Expert Systems With Applications 251 (2024) 123983

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A novel MIP model and a hybrid genetic algorithm for operation outsourcing in production scheduling with carbon tax policy

Melis Alpaslan Takan

Faculty of Engineering, Department of Industrial Engineering, Bilecik Şeyh Edebali University, TR, 11100 Bilecik, Turkey

ARTICLE INFO

Keywords:
Production scheduling
Operation outsourcing
Hybrid algorithm
Carbon footprint
Carbon tax

ABSTRACT

In the defense industry, outsourcing has become crucial during production stages, primarily because certain tasks involve complex operations that may exceed the capabilities of in-house machinery. This is often influenced by factors such as technical competencies and high production costs. On the other side, recent shifts in climate patterns and growing environmental concerns have led to a significant impact on scheduling decisions, especially influenced by the carbon footprint of production factories. Therefore, it is important for manufacturing factories to evaluate both economic and environmental costs. A Mixed Integer Programming (MIP) model is formulated to solve the real-life production scheduling problem including machine costs, operation outsourcing and carbon footprint of the schedule with minimizing total costs. Since the mathematical model has difficulty in finding solutions as the problem size increases, a Hybrid Genetic Algorithm (HGA) which combines Genetic Algorithm (GA) with Iterated Local Search (ILS) procedure is proposed to solve especially medium-sized and large-sized problems. The efficiency of the proposed algorithm is demonstrated through randomly generated test problems. Obtained results indicate the superiority of the proposed HGA over GA, showcasing its ability to generate high-quality solutions and reduce overall costs in a reasonable solution times.

1. Introduction

Scheduling problems within production processes are recognized as NP-hard combinatorial optimization problems, extensively investigated in various machine environments such as parallel machine, flow shop, or job shop scheduling as documented in the literature. Since the problem types variates, objectives and additional constraints also differs. The outsourcing option of jobs/operations has been an extensively discussed topic in the literature spanning various fields including engineering, economics, and management. Detailed cost analysis based on both economic and environmental criterias for operation outsourcing or not decision has become crucial especially for the production factories which have critical tasks to complete on critical due dates. Outsourcing option of operations generally arise because of the limited capacity of in-house machines. With the integration of outsourcing option, the following issues should also be considered (Qi, 2008): the selection of the outsourced/in-house operations; the scheduling of the in-house operations; the scheduling of the outsourced ones; and the transportation of the outsourced operations if possible (Guo & Lei, 2014). A manufacturer factory may prefer to assign all its operations to a subcontractor or to assign only some of operations for outsourcing by considering its other constraints. Skillful outsourcing can lead to reduced production timelines, lowered overall expenses, and enhanced organizational flexibility (Lee & Choi, 2011). The logistical difficulty (or high logistical cost) of transporting goods between the main manufacturer (in-house) and subcontractor is one of the key element of this problem. This approach gained recognition as a business strategy in 1989 (Drucker, 1989). A general outsourcing job model of single-stage scheduling problem was suggested by Qi (2008) with transportation considerations in the scheduling literature. Mokhtari et al. (2012) studied production scheduling with outsourcing scenarios and proposed a novel MIP model for these scenarios. The authors solved the problem by using the team process algorithm which is an effective evolutionary algorithm. Izadi et al. (2020) studied a new integrated production scheduling, routing, inventory and outsourcing (job) problem for parallel machine scheduling including setup times. They proposed a mixed integer linear programming model to minimize the total cost and applied hybrid algorithm by incorporating a genetic algorithm (GA) with dominance properties. Homayouni and Fontes (2021) studied a hybrid GA for integrated production and distribution scheduling problem with job outsourcing. They applied hill climbing algorithm which is similar with iterative local search algorithms that attempt to improve the incumbent

E-mail address: melis.alpaslan@bilecik.edu.tr.

https://doi.org/10.1016/j.eswa.2024.123983
Received 5 January 2024; Received in revised form 21 March 2024; Accepted 12 April 2024
Available online 16 April 2024

solution by searching in its neighborhood for a better solution. Pang et al. (2024) studied a bi-objective low-carbon economic scheduling model of a cogeneration system. To solve the problem, they improved bare-bones multi-objective particle swarm optimization to obtain the Pareto front of the low-carbon scheduling model. Ahmadizar and Amiri (2018) addressed a two-machine flow shop scheduling problem where jobs are released intermittently and outsourcing is allowed with the objective is to select a subset of jobs to be outsourced. Li et al. (2022) investigated a flexible job shop scheduling problem (FJSSP) with outsourcing operations with job priority constraints and proposed a sequence-based mathematical model by minimizing the weighted overdue days.

When environmental costs are examined in the scheduling literature, the significance of the carbon footprint becomes a pivotal factor in the manufacturing sector. In practical scenarios, factories might consider various objectives when making scheduling decisions like minimizing makespan, lateness, tardiness, and etc., alongside the consideration of the carbon footprint. The carbon footprint serves as a measure of the overall CO_2 emissions directly and indirectly linked to energy consumption (Liu & Huang, 2014). Renewable energy is an alternative option to non-renewable energy to reduce the carbon footprint of the production schedule (Wu et al., 2018). Given that manufacturing factories are acknowledged as major contributors to global warming, there is increasing pressure on them to reduce their carbon footprint as quick as they can. Recent studies demonstrated that energy-efficient scheduling is a highly effective and important approach to reduce the energy consumption to solve different types of scheduling problems. Therefore, many authors have considered energy criteria including additional constraints in their studies. The first study on energy-efficient scheduling of a single machine was conducted by Mouzon & Yildirim (2008). Fang et al. (2011) proposed a new mathematical programming model of the flow shop scheduling problem (FSSP) by considering the peak power load, energy consumption, and associated carbon footprint. Zhang et al. (2015) studied a FJSSP to minimize makespan, working load and carbon emissions caused by auxiliary material consumption and tool wear. A carbon footprint model of multi-job processing is established to quantify the carbon emission of different scheduling plans. Liu et al. (2017) proposed a multi-objective optimization model for minimizing carbon footprints of all products and makespan. To solve the proposed model, they designed a hybrid fruit fly optimization algorithm. Piroozfard et al. (2018) proposed a multi-objective FJSSP problem to minimize total carbon footprint and total late work criterion. Wei et al. (2022) proposed a novel energy-aware estimation model to compute different energy consumptions for different running conditions of a machine. They also formulated a multi-objective model for the dual FJSSP to minimize the makespan and total energy consumption. Rahman et al. (2022) considered an extension of the resource-constrained project scheduling problem and they represented as a bi-objective optimization problem with minimizing the total cost of the project and its carbon footprint. Trevino-Martinez et al. (2022) studied energy-carbon footprint optimization in sequence-dependent production scheduling. They developed a low energy-carbon cost sequence dependent job scheduling optimization model under a MIP formulation. Sagar et al. (2023) proposed an energy-aware production schedule model for flexible manufacturing systems which aims to minimize energy costs and carbon tax. They also added to the machine on/off strategy to reduce idle energy. He et al. (2023) developed a two-stage optimization strategy is adopted to achieve collaborative optimization of cutting parameters and production scheduling with integrated carbon footprint. Chen et al. (2023) considered an unrelated parallel machine scheduling problem with variable maintenance based on machine reliability to minimize the maximum completion time with using hybrid discrete spider monkey optimization algorithm.

Looking at these studies, growing concerns about climate change have transformed environmental practices and production strategies into more environmentally friendly regulations. In this context, it would serve countries' sustainable production goals and zero carbon targets if production factories choose more environmentally friendly methods within the scope of the energy sources they use (Atilgan & Azapagic, 2016) and if possible, develop strategies for this purpose. Therefore, the creation of deterrent penalties and the correct use of the carbon tax structure to be paid, is very crucial. This carbon tax is firstly introduced by Finland in the 1990s (Trevino-Martinez et al., 2022). A carbon tax operates as a form of pollution tax, acting as a mechanism in which consumers of carbon-based fuels assume financial accountability for the climate-related damage caused by the emission of carbon dioxide into the atmosphere. When this tax is set at a sufficiently high level, it transforms into a potent economic disincentive, compelling a shift towards a cleaner economy (Alegoz et al., 2021). Renewable energy plays a crucial role in reducing carbon footprints because it enables the

Table 1
Overview of the relevant literature.

Selected reference	Machine environment	Outsourcing option		Objective Function(s)					Solution Method
		Job	Operation	Tardiness	Makespan	Carbon cost	Energy cost or consumption	Other objective(s)	
Liu et al. (2023)	JSSP	√			√		√	√	SLISA
Bektur (2023)	DFSSP			√					NSGA-II
Xu et al. (2023)	DFJSSP		√		√	√		√	MO-HGABCA
Li et al. (2022)	FJSSP				√		√		HICA
Xu et al. (2021)	DFJSP		√		√	√		√	Hybrid GA&TS
Bouzid et al. (2021)	OAS			√		√			DIM
Babaee Tirkolaee et al. (2020)	FSSP	√					√	√	AFSA
Zheng et al. (2020)	FSSP				√		√		Hybrid ACO
Liu (2019)	FSSP	√			√			√	Hybrid VNS
Ramezanian et al. (2019)	GP-FSSP				√		√		CH
Lei et al. (2018)	FSSP			√			√		TLBO
Mokhtari & Hasani (2017)	FJSSP				√		√	√	EA

ACO: ant colony optimization; AFSA: artificial fish swarm algorithm; CH: constructive heuristic; DFJSSP: distributed flexible job shop scheduling problem; DFSSP: distributed flow shop scheduling problem; DIM: dynamic island model; EA: evolutionary algorithm; FJSSP: flexible job shop scheduling problem; FSSP: flow shop scheduling problem; GA: genetic algorithm; GP: green permutation; HGABCA: hybrid genetic artificial bee colony algorithm; HICA: hybrid imperialist competitive algorithm; MA: memetic algorithm; MO: multi objective; NSGA-II: non-dominated sorting genetic algorithm; PS: production scheduling; SLISA: self-learning interior search algorithm; TLBO: teaching–learning-based optimization; TS: tabu search; UPMS: unrelated parallel machine scheduling; VNS: variable neighborhood search.

generation of electricity without emitting significant amounts of greenhouse gases. This energy and carbon tax policies are also complementary strategies for reducing carbon footprints and mitigating climate change. Carbon tax policies directly target carbon footprints by placing a price on carbon emissions (Gómez et al., 2023; Ghorbanzadeh & Ranjbar, 2023). Carbon taxes provide a market-based mechanism to internalize the environmental costs of carbon emissions, while investments in renewable energy infrastructure accelerate the transition away from fossil fuels and towards sustainable energy sources which is very critical in today's world. Together, these policies can help countries achieve their emissions reduction targets and build a more resilient and sustainable future (Yağmur & Kesen, 2024).

