sustainability

MDPI

Article

Adaptive Genetic Algorithm Based on Fuzzy Reasoning for the Multilevel Capacitated Lot-Sizing Problem with Energy Consumption in Synchronizer Production

Shuai Wang [1], Jizhuang Hui [1,*] and Bin Zhu [1] and Ying Liu [2]

[1] Institute of Intelligent Manufacturing Systems, Chang'an University, Xi'an 710064, China; 2019025004@chd.edu.cn (S.W.); binzhu@chd.edu.cn (B.Z.)
[2] Institute of Mechanical and Manufacturing Engineering, School of Engineering, Cardiff University, Cardiff CF24 3AA, UK; liuy81@cardiff.ac.uk
* Correspondence: huijz@chd.edu.cn

Abstract: The multilevel capacitated lot-sizing problem (MLCLSP) is a vital theoretical problem of production planning in discrete manufacturing. An improved algorithm based on the genetic algorithm (GA) is proposed to solve the MLCLSP. Based on the solution results, the distribution of energy consumption in a synchronous production case is analyzed. In the related literature, the GA has become a much-discussed topic in solving these kinds of problems. Although the standard GA can make up for the defects of the traditional algorithm, it will lead to the problems of unstable solution results and easy local convergence. For these reasons, this research presents an adaptive genetic algorithm based on fuzzy theory (fuzzy-GA) to solve the MLCLSP. Firstly, the solving process of the MLCLSP with the fuzzy-GA is described in detail, where algorithms for key technologies such as the capacity constraint algorithm and the algorithm of solving fitness value are developed. Secondly, the auto-encoding of decision variables for MLCLSPs is studied; within this, the decision variables of whether to produce or not are encoded into a hierarchical structure based on the bill of material; combined with external demand, the decision variables of lot-sizing are constructed. Thirdly, the adaptive optimization process of parameters of the GA for the MLCLSP based on fuzzy theory is expounded, in which membership function, fuzzy rule, and defuzzification of the MLCLSP is mainly presented. Experimental studies using the processed dataset collected from a synchronizer manufacturer have demonstrated the merits of the proposed approach, in which the energy consumption distribution of the optimized production plan is given. The optimal lot-sizing is closer to the average value of the optimal value compared with the standard GA, which indicates that the proposed fuzzy-GA approach has better convergence and stability.

Keywords: energy consumption; fuzzy-GA; lot-sizing; multilevel; capacitated

Citation: Wang, S.; Hui, J.; Zhu, B.; Liu, Y. Adaptive Genetic Algorithm Based on Fuzzy Reasoning for the Multilevel Capacitated Lot-Sizing Problem with Energy Consumption in Synchronizer Production. *Sustainability* **2022**, *14*, 5072. https://doi.org/10.3390/su14095072

Academic Editors: Marc A. Rosen, Fu Gu, Jingxiang Lv and Shun Jia

Received: 8 March 2022
Accepted: 22 April 2022
Published: 22 April 2022

1. Introduction

Production of automobile synchronizers is typical discrete manufacturing, which is characterized by a production process that can be decomposed into multiple tasks. Each task goes through a series of discontinuous processes to obtain semi-finished products, and finally the semi-finished products obtained from each task are assembled to obtain finished products. It can be seen from the characteristics of production that production planning belongs to multilevel planning. Against the background of information and intelligence of manufacturing, the optimization and automatic generation of production plans are beneficial to reduce production resource costs and improve production efficiency.

At present, the competition of the global manufacturing industry is becoming more and more intense [1]. Meanwhile, the energy issue has also become more prominent for the manufacturing industry [2]. Therefore, on the premise of ensuring quality, reducing costs is an important way to improve the competitiveness of enterprises. Reasonable operational planning is the core of reducing production costs [3,4] and can be divided into: workshop scheduling that is the core of manufacturing execution system, production lot-sizing planning implemented by enterprise resource planning, and various combinations of these [5]. In most specific production scenarios, these problems are NP-hard problems, which have been proved by previous studies [6,7]. Moreover, rising energy prices are driving manufacturing companies to reduce their production energy consumption [8]. The optimization of the production plan can also improve energy efficiency, which could slow the cost increase in products caused by rising energy prices.

In the presented article, we focus on the algorithm of the production lot-sizing problem. This problem can be defined as the problem of minimizing production total cost (including production cost, inventory cost, and setup cost caused by the immediate setup of products on the same production line), while also meeting the customer demand under limited capacity [9–12]. Depending on whether the planning of the lot size is affected by the production capacity of the production resources, the lot-sizing problem can be divided into two groups: uncapacitated and capacitated. Lot-sizing problems can be divided into single level and multilevel according to the structure of the bill of material (BOM), which consists of subassembly, pieces, raw material, and finished products, and is a structured material list that describes the assembly relationship of components [13–15]. Since the single-level uncapacitated lot-sizing problem, known as the Wagner–Whitin problem, was proposed, there have been single-level capacitated lot-sizing problems, multilevel uncapacitated lot-sizing problems, and multilevel capacitated lot-sizing problems. Of these, the MLCLSP is the most complex and presents the most problems. Moreover, the issues of introducing overtime, backorders, carryovers, etc., further extends the above lot-sizing problems. For a more comprehensive understanding of this topic, we refer interested readers to the survey in [6]. As the energy crisis is becoming a challenge, energy consumption in industrial production has attracted increasing attention, such as in [16,17]. Hence, the fact that the energy consumption distribution of production is analyzed in the case of synchronizer application production.

This study is motivated by the problem faced by an automobile gearbox supplier in the Chinese city of Xi'an. The supplier is a large-scale professional production enterprise that has been committed to developing, manufacturing, and selling several series of synchronizers, including dozens of specifications of products. The synchronizer workshop production system of this supplier consists of parts production lines, assembly production lines, and production lot-sizing planning, which is a typical MLCLSP. The mathematical model of the MLCLSP is a mixed integer programming problem. The existing literature has proved that the MLCLSP is NP hard; the traditional methods include the branch and bound method, the heuristic method, and the search algorithm. These methods usually either take too long or the accuracy of the solution is not high enough. Obviously, the supplier is unable to adjust the production planning in time. In recent years, many bionic algorithms have been used to help solve the production planning problem, such as the ant colony algorithm, neural network algorithm, fruit fly algorithm, genetic algorithm, and so on [18–23]. Among them, the genetic algorithm has good efficiency and accuracy. However, the stability of the GA in solving the lot-sizing problem is relatively poor: when the algorithm is run many times in the same scenario, the results differ.

Motivated by the above application background, we considered improving the stability of the GA using the fuzzy inference system, which has the ability to realize automatic reasoning according to the rule of experience [24]. The major contributions of this study are listed as follows: firstly, the genetic algorithm based on fuzzy theory is used to deal with the MLCLSP for the first time. Secondly, the key technologies of the improved GA with the lot-sizing problem are studied, including the encoding method, a variable number of multipoint genetic operation methods, etc. Thirdly, fuzzy theory is used to improve the adaption of the algorithm, and the adaptive parameters suitable for solving

MLCLSP are explored. Meanwhile, the distribution of energy consumption in the production process is also explored.

The remainder of this paper is organized as follows. Section 2 briefly reviews the existing literature. Section 3 demonstrates how the fuzzy-GA solves the MLCLSP, including the problem statement and algorithm design. The experiment setup including experiment data, the design hypothesis of experiments, and the performance evaluation metrics of the algorithm are described in Section 4. Then, Section 5 presents the results and discussion of the experiment. Section 6 concludes the paper.

2. Literature Review—Existing Approaches for the MLCLSP

2.1. Decomposition-Based Approaches

The decomposition-based approaches divide the problem into submodules and tend to be more flexible than other procedures with respect to distributed computing. One can distinguish between problem-based decomposition methods and solution space-based decomposition algorithms. The former decomposes the original problem into subproblems. The MLCLSP is decomposed into a sequence of capacitated lot-sizing problems (CLSP) in the literature [25], then the CLSP is solved by a modified Dixon–Silver heuristic method. The MLCLSP is considered to be a serial bill-of-material structure with item-specific resources. The authors of [26] treated the multilevel problem by constructing a surrogate single-level problem, which is then solved by a greedy heuristic. Solution space-based decomposition algorithms decompose the entire solution space into subspaces, such as the branch and bound algorithm, the branch and cut algorithm, and the branch and selection algorithm, all of which generate part-new disjoint subsets of the solution space during the "branching" part. The authors of [27] solve multi-item capacitated lot-sizing problems with setup times using branch-and-cut, which gives strong valid inequalities, and then they implement these inequalities successfully in a branch-and-cut algorithm. The authors of [28] combine a data-driven process with the branching and selection process, solving a CLSP. One major research trend in the field of the solving methods of MLCLSP is the construction of heuristic methods.

2.2. Traditional Heuristic Methods

In heuristic methods, only parts of the solution space are explored and there is an attempt to find a good, feasible solution in a reasonable time; its purpose is to meet the requirement of a certain level of accuracy and to reduce the solution time. The authors of [29] implement three heuristic methods to generate a plan for the MLCLSP with availability constraints; within these methods, binary variables and constraints are relaxed. A Lagrangian heuristic is used to relax a two-level CLSP into a continuous knapsack problem, which is easily solved using bounded variable linear programs [30]. A Lagrangian heuristic is developed to solve the MLCLSP with linked lot sizes in [31]. To achieve high-quality solutions in reasonable computational time, [32] deals with the MLCLSP with backlogging, in which the LugNP method is used to fix a subset of the binary variable vector to obtain quickly a feasible candidate solution by heuristic methods. The authors of [33] combine variable neighborhood decomposition search and accurate mixed-integer programming to solve the MLCLSP, which uses the exact LP/MIP solvers ILOG CPLEX, and then the variables can be fixed according to the neighborhood search rule. A fix-and-optimize approach is proposed for solving the MLCLSP, then a variable neighborhood search (VNS) approach further improves the solution in [34]. Rolling-horizon and fix-and-relax heuristics are used in the process of solving the MLCLSP with sequence dependence to overcome computational infeasibility in [35].

2.3. Bionic Algorithms

Bionic algorithms denote the process of simulating natural phenomena or laws to solve problems, such as the bees algorithm, GA, and ant colony optimization algorithm (ACO), etc. The authors of [11] propose a hybrid algorithm of the bees algorithm and fix-and-optimize algorithm, in which the bees algorithm is implemented to avoid local optima. The authors of [36] combined an ant colony optimization algorithm with a fix and relax heuristic for solving the MLCLSP, in which ACO uses virtual pheromone concentrations that have been built up during previous iterations to single out promising subsets for the next iteration. To solve the MLCLSP with backlogging in the [37], a hybrid multi-population GA (HMPGA) is proposed, which combines a multi-population based on a fix-and-optimize heuristic and mathematical programming techniques. The authors of [38] propose a GA-based heuristic for CLSP with sequence-dependent setups, in which the rolling horizon heuristic improves genetic algorithm to overcome computational infeasibility. A GA hybridized with VNS solves CLSP with setup times indicated in [39], where GA generates the solutions and VNS improves the quality of solutions. The ACO algorithm is more suitable for searching paths on the graph and thus more inclined to deal with problems that can be translated into path optimization. In solving MCLSP problems, the bees algorithm is more suitable to improve other algorithms to avoid falling into the local extremum. GA is more suitable as the main algorithm to solve batch problems, as shown in references [37–39].

