OPTIMISATION OF A SOLAR-ASSISTED COMBINED COOLING, HEATING AND POWER SYSTEM USING META-HEURISTIC APPROACHES.

By

UKAEGBU, UCHECHI FAITHFUL

219120451

Ph.D. Thesis submitted in fulfilment of the requirements for the degree

Doctor of Engineering: Mechanical Engineering

In the Faculty of Engineering & the Built Environment,

at the

University of Johannesburg, South Africa.

Supervisor: Prof. Lagouge Tartibu

Co-supervisor: Prof. C.W. Lim

August 6, 2024.

ABSTRACT

Solar-assisted combined cooling, heating, and power (CCHP) systems offer a sustainable solution to address the escalating energy demand resulting from economic and population growth. This advancement has facilitated a reduction in excessive fuel consumption, thereby mitigating the associated impacts of climate change and global warming. These multi-generation systems undeniably possess the advantage of fuel efficiency, leading to enhanced overall system efficiency compared to individual cooling, heating, and power generation systems. However, optimizing these systems is imperative to realize optimal system design configurations that enhance their thermodynamic performance indicators. Hence, this research study aims to apply multi-objective optimization techniques to CCHP systems.

This work explored the use of three multi-objective optimization approaches to optimize the performance of a solar-assisted combined cooling, heating, and power system. It aimed to obtain optimal configurations comprising decision variables such as compression ratio, pinch point temperature difference, gas turbine inlet temperature, and combustion chamber inlet temperature to maximize the net power and exergy efficiency while minimizing CO_2 emission. This study also investigated the effects of each decision variable on the objective function of the combined cooling, heating, and power (CCHP) system via sensitivity analysis, understand the convergence trend of each decision variable, and evaluate its results with that obtained from related works.

This was carried out by first defining and initializing the objective functions, upper and lower boundaries of the decision variables as well as other relevant parameters. The randomly selected candidate solutions by the respective optimization algorithm were iteratively improved using techniques such as leader election, leader-prey, dynamic update, pack-hunting approaches, etc. Furthermore, evaluation processes were carried out by computing the objective values and the roulette wheel was predominantly used to select the new sets of solutions according to their fitness values. The optimization was carried out until the termination criteria which is the maximum number of iterations was attained and the results were analyzed with Pareto fronts presented to provide insight to the decision maker.

The grey wolf, Harris Hawks, and Antlion optimization techniques each generated 100 Pareto solutions, representing optimal trade-offs among conflicting objective functions. These solutions

offer decision-makers a diverse array of options. Analysis of the Pareto fronts revealed that minimizing CO_2 emissions contributes significantly to system efficiency while maximizing net power output does not necessarily result in higher efficiency. Sensitivity and convergence analyses indicated that optimizing a solar-assisted CCHP system involves minimizing compression ratio, pinch point temperature difference, and inlet combustion chamber temperature while maximizing gas turbine inlet temperature. Additionally, the compression ratio most significantly impacts CO_2 emissions, gas turbine inlet temperature affects exergy efficiency the most, and inlet combustion chamber temperature has the greatest influence on net power output.

Another set of six optimal solutions was generated by each optimization technique for validation purposes. The results obtained revealed that the Antlion optimization technique obtained an optimal configuration that yielded the best CO_2 emission and exergy efficiency of 49.40 gr/MJ and 45.45 % respectively. It also had the best runtime value of 3.02 seconds but produced a relatively lower net power output compared to the other methods. Also, the Harris Hawks had an optimal configuration that yielded the best net power output of 61.91 MW, but it also produced relatively higher CO_2 emission and exergy efficiency. The grey wolf technique, on the other hand, attempted to find the right balance between net power and exergy efficiency that had conflicting objectives. This is because, in contrast to the other optimization methods, it simultaneously achieved a relatively high exergy efficiency and net power output. However, it had the highest run-time value of 145.74 seconds. Furthermore, it was observed that the three meta-heuristic optimization techniques employed in this study produced, on average, reduced CO_2 emission, and high exergy efficiency, with negligible reduction in the net power (standard deviation of 0.39) when compared to the response surface method used for evaluation purposes. Hence, the optimization approaches presented in this study are all suitable for the multi-objective optimization of a solar-assisted CCHP system.

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to my supervisor and co-supervisor in the persons of Prof. Tartibu Lagouge and Prof. C.W. Lim for their expert guidance, encouragements, academic and moral support during my Ph.D. research study. Indeed, their supervision style provided me with the with the much-needed motivation to complete my research study despite the numerous challenges encountered.

I remain ever grateful to my spiritual father and mentor, the Most Rev. Prof. Daddy Hezekiah for his unwavering sacrifices and fervent prayers that inspire me to keep moving forward. I also thank my ever-loving parents, Engr. & Mrs. H.N. Ukaegbu for their unending support and encouragement all through my academic journey. I sincerely appreciate my siblings, Amara, Nenye, and Victor; my friends, Chinyere Ebom. I would not forget to thank Mr. & Mrs. Nnamdi Iwenofu for their tremendous support to me.

Finally, I would like to thank the Department of Mechanical and Industrial Engineering, University of South Africa for the wonderful platform to conduct this research study. I also want to thank the University Research Council for availing me of the Commonwealth scholarship as their financial contributions helped in the realization of this research work.

DEDICATION

This thesis is dedicated to God Almighty, the source of my strength and close confidant; and my parents, Engr. And Mrs. H.N Ukaegbu.

Table of contents

List of figures

List of Tables

Nomenclature

Symbol	Parameter
GT	Giga tonnes
CO_2	Carbon IV oxide
kW	Kilowatts
MW	Megawatts
$f_b(x)$	Objective functions
X	Design/search space
x_i	Decision variables
$\dot{P}_{net}$	net power output
$\dot{P}_{net,GT}$	net power from the gas turbines
$\dot{P}_{net,KC}$	net power from the kalina cycle
$\dot{P}_{GT-1}$	net power from gas turbine 1
$\dot{P}_{GT-2}$	net power from gas turbine 2
$\dot{P}_{GT-3}$	net power from gas turbine 3
$\dot{P}_{AC-1}$	net power to compressor 1
$\dot{P}_{AC-2}$	net power to compressor 2
$\dot{P}_{ST}$	net power from steam turbine
$\dot{P}_{pump-2}$	net power from pump 2 of the Kalina cycle

$\dot{m}_p$	mass flow rate of the combustion gases from the combustion chamber
$\dot{E}_{in}$	Input exergy
$\dot{E}_{fuel}$	Exergy of fuel
$\dot{E}_{coll}$	Exergy of solar collector
$\dot{Q}_{heating}$	Heating load
$\dot{Q}_{cooling}$	Cooling load
Cr	Compression ratio
Pp	Pinch point temperature difference
Gt	Inlet turbine temperature
Ct	Inlet combustion chamber temperature
$\mathcal{E}$	Exergy efficiency
$\vec{H}$	Coefficient vector
$\vec{N}$	Position vector
$\vec{N}_p$	Position vector of the prey
A	flight energy of the prey
A_0	initial flight energy of the prey
$Y(r+1)$	new position of the hawks based on r repetitions
Y_n	average position of the current hawk’s population
$Y(r)$	current position of the hawk
$\mathcal{J}$	prey’s random leap strength
LF	levy flight function

M_{Ant}	matrix for saving the location of each ant/prey
M_{OA}	matrix for storing the fitness value of each ant/prey
$M_{Antlion}$	matrix for saving the location of each ant lion
c^t	minimum of all variables at t^{th} iteration
d^t	maximum of all variables at t^{th} iteration
c_i^t	minimum of all variables at i^{th} ant
d_i^t	maximum of all variables at i^{th} ant
$Antlion_j^t$	position of the selected j^{th} ant lion at the i^{th} iteration
P_A^t	random walk around the ant lion chosen by the roulette wheel at the t^{th} iteration
P_E^t	random walk around the elite at the t^{th} iteration
Ant_i^t	position of the i^{th} ant at the t^{th} iteration

Chapter One – Introduction

1.1 Research background

Energy is pivotal to the economic growth of any country and its increased demand/production in recent times has led the extreme usage of fossil fuels such as petroleum, natural gas, coal, etc. The utilization of fossil fuels, as prime sources, also come with its drawback especially in the area of global warming due to greenhouse gas emission, its cost-intensiveness and depletion of its reserves (Nasrin et al. 2018; Wang, et al. 2020). This creates the niche for efficient systems to manage and improve energy conservation as well as renewable energy sources that could complement or replace fossil fuels.