In addressing the examined large-scale scheduling problems, metaheuristics and hybrid algorithms are predominantly employed as the solution methods. In the scheduling literature, there are numerous methods that have been proposed by scholars. The selection of an intelligent and effective optimization method with superior performance to solve the production scheduling problem with operation outsourcing becomes crucial. In particular, single solution-based metaheuristic methods such as Tabu Search (TS) and Simulated Annealing (SA) may be insufficient for solving NP-hard scheduling problems. On the other side, Genetic Algorithm (GA) has garnered worldwide attention. GAs are powerful optimization techniques for scheduling problems, offering advantages such as global exploration, population diversity, parallelism, robustness, adaptability, and efficient handling of large search spaces (Bi et al., 2020; Lu et al., 2017). GAs can be also easily adapted to different scheduling problems by customizing the genetic operators (crossover, mutation, selection) and fitness functions. This flexibility allows them to incorporate problem-specific knowledge and constraints effectively. Although GA demonstrates robust global search capabilities, it may encounter limitations when dealing with complex scheduling problems if it is applied solely to the specified problem. Therefore, many papers have shifted focus towards Hybrid Genetic Algorithms (HGA) to solve scheduling problems (Tutumlu & Saraç, 2023; Sun et al., 2023; Dai et al., 2022; Fekih et al., 2020; Xu et al., 2021; Qin et al., 2016; Chang et al., 2015). Iterated Local Search (ILS), operates by iteratively exploring the solution space, refining the solutions in each iteration by applying local search procedures and incorporating perturbation mechanisms to escape from local optima. ILS can also incorporate various metaheuristic techniques to guide the search process and efficiently explore large solution spaces without exhaustive enumeration in a reasonable computational time. There are many papers which are proposed different mathematical models and solved large-sized scheduling problems efficiently by using ILS (Avci, 2023; Oladzad-Abbasabady et al., 2023; Khedim et al., 2022; G.-de-Alba et al., 2022; Queiroga et al., 2021). By leveraging both the strengths of GA and ILS, HGA emerges as a superior choice for addressing the production scheduling problem with operation outsourcing described in this paper.

In Table 1 below, various studies in different machine environments are summarized based on some key characteristics such as ousourcing option (job or operation), objective function which consists of tardiness, makespan, carbon (emission) cost, energy consumption cost or others, and the solution method studied. As it is observed, the majority of these researches focused on minimizing makespan as the primary objective function (Liu et al., 2023; Li et al., 2022). Recently, carbon costs also considered in some papers for different machine environments (Zandi et al., 2020; Xu et al., 2021; Bouzid et al., 2021). As it is seen, there is a limited number of studies in the literature that have explored the scheduling considerations of both outsourcing operation and energy/carbon costs. Among them, Xu et al. (2023) proposed a mathematical model for operation outsourcing in a distributed flexible job shop problem. They tried to minimize total costs, makespan and carbon emission by hybrid genetic artificial bee colony algorithm. Liu et al. (2023) studied energy-aware job shop scheduling problem with outsourcing option to minimize the sum of completion time cost, job outsourcing cost and energy consumption cost. There are only a few studies which considers carbon footprint in the scheduling for different machine environments (see Fang et al., 2011; Xu et al., 2021; Trevino-Martinez et al., 2022; Rahman et al., 2022). As far as author awares, studies in the literature have not focused on the outsourcing of operations, the overall costs of machines and the energy sources used for the electricity consumption of the machines (factories) with considering the carbon tax.

This research is motivated by the production scheduling problem faced by a manufacturing factory of the defense industry in Turkey. The main objective is to examine the impact of outsourcing operations by considering the carbon footprint of the machines with the energy sources they use (renewable or non-renewable) and introducing a carbon tax to convert the objective into the cost function while concurrently considering the total economic costs, encompassing both total penalty cost of tardy jobs and total machine costs. In this respect, this study stands out from the literature. In this paper, a mixed integer linear programming model is developed for the problem at hand. The efficiency of the model at solving small-sized problems to optimality is presented. Due to the NP-hard nature of the problem, feasible solutions could not be found within reasonable time frames using the optimization solver as the dimension of the problem increases. Therefore, a Hybrid Genetic Algorithm (HGA) which integrates Genetic Algorithm (GA) and Iterated Local Search (ILS) is utilized for larger-sized problems. The effectiveness of HGA over classical GA is evident in nearly all obtained solutions. To emphasize the role of the carbon tax on the model, an analysis is conducted on a small-sized problem, and the detailed results obtained are presented (see Section 4.4). It is believed that the proposed MIP model will advance the literature by analyzing both economic and environmental consequences by providing high quality solutions.

The rest of the paper is outlined as follows. In Section 2, problem definition and the developed model is given. Section 3 presents GA, proposed HGA and its components in detail. Test problems and computational results are interpreted in Section 4. Finally, concluding remarks are provided in Section 5.

2. Problem definition and the developed mathematical model

Renewable energy sources used by manufacturing factories have become an important factor especially for zero emission targets and the use of these sources is encouraged by different countries. Renewable energy has the biggest role in greenhouse gas reduction targets. Among these energy sources, the proportion of solar energy in installation is the highest (International Energy Agency, 2023). Renewables like solar and wind will continue to expand rapidly, becoming the number one source of new power generation by 2025. On the other hand, fossil fuel still provides around 80 % of global energy demand. The type of fossil fuel used in the industrial factories varies by country, with coal being the main source in some countries, while in others both natural gas and coal are used in a mixed structure. When the literature is examined in the context of non-renewable energy, coal is used to be the primary energy source in Turkey until the 1970's and remains the country's primary domestic energy source (International Energy Agency, 2023).

Given the complex nature of production operations and the project-based critical characteristics within defense sector, outsourcing operations and identifying specialized subcontractors are crucial. The carbon footprint of each machine owned by the factories has been assessed by considering the energy sources used. Processing times for identical operations can vary across different machines. Carbon footprint coefficient also differs due to distinctions of the factories' (in-house machines or subcontractors) environmental sustainability, categorized as either using "renewable energy" or "non-renewable energy". In this context, the fact that the energy source used by the factory varies, such as coal, natural gas, solar, etc., will significantly change the environmental costs. In this problem, the factory with in-house machines is assumed to use the solar energy as a renewable energy source. The subcontractor machines use only coal or a mixture of coal and natural gas. Accordingly,

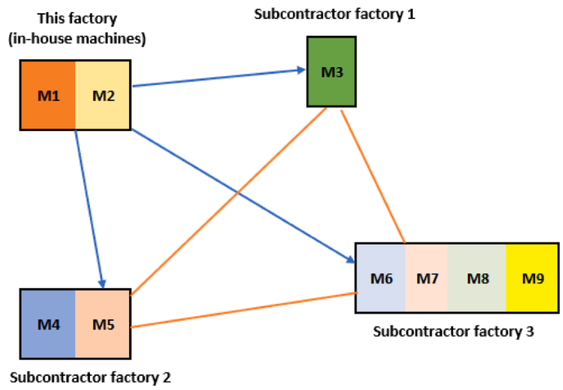

Fig. 1. An example of in-house machines and subcontractors.

carbon footprint coefficient is defined for each machine in the factories and this parameter is multiplied by carbon tax and converted into a cost function.

Since all machines may not process every operation, typically due to technical constraints, the problem under consideration incorporates machine eligibility restrictions (Bektur & Saraç, 2019). Also, the fact that in-house machines use renewable energy (solar energy in this case) while subcontractors that have machines which uses nonrenewable energy, complicates the problem. Considering the carbon footprint to lessen the environmental impacts on the scheduling plan which minimizes the environmental damage is one of the factory's objectives. For instance, assume that there is one machine in the factory that uses the solar energy with a carbon footprint coefficient of 0.05; whereas in subcontractor machines, electricity is obtained from a source using both coal and natural gas energy, resulting in a carbon footprint value average of 0.70. In this scenario, when focusing solely on the environmental consequences, it is essential to use in-house machine. Beside this, delays in defense industry projects can also lead to critical consequences; therefore, substantial penalty costs have been established as a deterrent to prevent significant job delays. Both in-house and subcontractor machines have specific machine costs. As the factory's machines operate on renewable energy, the cost of using these machines for the specified operations is relatively high. Other objective is to formulate a scheduling plan which minimizes overall costs for these machines whenever feasible by looking at the penalty costs for job delays.

In the studied problem, the objective function aims to minimize the total cost which has three different costs. The first term of the total cost function attempts to prevent job delays by imposing a penalty cost, as the failure to meet critical jobs at critical times. To demonstrate this, high penalty costs indicates critical jobs. The second term represents the total costs of machines. For in-house machines, this cost represents the maintanence cost, and since green technology machines have high investments, in-house machines have high costs than subcontractor-owned machines which are not environmentally-friendly. The third term involves multiplying the carbon footprint by a carbon tax. Due to the usage of different energy sources by subcontractor-machines, they have different carbon footprint values. Thus, the discussion revolves around strategies or investments (such as which energy sources can be used) that can reduce environmental damage, considering with the total economic costs, and finding some solutions to translate to the environmentally friendly options. To reflect the characteristics of the studied problem, an MIP model is established based on Mokhtari et al. (2012) but different by some assumptions based on a real-life manufacturing problem in the defense industry, described below.

- The subcontractor structure is treated as machines. For instance, while in-house factory may have 2 machines, a subcontractor may have 3 machines. As Fig. 1 shows, subcontractor factories can have different number of machines. The model evaluates on a machine-by-machine basis. Distances between machines are ignored within the same factory. The carbon footprint and electricity consumption of the machines are directly proportional to the energy source used by the factory they are connected to.
- Preemption of the operations is not allowed.
- Each job has a specific number of operations.
- A machine can process at most one operation at a time.
- All processing times of operations are deterministic.
- Each machine may not perform every operation of each job. This situation is represented in the model by the parameter e_{ijm}, which indicates whether machine m can perform operation j of job i or not (machine eligibility).
- In-house machines use renewable energy sources while subcontractors use non-renewable energy sources.

The sets, indeces, parameters, and decision variables of the developed model for the problem are presented below.

Notations:

Sets	
I	Set of jobs $i, u \in I$
J	Set of operations $j, v \in J$
M	Set of machines (in-house machine or subcontractor's machine) $m, s \in M$
H	Set of sequences $h, d \in H$
Parameters	
e_{ijm}	Machine eligibility (technical competence or capability of a machine) 1, if an operation j of job i can be processed on machine m; 0, otherwise
p_{ijm}	Processing time of machine m for operation j of job i
dd_i	Due date of job i
α_i	Penalty cost of job i
mac_{ijm}	Cost of machine m for processing the operation j of job i
$Carb_m$	Carbon footprint of machine m
$Cons_m$	Electricity consumption of machine m
C_{tax}	Carbon tax value
Decision variables	
$Carb_{max}$	Carbon footprint of the schedule
$Tard_i$	Tardiness of job i
t_{ij}	Starting time of operation j of job i
c_{ij}	Completion time of operation j of job i
$Cmax_i$	Maximum completion time of job i
y_{ijmh}	1, if an operation j of job i is processed on machine m in sequence h; 0, otherwise

Mathematical model:

$$\min z = f_1 + f_2 \tag{1}$$

$$f_1 = \sum_i \alpha_i Tard_i + \sum_i \sum_j \sum_m \sum_h mac_{ijm} p_{ijm} y_{ijmh} \tag{2}$$

$$f_2 = C_{tax} Carb_{max} \tag{3}$$

$$\sum_m \sum_h y_{ijmh} = 1 \ \forall i, j \tag{4}$$

$$\sum_i \sum_j y_{ijmh} \leq 1 \ \forall m, h \tag{5}$$

$$\sum_h y_{ijmh} \leq e_{ijm} \ \forall i, j, m \tag{6}$$

$$t_{ij+1} - t_{ij} - p_{ijs} - L_{sm} + M(2 - \sum_d y_{ij+1md} - \sum_h y_{ijsh}) \geq 0 \ \forall i, j, m, s; m \neq s \tag{7}$$

$$t_{ij+1} - t_{ij} - p_{ijm} + M(2 - \sum_d y_{ij+1md} - \sum_h y_{ijmh}) \geq 0 \ \forall i, j, m \tag{8}$$

$$t_{ij} - t_{uv} - p_{u,v,m} + M\left(2 - y_{ijmd} - y_{uvmh}\right) \geq 0 \ \forall i, j, u, v, m, h, d; i \neq u; j \neq v; d > h \tag{9}$$

