Vol. 9, No. 2, 2022, Pages 129-147

JIEMS
Journal of Industrial Engineering and Management Studies

journal homepage: www.jiems.icms.ac.ir

Lot-Sizing and scheduling in a flexible job-shop environment considering energy efficiency

Arezoo Osati[1], Esmaeil Mehdizadeh *[1], Sadoullah Ebrahimnejad[2]

[1] Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.
[2] Department of Industrial Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran.

Received: Mar 2022-12/ Revised: Jul 2022-12/ Accepted: Jul 2022-30

Abstract

The purpose of this paper is to optimize the integrated problem of lot-sizing and scheduling in a flexible job-shop environment considering energy efficiency. The main contribution of the paper is simultaneously considering lot-sizing and scheduling decisions, while accounting for energy efficiency. In order to achieve this objective, a mathematical model has been developed for integrated optimization of scheduling and lot-sizing problems. The developed model uses a big bucket approach and is presented as a mixed integer nonlinear problem (MINLP). The BARON solver in GAMS software has been used to solve the proposed MINLP model. By defining the relative optimality limit (OPTCR) of 0.05 for the termination criterion in BARON solver, GAMS has not been able to solve large problems at a specified time to achieve relative optimality. Therefore, due to the NP-hard nature of the problem, a new genetic-based evolutionary algorithm has been developed to solve the problem of large scale. In the developed algorithm, a different approach (instead of cross-over and mutation operators) is used to generate a new solution. By presenting and solving various problems, the efficiency of this algorithm for solving big problems is shown. Comparing the values of the objective function obtained from the genetic algorithm and the exact method shows that, especially in large problems, the genetic algorithm has been able to achieve a better solution than GAMS software in a limited time. It has also been shown that energy efficiency has a significant effect on the solution of the problem.

Keywords: lot-sizing; flexible job-shop; scheduling; genetic algorithm; energy efficiency.

Paper Type: Original Research

1. Introduction

Today, production planning is one of the most important problems in the manufacturing industry and decision making is one of the challenges of industry managers in this area (Ahmadi et al., 2016). Production planning is an activity that seeks to make the best use of production resources in a way that meets production objectives such as production requirements and expected sales in a time interval of time called the planning horizon (Brucker and Schlie., 1990). These problems are divided into three classes based on time intervals: long-term, medium-term and short-term. One of the most important and widely used problems in -production planning, which is a part of the medium-term problems, is the lot-sizing problem. This problem determines production levels and scheduling (Karimi et al., 2003).

Correct and appropriate calculation of the lot-size is one of the factors affecting the performance of the system, its efficiency, as well as maintaining the ability and competitiveness in the market. Therefore, the development and improvement of models and methods to determine the lot-size is very important. Lot-sizing problems can be categorized into different modes. This categorization includes production structure, including single-stage and multi-stage structures, production with limited or unlimited capacity, existance one or more resources, as well as one or more products, etc. (Bitran and Yanasse, 1982).

In recent years, manufacturing systems need plans for problems such as increasing product diversity, rapid changes in market and customer demand and the need for high flexibility, competitive environment, the need to reduce costs, reduce delivery time, etc. (Xiong and Fu., 2018). The details are more accurate and realistic, and for this reason, the researchers studied the problems of simultaneous lot-sizing optimization and scheduling by considering the conditions of scheduling problems and the sequence of operations in lot-sizing problems. On the other hand, with the increase in computing power of the new generation of computers and extensive research conducted to provide more efficient and faster solution methods, the concern of increasing computational complexity resulting from the combination of these two problems has been somewhat reduced (Rouhanineja, 2015).

*Corresponding Author: emqiau@yahoo.com

Copyright: Creative Commons Attribution 4.0
DOI: 10.22116/jiems.2022.333764.1479

In this study, we examine simultaneous optimization of lot-sizing and scheduling of operations in a flexible job-shop production environment.

Scheduling problems can be classified in different ways. For example, scheduling problems can be categorized as dynamic or static, deterministic or stochastic, single-product or multi-product, single-process or multi-process, etc. Depending on the number of operations required to process a task and the number of machines available for processing each operation, various flow patterns can be enumerated (Choy., 2011).

The problem of flexible job-shop production is an extension of the job-shop production problem first proposed by Brucker and Schlie (1990). The problem of job-shop production is one of the types of scheduling problems that seeks to determine the best sequence of work in a multi-machine environment when the therout of the works is predetermined and can be different. In the classic mode, there is n jobs that each job is to be processed by a maximum of m machine and the problem is to find the sequencing of tasks on each machine. The problem of job-shop production in generally NP-hard (Garey, 1976). However, in the flexible form of the problem, any operation at any station can be processed on a set of machines. Therefore, flexible job-shop production scheduling problems include two sub-problems. The first is to assign each operation to a machine from the set of machines allowed for that operation, and the second is to determine the sequence of all operations in such a way that the predetermined objectives are achieved. For this reason, having two variables of assignment and sequencing decision has a higher level of complexity than the problem of job-shop production with one variable of sequencing decision. Due to the nature of the flexible job-shop production environment, the problem studied in this research has a multi-stage and multi-machine structure. In this problem, we seek to determine the optimal or near-optimal production plan in a finite planning horizon while the capacity of machines is limited. In this problem, it is necessary to determine the sequence of production and the amount of production for each product regarding the sequence and allocation of products within a time interval (Rohaninejad, 2015).

Two problems that a production manager usually faces in managing a company at the planning level are: 1) Determine the number of products that should be produced in a time interval in order to minimize production costs, these costs include production, time interval and set-up costs. 2) Determining the best resource allocation and the best times to start and finish jobs in order to optimize some criteria such as minimizing the make-span, maximizing efficiency or operational capacity, etc. (Gomez et al., 2014). The first problem falls into the category of medium-term decision problems and is called the lot-sizing problem, while the second problem belongs to short -term decision problems and is called scheduling problem (Sifaleras et al., 2015) . These two categories of decision –making problems are very close to each other; because the output of lot-sizing is an input for sequence and scheduling. Proposed models for integrated lot-sizing and scheduling problems may be classified into small bucket and big bucket. In small bucket problems, first the larger time interval is broken down into smaller time intervals and then modeling is done (Baki et al., 2014). This method greatly increases the complexity of the problem. In models with a small bucket, it is usually assumed that one or a maximum of two products may be produced in each time interval. Models with a big bucket allow production of different products in one time interval, but do not specify the production sequence of these products (Baki et al., 2014). Capacitated Lot Sizing Problem (CLSP) is a model with a big bucket. Small-bucket lot-sizing problems include Discrete Lot sizing and Scheduling Problem (DLSP), Continuous Setup Lot-sizing Problem (CSLP), Proportional Lot-sizing and Scheduling Problem (PLSP). General Lot sizing and Scheduling Problem (GLSP) is a generall model that incorporates previously presented models (Zhang et al., 2011).

A production system usually has different steps to perform multiple operations in order to convert raw materials into the final product. In these systems, production planning, which aims to find the best way to use production resources to achieve production objectives on the planning horizon, will be important for system management (Zarrouk et al., 2019) . On the other hand, the traditional production planning problem considers performance indicators such as processing time, cost and quality as optimization objectives in production systems; however, it does not consider energy consumption and environmental impacts (Zhang et al., 2020) .

Flexible production planning with multi-stage, multi-product, multi-machine, and multi-time interval can be described as follows (Giglio et al., 2017) :

The demand for each type of product in each time interval is deterministic. During each time interval, a number of used products are returned to the factory where they are recycled and stored in the returned inventory. The number of return products in each time interval is provided by the forecasting unit and it is assumed that this number is deterministic. There is no disposal option in the system intended for return products and it is assumed that raw materials are always available in any quantity in any time interval. It is also assumed that there is no difference between final products obtained through production and those obtained through remanufacturing, and all final products are stored in a usable inventory to meet demand. A simple design of this system is shown in Figure 1.