Multi-generation systems are born out of the need for more sustainable generation of one or more forms of energy as output. They are mainly categorized, according to the number of outputs, into co-generation and tri-generation systems. Co-generation, also known as combined heat and power (CHP) system, involves the use of a power-generating unit to produce both power and thermal energy simultaneously. In conventional separate power generation systems, the heat energy would have been dissipated immediately to the atmosphere but is recovered in CHP systems to provide heating power thus enabling savings in fuel and GHG emission reduction (Ortiga, Bruno & Coronas, 2013). Irrespective of the fact that CHP systems have glaring advantages over conventional separate heat and power generation systems, they are susceptible to fluctuations in their system's overall efficiency especially during summer seasons when heating loads are at a minimum (Lozano, Carvalho, & Serra, 2011). A stable and constant power and thermal energy demand profile is essential to achieve maximum overall system efficiency all year round. Therefore, it is ideal to incorporate cooling systems into a CHP system to boost its efficiency hence the concept of combined cooling, heating, and power (CCHP) systems.

An energy management system that has gained research interest due to its fuel efficiency and reduced greenhouse emission rate is the combined cooling and heating power (CCHP) system. Tri-generation systems, also known as CCHP systems, are an extension of the CHP systems where electricity, cooling, and heating power are generated from a single energy source. It typically comprises a power generation unit/prime mover, cooling system, and heat recovery systems (Wang, et al. 2016). The prime mover generates mechanical power which is used to rotate the shaft of an electric motor for the generation of electricity. The CCHP involves the integration of various thermodynamic systems to produce two or more forms of energy in such a way that a ‘top system’ can be employed to drive a ‘bottom system’. A ‘top system’ in this context refers to systems such as gas turbines that require high degree of energy for their operation while a ‘bottom system’ such as the organic rankine, cycle, kalian cycle absorption chiller, etc., require a lower amount of energy (Mahdavi, Mojaver & Khalilarya, 2022). In other words, Thermal energy produced during power generation is used to drive the heating and cooling systems such as the absorption chiller, and heat recovery generation unit, and/or produce more power output using the organic Rankine or Kalina cycle. According to Wu & Wang (2006) who performed a comparative analysis between a conventional energy system and the CCHP system, it was established that the efficiency improved by about 33% owing to the cascade energy application of the CCHP system.

The integration of renewable energy sources either as an adjunct or replacement to fossil fuels is another talked about energy management system. According to the BP Energy Outlook (2023), wind and solar power would account for about two-thirds of the global power generation by 2050 and their rapid adoption would be fuelled by a fall in their costs. The solar energy source is predominately employed in CCHP systems, not necessarily as a standalone energy source due to its variability and the volatility of its radiation, to reduce the amount of fossil fuel consumption. In CCHP systems, the solar energy is converted to thermal energy via the solar thermal collectors which are either concentrating (such as the parabolic trough collector, linear Fresnel and compound parabolic concentrator) or non-concentrating (such as the flat plate and evacuated tube collector). Several pieces of literature have discussed the CCHP systems integrated with solar energy for multiple applications. The efficacy of a solar energy integrated CCHP over one powered by an internal combustion engine was confirmed by Yousefi, Ghodusinejad & Kasaeian (2017) who configured a solar-assisted CCHP system as it comprised the prime mover (internal combustion engine), thermal solar collector, absorption chiller, electric chiller and a boiler. Similarly, Zhang et al. (2018) proposed a hybrid CCHP system that yielded a 30.4% fuel saving with a 26% solar energy input.

The benefits of the solar-assisted CCHP systems cannot be over-emphasized, however their performance and design criteria can be improved through diverse optimization techniques. The optimization domain has applied several use cases in solving the challenge associated with rapid advancement in industrial technology (Nadimi-Shahraki et. al, 2022). According to Hansen, Muller & Koumoutsakos (2003), there are generally two types of optimizations: deterministic and metaheuristic algorithms. In the deterministic/heuristic approach, the gradient of the objective function is required, and a constant output is obtained for a given input. The metaheuristic, on the other hand, can solve several types of optimization problems by tuning its parameters and it does not yield a constant output for a given input. Two common types of metaheuristic optimization are the evolutionary and swarm optimization techniques (Abualigah, 2020). The evolutionary algorithm is guided by the Charles Darwin notion of mutation, crossover, and survival of the fittest while the swarm algorithm attempts to replicate the foraging/hunting behavior of a particular set of animals in a bid to accomplish a particular goal. The genetic algorithm, genetic programming, and differential evolution are valid examples of evolutionary algorithms. Prominent examples of swarm-intelligence algorithms are the Harris hawk, grey wolf, moth flame, ant colony, ant-lion optimization, etc.

Mirjalili, Jangir & Saremi (2017) defined optimization as the science of solving real problems by obtaining optimal design solutions through the use of machines/softwares. Real engineering problems are typically multi-objective in nature, and this implies that they require the optimization of more than one objective (Mirjalili et al.2016). Multi-objective functions are solved by arbitrarily assigning weights in a weighted-sum problem formulation and was employed by Zeng et al. (2015) and Song et al. (2020) to effectively improve the objectives of a CCHP system. The weight-based optimization or a priori method however has the drawback of requiring multiple runs and the need to always seek counsel from an expert/decision maker (Kim & De Weck, 2005). They can also be solved using the posteriori method that involves retaining the multi-objective formulation and obtaining the Pareto optimal solutions in a single run. There are quite a handful of optimization techniques in literature and they include the response surface method (Hasanzadeh et al 2022), Non-dominated sorting genetic algorithm-II (Ren, et al. 2019), Particle swarm optimization (Azaza & Wallin, 2017), Harris hawk optimization (Shehab et al 2022), Grasshopper optimization (Sharifian & Abdi , 2022), Ant-lion optimization (Ji et al 2022), Moth flame optimization (Xu et al 2023), and Greywolf optimization (Chen, Huang & Shahabi, 2021), etc. techniques. Ren, Wei & Zhai, (2021) employed the non-dominated sorting genetic algorithm to generate a Pareto front solution that translated to improved exergy efficiency and reduced greenhouse gas emission. Also, the greywolf optimization technique was used by Asgari et al. (2022) for the multi-objective optimization of net power, exergy efficiency and unit cost in a solar-based CCHP system.

Studies by Wang, Han & Guan, (2020) revealed that there are typical evaluation criteria that informs the optimization of hybrid CCHP systems, and they are the exergetic, economic and environmental factors. The exergetic factors comprise the exergy efficiency, energy efficiency, primary energy saving ratio, etc. The economic factors include, the product unit cost, total cost saving, net present value, etc. while the environmental factors are CO_2 emission and integrated performance.

Given the previous insights, the proposed research aims to utilize the grey wolf, Harris hawk, and antlion optimization techniques for multi-objective optimization. Specifically, the research focuses on optimizing net power output, exergy efficiency, and CO_2 emissions in a solar-assisted CCHP system.

1.2 Problem statement

Increasing population and economic growth are critical factors that are directly proportional to the global energy demand and consumption rate. These have adverse climate and health implications due to the emission of greenhouse gases among others. Environmental deterioration caused by these greenhouse gas emissions from power plants are a significant threat to the society influenced by the consequences of global warming. According to the IEA 2022 report, as shown in fig 1.1, CO_2 emission increased by 0.9% in 2022 peaking to an all-time high value of about 36.8GT. Also, Spahni et al. (2005) reported that electricity generation accounts for about 32% of CO_2 emissions followed by heating and cooling sources which accounts for 33% while transportation media accounts for 35% of the global CO_2 emission. This goes to prove that about 65% of CO_2 emission are due to power generation, heating, and cooling and these are necessities for human survival.

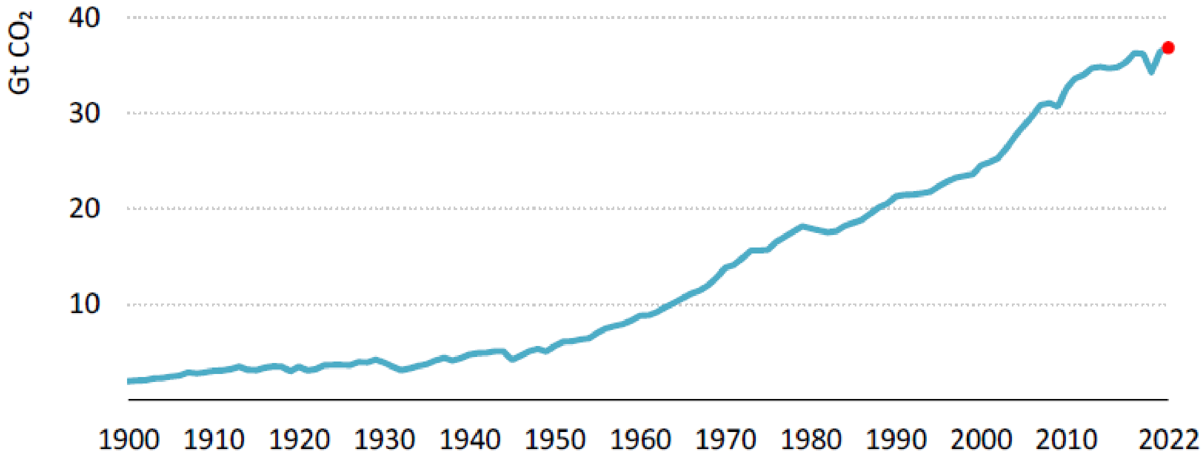


Figure 1.1: CO_2 emissions from energy combustion and industrial processes

While Combined Cooling, Heating, and Power (CCHP) systems offer a sustainable approach to energy conservation by reducing greenhouse emissions, heat loss, and operational costs, while simultaneously improving overall energy efficiency and reliability (Dincer & Zamfirescu, 2012), the pursuit of more optimal thermodynamic performance indicators remains an ongoing research endeavor. Optimization serves as a fundamental tool in engineering, allowing for the identification of the best decision variables to determine the optimal configuration, features, and operation of a given system. Hence, the proposed research study utilizes three meta-heuristic optimization techniques to maximize net power output and exergy efficiency while minimizing CO_2 emissions in a CCHP.