	JOB 1			JOB 2			JOB 3		
initial operations	1	1	1	2	2	2	3	3	3
random number generation for OS	0.1	0.52	0.9	0.8	0.69	0.75	0.2	0.3	0.78
sorting part	0.1	0.2	0.3	0.52	0.69	0.75	0.78	0.8	0.9
transformed result	1	3	3	1	2	2	3	2	1
transformed operation numbers	O_{11}	O_{31}	O_{32}	O_{12}	O_{21}	O_{22}	O_{33}	O_{23}	O_{13}

Fig. 2. The OS vector.

$$t_{uv} - t_{ij} - p_{i,j,m} + M\big(2 - (1 - y_{uvmh}) - (1 - y_{ijmd})\big) \geq 0 \ \forall i,j,u,v,m,h,d; i \neq u; j \neq v; h > d \tag{10}$$

$$c_{ij} \geq t_{ij} + \sum_m \sum_h p_{i,j,m} y_{ijmh} = 1 \ \forall i,j \tag{11}$$

$$Cmax_i \geq c_{ij} \ \forall i,j \tag{12}$$

$$Tard_i \geq Cmax_i - dd_i \quad \forall i \tag{13}$$

$$\sum_i \sum_j \sum_m \sum_h Carb_m Cons_m p_{i,j,m} y_{ijmh} \leq Carb_{max} \tag{14}$$

$$Carb_{max}, Cmax_i, c_{ij}, t_{ij} \geq 0 \quad \forall i,j \tag{15}$$

$$y_{ijmh} \in \{0,1\} \quad \forall i,j,m,h \tag{16}$$

(1) represents the objective function which is the sum of the costs described in equations (2) and (3); where (2) is the total tardiness cost of jobs and total machine costs; (3) is the carbon cost (tax) based on carbon footprint of the all production process. Constraint (4) describes that each operation of every job must be assigned to exactly one machine. (5) indicates that at most one operation of a job can be assigned to a machine in a given sequence. Constraint (6) states that; if the machine is not eligible, then no operation is assigned to that machine; if it is eligible, an operation assignment can be made. Equation (7) describes that if the successive operations of a job (such as j and $j+1$) are operated on different machines, the transportation time must be explicitly incorporated in the calculation of the start time. (If the operations are performed on different machines within the same factory, the transportation time is considered to be 0). Eq. (8) calculates the start times of consecutive operations when they are assigned to the same machine. Constraints (9) and (10) indicate the starting times by considering precedence relation of operations which are processed on different machines. Eq. (11) calculates the completion time of operation j of job i. Eq. (12) is the maximum completion time for each job. Constraint (13) indicates the tardiness of job i. Constraint (14) represents the carbon footprint of the schedule. Constraints (15) and (16) are sign constraints of the decision variables.

3. Proposed hybrid genetic algorithm

Production scheduling problems are classified as NP-hard, which means that finding an optimal solution to these problems becomes computationally infeasible as the size of the problem increases. Therefore, the hybridization strategy is primarily utilized to address the limitations of GA, specifically enhancing its local search capability and mitigating premature convergence. The hybrid genetic algorithm (HGA) suggested integrates two distinct algorithms, namely, Genetic Algorithm (GA) and Iterated Local Search (ILS). It aims to enhance the overall efficiency and potency of GA. In this context, all randomly generated test problems are solved both algorithms to make a wide comparison between the obtained solutions. In the following sub-sections, the solution representation of HGA, the creation of the initial population, calculation of the fitness function, ILS algorithm; selection, crossover, and mutation operators are explained in detail.

3.1. Classical genetic algorithm

Genetic algorithm (GA) operates on a population of potential solutions applying the principle of survival of the fittest to produce better approximations to a solution. GA is is a type of population-based metaheuristic designed to seek out effective solutions for intricate problems that generally lack exact solutions (Talbi, 2009).

The steps of GA are given below.

Initialization step. First, the chromosome structure should be determined for the problem being studied. Selection, crossover and mutation methods are chosen. Population size, crossover probability, mutation probability and termination criteria are determined.

Step 1. Get initial population randomly.
Step 2. Evaluate the population.
Step 3. While maximum generation number is not reached; Do Steps below:
Step 4. Select parents from current generation using roulette wheel selection.
Step 5. Apply crossover operators to operation sequence vector and machine assignment vector according to crossover rate, operators are POX crossover and uniform crossover respectively.
Step 6. Apply mutation operators to operation sequence vector and machine assignment vector according to the mutation rate.
Step 7. Evaluate new generation.

3.1.1. Creating the initial population

The initial population serves as the starting point for the evolutionary process, and its composition can significantly impact the performance of the algorithm. If the initial population lacks diversity, the genetic algorithm might converge too quickly to a suboptimal solution. To enhance diversity in the initial population, researchers apply some techniques such as random initialization, seeding with a variety of solutions, or incorporating domain-specific knowledge to ensure a well-distributed starting point. Balancing diversity with a representative coverage of the solution space is critical for achieving effective exploration throughout the evolutionary process. In this paper, random assignment is used, in which the operations randomly assigned to the machines. The detailed information is given in subsections below.

3.1.2. Solution representation

In this subsection, the structure of the chromosome employed will be elaborated which is influenced from the study of Li et al. (2022). The chromosome consists of two vector representation. First vector indicates the operation sequence (OS) vector and the second vector is the machine assignment (MA) vector. The OS vector reflects the order in which operations are processed, with values generated randomly within the range of 0 to 1. It is necessary to arrange these random values in ascending order and apply a transformation, as illustrated in Fig. 2, to establish the processing order. This vector enables the determination of the

Table 2
Eligible machines for a small example.

Job number	Operation number	Eligible Machines
1	1	M1,M3,M4
1	2	M2,M3,M4
1	3	M1,M2,M3,M4
2	1	M1,M2,M3,M4
2	2	M1,M2
2	3	M1,M2,M3,M4
3	1	M1,M2
3	2	M2,M4
3	3	M1,M2,M3,M4

processing sequence for operations across all jobs. The MA vector denotes the machine assigned to each operation which is randomly assigned with considering the eligibility property of the machine.

Let consider a small example for the solution representation. For instance, there are 3 jobs and 3 operations with 4 possible machines. The problem parameters are given in Table 2. The OS and MA vectors are illustrated in Fig. 2 and Fig. 3, respectively.O_{ij} indicates the operation j of job i.

3.1.3. Calculation of the fitness function

After creating the initial population, fitness function of the solution must be evaluated based on the total tardiness cost of each job. Then, machine costs and carbon taxes of the assigned machines are added to this cost function. Hence, fitness function calculation consists of two steps: First, calculating tardiness for each job. Second, calculating machine costs and carbon footprint costs by considering the carbon tax value.

3.1.4. Selection

In this paper, roulette wheel selection is used for parent selection in GA. This method allows selection of better fit individuals as parents proportionately to their fitness function values. The process is basically as follows: all fitness values in the population are getting normalized and accumulated. Thus, all values have normalized values between 0 and 1, and they are all have accumulated values. For instance, 3rd individual chromosome value is 0.34 and 4th one is 0.12, to accumulate 4th one which can be written as 0.46. That allows setting up intervals for all individuals. Then a random number between 0 and 1 generated, that number is used to select the individual as a parent.

3.1.5. Crossover

Crossover is one of the main evolutionary operators in GA. It is a genetic operator that combines two parent solutions to create one or more offspring. Since each chromosome contains two vectors, two crossover operators were used which are called Precedence Preserving Order-based Crossover (POX) applied for OS vector and Uniform Crossover (UC) applied for the MA vector, respectively (Moghadam et al., 2014).

POX Crossover: This is a technique commonly applied for permutation-based problems, such as scheduling. It involves selecting a subset of genes from each parent and preserving their order in the crossover. Sequencing operator vector refers to a representation of a solution, where the order of operations is encoded. The crossover operation is applied to the sequencing vectors of two selected chromosomes. The POX crossover is applied on the operation sequencing vector of two selected chromosomes for obtaining two offspring in each generation without changing the machine assignment part. It implies that the genetic information related to which machine each operation is assigned to remains unchanged during the crossover but the focus of the crossover is on the order or sequencing of operations. The application detail of POX is given below:

Step 1. Selection of Sub-Job (subj): A sub-job called subj, are randomly selected from the set of all jobs. This step is essential for determining which portions of the parents (Parent1 and Parent2) will be used to create the offspring.

Step 2. Copying Genes to Offspring (Child1 and Child2): Genes (representing operations) that correspond to subj is copied from Parent1 and Parent2 to the offspring chromosomes Child1 and Child2, respectively. The positions of these genes in Child1 and Child2 are determined by their positions in Parent1 and Parent2.

Step 3. Deletion of Genes from Parents: The genes that were copied to the offspring are then removed from their original positions in the Parent1 and Parent2. This ensures that there are no duplicate genes in the offspring chromosomes.

Step 4. Filling Empty Positions in the Offspring: The empty positions in Child1 and Child2, resulting from the removal of subj gene, are filled with the remaining genes from Parent2 and Parent1, respectively. The order of filling is maintained as per the order of genes in the parents.

Fig. 4. below describes the POX crossover.

Uniform Crossover (UC): This operator aims to generate two offspring in each generation while preserving the operation sequencing part. The steps of UC is given in the following:

Step 1. Make Sorting based on Operations. (For the two parent chromosomes, denoted as P1 and P2, the machine assignment vectors are sorted based on operations. The goal is to establish a consistent order of operations across the chromosomes.)

	JOB 1			JOB 2			JOB 3		
initial operations	1	1	1	2	2	2	3	3	3
random number generation for OS	0.1	0.52	0.9	0.8	0.69	0.75	0.2	0.3	0.78
sorting part	0.1	0.2	0.3	0.52	0.69	0.75	0.78	0.8	0.9
transformed result	1	3	3	1	2	2	3	2	1
transformed operation numbers	O_{11}	O_{31}	O_{32}	O_{12}	O_{21}	O_{22}	O_{33}	O_{23}	O_{13}

Fig. 3. The MA vector.

Parent 1	1	3	2	1	1	2	3	3	2	3	2	1
Parent 1	3	1	2	2	1	3	2	1	3	2	1	3
Child 1	1	3	2	1	1	2	3	2	3	2	3	1
Child 2	3	1	2	2	1	3	3	1	2	3	1	2

Fig. 4. POX Crossover.

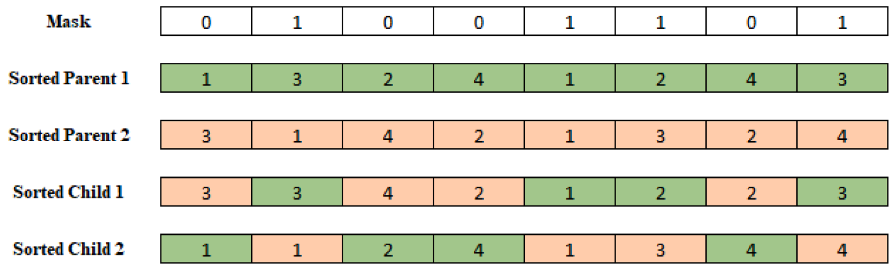

Fig. 5. Uniform Crossover.