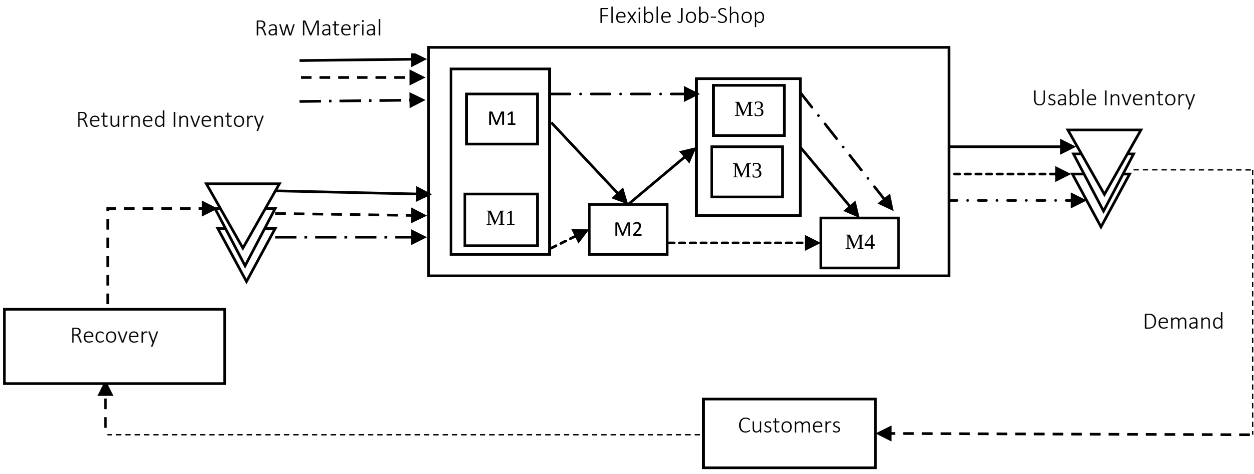

Figure 1: Integrated manufacturing/remanufacturing system (Giglio et al., 2017)

In this system, both raw and returned materials are transferred to a flexible job-shop system to follow their production processes. Several product classes are processed in the system, each product class has a specific precedence network and properties. Each class of products has a single processing path on different work stations, in each station (at least one of the stations) there are at least two similar machines in parallel to perform each step. The processing path is the same for both production and remanufacturing methods. Nominal processing times of products on different machines are fixed and predetermined. Each time interval has a duration (time capacity) and all machines must complete the allocated operation in a time interval in the available time. The main contribution of the paper is simultaneously considering lot-sizing and scheduling decisions, while accounting for energy efficiency. We also propose a novel genetic algorithm to solve the problems of large scale. The rest of the paper includes following sections: Section 2 presents literature review; Mathematical modelling and problem formulation is presented in Section 3; Section 4 entails experimental results; finally, conclusion and future research directions is presented in Section 5.

2. Literature Review

2.1. Flexible Job-shop Scheduling Problem (FJSP)

FJSP is a classic generalized job-shop model in which operations are performed on a set of flexible machines. The following is a brief overview of the FJSP literature.

FJSP was first proposed by Brucker and Schlie (1990). This paper used a polynomial time algorithm to solve FJSP. Brandimarte (1993) proposed a hierarchical and innovative two-tier Tabu Search approach to minimize completion time in FJSP. In addition, Hurink et al. (1994) proposed another Tabu Search algorithm to solve FJSP. Mixed integer optimization for FJSP with sequence-dependent setup was introduced by Choi and Choi (2002). Kacem et al. (2000) developed three dispatching rules to decide on the problem of sequencing and advanced genetic manipulation to improve the solution. To minimize machine idle time, Chan et al. (2006) proposed a resource constraint in FJSP. Gao et al. (2008) tried to jointly minimize the manufacturing time interval, the maximum machine workload, and the total workload using a genetic algorithm and a variable neighborhood descent. In addition, Pezzella et al. (2008) introduced and analyzed an improved genetic algorithm for FJSP. Chen et al. (2008) studied an industrial sample in a weapons factory using FJSP. To obtain the optimal make-span, Xing et al. (2010) worked on a hybrid algorithm based on the ant colony optimization algorithm. Chan and Choy (2011) designed a genetic algorithm-based job scheduler for the real-world FJSP with multiple products, parallel machines, and setup times. In another study, Mahdavi et al. (2010) proposed a decision support system for FJSP with industrial data. In addition, Zhang et al. (2011) combined global selection and local selection strategies and proposed a genetic algorithm to find an effective planning for FJSP. Al-Turki et al. (2011) proposed different dispatching rules for FJSP with setup time, batch processing, and uncertain data. Xiong and Fu (2018) designed a multi-agent system for the FJSP that seeks to reduce complexity and cost, improve flexibility, and increase robustness. In addition, Gao et al. (2015) examined the arrival of new work to the FJSP and developed effective exploratory methods to minimize the maximum workload and total workload of the machines. In addition, Ahmadi et al. (2016) developed an evolutionary algorithm for optimizing multi-objective FJSP with random machine failure.

2.2. Lot-Sizing with recycled products and remanufacturing

Over the past five decades, a great deal of research has been done on lot-sizing and scheduling of production systems. Drexl and Kimms (1997) and Jans and Degraeve(2008) presented a review of metaheuristic approaches to lot-sizing problems. Dynamic lot-sizing is presented, while in Buschkuhl et al. (2010) the authors categorize and review methods for solving lot-sizing problems with capacity constraints. However, few studies on lot-sizing in remanufacturing systems, called Dynamic Lot Sizing with Product Returns and Remanufacturing (DLSPR), have been reported in the literature review.

Richter and Sombrutzki (2000) refer to the production planning and dynamic inventory control model in both remanufacturing systems and hybrid production / remanufacturing systems. They provide the mathematical formulation for DLSP for these two types of systems. Richter and Weber (2001) expanded the previous work of Richter and Sombrutzki (2000) by adding the variables of remanufacturing and production costs. Golany et al. (2001) studied the production planning of a single product system with the option of reproducing in which the demand and rate of return are deterministic and there is also the option of disposal. Yang et al. (2005) focused on the concave cost version of the production planning problem proposed by Golani et al. (2001). They addressed the problem of unrestricted capacity in which all cost functions are concavet. In this study, it is stated that even if the costs are fixed, the problem is NP-hard. To solve the problem, the network flow type is formulated and an innovative method with polynomial time is used by using the special structure of the optimal extreme point solution. Teunter et al. (2006) studied the problem of capacity planning without capacity constraints, in which two different cases are considered for set-up costs; Joint setup costs for both production and remanufacturing, meaning that there is a single production line, or different setup costs when the production lines are separate. In case of common production line, a precise dynamic programming algorithm with polynomial time is used to solve the problem.

Teunter et al. (2009) developed rapid heuristic methods for the problems presented in the previous work of Teunter (2006). Pineyro and Viera (2009) address a dynamic lot-sizing problem with remanufacturing and disposal and propose a tabu search algorithm to solve the problem. Pineyro et al. (2010) examined the problem of lot-sizing in which the demand for new products and remanufacturing are different. Demand for reproducible products can be met by new products, but not vice versa. This one-way substitution comes with a cost to the model. This study also shows that the problem, even with fixed cost, is NP-hard. A tabu search-based method was provided for a near-optimal solution. Wang et al. (2011) examined the problem of lot-sizing in which separate production lines are considered for production and remanufacturing processes, and outsourcing is allowed to meet demand. This is a generalization of one of the problems raised in Teunter (2006) with separate product lines. In case of large amounts of return products, a dynamic programming method is adopted to obtain the optimal solution and the complexity of this method is expressed as O (T2) that T is the number of time intervals on the planning horizon. Baki et al. (2014) introduced a new mathematical formulation for DLSPR and proved that it is an NP-hard problem. They developed a dynamic programming method to solve it. Sifaleras et al. (2015) provide a variable neighborhood search for DLSP with product return and recycling. It is worth noting that all of the above works show the problem of lot-size with product return and remanufacturing without capacity constraints. Mehdizadeh and Fatehi (2017) proposed an algorithm for solving a fuzzy model which determines the lot-size of a single machine system.