1.3 Significance of study

This research aims to maximize the net power output, exergy efficiency, and minimize CO_2 emissions in a solar-based Combined Cooling, Heating, and Power (CCHP) system. It employs decision variables such as compression ratio, pinch point temperature difference, and inlet turbine and combustion chamber temperatures to enhance overall system efficiency and reduce operational costs. Additionally, the study seeks to mitigate greenhouse gas emissions, addressing their adverse health effects and contribution to global warming. By optimizing a solar-assisted CCHP system, the research aligns with efforts to reduce fossil fuel depletion rates. Notably, while previous studies have explored other optimization techniques for solar-based CCHP systems, this research introduces novel nature-inspired optimization approaches. The insights gained from the analysis and results are anticipated to inform decision-making processes in the realm of combined cooling, heating, and power generation systems.

1.4 Rationale and motivation of study

The motivation behind this study stems from the escalating greenhouse gas emissions, which pose significant threats to the ecosystem. Additionally, there is a pressing need for an optimized solar-based Combined Cooling, Heating, and Power (CCHP) system to enhance energy management effectively.

This study builds upon recent research conducted by Mahdavi, Mojaver & Khalilarya (2022), who utilized the Response Surface Method (RSM) to perform multi-objective optimization of exergy efficiency, net power, and CO_2 emissions. Their methodology yielded six optimal solutions through interaction effects among decision variables, with the best solution selected using the TOPSIS (technique for order preference by similarity to an ideal solution) method. The RSM involves developing a mathematical model to establish correlations between input variables and the objective function. Therefore, this work employed it to create a regression model capable of estimating net power output, exergy efficiency, and CO_2 emissions. This model serves as a foundation for the proposed research study.

This research illustrates the use of existing metaheuristics approaches, proposed by Mirjalili, (2023a), Mirjalili, (2023b), Heidari, (2019), Eshkevari, (2023), Mirjalili, (2023c) and Mirjalili, (2023d), for the problem formulation, optimization, and analysis related to the CCHP system. This study has addressed the optimization of a solar-assisted CCHP system. The performance metrics used to measure the efficiency of the system, when optimized individually, showed conflict in objective values hence a multi-objective approach was required. Therefore, three different meta-heuristic approaches were proposed to develop a mathematical programming model on MATLAB to optimize the CCHP system.

1.5 Research objectives

The proposed research study aims to optimize the net power output, exergy efficiency and CO_2 emission of a solar-assisted CCHP system using the Greywolf, Harris hawk and Ant-lion optimization techniques. The following objectives are geared towards achieving the above-mentioned aim.

1. To formulate mathematical programming models using Greywolf, Harris hawk, and ant-lion optimizers.
2. To maximize the net power and exergy efficiency while minimizing the CO_2 emission using the Greywolf, Harris hawk, and ant-lion using a single and multi-objective optimization approach.
3. To investigate, through a sensitivity analysis, the effects of the compression ratio, pinch point temperature, inlet combustion chamber temperature, and the inlet turbine temperature on the optimal net power, exergy efficiency, and CO_2 emission.

1.6 Research questions

The questions this research sets out to answer are as follows:

1. What optimal values of decision variables (compression ratio, pinch point temperature, inlet combustion chamber temperature, and the inlet turbine temperature) would effectively maximize the net power output and exergy efficiency while minimizing the CO_2 emission?
2. What are the effects of the decision variables on the optimal performance criteria?
3. How to formulate and solve a problem related to the CCHP system that involves conflicting objective functions?

1.7 Proposed research methodology

This section shows the procedures set out to actualize and successfully complete the research study. Although chapter three would provide a detailed methodology of the study, a summary is provided herein.

1. Problem definition and algorithm selection: The problem addressed in this study is the optimization of a solar-assisted Combined Cooling, Heating, and Power (CCHP) system using the Grey Wolf, Harris Hawks, and Ant Lion optimizers. Objectives include maximizing net power output, exergy efficiency, and minimizing CO_2 emissions while considering compression ratio, pinch point temperature difference, inlet combustion chamber, and inlet turbine temperatures. These metaheuristic algorithms were selected for their ability to handle complex optimization problems with multiple objectives and constraints and their capacity to meta-heuristically obtain optimal solutions within relatively quick computation times.
2. Initialization of solutions: The population of potential solutions was randomly initialized for each optimization algorithm with diverse set of feasible configurations for the CCHP system. The initialized solutions covered a wide range of the solution space while confined within the specified constraints.
3. Iterative improvement: The candidate solutions were iteratively improved by applying some operators or mechanisms with the view of maintaining a balance between exploration and exploitation. For the selected meta-heuristic algorithms, techniques such as leader selection, leader-prey, dynamic update, and pack-hunting approaches were employed.
4. Evaluation: The quality of each potential/candidate solution was evaluated using the defined objective functions. This was done by computing the objective value (net power

output, exergy efficiency, and CO_2 emission) from the obtained solutions to ascertain their effectiveness in solving the optimization problem.

5. Selection: This involved employing probabilistic selection mechanisms, especially the roulette wheel selection to choose new sets of solutions, based on their fitness values, during each iteration. The roulette wheel selection method was chosen due to the level of randomness it introduced to prevent arriving at a sub-optimal solution. Also, it was chosen due to its approach of assigning probability based on fitness values.
6. Termination Criterion: The termination criteria were defined to determine the point at which the optimization algorithm comes to a stop. For the selected metaheuristic methods, the maximum number of iterations was set as the termination criteria.
7. Post-processing and analysis: This was carried out post-optimization to analyze the obtained solutions that maximize net power output, and exergy efficiency and minimize CO_2 emissions. Pareto graphs were also presented to provide insight to the decision-maker in order to make informed decisions needed to optimize the solar-assisted CCHP system. Convergence analysis of the decision variables as well as sensitivity analysis were conducted to better understand the dynamics around the decision variables and the multi-objective optimization outcomes.
8. Documentation and reporting: The methodology, implementation details, and optimization results were reported in this thesis. Also, the research findings, conclusions, and insights drawn from the optimization process as well as limitations and future direction were duly documented.

1.8 Delimitation and limitation

This study is concerned with the single and multi-objective optimization of the net power output, exergy efficiency and CO_2 emission objective functions using the greywolf, harris hawk and ant-lion techniques. The effects of the decision variables on the objective functions would be analysed via a sensitivity analysis. This study does not focus on the development of objective functions, or the validation of optimal solutions reported.

1.9 Main contributions of the research

The main contributions of this research include:

1. Developing a mathematical programming model that describes the optimization of a CCHP systems with MATLAB.
2. Performing single and multi-objective optimization to compute the best configuration of the CCHP system that enhanced its performance.
3. Ascertaining the dynamics between the decision variables and multi-objective outcomes via sensitivity and co-analysis.

With the above contributions, the following publications were made:

1. Ukaegbu, U., Tartibu, L. and Lim, C.W., 2023. Multi-Objective Optimization of a Solar-Assisted Combined Cooling, Heating and Power Generation System Using the Greywolf Optimizer. *Algorithms*, *16*(10), p.463.
2. Ukaegbu, U., Tartibu, L. and Lim, C.W., 2024. Multi-Objective Optimization of a Solar-Assisted Combined Cooling, Heating and Power Generation System Using the Harris Hawks Optimizer- Submitted to Results in Engineering journal.
3. Ukaegbu, U., Tartibu, L. and Lim, C.W., 2024. Multi-Objective Optimization of a Solar-Assisted Combined Cooling, Heating and Power Generation System Using the Ant-lion Optimizer- Submitted to Results in Engineering journal.
4. Ukaegbu, U., Tartibu, L. and Lim, C.W., 2024. Multi-Objective Optimization of a Solar-Assisted Combined Cooling, Heating and Power Generation System Using the Ant-lion Genetic Algorithm Optimizer- Submitted to Results in Engineering journal

1.10 Research structure

This thesis is structured into six chapters to comprehensively describe the research study undertaken.