2.4. A Brief Summary

For a more comprehensive understanding of approaches to solving the MLCLSP, interested readers are referred to the surveys in [6,40]. Decomposition-based approaches need to decouple the association between each sub-module in the implementation process, and finding a decomposition method for specific instances of MLCLSP is often difficult. The traditional heuristic methods are mainly based on heuristic rules, in which the search of solution space is one-sided and leads to feasible solutions. However, the quality of solutions is bad, and finding specific heuristic rules for problems, which are often difficult to construct, is necessary. Although bionic algorithms are also heuristic algorithms, the search has a specific evolutionary orientation and randomness. Most solutions of MLCSLP implemented by bionic algorithms have better qualities. Genetic algorithms have been widely studied in solving production planning and scheduling problems. Although GA worked quite effectively for the lot-sizing problem, the average value of multiple results is utilized as the effective result when the results of CLSP solved by a GA-based algorithm are processed in [38,39]. The reason for this is to reach a local solution. Thus, reducing the difference of each running result is meaningful to improve the stability of the solution. This study uses fuzzy theory to enhance the probability of the crossover and mutation of GA, the algorithm's adaptability, and the solution's stability.

3. The Proposed Fuzzy-GA for MLCLSP

This section mainly describes the method of the fuzzy-GA for solving the MLCLSP. Fuzzy-GA denotes an adaptive genetic algorithm, which adjusts the crossover probability and mutation probability of parameters in GA by fuzzy theory in this study. The MLCLSP is described with the production process of the synchronizer, and the mathematical model of the MLCLSP is given. Two key techniques in the algorithm are emphasized. On the one hand, the coding method of the proposed algorithm based on hierarchy structure is explained. On the other hand, the adaptive optimization process of parameters based on fuzzy theory is expounded.

3.1. Problem Statement

This section starts with an example describing the synchronizers' production process, a core component of the automobile gearbox, as shown in Figure 1. For the convenience of describing the problem, only critical parts of the synchronizer are shown. The blank of the hub is sent into a processing workshop, which is composed of multiple non-equivalent machines. After processing, the synchronizer body can be used for assembly and stored in the buffer. In the same process, the gear ring and sleeve to be assembled are put into the buffer. The synchronizer body, gear ring, and sleeve are assembled in a particular proportion on the assembly line, composed of multiple non-equivalent machines. The finished products are put into storage for handling by next-level producers.

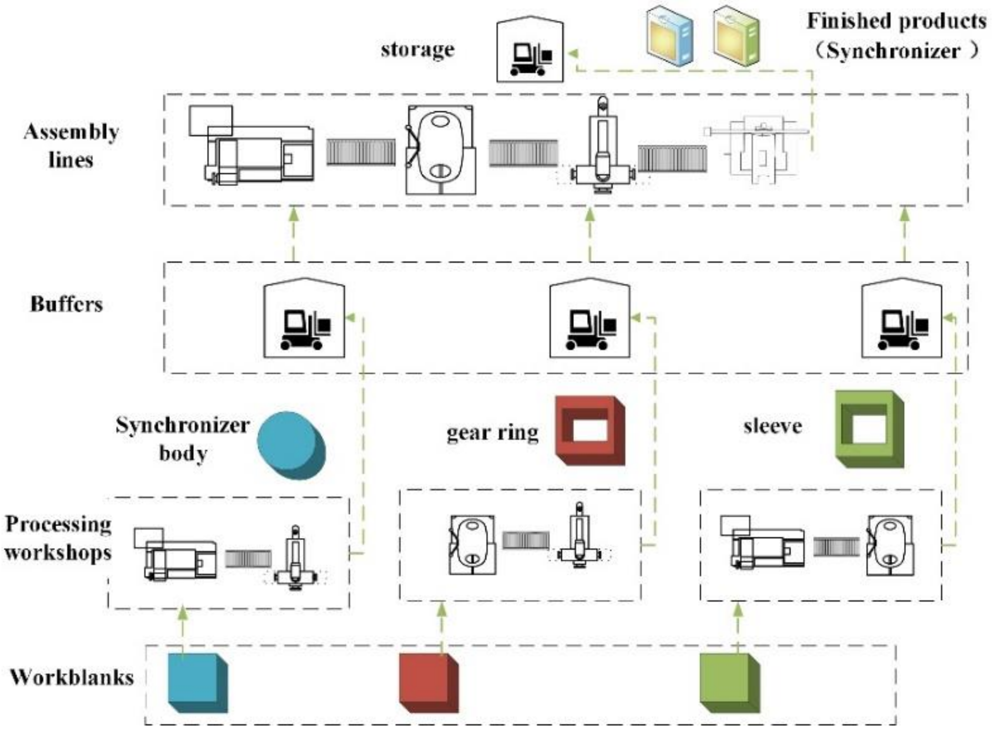


Figure 1. Example of synchronizer production.

In the production process of the synchronizers, the production plan for the next month should be made according to the demand, making the best weekly production lot-sizing of each type of synchronizer and corresponding components to balance inventory and setup costs. The lot-sizing of every product per week is affected by external order requirements, product dependencies in BOM, and equipment production capacity. This problem can be described as an MIP model. Parameters and decision variables used in the model are defined as follows.

By using the notation in Tables 1–3, the lot-sizing problem we considered in synchronizer production can be formulated as follows:

$$\text{Min } cost_{production} = \sum_{t=1}^{T}\sum_{i=1}^{n} c_{it}x_{it} + \sum_{t=1}^{T}\sum_{i=1}^{n} s_{it}Y_{it} + \sum_{t=1}^{T}\sum_{i=1}^{n} h_{it}I_{it} \quad (1)$$

s.t.

$$I_{i,t-1} + x_{it} - I_{it} \geq D_{it} + \sum_{j\in E(i)} R_{ij}x_{it}\ \forall i,t \quad (2)$$

$$\sum_{i=1}^{n}(ct_{kit}x_{it} + st_{kit}Y_{it}) \leq Ca_{kt}\forall k,t \quad (3)$$

$$I_{it}, x_{it} \geq 0\forall i,t \quad (4)$$

$$Y_{it} = \{0,1\} \forall i, t \tag{5}$$

Table 1. Indices and sets.

$t = 1,2, \ldots, T$:	Planning period
$i = 1,2, \ldots, n$:	Index for product
$k = 1,2, \ldots, M$:	Equipment number
$E(i)$:	Set of next-level products that need product i in the production process

Table 2. Parameters.

c_{it}:	Production cost unit product
s_{it}:	Setup cost per production of lot-sizing
h_{it}:	Inventory cost unit product
D_{it}:	External demand of product i in the planning period t
Ca_{kt}:	Production capacity of equipment resources k in the planning period t
R_{ij}:	The quantity of product i directly needed to produce one unit of product j (gozinto factor)
ct_{kit}:	The time cost of unit production of the product i in the equipment resource k during the planning period t
st_{kit}:	Time cost of production setup of the product i in the equipment resource k during the planning period t

Table 3. Decision variables.

x_{it}:	Output of product i during the planning period t
I_{it}:	Inventory of product i during the planning period t
Y_{it}:	Whether the product i is produced in the planning period t; $Y_{it} = 1$, if $x_{it} > 0$, $Y_{it} = 0$ otherwise.

3.2. Solving Model of MLCLSP with the Fuzzy-GA

When using the fuzzy-GA to deal with lot-sizing production problems, the general idea is to relax first. The capacity constraints are relaxed initially, the initial solution is obtained using the demand relationship, and the genetic operation is carried out. Then, the capacity constraint algorithm eliminates the solution that does not meet the constraints. The solution with the highest fitness is obtained on this basis and iterated several times until the optimization criteria are satisfied.

The MLCLSP solved in detail by the fuzzy-GA is given in Figure 2. First, the binary random variables Y_{it} are generated randomly, the initial lot-sizing is calculated, and Y_{it} stored as a hierarchical structure. Then, the capacity constraint algorithm is used to eliminate the individuals that do not meet the capacity constraint in the population, which is shown in Algorithm 1.

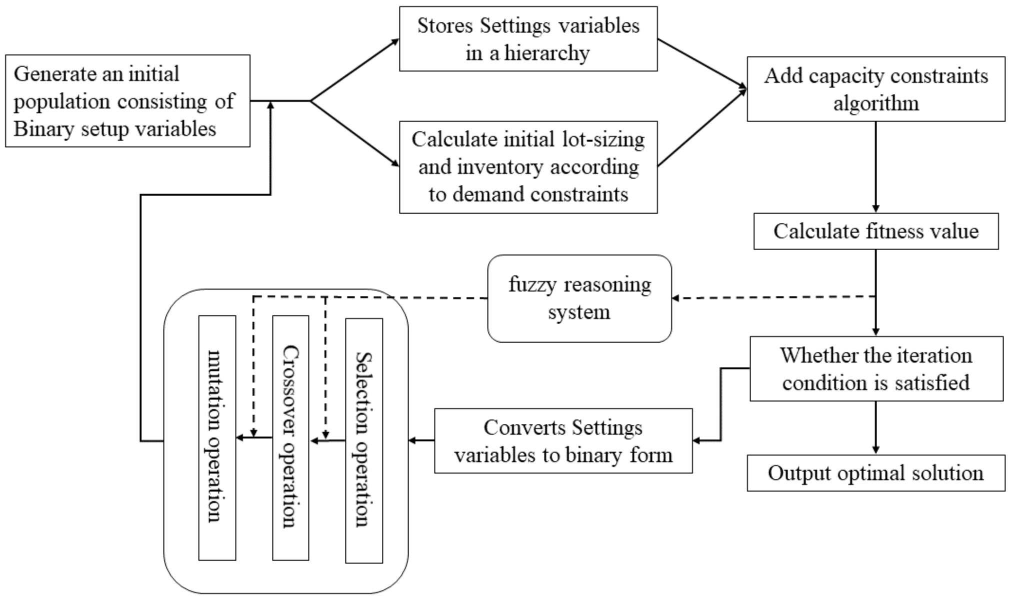

Generate an initial population consisting of Binary setup variables
Stores Settings variables in a hierarchy
Calculate initial lot-sizing and inventory according to demand constraints
Add capacity constraints algorithm
Calculate fitness value
fuzzy reasoning system
Whether the iteration condition is satisfied
Output optimal solution
Converts Settings variables to binary form
Selection operation
Crossover operation
mutation operation

Figure 2. Diagram of application of improved GA in lot-sizing production.