2.3. Problem of Integrated lot-sizing and scheduling

A General Lot sizing and Scheduling Problem (GLSP) for the flexible job-shop production problem is presented by Rouhaninejad (2015). Roshani et al. (2015) investigated a specific case of simultaneous lot-sizing and scheduling optimization by combining the lot-sizing problem and the scheduling problem in a flexible job-shop production environment. In order to determine a combination of two -year decision variables with the aim of minimizing system costs, a new hybrid integer programming model is presented as a big bucket model by combining genetic algorithm with particle swarm optimization algorithm. Rouhaninejad et al. (2016) also examine the problem of integrated lot-sizing and flexible job-shop scheduling. The problem involves simultaneous decision making about sequence operations, batch size, and machine allocation to operations to optimize a multi-objective function, including minimizing total system costs, machine workload, and make-span, while providing demand without shortages. Due to the complexity of the problem, a metaheuristics method was developed based on genetic algorithm and particle swarm optimization. In addition, the Taguchi method is used to set effective metaheuristic parameters. Sahraeian et al. (2017) proposed an integrated problem of lot-sizing and flexible job-shop production with limited capacity of machines and have used the small-bucket model to formulate the problem. A new harmony search algorithm has been developed to solve the problem. This paper also uses a mixed integer programming model.

Zarrouk et al. (2019) provide a two-stage particle swarm optimization algorithm for the flexible job-shop scheduling problem. The first stage determines the operation of the machines, while the second level determines the sequence of operations on the machines. To reduce the number of visited solutions, a boundary checking strategy is used based on the value of the objective function. This algorithm is tested on a considerable number of different standard problems with the existing algorithms for the flexible job-shop scheduling problem, and is shown to have acceptable performance. Zhang et al. (2020) designed a new two-stage framework with important scheduling features selection to explore the scheduling solution with only the features selected for flexible job-shop scheduling. The results show that their proposed genetic algorithm can successfully achieve a good scheduling with fewer decision variables. Hajibabaei and Behnamian (2021) investigated the effect of flexible resources on the problem of flexible job-shop scheduling with parallel machines and sequence - dependent setup times. Also, by presenting the linear programming model of complex integer program, they tried to minimize production, total delays, delivery time and inventory costs. After the model is solved by GAMS, due to the NP-Hard nature of problem, the Tabu Search algorithm is used for large problems. Finally, the obtained results are compared with the genetic algorithm.

As can it be seen in the literature presented above, none of the studies in the fields of flexible job-shop scheduling, lot-sizing, and remanufacturing have considered these three problems together. Therefore, the main contribution of this paper is to simultaneously consider lot-sizing, flexible job-shop scheduling and remanufacturing with the aim of minimizing production costs, remanufacturing costs and also minimizing energy costs consumed by machines. We also consider energy efficiency considerations in the integrated lot-sizing and scheduling of a flexible job-shop problem. In addition, we proposed a new genetic algorithm to solve the problems of large scale.

3. Problem formulation

This section introduces sets, parameters, and variables. In addition, the mathematical model is presented including constraints and objective functions.

3.1. Sets

$\mathcal{P} = \{1. \dots . P\}$ Set of product classes

$\mathcal{K} = \{1. \dots . K\}$ Set of machines

$\mathcal{T} = \{1. \dots . T\}$ Set of time intervals

$\mathcal{F} = \{M. R\}$Set of different production methods; manufacturing (M) and remanufacturing (R)

$\mathcal{H}_i = \{1. \dots . H_i\}$ Product operations set, i∈P

$O_i = \{K(1). K(2). \dots . K(o_{iH_i})\}$ A set of H_i sets of machines that are able to process materials for producing final products of class i∈P. Where the $K(o_{ih})$ is set of machines able to perform operations o_{ih} $(h \in \mathcal{H}_i)$.

3.2. Parameters

H_i: Number of operations required for the product i∈P

q_{ihk}^{f}: The nominal processing time of operations $h \in \mathcal{H}_i$ of product i∈P on the machine $k \in K(o_{ih})$when using the production method f∈F

e_{ihk}^{f}: Maximum compression time of operations $h \in \mathcal{H}_i$of product i∈P on the machine $k \in K(o_{ih})$ when using the production method f∈F

e_{ihk}^{f}: Number of returned products of class i∈P in the time interval t∈T

d_{it}: Demand for products of class i∈P in the time interval t∈T

v_{it}^{f}: The production cost for each unit of product i∈P in the time interval t∈T produced by the method f∈F

w_{ih}^{k}: setup cost of machine k to perform operations h on the product i∈P

s_{ih}^{k}: setup time of machine k to perform operations h on the product i∈P

$p_{hh'}^{i}$: Binary parameter, $p_{hh'}^{i}$=1 if the operation $h \in \mathcal{H}_i$ of the product i∈P is a precedence for the operation $h' \in \mathcal{H}_i$.

h_{it}^{u}: The time interval cost of a unit of returned inventory for product class i∈P at the time interval t∈T

h_{it}^{s}: The time interval cost of a unit of usable inventory for product class i∈P at the time interval t∈T

h_{it}^{b}: The shortage cost of a unit of product class i∈P at the time interval t∈T

c_t: Maximum time available in the time interval t∈T (production capacity)

cp_k: Unit cost for the energy consumption of the machine k∈K during the processing of the operations.

ci_k: Unit cost for power consumption of the machine k∈K in idle mode (which is spent by the machine for each unit of time spent in idle mode)

cc_k: Unit cost for machine k∈K power consumption to compress processing time (which is spent by the machine to reduce nominal processing time)

ψ: A large enough number

b_{it}^{f}: The maximum allowable production capacity of product i∈P in the time interval t∈T by method f∈F

3.3. Decision variables

3.3.1 Integer variables

$X_{it}^{f} \geq 0$: Amount of products of class i∈P produced in the time interval t∈T by the method of manufacturing (f=M) or remanufacturing (f=R)

$I_{it}^{u} \geq 0$: Invenory level of returned product i∈P in the time interval t∈T

$I_{it}^{s} \geq 0$: Available inventory level of the usable product i∈P in the time interval t∈T

$I_{it}^{b} \geq 0$: Shortage for the product i∈P in the time interval t∈T

3.3.2. Continuous variables

C_{PHS}: Total cost of production, time interval and setup

C_E: Total cost of energy

$PT_{ihkt}^{f} \geq 0$: Actual processing time of the operation $h \in \mathcal{H}_i$of product i∈P on the machine $k \in K(o_{ih})$ in the time interval t∈T with the method f∈F.

$Z_{ihkt}^{f} \geq 0$: Compressed amount of processing time of the operation $h \in \mathcal{H}_i$of product i∈P on the machine $k \in K(o_{ih})$ in the time interval t∈T with the method f∈F.

$ST_{ihkt}^{f} \geq 0$: Start time of the operation $h \in \mathcal{H}_i$of product i∈P on the machine $k \in K(o_{ih})$ in the time interval t∈T with the method f∈F.

CO_{kt}: Completion time of all operations assigned to the machine k∈K in the time interval t∈T

IT_{kt}: Machine k∈K idle time during time interval t∈T.

3.3.3. Binary variables

$\delta_{ihi'h'}^{ktfg} \in \{0.1\}$: Binary precedence variable. $\delta_{ihi'h'}^{ktfg}$=1 if operations $h'$of the product i'∈P in the time interval t∈T and production method g is scheduled on the machine k after operations h of the product i∈P with production method f. Otherwise $\delta_{ihi'h'}^{ktfg}$=0.

$y_{ihkt}^{f} \in \{0.1\}$: y_{ihkt}^{f}=1 if operations h of the product i∈P is scheduled in time interval t on the machine k using method f∈F. ; Otherwise y_{ihkt}^{f}=0

B_{it}^{f}: lot-size production binary variable. B_{it}^{f}=1 In case of producing non-zero amount of product i in the time interval t using the method f∈F. Otherwise B_{it}^{f}=0.