- Chapter two was focused on describing the theoretical concepts of a multi-generation system, exploring the classification and sub-systems of the CCHP system. Thereafter, the theory behind the application of solar energy in the CCHP system was described as well as related works employing this renewable energy type. Subsequently, the concept of optimization was introduced and related works employing the MOGWO, MOHHO, and MOALO in multi-generation systems were presented.
- In chapter three the system to be optimized was described as well as the decision variables, constraints, and objective functions. The three meta-heuristic optimization techniques were

extensively discussed to understand the governing concepts behind their development and implementation.

- Chapter four presented the multi-objective optimization of the solar-powered CCHP system under study using the greywolf optimization technique. The results obtained as well as the findings observed were reported.
- Chapter five presented the multi-objective optimization of the solar-powered CCHP system under study using the harris hawks optimization technique. The results obtained as well as the findings observed were reported.
- Chapter six presented the multi-objective optimization of the solar-powered CCHP system under study using the ant-lion optimization technique. The results obtained as well as the findings observed were reported.
- Chapter seven provided a general summary and conclusion of the research study, analyzing the strengths and weaknesses of the different optimization techniques employed.

Chapter Two-Literature Review

2.1 Introduction

This chapter aims to comprehensively review existing literature to facilitate a better understanding of the concepts and terminologies employed in this research study. It begins by elucidating the theoretical concepts of multi-generation systems, with a focus on Combined Cooling, Heating, and Power (CCHP) systems, exploring their classification and subsystems. Subsequently, the theory behind the integration of solar energy into CCHP systems is discussed, along with an overview of related works that have employed this renewable energy source.

Following this, the concept of optimization is introduced, emphasizing its significance in enhancing the performance of multi-generation systems. Related works employing Metaheuristic Grey Wolf Optimizer (MOGWO), Metaheuristic Harris Hawks Optimizer (MOHHO), and Metaheuristic Ant Lion Optimizer (MOALO) in multi-generation systems are highlighted. This includes a review of studies that have utilized these optimization algorithms to improve the efficiency, reliability, and sustainability of CCHP systems, with a particular focus on solar-assisted configurations.

2.2 Multi-generation systems

These are systems that are borne out of the need for a more sustainable generation of one or more forms of energy as output. They are mainly categorized, according to the number of outputs, into two, co-generation and tri-generation systems.

2.2.1 Co-generation systems

Co-generation, also known as combined heat and power (CHP) system, involves the use of a power-generating unit to produce both power and thermal energy simultaneously. Hence, two energy forms are generated from a single energy input/source. In conventional separate power generation systems, the heat energy would have been immediately dissipated to the atmosphere. In co-generation systems, however, they are recovered to provide heating power thus providing savings in fuel and also reducing the emission of greenhouse gases (Ortiga, Bruno & Coronas, 2013; Huicochea et al. 2011). These systems originated in the 1880s due to the active use of steam turbine generators and coal-fired boilers for electricity production in industrial plants (Díaz, Benito & Parise 2010).

Even though co-generation systems have obvious advantages over separate heat and power generation systems, they are susceptible to fluctuations in their system's overall efficiency especially during summer seasons when heating loads are at a minimum (Lozano, Carvalho & Serra, 2011). Hence, a stable and constant power and thermal energy demand profile is essential to achieve maximum overall system efficiency all year round. Therefore, it was ideal to incorporate cooling systems into a CHP system to boost its efficiency hence the concept of the combined cooling, heating, and power (CCHP) systems (Hernández-Santoyo & Sánchez-Cifuentes, 2003).

2.2.2 Tri-generation systems

Tri-generation systems, also known as combined cooling, heating, and power systems, are an extension of the CHP systems whereby electricity, cooling and heating power are generated from a single energy source. As shown in Figure 2.1, it typically comprises a power generation unit/prime mover, cooling system and heat recovery systems (Chen et al. 2014, Wang et al. 2016). The prime mover generates mechanical power which is used to rotate the shaft of an electric motor for the generation of electricity. Thermal energy produced during power generation is used to drive the heating and cooling systems such as the absorption chiller, and heat recovery generation unit, and/or produce more power output using the organic Rankine or Kalina cycle.

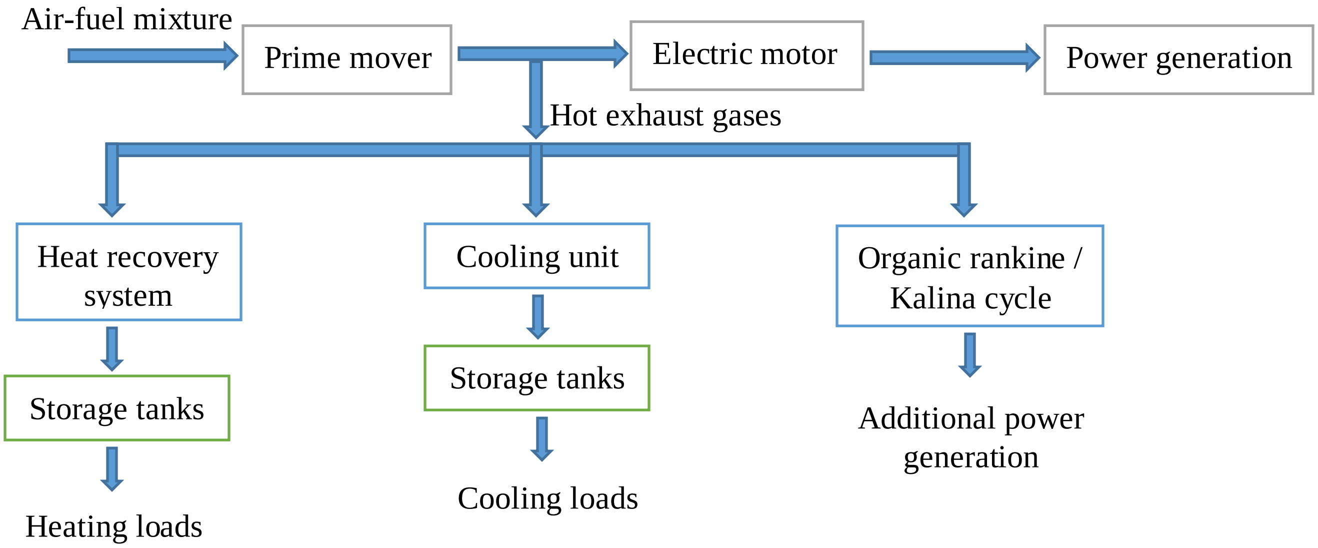


Fig 2.1: Typical CCHP system

2.3 Classification of CCHP systems

According to Wu & Wang (2006), CCHP systems can be classified based on three factors which are:

I. Type of prime mover employed.
II. The order in which the generated energy is used.
III. The industrial plant size.

I. **Classification according to the type of prime mover employed**: Prime movers are a fundamental part of the CCHP system and its selection according to Liu, Shi & Fang (2014) depends on the financial constraints, the size of the system, and available local resources. The different types of prime movers are (Wu & Wang 2006; Rosato, Sibilio & Ciampi, 2013): Stirling engine, gas turbine engines, combined cycle gas turbine engines, reciprocating internal combustion engines, microturbines, organic rankine cycle, steam turbine engines and fuel cells.

II. **Classification according to the energy generation order**: CCHP systems can be classified based on the order in which energy is used or the operational method employed as shown in fig 2.2 and 2.3. Based on this category, CCHP systems can wither be a topping or bottoming cycle (Ebrahimi, & Keshavarz,2012; Al-Sulaiman, Dincer & Hamdullahpur, 2012; Rosato, Sibilio & Ciampi, 2013; Oland, 2004).

    i. **Topping cycle**: In this system type, power generation is the primary concern in the sense that the air-fuel mixture is first of all used to generate electrical power. Subsequently, heat energy is produced as a secondary product to meet the heating and cooling loads of the CCHP system. This approach is predominantly utilized in multi-generation systems.

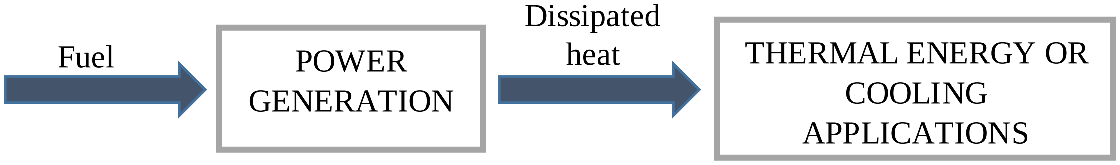


Fig 2.2: Schematic for a topping cycle

ii. **Bottoming cycle**: In this system type, thermal energy production is the priority concern of the CCHP system and afterward the dissipated heat is recuperated to produce the power output. Considering the fact that high-temperature gases are dissipated, bottoming cycles are appropriate in furnaces and kilns for the production of ceramics, steel, etc.

