Computers & Industrial Engineering 172 (2022) 108603

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Application of job shop scheduling approach in green patient flow optimization using a hybrid swarm intelligence

Masoumeh Vali [a], Khodakaram Salimifard [a], Amir H. Gandomi [b], Thierry J. Chaussalet [c,*]

[a] Computational Intelligence & Intelligent Research Group, Business & Economics School, Persian Gulf University, Bushehr 75168, Iran
[b] Faculty of Engineering & Information Technology, University of Technology Sydney, Ultimo, NSW, 2007, Australia
[c] Health and Social Care Modelling Group, School of Computer Science and Engineering, University of Westminster, London W1W 6UW, UK

ARTICLE INFO

Keywords:
Patient flow
Job shop scheduling
Swarm intelligence
Salp swarm algorithm
Chaotic map

ABSTRACT

With the increasing demand for hospital services amidst the COVID-19 pandemic, allocation of limited public resources and management of healthcare services are of paramount importance. In the field of patient flow scheduling, previous research primarily focused on classical-based objective functions, while ignoring environmental-based objective functions. This study presents a flexible job shop scheduling problem to optimize patient flow and, thereby, minimize the total carbon footprint, as the sustainability-based objective function. Since flexible job shop scheduling is an NP-hard problem, a metaheuristic optimization algorithm, called Chaotic Salp Swarm Algorithm Enhanced with Opposition-Based Learning and Sine Cosine (CSSAOS), was developed. The proposed algorithm integrates the Salp Swarm Algorithm (SSA) with chaotic maps to update the position of followers, the sine cosine algorithm to update the leader position, and opposition-based learning for a better exploration of the search space. generating more accurate solutions. The proposed method was successfully applied in a real-world case study and demonstrated better performance than other well-known metaheuristic algorithms, including differential evolution, genetic algorithm, grasshopper optimization algorithm, SSA based on opposition-based learning, quantum evolutionary SSA, and whale optimization algorithm. In addition, it was found that the proposed method is scalable to different sizes and complexities.

1. Introduction

Optimal flow, in terms of patient flow, is critical in providing quality care in healthcare environments, particularly hospitals. Enhancing patient flow is not only beneficial to healthcare providers, it also provides a way to refine health services and improve patient safety, outcomes, and satisfaction (Bacelar-Silva, Cox III, & Rodrigues, 2022; Leviner, 2020; Modi, 2007).

The wide use of complex equipment and technologies in various treatments, especially in hospitals, consumes a large amount of electricity and, thus, can increase CO_2 emissions (Brown, Buettner, & Canyon, 2012; MacNeill, Lillywhite, & Brown, 2017). CO_2 emissions from healthcare in the world's largest economies account for about 4 % of their national carbon footprints. Hospitals consume more energy than other nonresidential buildings per square meter of floor space, in part because of their continuous operation (Gaglia et al., 2007). According to Lancet Commission on Health and Climate Change, greenhouse gas (GHG) emissions of healthcare systems must be included as an indicator in assessments of health and climate (Watts et al., 2017). However, few studies have examined the emissions caused by the healthcare sector as well as potential mitigation strategies (McMichael, Neira, Bertollini, Campbell-Lendrum, & Hales, 2009; Watts et al., 2015).

Scheduling is used to allocate machines for various industrial processes and to determine processing sequences of products. Considering the emphasis on optimizing scheduling to reduce carbon emissions (Fang, Uhan, Zhao, & Sutherland, 2011; Liu & Huang, 2014; Yi, Li, Tang, & Wang, 2012; Zheng, Wang, & Wang, 2014), job shop scheduling problems (JSPs) are among the hardest combinatorial optimization problems even in a deterministic environment, in which all data are assumed to be fixed and precisely known in advance (Jain & Meeran, 1999). Since job shop scheduling optimization problems are NP-hard problems, intelligent algorithms (Jarosław, Czesław, & Dominik, 2013; Pan, 2012; Verma & Kaushal, 2017; Xiao & Konak, 2017) are commonly used. Pollard et. al. (Pollard, Paddle, Taylor, & Tillyard, 2014) proposed

* Corresponding author.
E-mail addresses: m.vali@mehr.pgu.ac.ir (M. Vali), salimifard@pgu.ac.ir (K. Salimifard), gandomi@uts.edu.au (A.H. Gandomi), chausst@westminster.ac.uk (T.J. Chaussalet).

https://doi.org/10.1016/j.cie.2022.108603
Received 27 July 2021; Received in revised form 21 July 2022; Accepted 22 August 2022
Available online 28 August 2022

a bottom-up modeling framework to help in the decision-making for both cost and carbon in healthcare, using data from a case study in Cornwall, UK. Research findings confirm that a bottom-up model is an efficient tool in the process of estimating and modeling the carbon footprint (CFP) of healthcare.

Most previous studies on patient flow scheduling primarily focused on classical objective functions, such as waiting time, length of stay, and patient throughput (Pham & Klinkert, 2008; Tai & Williams, 2012; White, Froehle, & Klassen, 2011; Wojtys, Schley, Overgaard, & Agbabian, 2009), but ignored environmental concerns. Therefore, this work applied a flexible job shop scheduling approach to model the green patient flow problem (GPFP). Considering that the job shop scheduling problem is NP-hard and complex, an improved intelligent algorithm is developed to prepare an efficient model.

The scientific contributions of this research are delivered through three different novelties in both modeling and solution. Firstly, to the best of our knowledge, it is the first paper that considers carbon emissions when patients are both receiving and waiting for treatments. It is important to note that while patients are kept waiting for the next treatment step, some electrical equipment is normally attached to them, resulting in electricity usage and CO2 production. Not only minimizing the sum of carbon emissions during the two periods is novel, but it is also critical from an application standpoint. Secondly, this study addresses carbon footprint in the patient flow, using an approach analogous to the FJSP. Thirdly, to solve the optimization problem we propose an improved evolutionary algorithm.

The remaining contents of this research are organized as follows. In Section 2, a literature review on the current studies is addressed. In Section 3, the case study is introduced. In Section 4, a bi-criterion green patient flow flexible job shop scheduling problem is described, and its mixed-integer programming model is constructed. In Section 5, the proposed algorithm, i.e. Chaotic Salp Swarm Algorithm Enhanced with Opposition-Based Learning and Sine Cosine (CSSAOS), is described. In Section 6, the datasets, parameter settings, and computational results are described, and the sensitivity analysis in terms of the number of patients in each category of ESI is presented. Finally, Section 7 discusses the findings of the research and draws conclusions based on the outputs of the research.

2. Related works

The delivery of healthcare services produces a surprising amount of greenhouse gas (GHG) emissions, which play an unequivocal factor in climate change and worldwide warming. Previous studies have addressed the environmental impacts of GHGs caused by the healthcare sector, primarily contributed to the large energy consumption of treatment to procedures. To illustrate, Gilliam et al. (Gilliam, Davidson, & Guest, 2008) estimated direct CO_2 emissions from laparoscopic surgeries, while Ryan and Nielsen (Ryan & Nielsen, 2010) determined the 20-year global warming potentials of three common anesthetic gases, including sevoflurane, isoflurane, and desflurane, through clinical scenarios to estimate the impacts on the environment. The consumption and generation of energy are associated with significant damages to the climate, environment, and, consequently, economy. In spite of the fact that power system dispatch incorporates a significate part in GHGs, especially CO_2 emissions, carbon capture power plants as a basic critical low-carbon generation option will have a vital effect on power system operation and dispatch. Ji et al. (Ji et al., 2013) introduced a model for low-carbon power system dispatch incorporating carbon capture power plants, demonstrating its effectiveness and validity using numerical examples based on an IEEE 118-bus tested system.

Based on a United Nations Framework Convention on Climate Change (UNFCCC) report, electricity production is responsible for 22 % of GHG emissions, 3 % of which is due to electric consumption by hospitals (Eckelman, Sherman, & MacNeill, 2018). In the United States, the healthcare system produces about 8–10 % of global GHG emissions (Chung & Meltzer, 2009; UNEP, 2012) while Canada is responsible for around 25 % (UNEP, 2012) climate change is one of the most important problems facing public health, healthcare services will continue to significantly generate GHGs. Therefore, reducing the environmental impacts and, of course, GHGs caused by the healthcare sector is one of the key responsibilities of the health sector in preventing global warming (Rossati, 2017). A previous study (Becker, 2012) estimated that American hospitals contributed 5.5 % of the total delivered energy to the commercial sector.

Concerning reports on the usage and amount of energy consumed in the healthcare sector (Becker, 2012; CBECS, 2012), optimization of this energy utilization can have a large impact on economic profit and, more importantly, on reducing greenhouse gases. Since the healthcare system has a high demand for electricity, such as for lighting, heating, ventilation, equipment, and air conditioning (Bi & Hansen, 2018; Bujak, 2010; Chirarattananon, Chaiwiwatworakul, Hien, Rakkwamsuk, & Kubaha, 2010; Chung & Meltzer, 2009; Renedo, Ortiz, Mañana, Silio, & Perez, 2006), medical centers have been recognized as one of the largest energy consumers and, thus, emitter of GHGs in the world. Particularly, hospitals consume more energy than other nonresidential buildings per square meter of floor space, as a result of their continuous operation (Gaglia et al., 2007). Based on a study by Yale School of Medicine (New Haven, CT) and Northeastern University (Boston, MA), the US healthcare system is a top producer of GHGs due to its energy consumption for electricity and heating (Review, 2016). To comply with the international (Kyoto and Paris) and national conventions to prevent global warming by reducing and managing emissions in such places is more probable. Ulli Weisz et al. (Weisz et al., 2020) proposed numerous untapped possibilities for reducing GHGs in healthcare services and determined six concrete steps toward sustainable healthcare that apply to most industrial countries. Moreover, life cycle assessments (LCA), as the most established approach to estimate GHG footprints from various treatments, individual products, locations, and industries, have been performed for associated industries (Belboom, Renzoni, Verjans, Léonard, & Germain, 2011; Eckelman, Mosher, Gonzalez, & Sherman, 2012; McGain & Naylor, 2014; Thiel et al., 2015; Usubharatana & Phungrassami, 2018), specific pharmaceuticals (McAlister et al., 2016; Parvatker et al., 2019; Wernet, Conradt, Isenring, Jiménez-González, & Hungerbühler, 2010), and medical procedures (Campion et al., 2012; Connor, Lillywhite, & Cooke, 2010; Danesh-Meyer, 2011; MacNeill et al., 2017).

Scheduling methods are based on operations research techniques, including optimization, mathematical modeling, forecasting, stochastic processes, and queue model. These techniques are used in scheduling staff, managing the patient flow, planning surgeries, and setting appointments.

Patient flow scheduling continuously remains one of the foremost vital issues within the healthcare framework (Hall, 2012). Liang et al. (Liang, Turkcan, Ceyhan, & Stuart, 2015) developed a discrete event simulation and mathematical programming model to evaluate the operational performance in an oncology clinic, which showed to reduce the total working times of clinics and patient waiting times while balancing resource utilization. Gupta and Denton (Gupta & Denton, 2008) introduced a state-of-the-art appointment scheduling system to manage access to service providers and demonstrated its potential for novel applications of Industrial Engineering and Operations Research (IE/OR) techniques.

Burdett and Kozan (Burdett & Kozan, 2018) proposed a flexible job shop scheduling (FJSS) model to effectively utilize hospital beds, operating rooms (OR), and other treatment spaces, considering patients, beds, hospital wards, and healthcare activities as jobs, single machines, parallel machines, and operations, respectively. The latter researchers developed hybrid and constructive metaheuristic algorithms to solve the FJSS problem and verified the potential of the scheduling model via integration in actual hospital information systems. Erhard et al. (Erhard, Schoenfelder, Fügener, & Brunner, 2018) reviewed the quantitative methods for physician scheduling in hospitals, including physician scheduling problems (e.g. staffing, rostering, re-planning, and personnel qualification) as well as shift types. The review, including 68 publications in operations research and management science fields, revealed the gaps that require future research activities.

As a result of increasing patient care and satisfaction, while reducing costs, Wright and Mahar (Wright & Mahar, 2013) examined the effect of centralizing scheduling decisions over departments in a clinic. The performance result of the centralized model showed an improvement in nurse management by 34 %, reduced overtime by 80 %, and minimized costs by just under 11 %. Other works (Piroozfard, Wong, & Wong, 2018; Zhang, Gu, & Jiang, 2015) on manufacturing emphasized GHGs reduction, especially CO_2 (environmental influence). For instance, Zhang et al. (Zhang et al., 2015) displayed a low-carbon scheduling model for the flexible job shop problem). The model was solved using a hybrid non-dominated sorting genetic algorithm II, and model efficiency was considering production components (i.e. makespan and machine workload) and environmental impact (i.e. carbon emission proven with some well-known benchmark instances and an actual case. In another research, Wang, et al. (Wang, Ding, Qiu, & Dong, 2011) displayed a low-carbon production scheduling system to minimize total CO_2 during the whole planning horizon, which was verified by computational experiments considering renewable energy.

Consequently, the main point of this investigation was to develop a mixed-integer programming model as a flexible job shop scheduling problem that can optimize an environmental-based objective function (total CFP) all over the scheduling time.

3. Case study

Our case study was conducted in Bushehr Heart Hospital (BHH) in southern Iran. The BHH has ten care units, including triage, cardiopulmonary resuscitation (CPR), inpatient emergency department (IED), coronary care unit I and II (CCUI and CCU II), post coronary care unit (PCCU), intensive care unit I and II (ICU I and ICU II), Catheterization Laboratory (Cath lab), and operating rooms (ORs). Also, it has two administration units, namely reception and discharge units, which have been conceptualized as workstations.