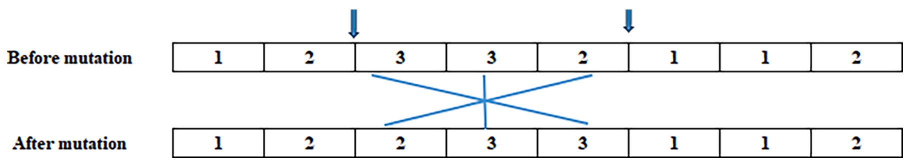

Fig. 6. An example of mutation process on OS vector.

Step 2. Recording Indices. (The indices of the sorted operations are recorded for both P1 and P2.)
Step 3. Creating a Random Mask (Alpha): A mask called "alpha" is created randomly. This alpha mask is a uniform binary vector with a length equal to the chromosome.
Step 4. Copying Genes to Offspring (C1 and C2): If the corresponding alpha value is 1, copy the gene from P1 to the corresponding position in offspring C1. If the alpha value is 0, copy the gene from P2 to the corresponding position in offspring C1. Similarly, for offspring C2, if alpha is 1, copy from P2; if alpha is 0, copy from P1.
Step 5. Re-sorting Offspring Based on Recorded Indices: Finally, both C1 and C2 are resorted based on the indices recorded before. This step ensures that the operation sequencing is restored to the original order.

UC method described here uses a random binary mask (alpha) to decide which genes to inherit from each parent. This process maintains the order of operations while introducing variability in the assignment of machines. The sorting and re-sorting (After the UC operations, we have the sorted child. The result is obtained by re-sorting this sorted child structure according to the initial indices.) The steps are crucial for maintaining the integrity of the operation sequencing during the crossover operation. Fig. 5 describes the application of UC.

3.1.6. *Mutation*

Mutation is a genetic operator that introduces small random changes in individual solutions to explore new regions of the solution space (Talbi, 2009). The role of the mutation operators are very critical in the flow of GA for maintaining genetic diversity and preventing the algorithm from getting stuck in local optima. Two types of mutation operators are used in this study which are assignment mutation operator and sequence mutation operator. Sequence mutation operator changes the sequencing property for operations in the chromosome. It is applied as follows: First, mutation index is selected randomly. Then the item in this index swapped by item at the index +2. The item which locates in the middle of the selected part, stays the same (see Fig. 6).

Assignment mutation operator changes the assignment of operations to machines. It is applied as follows: First, mutation index selected randomly again. Then, item at this index and following two items (items at index +1 and index +2) selected for mutation operation. Since eligibility constraints must be met, list of eligible machines fetched for each operation at index, index +1, and index +2. Last step is selecting new machines for respective positions randomly from eligible machine lists. Here, the coded program is always selecting among the machines that are not currently assigned respective index. If there is only one eligible machine, then the operation is reassigned to this machine (Yang, 2015). Fig. 7 is an example of a mutation process when considering that all machines are eligible.

3.2. *Iterated local search*

Iterated Local Search (ILS) is a metaheuristic algorithm outlined by Lourenço et al. (2010) which navigates local optimal space to locate a global optimum. This method involves introducing disturbances to the existing local optimum solution, then conducting a local search on this perturbed solution. If the outcome of the local search fulfills specific criteria, it becomes the new solution. This iterative process continues until a predefined stopping criterion is fulfilled. To produce the initial solutions for iterative local search algorithms, the affectation and sequence heuristics employed in generating the initial population for GAs are utilized. However, in the case of the ILS algorithm, only a single solution is generated. The local search process relies on two key operators: the affectation operator (AO) and the sequence operator (SO). The affectation operator alters the allocation of a set of operations to machines without altering their order on the machines. Conversely, the sequence operator modifies the sequencing of operations without changing their allocation to machines. The perturbation operator (PO) serves as a mutation-like operator, employed for the recreation of the chromosome when it is necessary. The general flow of these operators are given below, respectively.

The steps of AO is given below.

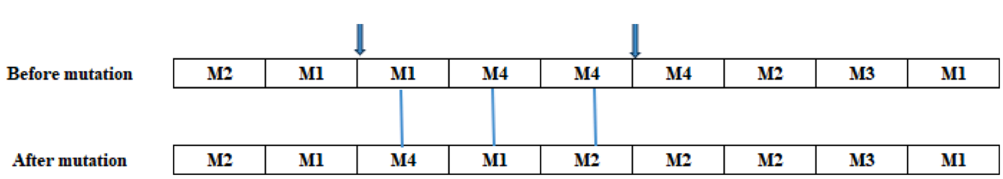

Fig. 7. An example of mutation process on MA vector.

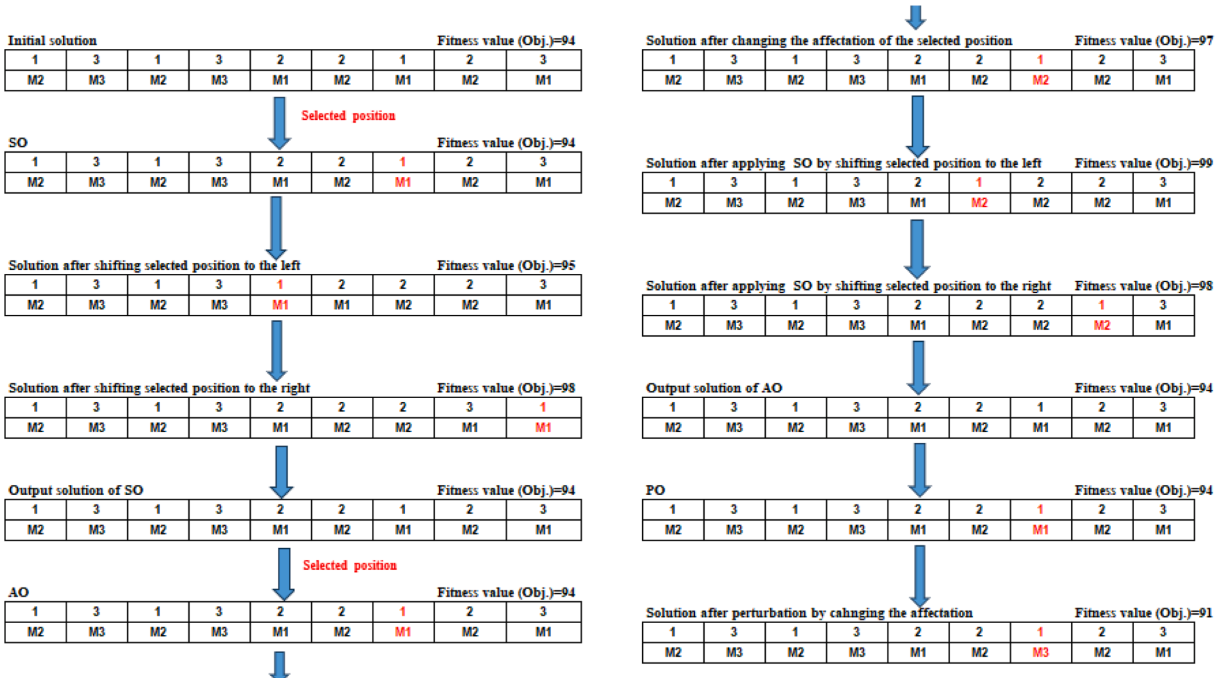

Fig. 8. Steps of ILS algorithm for one iteration.

Input: crossover probability: Pc, mutation probability: Pm, population size: Npop, number of generations: Ngen, and the carbon tax value: Ctax
Output: sum of tardiness and Gmax, sTotal
Initialization:
Let P(t) ← Ø and C(t) ← Ø, ∀t = 1, 2,.., Ngen, where P(t) and C(t) denote the parents and offspring in current generation t, respectively.
Let OBJi ← 0, ∀i= 1, 2, ..., Npop, where OBJi denotes the objective function value corresponding to chromosome i.

Let SBest ← Ø and ObjBest ← ∞ where SBest denotes the best solution and ObjBest denotes the objective function corresponding to the best solution.
01. **Begin**
02. Let t ← 1
03. Randomly generate an initial population, P(t)
04. **Repeat**
05. Sort P(t) in ascending order
06. Split P(t) in two by using elitism
07. **Repeat**
08. **If** this individual is elite, then, **if** this is Hybrid Genetic Algorithm then apply ILS to chromosome and add into C(t), **if** this is classic Genetic Algorithm, then add this individual directly to the C(t)
09. **If** this individual is not elite, then, randomly select two chromosomes from P(t), and apply crossover and mutation operators to child if probability conditions (Pc and Pm) met, and add generated child to C(t)
10. **Until** population size of C(t) = Npop
11. P(t + 1) ← C(t)
12. t ← t + 1
13. **Until** t = Ngen
14. sTotal ← ObjBest
15. **End**

Fig. 9. The pseudocode of the algorithms.

Step 1. Initialization.
Step 2. Reassign selected node to different machine.
Step 3. Calculate if the better solution is obtained. If Yes, Return obtained solution, Else Go to Step 4.
Step 4. Sequence operator applied to current chromosome, starting node is reassigned node.
Step 5. Check if the better solution is obtained. If Yes, Return obtained solution; Else Return the initial solution.

The steps of SO is shown below.

Step 1. Initialization.
Step 2. From the current node, find a node in left has the same machine without disrupting operation sequence.
Step 3. Determine whether the new chromosome has a better solution. If Yes, Return a new chromosome. If No, Go to Step 4.
Step 4. Is this node the leftmost node we can choose? If Yes, Go to Step 5. Else, Return to Step 2.
Step 5. From current node, find a node in right has the same machine without disrupting operation sequence
Step 6. Determine whether the new chromosome has a better solution or not. If Yes, Return a new chromosome. Else, Go to Step 7.
Step 7. Is this node the rightmost node we can choose? If Yes, Return the initial chromosome. Else, Go to Step 2.

The steps of PO is represented in the following.

Step 1. Initialization.
Step 2. Reassign Selected Node to Different Machine.
Step 3. Return Obtained Chromosome.

Based on this information, the general procedure of ILS is outlined as follows:

Step 1. Get initial chromosome from GA.
Step 2. Apply Sequence Operation to the initial chromosome.
Step 3. Apply Affectation Operation to the initial chromosome.
Step 4. Check if the Similarity Degree < Threshold. If Yes, Apply perturbation operator and Go to Step 5. Else, Directly Go to Step 5.
Step 5. Check the result of chromosome (Fitness Value) < Initial chromosome value or not. If Yes; New Chromosome = Result Chromosome. If No; New Chromosome = Initial Chromosome. Go to Step 6.
Step 6. Get a new chromosome. Check if a maximum number of generations are reached. If Yes, Return new chromosome. Else, Return to Step 2.

Steps of the ILS algorithm (Abderrabi et al., 2021) for one iteration of a small example is given below in Fig. 8. First line of the chromosome indicates the operation numbers of jobs, the second line represents the machine assignment.

The simplified pseudocode of the classical GA and the proposed HGA is depicted in Fig. 9.

3.3. Stopping criteria

Maximum number of generations in the main loop of HGA is used in this study. Once the algorithm has run for the specified maximum number of generations, it terminates. This condition prevents the algorithm from running indefinitely and allows for controlling the computational resources devoted to the optimization process.