Algorithm 1. Capacity constraint algorithm.

```
Input: $x_{it}$, $Y_{it}$; output: $x_{it}$
1.  According to production lot-sizing without capacity constraints, the demand $Q$ of each equipment resource can be obtained.
2.  Using the result of Step 1 and the capacity $Ca$ of equipment resource, the difference between capacity and demand can be obtained $QC(k,t) = Q - Ca, k = 1,\dots,K, t = 1,\dots,T$.
3.  Let $t = T$.
4.  According to the situation of $QC(k,t) > 0$, obtain the ordered set $K_t(n_{kt})$ of all the production equipment that does not meet the capacity constraints according to the serial number in the production stage $t$. $n_{kt}$ represents the set of equipment numbers that does not meet the capacity constraint at period $t$.
5.  For $k_j \in K_t, j = 1,\dots,n_{kt}$.
5.1 If $j = 1$,
{production lot size $x_{it}$ of period $t$ is obtained without capacity constraints sort in ascending order $Xup_{k_1t}$. Then, find the minimum lot size $Xup_t(1)$.
Calculate $C_{rk_1} = |QC(k_1,t)| - a_{k_11t}Xup_{k_1t}(1), i = 1,$
Else calculate $C_{rk_j} = |QC(k_j,t)| - \left(\sum_{m=1}^{i-1} a_{k_jmt}Xup_t(m) + a_{k_jit}x_{it}\right)$
if $C_{rk_j} \leq 0$.
End this cycle and jump to Step 5 for the next cycle.
End
End
5.2 While $(C_{rk_j} > 0)$,
{the corresponding lot-sizing of $Xup_t\ (i)$ is transferred to the production lot-sizing of $t-1$. Set the corresponding binary variable $Y_{it}$ to zero. Cyclic variable $i = i + 1$. Calculate $C_{rk_j} = C_{rk_j} - a_{k_jit}Xup_{k_jt}(i)$}.
5.3 Calculate $x_{it} = {C_{rk_j} + a_{k_jit}Xup_{k_jt}(i)} \Big/ {a_{k_jit}}$ as the production lot-sizing to be migrated to $t-1$.
End
6.  $t = t - 1$.
If $t > 0$, jump to Step 4.
Else, update $Q$. If $QC = Q - Ca \leq 0$, output updated lot-sizing, setup variable as result, and end the whole procedure;
Else, the solution cannot satisfy the capacity constraint and end the whole procedure.
End
```

End

For individuals that satisfy capacity constraints, fitness value is calculated as follows: Using the variables, Y_{it}, x_{it}, and I_{it} correspond to those meeting the capacity constraints to solve the target cost. The maximum value of all target costs in the population is selected and divided by the individual target cost as the individual fitness value. Algorithm 2 provides details of the algorithm.

Algorithm 2. Algorithm of solving fitness value.

Input: production lot-sizing x_it, inventory I_it, and setup variable Y_it; output: fitness value: f

1. The individual production lot-sizing x_{it} and inventory I_{it} in the population are transformed into a hierarchical structure.
2. By substituting the result of Step 1 into the objective function formula, the corresponding objective value of individuals in the population can be obtained $obj(i), i = 1,2,\ldots,n$ (n represents the number of chromosomes in the population).
3. Get the maximum objective value $max\{obj(i)\}$ in the population.
4. Calculate the fitness value $f = max\{obj(i)\}/obj(i)$.

On the one hand, individuals with the largest fitness value were selected as the best solution and judged whether the iteration condition was satisfied. If it was satisfied, the algorithm ends and provides the result. If not, the algorithm continues to convert the decision variables of the hierarchy to single-layer binary strings hierarchical structure in preparation for genetic operations. On the other hand, the crossover and mutation probability was automatically adjusted according to fitness values combined with fuzzy theory. Finally, the roulette method is used for selection operation. The crossover and mutation operation adopts multipoint operation with the variable number of operation points determined by Formula (6).

$$n = \begin{cases} 1 & round(\alpha l_c)=0 \\ round(\alpha l_c) & round(\alpha l_c)>0' \end{cases} \tag{6}$$

where $round()$ means to round to an integer, l_c denotes the length of the chromosome to be interlaced or mutated, and α is the proportional coefficient of operation point of mutation or crossover operation, which is a constant.

When the proposed algorithm is used to solve MLCLSP problems, the length of chromosomes is affected by the size of the production problem instance. Formula (6) ensures that the effects of crossover and mutation operations on population can be fully reflected when dealing with lot-sizing production problems of different scales. The fuzzy inference system obtains the mutation and crossover probability automatically during each iteration. The process of mutation operation and crossover operation is shown in Figure 3.

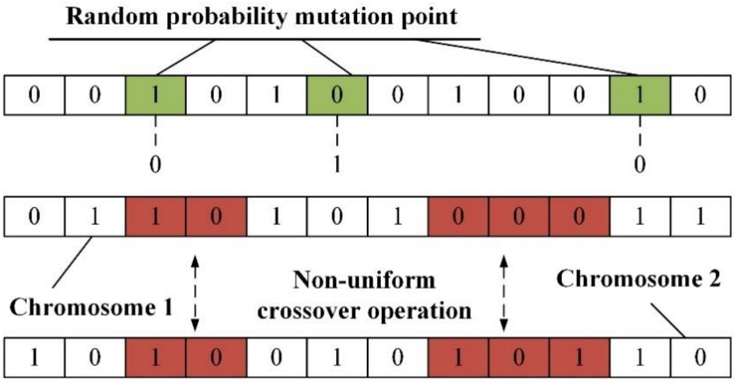


Figure 3. Schematic diagram of crossover and mutation.

Compared with the traditional GA, the fuzzy-GA first uses the fuzzy reasoning system to calculate the crossover probability and mutation probability of an individual according to the fitness value in each iteration cycle before carrying out the genetic operation to realize the automatic adjustment of crossover and mutation probability in the genetic operation of each iteration.

3.3. Encoder Based on Hierarchical Structure

According to the planning period and product type, the initial population with size n is randomly generated, and the chromosome in the population is the binary string with length $i \times t$ corresponding to the setup variable Y_{it}. This binary string indicates whether the product i is produced during the planning period t. The specific relationship is shown in Figure 4. Y_{ij} represents the product j at the BOM structure level i. Individuals in the population are arranged in sequence according to the plan period and then from top to bottom according to the BOM structure to form a binary string.

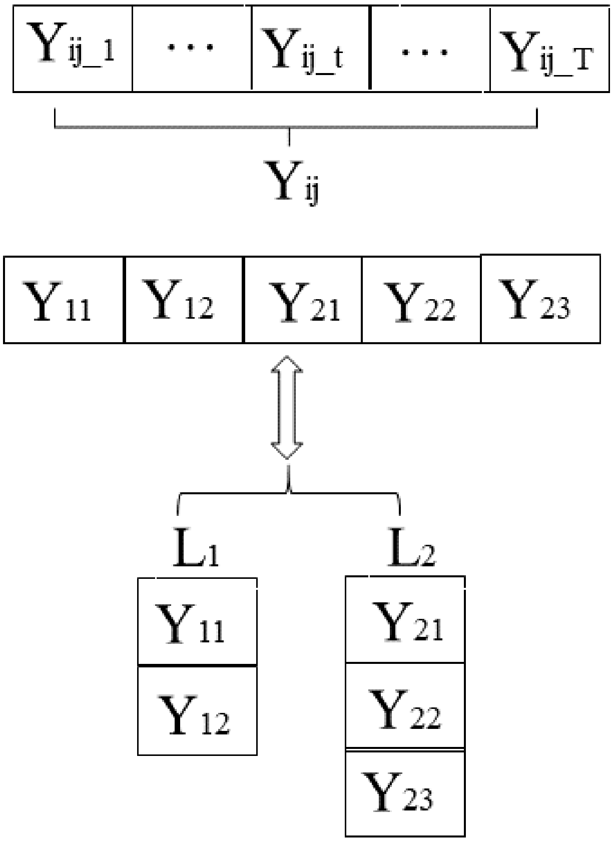


Figure 4. Relational graph of encoding and multilevel structure.

The chromosome generated by 1 is stored as a hierarchical structure Y_{it} according to the product BOM. The conversion relationship of them is shown in Figure 4. The data of whether the product is ready for production in each planning period in the BOM are stored in each level. Taking Y_{it}, quantity dependence R_{ij}, and external demand D_{it} as input variables, the lot-sizing algorithm constructed by demand relation (refer to Algorithm 3) is used to calculate the production decision variables x_{it}.

Algorithm 3. Algorithm of a lot-sizing solution according to the demand constraints.

```
Input: setup variable, product demand, and material quantity relationship $Y_{it}$, $D_{it}$, $R_{ij}$; output: production lot-sizing $x_{it}$
1. Get the number of layers L of the material structure by $Y_{it}$ structure.
2. Find the planning period $t$ of product $i$ corresponding to $Y_{it} = 1$, in the multilevel structure. Store the data from small to large in vector $T_i(j)$.
3. If the number of layers of product $i$ is equal to 1,
   Use demand balance relation $x_{it}\left(1, i, T_i(j)\right) = \sum_{t=T_i(j)}^{T_i(j+1)-1} D_{it}(i,t)$ and solve production lot-sizing $X_{it}$ of product$i$ at different planning periods.
   Else,
   use demand balance relation $x_{it}\left(1, i, T_i(j)\right) = \sum_{t=T_i(j)}^{T_i(j+1)-1} D_{it}(i,t) + \sum_{l_i=1}^{l-1} R_{ij} X_{it}\left(l_i, i, T_i(j)\right)$ and solve production lot-sizing $x_{it}$.
   End
4. $i = i + 1$. If $i$ is less than or equal to the total number of products, jump to Step 3; otherwise, end the program.
```

3.4. Adaptive Optimization Process of Parameters Based on Fuzzy Theory

The convergence and stability of the solution of traditional GAs are mainly affected by the crossover probability P_c and mutation probability P_m. P_c and P_m are fixed before the traditional GA runs, but they cannot be changed during the running process. During the algorithm's operation, the same mutation and crossover rates are used regardless of the fitness value of individuals in the population. However, from the characteristics of the genetic algorithm, in each genetic operation, for individuals with high fitness, appropriately reducing the crossover rate and mutation rate is conducive to improving the convergence speed of the algorithm. For individuals with low fitness, appropriately increasing the value of the two is conducive to the global search of the solution, reducing the possibility of the solution converging to the local optimum. Aiming at the problem that the GA is easy to use to obtain an optimal local solution and unstable in solving lot-sizing production problems, this section proposes to apply a fuzzy reasoning system to control P_c and P_m. In each genetic operation, the fuzzy inference system is used to infer different crossover and mutation rates for individuals with different adaptive values to improve the global search ability and convergence speed of the algorithm.