3.3.4. Mathematical model

$$Min\mathcal{C}_{PHS} + \mathcal{C}_E \tag{1}$$

St:

$$\mathcal{C}_{PHS} = \sum_{t\in T}\sum_{i\in P}\sum_{f\in\mathcal{F}} v_{it}^{f}X_{it}^{f} + \sum_{i\in P}\sum_{h\in H_i}\sum_{t\in T}\sum_{k\in K}\sum_{f\in\mathcal{F}} y_{ihkt}^{f}\, w_{ih}^{k} + \sum_{t\in T}\sum_{i\in P}(h_{it}^{u}I_{it}^{u} + h_{it}^{s}I_{it}^{s} + h_{it}^{b}I_{it}^{b}) \tag{2}$$

$$\mathcal{C}_E = \sum_{t\in T}\sum_{i\in P}\sum_{k\in K(o_{ih})}\sum_{h\in H_i}\sum_{f\in\mathcal{F}} c\,p_k PT_{ihkt}^{f} + \sum_{t\in T}\sum_{k\in\mathcal{K}} ci_k IT_{kt} + \sum_{t\in T}\sum_{k\in\mathcal{K}}\sum_{i\in P:k\in K(o_{ih})}\sum_{f\in\mathcal{F}}\sum_{h\in H_i} c_k Z_{ihkt}^{f} \tag{3}$$

$$IT_{kt} = CO_{kt} - \sum_{i\in P:k\in K(o_{ih})}\sum_{f\in\mathcal{F}}\sum_{h\in H_i} PT_{ihkt}^{f} \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K} \tag{4}$$

$$I_{it}^{u} - I_{i(t-1)}^{u} + X_{it}^{R} = r_{it} \quad \forall t\in\mathcal{T}.\forall i\in\mathcal{P} \tag{5}$$

$$I_{i(t-1)}^{s} - I_{it}^{s} + \sum_{f\in\mathcal{F}} X_{it}^{f} + I_{it}^{b} - I_{i(t-1)}^{b} = d_{it} \quad \forall t\in\mathcal{T}.\forall i\in\mathcal{P} \tag{6}$$

$$X_{it}^{f} \le b_{it}^{f} * B_{it}^{f} \quad \forall t\in\mathcal{T}.\forall i\in\mathcal{P}.\forall f\in\mathcal{F} \tag{7}$$

$$\sum_{h\in\mathcal{H}_i}\sum_{k\in\mathcal{K}} y_{ihkt}^{f} = |\mathcal{H}_i| * B_{it}^{f} \quad \forall t\in\mathcal{T}.\forall i\in\mathcal{P}.\forall f\in\mathcal{F} \tag{8}$$

$$\sum_{k\in K(o_{ih})} y_{ihkt}^{f} \le 1 \quad \forall t\in\mathcal{T}.\forall i\in\mathcal{P}.\forall h\in\mathcal{H}_i.\forall f\in\mathcal{F} \tag{9}$$

$$ST_{ihkt}^{f} \le y_{ihkt}^{f} * c_t \tag{10}$$

$$PT_{ihkt}^{f} \le y_{ihkt}^{f} * c_t \tag{11}$$

$$PT_{ihkt}^{f} = q_{ihk}^{f}X_{it}^{f} - Z_{ihkt}^{f} + y_{ihkt}^{f} * s_{ih}^{k} \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K}.\forall i\in\mathcal{P}: \mathcal{O}_i\cap\{k\}\neq\emptyset.\forall f\in\mathcal{F}.\forall h\in\mathcal{H}_i \tag{12}$$

$$Z_{ihkt}^{f} \le e_{ihk}^{f} y_{ihkt}^{f} \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K}.\forall i\in\mathcal{P}: \mathcal{O}_i\cap\{k\}\neq\emptyset.\forall f\in\mathcal{F}.\forall h\in\mathcal{H}_i \tag{13}$$

$$CO_{kt} \le c_t \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K} \tag{14}$$

$$ST_{ihkt}^{f} + PT_{ihkt}^{f} \le CO_{kt} \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K}.\forall i\in\mathcal{P}: \mathcal{O}_i\cap\{k\}\neq\emptyset.\forall f\in\mathcal{F} \tag{15}$$

$$p_{hh'}^{i} * \left(\sum_{k\in\mathcal{K}} ST_{ihkt}^{f} + PT_{ihkt}^{f}\right) \le \sum_{k\in\mathcal{K}} ST_{ih'kt}^{f} \quad \forall t\in\mathcal{T}.\forall i\in\mathcal{P}.\forall h.h'\in\mathcal{H}_i.\forall f\in\mathcal{F} \tag{16}$$

$$\psi\left(1-\delta_{ihi'h'}^{ktfg}\right) + ST_{i'h'kt}^{f} - ST_{ihkt}^{g} \ge PT_{ihkt}^{g} \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K}.\forall f.g\in\mathcal{F}.\forall i.i'\in\mathcal{P}: \mathcal{O}_i\cap\mathcal{O}_{i'}\cap\{k\} \neq\emptyset.\left((i\ge i')\wedge(f\neq g)\right)\vee\left((i>i')\wedge(f=g)\right).\forall h.h'\in\bigcup_{i=1}^{\mathcal{P}}\mathcal{H}_i \tag{17}$$

$$\psi\delta_{ihi'h'}^{ktfg} + ST_{ihkt}^{g} - ST_{i'h'kt}^{f} \ge PT_{i'h'kt}^{f} \quad \forall t\in\mathcal{T}.\forall k\in\mathcal{K}.\forall f.g\in\mathcal{F}.\forall i.i'\in\mathcal{P}: \mathcal{O}_i\cap\mathcal{O}_{i'}\cap\{k\}\neq\emptyset.((i \ge i')\wedge(f\neq g))\vee((i>i')\wedge(f=g)).\forall h.h'\in\bigcup_{i=1}^{\mathcal{P}}\mathcal{H}_i \tag{18}$$

In equation (1), total cost (objective function) is presented as sum of production cost ($\mathcal{C}_{PHS}$) and energy cost ($\mathcal{C}_E$). Equation (2) shows $\mathcal{C}_{PHS}$as sum of production, setup, inventory, and remanufacturing costs. The cost of energy is modeled by Equation (3), which is the sum of the costs paid for energy. constraint (4) calculates idle times of machines. Constraints (5) and (6) provide the dynamic equations of returned inventory and deliverable inventory. Constraint (7) ensures that B_{it}^{f}=1 when $X_{it}^{f} > 0$.

Constraint (8) causes that in case of production in each time interval (B_{it}^{f}=1), all the desired production operations will be performed in the same time interval, otherwise no operation will be performed for that product. Constraint (9) ensures that each operation in each time interval can be performed on a maximum of one machine. According to constraints (10) and (11), if the corresponding activity of the variable y_{ihkt}^{f} is performed, the start time and operation time of the relevant activity is less than the time capacity of the corresponding interval. Constraint (12) determines the compressed processing time of the operation on the machine k. Constraint (13) determines the upper limit of compression time. Constraint (14) causes all machines to perform their specified operations in the time available for the interval t. Constraint (15) specifies the processing time of each machine in each time interval. Constraint (16) ensures that each product follows a certain sequence. Constraints (17) and (18) determine the sequences of operations of h and $h'$of products i and j. It can be seen that in this modeling time intervals -are not broken down into smaller time intervals and therefore the big bucket model is used. Instead, model handles the sequence decisions by using the variables $\delta_{ihi'h'}^{ktfg}$.

4. Excremental Results

4.1. Numerical results

In this section, an illustrative example of the problem is presented and problems of different sizes are solved to measure computational performance of the model. Tables 1 to 6 provide some sample input parameters for the illustrative problem. To find the best solution, the model was solved in GAMS using the BARON solver. Following Tables and Figures 3 to 6 report the optimal solution, where the value of the objective function is 1480. To evaluate the performance of the proposed model in comparison with traditional lot sizing models that considers only production cost, this example is also solved without considering the cost of energy in the objective function. The obtained schedule is illustrated by Gantt charts in Figures 5 and 6. The value of the new objective function for this solution is 422. Table 9 shows the difference between the two solutions. As it can be seen in this table, by taking into account the cost of energy in the objective function, although the production cost increases (from 322 to 422), the cost of energy decrease by 670 units (from 1658 to 988) and total cost is reduced by 570 units (from 1980 to 1410). In addition, the total idle time of the machines is reduced from 561 to 88 time units. It can be concluded that the first solution, presented by the complete model reported, is superior to the second solution, which only considered the C_{PHS} cost.