4. Computational results

This section generates different-sized test problems to analyze the performance of the proposed mathematical model, classical GA, and the proposed HGA. The suggested mathematical model is coded by using GAMS 44.4 software and run using CPLEX solver. The classical GA and proposed HGA is implemented in Golang and executed on a computer with 8 GB of RAM and an Intel Core i5, 2.40 GHz CPU. The characteristics of the test problems, an example of a toy problem, the results obtained by CPLEX, classical GA and the proposed HGA for different sized problems and the impact of the carbon tax on the proposed mathematical model are given in the following subsections.

Table 3
Genetic parameters for GA and the proposed HGA.

Parameter	GA			HGA		
	Small sized	**Medium sized**	**Large sized**	**Small sized**	**Medium sized**	**Large sized**
Number of generations (NG)	200	750	1500	100	350	700
Population size (PS)	25	50	100	25	50	100
Mutation rate (P_m)	0.9	0.9	0.9	0.9	0.9	0.9
Crossover rate (P_c)	0.15	0.15	0.15	0.15	0.15	0.15

Table 4
Classical GA and the proposed HGA parameters for different sized problems.

Parameter	Small-sized	Medium-sized	Large-sized
Number of instances	20	20	20
Number of jobs (i)	3	10	40
Number of operations (j)	3	5	10
Number of machines (m)	4	10	20
p_{ijm}	Discrete U [2,10]	Discrete U [2,10]	Discrete U [2,10]
dd_i	20	40	70
mac_{ijm}	U[500,2000]	U[300,3000]	U[500,3000]
α_i	U[500,2000]	U[500,2000]	U[500,3000]

4.1. Generation of test problems

Since the actual data of the studied problem is inaccessible due to the confidentiality of production process and the complex structures of the machinery involved in defense industry, random generated instances are used which mimic real-life production conditions. This subsection generates different-sized test problems for the considered problem to analyze and compare the performances of the proposed mathematical model, classical GA, and the proposed HGA. The small-sized problems are solved with GAMS/CPLEX solver. Since the problem is NP-hard, the medium and large sized problems could not be solved within the given time limit 7200 seconds by CPLEX. Therefore, these problems are solved both with classical GA and the proposed HGA. Randomly generated different sized test problems include a total of 60 test problems, 20 for each size.

The efficiency of these algorithms is significantly impacted by the selected parameter values. Therefore, optimization of these parameters are critical for the quality of the obtained solutions for each size. Taguchi method is employed for optimizing these parameter values in which robustness is gauged as a means to pinpoint control factors that mitigate variability in a product or process by minimizing the impact of uncontrollable factors, commonly referred to as noise factors. L9 is chosen from the standard orthogonal array table. Thirty problems were used for pretests. These problems are constructed by selecting ten random problems from each size. All statistical analyses are performed by using Minitab 20 software. “Larger is better” signal-to-noise ratio (S/N) is used within the Taguchi method in order to identify the optimal

Table 5
Parameter values for the small-sized problem P1.

i	j	e_{ijm}	p_{ijm}	mac_{ijm}
1	1	M1,M2,M3,M4	9,8,7,6	1500,1000,500,600
1	2	M1,M2,M3,M4	9,5,3,5	1500,1000,400,600
1	3	M1,M2,M3,M4	9,9,7,9	2500,2000,1200,1000
2	1	M1,M2,M3,M4	9,8,5,4	1500,1000,500,550
2	2	M1,M2,M3,M4	8,7,3,4	1000,1000,550,450
2	3	M1,M2,M3,M4	8,9,9,7	1500,1200,600,800
3	1	M1,M2,M3,M4	9,7,4,3	1600,1500,500,800
3	2	M1,M2,M3,M4	8,7,6,6	2000,2000,1000,900
3	3	M1,M2,M3,M4	8,7,6,6	1000,1000,400,450

Table 6
Transportation times between different machines.

	M1	M2	M3	M4
M1	0	0	15	30
M2	0	0	15	30
M3	15	15	0	20
M4	30	30	20	0

parameter levels. The S/N ratio is computed via Eq. (17) where n depicts the number of observations in each trial and Y_i is the value of the objective function.

$$\frac{S}{N} = -10 \times \log\left(\sum_{i=1}^{n} Y_i^2\right) \Big/ n \tag{17}$$

The optimal levels of GA parameters are determined as indicated in Table 3. Other parameters used in both algorithms are given in Table 4.

4.2. Toy problem

In this subsection, a toy problem with three jobs, three operations and four machines are presented. The parameters of the problem are given in Table 5. First column of the Table 5 represents the job number and the second column is the operation number. Third column shows the eligible machines for the specified operation means that which machines can operate the specified operations. In this case, all machines can operate all operations with different processing times, machine costs and carbon footprint of the machines are also given. M1 and M2 are in-house machines which use renewable energy source (solar). Since they locate in the same factory, transportation times between these machines are assumed to be negligible. Also, the carbon footprint values of M1 and M2 are lower compared to M3 and M4, which belongs to different sucbontractors. M3 uses only coal energy and M4 uses a mixed combination of coal and natural gas energy, so the carbon footprints are high compared to M1 and M2. The penalty cost of tardiness for each job is taken 500, 1000 and 2000, respectively. Due dates for all jobs are assumed to be 20. Carbon footprint values are taken 0.05, 0.05, 0.85 and 0.80 (kgCO_2eq/kWh) for M1, M2, M3 and M4, respectively. Table 6 indicates the transportation times between different machines. Toy problem is solved by CPLEX and the solution is desribed below.

As it is observed from Fig. 10, when the carbon cost is assumed to be $65 and all machines can perform every operation for each job, the optimal machines are selected which are M1, M2 and M4, and the case involves the processing of consecutive operations on the same machine. The optimal solution does not include M3 due to its high carbon footprint value. The carbon footprint of the schedule which indicates $Carb_{max}$ is calculated as 2203, and the overall objective function is 278195.

4.3. Test results

To analyze the test results, 7200 s time limit is set and the optimal solutions obtained from CPLEX solver is used to compare the results of the proposed HGA and GA for the small sized test problems. Since the medium and large sized problems could not be solved in a given time limit, both algorithms are applied to solve these problems. Obtaining a computable lower bound close to the optimal solution is crucial for assessing the effectiveness of the algorithms. For classical GA and proposed HGA, the number of run is set 10. Elitism strategy is applied in both algorithms. The percentage improvement value is used to compare the best obtained solutions of CPLEX with GA and HGA for small sized problems; solutions of GA with HGA for the others. Its formula is given in Equation (18) as follows:

$$imp(method2 - method1) = \frac{Obj.(method1) - Obj.(method2)}{Obj.(method1)} \times 100 \tag{18}$$

The test results for small sized problems are given in Table 7, results of the medium sized problems in Table 8 and the results of large ones in Table 9. First column of the Tables indicates the instance name which are called “S-PX”, “M−PX” and “L-PX”. “S”,”M”and “L” represent the size of the problem which are small, medium and large. “PX” is the problem number for the specified size. For example, instance “L-P20” means that large sized problem number 20. Time values are given in milliseconds (ms) and seconds (s) in Tables 7, 8, and 9. For all sized problems, carbon tax value is considered as $65. In these Tables, the best (min) and the worst (max) solutions obtained by GA and HGA within 10 runs and the percentage differences between the obtained solutions for each size (which are calculated as in (19) and (20)) are also given.

$$dif(GA_{min} - GA_{max}) = \frac{Obj.(GA_{max}) - Obj.(GA_{min})}{Obj.(GA_{max})} \times 100 \tag{19}$$

$$dif(HGA_{min} - HGA_{max}) = \frac{Obj.(HGA_{max}) - Obj.(HGA_{min})}{Obj.(HGA_{max})} x100 \tag{20}$$

Four columns under GAMS/CPLEX in Table 7 which indicates the test results for small sized problems, are the objective function value, lower bound, run time in seconds, and percentage optimality gap, respectively. As shown in Table 7, CPLEX solver reached optimal solutions for all test problems within the time limit. The average run time is 1542.75 s. Solution time is mainly longer than this value in test instances such as S-P4, S-P8, S-P9, S-P19 and S-P20. Since the objective of this problem is to

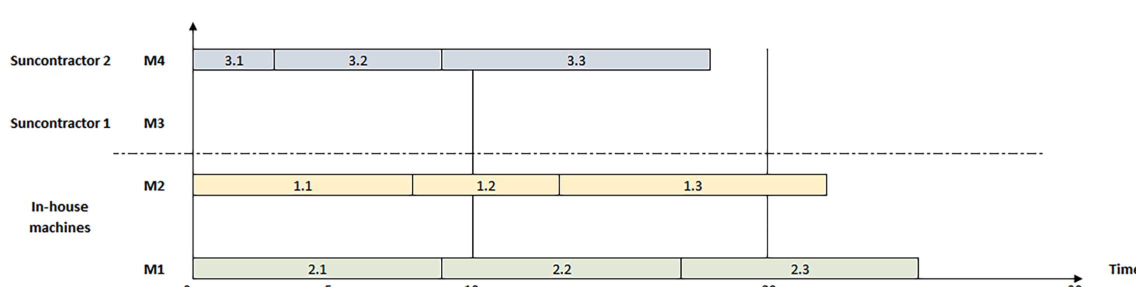

Fig. 10. Gantt chart of the toy problem.

Table 7
Computational results of GAMS/CPLEX for small sized test problems compared with GA and HGA.

Instance	GAMS/CPLEX				GA			HGA			Imp			Dif	
	Obj.	LB	Time(s)	GAP	Obj. (Min)	Obj. (Max)	Time (ms)	Obj. (Min)	Obj. (Max)	Time (s)	(HGA(min)-CPLEX)%	(GA(min)-CPLEX)%	(HGA(min)-GA(min))%	(GA(min)-GA(max))%	(HGA(min)-HGA(max))%
S-P1	278,195	278,195	53.19	0	278,195	294,886	89.71	278,195	284,195	6.75	0	0	0	5.7	2.1
S-P2	287,684	287,684	74.15	0	287,684	299,191	170.51	287,684	293,989	7.88	0	0	0	3.8	2.1
S-P3	306,295	306,295	60.13	0	332,825	355,609	165.44	306,295	317,554	25.66	0	−8.7	8.0	6.4	3.5
S-P4	293,425	293,425	4200	0	293,425	321,030	11.69	293,425	302,649	7.95	0	0	0	8.6	3.0
S-P5	309,405	309,405	436.25	0	320,735	345,872	69.77	309,405	322,226	7.88	0	−3.7	3.5	7.3	4.0
S-P6	297,055	297,055	720	0	305,285	308,937	182.65	297,055	315,566	29.36	0	−2.8	2.7	1.2	5.9
S-P7	306,295	306,295	296.29	0	326,285	360,798	55.23	306,295	328,906	10.74	0	−6.5	6.1	9.6	6.9
S-P8	300,970	300,970	4801.26	0	300,970	322,037	301.12	300,970	302,246	11.23	0	0	0	6.5	0.4
S-P9	293,425	293,425	4566.05	0	293,425	305,162	266.52	293,425	322,586	25.89	0	0	0	3.8	9.0
S-P10	311,690	311,690	30.56	0	311,690	336,625	100.25	311,690	319,645	135.2	0	0	0	7.4	2.5
S-P11	309,405	309,405	409.8	0	325,210	331,063	179.07	309,405	312,266	188.65	0	−5.1	4.9	1.8	0.9
S-P12	320,820	320,820	53.19	0	320,820	330,444	124.6	320,820	335,606	40.55	0	0	0.0	2.9	4.4
S-P13	328,810	328,810	74.15	0	333,569	341,962	89.71	328,810	341,946	55.21	0	−1.45	1.43	2.5	3.8
S-P14	336,844	336,844	60.13	0	336,844	367,054	170.58	336,844	359,286	135.2	0	0	0.0	8.2	6.2
S-P15	344,790	344,790	4200	0	344,790	372,373	165.44	344,790	358,597	198.56	0	0	0	7.4	3.9
S-P16	352,780	352,780	436.25	0	352,780	366,891	11.69	352,780	362,147	69.87	0	0	0	3.8	2.6
S-P17	360,770	360,770	720	0	360,770	378,808	69.77	360,770	367,854	84.53	0	0	0	4.8	1.9
S-P18	368,765	368,765	296.29	0	390,664	395,890	182.65	368,765	377,854	187.2	0	−5.9	5.6	1.3	2.4
S-P19	376,750	376,750	4801.26	0	376,750	406,890	66.4	376,750	400,896	174.32	0	0	0	7.4	6.0
S-P20	384,742	384,742	4566.05	0	384,742	403,979	301.12	384,742	401,285	69.66	0	0	0	4.8	4.1
Average	323445.75	323445.75	1542.75	0	328872.9	347275.05	138.696	323445.8	336364.95	73.6	0	−1.71	1.61	5.3	3.8

