

Article

A Modified Coronavirus Herd Immunity Optimizer for the Power Scheduling Problem

Sharif Naser Makhadmeh ¹, Mohammed Azmi Al-Betar ^{1,2}, Mohammed A. Awadallah ^{3,4}, Ammar Kamal Abasi ⁵, Zaid Abdi Alkareem Alyasseri ⁶, Iyad Abu Doush ^{7,8}, Osama Ahmad Alomari ⁹, Robertas Damaševičius ^{10,*}, Audrius Zajančauskas ¹⁰ and Mazin Abed Mohammed ¹¹

- ¹ Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman 346, United Arab Emirates; s.makhadmeh@ajman.ac.ae (S.N.M.); m.albetar@ajman.ac.ae or mohbetar@bau.edu.jo (M.A.A.-B.)
 - ² Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Irbid 21110, Jordan
 - ³ Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza P860, Palestine; ma.awadallah@alaqsa.edu.ps
 - ⁴ Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates
 - ⁵ School of Computer Sciences, Universiti Sains Malaysia, Gelugor 11800, Malaysia; ammar_abasi@student.usm.my
 - ⁶ Information Technology Research and Development Center (ITRDC), University of Kufa, Kufa 54001, Iraq; zaid.alyasseri@uokufa.edu.iq
 - ⁷ Computing Department, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya 20002, Kuwait; idoush@auk.edu.kw
 - ⁸ Computer Science Department, Yarmouk University, Irbid 21163, Jordan
 - ⁹ MLALP Research Group, University of Sharjah, Sharjah 346, United Arab Emirates; oalomari@sharjah.ac.ae
 - ¹⁰ Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania; audrius.zajanckauskas@vdu.lt
 - ¹¹ College of Computer Science and Information Technology, University of Anbar, Anbar 31001, Iraq; mazinalshujeary@uoanbar.edu.iq
- * Correspondence: robertas.damasevicius@vdu.lt



Citation: Makhadmeh, S.N.; Al-Betar, M.A.; Awadallah, M.A.; Abasi, A.K.; Alyasseri, Z.A.A.; Doush, I.A.; Alomari, O.A.; Damaševičius, R.; Zajančauskas, A.; Mohammed, M.A.

A Modified Coronavirus Herd Immunity Optimizer for the Power Scheduling Problem. *Mathematics* **2022**, *10*, 315. <https://doi.org/10.3390/math10030315>

Academic Editors: Sambor Guze and Enrico Zio

Received: 17 November 2021

Accepted: 11 January 2022

Published: 20 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: The Coronavirus herd immunity optimizer (CHIO) is a new human-based optimization algorithm that imitates the herd immunity strategy to eliminate of the COVID-19 disease. In this paper, the coronavirus herd immunity optimizer (CHIO) is modified to tackle a discrete power scheduling problem in a smart home (PSPSH). PSPSH is a combinatorial optimization problem with NP-hard features. It is a highly constrained discrete scheduling problem concerned with assigning the operation time for smart home appliances based on a dynamic pricing scheme(s) and several other constraints. The primary objective when solving PSPSH is to maintain the stability of the power system by reducing the ratio between average and highest power demand (peak-to-average ratio (PAR)) and reducing electricity bill (EB) with considering the comfort level of users (UC). This paper modifies and adapts the CHIO algorithm to deal with such discrete optimization problems, particularly PSPSH. The adaptation and modification include embedding PSPSH problem-specific operators to CHIO operations to meet the discrete search space requirements. PSPSH is modeled as a multi-objective problem considering all objectives, including PAR, EB, and UC. The proposed method is examined using a dataset that contains 36 home appliances and seven consumption scenarios. The main CHIO parameters are tuned to find their best values. These best values are used to evaluate the proposed method by comparing its results with comparative five metaheuristic algorithms. The proposed method shows encouraging results and almost obtains the best results in all consumption scenarios.

Keywords: discrete coronavirus herd immunity optimizer; power scheduling problem in smart home; multi-criteria optimisation; smart home; multi-objective optimisation problem

1. Introduction

The traditional grids cannot fulfill the rapid growth of users' power demand because of their primitive equipment and distribution systems, which can lead to blackouts in residential areas. This is because of the gap between power production and power demand, particularly in peak periods. Therefore, an alternative approach on the basis of smart technologies is proposed to address these issues, called the smart grid [1,2].

Smart grids are an improved generation of the traditional grids, mainly constructed to enhance communication, control, distribution, and delivery systems. These features improve the interaction between power supply companies and users by sending users the power and receiving feedback. The users' feedback allows supply companies to predict the power consumption of upcoming periods and produce enough power. Using smart grid technologies the power consumed at some periods, such as peak periods, is still high, which obliges power supply companies to operate more power plants to address this issue and maintain power systems efficiently. Accordingly, the cost of producing power and electricity tariffs will be increased which increases the electricity bill (EB) for users [1,3].

Power supply companies propose a new approach based on dynamic electricity tariffs to dispraise the power consumed and to reduce it at peak periods. This new approach is proposed to motivate users to reschedule smart home appliances' operation time to be operated at off-peak periods. Such an approach is called a dynamic pricing scheme. The dynamic pricing schemes generate dynamic electricity tariffs in which the tariffs are high and low during peak and off-peak periods, respectively. The most common dynamic pricing schemes are block rate (IBR), real-time price (RTP), time-of-use price, and critical period price [1,4,5].

The problem of rescheduling the appliances' operation time based on a dynamic pricing scheme(s) is called the power scheduling problem in the smart home (PSPSH). The main benefits that can be obtained by power supply companies and users when solving PSPSH are maintaining the stability of the power system. This is achieved by reducing the ratio between average and highest power demand which is called peak-to-average ratio (PAR), and reducing EB while considering the user comfort (UC) level [1].

The PSPSH is modeled as an optimization problem to optimally achieve its objectives, including reducing EB and PAR and improving UC level. The formulation of PSPSH is proposed in two forms on the basis of the objective function: single objective and multi-objective functions [2,6]. The single objective formulation is proposed to reduce only EB by ignoring the other objectives, whereas the multi-objective formulation considers reducing EB and increasing UC. However, the PAR's effect was not seriously considered in addressing PSPSH and the optimization processes, where only a few studies recognized it in the problem formulation [7–9].

Several optimization algorithms are adapted to address PSPSH optimally which are classified into exact, heuristic, and metaheuristic algorithms [1,10]. Metaheuristic algorithms are the most popular class due to their high performance in exploring search spaces and searching deeply to find optimal/near-optimal solution(s). Metaheuristic algorithms proved their efficiency in addressing optimization problems in different fields, such as software defect prediction [11], signal denoising [12–14], feature selection [15], production forecasting [16], brain–computer interface [17,18], human activity recognition [19], wind speed forecasting [20], document clustering [21–26], machine scheduling [27], heat transfer optimization [28], cybersecurity [29], topic extraction [30–32], battery useful life prediction [33], mobile network routing [34], biomedical image segmentation [35,36], gene selection [37], and Others [38].

Recently, a large number of metaheuristic algorithms inspired by viruses' behavior in nature are proposed. These algorithms are corona virus optimization [39], coronavirus optimization algorithm [40], virus spread optimization [41], virus colony search [42], and coronavirus herd immunity optimizer (CHIO) [43]. These algorithms presented a good performance in addressing optimization problems, where CHIO is the most successful due

to its dynamic and adjustable control parameters that allow it to investigate and explore search spaces efficiently [44–47].

In this paper, CHIO is adapted and modified to handle PPSH and achieve its objectives efficiently. CHIO is inspired by the herd immunity strategy to tackle the spreading of coronavirus pandemics (COVID-19). CHIO's primary goal is to find the best protection for society against the disease by transforming the bulk of the susceptible community that is not infected by the virus to become immune. Due to the discrete nature of PPSH, CHIO is adapted and modified to address the discrete PPSH and to achieve its objectives efficiently. CHIO is used for PPSH due to its powerful accomplishment in exploring rugged, constrained, and complex search spaces. In addition, it has a high ability to maintain the balance between exploitation and exploration in finding the optimal/near-optimal solution(s). Seven scenarios are used in the evaluation study to evaluate CHIO performance. Each scenario contains up to twenty-three smart appliances to be scheduled in accordance with a dynamic pricing scheme. The main CHIO parameters are tuned to find their best values. The results obtained by CHIO are statistically analyzed to evaluate and describe its performance clearly and carefully. In addition, CHIO's results are compared with that of five optimization algorithms, including genetic algorithm (GA), particle swarm optimization (PSO), grey wolf optimizer (GWO), wind-driven optimization (WDO), and differential evolution (DE).

The paper is structured as follows. Section 2 discusses PPSH background and formulations in terms of a single objective and multi-objective. Sections 3 and 4 provide a comprehensive description of the inspiration and adaptation of CHIO for PPSH, respectively. In Section 5, the experimental results of the proposed approach are presented and described. Section 6 concludes the paper.

2. Power Scheduling Problem in Smart Home

This section presents and discusses the most significant state-of-the-art studies that tackled PPSH using metaheuristic algorithms. In addition, a general formulation for PPSH including all its objectives is presented. Furthermore, PPSH formulation in terms of single objective and multi-objective are modeled.

2.1. Related Work

According to [1] a massive number of optimization algorithms were adapted to address PPSH including exact and metaheuristic algorithms. The metaheuristic algorithms are the most popular because they can explore rugged search spaces and find the best solutions. Accordingly, the most popular metaheuristic algorithms used to address PPSH are reviewed in this section.

The work Aslam et al. [48] implemented three popular metaheuristic algorithms, including GA, cuckoo search, and crow search algorithm to handle PPSH and find an optimal way to reduce PAR and EB. The proposed algorithms were tested in two scenarios, including 31 smart homes. Two of the most popular pricing schemes were used, including critical period price and RTP. The results showed the high performance of used algorithms in optimizing the values of PAR and EB.

A multi-objective approach was proposed to optimize UC and EB by Soares et al. [49]. GA was adapted for the proposed approach to achieve the optimal schedule for smart appliances. Six smart appliances were used within 36 h to calculate the power consumed. The results of the proposed method were presented and analyzed in different aspects. The proposed method achieved good results in reducing EB and improving the UC level considering the power system's stability.

Two metaheuristic algorithms were adapted to handle PPSH and optimize its objectives by Rasheed et al. [50]. These algorithms are WDO and PSO. The authors used several scenarios to evaluate the performance of the adapted algorithms within 24 h. WDO shows better performance than PSO in obtaining the best schedules.

GA for efficient scheduling in a smart home is applied in Zhao et al. [2]. The proposed approach combined two pricing schemes, including RTP and IBR, to achieve the best power system stability and EB. The problem was formulated as a multi-objective optimization problem to reduce EB and improve the UC level. In the simulation results, the proposed method significantly impacted the whole electricity system, where it obtained encouraging performance in achieving the objectives.

Two metaheuristic algorithms, including crow search algorithm and grasshopper optimization algorithm, were adapted to address PPSH optimally by Ullah et al. [51]. The adapted algorithms were tested and evaluated within 24 h using several scenarios. The results proved that the grasshopper optimization algorithm obtains high performance as it achieved better results than crow search algorithm and other metaheuristics algorithms.

An efficient model that accumulates two residential areas within a smart grid via a wide area network was proposed by Rahim et al. [52]. The proposed model main objectives are to reduce PAR, EB, execution time, and user discomfort. EBs were calculated based on two pricing schemes, including time-of-use price and IBR. The authors formulated the objectives as a multi-objective optimization function. Three optimization algorithms, including GA, ant colony optimization, and binary PSO, were tailored and evaluated on 13 home appliances within 24 h. The results demonstrated that GA excels ant colony optimization and binary PSO concerning EB.

A new model is proposed in Muralitharan et al. [53] to balance the amount of power consumed over a time horizon by applying the concept of a threshold limit and using a multi-objective evolutionary algorithm. The primary benefit of achieving such a goal is to minimize EB and user discomfort. The experiments were performed on ten appliances to assess the performance of the proposed approach. The threshold limit concept is applied during the scheduling process by switching off several appliances once the used power amount is exceeded the threshold limit. The results demonstrated that the proposed approach provided a good compromise between EB and UC level.

The GWO is adapted as a multi-objective optimization approach in Makhadmeh et al. [8] to tackle PPSH. The proposed approach seeks optimal scheduling that simultaneously reduces EB, PAR, and user discomfort. Seven scenarios were designed to evaluate the proposed approach's performance. RTP and IBR schemes were considered to calculate the EB because the IBR's mechanism solely caused dispersing power consumption over the time horizon that resulted from maintaining the power consumption in a certain limit for a specific time slot. The simulation results provide a comparison between GWO and GA on the basis of the datasets defined by the authors. GWO obtained better results than GA. GWO achieved significant savings in EB and PAR. Moreover, GWO compared against 19 state-of-the-art algorithms using the recommended settings related to the consumption profile considered in these algorithms. The results demonstrated that GWO nearly surpasses other algorithms in minimizing EB and PAR.