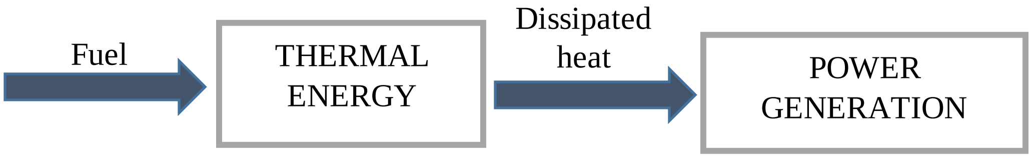


Fig 2.3: Schematic for a bottoming cycle

III. **Classification according to the industrial plant's size**: Different sizes of industrial plants exist and this is based on the amount of power required and the application of the CCHP system. They are the micro-scale, small-scale, medium-scale, and large-scale (Liu, Shi & Fang, 2014).

i. Micro-scale systems: These are CCHP systems in which their power generation is rated at a value below 20kW. Due to their low power output range, they are suitable for applications in residential and small commercial areas.

ii. Small-scale systems: These are CCHP systems in which their power-generating capacity ranges between 20kW and 1MW. They are commonly used in areas such as hospitals, supermarkets, universities, etc.

iii. Medium-scale systems: These are systems with a power generation capacity ranging between 1MW and 10MW. They are well-suitable for applications in big factories and schools.

iv. Large-scale systems: These are systems in which their power generation capacity is above 10MW. They are employed for use in large industries and rejected heat from them can be used to power schools and small communities.

2.4 Sub-systems of CCHP systems

The different component units of a typical CCHP system will be discussed in this section.

2.4.1 Prime mover

This can be defined as a system that converts other forms of energy (heat, electrical) to mechanical energy. A typical prime mover produces mechanical output in terms of the rotation of a shaft which is then coupled to an electric motor to generate electricity. The reciprocating internal combustion engines and the gas turbines are widely used for commercial purposes (Li et al. 2006). Other types such as the organic Rankine cycle, fuel cells, Stirling engines are still emerging and under research. Several types of prime movers have been discussed in this section and summarized in Table 2.1.

i. **Steam turbines**: The steam turbine produces rotary motion due to the heat energy from high pressure steam. It is one of the most versatile and ancient prime movers. It has been in existence for about 100 years after replacing steam engines as a result of its improved efficiency and cheap cost. With a power capacity above 50kW, they are suitable for application in large power plants and multi-generation systems. Steam turbines are typically bottoming cycle systems in the sense that thermal energy is the primary generated product while electricity generation is secondary. When integrated with a boiler, it can run on various types of fuels however, its delayed start-up time and poor electric efficiency limit its use in small-scale CCHP systems and other distributed energy applications (Wu & Wang, 2006). The steam turbine's thermodynamic cycle is known as the Rankine cycle. The rankine cycle is a typical power plant that comprises a thermal source (usually a boiler) that transforms water to pressurized steam. At first, water is pumped into the cycle at a high pressure, warmed up to the boiling point, and a few times above the boiling point at the same pressure. The high pressure is expanded at a multi-stage turbine where it is converted to steam at a lower pressure. Subsequently, the steam is dissipated into a condenser operating at a vacuum state or an intermediate distribution system that supplies steam for industrial or commercial applications. The condensed liquid is collected by the feed-water pump thus ensuring continuity of the cycle.

   The steam turbine is of two types namely: the back pressure and extraction condensing steam turbine. In the back-pressure steam turbine, high-pressure steam, greater or equal to the ambient, is extracted at intermediate stages of the turbine. In extraction condensing steam turbines, however, steam is extracted at the atmospheric pressure (Darrow et al. 2015). Selection of the type of steam turbine to be used is dependent on the amount of power/heat required, quality of heat as well as the cost considerations.

ii. **Reciprocating internal combustion engine (RICE):** This is a heat engine whereby the chemical reaction resulting from the combustion of an air-fuel mixture propels the piston, which through a crank-rod mechanism, produces rotary motion (Lévy, 1996). It is further categorized into the spark ignition engine which runs on natural gas or gasoline and compression ignition which runs on heavier fuels such as diesel oil.

It has a power generation capacity ranging from about 10kW to more than 5MW. Also, their cost- effectiveness, rapid start-up times, high reliability and high load adaptability make them suitable for CCHP systems used in industrial and university environments (Onovwiona & Ugursal 2006; Knight, Ugursal & Beausoleil-Morrison, 2005). However, its operation creates lots of noise, requires regular maintenance, and emits a large amount of greenhouse gas hence making it inappropriate for applications in residential areas (Kuhn, Klemeš & Bulatov, 2008). Waste thermal energy from the RICE which comprises the exhaust, engine jacket water, lube oil cooking, and turbocharger cooling can be employed to supply cooling power in CCHP systems. The reciprocating internal combustion engine proffers a high efficiency when compared to a gas turbine.

iii. **Gas turbine**: The gas turbine is a commonly used prime mover in multi-generation systems and runs on the Brayton thermodynamic cycle. It comprises an air compressor, combustion chamber, turbine and electric generator linked with a single shaft. In a Brayton cycle, atmospheric air is introduced into the compressor which compresses the air to an elevated pressure and temperature. The resulting compressed air enters the combustion chamber to be mixed with fuel and the subsequent combustion process which produces hot gases. The hot gases are introduced into the turbine where expansion takes place to rotate its blades. The rotary motion of turbine blades rotates its shaft and is transmitted to the electric generator to produce electricity. Also, waste exhaust gases from the turbine can be recovered to pre-heat the compressed air entering the combustion chamber through a heat recovery system. There are two types of gas turbines namely: the open-cycle gas turbine and the closed-cycle gas turbine (Petchers, 2020). In the open-cycle gas turbine, the conventional Brayton cycle is incorporated with a diffuser employed to introduce surplus air to the combustion chamber. In the closed-cycle gas turbine, helium or air circulating in a closed loop, instead of the hot gases, is used to rotate the blades of the turbine. Hence, thermal energy is transferred from the hot combustion gases to the air through a heat exchanger. Although closed-cycle gas turbine is less susceptible to turbine blade erosion, it is not commercially available (Ghaebi et al. 2011; Oland, 2004). The gas turbine is suitable for application in multi-generation systems and its development has witnessed rapid growth due to readily available natural gas, reduced cost of installation, rapid technological advancements, short gestation periods and reduced greenhouse gas emission (Lahoud, Al Asmar & Brouche, 2018). With a power generating capacity ranging from 500kW to about 100MW, their exhaust gases can be recuperated to drive heating and cooling systems as well as produce extra power output.

iv. **Microturbine**: This is a smaller version of the gas turbine capable of generating power output ranging from 30kW to 350kW and has the ability to run on a variety of gaseous and liquid fuels. With a similar principle of operation as the Brayton cycle, the micro turbine consists of a compressor, turbine, and recuperator. Its compact size low noise, reduced number of moving parts, minimal vibration, fast delivery time, and low maintenance requirement make it preferable over the gas turbine. Hence, it is suitable for distributed energy systems, particularly for multi-generations systems.

There are two configurations of the micro turbine and they are the single-shaft and double-shaft models. In the single-shaft micro turbine, the compressor, turbine, and electric generator are connected by a single shaft hence allowing for reliability. In contrast, for the double-shaft micro turbine, exhaust gases from the compressor-turbine assembly are used to drive another turbine which in turn powers an electric generator linked by a gearbox.

v. **Combined-cycle gas turbine**: This system is made up of double thermodynamic cycles linked with a working fluid (usually helium or air) and is operated at dissimilar temperatures. The most commonly used combined cycle system is known as the combined joule rankine system which involves the use of both gas and steam turbines. The working fluid is circulated in a closed loop, and gains thermal energy via a heat exchanger before getting to the turbine. The exhaust gases from the turbine can also be used to produce pressurized steam that generates additional power via a steam turbine. This integration can be employed for power generation on a large scale. According to MITCO (2006), this type of gas turbine has an efficiency of 70-90% and a power-to-heat ratio of about 0.6-2.

vi. **Stirling engine**: Unlike the reciprocating internal combustion engine, the Stirling engine operates on a Carnot cycle. It possesses an externally located combustion chamber ad confines the working fluid (helium or air) within a closed cycle where compression, heating, and expansion take place (Alanne & Saari, 2004). It can be categorized into the kinematic and free-piston Stirling engine and exists in three configurations- alpha, beta, and gamma types (Harrison, 2002). Generally, about a third of the thermal energy input into the Stirling engine is transformed to electric power with the remaining two-thirds dissipated to drive the cooling systems as exhaust gases. Although it is still under research and development as regards its application in multi-generation systems, the Stirling engine have some advantages over the RICE. It produces lower greenhouse gas emissions, is less noisy, and more flexible as it concerns fuel usage. However, it has the drawback of being capital-intensive, and yielding a lower electrical efficiency and power output when compared to the RICE.

vii. **Organic rankine cycle (ORC):** The ORC is based on a similar concept as the steam turbine. It operates on the rankine cycle to generate usable work from heat input using an organic working fluid of higher or lower boiling temperature. Using working fluids with low boiling points, it is capable of working as 'bottom' system that utilizes waste heat from multi-generation systems or boilers. In contrast, using working fluids with higher boiling points would make for the generation of useful work as well as dissipate waste heat for the operation of other sub-systems. Although still under research and development, its durable nature, simple configuration, cheap cost, and low operating temperature and pressure make it an ideal choice for multi-generation systems (Karellas & Schuster, 2008).

viii. **Fuel cells**: These are a different type of power generation system that generates electricity through the chemical reaction between hydrogen and oxygen with an electrolyte present thus producing water as a by-product (Wang & Nehrir, 2006). It is made up of the following (DOE, 2000).

a. Reformer responsible for extracting hydrogen,
b. Fuel stack – an electrolyte material sandwiched between the anode and cathode,
c. Inverter- to convert direct current to alternating current.