minimize the total cost (both economic and environmental costs), the negative improvement value represents the better performance of CPLEX over GA for the small sized problems in Table 7 (for instances S-P3, S-P5, S-P6, S-P7, S-P11, S-P13 and S-P18). On the other hand, HGA has obtained the optimal solutions in all small sized problems. However, since HGA incorporates ILS with GA and consists more inner iterations, the solution time is longer compared to classical GA. For instance, in problem S-P8, the optimal solution is 300,970 and the solution time of CPLEX is 4801.26 s. GA could find this optimal solution in 301.12 ms however HGA could find it in 11.23 s. The computational time of HGA is shorter than CPLEX for this problem.

Table 8 and 9 depict the test results for medium and large sized problems. Three columns under GA in these Tables indicate the minimum and maximum objective function values obtained by GA within 10 runs and run time in seconds, respectively. Three columns under HGA depict the minimum and maximum objective function values obtained by HGA within 10 runs and run time in seconds, respectively. In these Tables, column 7 presents the percentage improvement value between the best solutions of GA and the proposed HGA. Columns 8 and 9 show the difference between the minimum and maximum objective function values obtained with GA and HGA within 10 runs for each problem.

The obtained results for all medium sized test problems demonstrate that, HGA gives better solutions than the solutions obtained by classical GA. Average fitness function values are calculated 649140.55 by GA and 573387.5 by HGA. For problem M−P11, improvement of HGA over GA increases to 19.91 %. The results show that the performance of HGA is better than GA in all medium sized problems. Average solution time of HGA is 1150.73 s., while it is 3.73 s. for GA with worse quality solutions compared to HGA. For the problem M−P4, the difference between the minimum and maximum objective function values obtained with HGA within 10 runs is 5.66 %, while in GA, this value is 15.25 %.

It is remarkable that, as seen in Table 9, HGA produced better solutions than GA in all large-sized problems. For instance, in the L-P13, while the fitness value for GA is 3965830, HGA found 2881610, resulting in a 27.3 % improvement compared to GA. The improvement rate of all solutions achieved by applying HGA in large-sized problems compared to GA, are calculated to be an average of 17.4 %. This indicates a significant impact of the ILS operators incorporated by GA, on the solutions obtained compared to classical GA. Overall, when considering both economic and environmental costs, HGA could able to solve problems more effectively by producing higher-quality solutions.

From the result Tables given above, average objective function values are calculated for different sized problems and illustrated in Fig. 11. It is remarkable that average of the minimum objective function values obtained by HGA are better than the solutions of GA for each problem size. Improvement percentages for different sized problems of HGA over GA are also illustrated in Fig. 12.

As depicted in Fig. 13 below, HGA has captured a better solution faster compared to GA for the problem M−P3 (see Table 8). No improvement is observed after the 250th iteration for HGA, and the solution is found to be 570950. In GA, however, a solution of 706,715 is found at the 540th iteration. HGA has improved the solution of GA by 19.21 %.

4.4. The impact of the carbon tax on the proposed mathematical model

This subsection examines the carbon tax value to assess its influence on the proposed mathematical model. Below is a case study conducted for the small-sized problem P1, assuming that each operation of each job can be performed on every machine under the carbon tax analysis described below. The columns of Table 10 indicate the machine numbers, enegy source which machine uses, carbon footprint coefficient, electricity consumption of machine m and the average machine usage costs, respectively. In Table 10, carbon footprint values are determined by interviewing the production facilities based on the energy sources they use and the carbon taxes are taken from both

Table 8
Computational results of GA and HGA for medium sized test problems.

	GA			HGA			Imp	Dif	
	Obj. (min)	Obj. (max)	Time (s)	Obj. (min)	Obj. (max)	Time (s)	(HGA(min)-GA (min))%	(GA(min)-GA (max))%	(HGA(min)-HGA (max))%
M-P1	588,480	700,291	2.96	504,690	540,018	1500.27	14.24	15.97	6.54
M-P2	566,165	684,105	2.74	558,595	603,283	1260.47	1.34	17.24	7.41
M-P3	706,715	819,789	2.82	570,950	622,336	900.56	19.21	13.79	8.26
M-P4	602,745	711,239	2.99	521,155	552,424	1140.25	13.54	15.25	5.66
M-P5	570,395	667,362	2.74	541,950	585,306	1140.08	4.99	14.53	7.41
M-P6	536,490	643,788	2.72	491,290	540,419	1080.7	8.43	16.67	9.09
M-P7	649,815	766,782	3.13	573,590	602,270	1085.2	11.73	15.25	4.76
M-P8	570,930	650,860	2.7	548,020	585,421	1100.25	4.01	12.28	6.39
M-P9	763,855	890,795	5.71	636,395	687,307	1200.28	16.69	14.25	7.41
M-P10	610,170	671,187	2.63	543,525	602,442	1140.48	10.92	9.09	9.78
M-P11	687,558	756,314	2.45	550,655	600,214	1143.22	19.91	9.09	8.26
M-P12	707,178	784,968	3.53	657,785	697,252	1125.6	6.98	9.91	5.66
M-P13	660,154	726,169	6.61	564,915	593,161	1144.6	14.43	9.09	4.76
M-P14	764,578	863,973	6.56	672,045	719,088	1122.9	12.10	11.5	6.54
M-P15	667,890	724,716	3.58	591,750	627,255	1111.4	11.40	7.84	5.66
M-P16	805,469	894,071	5.15	725,698	756,497	1166.89	9.90	9.91	4.07
M-P17	569,871	672,448	2.39	499,687	549,656	1154.8	12.32	15.25	9.09
M-P18	550,269	631,804	2.88	466,589	523,916	1170.7	15.21	12.91	10.94
M-P19	639,870	755,047	3.9	589,674	642,745	1169.55	7.84	15.25	8.26
M-P20	764,214	894,130	6.5	658,791	718,082	1156.4	13.79	14.53	8.26
Average	649140.55	745491.9	3.73	573387.5	617454.6	1150.73	11.45	12.98	7.21

Table 9
Computational results of GA and HGA for large sized test problems.

	GA			HGA			Imp	Dif	
Instance	Obj. (min)	Obj. (max)	Time (s)	Obj. (min)	Obj. (max)	Time (s)	(HGA(min)-GA (min))%	(GA(min)-GA (max))%	(HGA(min)-HGA (max))%
L-P1	4,321,625	5,615,598	300.47	3,667,775	4,268,107	3360.08	15.1	23	14.1
L-P2	3,622,210	4,455,318	360.82	2,944,955	3,128,441	3660.5	18.7	18.7	5.9
L-P3	3,801,880	4,842,444	300.38	3,186,855	3,551,700	2700.18	16.2	21.5	10.3
L-P4	4,357,595	5,141,962	540.6	3,476,420	4,041,232	3300.85	20.2	15.3	14
L-P5	4,085,315	5,147,496	480.24	3,290,300	3,921,584	3480.78	19.5	20.6	16.1
L-P6	3,946,255	4,617,118	480.36	3,155,925	3,818,669	4500.9	20	14.5	17.4
L-P7	3,279,960	3,804,753	482.7	3,046,940	3,747,736	3960.43	7.1	13.8	18.7
L-P8	3,947,510	5,013,337	420.57	3,148,115	3,714,775	3360.29	20.3	21.3	15.3
L-P9	3,985,535	4,962,352	540.7	2,962,280	3,495,490	3720.21	25.7	19.7	15.3
L-P10	3,717,670	4,098,380	420.2	2,955,365	3,457,777	4260.7	20.5	9.3	14.5
L-P11	3,915,540	5,502,871	480.3	3,202,755	3,811,278	4140.28	18.2	28.8	16
L-P12	3,814,795	4,959,233	490.6	2,922,125	3,535,771	3960.55	23.4	23.1	17.4
L-P13	3,965,830	4,877,970	300.66	2,881,610	3,546,093	3840.88	27.3	18.7	18.7
L-P14	3,047,290	3,565,329	540.3	2,812,140	3,430,810	3985.55	7.7	14.5	18.0
L-P15	3,709,015	4,784,629	660.3	2,896,615	3,352,599	3748.9	21.9	22.5	13.6
L-P16	3,325,262	3,857,303	435.2	2,926,155	3,365,078	2680.45	12.0	13.8	13.0
L-P17	3,201,674	3,842,008	470.35	2,894,648	3,163,042	4022.27	9.6	16.7	8.5
L-P18	3,078,086	3,786,045	482.6	2,796,955	3,456,346	3664.23	9.1	18.7	19.1
L-P19	2,954,498	3,427,217	300.6	2,633,235	3,212,546	3685.2	10.9	13.8	18
L-P20	3,830,910	4,788,637	540.36	2,901,568	3,297,944	59.88	24.3	20	12
Average	3695422.8	4,554,500	451.4155	3035136.8	3565850.9	3504.6555	17.4	18.4	14.8

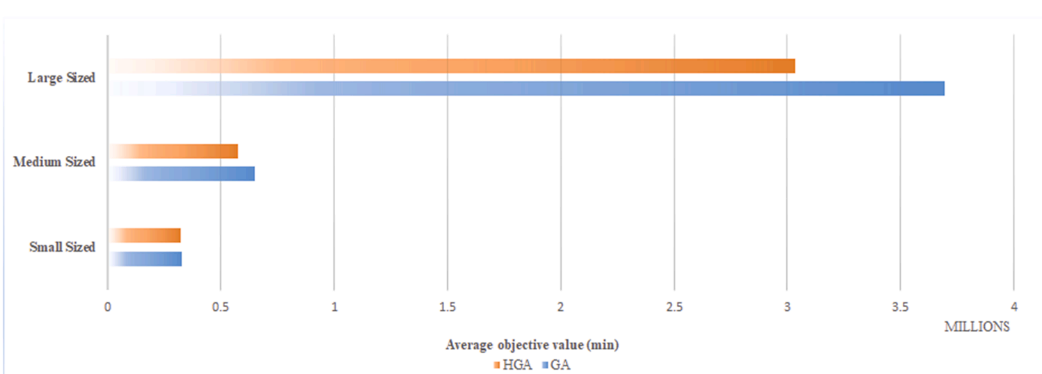

Fig. 11. Average objective function values of GA and HGA for different sized problems.