Figure 5 shows the schematic diagram of the fuzzy reasoning system. In each iteration of the proposed algorithm, the maximum fitness value of the chromosome in the population is different, leading to the uncertainty of the input value range during the fuzzy reasoning process. Differences in the input value ranges of fuzzy inference systems correspond to different membership functions. To ensure that the input of the fuzzy reasoning process in each iteration has the same domain and is easy to be realized by programming, scale transformation mapping of input variables is needed. Therefore, when fuzzy reasoning is carried out in each iteration, combined with the characteristics of fitness value in each iteration, the following scale transformation formula is performed:

$$f: obj'_{value} = \frac{obj_{value}}{max\{obj_{value}\}}, \tag{7}$$

where obj_{value} represents the fitness value corresponding to the chromosome in the population of each iteration, $max\{obj_{value}\}$ represents the maximum of all fitness values in the population, and $obj'_{value} \in [0,1]$ represents the output value after transformation. The same range of input variables is ensured in each reasoning process; they are all in the domain of [0, 1].

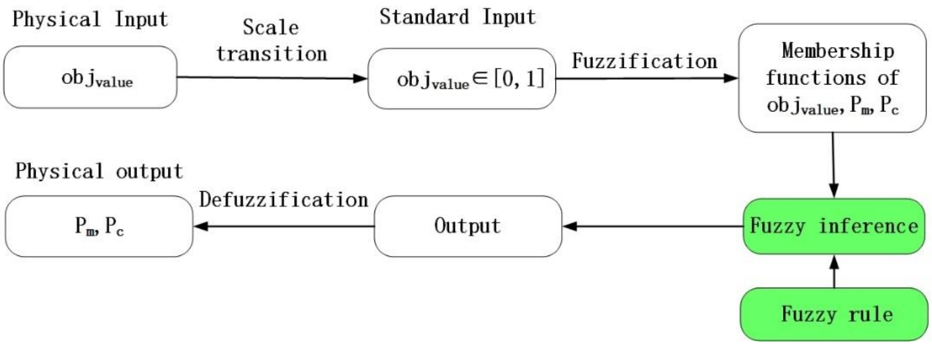


Figure 5. Diagram of fuzzy reasoning system.

In the second step, the variables of input and output are fuzzified into fuzzy language variables by fuzzy set, which uses the mapping of membership function to describe the variables so that the same variable can belong to multiple sets but have different degrees of membership. Fuzzification can be divided into fuzzy division and membership function determination. Fuzzy partition determines the number of fuzzy sets on the domain. The higher the number of sets is, the more precise the association between fuzzy sets and precise quantities is. However, fine division quickly increases the amount of calculation. According to input and output characteristics, three fuzzy sets are divided in the input and output domain, namely, PS, PM, and PB, which indicate that the value is small, moderate, and large, respectively. The membership functions of all fuzzy sets are Gaussian functions commonly used in fuzzy reasoning, such as Formula (8), but different parameters are selected.

$$mf(x,\ c_1,\ c_2) = e^{(x-\ c_2)^2/\ {c_1}^2}, \tag{8}$$

where x is a variable in the domain, and parameters c_1 and c_2 determine the rapid and slow change and the central position of the function curve, respectively.

In Formula (8), $c_1 = 0.2$ and $c_2 = 0$ are taken to obtain the small positive PS of the fuzzy language variable of input obj'_{value}, which is represented by $\mathrm{Imf_s}$. Similarly, $c_1 = 0.2$ and $c_2 = 0.5$ can be substituted into Formula (8) to obtain the membership function $\mathrm{Imf_m}$ corresponding to the positive middle PM of the input fuzzy language variable. The membership function $\mathrm{Imf_b}$ of the fuzzy variable PM of the input can be obtained by making $c_1 = 0.2$ and $c_2 = 1$ in Formula (8).

In the proposed algorithm, the mutation probability value is small. Thus, in the fuzzy reasoning process, the range of mutation probability P_m is [0, 0.003], [0, 0.03], or [0, 0.3]. The optimal range suitable for this problem will be determined by a trial method in Experiment 2 of the paper. Then, taking different c_1 and c_2 values for Formula (8) in the experiment, we can obtain the membership function $\mathrm{0mf_{mb}}$, $\mathrm{0mf_{mm}}$, and $\mathrm{0mf_{ms}}$ corresponding to fuzzy language variables PB, Pm, and PS of output P_m, respectively.

The domain of output crossover probability P_c in fuzzy inference system is [0, 1]. In Formula (8), $c_1 = 0.2$. In Experiment 2, the c_2 values of membership functions $\mathrm{0mf_{cb}}$, $\mathrm{0mf_{cm}}$ and $\mathrm{0mf_{cs}}$ corresponding to fuzzy language PB, and PM and PS for P_c are discussed.

Then, according to the fuzzy rules, the fuzzy input can obtain the fuzzy output through fuzzy reasoning. Finally, after defuzzification, the output can give the actual crossover probability and mutation probability output. The primary purpose of defuzzification is to transform the fuzzy output from fuzzy reasoning into the available actual production. In the algorithm, the centroid method is used as the defuzzification method, as shown in Formula (9):

$$p = \frac{\int_a^b z\mu_y(z)\,dz}{\int_a^b \mu_y(z)\,dz}, \tag{9}$$

where $[a, b]$ is the domain of output mutation or crossover probability. $\mu_y(z)$ is membership function of fuzzy sets in the domain. p represents the output after defuzzification.

In the implementation process, fuzzy inference and the establishment of fuzzy rules are the core steps, which are the green boxes shown in Figure 5.

Fuzzy rules are expert experiences or known rules expressed by fuzzy language variables. Through observation, in the iteration process of the algorithm, when the fitness value obj'_{value}, is small, slightly increasing the mutation rate P_m or increasing crossover probability P_c can improve the global search ability of the algorithm. When the fitness value is large, appropriately decreasing mutation rate P_m or crossover probability P_c can improve the convergence speed of the algorithm and improve the stability of the solution. The rules expressed by if−then are as follows:

If obj'_{value} is PS, then P_m and P_c are PB.
If obj'_{value} is PM, then P_m and P_c are PM.
If obj'_{value} is PB, then P_m and P_c are PS.

A fuzzy control table can be obtained according to the above empirical rules, as shown in Table 4.

Table 4. Rules of fuzzy inference.

IF	PS (obj'_{value})	PM (obj'_{value})	PB (obj'_{value})
THEN	PB (P_m)	PM (P_m)	PS (P_m)
THEN	PB (P_C)	PM (P_C)	PS (P_C)

Fuzzy reasoning is the process of obtaining the fuzzy output of the system by using fuzzy logic operations, such as implication and synthesis according to the fuzzy rules. The fuzzy relation model R_m and R_c of output P_m and P_c that is based on the max−min composition rule can be expressed as follows:

$$\begin{cases} R_m = (\mathrm{Imf_s} \cap \mathrm{Omf_{mb}}) \cup (\mathrm{Imf_m} \cap \mathrm{Omf_{mm}}) \cup (\mathrm{Imf_b} \cap \mathrm{Omf_{ms}}) \\ R_c = (\mathrm{Imf_s} \cap \mathrm{Omf_{cb}}) \cup (\mathrm{Imf_m} \cap \mathrm{Omf_{cm}}) \cup (\mathrm{Imf_b} \cap \mathrm{Omf_{cs}}) \end{cases}, \tag{10}$$

where intersection ∩ represents the smaller value of two operands and union ∪ indicates finding the maximum value of the two operands.

Formula 10 is established with fuzzy rules in Table 4. The operation within rules takes intersection, and the operation between rules takes union. Fuzzy decision-making is performed by the fuzzy relation model, which determines the output of the fuzzy system by combining the input of the fuzzy system with the fuzzy reasoning model. As shown in Formula (11),

$$\begin{cases} \mathrm{Omf}(P_m) = \cup_{obj'_{value}} [\mathrm{Imf}(obj'_{value}) \cap R_m] \\ \mathrm{Omf}(P_c) = \cup_{obj'_{value}} [\mathrm{Imf}(obj'_{value}) \cap R_c] \end{cases}, \tag{11}$$

where $\mathrm{Omf}(P_m)$ and $\mathrm{Omf}(P_c)$ denote the membership function of output and $\mathrm{Imf}(obj'_{value})$ denotes the membership function of the input.

4. Experimental Setup

4.1. Experimental Data and Design Hypothesis

This study designs three experiments to verify the algorithm's effectiveness, find the optimal parameters of the fuzzy reasoning algorithm, and illustrate the comprehensive performance of the algorithm. The experimental results are obtained in the environment of i5-7200U 2.71 GHz CPU and 4.00 GB memory, running MATLAB programming software version R2014b and the commercial software lingo 11.0.0.20.

4.1.1. Experiment 1 Validation of Fuzzy-GA in Solving MLCLSP

A two-level BOM material structure is constructed based on an actual production line of a synchronizer. The first level is composed of products 1 and 2. The second level is composed of products 3, 4, and 5. The proposed algorithm and the commercial software lingo were compared to verify the solution results. The production costs obtained from both methods are calculated to verify the algorithm's effectiveness. Table 5 shows the production demand of product $i(i = 1,\dots,5)$ in planning period $t(t = 1,2,3,4)$.

Table 5. Product demand D_{it}.

Product Number *i*\Planning Period *t*	**1**	**2**	**3**	**4**
1	28	14	9	32
2	6	1	1	17
3	1	9	4	11
4	17	13	15	17
5	18	2	4	10

Table 6 shows the production capacity of the equipment $k(k = 1,\dots,9)$ in each planning t period. Table 7 lists the quantity relationship between the secondary products and the primary products in the BOM.

Table 6. Capacity of production devices Ca_{kt}.

Equipment Number k \Planning Period t	**1**	**2**	**3**	**4**
1	5666	5483	5427	5539
2	5332	5437	5213	5147
3	5189	5088	5711	5225
4	5891	5676	5367	5148
5	5835	5274	5249	5982
6	5442	5948	5618	5596
7	5694	5944	5902	5961
8	5498	5342	5072	5554
9	5625	5717	5492	5336

Table 7. Product assembly ratio R_{ij} in BOM.

Level 2\Level 1 Product Number	**1**	**2**
3	2	1
4	0	2
5	1	1

The production setup time and the unit production time of product i on the equipment k at period t are shown in Table 8. Other parameters of the example, unit production cost, inventory cost, production setup cost, and initial inventory are given in Table 9.