PSO was adapted to tackle PPSH by Makhadmeh et al. [54]. The main target of PSO is to provide an adequate solution to PPSH by minimizing EB and PAR and maximizing UC level. Unlike the previous studies, this research added new factors, namely smart batteries, to provide more efficient scheduling to the appliances. The simulation results demonstrated that the added smart battery is significantly improved the results in terms of EB, PAR, and UC level. Furthermore, the performance of PSO was compared against GA and the results showed that PSO surpasses GA in terms of trade-offs between PPSH objectives.

Bacterial foraging optimization algorithm and strawberry algorithm were adapted to tackle PPSH by Khan et al. [55]. Having optimal scheduling of power for the smart home appliances entails an optimal minimizing of EB and PAR. To make this goal possible, bacterial foraging optimization algorithm and strawberry algorithm examined the possible permutation solutions for PPSH that efficiently minimize EB and PAR as much as possible. The simulation results showed that both optimization algorithms reduced total EB and PAR by scheduling the load from peak hours to off-peak hours. Results also showed that bacterial foraging optimization algorithm reduced EB better than the strawberry algorithm.

2.2. PSPSH Formulation

This section elaborates on the mathematical modelling of PSPSH. The section starts with smart home appliances classification to model power consumption. Subsequently, EB, PAR, and UC parameters are formulated. Finally, a multi-objective model that combined all PSPSH objectives is presented.

2.2.1. Power Consumption

Any smart home can have shiftable appliances (SAs) and non-shiftable appliances (NSAs). SAs can work autonomously, but users can set their time parameters ahead of time, such as when and how long it will be operated. As an illustration, users could set such parameters for the cloth dryer to work during a suitable time. On the other hand, NSAs need to be operated manually, and users cannot set their time parameters. As an illustration, users have no means to set periods to start the TV ahead of time.

This study investigates how to satisfy the objectives of PSPSH through setting SAs at times that meet users' requests while NSAs functions manually.

Let us say S and NS are SAs and NSAs vectors, as shown in Equations (1) and (2).

$$S = [s_1, s_2, \dots, s_m], \tag{1}$$

$$NS = [ns_1, ns_2, \dots, ns_q], \tag{2}$$

where s_1 represents the first SA in S , s_m is the last SA in S , ns_1 is the first NSA in NS , ns_q is the last NSA in NS , and m and q are the total number of SAs and NSAs, respectively.

In each smart home, the amount of consumed power of each SA can be modeled as follows:

$$PS = \begin{bmatrix} ps_1^1 & ps_2^1 & \dots & ps_m^1 \\ ps_1^2 & ps_2^2 & \dots & ps_m^2 \\ \vdots & \vdots & \dots & \vdots \\ ps_1^n & ps_2^n & \dots & ps_m^n \end{bmatrix}, \tag{3}$$

where ps_i^j is the amount of power consumed by s_i when we have a time interval t^j . t^j is the time interval within the time boundary T which as shown in Equation (4). n is the total number of time intervals in T

$$T = [t^1, t^2, \dots, t^n], \tag{4}$$

As mentioned earlier, users can set the time parameters for SA. These time parameters the allowing period for SAs to be operated (OTP) and their operation cycle (LOC). Users can set the starting time (OTP_s) and the ending time (OTP_e) of OTP as shown in Equations (5) and (6).

$$OTP_s = [OTP_{s1}, OTP_{s2}, \dots, OTP_{sm}], \tag{5}$$

$$OTP_e = [OTP_{e1}, OTP_{e2}, \dots, OTP_{em}], \tag{6}$$

where OTP_{s1} and OTP_{e1} symbolize the first appliance starting and ending time, respectively. On the other hand, OTP_{sm} and OTP_{em} is the last appliance starting and ending time, respectively.

For the second time parameter, LOC of SAs is presented as follows:

$$LOC = [l_1, l_2, \dots, l_m], \tag{7}$$

where l_1 symbolize the LOC of the first SA, and l_m is the LOC of the last SA. Moreover, the vectors St and Et have starting and ending time of SAs operations, respectively, (see Equations (8) and (9)). The time parameters presented previously are illustrated in Figure 1.

$$St = [st_1, st_2, \dots, st_m], \tag{8}$$

$$Et = [et_1, et_2, \dots, et_m], \tag{9}$$

where st_1 and et_1 symbolize the starting and ending activities of s_1 . In addition, st_m and et_m stand for the starting and ending activities of s_m .

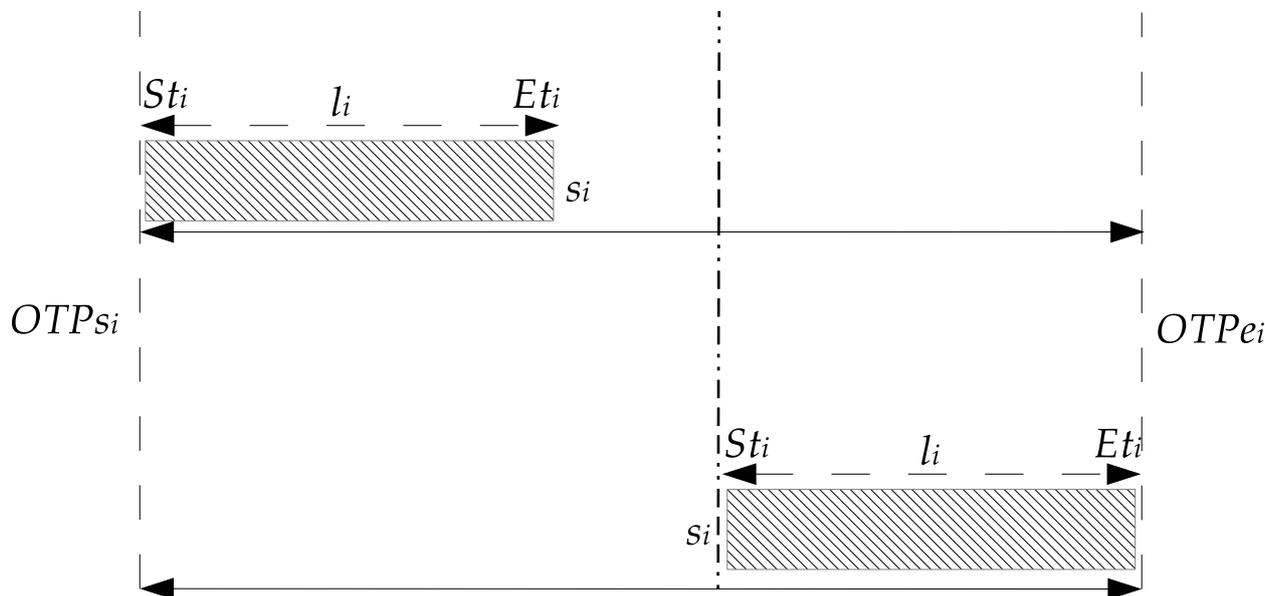


Figure 1. Time parameters illustration.

As discussed previously, users cannot set the time parameters for NSAs. As a result, the power consumption of NSAs (PNS) is modeled without taking into account T in the expression presented in Equation (10)

$$PNS = [pns_1, pns_2, \dots, pns_q], \tag{10}$$

where pns_1 symbolizes how much power is consumed by ns_1 , and pns_q is the power consumption of ns_q .

2.2.2. Electricity Bill (EB)

Reducing EB is one of the main reasons that trigger users to optimize power consumption. Equation (11) can be used to calculate EB for SAs. It is worth mentioning that EB for NSAs cannot be calculated because their operations time is not available.

$$EB = \sum_{j=1}^n \sum_{i=1}^m ps_i^j \times pc^j, \tag{11}$$

where pc^j represents the electricity cost at the time interval j .

In this paper, RTP is applied as a dynamic pricing scheme. The use of both RTP and IBR is followed because of the IBR efficiency in spreading the power consumption of SAs to preserve the stability of the power system [2]. The IBR have two levels of costs, including standard prices and high prices (see Equations (12) and (13)).

$$pc^j = \begin{cases} a^j & \text{if } 0 \leq ps^j \leq C \\ b^j & \text{if } ps^j > C \end{cases}, \tag{12}$$

where a^j is the standard price and b^j is the high price. ps^j is how much power is consumed by SAs during the period of time j , and C is a threshold between a^j and b^j .

$$b^j = \lambda \times a^j, \tag{13}$$

where λ is the ratio between a^j and b^j and it is a positive number.

2.2.3. Peak-to-Average Ratio (PAR)

PAR is the percentage between the highest power consumption and the average power consumption in T . Ideally, PAR will be reduced to balance power consumption and keep the power system stable. PAR is calculated using the following formula:

$$PAR = \frac{PS_{max}}{PS_{Avg}}, \tag{14}$$

where

$$PS_{Avg} = \frac{\sum_{j=1}^n ps^j}{n},$$

where PS_{max} represents the maximum power consumed by SA during T and PS_{avg} symbolizes the average power consumed within the same period.

2.2.4. User Comfort (UC) Level

UC level can be enhanced by shortening the waiting time when running SAs (waiting time rate (WTR)) [2]. Another factor for enhancing the UC level is to increase the power available to run NSAs within C (capacity power limit rate (CPR)) [8]. Both of these parameters are considered for scheduling to decrease the waiting time for SAs and maximize the power available for NSAs.

The first UC parameter WTR is mathematically modeled as follows:

$$WTR_i = \frac{st_i - OTPs_i}{OTPe_i - OTPs_i - l_i}, \quad \forall i \in S, \tag{15}$$

To compute the average WTR for all SAs we can use Equation (16).

$$WTR_{avg} = \frac{\sum_{i=1}^m (st_i - OTPs_i)}{\sum_{i=1}^m (OTPe_i - OTPs_i - l_i)}, \tag{16}$$

The second UC parameter CPR is mathematically modeled as follows:

$$CPR^j = \frac{\sum_{k=1}^q ONA_k^j}{q}, \tag{17}$$

where ONA symbolize how many NSAs have operation power that is beyond the available power at the time j . It is mathematically modeled as follows:

$$ONA_k^j = \begin{cases} 0 & \text{if } PNS_k < AP^j \\ 1 & \text{if } Otherwise, \end{cases} \tag{18}$$

where AP^j represents the available power to operate NSAs at time j . Please note that AP^j is computed based on how much power is consumed by SAs at any time j and C using the formula:

$$AP^j = C - PS^j, \tag{19}$$

The computation of the mean of CPR for T is mathematically modeled as follows:

$$CPR_{avg} = \frac{\sum_{j=1}^n \sum_{k=1}^q ONA_k^j}{q \times n}, \tag{20}$$

Observe that the range of values for WTR_{avg} and CPR_{avg} is within the interval 0 and 1. As a result, the ratio of UC can be computed using the following formula:

$$UC_p = (1 - (\frac{WTR_{avg} + CPR_{avg}}{2})) \times 100\%, \quad (21)$$

2.3. Multi-Objective Function

In this study, the objective function of PSPSH is formulated as a multi-objective function by formulating PSPSH as a multi-objective optimization problem. The primary aim of this formulation is to optimize all PSPSH objectives, including EB, PAR, WTR, and CPR, simultaneously. As mentioned previously, all these objectives affect users and power supply companies at the same time, where users need to be motivated to schedule their appliances operations by minimizing EB and UC level, which contains WTR and CPR. There is a trade-off between optimizing EB and UC, where minimizing EB could decrease UC and vice versa. In contrast, power supply companies achieve benefits from the proposed formulation by optimizing PAR. PAR parameter is used to minimize the highest power consumption for users, accordingly, enhance power generation and distribution systems of the supply companies.

A non-Pareto scalarization approach, called the weighted sum, is used for the multi-objective function due to ineffectiveness of Pareto optimality to deal with problems contain more than three objectives, such as PSPSH [56–58]. The weighted sum method is used for PSPSH due to its easy implementation, simplicity, non-complexity, and wide use by PSPSH literature [59–63]. The formulation of PSPSH's multi-objective function is modeled based on Equations (11), (14), and (16) as follows:

$$\min F(X) = w_1 \times \frac{EB}{EB + A} + w_2 \times \frac{PAR}{PAR + B} + w_3 \times WTR_{avg} + w_4 \times CPR_{avg}, \quad (22)$$

where A and B represent two positive numbers, and w_1, w_2, w_3 , and w_4 are weight parameters denoting importance of each objective function.

3. Coronavirus Herd Immunity Optimizer

Recently, a new natural-inspired human-based metaheuristic algorithm called CHIO has been proposed by Al-Betar et al. [43]. The idea is inspired by the herd immunity used as a mechanism to stop the Coronavirus pandemic. In herd immunity, most of the population must be infected and recovered from Coronavirus. This partial population will shield the remaining individuals from being infected where the immune individual stands as a firewall to prevent the infection of susceptible individuals. Such inspiration is formulated as an optimization algorithm and tested using various standard test functions and engineering problems. In this section, the inspiration of CHIO is illustrated, and its procedural optimization steps are provided.