Table 1: Sample problem specifications

Number of product classes	Number of time intervals	Number of machines	Number of product 1 activities	Number of product 2 activities
2	2	3	4	5

Table 2: Machines capable of processing each activity

Product class	Activity	Machine		
		1	2	3
1	1	1		1
1	2	1	1	
1	3	1		1
1	4	1	1	1
2	1		1	1
2	2		1	
2	3	1		1
2	4	1	1	
2	5			1

Table 3: Number of returned products in each time interval

Product \ time interval	1	2
1	10	14
2	8	10

Table 4: Number of product demand in each time interval

Product \ time interval	1	2
1	15	16
2	12	18

Table 5: Production costs for unit of product

Product class	Time interval	Production method	
		manufacturing	remanufacturing
1	1	4	2
1	2	6	4
2	1	5	3
2	2	8	5

Table 6: Time capacity of each time interval

Time interval	1	2
Time capacity	250	300

The precedence relationships between the activities of each product class are as follows:

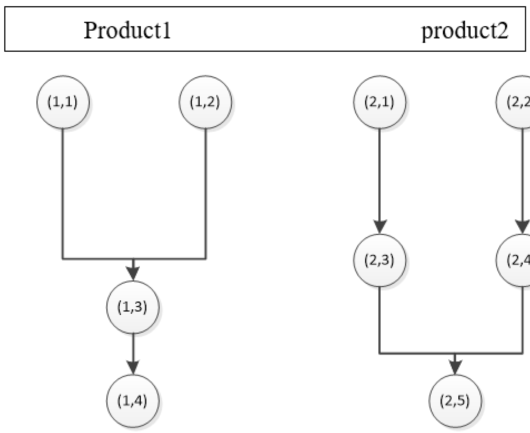

Figure 2: Precedence relationships between the activities of each product class

4.1.1. Solving the sample problem considering energy efficiency

Table 7 shows the amount of production in each time interval (taking into account energy efficiency). It can be seen from this table that no class 2 products were produced in the second time interval.

Table 7: Production in each time interval (taking into account energy efficiency)

Product class	Time interval	Production method	
		Manufacturing	Remanufacturing
1	1	5	10
1	2	2	14
2	1	22	8

The schedule of activities for each time interval is shown as Gantt charts in Figures 3 and 4. Purple bars indicate manufacturing operations and green bars indicate remanufacturing operations. In addition, the numbers displayed on each activity are presented as ordered tuples (i.h.k.t.f) to identify activities.

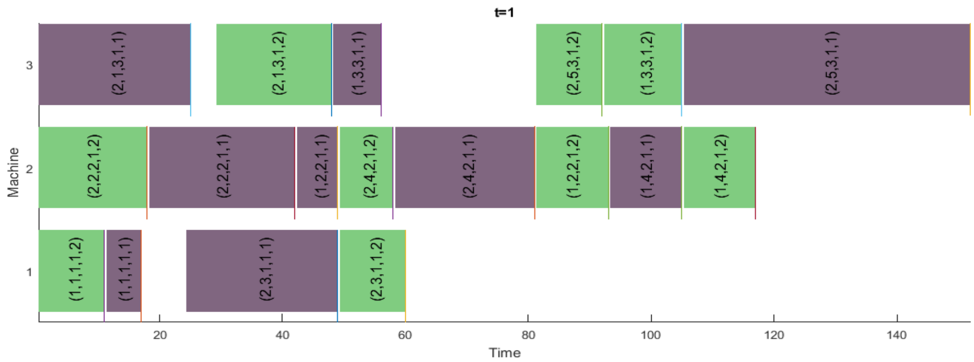

Figure 3: Scheduling of activities in the first-time interval (taking into account energy efficiency)

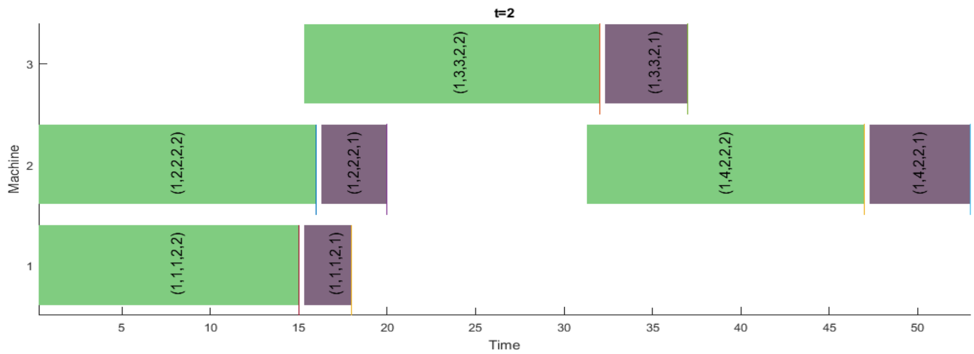

Figure 4: Scheduling of activities in the second time interval (taking into account energy efficiency)

4.1.2. Solving the sample problem without considering energy efficiency

Table 8 shows the amount of production in each time interval (without considering energy efficiency).

Table 8: Production rate in each time interval (excluding energy efficiency)

Product class	Time interval	Production method	
		manufacturing	remanufacturing
1	1	5	10
1	2	2	14
2	1	4	8
2	2	8	10

The schedule of activities for each time interval is shown as Gantt charts in Figures 5 and 6. Purple bars indicate manufacturing operations and green bars indicate remanufacturing operations. In addition, the numbers displayed on each activity are presented as ordered tuples (i,h,k,t,f) to identify activities.

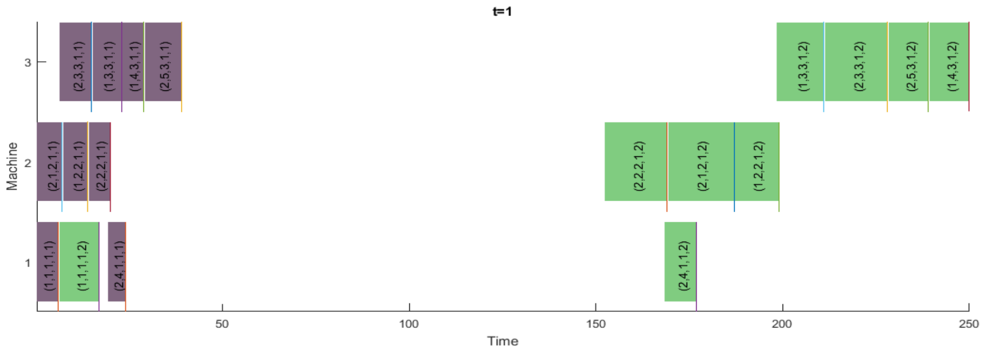

Figure 5: Scheduling activities in the first time interval (without considering energy efficiency)

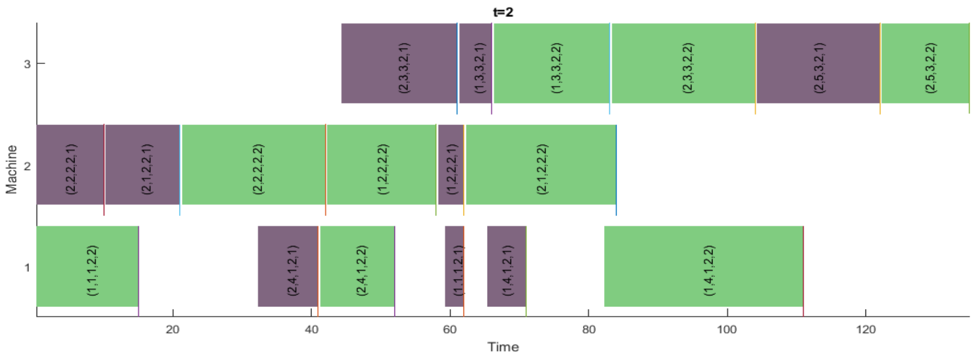

Figure 6: Scheduling of activities in the second time interval (without considering energy efficiency)

Table 9: Comparison of two solutions (with and without energy efficiency)

	Total cost	Energy cost	Production costs	Total idle time of machines
Considering energy efficiency	1480	988	422	88
Without considering energy efficiency	1980	1658	322	561

In order to evaluate the efficiency of the model for solving problems of different sizes, 28 problems have been solved with different sizes. The results of solving these problems can be seen in Table 10. By using a specified value (0.05) for relative optimality as algorithm termination criterion in the solver BARON, GAMS is unable to solve large-scale problems. From table 10, it can be seen that problems 1 to 11 have reached a relative optimality of less than 0.05 in less than 100 seconds. Problems 12 and 13 have obtained feasible solutions in 100 seconds, but these solutions have relative optimality more than 0.05 and are away from optimality. Problems 14 to 20 have been solved in 200 seconds, but these solutions also have relative optimality criterion greater than 0.05. Problems 21 to 26 have been solved in 300 seconds, but these solutions also have relative optimality criterion greater than 0.05. Finally, problems 27 and 28 have been solved in 500 seconds, but these solutions also have relative optimality criterion greater than 0.05.