Based on the Emergency Severity Index (ESI), patients were categorized into resuscitation (ESI1), emergent (ESI2), urgent (ESI3), less urgent (ESI4), and non-urgent (ESI5) cases (Tanabe, Gimbel, Yarnold, & Adams, 2004). Using the obtained ESI of all patients, the routes that patients must follow from arrival time to discharge from the hospital were determined and are shown in Table 11 (in Appendix). As an example, a patient who is categorized in ESI1 category is immediately moved to CPR. If treatment is successful, the patient is transferred to the inpatient unit and then to the Cath lab. A patient with acute cardiovascular disease (ACS) and experiencing severe chest pain is categorized into ESI2 and is taken to Cath lab for angiography, then if necessary, receives Percutaneous Coronary Intervention (PCI). A patient who is categorized in ESI3 is transferred to the inpatient unit, but if needed angiography, he/she is taken to the Cath lab. A patient categorized into ESI4 is admitted to the emergency department for up to a 6-h stay and will be discharged if the treatment is successful, else he/she is transferred to the inpatient unit. A patient who is in the ESI5 category is referred to an outpatient clinic. The whole process is conceptually considered as patient flow. The average time of using the medical equipment in each care unit according to one patient and the amount of electricity consumed by the equipment (in KW/h) are given in Table 10 and Table 11 (in Appendix), respectively.

4. Mathematical formulation of green patient flow scheduling

In this research, an application of flexible job shop scheduling in green patient flow management (GPFM) was studied. A flexible job shop scheduling problem (FJSP) could be a generalized form of classical job shop scheduling (Pezzella, Morganti, & Ciaschetti, 2008), where each operation can be executed on a qualified machine. To comply with the methodology, we considered patients as jobs, treatment operations as operations, and medical equipment as machines.

To model the problem, we have used the analogy of the flexible job shop problem. It means that we see the problem as an FJSP instance, where patients resemble jobs, and treatment operations were seen as operations of a specific job. However, we did not intend to use all modeling aspects of the FJSP. In fact, the FJSP analogy helped us to understand the system and also helped in defining the solution procedure.

The FJSP is known to be an NP-hard combinatorial optimization problem (Garey, Johnson, & Sethi, 1976). FJSP consists of two sub-problems, the assignment and the scheduling problems. Using this FJSP analogy, we have considered that patient flow problem comprises two sub-problems, which are patient routing and sequencing. The patient routing problem, with the pseudocode represented in Algorithm 4, is concerned with assigning proper medical equipment to treatment operations from a set of eligible medical equipment to execute the treatment operations; the second sub-problem (sequencing) including algorithms 2 and 5 of section 5.6 involves ordering all treatment operations on all medical equipment. The total amount of CO_2 delivered by medical equipment is calculated from both processes.

4.1. Assumptions of the proposed method

The green patient flow as a flexible job shop problem is defined as follows. Consider a set of n given patients $J = \{J_{j=1}, J_2, J_3, \cdots, J_n\}$, where each patient consists of s_j sequenced treatment operations $O_j = \{O_{l=1,j}, O_{2,j}, O_{3,j}, \cdots, O_{s_j,j}\}$. Notation $O_{l,j}$ denotes the l^{th} *the* treatment operation of patient j, which should be executed on one medical equipment M_i from a set of eligible medical equipment $MC_{l,j} \subset \mathrm{M}$ ($M = \{M_{i=1}, M_2, M_3, \cdots, M_m\}$) with known execution time. The assumptions and constraints of the proposed problem are summarized as follows:

(1). Simultaneously executing more than one treatment operation on one piece of medical equipment is not permitted, i.e. to avoid overlap in treatment operations.

(2). A piece of medical equipment cannot be used in more than one treatment operation. Thus, more than one patient may be under the same treatment operation only if the required equipment is available for each of them, but a single piece of equipment cannot be shared between two or more operations or patients.

(3). Treatment operations are non-preemptable, i.e. once a treatment operation is commenced on medical equipment, it cannot be hindered or delayed.

(4). Medical equipment or patients are independent of each other, i.e. there is no relation among distinctive medical equipment or patients. In any case, priority connections and mechanical arrangements must be considered between the treatment operations of the same patients.

(5). A boundless buffer size between medical equipment is accepted.

(6). Medical equipment and patients are accepted to be accessible from the start.

(7). Medical equipment breakdown and preventive maintenance are not considered, i.e. types of medical equipment are ceaselessly accessible.

(8). Emitted CFP per kilowatt-hour is accepted to be constant.

(9). The length of stay in each care unit is equal to the length of bed occupancy in that care unit.

4.2. Mathematical model

To find the optimal solution for the green patient flow management scheduling (GPFMS) at BHH, mixed-integer linear programming (MIP) model was formulated. The GPFMS is similar to the flexible job shop scheduling problem (FJSP), which was used as a general framework for modeling patient flow.

The following lists include the indices, parameters, decision variables, and the mathematical model considered in this work. In addition, the constraints, notation, and parameters of the mathematical model are provided.

Indices:

i	Medical equipment index $(1, 2, \cdots, m)$
q	Bed index in inpatient units $(1, 2, \cdots, Q)$
i_q	Medical equipment related to treatment operation of bed q (1,2,…,m_q)
j	Patient index $(1, 2, \cdots, n)$
l	Index of treatment operations $(1, 2, \cdots, s_j)$
p	Priority index $(1, 2, \cdots, u_i)$
k	Care and administration unit index (1, 2,…, K)

Parameters:

A_j	Arrival time of patient j
m	Total number of medical equipment
n	Total number of patients
s_j	Total number of treatment operation s for the j^{th} patient
$MC_{l,j}$	A set of capable medical equipment for the l^{th} treatment operation of patient j
$P^i_{l,j}$	Processing time for the l^{th} the treatment operation of patient j that should be executed on medical equipment i
Pm_i	Power consumption of medical equipment i in the processing condition
EF	Quantity of emitted CFP per kilowatt-hour
B	It is assumed as a big number
C_k	The capacity of the care unit (Number of beds in the treatment unit(

Decision variables:

WT^q_j	Waiting time of patient j on inpatient bed q
$F^i_{l,j}$	Finish time of the l^{th}*the* treatment operation of patient j on medical equipment i
$BO^i_{l,j}$	Beginning time of the l^{th} the treatment operation of patient j on medical equipment i
Bm^p_i	Beginning time of the i^{th} medical equipment in the p^{th} priority
Z_1	The total emitted CFP of the solution (TECF)
$X_{l,j,i,p} =$	$\begin{cases} 1, \textit{If the the } l^{th} \textit{ treatment operation of patient } j \textit{ is executed on the } i^{th} \textit{medicalequipment in the } p^{th} \textit{ priority} \\ 0, \textit{otherwise} \end{cases}$
$h_{l,j,i} =$	$\begin{cases} 1, \textit{If the the } l^{th} \textit{ the treatment operation of patient } j \textit{ is assigned on the } i^{th} \textit{ bed} \\ 0, \textit{otherwise} \end{cases}$
$U_{i_q,j} =$	$\begin{cases} 1, 1, \textit{If patient } j \textit{ has used medical equipment related to treatment operation of bed } q. \\ 0, \textit{otherwise} \end{cases}$

The mathematical model consists of the objective function and constraints to capture the reality of the system. The modeling requirements of FJSP are considered in constraints defined in Eqs. (4, 6, 7, 11, 12). Constraints (8, 9, 13) represent the assignment sub-problem while the sequencing constraints are defined in constraints (5, 10, 14, 15).

$$\min Z = \sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{l=1}^{s_j}\sum_{p=1}^{u_i} EF^*Pm_i{}^*P^i_{l,j}{}^*X_{l,j,i,p} +.$$

$$\sum_{j=1}^{n}\sum_{i_q=1}^{m_q}\sum_{l=1}^{s_j} EF^*Pm_{i_q}{}^*P^{i_q}_{l,j}{}^*WT^q_j{}^*U_{i_q,j} \tag{3}$$

$$BO^i_{l,j} \geq A_j, l = 1; \forall i = 1, 2, \cdots, m; \forall j = 1, 2, \cdots, n \tag{4}$$

$$BO^i_{l,j} + P^i_{l,j} \leq BO^{i'}_{l+1,j}, \ \forall i = 1, \cdots, i', \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j - 1 \tag{5}$$

$$Bm^p_i + P^i_{l,j}X_{l,j,i,p} \leq Bm^{p+1}_i, \ \forall i = 1, \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j, \forall p = 1, 2, \cdots, u_i - 1 \tag{6}$$

$$Bm^p_i \leq BO^i_{l,j} + \left(1 - X_{l,j,i,p}\right)B, \ \forall i = 1, \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j, \forall p = 1, 2, \cdots, u_i \tag{7}$$

$$X_{l,j,i,p} + (X_{l,j,i,p} - 1)B \leq h_{l,j,i}, \ \forall i = 1, \cdots, m; \ \forall j = 1, 2, \cdots, n; \ \forall l = 1, 2, \cdots, s_j, \ \forall p = 1, 2, \cdots, u_i \tag{8}$$

$$\sum_j\sum_l X_{l,j,i,p} = 1, \ \forall i = 1, 2, .., m; \forall p = 1, 2, \cdots, u_i \tag{9}$$

$$\sum_{i \in MC_{l,j}}\sum_p X_{l,j,i,p} = 1, \ \forall j = 1, 2, .., n; \forall l = 1, 2, \cdots, s_j \tag{10}$$

$$MC_{l,j} \subset M, \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j \tag{11}$$

$$F^i_{l,j} = BO^i_{l,j} + \sum_j\sum_{q \in l} P^i_{q,j}X_{l,j,i,p}, \forall i = 1, 2, \cdots, m; \forall l = 1, 2, \cdots, s_j, \forall p = 1, 2, \cdots, u_i \tag{12}$$

$$\sum_q\sum_j\sum_l h_{q,j,l} \leq C_k, \forall k \in \{1, 2, \cdots, K\}, \tag{13}$$

$$WT^q_j = BO^{q+1}_{l,j} - F^q_{l-1,j}, \ \forall i = 1, 2, \cdots, m; \forall j = 1, 2, \cdots, n, \forall l = 1, 2, \cdots, s_j, \forall q = 1, 2, \cdots, Q \tag{14}$$

$$BO^i_{l,j} \geq 0, Bm^p_i \geq 0, \forall i = 1, 2, \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j, \forall p = 1, 2, \cdots, u_i \tag{15}$$

$$F^i_{l,j} > 0, \forall i = 1, 2, \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j \tag{16}$$

$$X_{l,j,i,p} \in \{0, 1\}, \ \forall i = 1, 2, \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j, \forall p = 1, 2, \cdots, u_i \tag{17}$$

$$h_{l,j,i} \in \{0, 1\}, \ \forall i = 1, 2, \cdots, m; \forall j = 1, 2, \cdots, n; \forall l = 1, 2, \cdots, s_j \tag{18}$$

The objective function, defined in Eq. (3), accounts for the minimization of the total emitted CFP obtained by calculating the emitted CFP of medical equipment for each patient and the carbon produced due to the patient's waiting time in the care units. While it is possible to assign different weights to the two components of the objective function, we have used here equal weights to reflect the hospital authority's preference not to prioritize either component.

Eq. (4) guarantees that the beginning time of the first patient's treatment operation will be longer than the arrival time. Eq. (5) ensures that the start time of a preceding treatment operation ($BO^i_{l,j}$) plus its execution time ($P^i_{l,j}$) is less than or equal to the start time of the beginning time of the consequent treatment operation of the same patient

Table 1
Points of interest of Chaotic Maps applied on CSSAOS.

No.	Map Name	Map Equation
1	Logistic map	$N(i+1) = 4N(i)(1 - N(i))$
2	Cubic map	$N(i+1) = 2.59\, N(i)(1 - N(i)^2)$
3	Sine map	$N(i+1) = \sin(\pi N(i))$
4	Sinusoidal map	$N(i+1) = 2.3\, N(i)^2 \sin(\pi N(i))$
5	Singer map	$N(i+1) = 1.073(7.86\, N(i) - 23.31 N(i)^2 + 28.75 N(i)^3 - 13.302875 N(i)^4)$
6	Tent map	$N(i+1) = \begin{cases} \frac{N(i)}{0.4}, 0 < N(i) \leq 0.4 \\ \frac{1 - N(i)}{0.6}, 0.4 < N(i) \leq 1 \end{cases}$
7	Gaussian map	$N(i+1) = \begin{cases} 0, N(i) = 0 \\ \left(\frac{1}{N(i)}\right) mod(1), N(i) \neq 0 \end{cases}$
8	Chebyshev map	$N(i+1) = \cos(0.5\, cos^{-1} N(i))$
9	Bernoulli map	$N(i+1) = \begin{cases} \frac{N(i)}{0.6}, 0 < N(i) \leq 0.6 \\ \frac{(N(i) - 0.6)}{0.4}, 0.6 < N(i) \leq\leq 1 \end{cases}$
10	Circle map	$N(i+1) = N(i) + 0.5 - \frac{1.1}{\pi} \sin(2\pi N(i)) mod(1)$

$(BO^{i'}_{l+1,j})$, i.e. to fulfill the priority relationship between diverse treatment operations of the same understanding whereas guaranteeing no relation between treatment operations of distinctive patients.