Table 8. Time cost $st_{kit}\backslash ct_{kit}$ of setup\production per unit.

Equipment	**Product**	**Planning Period *t***			
Number *k*	**Number *i***	**1**	**2**	**3**	**4**
1	1	34\2	31\4	32\3	30\8
	2	32\9	33\7	32\6	35\9
	3	31\3	33\8	33\9	34\9
	4	31\4	33\5	31\6	32\8
	5	33\7	35\7	32\4	30\3
2	1	35\8	31\8	31\4	30\2
	2	30\2	33\5	33\5	33\6
	3	32\2	31\5	35\5	33\6
	4	34\4	34\2	35\2	35\9
	5	32\3	31\7	35\4	33\7
3	1	33\7	35\4	34\7	30\3
	2	34\3	30\6	30\2	32\4
	3	31\3	31\8	35\9	32\4
	4	30\7	34\4	31\5	31\2
	5	33\3	30\7	33\3	33\9
4	1	32\5	32\6	32\7	33\7
	2	34\3	30\4	30\6	30\7
	3	33\9	31\3	32\7	33\3
	4	33\3	33\9	30\9	34\7
	5	35\6	33\8	33\8	31\8
5	1	33\4	32\6	32\6	34\8
	2	33\8	31\2	32\2	33\5
	3	34\7	31\5	34\8	31\5
	4	31\4	33\4	34\5	31\9
	5	30\2	33\5	33\5	30\8
6	1	31\6	32\8	33\8	34\7
	2	31\6	34\5	31\4	31\4
	3	31\9	33\8	33\3	34\8
	4	31\3	31\9	33\3	34\3
	5	35\3	34\8	32\4	31\8
7	1	31\4	32\5	31\2	30\9
	2	34\5	33\3	32\3	32\5
	3	34\6	33\9	32\4	33\8
	4	32\3	32\3	35\4	34\3
	5	33\8	31\2	31\3	31\7
8	1	34\5	34\6	31\3	31\9
	2	33\8	31\7	32\7	33\6
	3	30\7	33\7	35\8	32\6
	4	30\2	32\6	32\6	31\2
	5	30\7	33\3	32\7	33\6
9	1	30\8	32\4	30\8	35\2
	2	30\5	34\3	32\4	35\7
	3	35\7	32\7	34\7	32\3
	4	33\7	34\5	31\3	32\5
	5	30\2	32\7	35\5	35\9

Table 9. Other model parameters.

Parameters\ Production Number	**1**	**2**	**3**	**4**	**5**
Production cost c_{it}	39	36	31	37	35
Inventory cost h_{it}	59	48	32	53	40
Setup cost s_{it}	123	110	117	104	127
Initial inventory I_{i0}	5	18	18	1	22
Energy costs	1.5	1.2	2	2.2	1.8

4.1.2. Experiment 2 Adaptive Parameters of the Fuzzy-GA

In this experiment, the dataset from Experiment 1 is still used. The central parameters c_2 of the output membership function greatly influence the algorithm performance in the fuzzy reasoning system. Thus, the parameters c_2 suitable for crossover probability P_c and mutation probability P_m are determined in the process of solving the MLCLSP with the proposed algorithm. Multiple sets of c_2 values are used to solve the problem. By comparing the results, the better operation parameters of the algorithm are given.

Table 10 shows the specific values of the central control parameters c_2 of the membership functions in the language variable set corresponding to the mutation probability in the fuzzy reasoning system in three cases, P_{m1}, P_{m2}, and P_{m3}, which represent parameters of different orders of magnitude.

Table 10. Parameters of mutation with different magnitudes.

Fuzzy Sets\Magnitude	$\boldsymbol{P_{m1}}$	$\boldsymbol{P_{m2}}$	$\boldsymbol{P_{m3}}$
PS	0.01	0.1	0.001
PM	0.02	0.2	0.002
PB	0.03	0.3	0.003

Table 11 shows the specific values of the central control parameters c_2 of each membership function in the language variable set corresponding to the crossover probability in the fuzzy reasoning system under three cases P_{c1}, P_{c2}, P_{c3}. A PM equal to 0.7 means that the chromosome crossover probability of the intermediate fitness value tends to be large in each iteration population. A PM equal to 0.3 means that the chromosome crossover probability of the intermediate fitness value tends to be small in each iteration population. A PM equal to 0.5 is between the two.

Table 11. Parameters of crossover with different magnitude.

Fuzzy Sets\Magnitude	$\boldsymbol{P_{c1}}$	$\boldsymbol{P_{c2}}$	$\boldsymbol{P_{c3}}$
PS	0.1	0.1	0.1
PM	0.7	0.3	0.5
PB	0.9	0.9	0.9

4.1.3. Experiment 3 Performance of Fuzzy-GA in Solving MLCLSP

On the basis of the production examples of different scales, the MLCLSP was solved by GA and the proposed algorithm, and the superiority of the proposed algorithm was verified. This experiment mainly verifies whether the algorithm's speed, quality, and stability are improved for the proposed algorithm in solving the MLCLSP. The response speed of the improved algorithm is verified by comparing the solution time. The comparison of the best solution is used to verify the quality of the solution. The difference between the average and the best solution is used to verify the stability of the solution.

Computational experiments were conducted on three groups of test instances (i.e., N(5,10,20)-T5-M9, N20-T(5,7,10)-M9, and N20-T10-M(9,15,20), representing different problem scales, where the numbers represent products, periods, and machines, respectively). These groups are similar to the data structure of Experiment 1. N(5,10,20) means N5, N10, and N20. Thus, each combination has three instances, producing seven instances in total. According to the synchronizer generation process, all instances have a two-level BOM structure. Although the real-life data are unavailable because of confidentiality, the setting of those instances was simulated from the practice in a production line of a synchronizer assembly manufacturing company in China.

4.2. Performance Evaluation Metrics

Different performance characteristics require different methods to measure. This study mainly examined the effectiveness, optimal parameter selection, and stability of the algorithm. Thus, the corresponding measurement methods for these performance characteristics are given below.

The proposed algorithm is a random algorithm, which has a certain difference in the results of each solution for the same instance. The optimal total cost of the MLCLSP solved by the proposed algorithm is the expected value of n calculations as follows:

$$E(f) = \frac{\sum_{i=0}^{n} f_i}{n} \tag{12}$$

Assuming that the optimal cost of the commercial software lingo solution is denoted as f_l, then the effectiveness of the proposed algorithm can be measured by Formula (13).

$$e_1 = \frac{|E(f) - f_l|}{E(f)} \tag{13}$$

Adjusting the mutation probability can improve the defect that easily falls into the optimal local solution for the GA. Thus, the selection of the central parameters c_2 of the output membership function of mutation rate takes the minimum value of the optimal cost under different parameters as the measurement index. Assuming that $f(P_{mi})$ is the optimal cost of the proposed algorithm in the case of $P_{mi}(i = 1, 2, 3)$, the measurement index can be written as follows:

$$e_2 = min\{E\big(f(P_{mi})\big)\} \tag{14}$$

Adjusting the crossover probability can improve the stability of the solution, which means the amplitude of the target value will weaken as the number of iterations increases. Thus, the central parameters c_2 of the output membership function of crossover rate with the highest density of curves is selected as the best parameter, according to the different densities of curves in the figure of optimization target changing with the number of iterations under the case of different $P_{ci}(i = 1, 2, 3)$.

In Experiment 3, the optimal value of different scales takes the minimum objective value of multiple solutions as Formula (15). The average optimal value is the expected value of various solutions, which can be expressed by Formula (12).

$$f_{ov} = min\{f_1, f_2, \ldots, f_n\} \tag{15}$$

The stability of the proposed algorithm can be expressed by the difference between the average value and the best value as Formula (16).

$$e_3 = |E(f) - f_{ov}| \tag{16}$$

Assuming that elapsed time of the proposed algorithm denotes t_i, the proposed algorithm's computational speed is measured by the average speed of multiple algorithm solutions, as shown in Formula (17).

$$t_{run} = \frac{\sum_{i=0}^{n} t_i}{n} \tag{17}$$

5. Results and Discussion

5.1. Results and Discussion for Experiment 1

The number of iterations is 100, and the population size is 30. The proportional coefficient of mutation operation $\alpha_m = 0.001$, the proportional coefficient of crossover operation $\alpha_c = 0.01$, and P_{m1} and P_{c2} in the fuzzy reasoning system are selected. Using the proposed algorithm, the optimal value of comprehensive cost is 18,320 by running the example, and the corresponding lot-sizing production results are shown in Table 12.

Table 12. Production lot-size x_{it} under optimal objective value.

Product Number i \Planning Period t	**1**	**2**	**3**	**4**
1	28	14	9	32
2	7	0	1	17
3	64	37	23	92
4	31	13	17	51
5	53	16	14	59

Verifying that the production lot size x_{it} can meet external demand in Table 5, and capacity requirements in Table 6 are easy, indicating the algorithm's feasibility. The commercial soft lingo is used for solving, and the optimal value is 18,203. The difference is 150, and the relative error is 0.639% between the two, which shows the proposed algorithm's effectiveness.

As shown in Figure 6, the legend number corresponds to the product number. The abscissa is the production planning period. The ordinate is the product production quantity, which shows the production quantity change rule of various products in the same planning period. Figure 7 reflects the proportion distribution of energy consumption of different products in the total energy consumption in each period, where the different colors are consistent with Figure 6. Combining Figures 6 and 7, the distributions of energy consumption are consistent with the size of product lot-sizing but show some differences, such as products 4 and 5 in the fourth period, mainly caused by the difference in energy consumption costs between them.