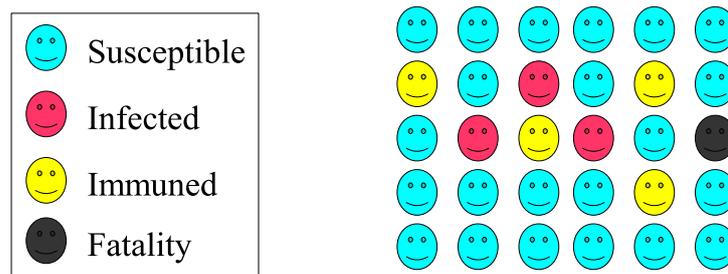
3.1. Inspiration

In biology, the virus's replications and spreading are essential features where the host (infected) individual can easily transmit it to other individuals through direct communication. In December 2019, a novel respiratory Coronavirus (2019-nCov) was identified in Wuhan, China. Therefore, the World Health Organization announced the name of the new contagious disease to be Corona Virus Disease (COVID-19) [64]. The pandemic is very quickly spread worldwide and affects their economic, political, social sides. The incubation period of COVID-19 from the time individual caught the virus until the symptoms first appeared was estimated from 2 to 14 days. Currently, no antiviral treatment is yet recommended for the COVID-19 infection. The treatment depends on the immunity system of the infected individuals. Until discovering an approved vaccine as antiviral to COVID-19 infection, some countries protect their population by directing them to follow the health care standards such as wearing masks, committing to social distancing, lockdown, etc. Other

countries, like the UK, preferred to implement herd immunity rules to yield a self-protected population and thus control the COVID-19 epidemic outbreak [65].

The social distancing strategy is recommended by World Health Organization to slow down the virus spreading. The COVID-19 can be transmitted from individual to individual through direct contact with the predefined distance estimated around 1.8 m or indirect contact through objects or surfaces around the persons. The transmitting media is essentially the small droplets from the mouth or nose when the infected person sneezes, coughs, or exhales. Social distancing is affected by the basic reproduction rate, which is the number of people that can catch COVID-19 from an infected person. The higher the basic reproduction rate the quicker the spreading of the virus; thus, the fatality rate will be increased. The fatality rate determines the percentage of the infected individuals to die. This is affected by the immunity systems of the individuals. Older people and those with chronic diseases have a higher fatality rate. Therefore, the population’s average age is an essential factor for recovering or not [66].

In the herd immunity principle, the virus is transmitted from one individual to others until most of the population is infected and recovered. Therefore, the protected individuals’ immunity systems can stop the spreading of COVID-19 from being transmitted to the susceptible individuals [67]. This is visualized in Figure 2, where the immune individuals can downsize the virus from spreading from infected individuals to the susceptible ones. The immune individuals shield the susceptible ones as a firewall against virus spreading.



Infectious agent pass freely from infected to susceptible



Infectious agent cannot pass freely from immuned to susceptible



Figure 2. Herd immunity of COVID-19.

The herd immunity threshold is estimated by 60%, which determines the percentage of population individuals shall be immune to protect other susceptible individuals.

Normally herd immunity can be implemented as follows [65,67,68] :

- A group of infected individuals infects a group of susceptible individuals.
- The majority of the infected people recover and gain immunity against COVID-19, and a low rate are dying.
- The immune individual will stop the virus from spreading; thus, the population is protected.

3.2. Optimization Steps of CHIO

In terms of optimization, CHIO is described as six steps illustrated below. The pseudo-code is given in Algorithm 1, and the flowchart describing the flow of work is provided in Figure 3.

Algorithm 1 CHIO pseudo-code.

```

1: {———— Step 1: Initialize the CHIO parameters ————}
2: Initialize the parameters ( $HIP$ ,  $S_r$ , and  $Max_{Age}$ ).
3: {———— Step 2: Generate herd immunity population ————}
4:  $x_o^y = lb_o + (ub_o - lb_o) \times U(0, 1)$ ,  $\forall o = 1, 2, \dots, d$ , and  $\forall y = 1, 2, \dots, HIP$ .
5: Calculate the fitness of each search agent
6: Set  $\mathcal{ST}_y = 0 \quad \forall y = 1, 2, \dots, HIP$ .
7: Set  $\mathcal{A}_y = 0 \quad \forall y = 1, 2, \dots, HIP$ .
8: {———— Step 3: Herd immunity evolution ————}
9: while ( $t \leq I$ ) do
10:   for  $y = 1$  to  $HIP$  do
11:      $is\_Corona(x^y(t + 1)) = \text{false}$ 
12:     for  $o = 1$  to  $d$  do
13:       if ( $r < \frac{1}{3} \times BR_r$ ) then
14:          $x_o^y(t + 1) = C(x_o^y(t))$     {See Equation (26)}
15:          $is\_Corona(x^y(t + 1)) = \text{true}$ 
16:       else if ( $r < \frac{2}{3} \times BR_r$ ) then
17:          $x_o^y(t + 1) = N(x_o^y(t))$     {See Equation (28)}
18:       else if ( $r < BR_r$ ) then
19:          $x_o^y(t + 1) = R(x_o^y(t))$     {See Equation (30)}
20:       else
21:          $x_o^y(t + 1) = x_o^y(t)$ 
22:       end if
23:     end for
24:     {———— Step 4: Update herd immunity population ————}
25:     if ( $f(x^y(t + 1)) \leq f(x^y(t))$ ) then
26:        $x^y(t) = x^y(t + 1)$ 
27:     else
28:        $\mathcal{A}_y = \mathcal{A}_y + 1$ 
29:     end if
30:     if  $f(x^y(t + 1)) < \frac{f(x)^y(t+1)}{\Delta f(x)} \wedge \mathcal{ST}_y = 0 \wedge is\_Corona(x^y(t + 1))$  then
31:        $\mathcal{ST}_y = 1$ 
32:        $\mathcal{A}_y = 1$ 
33:     end if
34:     if  $f(x^y(t + 1)) > \frac{f(x)^y(t+1)}{\Delta f(x)} \wedge \mathcal{ST}_y = 1$  then
35:        $\mathcal{ST}_y = 2$ 
36:        $\mathcal{A}_y = 0$ 
37:     end if
38:     {———— Step 5: Fatality condition ————}
39:     if ( $(\mathcal{A}_y \geq Max_{Age}) \wedge (\mathcal{ST}_y == 1)$ ) then
40:        $x_o^y = lb_o + (ub_o - lb_o) \times U(0, 1)$ ,  $\forall o = 1, 2, \dots, d$ .
41:        $\mathcal{ST}_y = 0$ 
42:        $\mathcal{A}_y = 0$ 
43:     end if
44:   end for
45:    $t = t + 1$ 
46: end while

```

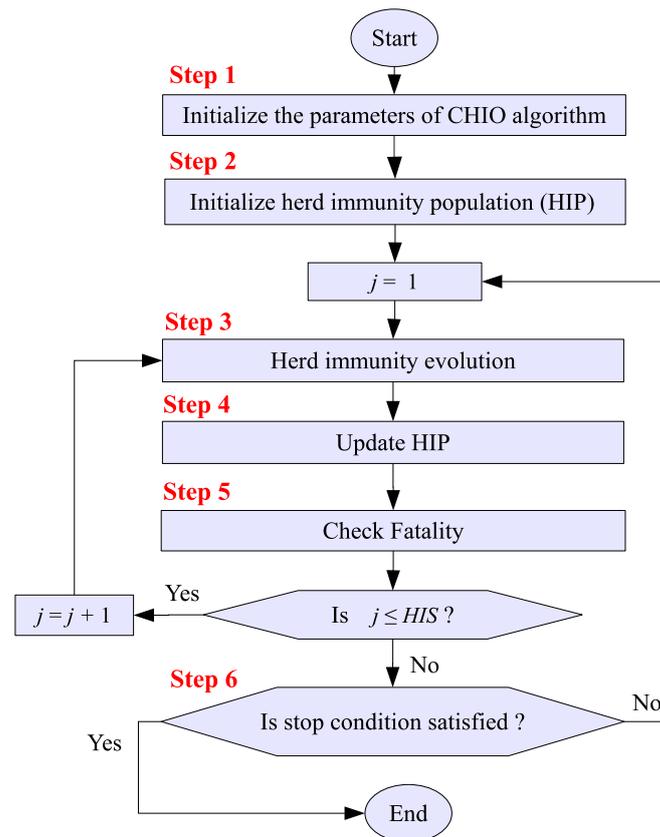


Figure 3. The flowchart of CHIO algorithm.

Step 1: Initialization

Initially, any problem must be modeled as an optimization problem to be addressed by optimization algorithms. In addition, two main parts should be identified, including the solution representation and the objective function. The solution of a constrained optimization problem can be formulated as a vector $x = (x_1, x_2, \dots, x_d)$ of d decision variables. Each decision variable has its value range $x_o \in X_o$ where $X_o \in [x_o^{min}, x_o^{max}]$. where x_o^{min} denotes the minimum value range and x_o^{max} denotes the maximum value range. Accordingly, the objective function of the solution x can be formulated as follows:

$$\begin{aligned}
 &\min \quad f(x) \\
 &S.t. \quad g_b(x) = w_b \quad b \in (1, 2, \dots, a) \\
 &\quad \quad h_z(x) = e_z \quad z \in (1, 2, \dots, l)
 \end{aligned} \tag{23}$$

where $f(x)$ is the objective function to measure the fitness of solution x . $g_b(x) = w_b$ is the set of a th equality constraints while $h_z(x) = e_z$ is the set of l th inequality constraints. CHIO has two types of parameters: algorithmic and control. The algorithmic parameters are maximum number of iterations (I), herd immunity population size (HIP), and C_0 which is the initial infected cases. CHIO also has two control parameters, including basic reproduction rate (BR_r) denotes the rate of transmitting the virus from individual to another and max age (Max_{Age}) which determines the status of infected individual according to its infection age.

Step 2: Initialize herd immunity population

The herd immunity population (HIP) is a memory matrix of size $d \times HIP$ stored in CHIO individuals. These individuals are initialized normally concerning the equality

and inequality constraints. HIP matrix is represented in Equation (24). The objective function values (or immunity rates) are computed for all initialized solutions using Equation (23).

$$\mathbf{HIP} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^{HIP} & x_2^{HIP} & \dots & x_d^{HIP} \end{bmatrix} \cdot \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HIP}) \end{bmatrix}. \tag{24}$$

To keep tracking the status of HIP individuals, the status vector $ST = (ST_1, ST_2, \dots, ST_{HIP})$ of size HIP is initialized by zeros as many as $HIP - C_0$ and ones as many as C_0 . Please note that zeros and ones refer to the susceptible and infected cases, respectively.

Step 3: Herd immunity evolution

In this step, A new CHIO solution is generated based on three rules, which are discussed below.

Infected case: the decision variable o in the solution y , such as $x_o^y(t + 1)$, will be modified based on social distancing calculated based on the difference between the current decision variable and a decision variable taken from any infected case with a probability $r \in [0, \frac{1}{3}BR_r)$, such as:

$$x_o^y(t + 1) = C(x_o^y(t)) \tag{25}$$

where

$$C(x_o^y(t)) = x_o^y(t) + r \times (x_o^y(t) - x_o^c(t)) \tag{26}$$

where $x_o^c(t)$ is selected from infected case x^c whereby the status vector ($ST_c = 1$).

Susceptible case: the decision variable o in the solution y will be modified based on social distancing calculated the difference between the current decision variable and a decision variable taken from any susceptible case with a probability of $r \in [\frac{1}{3}BR_r, \frac{2}{3}BR_r)$, such as:

$$x_o^y(t + 1) = N(x_o^y(t)) \tag{27}$$

where

$$N(x_o^y(t)) = x_o^y(t) + r \times (x_o^y(t) - x_o^h(t)) \tag{28}$$

where $x_o^h(t)$ is selected from any susceptible case x^h based on the status vector ($ST_h = 0$).

Immune case: the decision variable o in the solution y will be modified based on social distancing calculated the difference between the current decision variable and a decision variable from any immune case with a probability of $r \in [\frac{2}{3}BR_r, BR_r)$ such as:

$$x_o^y(t + 1) = R(x_o^y(t)) \tag{29}$$

where

$$R(x_o^y(t)) = x_o^y(t) + r \times (x_o^y(t) - x_o^v(t)) \tag{30}$$

where $x_o^v(t)$ is selected from the best immune case x^v with regard to the status vector (ST) such that:

$$f(x^v) = \min_{y \sim \{Y | ST_Y=2\}} f(x^y).$$

In summary, the operations of the three rules can be formulated as follows:

$$x_o^y(t+1) \leftarrow \begin{cases} x_o^y(t) & r \geq BR_r \\ C(x_o^y(t)) & r < \frac{1}{3} \times BR_r. \quad // \text{infected case} \\ N(x_o^y(t)) & r < \frac{2}{3} \times BR_r. \quad // \text{susceptible case} \\ R(x_o^y(t)) & r < BR_r. \quad // \text{immune case} \end{cases} \quad (31)$$

Step 4: Updating HIP

The fitness value (or immunity rate) of each generated solution $f(x^y(t+1))$ is computed using fitness function. The generated solution $x^y(t+1)$ replaces the current one $x^y(t)$, when $f(x^y(t+1)) < f(x^y(t))$. In case the replacement is done, the Age value of such solution is increment by one (i.e., $A^y = A^y + 1$) if the current solution $x^y(t)$ is infected case ($ST^y = 1$).