Table 10: Solving problems in different sizes

Problem	K |	| T |	| H |		Time to solve the problem	Relative optimality criterion	GAMS Objective
			H1	H2			
1	2	2	3	4	1/76	0/047	1153
2	2	2	4	3	2/85	0/048	1151
3	2	2	4	4	3/64	0/048	1246
4	2	3	3	3	3/93	0/048	1451
5	2	1	3	4	4	0/047	491
6	2	1	4	5	7	0/048	577
7	3	1	4	5	12	0/04	573
8	3	2	4	3	15	0/049	1100
9	3	2	3	4	22	0/049	1132
10	2	4	4	3	24/6	0/049	2024
11	3	2	4	4	35	0/049	1239
12	3	2	4	5	100	0/14	1410
13	3	2	5	5	100	0/27	1516
14	4	2	4	4	200	0/51	1106
15	4	4	4	4	200	0/53	2295
16	4	4	5	5	200	0/64	3257
17	4	3	3	3	200	0/2	1431
18	4	3	4	4	200	0/46	1831
19	4	4	4	3	200	0/4	2082
20	4	3	3	3	200	0/44	1955
21	4	4	6	6	300	0/79	5849
22	3	4	3	4	300	0/36	2403
23	2	4	5	5	300	0/6	2955
24	3	4	5	5	300	0/47	2928
25	3	5	3	4	300	0/36	3050
26	5	4	3	4	300	0/59	2656
27	5	4	5	5	500	0/66	3032
28	2	3	4	4	500	0/48	1745

4.2. Proposed genetic-based algorithm

Due to the inability of GAMS in problem solving with the growth of problem size (Table 10), a genetic-based algorithm has been used to solve problems of different sizes. The overall process for solving the problem is shown in the flow chart of Figure 7. The process of the algorithm begins with generating random values for the amount of manufacturing and remanufacturing of products, -and the execution steps are as follows:

The initial production of each solution in each time interval is considered as U multiplied by half the demand in that time interval, where U is a random number between 0.9 and 1.1. These values are determined using trial and error. The solution for x is displayed as a three-dimensional matrix with dimensions P*T*F.

Inventory, returned inventory and inventory shortage are calculated according to the specified production amount (x) and Equations 5 and 6.
Activities of the manufactured products in each time interval are randomly assigned to machines that are capable of performing them.
The execution time of each activity (Pt) is determined according to Equation 12.
According to the activities selected for execution and the time calculated in step 4, the sequence of activities is determined randomly.
Based on the objective function of Equation 1, the generated solutions are evaluated and sorted in ascending order.
Termination condition is checked. The termination condition in this algorithm is defined as achieving the maximum allowable time or achieving a better solution than the solution obtained from the exact solution by GAMS. If the termination conditions of the algorithm occur, the algorithm will stop. Otherwise step 8 is executed.
The new solution generation procedure is performed as follows until the specified number of new solutions is generated:
Random selection of production (x) of two solutions x^1 And x^2 among the available solutions
Generating random parameters $a = uniform(\alpha, \beta)$and $b = uniform(\theta, \gamma)$
value α=-3, θ=1, β=1, and γ=2 are selected by trial and error in the present algorithm. Random number $\rho = \mathrm{round}(uniform(\mathrm{a}, \mathrm{a}+\mathrm{b}))$ is defined. Then the matrix r is defined (with dimensions equal to matrices $x^1$and x^2) that all its elements are equal ρ.
Modify the solutions $x^1$and x^2as follows:
$x_{new}^1 = \max(0, x^1 + r)$
$x_{new}^2 = \max\ (0, x^1 - r)$
The combine generated solutions with the previous solutions and go to step 2.

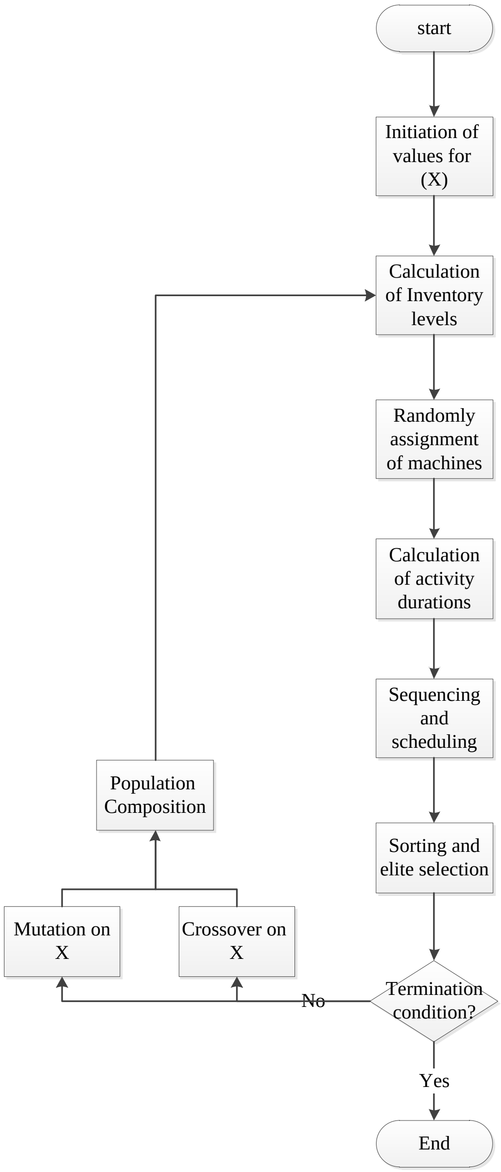

Figure 7: Flow chart of the genetic-based algorithm

Table 11 shows the performance of the proposed genetic-based algorithm along with the performance of GAMS software to solve the problem. The termination condition of the genetic algorithm is considered to be reaching the time equal to the exact solution time or achievement of the solution with the objective function less than the objective function of GAMS. From this table, it can be seen that the genetic algorithm has not been able to get better solutions than GAMS (at the specified time) for problems 1 to 6. However, the solution from the genetic algorithm in Problem 7 is better than the exact solution (in limited time). The genetic algorithm has not been able to solve the problems 8 and 9 to achieve a better objective function than GAMS (in limited time). However, problems 10 to 12 are solved more efficiently than GAMS by genetic algorithm. Problems 14 to 28 also show the efficiency of the proposed meta-heuristic algorithm for solving large problems compared to GAMS. The population size parameter (Npop) was selected according to the problem size and is reported in Table 11.