Eq. (6) guarantees that medical equipment cannot commence preparing diverse treatment operations at the same time, i.e. to prevent overlap in medical equipment. Eq. (7) guarantees that the medical equipment for processing the l^{th} the treatment operation of patient j $(O_{l,j})$ is empty and its previous treatment operation $O_{l-1,j}$ is already processed, i.e. to prevent overlap in treatment operations. In Eq. (8), the appropriate type of medical equipment is determined for each treatment operation of patients. Eq. (9) assigns the treatment operations of the patients to their medical equipment, and subsequently, all treatment operations are arranged on all medical equipment while taking into account the priority of treatment operations on medical equipment. Eq. (10) limits each treatment operation to be executed on a single capable medical equipment with one priority, i.e. each assigned medical equipment for processing a treatment operation has one priority.

Eq. (11) ensures that the eligible set of medical equipment for executing $O_{l,j}$ is from the given set of medical equipment $MC_{l,j}$. Eq. (12) specifies that once a treatment operation is commenced, it should be performed without interruption, i.e. treatment operations are non-preemptable. Eq. (13) indicates that the number of occupied beds in each hospitalized care unit shall be less than or equal to the capacity of that hospitalized care unit. Eq. (14) shows the waiting time of patient j on an inpatient bed q. Eq. (15) indicates that the start time of $O_{l,j}$ on medical equipment i and start time of medical equipment i in priority p is bigger than or equal to zero. Eq. (16) express that the finish time of $O_{l,j}$ on medical equipment i should be bigger than or equal to zero. Eqs. (17)-(18) specify that the decision variables including $X_{l,j,i,p}$ and $h_{l,j,i}$ are binary.

As stated in Section 4, the objective function minimizes carbon emitted both during the treatment process and during the time patients are waiting for the next treatment step. As the objective function has two components, it could be modeled with a bi-objective optimization model and then solved using multi-objective metaheuristic algorithms. However both components are of environmental concern and are independent of each other, i.e. an increase in one component does not result in a decrease in the other. In fact, a multi-objective solution algorithm could be used if the optimal decisions need to be taken in the presence of tradeoffs between two or more conflicting objectives.

Although in an FJSP, machine idle time is of concern, in this research beds and equipment idle time have not been concerned. The reason is that there is no CO_2 emission as long as they are not used in the treatment process. The usage time of equipment is based on the patient's ESI, as shown in Table 11.

5. Proposed CSSAOS algorithm

To solve the problem at hand, an evolutionary algorithm, called Chaotic Salp Swarm Algorithm Enhanced with Opposition-Based Learning and Sine Cosine (CSSAOS), is proposed to solve the presented objective flexible job shop scheduling problem (FJSP). The SSA, recently introduced by Seyedali Mirjalili et al. (Mirjalili et al., 2017), is prone to problems, such as a slow convergence rate and local optimal solution, like most metaheuristic algorithms. To solve these problems, SSA is integrated with the chaotic maps, sine cosine algorithm (SCA), and opposition-based learning (OBL).

In this area, the proposed strategy is presented, at that point, the encoding and decoding of a solution are completely clarified as a vital portion of the algorithm. Hence, a stepwise outline of the calculation is given. To depict the proposed algorithm, an outline of SSA, SCA, OBL, and chaotic maps are brought as takes after.

5.1. Salp swarm algorithm

SSA mirrors the swarm behavior of salps, alter their position design utilizing quick agreeable changes to rummage around for nourishment (Anderson & Bone, 1980; Sutherland & Weihs, 2017). The populace of salps is partitioned into two bunches: leader and followers. The leader flies at the front of the chain to assign another movement for the rest of the salps, or devotees, to mimic. The position of salps is characterized in an n-dimensional search space, where n is the number of variables of a given problem and food source F is the target. Thus, the position of all salps is stored in a two-dimensional matrix called x. Based on the taking after equation, the position of the leader is overhauled:

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right)c_3 \geq 0 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right)c_3 < 0 \end{cases} \tag{19}$$

where x_j^1 is the position of the primary salp (called leader) in the j^{th} dimension; F_j is the position of the food source within the j^{th} dimension; ub_j is the upper bound of the j^{th} dimension; lb_j is the lower bound of the j^{th} dimension and c_1, c_2, and c_3 are random numbers. The coefficient c_1 is the most important parameter in SSA because it balances exploration and exploitation, which is characterized as follows:

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{20}$$

where l is the current iteration; and L is the maximum number of iterations. Newton's law of motion is used to update the position of the follower as:

$$x_j^i = \frac{1}{2}at^2 + v_0 t \tag{21}$$

where $i \geq 2, x_j^i$ is the position of the i^{th} follower salp within the j^{th} dimension; t is time; v_0 is the initial speed; and $a = \frac{v_{final}}{v_0}$ where $v = \frac{x - x_0}{t}$. Since the time for optimization is an iteration, the discrepancy between iterations is equal to 1. Considering $v_0 = 0$, the position of the i^{th} follower salp in the j^{th} dimension is expressed as follows:

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) i \geq 2 \tag{22}$$

5.2. Sine cosine calculation for upgrading the leader's position

The SCA could be a population-based optimization method and a modern metaheuristic calculation (Mirjalili, 2016), the solutions are updated based on the sine or cosine function as in equations (23):

$$X_i = \begin{cases} X_i + r_1 * \sin(r_2) * |r_3 P_i - X_i| r_4 < 0.5 \\ X_i + r_1 * \cos(r_2) * |r_3 P_i - X_i| r_4 \geq 0.5 \end{cases} \tag{23}$$

where P_i is the destination solution, X_i is the current solution; $||$ demonstrates the absolute value; and r_1, r_2, r_3 and r_4 are random variables. The parameter r_1 is upgraded utilizing Eq. (24) to adjust exploration and exploitation (Mirjalili, 2016), as follows:

$$r_1 = a - t\frac{a}{T} \tag{24}$$

where T is the maximum number of iterations; a is a constant; and t is the current iteration. The r_2 is a random variable utilized to discover the direction of the movement of the next solution (i.e. if it moves towards or away from P_i). Moreover, r_3 is a random variable that gives random weights for P_i to stochastically emphasize ($r_3 > 1$) or deemphasize ($r_3 < 1$) the impact of desalination in characterizing the distance. Moreover, r_4 is utilized to switch between the sine and cosine functions as in Eq. (23).

5.3. Chaotic maps

In recent studies, chaotic generators have been chosen over random number generators (RNGs) as RNGs are not completely random (Caponetto, Fortuna, Fazzino, & Xibilia, 2003). Therefore, this work employed various chaotic maps to update the followers' position and form a new solution, as listed in Table 1.

The chaotic maps are used to update the followers' position. For each follower i, its next position is calculated using Eq. (25).

$$x_j^i = N(i+1) * \left(\frac{x_j^i + x_j^{i-1}}{2}\right) i \geq 2 \tag{25}$$

where $N(i+1)$ is calculated using a chaotic map taken from Table 1; and x_j^i is the position of the i^{th} follower salp within the j^{th} dimension. Other than that, the salp population X is built by the chaotic maps utilizing Eq. (26).

$$x_j^i = N(i+1) * x_j^i \tag{26}$$

5.4. Opposition-based learning

OBL describes a contrary solution to the current solution, and after that assesses the fitness function to accept or reject the new solution (Tizhoosh, 2005), as follows:

$$\overline{x_j^i} = U_j + L_j - x_j^i, j = 1, \cdots, dimention \tag{27}$$

where $\overline{x_j^i}$ is the opposite vector from the real vector x_j^i; and x_j^i is characterized as a real number over the interval $x_j^i \in [L_j, U_j]$.

5.5. Chaotic salp swarm algorithm enhanced with opposition-based learning and sine cosine

Within the proposed strategy, SCA is utilized for upgrading the position of the leader, the chaotic maps to update the position of followers, and OBL for a better exploration of the search space, generating more accurate solutions. Based on the 10 chaotic maps provided in Table 1, different CSSAOS algorithms, CSSAOS1 to CSSAOS10, are introduced. The pseudo-code of the CSSAOS algorithm is outlined in Algorithm 1.

Algorithm 1. Pseudo-code of CSSAOS

```
Initialize the randomly generated population of the salp swarm $X_{ini}(ini = 1, 2, \cdots, n)$
Calculate opposite point $X_{oi}$ of $X_i$.
Calculate the fitness value $f(X_{ini})$ and $f(X_{oi})$ of $X_{ini}$ and $X_{oi}$.
If $f(X_{oi}) \leq f(X_{ini})$
$X_{ini} = X_{oi}$
else
Using the chaotic maps (Eq. (26)) to form a new population of the salp swarm $X_i$ of $X_{ini}$
If $f(X_i) \leq f(X_{ini})$
$X_{ini} = X_i$
end if
$X^*$ = the best search agent.
while(end condition is not satisfied)
Update $r_1$ by Eq. (6)
  For each salp ($x_i$)
    If (i==1)
      Update the position of the leading salp by Eq. (23)
    else
      Update the position of the follower salp by Eq.(25)
    end
  end
reposition the salps which go out search space based on lower and upper bounds
of problem variables
Update X* if there is a better solution.
Using opposition-based learning to form another new solution $X_{new1}$ by Eq. (27)
Calculate the fitness value $f(X_{new1})$, $f(X^*)$ of $X_{new1}$ and $X^*$
  If $f(X_{new1}) \leq f(X^*)$
    $X^* = X_{new1}$
  else
    Using chaotic maps to form another new solution $X_{new2}$ by Eq. (26)
      If $f(X_{new2}) \leq f(X^*)$
        $X^* = X_{new2}$
  end
end
end while
Return the best solution X* and its fitness value f(X*).
```

5.6. Chromosomes representation

The FJSP is an NP-hard problem consisting of two sub-problems, which are the assignment and the scheduling problems. The FJSP analogy is utilized to see the system in which a different number of jobs (patients) is be processed (treated) on different numbers of machines (treatment units) at the same time.

In this research, two chromosomes are defined to represent a solution. First, the routing chromosome captures the path that will be followed by patients based on their ESIs. Second, the ranking chromosome is used to prioritize bed allocation to patients in all units.

The whole process can be summarized in three steps as follows:

(1). Define the patient path chromosome and priority chromosomes.
(2). Schedule bed allocation to the patient based on priority chromosomes.
(3). Determine the amount of carbon consumed by each patient, which is the total carbon produced by the services and waiting time.

In the first step, the algorithm sets a uniformly distributed random number $r \in U(0, 1)$ to each gene of the chromosomes (random numbers are different for the routing and ranking chromosomes). The chromosomes are divided into n (total number of patients) segments, each representing one patient. The number of genes in each segment is equal to the number of treatment units that each patient will go through in the treatment operation.

In the second step, the beds are assigned to the patients by decoding the first chromosome using Eq. (28):

$$f(x) : [0, 1] \rightarrow [q_l, q_h]; \; xnew = \lfloor (q_h - q_l)x + (q_l + 1), n \rfloor \tag{28}$$

Table 2
Treatment pathways.

Treatment pathways	ESI
treatment unit1 → treatment unit 3 →treatment unit2→treatment unit 4	1
treatment unit 2 →treatment unit 3→treatment unit 1	2
treatment unit 1 →treatment unit 3→treatment unit 4	3
treatment unit 2→treatment unit 4	4
treatment unit 1→treatment unit 2	5

where $\lfloor . \rfloor$ is the floor function; and q_l and q_h indicate the first and last bed index of the corresponding care unit, respectively. The amount of carbon dioxide produced is calculated in the third step via Algorithm 6.

The most rationale of the proposed calculation is portrayed in Algorithm 2, which consists of six steps, each with a specific set of operations.

Algorithm 2. Pseudo-code of Top-Level Algorithm

```
Grand = setRandomGene() (for Routing Chromosome and Rank Chromosome)
Grank = rankGenes() (for priority chromosomes)
Prank = setPatientOrder()
Baloc = allocateBeds()
Psche = schedulePatients()
Pcarbon = CalculateCarbon()
```

Both routing and ranking chromosomes are initialized with random numbers, which are defined as Grand. The rank chromosome is valued using Algorithm 3.

Algorithm 3. Pseudo-code of setGenesOrder()

```
DO
  rg(1) = rand()
  FOR k = 2: chromosomeLength
    rg(k) = rand() & rg(k) <> rg(k-1)
  END FOR
//assigns a rank value to each gene based on its randomly-set value
  FOR i = 1: chromosomeLength
    g(i) = rank(rg(i))
  END FOR
END DO
```

Bed assignment is based on patients' priorities. A patient with higher priority should be allocated to a bed before other patients. Algorithm 4 is used to determine the priority of patients. This algorithm defines the ordered set of patients based on their priority. The priority then will be used for sequencing patients and allocating beds.

Algorithm 4. Pseudo-code of setPatientOrder()

```
DO
  FOR each patient p
    DO
      Sg(p) = Find smallest rank among its gene
    END DO
      sortSg[] = Sort Sg(p)s
  END FOR
  // the ordered set of patients
  patientsOrder = SortSg[]
  OUTPUT patientsOrder
END DO
```

In allocating beds to the patient, it is critical to consider the plausibility of patient boarding, due to the unavailability of required beds in the destination care unit. Based on the assignment sub-problem of the FJSP, Algorithm 5 is used to allocate beds to the patient and determine if the patient needs to wait for an unoccupied bed. This algorithm ensures that simultaneous execution of more than one treatment operation using a specific medical device is not allowed.