In addition, CHIO updates the status vector (ST^y) for generated solution x^y based on the herd immune threshold formulated in Equation (34)

$$ST^y \leftarrow \begin{cases} 1 & \text{if } f(x^y(t+1)) < \frac{f(x^y(t+1))}{\Delta f(x)} \wedge ST_y = 0 \wedge is_Corona(x^y(t+1)) \\ 2 & \text{if } f(x^y(t+1)) > \frac{f(x^y(t+1))}{\Delta f(x)} \wedge ST_y = 1 \end{cases} \quad (32)$$

Please note that $is_corona(x^y(t+1))$ is a binary value set to one if $x^y(t+1)$ inherited a value from any infected case. The $\Delta f(x)$ is the fitness mean value of the individuals in HIP, such as $\frac{\sum_{o=1}^{HIP} f(x_o)}{HIP}$.

Step 5: Check Fatality

This step decides whether the infected cases (i.e., $x^y(t+1) \wedge (ST_y == 1)$) are dead or immune. This is specified by the parameter Max_{Age} . When the infected case's age exceeds the Max_{Age} limit such that $A_y \geq Max_{Age}$ without any improvement, the infected case will be died (or be removed from the HIP) and it is regenerated from scratch. Furthermore, the A_y and ST_y are set to zero. This is the main operator for diversification.

Step 6: Stop condition

The evolution steps (Step 3 to Step 5) are repeated until the HIP is only contained in either susceptible or immune cases but not infected cases. Typically, the maximum number of iterations is used as a stopping criterion.

4. The Proposed CHIO-PSPSH

In this section, CHIO is modified and adapted to address PSPSH in terms of discrete search space. CHIO is used for PSPSH due to its high and robust performance in addressing large-scale search spaces like the one of PSPSH. CHIO proved its robustness when applied to tackle different research fields, such as travelling salesman problems [69], vehicle routing problems [44], feature selection problems [45], wheel motor design problems [70], intrusion detection systems [46], and others [47,71,72]. CHIO has achieved its popularity since its foundation due to its dynamic parameters that allow it to explore search spaces efficiently and find optimal solutions. In addition, CHIO contains two adjustable control parameters that enhance its searchability for better investigation.

The adaptation of CHIO for PSPSH contains several steps that describe its behaviour in finding optimal/near-optimal solutions. These steps are deeply discussed below.

Step 1: Initialize CHIO-PSPSH parameters

The first step of adapting CHIO to handle PSPSH is initializing CHIO and PSPSH parameters. The main parameters of PSPSH that must be initialized are the home

appliances **S** and **NS**, length of operation cycle **LOC**, time horizon **T**, power required by appliance **PS** and **PNS**, boundary of operation time periods **OTPs** and **OTPe**, and electricity prices **pc**.

For the CHIO parameters are the maximum number of iterations (**I**), population size (**HIP**), Spreading rate (BR_r), Max Age (Max_{Age}), and number of initial infected (C_0).

Step 2: Initialize CHIO-PSPSH population

PSPSH’s solutions are generated randomly in this step by the CHIO considering the discrete nature of PSPSH. The solutions are presented in a vector of length m containing the SA’s starting time st . Figure 4 shows an example of the solution representation. The solution’s length is 5 ($m = 5$), and each bit in the solution contains st of an SA in the range $[0, 24 - l]$.

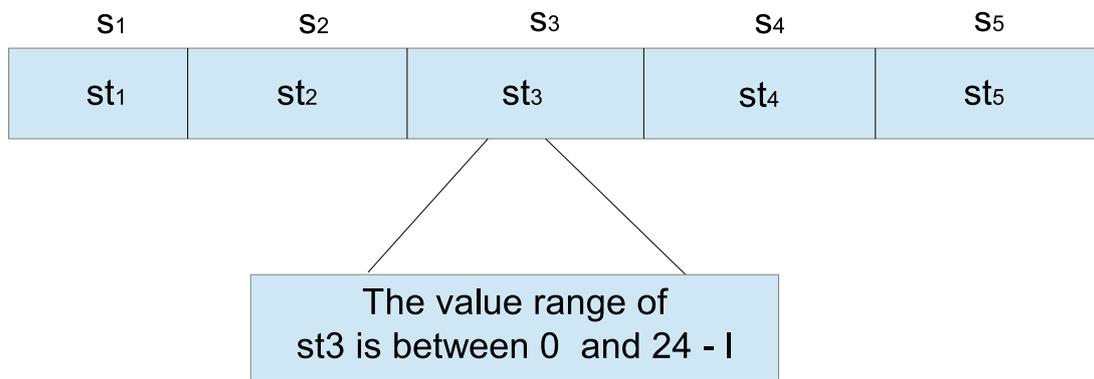


Figure 4. Solution representation of PSPSH.

The PSPSH population contains HIP number of solutions, as shown in Equation (33).

$$PSPSH\ population = \begin{bmatrix} st_1^1 & st_2^1 & \dots & st_m^1 \\ st_1^2 & st_2^2 & \dots & st_m^2 \\ \vdots & \vdots & \dots & \vdots \\ st_1^N & st_2^N & \dots & st_m^{HIP} \end{bmatrix}, \tag{33}$$

where st_i^y denote st of SA i in solution y .

Step 3: Calculate fitness function

Each solution is evaluated based on Equation (22) in this step. After the evaluation, the best solution is assigned with the highest immunity rate ($BestSol$).

Step 4: Coronavirus herd immunity evolution

This step is the primary step of CHIO. The SA i in the solution y may be affected by social distancing or remain the same based on BR_r using three rules as follows:

$$st_i^y(t + 1) \leftarrow \begin{cases} st_i^y(t) & r \geq BR_r \\ C(st_i^y(t)) & r < \frac{1}{3} \times BR_r. \quad // \text{infected case} \\ N(st_i^y(t)) & r < \frac{2}{3} \times BR_r. \quad // \text{susceptible case} \\ R(st_i^y(t)) & r < BR_r. \quad // \text{immune case} \end{cases} \tag{34}$$

Step 5: Update herd immunity population

In this step, the fitness value of each PSPSH’s generated solution $st_i^y(t + 1)$ is calculated and replaced the current solution $st_i^y(t)$ if achieved a fitter value, such that $f(st^y(t + 1)) < f(st^y(t))$. The age vector \mathcal{A}^y is also increased by one if $ST^y = 1$. In addition, the status vector (ST^y) is updated for each solution using Equation (34).

As mentioned previously, the CHIO is modified to deal with discrete optimization problems. Therefore, in this step, once the PSPSH population is updated, all values in the population will be converted from continuous to discrete values.

Step 6: Fatality cases

This step decides whether the infected solution (i.e., $st^y(t + 1) \wedge (ST^y == 1)$) is dead or immune. This is specified by the parameter Max_{Age} . When the Age of the infected solutions exceed the Max_{Age} limit without any improvement, the infected solutions are removed from the population and regenerated. Furthermore, the age and ST^y are set to zero.

Step 7: Stop criterion

CHIO repeats Step 3 to Step 6 until reach the algorithm stop criterion. Figure 5 presents the flowchart of adapting CHIO for PSPSH. Table 1 shows the mapping between the components of PSPSH and CHIO in the optimization processes.

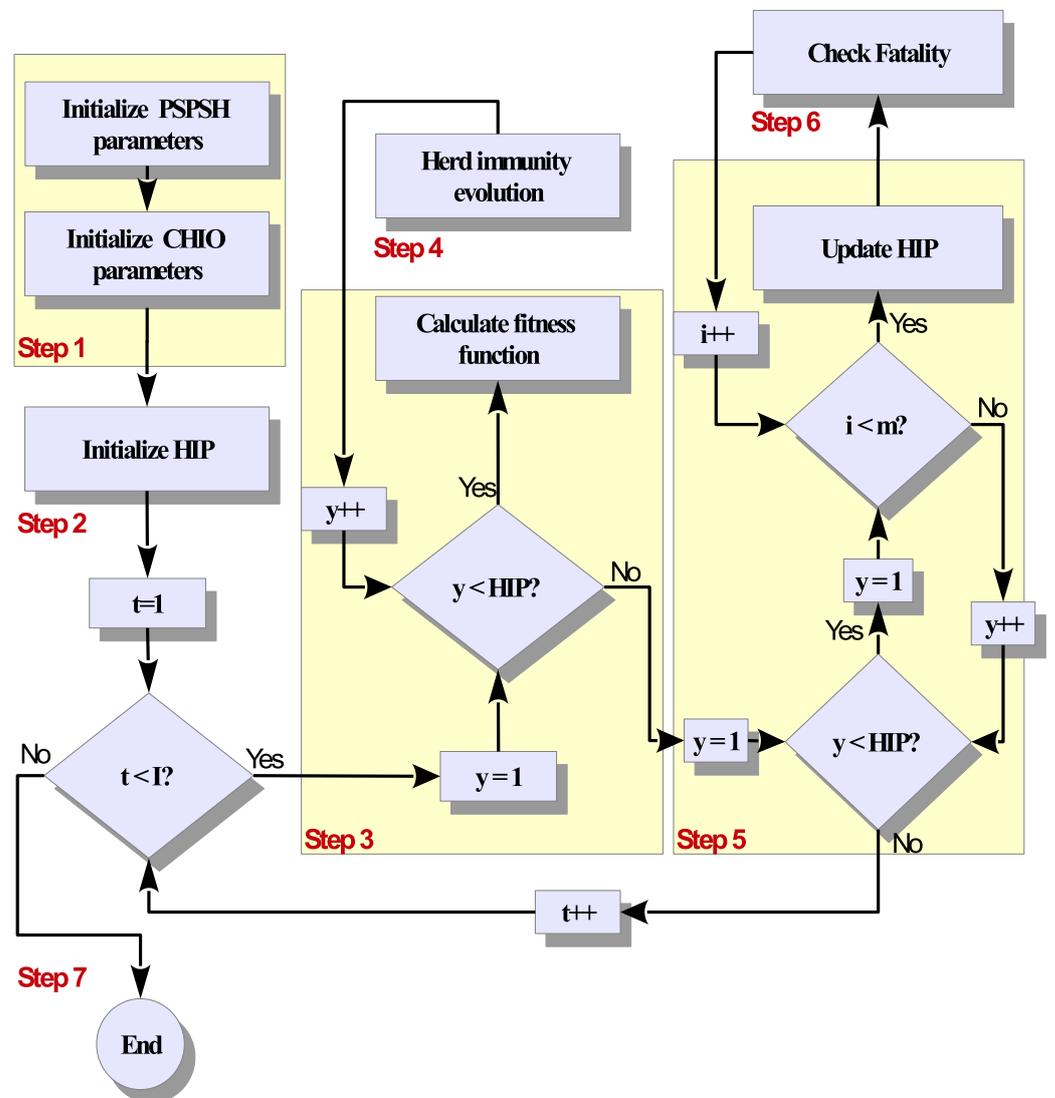


Figure 5. The flowchart of CHIO-PSPSH.

Table 1. Mapping between the components of PSPSH and CHIO.

PSPSH Process	CHIO Process
Candidate Solution	Infected, susceptible, and immune cases
Social Distancing	Pick a random case and rely on the basic reproduction rate
Optimal Solution	immune cases
Fitness Value	Immunity rate
Fatality Rate	Reach the maximum age
Population	HIP

5. Experimental Results

In this section, the proposed CHIO-PSPSH method is experimentally examined to analyze and evaluate the CHIO performance in addressing PPSH and achieving its objectives using seven different consumption scenarios. The main CHIO parameters are tuned to find their best values. CHIO-PSPSH performance is statistically evaluated to track its behaviour in obtaining the best schedules. CHIO-PSPSH’s results are compared with that of five optimization methods, including GA, PSO, GWO, WDO, and DE, using the same consumption scenarios.

5.1. Experimental Design

The proposed CHIO-PSPSH is tested using seven different consumption scenarios and each scenario is evaluated separately. As mentioned previously, the dynamic pricing scheme used in this study is a combination scheme of the RTP and IBR schemes. The RTP scheme is adopted from Commonwealth Edison Company from 1st to 7th of June, 2016 (7 days). λ and C are used by the IBR scheme and are set to be 1.543 and 0.0333 per time slot, respectively, (Equation (12)) [2,7,54].

The main parameters and characteristics of smart home appliances used in this study (i.e., SAs and NSAs) are highlighted and presented in Tables 2 and 3. Up to 36 SA and 14 NSA are used by each of the seven scenarios, as shown in Table 4. In this table, the Scenario’s column refers to the scenario number, and the Appliances’ column refers to the appliance number in Table 2.

Table 2. Main characteristics of the appliance time parameters.

NO.	Appliances	LOC	OTPs–OTPe	NO.	Appliances	LOC	OTPs–OTPe
1	Dishwasher	105	540–780	19	Dehumidifier	30	1–120
2	Dishwasher	105	840–1080	20	Dehumidifier	30	120–240
3	Dishwasher	105	1200–1440	21	Dehumidifier	30	240–360
4	Air Conditioner	30	1–120	22	Dehumidifier	30	360–480
5	Air Conditioner	30	120–240	23	Dehumidifier	30	480–600
6	Air Conditioner	30	240–360	24	Dehumidifier	30	600–720
7	Air Conditioner	30	360–480	25	Dehumidifier	30	720–840
8	Air Conditioner	30	480–600	26	Dehumidifier	30	840–960
9	Air Conditioner	30	600–720	27	Dehumidifier	30	960–1080
10	Air Conditioner	30	720–840	28	Dehumidifier	30	1080–1200
11	Air Conditioner	30	840–960	29	Dehumidifier	30	1200–1320
12	Air Conditioner	30	960–1080	30	Dehumidifier	30	1320–1440
13	Air Conditioner	30	1080–1200	31	Electric Water Heater	35	300–420
14	Air Conditioner	30	1200–1320	32	Electric Water Heater	35	1100–1440
15	Air Conditioner	30	1320–1440	33	Coffee Maker	10	300–450
16	Washing Machine	55	60–300	34	Coffee Maker	10	1020–1140
17	Clothes Dryer	60	300–480	35	Robotic Pool Filter	180	1–540
18	Refrigerator	1440	1–1440	36	Robotic Pool Filter	180	900–1440

Table 3. NSAs used in the scheduling.