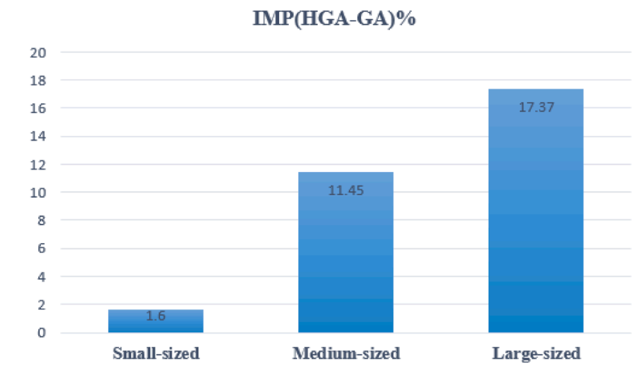

Fig. 12. Improvement percentages for different sized problems of HGA over GA.

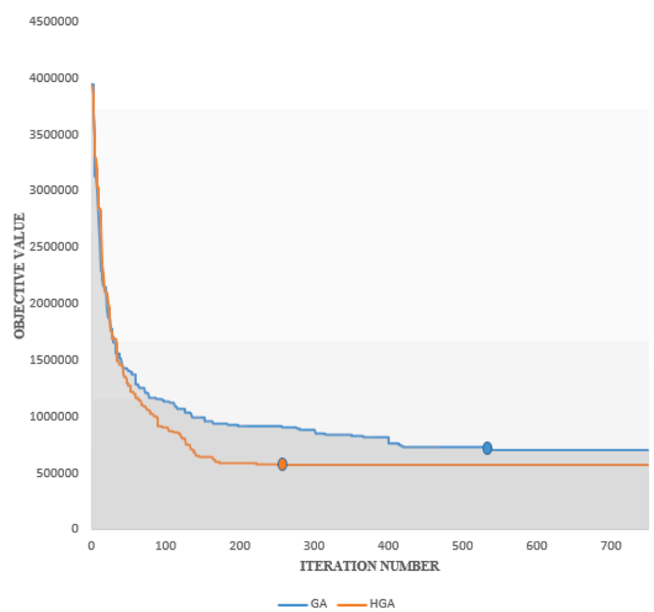

Fig. 13. Convergence Analysis of the Fitness Function for GA and HGA for M−P3.

Table 10
Machine characteristics for the problem S-P1 (International Energy Agency, 2023; International Renewable Energy Agency, 2023).

Machine number	Energy source used by machine	Carbon footprint coefficient	Electricity consumptionof machine m (kWh)	Average cost of machine
M1	Solar energy	0.05	60	1750
M2	Solar energy	0.05	80	1500
M3	Coal energy	0.85	200	500
M4	Coal & natural gas energy	0.80	170	600

international reports and literature (International Energy Agency, 2023; International Renewable Energy Agency, 2023; Trevino-Martinez, et al., 2022). Table 11 illustrates the sensitivity of the proposed model to understand the effect of carbon taxes on the obtained solutions. First column indicates the different carbon taxes which are assumed as \$65, \$150 and \$300 per Ton equivalent of CO_2 (tCO_2e). The obtained results of GAMS/CPLEX, GA and HGA are also given. When the carbon tax is \$65, model chooses M1, M2 and M4. If this tax is \$150 or \$300 (when carbon tax rises sharply), in-house machines 1 and 2 which use renewable energy are selected. This result shows that carbon tax has a great influence on the model to select more environmentally-friendly decisions.

As seen in this example, when the carbon tax is sufficiently high, the model chooses more environmentally friendly machines despite higher machine costs. For instance, when carbon tax is 65, the objective function is 278,195 with selected machines M1, M2 and M4. The carbon footprint of the schedule ($Carb_{max}$) is calculated 2203. The third machine is excluded by the model due to its elevated carbon footprint and higher electricity consumption which is 200 kWh. If the carbon tax is equal to 300, only in-house machines are used and the carbon footprint of the schedule is calculated as 247. By giving a significant carbon tax, there is an 88.7 % improvement in the schedule's carbon footprint. This underscores the significant role of determining carbon tax policies and selecting renewable energy sources. As the carbon tax increases, model avoids outsourcing operations (since main factory uses solar energy for the electricity consumption) if possible. Therefore, accurately determining the carbon tax is crucial for the proposed model when considering both economic and environmental costs. For factories inclined towards more environmentally friendly practices and solving scheduling problems by outsourcing operations, this study can be considered as a valuable guide.

5. Conclusions

Production scheduling plays a crucial role in achieving energy efficiency within a manufacturing system, particularly when viewed from an operations management perspective. In order to address the challenges posed by global climate changes, it is also crucial for managers to contemplate the environmental impact of their decisions. To enhance the implementation of critical clean energy solutions in addressing real-life scheduling issues, incentives are essential. These can be realized through governmental interventions, including the reduction of initial and operational expenses for clean alternatives through technological and financial advancements, or the adjustment of pricing, taxation, and subsidy systems related to fossil fuels. This research contributes an important perspectives to the decision makers on enhancing the operational scheduling by taking into account economic factors like total penalty costs for tardy jobs wtih machine costs along with environmental considerations such as the carbon footprint of the schedule. The suggested mathematical model proved ineffective in identifying feasible solutions due to the inherent problem structure in cases involving medium and large-sized problems. Therefore, HGA which integrates GA and ILS, has been proposed for the first time to the described problem. The results indicate that HGA can efficiently generate highly successful solutions within a reasonable time frame. Obtained solutions also offer valuable insights into the impact of carbon footprint which is an

Table 11
The impact of the carbon tax on the proposed mathematical model.

	GAMS/CPLEX			GA			HGA		
Carbon Tax	**Obj.**	**Selected Machines**	**Time (s)**	**Obj.**	**Selected Machines**	**Time (ms)**	**Obj.**	**Selected Machines**	**Time (s)**
65	278,195	M1,M2,M4	4000.86	278,195	M1,M2,M4	145.2	278,195	M1,M2,M4	20.16
150	318,050	M1,M2	4000.037	318,050	M1,M2	201.5	318,050	M1,M2	22.30
300	355,100	M1,M2	519.285	355,100	M1,M2	164.3	355,100	M1,M2	18.45

essential factor for manufacturing factories striving to reduce environmental effects in today's world. The developed model in this research integrates economic and environmental costs into a unified objective function. The precise determination of the carbon tax value holds significant importance in the model. When addressing this problem, the implemented HGA algorithm outperformed the GA, showcasing notably superior results. The enhancement reached up to 17.37 %, particularly evident in larger-scale problems. As the proposed model shows, the carbon tax acts as a disincentive factor for more environmentally friendly scheduling options, even if other economic costs are too high. Manufacturing factories face with the decision juncture regarding how to strategically invest energy sources, aiming to both minimize environmental costs and assess economic costs simultaneously. The following future research avenues can be investigated. First, it will be interesting to search the influence of environmental based scheduling with respect to other objective functions (such as total energy consumption of machines when they are at the idle state). It can also be interesting to consider the vehicles (electric vehicles or not) used for the transportation between different factories. Finally, various heuristics/metaheuristics may be applied for the proposed model to make a wide comparison.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Abderrabi, F., Godichaud, M., Yalaoui, A., Yalaoui, F., Amodeo, L., Qerimi, A., & Thivet, E. (2021). Flexible job shop scheduling problem with sequence dependent setup time and job splitting: Hospital catering case study. *Applied Sciences, 11*(4), Article 1504. https://doi.org/10.3390/app11041504

Ahmadizar, F., & Amiri, Z. (2018). Outsourcing and scheduling for a two-machine flow shop with release times. *Engineering Optimization, 50*(3), 483–498. https://doi.org/10.1080/0305215X.2017.1325483

Alegoz, M., Kaya, O., & Bayindir, Z. P. (2021). A comparison of pure manufacturing and hybrid manufacturing–remanufacturing systems under carbon tax policy. *European Journal of Operational Research, 294*(1), 161–173. https://doi.org/10.1016/j.ejor.2021.01.018

Atilgan, B., & Azapagic, A. (2016). Assessing the environmental sustainability of electricity generation in Turkey on a life cycle basis. *Energies, 9*(1), Article 31. https://doi.org/10.3390/en9010031

Avci, M. (2023). An effective iterated local search algorithm for the distributed no-wait flowshop scheduling problem. *Engineering Applications of Artificial Intelligence, 120*, Article 105921. https://doi.org/10.1016/j.engappai.2023.105921

Babaee Tirkolaee, E., Goli, A., & Weber, G. W. (2020). Fuzzy mathematical programming and self-adaptive artificial fish swarm algorithm for just-in-time energy-aware flow shop scheduling problem with outsourcing option. *IEEE Transactions on Fuzzy Systems, 28*(11), 2772–2783. https://doi.org/10.1109/TFUZZ.2020.2998174

Bektur, G. (2023). A variant of the NSGA II for the speed scaling distributed flow shop scheduling problem with total tardiness minimisation. *Journal of Intelligent and Fuzzy Systems, 45*(1), 1209–1222. https://doi.org/10.3233/JIFS-224199

Bektur, G., & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers and Operations Research, 103*, 46–63. https://doi.org/10.1016/j.cor.2018.10.010

Bi, H., Lu, F., Duan, S., Huang, M., Zhu, J., & Liu, M. (2020). Two-level principal–agent model for schedule risk control of IT outsourcing project based on genetic algorithm. *Engineering Applications of Artificial Intelligence, 91*, Article 103584. https://doi.org/10.1016/j.engappai.2020.103584

Bouzid, M., Masmoudi, O., & Yalaoui, A. (2021). Exact methods and heuristics for order acceptance scheduling problem under time-of-use costs and carbon emissions. *Applied Sciences (Switzerland), 11*(19), Article 8919. https://doi.org/10.3390/app11198919

Chang, H. C., Chen, Y. P., Liu, T. K., & Chou, J. H. (2015). Solving the flexible job shop scheduling problem with makespan optimization by using a hybrid taguchi-genetic algorithm. *IEEE Access, 3*, 1740–1754. https://doi.org/10.1109/ACCESS.2015.2481463

Chen, Y., Zhong, L., Shen, C., Mumta, J., & Chou, F. D. (2023). Joint optimization of production and maintenance scheduling for unrelated parallel machine using hybrid discrete spider monkey optimization algorithm. *International Journal of Industrial Engineering Computations, 14*(3), 539–554. https://doi.org/10.5267/j.ijiec.2023.4.001

Dai, L., Lu, H., Hua, D., Liu, X., Chen, H., Glowacz, A., Królczyk, G., & Li, Z. (2022). A novel production scheduling approach based on improved hybrid genetic algorithm. *Sustainability (Switzerland), 14*(18), Article 11747. https://doi.org/10.3390/su141811747

Drucker, P. (1989). Sell the mailroom. *Wall Street Journal, 25*.