They have a low greenhouse emission rate, require little maintenance, are less noisy, and exhibit a high efficiency under partial loads. There are five types: the solid oxide fuel cell (SOFC), commonly used in multi-generation systems, are operated at high temperatures of about 1000ºc and this warrants the use of a ceramic-based electrolyte. They can obtain fuel-to-electricity efficiencies of 45-60%. The alkaline fuel cell makes use of potassium hydroxide solution as its electrolyte operating at temperatures ranging from 60ºc-80ºc. The phosphoric acid fuel cell can attain fuel-to-electricity efficiency of 35-45% while operating at 100ºc-200ºc. Furthermore, the proton exchange membrane fuel cell possesses a high-power density, fast starting times, and provide rapid response to fluctuations in power requirements. Operating at temperatures between 50ºc and 100ºc, it employs solid electrolyte plastic sheets with a small thickness that allows the flow of hydrogen ions. The molten carbonate fuel cell, which uses a mixture of molten carbonate salt as electrolyte, is able to achieve low heating value efficiencies ranging from 45-60%.

Table 2.1: Selected prime movers comparison (Liu, Shi & Fang 2014)

Prime mover	**Size**	**Advantages**	**Disadvantages**	**Greenhouse gas emissions**	**Preferences and use cases**
Reciprocating internal combustion engine	10kW-5MW	• Cost effective • Rapid start-up time • Highly reliable • Excellent load following • Excellent partial efficiency	• Needs to be maintained regularly. • Noisy during operation	• High NO_x when diesel is being used. • Natural gas is preferable	• Operated with absorption cooling systems. • Small and medium-scale applications
Gas turbine	500kW-100MW	• High-grade exhaust heat • Reduced cost of installation	• Insufficiently small partial efficiency	• NO_x 25 parts per million • CO 10 to 50 parts per million	• Applicable where a large amount of thermal energy is required. • Large-scale
Steam turbine	50kW-250MW	• Flexible usage of fuel	• Poor electric efficiency	• Based on the fuel type	• Suitable where thermal energy is of

					priority and electricity is a secondary product. • Large-scale
Microturbine	1kW-350kW	• Flexible usage of fuel • Rapid rate of rotation • Small size • Minimal noise and vibration. • Little maintenance requirement	• Lengthy startup period • Cost intensive	• NO_x < 10 parts per million	• Suitable for distributed energy systems. • Micro to small-scale applications
Stirling engine	About 100kW	• More secure with minimal noise. • Flexible usage of fuel • Lengthy service period	• Cost intensive • Difficulty in tuning its power output	• Smaller than the reciprocating internal combustion engine.	• Small-scale applications
Fuel cell	0.5kW-1.2MW	• Minimal noise • Highly reliable	• Greenhouse gas emission associated	• small	Micro to medium-scale applications

		• Excellent efficiency	with the production of hydrogen		

2.4.2 Heat recovery system

In multi-generation systems, the heat recovery system is used to trap thermal energy from a 'top' system's exhaust gas/liquid stream and convert it to usable energy. Hence, reducing the fuel consumption rate and improving the overall efficiency. It is made up of heat exchangers of different types that are employed according to the need application. Heat recovery systems can be categorized as either fired or unfired and would be discussed herein.

i. **Unfired heat recovery system**: This system captures thermal energy dissipated from a prime mover and cannot create further heat hence is not a source of greenhouse gas emission as no fuel consumption takes place (ASHRAE, 2000). Example of this type of system are the heat exchangers, unfired heat recovery steam generator (HRSG), mufflers, regenerators, and recuperator.
    a. **Unfired heat recovery steam generator**: This performs the duty of heat exchangers because it utilizes the thermal energy from hot gases to generate high temperature gas and liquid. It exists as either a water-tube heat-recovery steam generator or a fire-tube heat recovery steam generator (Kitto, 2005).
    b. **Heat recovery muffler**: This type of system is used to regain thermal energy from the exhaust of reciprocating internal combustion engines. Similar to the HRSG, it is able to produce high-temperature liquid and pressurized gases from hot gases with a temperature of 370^{o}c-540^{o}c at a low noise.
    c. **Recuperator**: This is normally known as an air heater. It is a gas-to-gas heat exchanger that captures heat using its plate or tube feature hence transforming heat from the hot gases to the combustion air without mixing both gases. It is employed in most gas turbines (Behbahani-Nia, Bagheri &Bahrampoury, 2010)
    d. **Regenerator:** This is a gas-to-gas heat exchanger that allows the transfer of thermal energy to combustion air when a heat storage tank is in turn left open to hot and cold streams with the aid of a switching tool (Oland, 2004).
    e. **Heat exchanger**: This constitutes other types of heat exchangers that are utilized for the recovery of heat such as the shell and tube, plate, fins, and micro-channeled heat exchangers.

ii. **Fired heat recovery system**: This type of system performs the function of capturing thermal energy dissipated from an independent source. It also generates further thermal energy through the combustion of fuel. There exist two main types of fired heat recovery systems used in multi-generation systems and they include:

a. **Fired heat recovery steam generator**: This is normally employed in prime movers with high-temperature exhaust gases such as the gas turbine multi-generation system. It is incorporated with a ductwork that connects the HRSG to the turbine. This duct work contains a burner that increases the incoming temperature of exhaust gases to about 980ºc (Oland, 2002).
b. **Boiler**: The main concept behind incorporating boilers in multi-generation systems as heat recovery systems is to generate a large amount of steam with low fuel using the heat dissipated from a prime mover. Commonly referred to as a waste heat recovery boiler, it is classified as either a water-tube or fire-tube boiler. The design of its system are according to the fuel type, pollution control technology, and other economic factors (Oland, 2004).

2.4.3 Cooling system

This is also known as a thermally activated system and utilizes the thermal energy from exhaust gases, hot water and steam from a prime mover to provide cooling and air-conditioning functions. This is achieved by a phenomenon called 'sorption cooling' whereby absorption and adsorption activities are used to produce thermal compression in contrast to mechanical compression generated in traditional refrigeration processes. There are three main types of cooling systems employed in multi-generation systems and they are absorption cooling, adsorption cooling and desiccant humidifier. Since each of these cooling system types has their appropriate working temperature, their selection for CCHP applications depends on the temperature of the prime mover's exhaust gas. Table 2.2 presents a comparison between the different cooling systems.

i. **Absorption cooling**: Its working cycle originated in the 1700s when it was discovered that the evaporation of clean water, from a vessel housed in an empty container with sulphuric acid, could produce ice (Gosney, 1982). The absorption cycle comprises four elements: an absorbent, a generator, condenser, and an evaporator. The refrigerant and absorbent work as a pair. In contrast to traditional vapor compression systems, absorption systems use the thermal energy produced by the generator for the compression of the refrigerant vapor. Furthermore, the principle of operation for the absorption chiller is based on the absorption and separation process as shown in Figures 2.4 and 2.5. During the absorption process, the temperature and pressure of the refrigerant reduces, and heat is dissipated to the environment. The absorption process continues until the solution becomes saturated. The separation process, which is the opposite of the absorption, acts to separate the refrigerant from the solution in order to ensure the continuity of the absorption process. Thermal energy is used to heat the solution thus drying out the refrigerant from it. The resulting refrigerant vapor from the drying process goes through the condensation process in a heat exchanger to begin a new absorption cycle.

Because the performance of the absorption cooling system depends on the properties of the working fluid, the selected working fluid must be chemically stable, non-poisonous, and should not explode easily. A study by Macriss, Gutraj & Zawacki (1988) revealed that there are about 40 refrigerant compounds and about 200 absorbent compounds suitable to be used as working fluid for an absorption chiller. Nevertheless, the lithium-bromide/water (LiBr/water) and ammonia/water (NH_3/water) are widely employed.