No.	Appliances	Power (kW)
1	Light [52]	0.6
2	Attic Fan [73]	0.3
3	Table Fan [73]	0.8
4	Iron [52]	1.5
5	Toaster [73]	1
6	Computer Charger [73]	1.5
7	Cleaner [2]	1.5
8	TV [73]	0.3
9	Hair Dryer [73]	1.2
10	Hand Drill [73]	0.6
11	Water Pump [73]	2.5
12	Blender [73]	0.3
13	Microwave [52]	1.18
14	Electric Vehicle [74]	1

Table 4. Main characteristics of the scenarios.

S #	Appliances
S #1	1, 3, 4, 5, 6, 7, 15, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35
S #2	1, 2, 4, 5, 6, 7, 10, 11, 12, 18, 25, 26, 27, 28, 29, 31, 32, 33, 34, 36
S #3	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35
S #4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
S #5	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35
S #6	1, 2, 3, 8, 9, 10, 11, 12, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35
S #7	1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36

The proposed CHIO-PSPSH is tested 30 independent times to make a fair comparison. In each test, the algorithm ran 1000 iterations since 1000 generations are sufficient for the convergence of the algorithm [7,75]. Table 5 shows CHIO parameters setting. Please note that RB_r and Max_{Age} parameters are not mentioned in the table because these parameters will be tuned to find their best values.

Table 5. Parameters of CHIO algorithms.

Parameter	Value
Population Size (HIS)	40
Max Iteration (I)	1000

5.2. CHIO Parameters Analyzation

CHIO is analyzed and evaluated using seven consumption scenarios to show its performance in addressing PSPSH. As discussed previously, CHIO’s sensitivity is based on two control parameters RB_r and Max_{Age} . Therefore, the two control parameters are studied and analyzed in this section to find their best values in achieving the proposed CHIO-PSPSH optimal schedules.

The effect of RB_r and Max_{Age} are studied using four values. These values are 0.005, 0.05, 0.1, and 0.5 for RB_r , and 50, 100, 300, and 500 for Max_{Age} [43]. Each value is evaluated using PSPSH objectives, including EB, PAR, WTR, and CPR, as shown in Table 6. The table presents the results obtained by all possible experiments using all possible suggested values of the control parameters, where each value of the first parameter is compared with all values of the second parameter.

Table 6. CHIO Parameters Analysis.

		0.005				0.05				0.1				0.5			
	S #	EB	PAR	WTR	CPR												
50	S #1	43.4641	2.6807	0.2242	0.3227	43.8147	2.6044	0.0094	0.3127	43.7384	2.6045	0.0087	0.3129	43.5523	2.6045	0.0458	0.3164
	S #2	64.1402	2.6446	0.2086	0.3462	64.6208	2.4323	0.0124	0.3453	64.4169	2.4254	0.0147	0.3455	64.2613	2.4169	0.0297	0.3460
	S #3	65.9196	2.3865	0.2182	0.3934	66.4258	2.2181	0.0150	0.3803	66.3190	2.2117	0.0144	0.3805	66.3121	2.2225	0.0405	0.3832
	S #4	63.2499	2.5270	0.2356	0.5101	63.3545	1.8744	0.0431	0.5066	63.3702	1.9665	0.0388	0.5096	62.8679	2.0783	0.0838	0.5135
	S #5	46.5781	2.4151	0.2125	0.3929	46.7815	2.2203	0.0157	0.3806	46.7853	2.2160	0.0124	0.3801	46.6086	2.2139	0.0423	0.3843
	S #6	52.4087	2.5095	0.2031	0.3570	52.6468	2.5175	0.0095	0.3490	52.4176	2.5040	0.0143	0.3500	52.2710	2.5060	0.0388	0.3521
	S #7	63.3368	2.3054	0.2535	0.4800	63.0784	2.004	0.0386	0.4661	63.2179	2.0385	0.0337	0.4656	62.9447	2.0086	0.0763	0.4729
100	S #1	43.5971	2.7084	0.2041	0.3220	43.8176	2.6044	0.0076	0.3122	43.8085	2.6045	0.0089	0.3128	43.4891	2.6024	0.0379	0.3164
	S #2	63.4767	2.5924	0.2266	0.3489	64.4900	2.4248	0.0136	0.3454	64.290	2.4199	0.0123	0.3456	64.5717	2.4332	0.0304	0.3460
	S #3	66.4986	2.3235	0.2269	0.3931	66.6698	2.2246	0.0123	0.3791	66.4452	2.216	0.0138	0.3801	66.0401	2.2203	0.0408	0.3844
	S #4	62.8937	2.4846	0.2365	0.5113	63.3123	1.9471	0.0373	0.5104	63.2181	1.9723	0.0388	0.5104	62.7356	2.0188	0.0875	0.5157
	S #5	46.4773	2.4009	0.2209	0.3931	46.7609	2.2160	0.0154	0.3802	46.4528	2.2117	0.0190	0.3817	46.7672	2.2273	0.0456	0.3841
	S #6	51.9756	2.5213	0.2208	0.3608	52.5642	2.5175	0.0116	0.3491	52.1948	2.4944	0.0153	0.3508	52.2699	2.5060	0.0326	0.3515
	S #7	63.1163	2.3252	0.2346	0.4784	63.3557	2.0195	0.0297	0.4643	63.0719	2.0152	0.0362	0.4661	62.8943	2.0117	0.0723	0.4724
300	S #1	43.5265	2.6543	0.2239	0.3224	43.8004	2.6044	0.0074	0.3126	43.8163	2.6045	0.006	0.3123	43.6746	2.6024	0.0352	0.3156
	S #2	64.3299	2.5915	0.2181	0.3473	64.4096	2.4244	0.0112	0.3454	64.2427	2.4199	0.0129	0.3456	64.4953	2.4341	0.026	0.3459
	S #3	66.0201	2.3541	0.2266	0.3944	66.7045	2.2267	0.0102	0.3790	66.6802	2.2246	0.0102	0.3791	66.3345	2.2246	0.0425	0.3838
	S #4	63.1029	2.4434	0.2352	0.5131	63.0111	1.9239	0.0386	0.5118	63.2983	1.9223	0.0404	0.5112	62.5525	2.0766	0.0785	0.5132
	S #5	46.3968	2.3109	0.2172	0.3955	46.6641	2.2117	0.0167	0.3804	46.4139	2.2010	0.0159	0.3816	46.4839	2.216	0.0379	0.3847
	S #6	52.6838	2.5650	0.1999	0.3565	52.4779	2.5117	0.0102	0.3494	52.1136	2.4887	0.0141	0.3509	52.2548	2.506	0.0295	0.3516
	S #7	63.2469	2.4141	0.2121	0.4740	63.2371	2.0133	0.0296	0.4647	63.1973	2.0118	0.0325	0.4653	62.3167	2.0149	0.081	0.4751
500	S #1	43.3217	2.603	0.2088	0.3234	43.8209	2.6044	0.0066	0.3123	43.8142	2.6045	0.0059	0.3124	43.6684	2.6024	0.0251	0.315
	S #2	64.04	2.6589	0.1998	0.3479	64.6379	2.4359	0.0098	0.3453	64.2235	2.4176	0.0138	0.3456	64.5138	2.4321	0.0284	0.346
	S #3	65.9527	2.3322	0.216	0.3919	66.6666	2.2246	0.0107	0.3791	66.642	2.2225	0.0111	0.3795	66.2984	2.223	0.0451	0.3843
	S #4	63.6569	2.5122	0.2225	0.5076	63.3165	1.9488	0.0350	0.5097	63.0808	1.9197	0.0373	0.5113	62.8417	1.9618	0.0883	0.5173
	S #5	46.5366	2.3837	0.1951	0.3933	46.5411	2.2095	0.0143	0.3808	46.4481	2.2031	0.0149	0.3811	46.3428	2.2165	0.0443	0.3855
	S #6	52.532	2.5412	0.1905	0.358	52.0938	2.4944	0.0126	0.3508	52.3337	2.5002	0.0103	0.35	52.4137	2.5156	0.0321	0.3517
	S #7	62.9267	2.2803	0.2238	0.4807	63.2214	2.0288	0.0304	0.4647	63.0749	2.0071	0.0337	0.466	62.5727	2.0133	0.0839	0.4755

The table proves the robust performance of CHIO when the values of its control parameters RB_r and Max_{Age} are 0.05 and 50, respectively. It obtained the best results six times including best PAR two times, best WTR one time, and best CPR 3 times. Table 7 shows the values of CHIO’s control parameters that achieve the best reduction for each PSPSH objective in each scenario. For example, the best EB for the first scenario is achieved when RB_r and Max_{Age} are 0.05 and 500, respectively. Accordingly, the best control parameters’ values that obtain the best PSPSH schedules are 0.05 and 50 for RB_r and Max_{Age} , respectively. Therefore, these values will be considered in the next stage of the experiment and evaluation.

Table 7. CHIO’s control parameters for PSPSH objectives.

S #	EB	PAR	WTR	CPR
S #1	0.005/500	0.5/100, 300, 500	0.1/500	0.05/100
S #2	0.005/100	0.5/50	0.05/500	0.05/50, 500
S #3	0.005/50	0.1/50	0.05, 0.1/300	0.05/300
S #4	0.5/300	0.05/50	0.05/500	0.05/50
S #5	0.5/500	0.1/300	0.1/50	0.1/50
S #6	0.005/100	0.1/300	0.05/50	0.05/50
S #7	0.5/300	0.05/50	0.05/300	0.05/100

5.3. Illustrative Example

For CHIO-PSPSH behaviour further explanation, an illustrative example of how CHIO components deal with PPSH is presented in this section. As mentioned previously, adapting CHIO to deal and address PPSH contains six main steps, as shown in Figure 5.

In the first step (Initialize CHIO-PSPSH parameters), all PPSH and CHIO parameters are initialized. For PPSH, the parameters are initialized as: $T = 1440$, PS and $PNS =$ the power required for each SA and NSA, which are taken from Tables 2 and 3, respectively, S and $NS =$ random number $\in \mathbb{Z}$ according to $OTPs$ and $OTPe$ which are initialized using Table 2, and $pc =$ values extracted from Equation (12). For CHIO parameters, $I = 100$, $HIP = 5$, $C_0 = 1$, $BR_r = 0.05$, and $Max_{Age} = 50$.

In the second step (Initialize CHIO-PSPSH population), PPSH’s solutions are generated. The population contains five solutions (HIS), and each solution contains five elements (number of appliances), as shown in Equation (35).

$$\begin{matrix}
 [Sol_1 & | & 570 & 1218 & 42 & 198 & 275] \\
 [Sol_2 & | & 596 & 1200 & 81 & 129 & 283] \\
 [Sol_3 & | & 597 & 1317 & 49 & 122 & 312] \\
 [Sol_4 & | & 556 & 1313 & 61 & 137 & 273] \\
 [Sol_5 & | & 574 & 1309 & 86 & 171 & 317]
 \end{matrix} \tag{35}$$

In step 3 (Calculate fitness function), the solutions are evaluated based on the objective function presented in Equation (22). After the evaluation, the solution with the fittest value is assigned as the best solution (highest immunity rate).

Steps four (Coronavirus herd immunity evolution) and five (Update herd immunity population) modify the cases and update the solutions. There are three main cases in CHIO, including normal/susceptible, infected, and immune/recovered. These cases are determined and updated iteratively for each solution according to Equation (34). As mentioned previously in this example, the number of iterations I is set to 100. Therefore, the CHIO will update the solutions and their status 100 times to reach its optimal solution if the conditions in Equation (34) are achieved. The solutions and their cases for the 100 iterations are presented in Table 8. As shown in the table, in the first iteration, the solution with the fittest value is assigned as the best solution, which is 0.4117, and the normal, infected, and recovered cases are assigned by 4, 1, and 0, respectively. In iteration 8, CHIO found a better solution with a better fitness value, and the three cases are still the same.

Furthermore, in iteration 14, the best solution is changed without changing the solutions cases. The solutions cases start changing in iteration 33, where the normal and infected solutions become 3 and 2, respectively. In iteration 39, the solutions cases are changed to 3, 1, and 1 for the normal, infected, and recovered solutions. A new solution is infected in iteration 58, and the best solution value becomes 0.4049. Another case of a solution is changed from infected to recovered in iteration 60, and the best solution is updated as well. The best solution is also updated in iteration 77. In iteration number 83, a solution dies, and a new solution is generated to update the best solution. The CHIO attempts to update the population is continued until iteration 100.