Table 11: Performance of the proposed genetic-based algorithm compared with GAMS

Problem	| K |	| T |	| H |		GAMS CPU-Time	Relative optimality criterion (optcr)	GAMS objective	GA objective	GA CPU-time	Npop
			H1	H2						
1	2	2	3	4	1/76	0/047	1153	2159	1/76	10
2	2	2	4	3	2/85	0/048	1151	2396	2/85	10
3	2	2	4	4	3/64	0/048	1246	4262	3/64	10
4	2	3	3	3	3/93	0/048	1451	1478	3/93	10
5	2	1	3	4	4	0/047	491	520	4	70
6	2	1	4	5	7	0/048	577	625	7	70
7	3	1	4	5	12	0/04	573	570	10	70
8	3	2	4	3	15	0/049	1100	1672	15	50
9	3	2	3	4	22	0/049	1132	1260	22	50
10	2	4	4	3	24/6	0/049	2024	1995	21	50
11	3	2	4	4	35	0/049	1239	1141	28	50
12	3	2	4	5	100	0/14	1410	1372	7	50
13	3	2	5	5	100	0/27	1516	1573	100	70
14	4	2	4	4	200	0/51	1106	1075	56	70
15	4	4	4	4	200	0/53	2295	2216	96	70
16	4	4	5	5	200	0/64	3257	2955	151	70
17	4	3	3	3	200	0/2	1431	1426	96	70
18	4	3	4	4	200	0/46	1831	1658	112	70
19	4	4	4	3	200	0/4	2082	1689	57	70
20	4	3	3	3	200	0/44	1955	1911	18	120
21	4	4	6	6	300	0/79	5849	5326	192	70
22	3	4	3	4	300	0/36	2403	2366	73	70
23	2	4	5	5	300	0/06	2955	2865	110	100
24	3	4	5	5	300	0/47	2928	2912	213	100
25	3	5	3	4	300	0/36	3050	3565	300	200
26	5	4	3	4	300	0/59	2656	2637	26	150
27	5	4	5	5	500	0/66	3032	2754	254	100
28	2	3	4	4	500	0/048	1745	1623	318	100

Figure 8 shows a comparison of the values of the optimal objective functions obtained by the genetic algorithm and GAMS software. It can be seen from this figure that (especially in larger problems) the genetic algorithm has in most cases been able to achieve a better solution than GAMS software in less time (According to the limited time, the solutions of GAMS are not necessarily optimal).

It should be noted that one of the conditions for terminating the genetic algorithm is to achieve a solution with a better objective function or equal to the exact solution. Therefore, due to the fact that the solutions obtained from GAMS software have been solved up to an optimality distance of 0.05 (or more), the genetic algorithm has been able to achieve better solutions in some cases (due to time constraints).

In addition, Figure 9 shows percent time improvement caused by genetic algorithm compared to GAMS software. From this figure, it can be seen that although for small problems (problems 1 to 7) no improvement has been achieved in the solution time, but in larger problems, the genetic algorithm has been able to improve the problem-solving time. In addition, the percentage of improvement in the genetic algorithm compared to GAMS software increases with problem size.

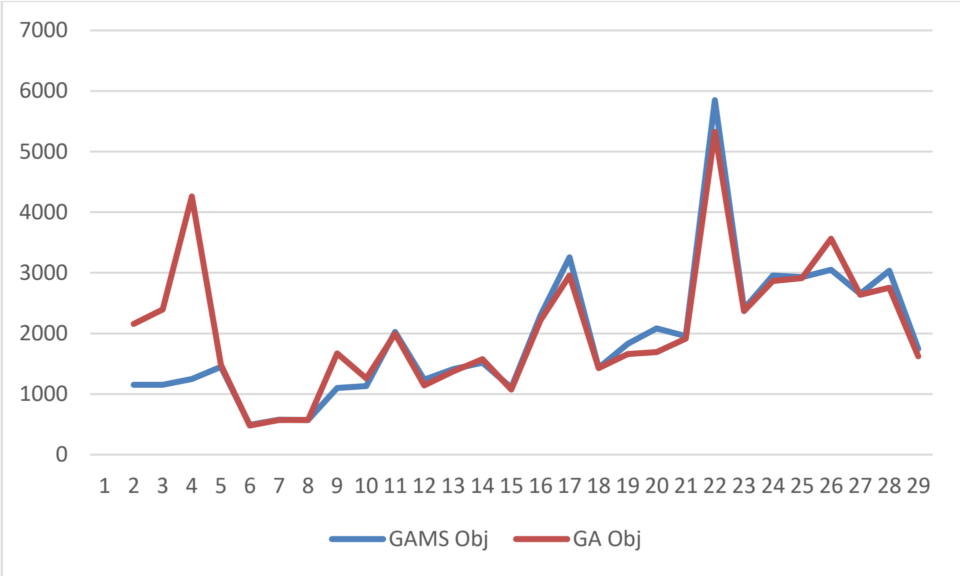

Figure 8: Comparison of the objective function obtained from the genetic algorithm and GAMS software

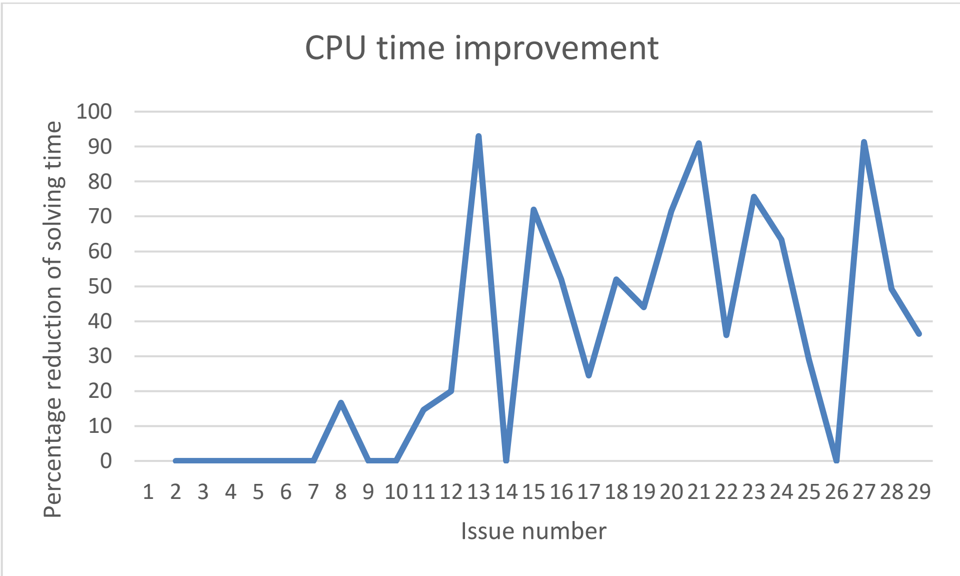

Figure 9: Percentage of reduction of CPU time using genetic algorithm

5. Conclusion

In this paper, the problem of integrated lot-sizing and scheduling in a flexible job-shop environment was modeled and solved by considering energy efficiency. To this end, a mathematical model has been developed for the integrated optimization of scheduling and lot-sizing problems. The developed model used a big bucket approach and was presented in the form of a mixed integer nonlinear programming (MINLP). To find the best solution, the model was solved by using GAMS. An illustrative example was presented and problems of different sizes where solved to assess the computational performance of the model.
To evaluate the performance of the proposed model in comparison with traditional models of integrated lot-sizing and scheduling that consider only production costs, an example without considering the cost of energies in the objective function was also solved. Solving this sample problem showed that considering energy efficiency has a significant effect on the resulting solution. More precisely, the results of solving a sample problem show that by entering the energy cost in the objective function, the cost of production increases (from 322 to 422), and the energy cost has 670 units of decrease (from 1658 to 988); however, these results in 570 units of reduction in

total cost (from 1980 to 1410). The results for the sample problem also show that the total idle time of the machines is reduced by 473 units (from 561 to 88).

Due to the NP-hard nature of the problem under study, an evolutionary algorithm based on a genetic algorithm has been developed. However, it uses a different approach to generate a new solution (instead of cross-over and mutation operators). The efficiency of this algorithm for solving larger problems is shown by solving different problems. Comparison of the values of the optimal objective function obtained from the genetic algorithm and GAMS software shows that (especially in larger problems) the genetic algorithm has been able to achieve a better solution than GAMS software in less time. In addition, although GAMS software has less time to solve the small problems, but in larger problems, the genetic algorithm has been able to improve the problem-solving time. Also, the percentage of reduction in time for genetic algorithm compared to GAMS software increases as the problem gets larger.

Considering sequence-dependent setup times, random production times or random demand, and modeling the nonlinear model as integer linear models, as well as providing exact or heuristic solutions to large-scale problems which can be solved in acceptable time may be considered as future research directions.

Managerial Insight

In this paper, we considered energy efficiency concerns in the integrated problem of lot-sizing and scheduling of flexible job-shop. Our results show that considering energy efficiency concerns can significantly help cost reduction in a flexible job-shop environment. More specifically, the energy cost decreases by taking into account the cost of energy in the objective function, while the production cost increases. However, this trade of can cause significant reduction in total operational cost. These results show that energy costs should be explicitly considered in integrated scheduling and lot-sizing problems in flexible job-shop environment. In addition, our proposed metaheuristic algorithm can help solving the problems of real-world problems with large scales.