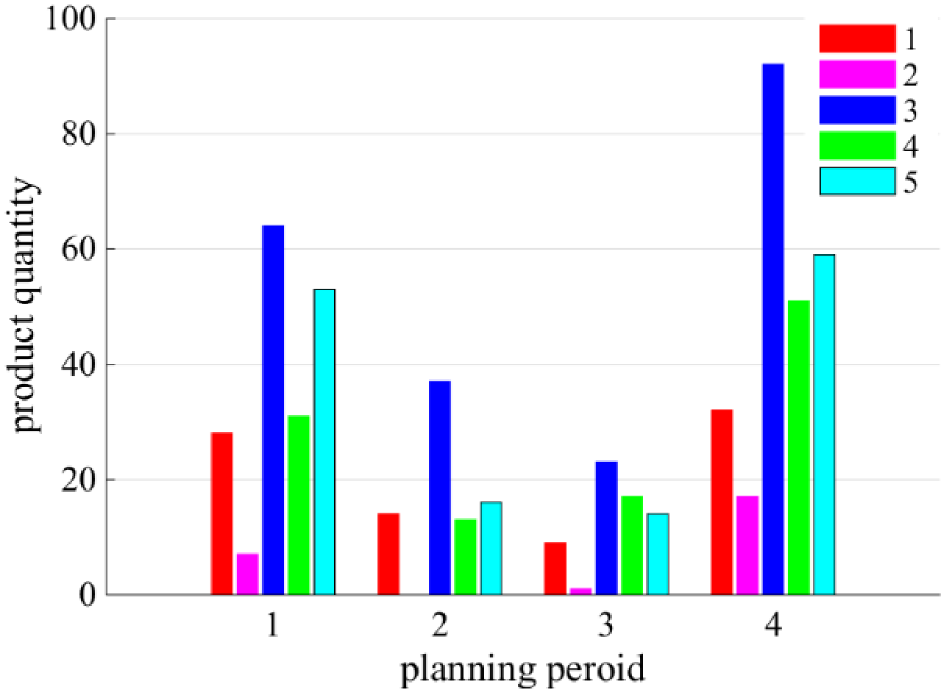
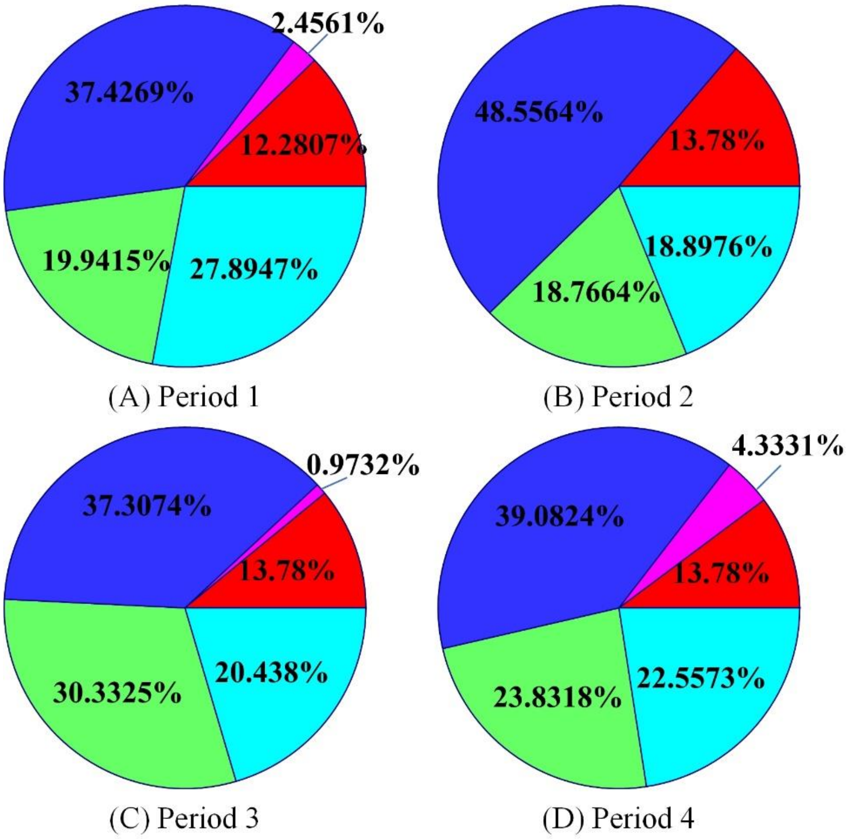


Figure 6. Column diagram of production quantity of different planning periods.



(A) Period 1

(B) Period 2

(C) Period 3

(D) Period 4

Figure 7. Energy costs of all products for: (**A**) planning period 1; (**B**) planning period 2; (**C**) planning period 3; (**D**) planning period 4.

The legend number in Figure 8 corresponds to the production planning period, the abscissa is the product type number, and the ordinate is the product production quantity. Figure 8 describes the production quantity of the same product in a different production period. Through the classified display of the two results, the lot-sizing distribution of different products in each planning period can be displayed clearly. Figure 9 shows the proportion of energy consumption of each period in the total energy consumption for various products, in which the different colors are consistent with Figure 8. Comparing Figures 8 and 9, different products have a large proportion of energy consumption in period 4. Thus, optimizing energy consumption can focus on the planning of period 4.

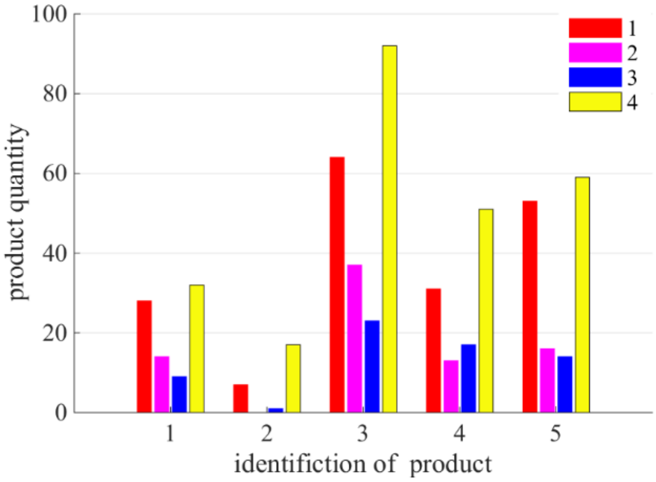


Figure 8. Quantity column diagram of different productions.

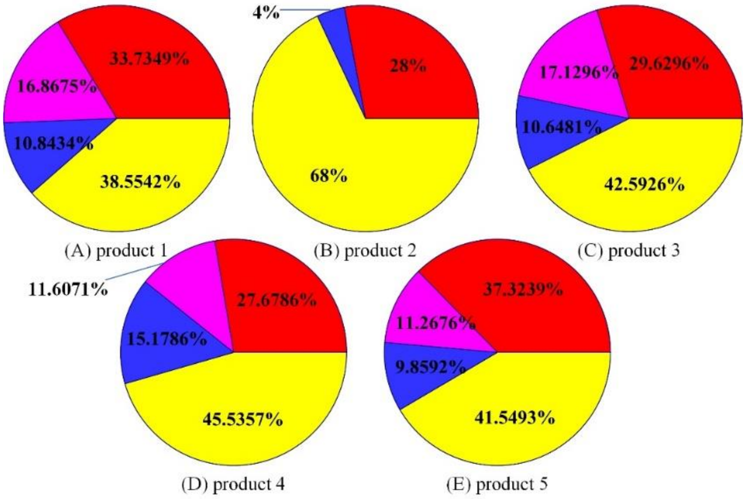


Figure 9. Energy costs of synchronizers for: (**A**) product 1; (**B**) product 2; (**C**) product 3; (**D**) product 4; (**E**) product 5 in the planning horizon.

5.2. Results and Discussion for Experiment 2

The results in Figure 10 are randomly run five times and are drawn by taking the minimum target value in each case. Three smooth curves are the fitting results of three actual curves in the figure, obtained using a six-order polynomial function. The fitting curve $P_{m3} - fitting$ shows that the variation trend of the solution tends to be flat and the worst effect of the optimal solution is when the magnitude of the mutation parameter is too small. When the magnitude of the mutation parameter is the largest, as shown in the curve $P_{m2} - fitting$, the change of the target value has a sharp convergence with the number of iterations, and the overall target value tends to decrease. However, the final convergence result is poorer than curve $P_{m1} - fitting$. When the mutation parameter takes the middle value, according to the curve $P_{m1} - fitting$, the intensity of the overall trend of the target value is slightly worse than that of curve $P_{m2} - fitting$, but the effect of the optimal solution is the best of the three. Thus, when the proposed fuzzy-GA is used to solve a lot-sizing problem, P_{m1} groups of parameters can be selected.

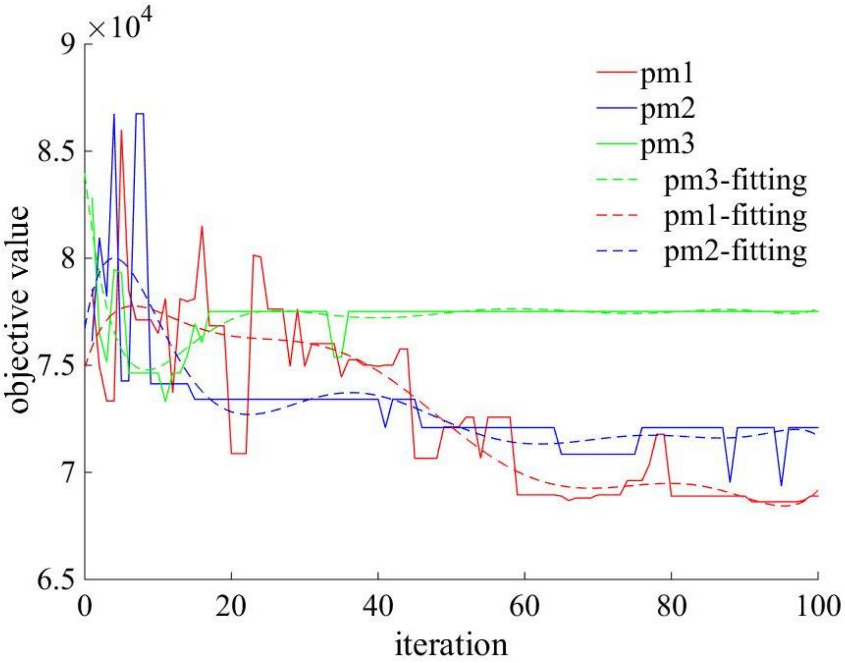


Figure 10. Comparison of results with different parameters of mutation.

Figures 11–13 show the parameters in the case of P_{c1}, P_{c2}, and P_{c3}. When the fuzzy-GA is run many times, the target value changes with the number of iterations. Compared with P_{c3}, the trend of the curves P_{c1} and P_{c2} between the target value and the number of iterations is better. Although the optimal results are close to each other with different parameters, the curves are more concentrated in the case of P_{c2}, P_{c1}, and P_{c3} and are relatively scattered. Therefore, the fuzzy-GA algorithm has better stability for lot-sizing problem solving under the parameter of P_{c2}.

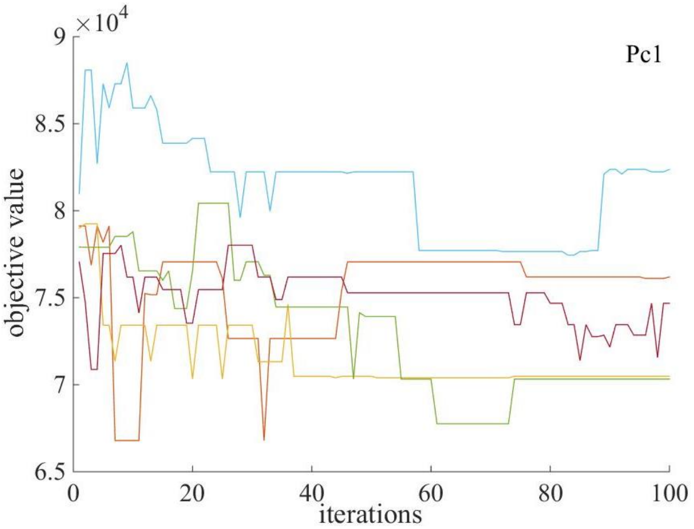


Figure 11. Results with parameters P_{c1} of crossover.