In the last step, the stop criterion is checked.

Table 8. Mapping between the components of PSPSH and CHIO.

Iteration	S1	S2	S3	S4	S5	FF	Normal	Infected	Recovered	Decision
1	570	1218	42	198	275	0.4117	4	1	0	Best solution
2	570	1218	42	198	275	0.4117	4	1	0	Do not replace best solution
5	570	1218	42	198	275	0.4117	4	1	0	Do not replace best solution
8	557	1266	12	180	195	0.4100	4	1	0	Replace best solution
14	557	1266	1	180	195	0.4055	4	1	0	Replace best solution
33	557	1266	1	180	195	0.4055	3	2	0	Do not replace best solution
39	557	1266	1	180	195	0.4055	3	1	1	Do not replace best solution
58	576	1298	6	187	241	0.4049	2	2	1	Replace best solution
60	567	1263	33	137	260	0.3897	2	1	2	Replace best solution
77	539	1213	10	168	260	0.3875	2	1	2	Replace best solution
83	541	1225	38	183	254	0.3813	3	0	2	Replace best solution
100	541	1225	38	183	254	0.3813	3	0	2	Replace best solution

5.4. Experimental Evaluation

In this section, the effect of CHIO on EB, PAR, WTR, and CPR reduction is evaluated and compared with other five popular metaheuristics, including GA, PSO, WDO, DE, and GWO.

5.4.1. Algorithms Effect on EB

The algorithm’s effect on EB value is presented and compared. In addition, the overall reduction is included in this comparison to present the best algorithm in reducing EB.

The average EB for seven scenarios is reduced to (55.6472), (56.1549), (57.5272), (57.7394), (56.8562), and (57.2460) using GA, PSO, WDO, DE, GWO, and CHIO, respectively. EBs obtained by these algorithms for the seven scenarios are compared in Table 9. The table shows a comparison between the algorithms in each scenario and their average EB.

Table 9. EB reduction comparison. Best values are shown in bold.

S #	GA	PSO	WDO	DE	GWO	CHIO
S #1	42.8801	42.9128	43.2542	43.8137	43.5041	43.8147
S #2	60.3058	62.3731	65.1472	64.9603	64.5597	64.6208
S #3	65.2185	65.1080	66.1703	67.4365	66.1138	66.4258
S #4	62.2197	62.7878	64.6019	64.2647	62.5916	63.3545
S #5	45.6463	45.7852	46.4160	47.5699	46.2879	46.7815
S #6	51.6958	51.9391	52.6767	52.7780	52.2998	52.6468
S #7	61.5644	62.1783	64.4241	63.3523	62.6367	63.0784
Average	55.6472	56.1549	57.527261	57.73940	56.85627	57.2460

Please note that the PSO outperforms all other algorithms in six scenarios, including scenario numbers 1, 2, 3, 5, 6, 7, in reducing EB, whereas GWO performs better in reducing EB in the fourth scenario. In addition, PSO achieves the best overall reduction, where it obtains a better average EB than the other algorithms.

5.4.2. Algorithms Effect on PAR

This section presents PAR reduction on the basis of the considered algorithms. The PAR values obtained by all compared algorithms for seven scenarios are presented in Table 10. The table presents the average PAR obtained by each algorithm to show which algorithm is better in reducing the overall PAR values. The average PAR value obtained by the compared algorithms are reduced to (2.3795), (2.3649), (2.5335), (2.3074), (2.3280), and (2.2672) using GA, PSO, WDO, DE, GWO, and CHIO, respectively.

Table 10. PAR reduction comparison. Best values are shown in bold.

S #	GA	PSO	WDO	DE	GWO	CHIO
S #1	2.6074	2.5943	2.6668	2.6044	2.6002	2.6044
S #2	2.4305	2.4037	2.8451	2.4451	2.4451	2.4323
S #3	2.3213	2.2409	2.2650	2.2310	2.2267	2.2181
S #4	2.2516	2.3363	2.5033	2.1096	2.2277	1.8744
S #5	2.2799	2.2458	2.2310	2.2310	2.2310	2.2203
S #6	2.5055	2.5055	2.6006	2.5233	2.5233	2.5175
S #7	2.2604	2.2275	2.6226	2.0071	2.0423	2.0039
Average	2.3795	2.3649	2.5335	2.3074	2.3280	2.2672

Notably, GA obtains the best PAR value in scenario number 6, whereas PSO outperforms the compared algorithms in the first and second scenarios. However, CHIO achieves the best PAR value in scenarios 3, 4, 5, and 7. In addition, CHIO performs better in reducing the overall PAR value, where it obtains the best average PAR value compared with the other algorithms.

5.4.3. Algorithms Effect on UC level

In this section, the effect of the compared algorithms on the basis of the two parameters of UC (i.e., WTR and CPR) is evaluated and compared to show their performance in improving the schedules in terms of UC enhancement.

Tables 11 and 12 present WTR and CPR values obtained by all compared algorithms for the seven scenarios. In addition, the average WTR and CPR for the seven scenarios are presented in the tables to investigate which algorithm can achieve a better overall reduction in terms of WTR and CPR.

Table 11. Comparison of WTR reduction. Best values are shown in bold.

S #	GA	PSO	WDO	DE	GWO	CHIO
S #1	0.3901	0.3523	0.0343	0.0150	0.0302	0.0094
S #2	0.3709	0.3659	0.0149	0.0046	0.0242	0.0124
S #3	0.3719	0.3673	0.0438	0.0281	0.0378	0.015
S #4	0.3900	0.4036	0.0200	0.0293	0.0579	0.0431
S #5	0.3677	0.3596	0.0405	0.0281	0.0396	0.0157
S #6	0.3890	0.3636	0.0253	0.0136	0.0268	0.0095
S #7	0.3853	0.3881	0.0092	0.0260	0.0587	0.0386
Average	0.3807	0.3715	0.0269	0.0207	0.0393	0.0205

Table 11 presents the CHIO’s robust performance when compared with the other algorithms in reducing WTR, where it obtains the best WTR value in four scenarios. In addition, CHIO achieves the best average reduction when compared with all considered algorithms. In terms of reducing CPR value, WDO obtains the best results, where it achieves the best CPR values in scenario numbers 2, 3, 4, 5, and 7. In addition, it obtains the best average CPR reduction.

Table 12. Comparison of CPR reduction. Best values are shown in bold.

S #	GA	PSO	WDO	DE	GWO	CHIO
S #1	0.3248	0.3246	0.3143	0.3122	0.3148	0.3127
S #2	0.3599	0.3535	0.3429	0.3449	0.3455	0.3453
S #3	0.3957	0.3959	0.3799	0.3811	0.3815	0.3803
S #4	0.5200	0.5121	0.4946	0.5044	0.5033	0.5066
S #5	0.3969	0.3956	0.3801	0.3811	0.3822	0.3806
S #6	0.3614	0.3594	0.3489	0.3481	0.3504	0.349
S #7	0.4806	0.4797	0.4492	0.4628	0.4688	0.4661
Average	0.4056	0.4030	0.3871	0.3907	0.3924	0.3915

CHIO obtains the highest UC level in scenario numbers 1, 3, 5, and 6. As shown in Table 13, WDO achieved the best UC level in the fourth and seventh scenarios, and DE outperforms other algorithms in the second scenario. In terms of average UC level, DE outperforms all compared algorithms, where it improves UC level by up to (18.74%), (18.15%), (0.13%), (1.01%), and (0.03%) when compared with GA, PSO, WDO, GWO, and CHIO, respectively. Notable, CHIO obtains the second best average UC level, where it outperforms all algorithms except the DE algorithm. These results prove CHIO's robust performance in improving UC level when compared with GA, PSO, WDO, and GWO.

Table 13. Comparison of UC_p reduction. Best values are shown in bold.

S #	GA	PSO	WDO	DE	GWO	CHIO
S #1	64.2510	66.1500	82.5642	83.6358	82.7463	83.895
S #2	63.4564	64.0215	82.1037	82.5222	81.5124	82.115
S #3	61.6160	61.8364	78.8101	79.5370	79.0296	80.235
S #4	54.4975	54.2085	74.2622	73.3118	71.9334	72.515
S #5	61.7660	62.2313	78.9602	79.5370	78.9061	80.185
S #6	62.4762	63.8421	81.2855	81.9100	81.1313	82.075
S #7	56.6993	56.6041	77.0780	75.5528	73.6213	74.765
Average	60.6803	61.2705	79.2948	79.4295	78.4115	79.3976

5.4.4. Performance and Statistical Evaluation

The obtained results show that some algorithms obtain good results in reducing EB but achieve the worst results in reducing WTR and CPR, as shown by GA and PSO results. In addition, CHIO outperforms all algorithms in reducing PAR and WTR, whereas DE achieves the best results in reducing CPR. Accordingly, finding the best overall reduction for all PPSH objectives by all algorithms is necessary to show the algorithm with the best performance in obtaining the best solution in terms of all objectives.

Table 14 shows the fitness values (FF) obtained by all algorithms with reduction for all objectives. The table proves that CHIO algorithm achieves high performance as it finds the best PPSH schedules and it achieves the best FF for all scenarios except the second scenario.

Figure 6 illustrates all algorithms performance using the convergence behaviour toward an optimal solution for the seven scenarios. The figure shows that CHIO converges towards the optimal solution better than the other algorithms in scenarios 1, 3, 4, 5, 6, and 7. This situation is due to CHIO's robust performance in balancing exploitation and exploration when finding the optimal solution. In addition, the figure shows that the convergence rate of CHIO is lower than that of the other algorithms for the same scenarios.

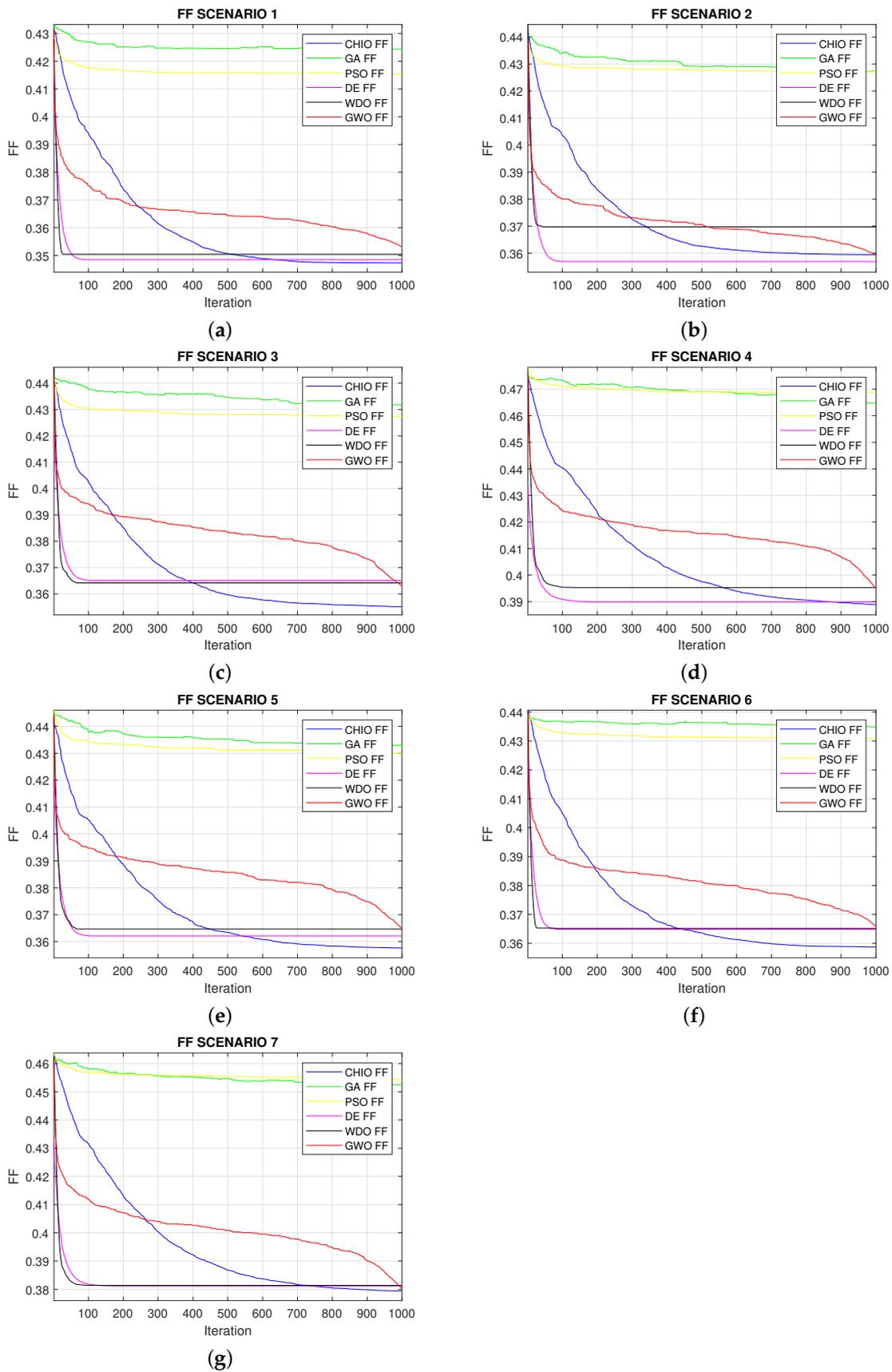


Figure 6. CB for the methods.