Reference

Ahmadi, E., Zandieh, M., Farrokh, M., & Emami, S. M. (2016). A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. Computers & Operations Research, 73, 56-66.

Al-Turki, U. M., Arifusalam, S., El-Seliaman, M., & Khan, M. (2011). Resource allocation, batching and dispatching in a stochastic flexible job shop. In Advanced Materials Research (Vol. 264, pp. 1758-1763). Trans Tech Publications Ltd.

Baki, M. F., Chaouch, B. A., & Abdul-Kader, W. (2014). A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery. Computers & Operations Research, 43, 225-236.

Bitran, G. R., & Yanasse, H. H. (1982). Computational complexity of the capacitated lot size problem. Management Science, 28(10), 1174-1186.

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. Annals of Operations research, 41(3), 157-183.

Brucker, P., & Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. Computing, 45(4), 369-375.

Buschkühl, L., Sahling, F., Helber, S., & Tempelmeier, H. (2010). Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. Or Spectrum, 32(2), 231-261.

Chan, F. T. S., Wong, T. C., & Chan, L. Y. (2006). Flexible job-shop scheduling problem under resource constraints. International journal of production research, 44(11), 2071-2089.

Chan, F. T., & Choy, K. L. (2011). A genetic algorithm-based scheduler for multiproduct parallel machine sheet metal job shop. Expert systems with Applications, 38(7), 8703-8715.

Chen, J. C., Chen, K. H., Wu, J. J., & Chen, C. W. (2008). A study of the flexible job shop scheduling problem with parallel machines and reentrant process. The International Journal of Advanced Manufacturing Technology, 39(3), 344-354.

Choi, I. C., & Choi, D. S. (2002). A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups. Computers & Industrial Engineering, 42(1), 43-58.

Drexl, A., & Haase, K. (1995). Proportional lotsizing and scheduling. International Journal of Production Economics, 40(1), 73-87.

Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling—survey and extensions. European Journal of operational research, 99(2), 221-235.

Gao, J., Sun, L., & Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35(9), 2892-2907.

Gao, K. Z., Suganthan, P. N., Tasgetiren, M. F., Pan, Q. K., & Sun, Q. Q. (2015). Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion. Computers & Industrial Engineering, 90, 107-117.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. Mathematics of operations research, 1(2), 117-129.

Giglio, D., Paolucci, M., & Roshani, A. (2017). Integrated lot sizing and energy-efficient job shop scheduling problem in manufacturing/remanufacturing systems. Journal of cleaner production, 148, 624-641.

Golany, B., Yang, J., & Yu, G. (2001). Economic lot-sizing with remanufacturing options. IIE transactions, 33(11), 995-1004.

Gomez Urrutia, E. D., Aggoune, R., & Dauzère-Pérès, S. (2014). Solving the integrated lot-sizing and job-shop scheduling problem. International Journal of Production Research, 52(17), 5236-5254.

Hajibabaei, M., & Behnamian, J. (2021). Flexible job-shop scheduling problem with unrelated parallel machines and resources-dependent processing times: a tabu search algorithm. International Journal of Management Science and Engineering Management, 16(4), 242-253.

Hurink, J., Jurisch, B., & Thole, M. (1994). Tabu search for the job-shop scheduling problem with multi-purpose machines. Operations-Research-Spektrum, 15(4), 205-215.

Jans, R., & Degraeve, Z. (2008). Modeling industrial lot sizing problems: a review. International Journal of Production Research, 46(6), 1619-1643.

Kacem, I., Hammadi, S., & Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(1), 1-13.

Karimi, B., Ghomi, S. F., & Wilson, J. M. (2003). The capacitated lot sizing problem: a review of models and algorithms. Omega, 31(5), 365-378.

Mahdavi, I., Shirazi, B., & Solimanpur, M. (2010). Development of a simulation-based decision support system for controlling stochastic flexible job shop manufacturing systems. Simulation Modelling Practice and Theory, 18(6), 768-786.

Mehdizadeh, E., & Fatehi-Kivi, A. (2017). A Vibration Damping Optimization Algorithm for Solving the Single-item Capacitated Lot-sizing Problem with Fuzzy Parameters. International Journal of Industrial Engineering & Production Research, 28(1), 33-45.

Pezzella, F., Morganti, G., & Ciaschetti, G. (2008). A genetic algorithm for the flexible job-shop scheduling problem. Computers & operations research, 35(10), 3202-3212.

Piñeyro, P., & Viera, O. (2009). Inventory policies for the economic lot-sizing problem with remanufacturing and final disposal options. Journal of Industrial & Management Optimization, 5(2), 217.

Pineyro, P., & Viera, O. (2010). The economic lot-sizing problem with remanufacturing and one-way substitution. International Journal of Production Economics, 124(2), 482-488.

Richter, K., & Sombrutzki, M. (2000). Remanufacturing planning for the reverse Wagner/Whitin models. European Journal of Operational Research, 121(2), 304-315.

Richter, K., & Weber, J. (2001). The reverse Wagner/Whitin model with variable manufacturing and remanufacturing cost. International Journal of Production Economics, 71(1-3), 447-456.

Rohaninejad, M., Kheirkhah, A., & Fattahi, P. (2015). Simultaneous lot-sizing and scheduling in flexible job shop problems. The International Journal of Advanced Manufacturing Technology, 78(1), 1-18.

Rohaninejad, M., Sahraeian, R., & Nouri, B. V. (2016). Multi-objective optimization of integrated lot-sizing and scheduling problem in flexible job shops. RAIRO-Operations Research, 50(3), 587-609.

Roshani, A., Giglio, D., & Paolucci, M. (2016). A simulated annealing approach for the capacitated dynamic lot sizing problem in a closed remanufacturing system. IFAC-PapersOnLine, 49(12), 1496-1501.

Sahraeian, R., Rohaninejad, M., & Fadavi, M. (2017). A new model for integrated lot sizing and scheduling in flexible job shop problem. Journal of Industrial and Systems Engineering, 10(3), 72-91.

Sifaleras, A., Konstantaras, I., & Mladenović, N. (2015). Variable neighborhood search for the economic lot sizing problem with product returns and recovery. International Journal of Production Economics, 160, 133-143.

Teunter, R. H., Bayindir, Z. P., & Den Heuvel, W. V. (2006). Dynamic lot sizing with product returns and remanufacturing. International Journal of Production Research, 44(20), 4377-4400.

Teunter, R., Tang, O., & Kaparis, K. (2009). Heuristics for the economic lot scheduling problem with returns. International Journal of Production Economics, 118(1), 323-330.

Wang, N., He, Z., Sun, J., Xie, H., & Shi, W. (2011). A single-item uncapacitated lot-sizing problem with remanufacturing and outsourcing. Procedia Engineering, 15, 5170-5178.

Xing, L. N., Chen, Y. W., Wang, P., Zhao, Q. S., & Xiong, J. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. Applied Soft Computing, 10(3), 888-896.

Xiong, W., & Fu, D. (2018). A new immune multi-agent system for the flexible job shop scheduling problem. Journal of Intelligent Manufacturing, 29(4), 857-873.

Yang, J., Golany, B., & Yu, G. (2005). A concave-cost production planning problem with remanufacturing options. Naval Research Logistics (NRL), 52(5), 443-458.

Zarrouk, R., Bennour, I. E., & Jemai, A. (2019). A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem. Swarm Intelligence, 13(2), 145-168.

Zhang, F., Mei, Y., Nguyen, S., & Zhang, M. (2020). Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling. ieee transactions on cybernetics, 51(4), 1797-1811.

Zhang, G., Gao, L., & Shi, Y. (2011). An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications, 38(4), 3563-3573.

This article can be cited: Osati, A., Mehdizadeh, E., Ebrahimnejad, S., (2022). Lot-Sizing and Scheduling in a Flexible Job-Shop Environment Considering Energy Efficiency. Journal of Industrial Engineering and Management Studies, Vol. 9, No.2, pp. 129-147.