Algorithm 5. Pseudo-code of allocateBeds()

```
DO
  FOR each patient p in patientsOrder
    FOR each unit required for patient p
      IF a bed b is available THEN
        Allocate bed b to patient p
        SET allocPatient = NOW()
      ELSE
        SET patient p in boarding
        SET patientWait = NOW()
      END IF
    END FOR
    OUTPUT allocation gene, boarding patients
  END FOR
END DO
```

The timing of the whole care process needs to be scheduled. Using the concept of the scheduling sub-problem of the FJSP, Algorithm 6 determines the total time that a bed is occupied by a patient and the boarding time of a patient. It shows that the length of stay of a patient in each treatment unit is equal to the duration of bed occupancy in that unit. Also, the priority relationship between different treatment operations shows that there is no connection between different treatment operations. If the next ward does not have an unoccupied bed, the patient will wait in the previous unit until a bed becomes available in the destination unit.

Algorithm 6. Pseudo-code of schedulePatients()

```
DO
  FOR each patient p
    INITIALIZE bedStart, bedEnd, bedWait
    FOR each bed b
      bedStart = allocPatient
      bedWait = NOW() - boardingTime
      bedEnd = bedStart + patientOperationDuration
    END FOR
  END FOR
  OUTPUT bedStart(p), bedWait(p), bedEnd(p)
END DO
```

The amount of carbon consumed due to the use of electrical equipment during services and boarding time is calculated using Algorithm 7.

Algorithm 7. Pseudo-code for calculating carbon per bed

```
DO
  FOR each bed assigned to each patient j
    The produced carbon is calculated by using Eq. (29)
  End FOR
    The total carbon produced per bed is calculated by using Eq. (30)
End DO
```

$$EF^{*}Pm_{\boldsymbol{i}}{}^{*}P^{\boldsymbol{i}}_{l,j}{}^{*}X_{l,j,i} + EF^{*}Pm_{\boldsymbol{i_q}}{}^{*}P^{\boldsymbol{i_q}}_{l,j}{}^{*}WT^{q}_{j}{}^{*}U_{\boldsymbol{i_q},j} \tag{29}$$

$$\sum_{i=1}^{m} EF^{*}Pm_{\boldsymbol{i}}{}^{*}P^{\boldsymbol{i}}_{l,j}{}^{*}X_{l,j,i} + \sum_{\boldsymbol{i_q}=1}^{m_q} Pm_{\boldsymbol{i_q}}{}^{*}P^{\boldsymbol{i_q}}_{l,j}{}^{*}WT^{q}_{j}{}^{*}U_{\boldsymbol{i_q},j}$$

Eq. (29) shows the amount of carbon dioxide produced during the treatment and boarding of each patient for each bed. Eq. (30) determines the total CO_2 produced from the treatment of a patient.

5.7. A non-trivial example

Suppose we have 5 patients with 4 treatment units and 15 beds, and 5 treatment pathways are defined based on the ESI of patients as represented in Table 2.

The bed in each treatment unit is marked with a numerical index. Beds with index numbers 1 to 4 belong to treatment unit 1; beds with

.43	.27	.42	.79	■ .65	.52	■ .83	.11	■ .25	.50	.93	■ .85	.78	.02	.67

Fig. 1. The chromosome representing patients' paths.

	Patient 1				■ Patient 2		■ Patient 3		■ Patient 4			■ Patient 5			
treatment pathways	1	3	2	4	2	4	1	2	2	3	1	1	3	2	4
B_{lm}	1	8	3	12	3	12	1	3	3	8	1	1	8	3	12
B_{ln}	2	11	7	15	7	15	2	7	7	11	2	2	11	7	15
chromosome	.43	.27	.42	.79	.65	.52	.83	.11	.25	.50	.93	.85	.78	.02	.67

Fig. 2. Matrix to assign beds to patients.

2	9	5	15	■ 6	14	■ 2	4	■ 4	10	2	■ 2	11	3	14

Fig. 3. Decoded allocation chromosome.

.45	.3	.44	.81	■ .67	.54	■ .85	.13	■ .27	.52	.95	■ .87	. 8	.08	.69

Fig. 4. Rank-based chromosome.

6	4	5	12	■ 9	8	■ 13	2	■ 3	7	15	■ 14	11	1	10

Fig. 5. Rank-based allocation.

Table 3
Average time and bed numbers in each treatment unit.

ESI	Unit Number			
	1	2	3	4
1	[82,1–3]	[450,4–5]	[5760,6–10]	[4320,11–15]
2	[300,1–3]	[380, 4–5]	[5040, 6–10]	[2880, 11–15]
3	[240,1–3]	[190, 4–5]	[4320, 6–10]	[1440, 11–15]
4	[240,1–3]	[0, 4–5]	[2160, 6–10]	[1440, 11–15]
5	[0,1–3]	[0, 4–5]	[0, 6–10]	[0, 11–15]

Table 4
Time of allocating the bed to patient 5.

Bed index	operation Start time	Operation finish time	Boarding Time
2	19	101	0
11	101	4421	0
3	4421	4503	131
14	4634	8964	0

index numbers 5 to 6 belong to treatment unit 2; beds with index numbers 7 to 10 belong to treatment unit 3; and beds with index numbers 10 to 15 belong to treatment unit 4. Also, assume that patients 1 and 5 are categorized in ESI1, patient 2 in ESI4, patient 3 in ESI3, and patient 4 in ESI2 categories. The arrival times of patients 1–5 are 67, 1034, 108, 97, and 19 min, respectively. The chromosome for the above five patients, generated using *setRandomGene()*, is illustrated in Fig. 1.

As seen in Fig. 1, the chromosome is divided into five segments, each segment represents one patient. Therefore, the *chromosome* represents five patients. The number of genes in each segment is equal to the number of treatment units that the patient will go through in the treatment process.

In the next stage of the procedure, the beds are assigned to patients by decoding the following matrix:

In Fig. 2, the first row of the matrix shows the treatment pathways for the five patients. The second row represents the index number of the first bed in the corresponding treatment unit (B_{lm}), while the third row represents the index of the last bed of the treatment unit (B_{ln}). The chromosome is represented in the fourth row of the matrix. Equation (31) is used to convert the matrix to the decoded chromosome:

$$f(x) : [0, 1] \rightarrow [B_{lm}, B_{ln}] \tag{31}$$

$$xnew = \lfloor (B_{ln} - B_{lm})x + (B_{lm} + 1), B_{ln} \rfloor$$

where ⌊ ⌋ is the correct component function; B_{lm} is the index number of the first bed; and B_{ln} is the index number of the last bed of the corresponding treatment unit *l*. Fig. 3 presents the allocation of beds for each patient.

To consider the priority of each patient in assigning beds and performing the required treatment, the ranking chromosome is defined in Fig. 4.

As shown in Fig. 5, using rank-based *rankGenes()*, the priority of the patient is determined.

For example, patient 5 with rank 1 in locus 3 is the first to be allocated on bed number 3, as seen in Fig. 5. If the target gene of the corresponding segment of the decoded allocation chromosome is in locus 1, the bed will be allocated directly. Otherwise, if the target gene of the corresponding segment is in locus 2 or greater, all the genes before the target should be allocated as well to ensure that the patient is allocated all the required beds in the corresponding treatment units. Using the same procedures, all beds are allocated to all patients.

In Table 3, cell [82, 1–3] shows that a patient in ESI1 (row 1) goes to unit 1, and the total time of the care process will be 82 min. If available, beds 1, 2, or 3 can be allocated to the patient. The other cells can be read

Table 5
level of parameters for Taguchi method.

parameter	level of parameters		
	lower	median	highest
MaxIt	50	100	150
Number of agents	30	40	50

accordingly.

As seen in Table 4, patient 5 arrives at time 19 min in unit 2 and is allocated bed 2. He then goes to unit 4 with no waiting time and then back to unit 1 at time 4421. The care process is finished in unit 1 at time 4503, then the patient goes to unit 4 and is allocated bed 14. Since there is no vacant bed, the patient must wait 131 min (his boarding time) until the bed is available at 4634. The whole care process is finally completed at time 8964. In order to calculate the total carbon footprint for the patient, Eq. (30) is used.

6. Experimental results and analysis

To illustrate the legitimacy of the displayed model and the viability of the proposed solution approach, a few numerical tests of diverse sizes were executed with the CSSAOS algorithms. The results are compared to a few other metaheuristic algorithms including Differential Evolution (DE) (Chakraborty, 2008), Genetic Algorithm (GA) (Mirjalili, 2019), Grasshopper Optimization Algorithm (GOA) (Saremi, Mirjalili, & Lewis, 2017), Salp Swarm Algorithm based on opposition-based learning (OSSA) (Bairathi & Gopalani, 2018), Quantum Evolutionary Salp Swarm Algorithm (QSSA) (Chen, Dong, Ye, Chen, & Liu, 2019), Salp Swarm Algorithm (SSA), and Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016).

6.1. Parameter tuning

To detect the optimum level for parameters of algorithms, the Taguchi method (Li & Kwan, 2004) was used. Also, the signal-to-noise ratio (Fattahi, Hajipour, & Nobari, 2015), as stated in Eq. (32), was

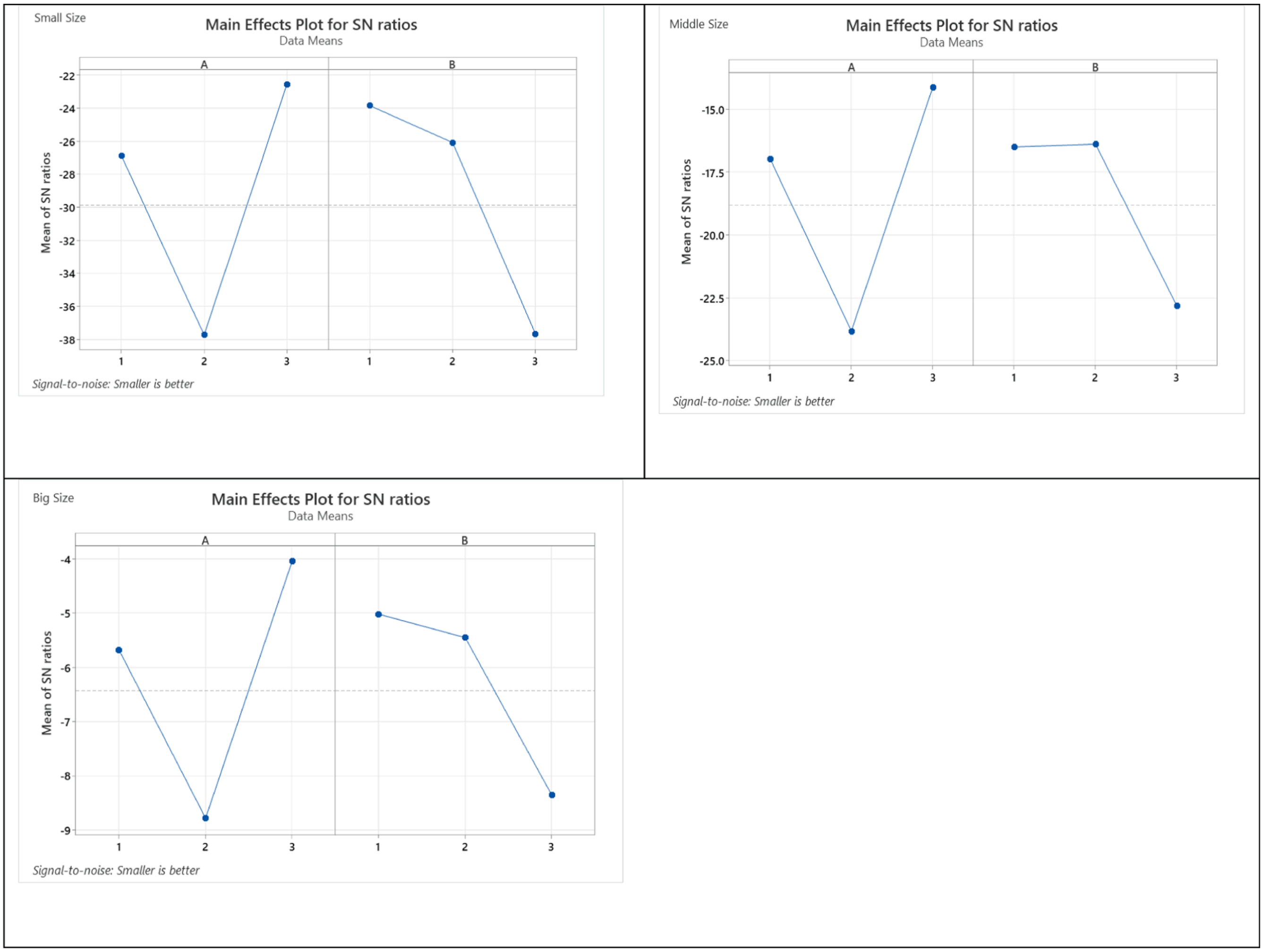

Fig. 6. The output of the Taguchi method.

Table 6
Solving test problem in different sizes.