Table 14. Comparison of FF reduction. Best values are shown in bold.

S #	GA	PSO	WDO	DE	GWO	CHIO
S #1	0.4244	0.4153	0.3504	0.3484	0.3531	0.3473
S #2	0.4274	0.4270	0.3697	0.3569	0.3593	0.3598
S #3	0.4316	0.4273	0.3641	0.3651	0.3630	0.3551
S #4	0.4647	0.4686	0.3953	0.3898	0.3952	0.3888
S #5	0.4329	0.4300	0.3646	0.3620	0.3649	0.3576
S #6	0.4346	0.4308	0.3651	0.3648	0.3658	0.3586
S #7	0.4525	0.4542	0.3813	0.3811	0.3803	0.3793

A statistical comparison between the results obtained by all algorithms is presented in Table 15 to investigate any statistically significant difference that exists between the results obtained by CHIO and the compared algorithms. The CHIO’s results are compared with the other algorithms due to its encouraging results. As it achieved the best FF in six scenarios. The Wilcoxon signed-rank test is used for this statistical comparison. The p-value is used to estimate whether or not there is a significant difference between the results of the algorithms. A significant difference can be obtained if the p-value is lower than or equal to 0.05. Otherwise, the difference in the results is insignificant. Table 15 provides a statistical comparison based on the algorithms’ FF. The table indicates that CHIO considerably decreases FF when it is compared to the other algorithms, with a meaningful difference in all cases. Please note that the p-value is not presented for DE and GWO in the second scenario because these algorithms obtained better FF than CHIO in this scenario.

Table 15. Comparison of p-value.

S #	GA	PSO	WDO	DE	GWO
S #1	1.7343×10^{-6}	1.7343×10^{-6}	0.2802	0.7813	0.0333
S #2	1.7343×10^{-6}	1.7343×10^{-6}	0.0015	—	—
S #3	1.7343×10^{-6}	1.7343×10^{-6}	0.0020	3.5888×10^{-4}	4.1955×10^{-4}
S #4	1.7343×10^{-6}	1.7343×10^{-6}	0.0028	0.6583	0.0050
S #5	1.7343×10^{-6}	1.7343×10^{-6}	0.0185	0.0148	0.0026
S #6	1.7343×10^{-6}	1.7343×10^{-6}	0.0104	0.0125	0.0021
S #7	1.7343×10^{-6}	1.7343×10^{-6}	0.4908	0.5304	0.3743

6. Conclusions and Future Work

In power systems, scheduling the smart home appliances to proper operation periods concerning the dynamic pricing scheme(s) is known as PPSH. This scheduling problem is essential to control the power for supply companies and their users to reduce PAR and EB and to ensure UC level. PPSH is formulated as a constrained optimization problem with multi-objective features. Several optimization algorithms have been used to tackle PPSH. Due to their impressive characteristics, metaheuristic algorithms are successful in tackling PPSH. Quite recently, CHIO was established as a human-based optimization algorithm that imitates the herd immunity strategy to stop the spread of the COVID-19 disease. In this paper, CHIO is adapted to address PPSH (CHIO-PPSH) and achieve its objectives efficiently due to its ability in achieving the right balance of exploitation and exploration, thus finding the optimal/near-optimal solution(s).

The proposed CHIO-PPSH is examined using a dataset that contains 36 home appliances and seven scenarios. The main CHIO parameters, including RB_r and Max_{Age} are tuned to find their best values. The results prove CHIO’s robust performance when the values of its control parameters RB_r and Max_{Age} are 0.05 and 50, respectively. These best values are used in the evaluation of the proposed method by comparing its results with the other five metaheuristic algorithms, including GA, PSO, WDO, DE, and GWO. These methods are compared in terms of all PPSH objectives (i.e., EB, PAR, WTR, and CPR) to show the best method in optimizing the problem. Generally speaking, the proposed

CHIO shows the high performance when addressing PPSH, where it obtains the best PAR and WTR results compared with the other methods. In addition, it achieves the best results and convergence for the multi-objective function reduction. However, in terms of EB and CPR, PSO and WDO achieve the best values. In addition, DE yields the best UC improvement results. A statistical comparison between the results obtained by CHIO against the compared methods is conducted to detect any statistically significant difference between the obtained results. The statistical comparison proves the significant difference in the results obtained favoring CHIO.

Possible future directions can be considered to improve the performance of the proposed method and the quality of the PPSH solutions. The future directions are summarized as follows:

- As mentioned previously, CHIO contains two adjustable control parameters, including BR_r and Max_{Age} , where their values are changing from an optimization problem to another. These parameters are tuned in Section 5.2 based on the recommended values suggested by Al-Betar et al. [43] ($BR_r = 0.005, 0.05, 0.1, 0.5$, $Max_{Age} = 50, 100, 300, 500$) to find their best values for PPSH. However, such parameters best values for PPSH may not one of the suggested values, for instance, $BR_r = 0.001$ and $Max_{Age} = 30$. Accordingly, new tuning experiments for such control parameters can be conducted to investigate and find better best values. More experiments required a more illustrative presentation approach. Therefore, visualizing the new conducted results in a graph could enhance the presentation of the results.
- Due to the high constraints of the PPSH that restrict the optimization processes of the algorithms, new power sources, such as storage systems and renewable energy sources, can be integrated with the smart home components to enhance the solutions and schedules of PPSH.
- Due to the weak performance of CHIO in optimizing EB, CPR, and UC level, the proposed CHIO can be hybridized with components of other efficient methods to enhance its performance and achieve better solutions.

Author Contributions: Conceptualization, S.N.M.; Data curation, S.N.M. and M.A.A.; Formal analysis, S.N.M., M.A.A.-B., M.A.A., A.K.A., Z.A.A.A., I.A.D., O.A.A., R.D., A.Z. and M.A.M.; Funding acquisition, A.Z.; Investigation, S.N.M., M.A.A.-B., M.A.A., A.K.A., I.A.D. and O.A.A.; Methodology, S.N.M., M.A.A.-B. and M.A.A.; Software, S.N.M. and Z.A.A.A.; Supervision, M.A.M.; Validation, S.N.M., M.A.A.-B., M.A.A., A.K.A., Z.A.A.A., I.A.D., O.A.A. and R.D.; Visualization, S.N.M., M.A.A.-B., M.A.A., A.K.A., Z.A.A.A., I.A.D. and O.A.A.; Writing—original draft, S.N.M., M.A.A.-B., M.A.A., Z.A.A.A., I.A.D., O.A.A. and R.D.; Writing—review and editing, A.Z. and M.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

CHIO	Coronavirus Herd Immunity Optimizer
CPR	Capacity Power Limit Rate
ED	Differential Evolution
EB	Electricity Bill
FF	Fitness Values
GA	Genetic Algorithm
GWO	Grey Wolf Optimizer

HIP	Herd Immunity Population
IBR	Inclining Block Rates
NSA	Non-Shiftable Appliance
PAR	Peak-to-Average Ratio
PSO	Particle-Swarm Optimization
PSPSH	Power Scheduling Problem in Smart Home
RTP	Real-Time Price
SA	Shiftable Appliance
UC	User Comfort
WDO	Wind-Driven Optimization
WTR	Waiting Time Rate

References

- Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Abasi, A.K.; Alyasseri, Z.A.A. Optimization methods for power scheduling problems in smart home: Survey. *Renew. Sustain. Energy Rev.* **2019**, *115*, 109362. [CrossRef]
- Zhao, Z.; Lee, W.C.; Shin, Y.; Song, K.B. An optimal power scheduling method for demand response in home energy management system. *IEEE Trans. Smart Grid* **2013**, *4*, 1391–1400. [CrossRef]
- Yan, Y.; Qian, Y.; Sharif, H.; Tipper, D. A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE Commun. Surv. Tutorials* **2013**, *15*, 5–20. [CrossRef]
- Khan, A.R.; Mahmood, A.; Safdar, A.; Khan, Z.A.; Khan, N.A. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renew. Sustain. Energy Rev.* **2016**, *54*, 1311–1322. [CrossRef]
- Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S. An optimal power scheduling for smart home appliances with smart battery using grey wolf optimizer. In Proceedings of the 2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 23–25 November 2018; pp. 76–81.
- Iftikhar, H.; Asif, S.; Maroof, R.; Ambreen, K.; Khan, H.N.; Javaid, N. Biogeography Based Optimization for Home Energy Management in Smart Grid. In Proceedings of the International Conference on Network-Based Information Systems, Toronto, ON, Canada, 24–26 August 2017; pp. 177–190.
- Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Abasi, A.K.; Alyasseri, Z.A.A. A novel hybrid grey wolf optimizer with min-conflict algorithm for power scheduling problem in a smart home. *Swarm Evol. Comput.* **2021**, *60*, 100793. [CrossRef]
- Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S. Multi-objective power scheduling problem in smart homes using grey wolf optimiser. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3643–3667. [CrossRef]
- Makhadmeh, S.N.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Abasi, A.K.; Khader, A.T.; Damaševičius, R.; Mohammed, M.A.; Abdulkareem, K.H. Smart Home Battery for the Multi-Objective Power Scheduling Problem in a Smart Home Using Grey Wolf Optimizer. *Electronics* **2021**, *10*, 447. [CrossRef]
- Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Abasi, A.K. A min-conflict algorithm for power scheduling problem in a smart home using battery. In Proceedings of the 11th National Technical Seminar on Unmanned System Technology, Singapore, 2–3 December 2019; Springer: Singapore, 2020; pp. 489–501.
- Khurma, R.A.; Alsawalqah, H.; Aljarah, I.; Elaziz, M.A.; Damaševičius, R. An enhanced evolutionary software defect prediction method using island moth flame optimization. *Mathematics* **2021**, *9*, 1722. [CrossRef]
- Alyasseri, Z.A.A.; Khader, A.T.; Al-Betar, M.A.; Abasi, A.K.; Makhadmeh, S.N. EEG signals denoising using optimal wavelet transform hybridized with efficient metaheuristic methods. *IEEE Access* **2019**, *8*, 10584–10605. [CrossRef]
- Alyasseri, Z.A.A.; Khader, A.T.; Al-Betar, M.A.; Papa, J.P.; Alomari, O.A.; Makhadmeh, S.N. An efficient optimization technique of eeg decomposition for user authentication system. In Proceedings of the 2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS), Kuching, Malaysia, 24–26 July 2018; pp. 1–6.
- Alyasseri, Z.A.A.; Khadeer, A.T.; Al-Betar, M.A.; Abasi, A.; Makhadmeh, S.; Ali, N.S. The effects of EEG feature extraction using multi-wavelet decomposition for mental tasks classification. In Proceedings of the International Conference on Information and Communication Technology, Singapore, 16–18 August 2019; pp. 139–146.
- Xie, Q.; Cheng, G.; Zhang, X.; Lei, P. Feature selection using improved forest optimization algorithm. *Inf. Technol. Control* **2020**, *49*, 289–301. [CrossRef]
- Alrassas, A.M.; Al-Qaness, M.A.A.; Ewees, A.A.; Ren, S.; Elaziz, M.A.; Damaševičius, R.; Krilavičius, T. Optimized anfis model using aquila optimizer for oil production forecasting. *Processes* **2021**, *9*, 1194. [CrossRef]
- Alyasseri, Z.A.A.; Khader, A.T.; Al-Betar, M.A.; Papa, J.P.; Alomari, O.A.; Makhadmeh, S.N. Classification of eeg mental tasks using multi-objective flower pollination algorithm for person identification. *Int. J. Integr. Eng.* **2018**, *10*. Available online: <https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/3478> (accessed on 26 June 2021). [CrossRef]
- Alyasseri, Z.A.A.; Khader, A.T.; Al-Betar, M.A.; Abasi, A.K.; Makhadmeh, S.N. EEG signal denoising using hybridizing method between wavelet transform with genetic algorithm. In Proceedings of the 11th National Technical Seminar on Unmanned System Technology, Singapore, 2–3 December 2019; Springer: Singapore, 2020; pp. 449–469.