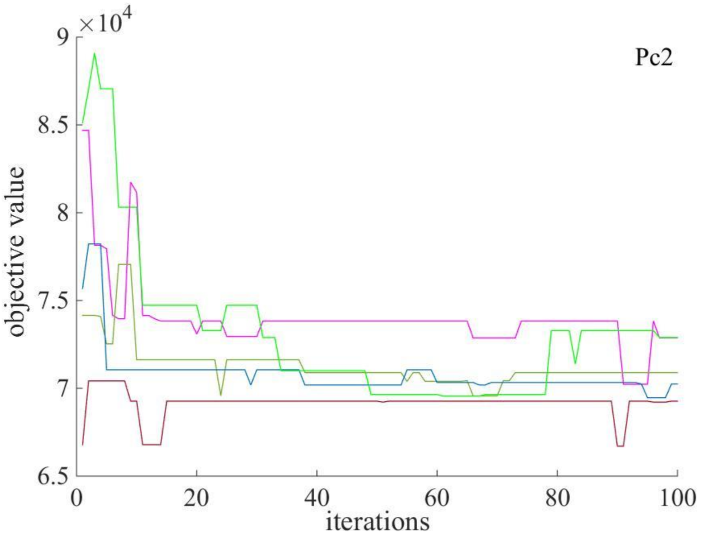


Figure 12. Results with parameters P_{c2} of crossover.

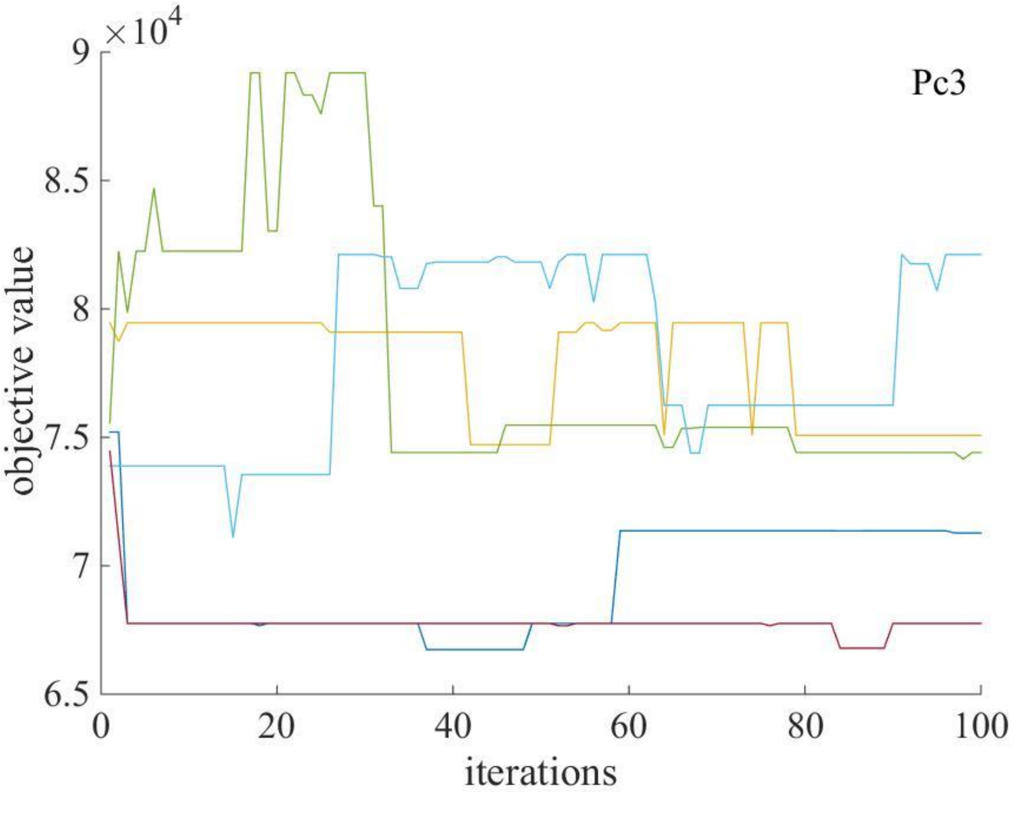


Figure 13. Results with parameters P_{c2} of crossover.

5.2. Results and Discussion for Experiment 3

In this experiment, the parameter of the mutation probability membership function adopts P_{m1}, and the parameter of the crossover probability membership function adopts P_{c2}. Table 13 shows the results of solving the MLCLSP by the general GA and the fuzzy-GA, which includes different production scales indicated by product type, planning period, and the number of devices.

Table 13. Comparison of results between GA and fuzzy-GA.

			Traditional GA				Fuzzy-GA			
Product Type N	**Planning Period T**	**Number of Devices M**	**Optimal Value**	**Average Optimal Value**	**Iterations**	**Running time(s)**	**Optimal Value**	**Average Optimal Value**	**Iterations**	**Running Time(s)**
5	5	9	90,101	94,619	95	4.5174	87,629	90,839	104	4.3237
10	5	9	256,618	263,514	111	5.7030	242,461	246,960	105	5.5393
20	5	9	999,289	1,061,107	94	7.0897	908,033	944,982	97	7.6957
20	7	9	1,406,511	1,457,795	107	9.8041	1,349,712	1,378,720	93	9.2876
20	10	9	1,950,637	2,115,856	100	12.7491	2,005,225	2,044,097	90	11.3101
20	10	15	1,985,502	2,020,672	95	13.3185	1,888,792	1,905,142	85	12.2024
20	10	20	2,515,585	2,596,667	112	16.4103	2,410,076	2,440,758	97	15.5662

The iterations in Table 13 represent the average number of iterations for 10 runs of the algorithm. Iterations are the number of iterations based on the criterion that the optimization objective has not been optimized for 50 consecutive times during the algorithm's operation. The average optimal value and running time in Table 6 are the average value of the algorithm running 10 times. The optimal value is the minimum of 10 running results. The smaller the target value, the better the performance of the corresponding lot-sizing solution. The table shows that, when the planning period and the number of devices are the same, with the increase in product types, fuzzy-GA's optimal value is smaller than that of the traditional GA. Therefore, the improved genetic algorithm has better optimization ability for different product types. The average optimal value of the fuzzy-GA is smaller when the production scale is the same, which indicates that the improved genetic algorithm still shows superior stability. At the same time, the number of iterations fluctuates less, and the running time is less, which shows that the algorithm has better stability for the increase in product types in solving lot-sizing problems. Table 13 also shows that, when the product type and the number of pieces of equipment are fixed, compared with the traditional GA, the optimal value and average optimal value of the fuzzy-GA are smaller, which indicates that the result performance of the improved genetic algorithm is better when the scale of the planning period is increased. The iterations of the two algorithms have slight fluctuation, and the running time is similar, which indicates that the running time of the two algorithms is relatively stable when the scale of the planning period increases, but the running time of the improved algorithm is slightly shorter. When product types and planned periods are fixed, Table 13 shows that, with the increase in the number of devices, the improved genetic algorithm solution has a better effect, the iteration times are more stable, and the algorithm runs faster.

In general, with the increase in the number of products, planning periods, and devices, compared with the traditional GA, the fuzzy-GA has a better effect on the optimal and average values of the target, more stable iterations, and less running time.

Table 14 shows the optimal value and running time of calculation examples of different scales under the proposed iteration stop condition, iteration 100 times, iteration 300 times, and iteration 500 times, respectively. In order to reduce the influence caused by the randomness of the algorithm, the optimal value is the optimal value of 10 calculations, and the running time is the average of 10 calculations. As can be seen from Table 14, when the number of iterations is constant, the iteration time increases with the increase of production scale. At the same scale, the optimal value becomes better and better with the increase of iterations, but calculation time is also increasing. Under different scales, the optimal value of the proposed stop condition of iteration is similar to the result performance of 300 iterations, but requires less calculation time.

Table 14. Results of different iterations with fuzzy-GA.

Product Type N	Planning Period T	Number of Devices M	The Proposed Iteration Condition		100 Times		300 Times		500 Times	
			Optimal Value	Running Time(s)	Optimal Value	Running Time(s)	Optimal Value	Running Time(s)	Optimal Value	Running Time(s)
5	5	9	87,629	4.3237	88,748	3.3064	88,080	9.7639	86,276	16.6609
10	5	9	242,461	5.5393	244,647	4.3638	247,769	13.0195	241,662	21.7165
20	5	9	908,033	7.6957	975,467	6.4136	947,128	21.6976	918,993	33.2535
20	7	9	1,349,712	9.2876	1,402,439	8.2405	1,345,295	24.7846	1,214,054	42.3357
20	10	9	2,005,225	11.3101	2,081,146	10.7705	1,877,778	33.7112	2,074,800	54.1895
20	10	15	1,888,792	12.2024	2,148,849	11.9693	1,984,062	37.0395	1,977,120	58.4784
20	10	20	2,410,076	15.5662	2,663,989	12.7645	2,512,804	38.8569	2,330,056	66.2304

Table 15 presents the performance comparison with other methods; the mean of iterations indicates the average number of iterations the algorithm runs to obtain the best value in all calculation examples. This indicator reflects the search capability of the algorithm. References [4,21] do not give the number of iterations of the algorithm. The mean of deviation is the expected ratio between the average and best values of the calculation examples in the dataset and the average values. This indicator can reflect the possibility of obtaining the optimal global solution of the algorithm. The smaller the value is, the greater the likelihood of obtaining the optimal solution during each algorithm operation. The range of variation indicates the difference between the maximum and the minimum values of the deviation, reflecting the fluctuation range of understanding. The smaller the value of the indicator, the more stable the solution. These two indicators together reflect the algorithm's global search capability and stability. Table 15 reveals that each proposed algorithm indicator is superior to the algorithm in references [21,39] and the PSO algorithm in reference [4]. Compared with the GLNPSO algorithm in reference [4], the global optimization ability of the proposed literature is slightly worse, but the stability of the solution is better.

Table 15. Performance comparison with other methods.