	Test Problem1: Small Size				**Test Problem 2: Small Size**			
Algorithms	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**
CSSAOS1	4.95462*	5.09823	0.25243	3.42719	10.50450	11.53924*	0.64586	4.19587
CSSAOS2	4.95462*	5.02522	0.14214	3.19873	10.30168*	11.78454	0.65711	4.46177
CSSAOS3	4.95462*	5.15481	0.26735	3.07558*	10.71088	11.65881	0.61619	4.25497
CSSAOS4	4.95462*	5.00422	0.13483	4.66231	10.57119	11.76771	0.65537	6.74461
CSSAOS5	4.95462*	5.08514	0.19293	3.16068**	10.54771	11.64095	0.65898	4.17811
CSSAOS6	4.95462*	5.14663	0.28283	3.22157	10.52941	11.60803	0.60690	4.03307*
CSSAOS7	4.95462*	5.11154	0.23837	3.19282	10.73010	11.87738	0.58545	4.20391
CSSAOS8	4.95462*	5.02351**	0.11506**	3.23448	10.64256	11.67628	0.64038	4.32907
CSSAOS9	4.95462*	5.05425	0.18648	3.28945	10.57424	11.66554	0.56254**	4.35221
CSSAOS10	4.95462*	5.07098	0.22419	3.215317	10.77188	12.09493	0.73497	4.08398**
SSA	4.95462*	5.12357	0.30233	14.27568	10.81245	11.79960	0.59210	14.61729
QSSA	5.04423	5.45239	0.29695	15.31335	12.08887	13.56507	0.68118	15.87838
OSSA	4.95462*	4.98199*	0.03608*	14.63205	11.01619	11.75203	0.47273*	16.30203
GA	4.95462*	5.04230	0.22364	16.76782	10.97684	11.69417	0.57902	16.85738
DE	4.98024**	5.66531	0.34963	16.35999	12.70304	14.42354	0.86426	15.80508
GOA	4.95462*	5.57136	0.32336	17.61514	11.57251	12.82854	0.96563	20.02698
WOA	4.95462*	5.28127	0.36037	18.12555	10.34580**	11.59737**	0.63316	16.25468

	Test Problem 3: Small Size				**Test Problem 4: Middle Size**			
Algorithms	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**
CSSAOS1	27.58613	29.29741**	1.08608	5.51658	45.70465	50.85359	2.95479**	7.84334**
CSSAOS2	27.05187	29.29169*	1.45029	5.68013	77.6347	86.17899	4.63922	8.25795
CSSAOS3	27.67885	29.39274	1.03426**	5.48280	77.10125	84.79355	4.61847	8.76991
CSSAOS4	27.01771	29.45030	1.37642	11.76362	79.4265	88.01630	5.41550	11.58847
CSSAOS5	27.46443	29.31886	1.15866	5.03614**	77.56566	85.19823	3.14258	7.80384*
CSSAOS6	26.86968*	29.40598	1.25169	5.04853	79.52092	87.27901	6.02904	8.07422
CSSAOS7	27.58277	29.90877	1.04923	5.24388	78.40614	85.73644	5.03897	8.45849
CSSAOS8	27.76425	29.36942	1.13383	5.38938	79.78749	86.74487	4.95683	7.99706
CSSAOS9	27.61693	29.46765	1.44061	4.97582*	77.31262	84.26181	4.34442	8.07542
CSSAOS10	27.73863	30.42685	1.97994	5.41004	45.70465	50.85359	2.95479**	7.84334**
SSA	27.57067	29.47222	1.14195	15.03834	42.03479	49.29641	3.28988	17.65376
QSSA	30.32645	33.73051	2.78883	16.10209	47.65076	57.83452	4.14308	20.07047
OSSA	27.45752	29.77187	1.58429	15.55314	41.89195**	47.93723**	3.17872	20.88283
GA	27.03408	28.49410	0.90817*	17.32577	40.86146*	45.53817*	2.42029*	22.44738
DE	32.22447	37.22198	2.55796	15.52141	55.998	64.27059	5.43102	20.33557
GOA	28.25957	31.94160	2.64559	21.67295	47.93237	56.68003	4.91696	34.70050
WOA	26.97430**	30.04258	1.99157	16.36165	42.97897	48.08904	3.07787	23.08921

	Test Problem 5: Middle Size				**Test Problem 6: Middle Size**			
Algorithms	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**
CSSAOS1	55.49790	61.63620	3.83545	10.445	120.7820	138.94506	9.77412	13.42250
CSSAOS2	54.23062	61.06017	3.47999*	9.87299**	127.20868	138.76798	7.95621	13.38620
CSSAOS3	52.00565	60.41638**	3.63744**	14.45141	122.27633	139.73413	9.59079	13.53226
CSSAOS4	52.37256	61.06365	5.05545	14.35272	123.34576	138.55766	7.56643	13.34842
CSSAOS5	52.28594	61.85291	4.25359	9.92305	124.20748	137.22265	6.69154**	12.39329*
CSSAOS6	55.87152	62.14883	3.66387	10.48845	122.38491	140.89066	9.29536	12.97873
CSSAOS7	56.15700	65.29622	4.57686	9.60264*	129.54956	138.87065	6.29750*	14.62203
CSSAOS8	53.83961	62.46868	3.87562	10.14524	128.80211	142.13265	8.91109	12.66591
CSSAOS9	54.30758	61.58231	4.83550	14.36969	122.37850	137.09478**	7.51015	12.53642
CSSAOS10	56.35220	64.56032	4.39567	10.44493	125.42942	140.38920	7.86834	12.51933**
SSA	50.51542	60.79819	4.78161	19.0341	119.11378**	138.49742	12.05787	17.62483
QSSA	64.80552	73.02086	4.78595	21.665	141.30924	168.28082	15.12134	19.06571
OSSA	46.01047*	62.15536	6.65803	21.0956	119.36846	137.21327	12.13792	27.56230
GA	48.49967**	54.22031*	4.11578	22.4497	114.33240*	126.03731*	8.490391	19.99443
DE	70.18680	82.92114	5.18139	17.6365	151.92629	182.23337	16.22329	18.34336
GOA	58.29780	66.16209	5.48442	28.9475	122.05754	149.95388	16.28922	31.98862
WOA	53.23398	63.63002	5.00261	23.6593	128.83738	140.94750	9.218527	19.67783

	Test Problem 7: Big Size				**Test Problem 8: Big Size**			
Algorithms	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**	**Best**	**Mean**	**Std**	**Execution Time (*Sec.*)**
CSSAOS1	289.6288	327.4334	17.1453	23.7038	460.45341	546.41045	39.46089	32.03189
CSSAOS2	283.3795	320.9263	20.0607	21.7803*	449.68285	532.11962	37.05999	32.91690
CSSAOS3	277.2591	319.4264	16.91791	26.1740	483.78683	546.70642	31.09164*	46.61300
CSSAOS4	272.7691	319.2161	16.9036**	30.8315	466.65355	542.48553	44.85309	33.34647
CSSAOS5	281.3526	327.6406	19.4602	22.2118	486.68179	549.76931	40.43495	33.18148
CSSAOS6	272.6986**	326.8937	23.9196	23.6171	467.74606	546.61403	59.55102	28.89418
CSSAOS7	274.9710	325.7727	22.4014	21.8263	460.31118	531.40664**	38.33963	28.69562
CSSAOS8	275.2227	321.2622	18.1463	22.2729	492.62817	549.92323	36.08036	30.52653

(*continued on next page*)

Table 6 (*continued*)

	Test Problem1: Small Size				Test Problem 2: Small Size			
Algorithms	Best	Mean	Std	Execution Time (*Sec.*)	Best	Mean	Std	Execution Time (*Sec.*)
CSSAOS9	291.1419	319.3040	16.5973*	29.0485	574.89917	687.88371	48.67681	29.70536
CSSAOS10	290.7308	324.8677	22.6660	23.1340	471.67457	557.28976	47.22528	29.38603
SSA	281.97	325.9626	25.56847	22.7897	466.984178	544.94682	43.70286	23.29909*
QSSA	345.9554	398.0477	21.54866	26.6569	525.732261	703.28816	60.20151	25.51017
OSSA	281.97	325.9626	25.56847	34.5173	447.957363**	546.29674	41.52890	25.18711
GA	275.3897	310.0742*	24.67966	27.2333	387.696073*	509.31433	40.757151	26.36991
DE	376.901	419.9076	31.0641	23.0523	595.884193	731.56177	71.52201	23.97885**
GOA	281.626	347.7392	27.2258	45.0247	475.404008	565.48033	46.98423	57.19017
WOA	268.5507*	318.0599**	21.4426	21.7839**	467.13434	520.10333*	31.26564**	24.87001

	Test Problem 9: Big Size			
Algorithms	Best	Mean	Std	Execution Time (*Sec.*)
CSSAOS1	534.1640*	688.7093	53.7634	51.4537
CSSAOS2	619.2545	696.3099	38.9450**	36.2628
CSSAOS3	636.9109	692.4627	36.1997*	52.7674
CSSAOS4	619.9953	703.7494	42.0843	53.7443
CSSAOS5	597.4677	688.6009	58.1584	36.7552
CSSAOS6	603.1611	683.1784	40.0977	44.4478
CSSAOS7	593.4231	686.3068	46.9671	37.7280
CSSAOS8	562.5483	697.5292	48.1113	36.9116
CSSAOS9	574.8991	687.8837	48.6768	45.7312
CSSAOS10	574.7606	675.8552	57.5630	39.5215
SSA	601.1139	680.6272	46.10871	32.3635
QSSA	727.0634	860.2973	70.34379	32.2286
OSSA	593.8683	664.5131	39.86693	35.7876
GA	579.0285	643.7635**	41.8699	38.0359
DE	724.1793	910.0161	90.47128	44.3503
GOA	615.4905	700.2674	56.81808	61.297
WOA	548.0802**	640.9678*	50.68391	26.9566*

utilized to calculate the response variations.

$$\frac{S}{N} = -10^{*}\log\left(S\left(Y^{2}/\mathrm{N}\right)\right) \tag{32}$$

where N and Y indicate the number of orthogonal arrays and the response, respectively. Table 5 gives a general view of the level of parameters for the Taguchi method.

Orthogonal arrays are used in the Taguchi method with the aim of studying all factors concurrently. The L9 design is used for all CSSAOS1, CSSAOS2, …, and CSSAOS10 algorithms. As illustrated in Fig. 6, the best

Table 7
Aggregate results of the Wilcoxon Signed-Rank Test.

		CSSAOS1	CSSAOS2	CSSAOS3	CSSAOS4	CSSAOS5	CSSAOS6	CSSAOS7	CSSAOS8	CSSAOS9	CSSAOS10	Total
CSSAOS1	better	–	7	5	8	7	7	5	0	5	6	**50**
	as good as	–	0	0	0	0	0	0	0	0	0	0
	worse	–	2	4	1	2	2	4	9	4	3	31
CSSAOS2	better	2	–	2	5	4	3	3	0	3	4	26
	as good as	0	–	0	0	0	0	0	0	0	0	0
	worse	7	–	7	4	5	6	6	9	6	5	55
CSSAOS3	better	4	7	–	5	5	7	3	1	5	5	42
	as good as	0	0	–	0	0	0	0	0	0	0	0
	worse	5	2	–	4	4	2	6	8	4	4	39
CSSAOS4	better	1	4	4	–	5	5	3	0	3	4	29
	as good as	0	0	0	–		1	0	0	0	0	1
	worse	8	5	5	–	4	3	6	9	6	5	51
CSSAOS5	better	3	5	4	4	–	5	3	0	1	4	29
	as good as	0	0	0	0	–	0	0	0	0	0	0
	worse	6	4	5	5	–	4	6	9	8	5	52
CSSAOS6	better	2	5	2	3	4	–	3	0	3	4	26
	as good as	0	0	0	1	0	–	0	0	0	0	1
	worse	7	4	7	5	5	–	6	9	6	5	54
CSSAOS7	better	4	5	6	6	6	6	–	1	4	4	42
	as good as	0	0	0	0	0	0	–	0	0	0	0
	worse	5	4	3	3	3	3	–	8	5	5	39
CSSAOS8	better	9	9	8	9	9	9	7	–	7	8	**75**
	as good as	0	0	0	0	0	0	0	–	0	0	0
	worse	0	0	1	0	0	0	2	–	2	1	6
CSSAOS9	better	4	6	4	6	7	6	5	1	–	4	43
	as good as	0	0	0	0	0	0	0	0	–	0	0
	worse	5	3	5	3	2	3	4	8	–	5	38
CSSAOS10	better	3	5	4	4	5	5	5	1	5	–	37
	as good as	0	0	0	0	0	0	0	0	0	–	0
	worse	6	4	5	5	4	4	4	8	4	–	44