19. Helmi, A.M.; Al-Qaness, M.A.A.; Dahou, A.; Damaševičius, R.; Krilavičius, T.; Elaziz, M.A. A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors. *Entropy* **2021**, *23*, 1065. [CrossRef]
20. Altan, A.; Karasu, S.; Zio, E. A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. *Appl. Soft Comput.* **2021**, *100*, 106996. [CrossRef]
21. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. Link-based multi-verse optimizer for text documents clustering. *Appl. Soft Comput.* **2019**, *87*, 106002. [CrossRef]
22. Alrosan, A.; Alomoush, W.; Alswaitti, M.; Alissa, K.; Sahran, S.; Makhadmeh, S.N.; Alieyan, K. Automatic Data Clustering Based Mean Best Artificial Bee Colony Algorithm. *Cmc-Comput. Mater. Contin.* **2021**, *68*, 1575–1593. [CrossRef]
23. Alyasseri, Z.A.A.; Abasi, A.K.; Al-Betar, M.A.; Makhadmeh, S.N.; Papa, J.P.; Abdullah, S.; Khader, A.T. EEG-Based Person Identification Using Multi-Verse Optimizer as Unsupervised Clustering Techniques. In *Evolutionary Data Clustering: Algorithms and Applications*; Springer: Singapore, 2021; p. 89. Available online: https://www.google.com/books?hl=en&lr=&id=20wfEAAAQBAJ&oi=fnd&pg=PA88&dq=EG-Based+Person+Identification+Using+Multi-Verse+Optimizer+as+++Unsupervised+Clustering+Techniques.&ots=U329jhbpaf6&sig=KEDQzcxWM_jcun6McESW5PYF01w (accessed on 1 November 2021).
24. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. A Text Feature Selection Technique based on Binary Multi-Verse Optimizer for Text Clustering. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 1–6.
25. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. An improved text feature selection for clustering using binary grey wolf optimizer. In Proceedings of the 11th National Technical Seminar on Unmanned System Technology, Singapore, 2–3 December 2019; Springer: Singapore, 2021; pp. 503–516.
26. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Makhadmeh, S.N.; Al-laham, M.; Naim, S. A Hybrid Salp Swarm Algorithm with β -Hill Climbing Algorithm for Text Documents Clustering. In *Evolutionary Data Clustering: Algorithms and Applications*; Springer: Singapore, 2021; p. 129. Available online: <https://www.google.com/books?hl=en&lr=&id=20wfEAAAQBAJ&oi=fnd&pg=PA129&dq=A+Hybrid+Salp+Swarm+Algorithm+with+%24%5Cbeta%24-Hill+Climbing+Algorithm+++for+Text+Documents+Clustering.&ots=U329jhclec&sig=8XSWNweSeqR0I10YIPt4Y75Ic> (accessed on 1 November 2021).
27. Jouhari, H.; Lei, D.; Al-qaness, M.A.A.; Abd Elaziz, M.; Damaševičius, R.; Korytkowski, M.; Ewees, A.A. Modified Harris Hawks optimizer for solving machine scheduling problems. *Symmetry* **2020**, *12*, 1460. [CrossRef]
28. Ksiazek, K.; Połap, D.; Woźniak, M.; Damaševičius, R. Radiation heat transfer optimization by the use of modified ant lion optimizer. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017-Proceedings, Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–7.
29. Khan, D.M.; Aslam, T.; Akhtar, N.; Qadri, S.; Khan, N.A.; Rabbani, I.M.; Aslam, M. Black hole attack prevention in mobile ad hoc network (Manet) using ant colony optimization technique. *Inf. Technol. Control* **2020**, *49*, 308–319. [CrossRef]
30. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Makhadmeh, S.N. A novel hybrid multi-verse optimizer with K-means for text documents clustering. *Neural Comput. Appl.* **2020**, *32*, 17703–17729. [CrossRef]
31. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Makhadmeh, S.N. An ensemble topic extraction approach based on optimization clusters using hybrid multi-verse optimizer for scientific publications. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 2765–2801. [CrossRef]
32. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. A novel ensemble statistical topic extraction method for scientific publications based on optimization clustering. *Multimed. Tools Appl.* **2021**, *80*, 37–82. [CrossRef]
33. Wang, H.; Song, W.; Zio, E.; Kudreyko, A.; Zhang, Y. Remaining useful life prediction for Lithium-ion batteries using fractional Brownian motion and Fruit-fly Optimization Algorithm. *Meas. J. Int. Meas. Confed.* **2020**, *161*, 107904. [CrossRef]
34. Subramaniam, S.; Perumal, T. Statistical Markov model based natural inspired glowworm swarm multi-objective optimization for energy efficient data delivery in MANET. *Inf. Technol. Control* **2020**, *49*, 333–347.
35. Alrosan, A.; Alomoush, W.; Norwawi, N.; Alswaitti, M.; Makhadmeh, S.N. An improved artificial bee colony algorithm based on mean best-guided approach for continuous optimization problems and real brain MRI images segmentation. *Neural Comput. Appl.* **2020**, *33*, 1671–1697. [CrossRef]
36. Khan, M.A.; Sharif, M.; Akram, T.; Damaševičius, R.; Maskeliūnas, R. Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization. *Diagnostics* **2021**, *11*, 811. [CrossRef]
37. Alomari, O.A.; Makhadmeh, S.N.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Doush, I.A.; Abasi, A.K.; Awadallah, M.A.; Zitar, R.A. Gene selection for microarray data classification based on Gray Wolf Optimizer enhanced with TRIZ-inspired operators. *Knowl.-Based Syst.* **2021**, *223*, 107034. [CrossRef]
38. Alyasseri, Z.A.A.; Al-Betar, M.A.; Awadallah, M.A.; Makhadmeh, S.N.; Abasi, A.K.; Doush, I.A.; Alomari, O.A. A hybrid flower pollination with β -hill climbing algorithm for global optimization. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, in press. [CrossRef]
39. Salehan, A.; Deldari, A. Corona virus optimization (CVO): A novel optimization algorithm inspired from the Corona virus pandemic. *J. Supercomput.* **2021**, 1–32. [CrossRef]
40. Martínez-Álvarez, F.; Asencio-Cortés, G.; Torres, J.F.; Gutiérrez-Avilés, D.; Melgar-García, L.; Pérez-Chacón, R.; Rubio-Escudero, C.; Riquelme, J.C.; Troncoso, A. Coronavirus optimization algorithm: A bioinspired metaheuristic based on the COVID-19 propagation model. *Big Data* **2020**, *8*, 308–322. [CrossRef]
41. Li, Z.; Tam, V. A novel meta-heuristic optimization algorithm inspired by the spread of viruses. *arXiv* **2020**, arXiv:2006.06282.

42. Li, M.D.; Zhao, H.; Weng, X.W.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* **2016**, *92*, 65–88. [[CrossRef](#)]
43. Al-Betar, M.A.; Alyasseri, Z.A.A.; Awadallah, M.A.; Doush, I.A. Coronavirus herd immunity optimizer (CHIO). *Neural Comput. Appl.* **2020**, *33*, 5011–5042. [[CrossRef](#)] [[PubMed](#)]
44. Dalbah, L.M.; Al-Betar, M.A.; Awadallah, M.A.; Zitar, R.A. A modified coronavirus herd immunity optimizer for capacitated vehicle routing problem. *J. King Saud-Univ.-Comput. Inf. Sci.* **2021**, *in press*. [[CrossRef](#)]
45. Alweshah, M.; Alkhalailah, S.; Al-Betar, M.A.; Bakar, A.A. Coronavirus herd immunity optimizer with greedy crossover for feature selection in medical diagnosis. *Knowl.-Based Syst.* **2022**, *235*, 107629. [[CrossRef](#)] [[PubMed](#)]
46. Mahboob, A.S.; Shahhoseini, H.S.; Moghaddam, M.R.O.; Yousefi, S. A Coronavirus Herd Immunity Optimizer For Intrusion Detection System. In Proceedings of the 2021 29th Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 18–20 May 2021; pp. 579–585.
47. Neto, A.S.A. Optimization inspired on herd immunity applied to Non-hierarchical grouping of objects. *Rev. Inform. Teór. E Apl.* **2021**, *28*, 50–65. [[CrossRef](#)]
48. Aslam, S.; Iqbal, Z.; Javaid, N.; Khan, Z.A.; Aurangzeb, K.; Haider, S.I. Towards Efficient Energy Management of Smart Buildings Exploiting Heuristic Optimization with Real Time and Critical Peak Pricing Schemes. *Energies* **2017**, *10*, 2065. [[CrossRef](#)]
49. Soares, A.; Antunes, C.H.; Oliveira, C.; Gomes, Á. A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy* **2014**, *77*, 144–152. [[CrossRef](#)]
50. Rasheed, M.B.; Javaid, N.; Ahmad, A.; Khan, Z.A.; Qasim, U.; Alrajeh, N. An efficient power scheduling scheme for residential load management in smart homes. *Appl. Sci.* **2015**, *5*, 1134–1163. [[CrossRef](#)]
51. Ullah, I.; Hussain, I.; Singh, M. Exploiting grasshopper and cuckoo search bio-inspired optimization algorithms for industrial energy management system: Smart industries. *Electronics* **2020**, *9*, 105. [[CrossRef](#)]
52. Rahim, S.; Javaid, N.; Ahmad, A.; Khan, S.A.; Khan, Z.A.; Alrajeh, N.; Qasim, U. Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources. *Energy Build.* **2016**, *129*, 452–470. [[CrossRef](#)]
53. Muralitharan, K.; Sakthivel, R.; Shi, Y. Multiobjective optimization technique for demand side management with load balancing approach in smart grid. *Neurocomputing* **2016**, *177*, 110–119. [[CrossRef](#)]
54. Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Abasi, A.K. Particle swarm optimization algorithm for power scheduling problem using smart battery. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 672–677.
55. Khan, H.N.; Iftikhar, H.; Asif, S.; Maroof, R.; Ambreen, K.; Javaid, N. Demand side management using strawberry algorithm and bacterial foraging optimization algorithm in smart grid. In *International Conference on Network-Based Information Systems*; Springer: Cham, Switzerland, 2017; pp. 191–202.
56. Farina, M.; Amato, P. A fuzzy definition of “optimality” for many-criteria optimization problems. *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Hum.* **2004**, *34*, 315–326. [[CrossRef](#)]
57. López Jaimes, A.; Coello Coello, C.A. Some techniques to deal with many-objective problems. In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, Montreal, QC, Canada, 8–12 July 2009; pp. 2693–2696.
58. Farina, M.; Amato, P. Fuzzy optimality and evolutionary multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 58–72.
59. Marler, R.T.; Arora, J.S. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **2004**, *26*, 369–395. [[CrossRef](#)]
60. Fei, Z.; Li, B.; Yang, S.; Xing, C.; Chen, H.; Hanzo, L. A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 550–586. [[CrossRef](#)]
61. Cho, J.H.; Wang, Y.; Chen, R.; Chan, K.S.; Swami, A. A Survey on Modeling and Optimizing Multi-Objective Systems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1867–1901. [[CrossRef](#)]
62. Simon, D. *Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
63. Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Eng.* **2018**, *5*, 1502242. [[CrossRef](#)]
64. Lai, C.C.; Shih, T.P.; Ko, W.C.; Tang, H.J.; Hsueh, P.R. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and corona virus disease-2019 (COVID-19): The epidemic and the challenges. *Int. J. Antimicrob. Agents* **2020**, *55*, 105924. [[CrossRef](#)]
65. Kwok, K.O.; Lai, F.; Wei, W.I.; Wong, S.Y.S.; Tang, J.W. Herd immunity—estimating the level required to halt the COVID-19 epidemics in affected countries. *J. Infect.* **2020**, *80*, e32–e33. [[CrossRef](#)]
66. Remuzzi, A.; Remuzzi, G. COVID-19 and Italy: What next? *Lancet* **2020**, *395*, 1225–1228. [[CrossRef](#)]
67. Ribeiro, G.S.; Hamer, G.L.; Diallo, M.; Kitron, U.; Ko, A.I.; Weaver, S.C. Influence of herd immunity in the cyclical nature of arboviruses. *Curr. Opin. Virol.* **2020**, *40*, 1–10. [[CrossRef](#)]
68. Fine, P.E. Herd immunity: History, theory, practice. *Epidemiol. Rev.* **1993**, *15*, 265–302. [[CrossRef](#)]
69. Dalbah, L.M.; Al-Betar, M.A.; Awadallah, M.A.; Zitar, R.A. A coronavirus herd immunity optimization (chio) for travelling salesman problem. In *International Conference on Innovative Computing and Communications*; Springer: Singapore, 2022; pp. 717–729.
70. Kumar, C.; Magdalin Mary, D.; Gunasekar, T. MOCHIO: A novel Multi-Objective Coronavirus Herd Immunity Optimization algorithm for solving brushless direct current wheel motor design optimization problem. *Automatika* **2021**, *63*, 149–170. [[CrossRef](#)]

71. Naderipour, A.; Abdullah, A.; Marzbali, M.H.; Nowdeh, S.A. An improved corona-virus herd immunity optimizer algorithm for network reconfiguration based on fuzzy multi-criteria approach. *Expert Syst. Appl.* **2022**, *187*, 115914. [[CrossRef](#)] [[PubMed](#)]
72. Amini, S.; Ghasemi, S.; Golpira, H.; Anvari-Moghaddam, A. Coronavirus Herd Immunity Optimizer (CHIO) for Transmission Expansion Planning. In Proceedings of the 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Bari, Italy, 7–10 September 2021; pp. 1–6.
73. Central Maine Diesel Inc. *Generator Sales*; Central Maine Diesel Inc.: Hampden, ME, USA, 2017.
74. Ogwumike, C.; Short, M.; Abugchem, F. Heuristic optimization of consumer electricity costs using a generic cost model. *Energies* **2015**, *9*, 6. [[CrossRef](#)]
75. Forsati, R.; Mahdavi, M.; Shamsfard, M.; Meybodi, M.R. Efficient stochastic algorithms for document clustering. *Inf. Sci.* **2013**, *220*, 269–291. [[CrossRef](#)]