Fang, K., Uhan, N., Zhao, F., & Sutherland, J. W. (2011). A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems, 30*(4), 234–240. https://doi.org/10.1016/j.jmsy.2011.08.004

Fekih, A., Hadda, H., Kacem, I., & Hadj-Alouane, A. B. (2020). A hybrid genetic Tabu search algorithm for minimising total completion time in a flexible job-shop scheduling problem. *European Journal of Industrial Engineering, 14*(6), 763–781. https://doi.org/10.1504/EJIE.2020.112479

G.-de-Alba, H., Nucamendi-Guillén, S., & Avalos-Rosales, O. (2022). A mixed integer formulation and an efficient metaheuristic for the unrelated parallel machine scheduling problem: Total tardiness minimization. *EURO Journal on Computational Optimization, 10*, Article 100034. https://doi.org/10.1016/j.ejco.2022.100034

Ghorbanzadeh, M., & Ranjbar, M. (2023). Energy-aware production scheduling in the flow shop environment under sequence-dependent setup times, group scheduling and renewable energy constraints. *European Journal of Operational Research, 307*(2), 519–537. https://doi.org/10.1016/j.ejor.2022.09.034

Gómez, J., Chicaiza, W. D., Escaño, J. M., & Bordons, C. (2023). A renewable energy optimisation approach with production planning for a real industrial process: An application of genetic algorithms. *Renewable Energy, 215*, Article 118933. https://doi.org/10.1016/j.renene.2023.118933

Guo, X., & Lei, D. (2014). Bi-objective job shop scheduling with outsourcing options. *International Journal of Production Research, 52*(13), 3832–3841. https://doi.org/10.1080/00207543.2013.848488

He, B., Liu, R., & Li, T. (2023). Integrated carbon footprint with cutting parameters for production scheduling. *Journal of Cleaner Production, 412*, Article 137307. https://doi.org/10.1016/j.jclepro.2023.137307

Homayouni, S. M., & Fontes, D. B. M. M. (2021). Production and transport scheduling in flexible job shop manufacturing systems. *Journal of Global Optimization, 79*(2), 463–502. https://doi.org/10.1007/s10898-021-00992-6

International Energy Agency. (2023). World Energy Outlook 2023. Retrieved from https://www.iea.org/reports/world-energy-outlook-2023. Accessed March 20, 2024.

International Renewable Energy Agency. (2023). Renewable Energy Statistics 2023. Retrieved from https://www.irena.org/Publications/2023/Jul/Renewable-energy-statistics-2023. Accessed March 20, 2024.

Izadi, L., Ahmadizar, F., & Arkat, J. (2020). A hybrid genetic algorithm for integrated production and distribution scheduling problem with outsourcing allowed. *International Journal of Engineering, Transactions B: Applications, 33*(11), 2285–2298. https://doi.org/10.5829/ije.2020.33.11b.19

Khedim, A. O., Souier, M., & Sari, Z. (2022). Combinatorial artificial bee colony algorithm hybridised with a new release of iterated local search for job shop scheduling problem. *International Journal of Operational Research, 44*(4), 435–461. https://doi.org/10.1504/ijor.2022.125128

Lee, K., & Choi, B. C. (2011). Two-stage production scheduling with an outsourcing option. *European Journal of Operational Research, 213*(3), 489–497. https://doi.org/10.1016/j.ejor.2011.03.037

Lei, D., Gao, L., & Zheng, Y. (2018). A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop. *IEEE Transactions on Engineering Management, 65*(2), 330–340. https://doi.org/10.1109/TEM.2017.2774281

Li, H., Wang, X., & Peng, J. (2022). A hybrid differential evolution algorithm for flexible job shop scheduling with outsourcing operations and job priority constraints. *Expert Systems with Applications, 201*, Article 117182. https://doi.org/10.1016/j.eswa.2022.117182

Liu, C. H., & Huang, D. H. (2014). Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *International Journal of Production Research, 52*(2), 337–352. https://doi.org/10.1080/00207543.2013.825740

Liu, L. (2019). Outsourcing and rescheduling for a two-machine flow shop with the disruption of new arriving jobs: A hybrid variable neighborhood search algorithm. *Computers and Industrial Engineering, 130*, 198–221. https://doi.org/10.1016/j.cie.2019.02.015

Liu, Q., Zhan, M., Chekem, F. O., Shao, X., Ying, B., & Sutherland, J. W. (2017). A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint. *Journal of Cleaner Production, 168*, 668–678. https://doi.org/10.1016/j.jclepro.2017.09.037

Liu, X., Liu, L., & Jiang, T. (2023). A self-learning interior search algorithm based on reinforcement learning for energy-aware job shop scheduling problem with outsourcing option. *Journal of Intelligent and Fuzzy Systems, 44*(6), 10085–10100. https://doi.org/10.3233/JIFS-224624

Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). International series in operations research & management science. In M. Gendreau, & J. Y. Potvin (Eds.), *Iterated Local Search: Framework and Applications* (pp. 363–397). Boston, MA: Springer.

Lu, F., Bi, H., Huang, M., & Duan, S. (2017). Simulated Annealing Genetic Algorithm Based Schedule Risk Management of IT Outsourcing Project. Mathematical Problems in Engineering, 2017, Article 6916575. 10.1155/2017/6916575.

Moghadam, A. M., Wong, K. Y., & Piroozfard, H. (2014). An efficient genetic algorithm for flexible job-shop scheduling problem. *Poster session presentation at the meeting of*

IEEE International Conference on Industrial Engineering and Engineering Management, Selangor, Malaysia.

Mokhtari, H., & Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Computers and Chemical Engineering, 104*, 339–352. https://doi.org/10.1016/j.compchemeng.2017.05.004

Mokhtari, H., Kamal Abadi, I. N., & Amin-Naseri, M. R. (2012). Production scheduling with outsourcing scenarios: A mixed integer programming and efficient solution procedure. *International Journal of Production Research, 50*(19), 5372–5395. https://doi.org/10.1080/00207543.2011.627687

Mouzon, G., & Yildirim, M. B. (2008). A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering, 1*(2), 105–116. https://doi.org/10.1080/19397030802257236

Oladzad-Abbasabady, N., Tavakkoli-Moghaddam, R., Mohammadi, M., & Vahedi-Nouri, B. (2023). A bi-objective home care routing and scheduling problem considering patient preference and soft temporal dependency constraints. *Engineering Applications of Artificial Intelligence, 119*, Article 105829. https://doi.org/10.1016/j.engappai.2023.105829

Pang, X., Wang, Y., Yang, S., Liu, W., & Yu, Y. (2024). A bi-objective low-carbon economic scheduling method for cogeneration system considering carbon capture and demand response. *Expert Systems with Applications, 243*, Article 122875. https://doi.org/10.1016/j.eswa.2023.122875

Piroozfard, H., Wong, K. Y., & Wong, W. P. (2018). Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resources, Conservation and Recycling, 128*, 267–283. https://doi.org/10.1016/j.resconrec.2016.12.001

Qi, X. (2008). Coordinated logistics scheduling for in-house production and outsourcing. *IEEE Transactions on Automation Science and Engineering, 5*(1), 188–192. https://doi.org/10.1109/TASE.2006.887159

Qin, Q., Cheng, S., Zhang, Q., Li, L., & Shi, Y. (2016). Particle Swarm Optimization with Interswarm Interactive Learning Strategy. *IEEE Transactions on Cybernetics, 46*(10), 2238–2251. https://doi.org/10.1109/TCYB.2015.2474153

Queiroga, E., Pinheiro, R. G. S., Christ, Q., Subramanian, A., & Pessoa, A. A. (2021). Iterated local search for single machine total weighted tardiness batch scheduling. *Journal of Heuristics, 27*(3), 353–438. https://doi.org/10.1007/s10732-020-09461-x

Rahman, H. F., Servranckx, T., Chakrabortty, R. K., Vanhoucke, M., & El Sawah, S. (2022). Manufacturing project scheduling considering human factors to minimize total cost and carbon footprints. *Applied Soft Computing, 131*, Article 109764. https://doi.org/10.1016/j.asoc.2022.109764

Ramezanian, R., Vali-Siar, M. M., & Jalalian, M. (2019). Green permutation flowshop scheduling problem with sequence-dependent setup times: A case study. *International Journal of Production Research, 57*(10), 3311–3333. https://doi.org/10.1080/00207543.2019.1581955

Sagar, K. V., Jerald, J., & Khan, M. A. (2023). An energy-aware optimisation model to minimise energy consumption and carbon footprint in a flexible manufacturing system. *International Journal on Interactive Design and Manufacturing, 1–12*. https://doi.org/10.1007/s12008-023-01352-2

Sun, K., Zheng, D., Song, H., Cheng, Z., Lang, X., Yuan, W., & Wang, J. (2023). Hybrid genetic algorithm with variable neighborhood search for flexible job shop scheduling problem in a machining system. *Expert Systems with Applications, 215*, Article 119359. https://doi.org/10.1016/j.eswa.2022.119359

Talbi, E. G. (2009). *Metaheuristics: From Design to Implementation*. John Wiley & Sons.

Trevino-Martinez, S., Sawhney, R., & Shylo, O. (2022). Energy-carbon footprint optimization in sequence-dependent production scheduling. *Applied Energy, 315*, Article 118949. https://doi.org/10.1016/j.apenergy.2022.118949

Tutumlu, B., & Saraç, T. (2023). A MIP model and a hybrid genetic algorithm for flexible job-shop scheduling problem with job-splitting. *Computers and Operations Research, 155*, Article 106222. https://doi.org/10.1016/j.cor.2023.106222

Wei, Z., Liao, W., & Zhang, L. (2022). Hybrid energy-efficient scheduling measures for flexible job-shop problem with variable machining speeds. *Expert Systems with Applications, 197*, Article 116785. https://doi.org/10.1016/j.eswa.2022.116785

Wu, X., Shen, X., & Cui, Q. (2018). Multi-objective flexible flow shop scheduling problem considering variable processing time due to renewable energy. *Sustainability, 10*(3), Article 841. https://doi.org/10.3390/su10030841

Xu, W., Hu, Y., Luo, W., Wang, L., & Wu, R. (2021). A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission. *Computers and Industrial Engineering, 157*, Article 107318. https://doi.org/10.1016/j.cie.2021.107318

Xu, W., Wang, L., Liu, D., Tang, H., & Li, Y. (2023). Solving distributed low carbon scheduling problem for large complex equipment manufacturing using an improved hybrid artificial bee colony algorithm. *Journal of Intelligent and Fuzzy Systems, 45*(1), 147–175. https://doi.org/10.3233/JIFS-223435

Yağmur, E., & Kesen, S. E. (2024). Integrated production scheduling and vehicle routing problem with energy efficient strategies: Mathematical formulation and metaheuristic algorithms. *Expert Systems with Applications, 237*, Article 121586. https://doi.org/10.1016/j.eswa.2023.121586

Yang, Y. (2015). A modified biogeography-based optimization for the flexible job shop scheduling problem. *Mathematical Problems in Engineering, 2015*, Article 184643. https://doi.org/10.1155/2015/184643

Zandi, A., Ramezanian, R., & Monplaisir, L. (2020). Green parallel machines scheduling problem: A bi-objective model and a heuristic algorithm to obtain Pareto frontier. *Journal of the Operational Research Society, 71*(6), 967–978. https://doi.org/10.1080/01605682.2019.1595190

Zhang, C., Gu, P., & Jiang, P. (2015). Low-carbon scheduling and estimating for a flexible job shop based on carbon footprint and carbon efficiency of multi-job processing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 229*(2), 328–342. https://doi.org/10.1177/0954405414527959

Zheng, X., Zhou, S., Xu, R., & Chen, H. (2020). Energy-efficient scheduling for multi-objective two-stage flow shop using a hybrid ant colony optimisation algorithm. *International Journal of Production Research, 58*(13), 4103–4120. https://doi.org/10.1080/00207543.2019.1642529