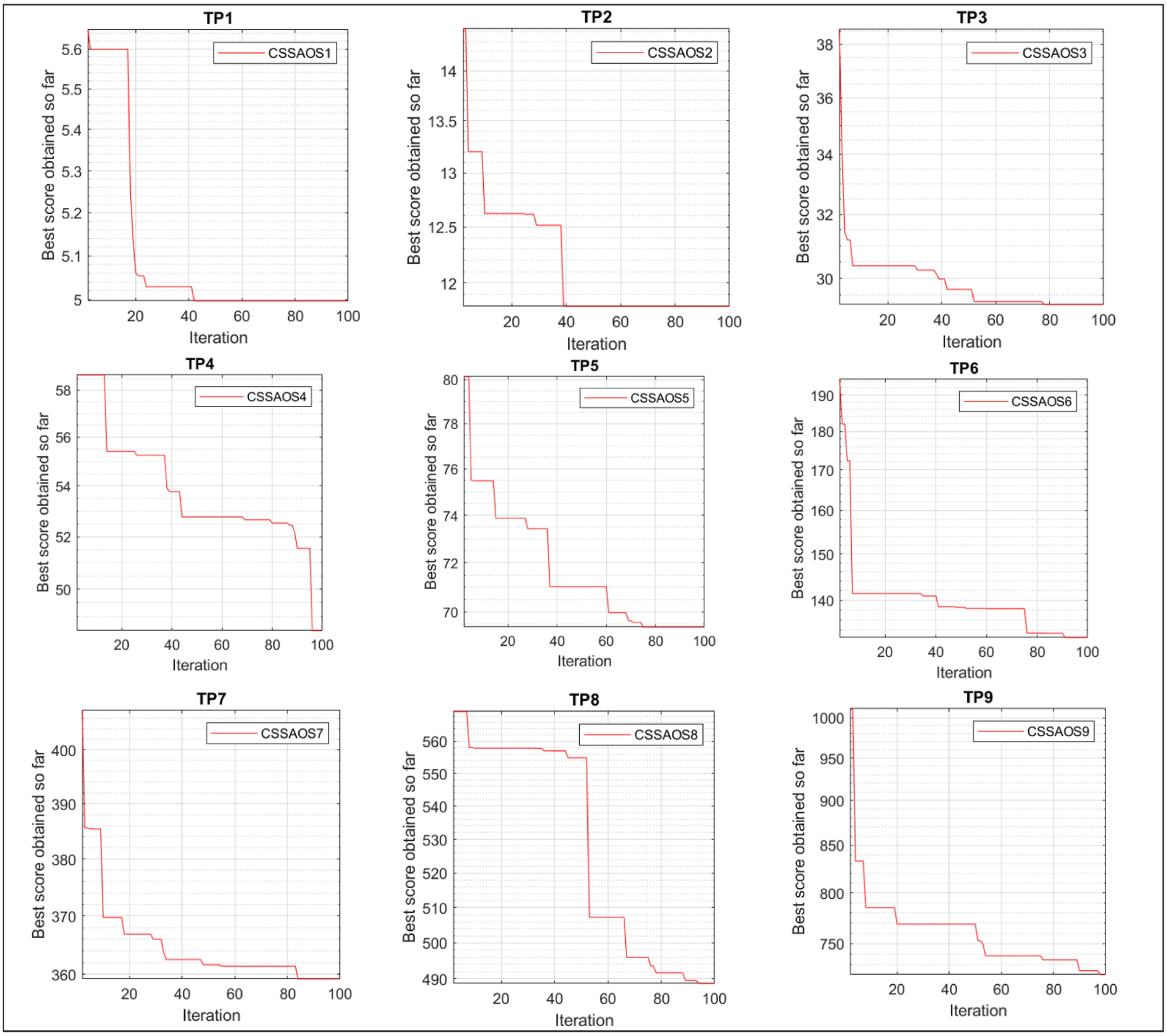

Fig. 7. Convergence curves of CSSAOS in different test problems.

Table 8
P-value obtained from Wilcoxon Signed-Rank Test.

Test Problem	SSA	QSSA	OSSA	GA	DE	GOA
TP 1	0.465	3.90E−07	4.64E−02	294E−02	1.59E−08	2.43E−07
TP 2	6.15E-02	2.61E−10	8.24E−02	0.2642	6.69E−11	1.07E−07
TP 3	0.7283	1.33E−10	0.3555	4.4E−03	3.02E−11	1.43E−05
TP 4	0.1413	2.60E−08	8.56E−04	1.10E−08	6.07E−11	2.32E−06
TP 5	2.25E-04	1.73E−07	5.75E−02	3.82E−10	4.98E−11	0.7172
TP 6	0.5011	8.10E−10	0.2643	2.32E−06	5.49E−11	9.5E−03
TP 7	0.7845	6.07E−11	0.7845	0.0018	3.02E−11	1.8E−03
TP 8	0.7618	2.87E−10	0.9117	7.30E−04	4.08E−11	9.63E−02
TP 9	0.3871	1.61E−10	4.36E−02	1.04E−04	1.09E−10	0.7283

levels are determined for CSSAOS algorithms. According to the results of the Taguchi method, the best level for *MaxIt* is the median level (100 repetitions) and the best level for the number of agents is the lower level, i.e. 30.

6.2. Results

The standard deviation, elapsed time, and mean value of the best solutions of 30 independent runs were employed as the performance metrics. Specifically, standard deviation indicates the stability of CSSAOS during all the runs, and the mean value indicates the expected optimal value between all the independent runs. Elapsed time refers to

Table 9
Result of two scenarios on different test problems.

Algorithms	TP 1		Algorithms	TP 2	
	Scenario 1	Scenario 2		Scenario 1	Scenario 2
CSSAOS1	43.41714	106.54069	CSSAOS1	50.67028	107.09338*
CSSAOS2	43.44108	106.90035	CSSAOS2	50.66570	108.64819
CSSAOS3	43.30039	105.81490*	CSSAOS3	50.38541	108.29135
CSSAOS4	43.49327	107.44247	CSSAOS4	50.58236	109.03749
CSSAOS5	43.34817	105.85339	CSSAOS5	50.34261	107.70121
CSSAOS6	43.29790*	106.04966	CSSAOS6	50.41639	107.96314
CSSAOS7	43.38332	106.64062	CSSAOS7	50.32690	108.96405
CSSAOS8	43.51448	107.95853	CSSAOS8	51.15230	109.16154
CSSAOS9	43.46609	106.64860	CSSAOS9	50.22408*	108.13719
CSSAOS10	43.52413	107.28868	CSSAOS10	50.46925	112.08073
Algorithms	TP 3		Algorithms	TP 4	
	Scenario 1	Scenario 2		Scenario 1	Scenario 2
CSSAOS1	118.99952	119.90934	CSSAOS1	156.55615	163.43364
CSSAOS2	117.78252	118.60266	CSSAOS2	156.45330	165.21944
CSSAOS3	118.05346	120.30361	CSSAOS3	154.74505*	165.84541
CSSAOS4	119.67115	120.56975	CSSAOS4	160.40214	162.37491
CSSAOS5	117.43848	118.39498*	CSSAOS5	156.09889	163.64796
CSSAOS6	117.86025	119.54019	CSSAOS6	155.15784	161.98745
CSSAOS7	120.11830	119.21380	CSSAOS7	155.48179	161.64331*
CSSAOS8	122.13965	123.40388	CSSAOS8	165.45007	167.39219
CSSAOS9	116.69701*	119.37925	CSSAOS9	157.04802	163.25710
CSSAOS10	119.10560	120.74829	CSSAOS10	157.40742	162.95838
Algorithms	TP 5		Algorithms	TP 6	
	Scenario 1	Scenario 2		Scenario 1	Scenario 2
CSSAOS1	169.64204*	187.87966	CSSAOS1	213.08597	262.01385
CSSAOS2	170.43839	187.14951	CSSAOS2	213.34596	261.14971
CSSAOS3	172.11505	191.51691	CSSAOS3	207.90385*	260.07952
CSSAOS4	172.08474	184.53374*	CSSAOS4	210.68178	256.91591*
CSSAOS5	169.58989	188.67022	CSSAOS5	209.39523	264.49379
CSSAOS6	173.22762	189.24345	CSSAOS6	208.50304	260.69338
CSSAOS7	172.79874	187.88798	CSSAOS7	212.97943	263.79808
CSSAOS8	180.49498	192.60615	CSSAOS8	221.37749	276.75575
CSSAOS9	170.61923	188.68390	CSSAOS9	210.93482	261.79015
CSSAOS10	174.05459	191.15471	CSSAOS10	211.90056	263.35296
Algorithms	TP 7		Algorithms	TP 8	
	Scenario 1	Scenario 2		Scenario 1	Scenario 2
CSSAOS1	319.03591*	409.00587	CSSAOS1	684.19235	813.63790
CSSAOS2	323.50533	399.20926*	CSSAOS2	681.44988*	822.10362
CSSAOS3	322.20081	400.82711	CSSAOS3	705.90555	780.94755
CSSAOS4	320.11355	401.05166	CSSAOS4	682.04598	782.22599
CSSAOS5	320.66377	403.82135	CSSAOS5	686.41675	809.66363
CSSAOS6	319.57062	400.46126	CSSAOS6	697.15683	813.06987
CSSAOS7	322.71439	407.09836	CSSAOS7	712.37358	819.84531
CSSAOS8	323.69079	408.92380	CSSAOS8	708.38740	799.46035
CSSAOS9	321.39431	408.27493	CSSAOS9	707.89509	769.66599*
CSSAOS10	321.70233	407.13588	CSSAOS10	685.29395	832.22004
Algorithms	TP 9				
	Scenario 1	Scenario 2			
CSSAOS1	792.42895	1253.81253			
CSSAOS2	778.32621	1260.52903			
CSSAOS3	788.03690	1264.37061			
CSSAOS4	764.58578*	1222.82759			
CSSAOS5	788.11861	1232.90053			
CSSAOS6	776.34324	1232.08028			
CSSAOS7	824.93410	1268.45680			
CSSAOS8	821.77483	1260.93219			
CSSAOS9	802.31085	1189.46387*			
CSSAOS10	800.58965	1223.91314			

Table 10
Medical equipment in each unit.

Department		Medical Equipment	Number of electrical equipment	Demanding Time to use electrical equipment per patient based on ESI (minutes)		
				ESI 1	ESI 2	ESI 3 & ESI 4
Emergency Department	Triage	Monitoring of Vital Signs	1	3	2	1
	CPR	CPR Beds	1	120	82	45
		Monitoring of Vital Signs	1	90	82	45
		Syringe Pump	3	75	50	30
		General Motorized Suction	2	40	30	20
		Electro Shock	2	1.5	1	0.5
		Portable Ventilator	1	80	50	10
		Blood Gas Analyzer	1	7	5	3
	IED	IED Beds	7	360	300	240
		Monitoring of Vital Signs	7	360	300	240
		Syringe Pump	7	360	300	240
		Electrocardiograph	2	5	4	3
		Echocardiograph	1	20	17	15
Cath lab	Angiography	Angiography bed	1	30	20	10
		Angiographic Injector	1	3	2	1
		CARM radiography	1	30	20	10
		Angiographic specialized monitors	6	30	20	10
	Recovery	Recovery Beds	8	420	360	180
		Syringe pump	10	420	360	180
		Monitoring of vital signs	8	420	360	180
	Angioplasty	Angiography bed	1	90	60	15
		Angiographic Injector	1	3	2	1
		CARM radiography	1	90	60	15
		Angiographic specialized monitors	3	90	60	15
ICU	ICU I	ICU Beds	8	7200	5760	4320
		Portable Ventilator	8	2880	2160	1440
		Motorized Suction Surgery	4	1440	1330	1220
		Monitoring of Vital Signs	8	7200	5760	4320
		Syringe pump	10	7200	5760	4320
		Mattress with air pump	8	7200	5760	4320
		Multichannel electrocardiograph	1	5	4	3
		Electro Shock	2	1.5	1	0.5
		Blood Gas Analyzer	1	7	5	3
		Echocardiograph	3	20	17	15
		Warm touch	8	7200	5760	4320
		Electrocardiograph	1	5	4	3
	ICU II	ICU Beds	5	4320	2880	1440
		Portable Ventilator	5	4320	2880	1440
		Motorized Suction Surgery	3	20	10	5
		Monitoring of Vital Signs	5	4320	2880	1440
		Syringe pump	7	4320	2880	1440
		Mattress with air pump	5	4320	2880	1440
		Multichannel electrocardiograph	1	5	4	3
		Electro Shock	2	1.5	1	0.5
		Balloon Pump*	2	4320	2880	1440
		Blood Gas Analyzer	1	7	5	3
		Electrocardiograph	1	5	4	3
		Echocardiograph	1	20	17	15
CCU	CCU I	CCU Beds	12	5760	5040	4320
		Monitoring of vital signs	12	5760	5040	4320
		Portable Ventilator	1	1440	1330	1220
		Flowmeter O2	12	1440	1330	1220
		Multichannel electrocardiograph	1	5	4	3
		Syringe pump	12	5760	5040	4320
		Motorized Suction Surgery	4	3	2	1
		Electrocardiograph	1	5	4	3
		Echocardiograph	1	20	17	15
	CCU II	CCU Beds	4	2880	2160	1440
		Monitoring of vital signs	4	2880	2160	1440
		Portable Ventilator	2	1440	1330	1220
		Flowmeter O2	2	1440	1330	1220
		Multichannel electrocardiograph	1	5	4	3
		Syringe pump	4	2880	2160	1440
		Motorized Suction Surgery	1	3	2	1
		Mattress with air pump	2	2880	2160	1440
		Electrocardiograph	1	5	4	3
		Echocardiograph	1	20	17	15
PCCU		PCCU Beds	14	4320	2880	1440
		Syringe pump	14	4320	2880	1440
		Motorized Suction Surgery*	2	20	15	10
		Monitoring of vital signs	14	4320	2880	1440
		Multichannel electrocardiograph	1	5	4	3

(*continued on next page*)

Table 10 (*continued*)

Department	Medical Equipment	Number of electrical equipment	Demanding Time to use electrical equipment per patient based on ESI (minutes)		
			ESI 1	ESI 2	ESI 3 & ESI 4
	Echocardiograph	1	20	17	15
OR	Hydraulic operating room bed	2	360	300	240
	Autoclaves	1	135	100	45
	Electric Sternum Saw	2	4	3	2
	Monitoring of Vital Signs	3	360	300	240
	Medical monitor	4	360	300	240
	Anesthesia Machine	2	360	300	240
	Syringe Pump	10	360	300	240
	Motorized Suction Surgery	4	40	30	20
	Blood Shaker	1	40	30	20
	Two-cavity ceiling flashing light	2	360	300	240
	Cerebral cortex	1	360	300	240
	Automatic coagulation timer	3	15	10	5
	Trans esophageal Echocardiogram (TEE)	1	120	60	30
	Electro Counter	1	60	40	30
	Heart-lung machine	2	180	150	130
	Blood warmer	1	20	15	10
	Salt Set	1	40	30	20
	laryngoscope	2	3	2	1
Reception	Portable Monitor	1	15	10	5
Discharge	Portable Monitor	1	20	15	10

the average total time (in seconds) that each algorithm needs to run in order to determine the computation cost of each algorithm. For providing a fair comparison, the main controlling parameters of these algorithms, i.e. maximum iteration and the number of search agents, were set equal to 100 and 30, respectively.