	GA-VNS [39]	COA [21]	PSO [4]	GLNPSO [4]	The Proposed Algorithm
Mean of iterations	100	-	-	-	26
Mean of deviation	0.185	0.062	0.038	0.017	0.024
Range of variation	0.288	0.165	0.297	0.066	0.027

6. Conclusions and Prospects

This study realizes solving the MLCLSP by the fuzzy-GA, which analyzes the results of different values of parameters of mutation membership function and crossover membership function in the fuzzy-GA and gives the specific parameters suitable for the MLCLSP. In the different production scales, the GA and the fuzzy-GA are run, and the optimal solution, the average value of the optimal solution, the iterations, and the running time are recorded. Compared with the traditional GA, the proposed algorithm can improve the solution speed, stability, and accuracy of the solution. The distribution of energy consumption is given for the case of the production of the synchronizer, which can provide a reference for relevant enterprises to reduce energy consumption.

Although the setup time is considered in the MLCLSP, the setup carryover during the planning period is not considered in order to reduce the difficulty of problem-solving, which actually exists in the synchronizer production. Moreover, the scheduling optimization in each planning period is also not considered when dealing with the MLCLSP in this study. Therefore, the proposed algorithm in this paper may not be effective for such cases.

Future work will focus on improving algorithms which can be used to solve the MLCLSP with setup carryover or scheduling optimization.

Author Contributions: Conceptualization, S.W. and B.Z.; funding acquisition, J.H.; methodology, S.W.; project administration, J.H.; supervision, J.H.; writing—original draft, S.W.; writing—review and editing, B.Z. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the intelligent manufacturing support project for major science and technology programs of Shaanxi Province grant number Grant No. 2018zdzx01-01-01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data involved in this study are available through the corresponding author.

Acknowledgments: The authors are grateful to the financial support of the intelligent manufacturing support of the intelligent manufacturing support project for major science and technology programs of Shaanxi Province (Grant No. 2018zdzx01-01-01).

Conflicts of Interest: All authors declare no conflict of interest.

References

1. Awad, M.I.; Hassan, N.M. Joint decisions of machining process parameters setting and lot-size determination with environmental and quality cost consideration. *J. Manuf. Syst.* **2018**, *46*, 79–92.
2. Jia, S.; Yuan, Q.; Cai, W.; Li, M.; Li, Z. Energy modeling method of machine-operator system for sustainable machining. *Energy Convers. Manag.* **2018**, *172*, 265–276.
3. Klement, N.; Abdeljaouad, M.A.; Porto, L.; Silva, C. Lot-Sizing and Scheduling for the Plastic Injection Molding Industry—A Hybrid Optimization Approach. *Appl. Sci.* **2021**, *11*, 1202.
4. Boonmee, A.; Sethanan, K. A GLNPSO for multi-level capacitated lot-sizing and scheduling problem in the poultry industry. *Eur. J. Oper. Res.* **2016**, *250*, 652–665.
5. Berretta, R.; Rodrigues, L.F. A memetic algorithm for a multistage capacitated lot-sizing problem. *Int. J. Prod. Econ.* **2004**, *87*, 67–81.
6. Buschkühl, L.; Sahling, F.; Helber, S.; Tempelmeier, H. Dynamic capacitated lot-sizing problems: A classification and review of solution approaches. *Or Spectr.* **2010**, *32*, 231–261.
7. Duda, J.; Stawowy, A. Optimization Methods for Lot-Sizing Problem in an Automated Foundry. *Arch. Metall. Mater.* **2013**, *58*, 863–866.
8. Jia, S.; Yuan, Q.; Lv, J.; Liu, Y.; Ren, D.; Zhang, Z. Therblig-embedded value stream mapping method for lean energy machining. *Energy* **2017**, *138*, 1081–1098.
9. Jinxing, X.; Jiefang, D. Heuristic genetic algorithms for general capacitated lot-sizing problems. *Comput. Math. Appl.* **2002**, *44*, 263–276.
10. Yijun, L.; David, L. Self-adaptive randomized constructive heuristics for the multi-item capacitated lot-sizing problem. *arXiv* **2021**, arXiv:2103.04199.
11. Furlan, M.M.; Santos, M.O. BFO: A hybrid bees algorithm for the multi-level capacitated lot-sizing problem. *J. Intell. Manuf.* **2017**, *28*, 929–944.
12. Kirschstein, T.; Meisel, F. A dynamic multi-commodity lot-sizing problem with supplier selection storage selection and discounts for the process industry. *Eur. J. Oper. Res.* **2019**, *279*, 393–406.
13. Cunha, J.O.; Kramer, H.H.; Melo, R.A. On the computational complexity of uncapacitated multi-plant lot-sizing problems. *Optim. Lett.* **2021**, *15*, 803–812.
14. Han, Y.; Kaku, I.; Tang, J.; Dellaert, N.; Cai, J.; Li, Y. A Comparison of Particle Swarm Optimizations for Uncapacitated Multilevel Lot-sizing Problems(Theory and Methodology). *J. Jpn. Ind. Manag. Assoc.* **2010**, *61*, 203–213.
15. Seeanner, F. Multi-Stage Simultaneous Lot-Sizing and Scheduling—Planning of Flow Lines with Shifting Bottlenecks; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013. https://doi.org/10.1007/978-3-658-02089-7.
16. Jia, S.; Yuan, Q.; Cai, W.; Lv, J.; Hu, L. Establishing prediction models for feeding power and material drilling power to support sustainable machining. *Int. J. Adv. Manuf. Technol.* **2019**, *100*, 2243–2253.
17. Jia, S.; Cai, W.; Liu, C.; Zhang, Z.; Bai, S.; Wang, Q.; Li, S.; Hu, L. Energy modeling and visualization analysis method of drilling processes in the manufacturing industry. *Energy* **2021**, *228*, 120567.

18. Zohali, H.; Naderi, B.; Mohammadi, M. The economic lot scheduling problem in limited-buffer flexible flow shops: Mathematical models and a discrete fruit fly algorithm. *Appl. Soft Comput.* **2019**, *80*, 904–919.
19. Farooq, B.; Bao, J.; Raza, H.; Sun, Y.; Ma, Q. Flow-shop path planning for multi-automated guided vehicles in intelligent textile spinning cyber-physical production systems dynamic environment. *J. Manuf. Syst.* **2021**, *59*, 98–116.
20. Che, Z.H.; Chiang, T.A.; Lin, T.T. A multi-objective genetic algorithm for assembly planning and supplier selection with capacity constraints. *Appl. Soft Comput.* **2021**, *101*, 107030.
21. Hadi, A.S.; Mehdi, R.B. Solving a flexible job shop lot sizing problem with shared operations using a self-adaptive COA. *Int. J. Prod. Res.* **2021**, *59*, 483–515.
22. Andrés, A.; Mingyuan, C. Multi-level lot-sizing with raw-material perishability, deterioration, and lot-sizing ordering: An application of production planning in advanced composite manufacturing. *Comput. Ind. Eng.* **2020**, *145*, 106484.
23. Sikora, R. A genetic algorithm for integrating lot-sizing and sequencing in scheduling a capacitated flow line. *Comput. Ind. Eng.* **1996**, *30*, 969–981.
24. Soltanali, H.; Rohani, A.; Tabasizadeh, M.; Abbaspour-Fard, M.H.; Parida, A. An improved fuzzy inference system-based risk analysis approach with application to automotive production line. *Neural Comput. Appl.* **2020**, *32*, 10573–10591.
25. Tempelmeier, H.; Helber, S. A heuristic for dynamic multi-item multi-level capacitated lot-sizing for general product structures. *Eur. J. Oper. Res.* **1984**, *75*, 296–311.
26. Boctor, F.F.; Poulin, P. Heuristics for N-product, M-stage, economic lot sizing and scheduling problem with dynamic demand. *Int. J. Prod. Res.* **2005**, *43*, 2809–2828.
27. Miller, A.J.; Nemhauser, G.L.; Savelsbergh, M. *Solving Multi-Item Capacitated Lot-Sizing Problems with Setup Times by Branch-and-Cut*; CORE Discussion Papers; Université Catholique de Louvain: Ottignies-Louvain-la-Neuve, Belgium, 2000.
28. Zhang, C.; Zhang, D.; Wu, T. Data-driven Branching and Selection for Lot-sizing and Scheduling Problems with Sequence-dependent Setups and Setup Carryover. *Comput. Oper. Res.* **2021**, *132*, 105289.
29. Ramezanian, R.; Saidi-Mehrabad, M.; Fattahi, P. MIP formulation and heuristics for multi-stage capacitated lot-sizing and scheduling problem with availability constraints. *J. Manuf. Syst.* **2013**, *32*, 392–401.
30. Verma, M.; Sharma, R. Lagrangian relaxation and bounded variable linear programs to solve a two-level capacitated lot sizing problem. In Proceedings of the 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, India, 8–10 April 2011; Volume 6, pp. 188–192.
31. Tempelmeier, H.; Buschkühl, L. A heuristic for the dynamic multi-level capacitated lot-sizing problem with linked lot-sizes for general product structures. Or Spectrum **2009**, *31*, 385–404.
32. Wu, T.; Shi, L.; Geunes, J.; Akartunalı; K An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging. *Eur. J. Oper. Res.* **2011**, *214*, 428–441.
33. Zhao, Q.; Xie, C.; Xiao, Y. A variable neighborhood decomposition search algorithm for multi-level capacitated lot-sizing problems. *Electron. Notes Discret. Math.* **2012**, *39*, 129–135.
34. Chen, H. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega* **2015**, *56*, 25–36.
35. Mohammadi, M.; Fatemi Ghomi SM, T.; Karimi, B.; Torabi, S.A. Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lot-sizing problem with sequence-dependent setups. *J. Intell. Manuf.* **2010**, *21*, 501–510.
36. Pitakaso, R.; Almeder, C.; Doerner, K.F.; Hartl, R.F. Combining exact and population-based methods for the constrained multi-level lot-sizing problem. *Int. J. Prod. Res.* **2006**, *44*, 4755–4771.
37. Toledo, C.F.M.; De Oliveira, R.R.R.; França, P.M. *A Hybrid Multi-Population Genetic Algorithm Applied to Solve the Multi-Level Capacitated Lot Sizing Problem with Backlogging*; Elsevier Science Ltd.: Amsterdam, The Netherlands, 2013; Volume 40, pp. 910–919.
38. Mohammadi, M.; Ghomi, S.M.T.F. Genetic algorithm-based heuristic for capacitated lot-sizing problem in flow shops with sequence-dependent setups. *Expert Syst. Appl.* **2011**, *38*, 7201–7207.
39. Duda, J. A hybrid genetic algorithm and variable neighborhood search for multi-family capacitated lot-sizing problem. *Electron. Notes Discret. Math.* **2017**, *58*, 103–110.
40. Quadt, D.; Kuhn, H. Capacitated lot-sizing with extensions: A review. *4OR* **2008**, *6*, 61–83.