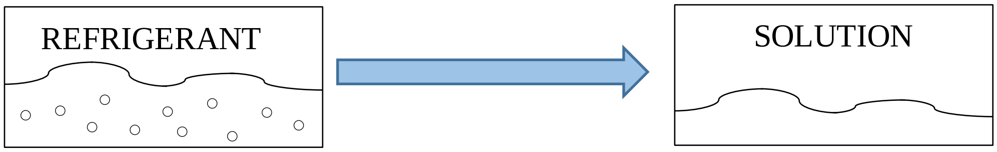


Fig 2.4: Absorption process (adapted from Liu, Shi & Fang, 2014)

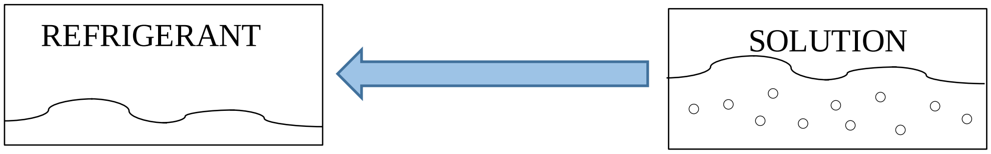


Fig 2.5: Separation process (adapted from Liu, Shi & Fang, 2014)

ii. **Adsorption cooling**: Its working cycle originated in 1848 when Faraday discovered the concept of adsorption refrigeration influenced by the adsorption of ammonia on AgCl (Critoph & Zhong, 2005). The difference between an absorption and adsorption cooling system is the fact that the latter can operate on low temperature exhaust fluid. Although not as developed for application in multi-generation systems as the absorption cooling system, the adsorption chillers employ the use of solid adsorbent beds for the adsorption and desorption of a refrigerant. Similar to the absorption chiller temperature changes in the adsorbent are based on the amount of refrigerant vapor adsorbed and desorbed.

The adsorption cycle comprises a solid adsorbent bed, condenser, expansion valve and evaporator (Yong & Wang, 2007). Also, its process can be categorized into absorbent heating and desorption process and the adsorption process. In the absorbent heating and desorption process, the adsorbent bed is combined with a condenser and low temperature is used to condense the refrigerant. The second process involves integrating an evaporator to the adsorbent bed to produce the cooling effect. The most widely used working fluids for the adsorption cooling are the silica gel-water, zeolite-water and activated carbon-methanol.

iii. **Desiccant humidifier**: This is used to reduce the humidity of the atmosphere through the use of materials capable of attracting and retaining moisture. It utilizes waste heat to improve air-quality, decrease cooling loads and provide thermal comfort through sensible cooling and air humidity control.

Desiccant humidifiers are of two types; the solid and liquid desiccant humidifier. The solid type employs a rotary wheel incorporated with a desiccant substance that allows for the removal of latent load using adsorption. In the liquid type, dehumidification of air is carried out by a liquid desiccant and a regenerator is utilized to recover the desiccant solution using thermal input. Solid desiccant humidifier is applied for commercial purposes and is more advanced in terms of its development unlike the liquid.

Table 2.2: Comparisons between various cooling systems (Liu, Shi & Fang 2014)

Cooling system	**Cooling power**	**Advantages**	**Disadvantages**	**Coefficient of performance**	**Preference**
Absorption chiller	10kW-1MW	• Powered with steam • Produces minimal noise. • Capable of being powered by low-temperature heat • Minimal greenhouse gas emission	• Lower efficiency when compared with a compression-driven chiller	About 1.2	• Evaporation temperature of Lithium bromide/water: 5ºc to 10ºc. • Evaporation temperature of Ammonia/water: < 0ºc. • Small to large-scale application.
Adsorption chiller	5.5-500kW	• Powered with steam. • Compact size • No problem with corrosion	• Strictly powered by high-temperature heat. • Cost intensive	0.6	• Small-scale application

		or crystallization • Minimal noise. • Does not require lubrication. • Minimal greenhouse gas emission			
Desiccant humidifier	N/A	• Possible to regulate humidity without affecting the temperature. • Decreased demand for mechanical cooling	• Cost intensive. • Needs to be regularly maintained	N/A	• Solid: heating, ventilation, and cooling (HVAC) system. • Liquid: application in residential and industrial areas.

2.4.4 Kalina cycle

The Kalina cycle, which is an extension of the Rankine cycle, was proposed by Alexander I. Kalina in the 1800s for producing electricity. It employs low-temperature exhaust gas/liquid streams and uses the ammonia-water as a working fluid (Kalina, A.I., 1984). A typical Kalina cycle comprises a steam turbine, heat exchanger, recuperator, evaporator, condenser, pump, separator, etc. Studies reveal that the ammonia water is selected as the working fluid for the following reasons (Ogriseck, 2009; Mirolli, 2007):

i. It enables a more effective utilization of the exhaust thermal energy under greater pressure by forcing boiling to begin at a lower pressure.
ii. It allows for variation in the constituents of the mixture via the distillation. The variation in the mixture's temperature decreases heat transfer losses hence improving the kalina cycle's efficiency.
iii. It enables the usage of standard steam turbine components because the molecular weights of ammonia and water are alike which makes the ammonia-water vapor act as steam.
iv. It allows for the utilization of standard materials such as carbon steel and high-temperature alloys.
v. Ammonia is widely available, reasonably priced as well as non-toxic to the environment.

A Study by Jurgen (1986) suggests that the Kalina cycle assists in transforming into electricity about 45% of heat input from a direct-fired system and about 52% from a combined-cycle plant. This is higher than the 35% and 44% respectively obtainable from the Rankine cycle. Also, in industrial exhaust heat use cases, the Kalina cycle generates 32% additional power when compared with the traditional Rankine cycle. The opposite is however obtainable in a small-scale direct-fired biomass-based multi-generation system where the Rankine cycle exhibits a better performance than the Kalina cycle.

Figure 2.6 shows a Kalina cycle driven by exhaust gases (1, 2) to a boiler. Work output (4) is generated from the expansion of overheated ammonia water (3) in a turbine. The exhaust from the turbine (5) is subsequently cooled (6, 7, 8), mixed with the working fluid (9, 10), and left to condense (11) using cooling water in the absorber (12, 13). Furthermore, the saturated liquid undergoes compression (14) and heating (15, 16, 17, and 18) before being separated into ammonia-poor liquid (19) and ammonia-rich vapor (22). The ammonia-poor liquid is then subjected to cooling (20, 21) and depressurization in a throttle. The ammonia-rich vapor undergoes cooling (23) and is mixed with the initial condensate (24) to boost the ammonia composition (25). Finally, the mixture is allowed to cool (26), and condense (27) using cooling water (28, 29). It is then compressed (30) before being supplied to the boiler (31) through its feed water heater for continuity of the cycle.