The average number of patients admitted to the emergency department of BHH between August 2018 and August 2019 was 6996, of which 10 % were categorized into ESI1 (3 % for route 11, 4 % for route 12, and 3 % for route 13) and another 15 % into ESI2 (2 % for route 21, 5 % for route 22, and 8 % for route 23). For the others, 20 % went to ESI3 (15 % for route 31 and 5 % for route 32), 25 % to ESI4, and 30 % to ESI5.

To examine the applicability of the method, nine different test problems grouped as small, medium, and large-size problems were considered. Each test problem exemplifies a different level of complexity, different planning horizon, different number of patients, and different percentages of patients with different ESIs. The number of beds and electrical equipment and their electricity consumption were also considered, which is provided in Table 10 and Table 11 (in Appendix).

The comparison results in Table 6 verify that CSSAOS performs slightly better than the other well-known metaheuristic algorithms. The * and ** symbols represent the smallest and the second smallest value in each column, respectively.

The Wilcoxon Signed-Rank test is used in order to perform a pairwise comparison and the comparison are made between CSSAOS1, CSSAOS2, …, and CSSAOS10 algorithms using rank-sum values. The result of the pairwise comparison is given in Table 7. In summary, from the results of 9 test problems, it could be easily concluded that CSSAOS8 is very efficient and produces very competitive results based on quality and reliability criteria. In fact, CSSAOS8 strongly competes with the current state-of-the-art algorithms.

The convergence curves of different CSSAOS algorithms on different test problems TP1,…, TP9 (small, medium, and big sizes) are shown in Fig. 7. The behavior of the average fitness of all agents demonstrates the capability of CSSAOS in achieving very good results during the solution process.

The hypothesis test is performed to show whether the mean values of the superior algorithms are significantly different. The Wilcoxon Signed-Rank Test (James & Li, 2015) reveals the performance of the proposed algorithm with the other well-known algorithms. To this aim, the null hypothesis (H_0) and the alternative hypothesis (H_1) can be used to determine the significant level of rejecting the null hypothesis, which is 0.01:

$$\begin{cases} H_0 : \mu_1 = \mu_2 \\ H_1 : \mu_1 \neq \mu_2 \end{cases}$$

According to Table 8, H_0 is rejected at the 99 % significance level, while the acceptance of H_1 implies that the obtained optimal values of our proposed algorithm are distinctive from those of the other algorithms.

6.3. *Sensitivity analysis*

Considering the growing trend of cardiovascular diseases worldwide (such as arrhythmias, aorta disease, congenital heart disease, coronary artery, heart attack, heart failure, and cardiomyopathy), it is pertinent to study the effect of the increasing number of high-risk patients (ESI1, ESI2) on the amount of CO_2 emissions in the hospital. For this, two scenarios are defined. In the first scenario, the percentage of patients in ESI1, ESI2, ESI3, ESI4, and ESI5 are 12 %, 17 %, 18 %, 23 %, and 30 %, respectively. In the second scenario, the percentages are 15 %, 20 %, 15 %, 20 %, and 30 %, accordingly. The results of 30 independent runs are reported in Table 9.

As can be seen in Table 9, for each test problem and each scenario, there is an algorithm that performs superior to the other algorithms. Considering the increase in high-risk patients due to the COVID-19 pandemic, two scenarios have been designed. In the second scenario, the percentage of high-risk patients is higher than in the first scenario. According to Fig. 8, as the number of patients in a serious ESI category increases, the amount of CO_2 emitted grows exponentially. Although,

Table 11
Electricity Consumption (KW) of medical equipment in the BHH.

Department		Medical Equipment	Electricity Consumption (KW)
Emergency Department	Triage	Monitoring of Vital Signs	0.14
		Monitoring of Vital Signs	0.14
		Syringe Pump	0.03
		General Motorized Suction	0.15
		Electro Shock	0.22
		Portable Ventilator	0.53
		Blood Gas Analyzer	0.02
		Monitoring of Vital Signs	0.14
		Syringe Pump	0.03
		Electrocardiograph	0.14
		Echocardiograph	1
Cath lab	Angiography	Angiographic Injector	0.26
		CARM radiography	0.50
		Angiographic specialized monitors	0.24
	Recovery	Syringe pump	0.03
		Monitoring of vital signs	0.14
	Angioplasty	Angiographic Injector	0.26
		CARM radiography	0.50
		Angiographic specialized monitors	0.24
ICU	ICU I	Portable Ventilator	0.53
		Motorized Suction Surgery	0.15
		Monitoring of Vital Signs	0.14
		Syringe pump	0.03
		Mattress with air pump	0.02
		Multichannel electrocardiograph	0.02
		Electro Shock	0.22
		Blood Gas Analyzer	0.02
		Echocardiograph	1
		Warm touch	1.2
		Electrocardiograph	0.14
	ICU II	Portable Ventilator	0.53
		Motorized Suction Surgery	0.15
		Monitoring of Vital Signs	0.14
		Syringe pump	0.03
		Mattress with air pump	0.02
		Multichannel electrocardiograph	0.02
		Electro Shock	0.22
		Blood Gas Analyzer	0.02
		Electrocardiograph	0.14
		Echocardiograph	1
		Carotid Sono	1
CCU	CCU I	Monitoring of vital signs	0.14
		Portable Ventilator	0.53
		Flowmeter O2	2.20
		Multichannel electrocardiograph	0.14
		Syringe pump	0.03
		Motorized Suction Surgery	0.15
		Pacemaker	0.01
		Electro Shock	0.22
		Mattress with air pump	0.02
		Electrocardiograph	0.14
		Echocardiograph	1
	CCU II	Monitoring of vital signs	0.14
		Portable Ventilator	0.53
		Flowmeter O2	13.2
		Multichannel electrocardiograph	0.02
		Syringe pump	0.03
		Motorized Suction Surgery	0.15
		Pacemaker	0.01
		Electro Shock	0.22
		Mattress with air pump	0.02
		Electrocardiograph	0.14
		Echocardiograph	1
PCCU		Syringe pump	0.03
		Motorized Suction Surgery	0.15
		Monitoring of vital signs	0.14
		Multichannel electrocardiograph	0.03
		Electro shock	0.22
		Mattress with air pump	0.02
		Echocardiograph	1
OR		Autoclaves	1
		Electric Sternum Saw	1
		Monitoring of Vital Signs	0.14
		Medical monitor	0.22
		Anesthesia Machine	0.02
		Syringe Pump	0.03
		Motorized Suction Surgery	0.15
		Blood Shaker	0.02
		Two-cavity ceiling flashing light	0.17
		Cerebral cortex	0.12
		Automatic coagulation timer	6.50
		Trans esophageal Echocardiogram (TEE)	1
		Electro Counter	0.85
		Heart-lung machine	0.22
		Blood warmer	0.28
		Salt Set	3.52
		laryngoscope	2.20
Reception/ Discharge		Portable Monitor	0.21

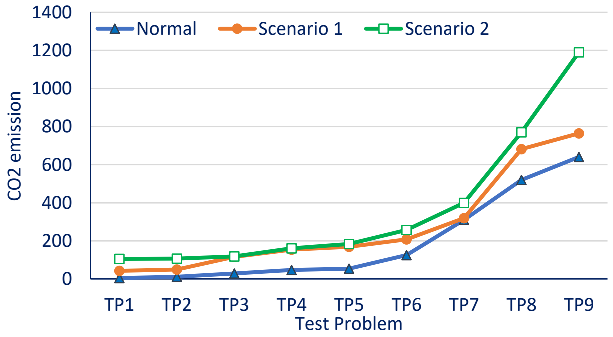

Fig. 8. Comparative results on different test problems.

the rate of growth in scenario 2 is sharper than in scenario 1 and the normal situation.

6.4. *Managerial insights*

This study provided the hospital with managerial insights for better patient flow and the environmental impact of the treatment process. The issue with the current operating conditions and the flow of the patient system we identified during the course of the study led to a meeting with hospital authorities to highlight the hospital's responsibility in complying with low-carbon healthcare. Our optimization results confirmed that patient waiting times would be reduced if the proposed algorithm was used. This would assist the hospital to achieve a better service level with less waiting time and length of stay. Furthermore, hospital administrators agreed that the research results could be applied, as they addressed both operational and environmental concerns. They highlighted the importance of implementing an integrated web-based scheduling system across the hospital to help the hospital offer timely services to patients while respecting its environmental commitment.

7. Conclusion

The complexity and cost of healthcare systems, especially in hospitals, are becoming increasingly concerning due to the emergence of complex equipment and technologies. In addition, numerous treatments consume large amounts of electricity and, therefore, indirectly increase CO_2 emissions(MacNeill et al., 2017). Therefore, this research proposes an NP-hard carbon-efficient flexible job shop scheduling problem and a metaheuristic optimization algorithm called CSSAOS. The algorithm integrates SSA with chaotic maps to update the position of followers, the sine cosine algorithm to update the leader position, and opposition-based learning for a better exploration of the search space, generating more accurate solutions. Based on a real-world case study, the results indicate that the proposed CSSAOS algorithm exhibited better performance to solve problems with a complex search space compared to DE, GA, GOA, OSSA, QSSA, SSA, and WOA.

We have applied the CSSAOS algorithm to solve a green patient flow problem. This research could be extended in two directions. From the problem viewpoint, more details, such as resource management and staff rostering, could be added to the patient flow. Also, energy management and policies, such as replacing equipment with different technology, could be considered. From a solution viewpoint, one can compare the proposed algorithm and its variants with other multi-objective metaheuristics such as NSGA-II and MOPSO. One could also consider extending the proposed CSSAOS algorithm to a multi-objective algorithm.

CRediT authorship contribution statement

Masoumeh Vali: Conceptualization, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Khodakaram Salimifard:** Conceptualization, Formal analysis, Investigation, Writing – original draft, Supervision. **Amir H. Gandomi:** Writing – review & editing. **Thierry J. Chaussalet:** Formal analysis, Investigation, Writing – review & editing.

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.

Acknowledgement

None.

Appendix

See Fig. 10, Table 10, Table 11.

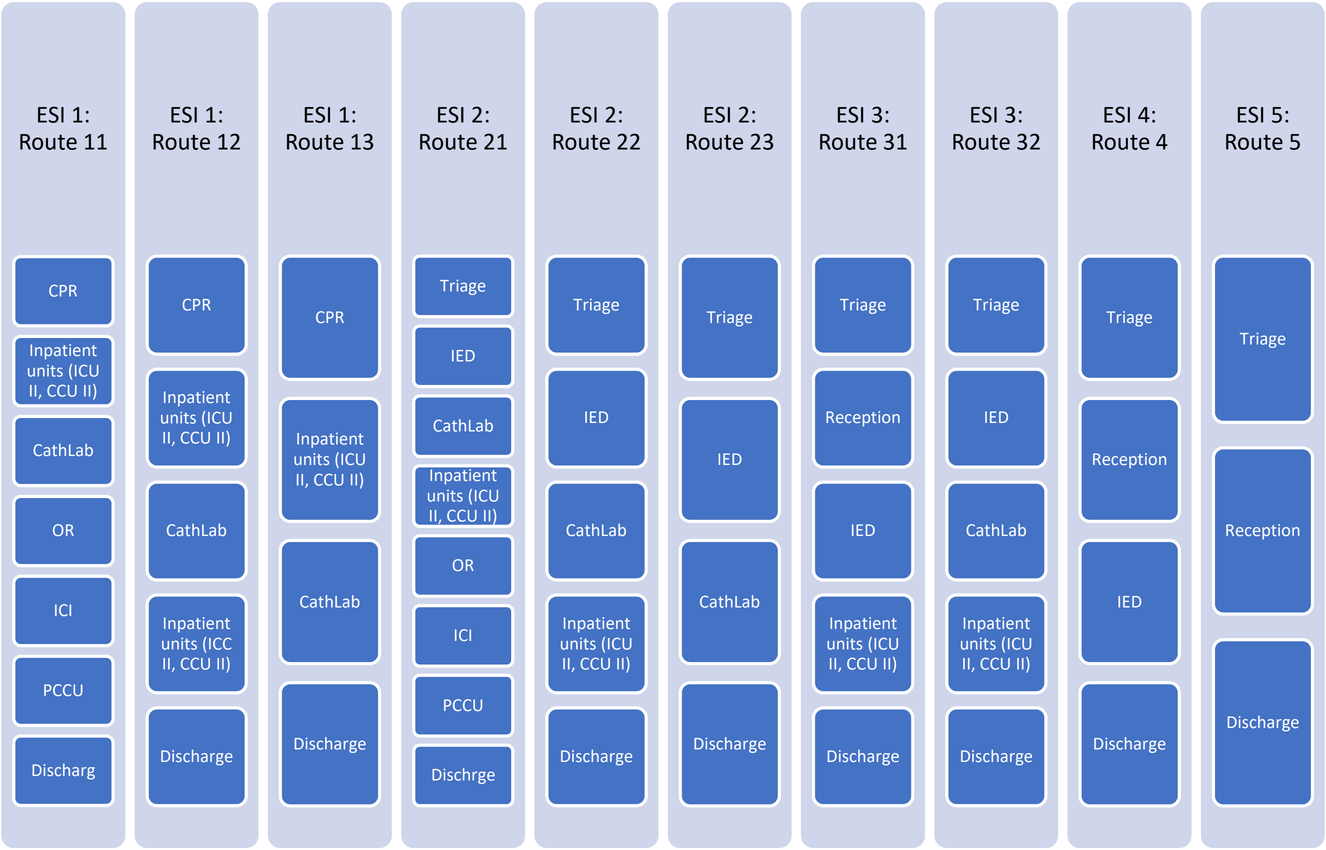

Fig. 10. Patient flow routes based on his/her ESI.

References

Anderson, P., & Bone, Q. (1980). Communication between individuals in salp chains. II. Physiology. *Proceedings of the Royal Society of London. Series B. Biological Sciences, 210*(1181), 559–574.

Bacelar-Silva, G. M., Cox, J. F., III, & Rodrigues, P. P. (2022). Outcomes of managing healthcare services using the Theory of Constraints: A systematic review. *Health Systems, 11*(1), 1–16.

Bairathi, D., & Gopalani, D. (2018). Opposition based salp swarm algorithm for numerical optimization. *Paper presented at the International Conference on Intelligent Systems Design and Applications*.

Becker (Producer). (2012, 9 6). 18 Statistics on Hospital Energy Consumption. *BECKER'S HOSPITAL REVIEW*. Retrieved from https://www.beckershospitalreview.com/strategic-planning/18-statistics-on-hospital-energy-consumption.html.

Belboom, S., Renzoni, R., Verjans, B., Léonard, A., & Germain, A. (2011). A life cycle assessment of injectable drug primary packaging: Comparing the traditional process in glass vials with the closed vial technology (polymer vials). *The International Journal of Life Cycle Assessment, 16*(2), 159–167.

Bi, P., & Hansen, A. (2018). Carbon emissions and public health: An inverse association? *The Lancet Planetary Health, 2*(1), e8–e9.

Brown, L. H., Buettner, P. G., & Canyon, D. V. (2012). The energy burden and environmental impact of health services. *American Journal of Public Health, 102*(12), e76–e82.

Bujak, J. (2010). Heat consumption for preparing domestic hot water in hospitals. *Energy and Buildings, 42*(7), 1047–1055.

Burdett, R. L., & Kozan, E. (2018). An integrated approach for scheduling health care activities in a hospital. *European Journal of Operational Research, 264*(2), 756–773.

Campion, N., Thiel, C. L., DeBlois, J., Woods, N. C., Landis, A. E., & Bilec, M. M. (2012). Life cycle assessment perspectives on delivering an infant in the US. *Science of the Total Environment, 425*, 191–198.

Caponetto, R., Fortuna, L., Fazzino, S., & Xibilia, M. G. (2003). Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation, 7*(3), 289–304.

CBECS (Producer). (2012, 8 17). Energy Characteristics and Energy Consumed in Large Hospital Buildings in the United States in 2007 *eia*. Retrieved from https://www.eia.gov/consumption/commercial/reports/2007/large-hospital.php.

Chakraborty, U. K. (2008). *Advances in differential evolution* (Vol. 143). Springer.

Chen, R., Dong, C., Ye, Y., Chen, Z., & Liu, Y. (2019). QSSA: Quantum Evolutionary Salp Swarm Algorithm for Mechanical Design. *IEEE Access, 7*, 145582–145595.

Chirarattananon, S., Chaiwiwatworakul, P., Hien, V. D., Rakkwamsuk, P., & Kubaha, K. (2010). Assessment of energy savings from the revised building energy code of Thailand. *Energy, 35*(4), 1741–1753.

Chung, J. W., & Meltzer, D. O. (2009). Estimate of the carbon footprint of the US health care sector. *JAMA, 302*(18), 1970–1972.

Connor, A., Lillywhite, R., & Cooke, M. (2010). The carbon footprint of a renal service in the United Kingdom. *QJM: International Journal of Medicine, 103*(12), 965–975.

Danesh-Meyer, H. V. (2011). Neuroprotection in glaucoma: Recent and future directions. *Current Opinion in Ophthalmology, 22*(2), 78–86.

Eckelman, M., Mosher, M., Gonzalez, A., & Sherman, J. (2012). Comparative life cycle assessment of disposable and reusable laryngeal mask airways. *Anesthesia & Analgesia, 114*(5), 1067–1072.

Eckelman, M. J., Sherman, J. D., & MacNeill, A. J. J. P. M. (2018). Life cycle environmental emissions and health damages from the Canadian healthcare system: An economic-environmental-epidemiological analysis. *PLoS Medicine, 15*(7), e1002623.

Erhard, M., Schoenfelder, J., Fügener, A., & Brunner, J. O. (2018). State of the art in physician scheduling. *European Journal of Operational Research, 265*(1), 1–18.

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.

Fattahi, P., Hajipour, V., & Nobari, A. (2015). A bi-objective continuous review inventory control model: Pareto-based meta-heuristic algorithms. *Applied Soft Computing, 32*, 211–223.

Gaglia, A. G., Balaras, C. A., Mirasgedis, S., Georgopoulou, E., Sarafidis, Y., & Lalas, D. P. (2007). Empirical assessment of the Hellenic non-residential building stock, energy consumption, emissions and potential energy savings. *Energy Conversion and Management, 48*(4), 1160–1175.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research, 1*(2), 117–129.

Gilliam, A., Davidson, B., & Guest, J. (2008). The carbon footprint of laparoscopic surgery: Should we offset? *Surgical Endoscopy, 22*(2), 573.

Gupta, D., & Denton, B. (2008). Appointment scheduling in health care: Challenges and opportunities. *IIE Transactions, 40*(9), 800–819.

Hall, R. W. (2012). *Handbook of healthcare system scheduling*. Springer.

Jain, A. S., & Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research, 113*(2), 390–434.

James, J., & Li, V. O. (2015). A social spider algorithm for global optimization. *Applied Soft Computing, 30*, 614–627.

Jarosław, P., Czesław, S., & Dominik, Ż. (2013). Optimizing bicriteria flow shop scheduling problem by simulated annealing algorithm. *Procedia Computer Science, 18*, 936–945.

Ji, Z., Kang, C., Chen, Q., Xia, Q., Jiang, C., Chen, Z., & Xin, J. (2013). Low-carbon power system dispatch incorporating carbon capture power plants. *IEEE Transactions on Power Systems, 28*(4), 4615–4623.

Leviner, S. (2020). Patient flow within hospitals: A conceptual model. *Nursing Science Quarterly, 33*(1), 29–34.

Li, J., & Kwan, R. S. (2004). A meta-heuristic with orthogonal experiment for the set covering problem. *Journal of Mathematical Modelling and Algorithms, 3*(3), 263–283.

Liang, B., Turkcan, A., Ceyhan, M. E., & Stuart, K. (2015). Improvement of chemotherapy patient flow and scheduling in an outpatient oncology clinic. *International Journal of Production Research, 53*(24), 7177–7190.

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.

MacNeill, A. J., Lillywhite, R., & Brown, C. J. (2017). The impact of surgery on global climate: A carbon footprinting study of operating theatres in three health systems. *The Lancet Planetary Health, 1*(9), e381–e388.

McAlister, S., Ou, Y., Neff, E., Hapgood, K., Story, D., Mealey, P., & McGain, F. (2016). The Environmental footprint of morphine: A life cycle assessment from opium poppy farming to the packaged drug. *BMJ Open, 6*(10), Article e013302.

McGain, F., & Naylor, C. (2014). Environmental sustainability in hospitals–a systematic review and research agenda. *Journal of Health Services Research & Policy, 19*(4), 245–252.

McMichael, A. J., Neira, M., Bertollini, R., Campbell-Lendrum, D., & Hales, S. (2009). Climate change: A time of need and opportunity for the health sector. *The Lancet, 374*(9707), 2123–2125.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133.

Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks* (pp. 43–55). Springer.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Modi, J. A. (2007). Hall, Randolph W., Ed. 2006. Patient Flow: Reducing Delay in Healthcare Delivery. *Interfaces, 37*(5), 490–492.

Pan, W.-T. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems, 26*, 69–74.

Parvatker, A. G., Tunceroglu, H., Sherman, J. D., Coish, P., Anastas, P., Zimmerman, J. B., & Eckelman, M. J. (2019). Cradle-to-gate greenhouse gas emissions for twenty anesthetic active pharmaceutical ingredients based on process scale-up and process design calculations. *ACS Sustainable Chemistry & Engineering, 7*(7), 6580–6591.

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

Pham, D.-N., & Klinkert, A. (2008). Surgical case scheduling as a generalized job shop scheduling problem. *European Journal of Operational Research, 185*(3), 1011–1025.

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.

Pollard, A., Paddle, J., Taylor, T., & Tillyard, A. (2014). The carbon footprint of acute care: How energy intensive is critical care? *Public Health, 128*(9), 771–776.

Renedo, C., Ortiz, A., Mañana, M., Silio, D., & Perez, S. (2006). Study of different cogeneration alternatives for a Spanish hospital center. *Energy and Buildings, 38*(5), 484–490.

Review, H. (Producer). (2016, June 14). Hospital Review. *Becker's Healthcare*. Retrieved from https://www.beckershospitalreview.com/population-health/us-healthcare-system-is-a-top-producer-of-greenhouse-gas-emissions.html.

Rossati, A. (2017). Global warming and its health impact. *The International Journal of Occupational and Environmental Medicine, 8*(1), 7.

Ryan, S. M., & Nielsen, C. J. (2010). Global warming potential of inhaled anesthetics: Application to clinical use. *Anesthesia & Analgesia, 111*(1), 92–98.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47.

Sutherland, K. R., & Weihs, D. (2017). Hydrodynamic advantages of swimming by salp chains. *Journal of The Royal Society Interface, 14*(133), 20170298.

Tai, G., & Williams, P. (2012). Optimization of scheduling patient appointments in clinics using a novel modelling technique of patient arrival. *Computer Methods and Programs in Biomedicine, 108*(2), 467–476.

Tanabe, P., Gimbel, R., Yarnold, P. R., & Adams, J. G. J. J. o. E. N. (2004). The Emergency Severity Index (version 3) 5-level triage system scores predict ED resource consumption. *30*(1), 22-29.

Thiel, C. L., Eckelman, M., Guido, R., Huddleston, M., Landis, A. E., Sherman, J., … Bilec, M. M. (2015). Environmental impacts of surgical procedures: Life cycle assessment of hysterectomy in the United States. *Environmental Science & Technology, 49*(3), 1779–1786.

Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. *Paper presented at the International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*.

UNEP. (2012). *The Emissions Gap Report 2012*. Retrieved from https://climateanalytics.org/media/2012gapreport.pdf.

Usubharatana, P., & Phungrassami, H. (2018). Carbon footprints of rubber products supply chains (fresh latex to rubber glove). *Applied Ecology and Environmental Research, 16*(2), 1639–1657.

Verma, A., & Kaushal, S. (2017). A hybrid multi-objective particle swarm optimization for scientific workflow scheduling. *Parallel Computing, 62*, 1–19.

Wang, X., Ding, H., Qiu, M., & Dong, J. (2011). A low-carbon production scheduling system considering renewable energy. *Paper presented at the Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*.

Watts, N., Adger, W. N., Agnolucci, P., Blackstock, J., Byass, P., Cai, W., … Cooper, A. (2015). Health and climate change: Policy responses to protect public health. *The Lancet, 386*(10006), 1861–1914.

Watts, N., Adger, W. N., Ayeb-Karlsson, S., Bai, Y., Byass, P., Campbell-Lendrum, D., … Depledge, M. (2017). The Lancet Countdown: Tracking progress on health and climate change. *The Lancet, 389*(10074), 1151–1164.

Weisz, U., Pichler, P.-P., Jaccard, I. S., Haas, W., Matej, S., Bachner, F., … Weisz, H. (2020). Carbon emission trends and sustainability options in Austrian health care. *Resources, Conservation and Recycling, 160*, Article 104862.

Wernet, G., Conradt, S., Isenring, H. P., Jiménez-González, C., & Hungerbühler, K. (2010). Life cycle assessment of fine chemical production: A case study of pharmaceutical synthesis. *The International Journal of Life Cycle Assessment, 15*(3), 294–303.

White, D. L., Froehle, C. M., & Klassen, K. J. (2011). The effect of integrated scheduling and capacity policies on clinical efficiency. *Production and Operations Management, 20*(3), 442–455.

Wojtys, E. M., Schley, L., Overgaard, K. A., & Agbabian, J. (2009). Applying lean techniques to improve the patient scheduling process. *Journal for Healthcare Quality, 31*(3), 10–16.

Wright, P. D., & Mahar, S. (2013). Centralized nurse scheduling to simultaneously improve schedule cost and nurse satisfaction. *Omega, 41*(6), 1042–1052.

Xiao, Y., & Konak, A. (2017). A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem. *Journal of Cleaner Production, 167*, 1450–1463.

Yi, Q., Li, C., Tang, Y., & Wang, Q. (2012). A new operational framework to job shop scheduling for reducing carbon emissions. *Paper presented at the 2012 IEEE International Conference on Automation Science and Engineering (CASE)*.

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.

Zheng, X.-L., Wang, L., & Wang, S.-Y. (2014). A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. *Knowledge-Based Systems, 57*, 95–103.