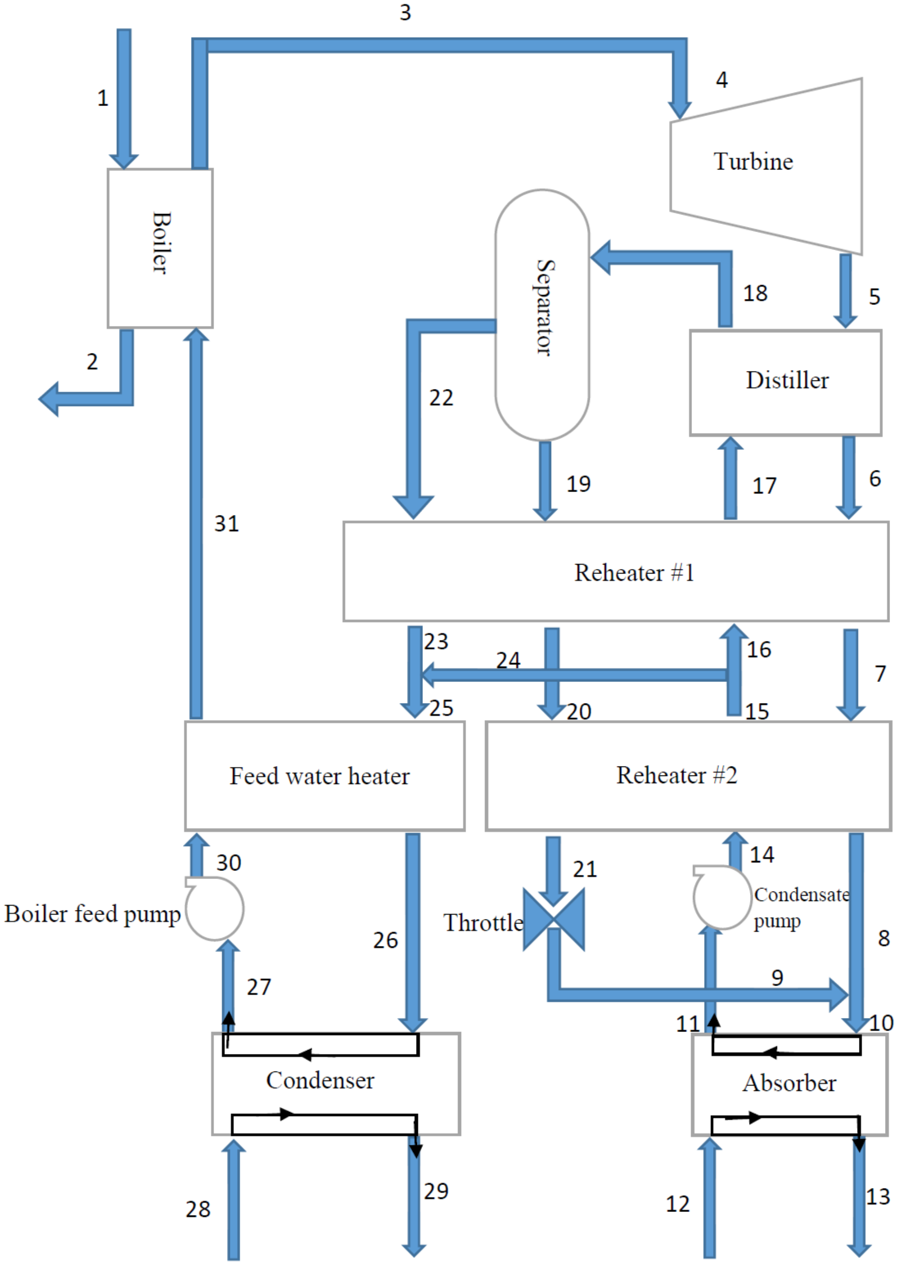


Fig 2.6: Schematic of a typical Kalina cycle (adapted from Zhang, He & Zhang, 2012)

2.5 Solar-based CCHP systems

Solar thermal and solar photovoltaic technologies have been extensively researched as viable solar harvesting options for power generation in a bid to reduce fossil fuel consumption. A photovoltaic panel (PV) can be described as a semiconductor comprising diodes and resistors that convert energy from the sun into electricity (Shahverdian et al. 2021). Solar thermal collectors, on the other hand, convert radiation from the sun into thermal energy. PV is not commonly used in multi-generation systems due to its relatively high cost and low conversion efficiency unlike solar thermal collectors (Sharma & Bhattacharya, 2022).

Solar thermal collectors can be classified as either concentrating or non-concentrating collectors. Non-concentrating solar collectors supply heat at low temperature, an examples of which are the flat plate and evacuated tube collectors. The flat plate collector collects photons via an absorber surface, and transforms them to heat which is subsequently transported in fluid tubes. Evacuated tube collectors operate in the same way but an evacuated tube is used to replace the absorber surface.

The concentrating solar thermal collector supplies thermal energy at a high temperature and this makes them suitable for application in multi-generation systems. It produces steam for a steam turbine prime mover and also pre-heats combustion gases for a gas turbine prime mover. Examples of concentrating solar collectors are the parabolic trough collector (PTC), linear Fresnel reflector (LFR), Heliostat field collector (HFC) parabolic dish collector (PDC), and compound parabolic collector (CPC).

Considering other concentrating collectors, the PTC is predominantly employed for supplying thermal energy. It is made up of a parabolic long trough-shaped reflector that focuses radiation from the sun on a tube that runs through the focal line of the mirror. The tube is made up of black metal and coated with a material that boosts solar absorption while decreasing heat emittance. For further heat loss prevention, the tube is enclosed in a glass pipe and a vacuum is placed in the area between the black metal tube and the glass pipe. Also, the glass pipe is frequently covered with anti-reflective material to improve transmistivity. The working fluid normally used is either an absorbent oil or water and it flows through the tube. The heat reflected by the trough is absorbed by the working fluid which then transfers the thermal energy to another fluid via a heat exchanger. The optimal efficiency, known as the amount of radiation absorbed by the system, for a PTC is about 70-80% (Fanchi, 2004).

The PDC is made up of a mirror array mounted on a two-axis sun-tracking system to ensure the convergence of light on the dish focal point. A receiver is locked on the focal point of the dish where radiation from the sun is being focused, in the form of heat. Power can be generated by situating a heat engine (such as the Stirling engine, or steam turbine) close to the dish focal point to directly collect thermal energy from the receiver to produce electricity from mechanical energy. Also, the working fluid can be connected to the receivers of several dishes and directed to flow to a central power generating unit thus supplying its acquired thermal energy. The PDC is capable of attaining a concentrated high light of about 1500ºC generating electricity of up to a few kilowatts (Karabulut, 2009).

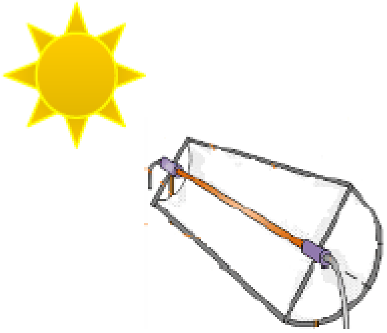
The HFC comprises sets of reception/tracking mirrors that are arranged in concentric rings round about a central tower. The heliostat focuses the radiation from the sun on a receiver located on the tower) that houses a working fluid flowing in a closed circuit. A typical heliostat has a surface area of 50m to 150 m2 and is mounted on a two-axis tracking tower usually 75m to 150m tall. Table 2.3 provides a summary of the three main solar thermal collectors while Table 2.4 shows their pictorial form.

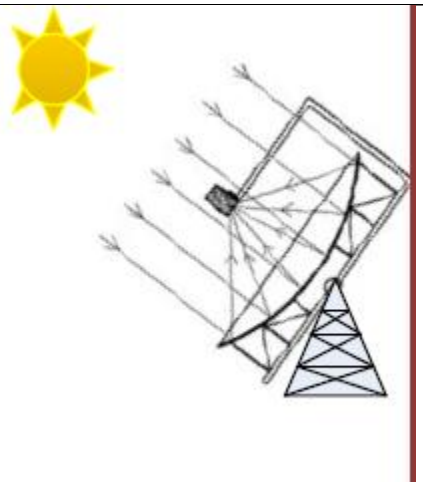
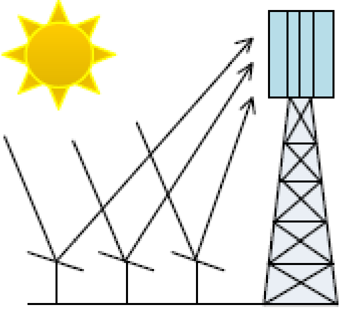
Table 2.3: Summary of concentrating solar thermal collectors (Kalogirou, 2004; Barlev, Vidu & Stroeve, 2011)

Concentrating Solar collector	**System description**	**Operating temperature(ºC)**	**Concentration ratio(sun)**	**Thermodynamic efficiency**
Parabolic trough collector	The sun radiation is concentrated and transferred to a heat transfer fluid flowing in a metal pipe using a parabola sheet formed of a reflective material**.**	50-400	15-45	Low
Parabolic dish collector	Mounted on a two-axis sun tracking system to ensure the	150-1500	100-1000	High

	convergence of light on the dish focal point. A receiver is locked on the focal point of the dish where radiation from the sun is being focused in the form of heat			
Heliostat field	A sizable array of heliostats used to focus solar flux over an energy reservoir installed on a tower or combined with a steam power cycle.	300-2000	100-1500	High

Table 2.4: Schematic and pictorial form of concentrating solar thermal collectors (Kalogirou, 2004; Barlev, Vidu & Stroeve, 2011)

Concentrating Solar collector	Schematic	Pictorial form
Parabolic trough collector		

Parabolic dish collector		
Heliostat field		

2.6 Solar-based Tri-generation systems

There have been various research studies concerning the integration of renewable energy sources with the CCHP systems and according to Bai & Xu (2018), solar energy is the most commonly employed renewable energy. The works of literature reviewed in this section draws similar conclusions on the efficiency of the solar-based CCHP system over the separate power, heating and cooling systems.

Yan et al. (2020) proposed a solar-assisted CCHP system that employed the phosphoric acid fuel cell as the prime mover. The energy and exergy analysis of the system with and without solar energy integration was reported as well as other parametric analysis performed. Among other findings, it was discovered that increasing the reforming temperature, which translates to an increase in the solar energy input, improves the electrical power output and overall exergy of the system.

Li et al (2020) evaluated the energy, economic, and environmental factors of a low-grade CCHP system by integrating photovoltaic thermal collector with a solar absorption-subcooled compression cooling system with data obtained from three cities. The proposed approach achieved an improved electricity saving value and system performance in areas with high solar irradiation.

Saini, Singh & Sarkar (2020) presented a solar-based CCHP system for application in a small building. The authors also carried out the exergy, economic, and environmental analysis of the proposed system as well as parametric assessments. Their approach yielded improved exergy efficiency, total cost, and CO_2 reduction. It was also pointed out that increased generator and condenser temperature translates to the improved exergy efficiency of a tri-generation system.

Wang et al. (2021) developed a solar energy device integrated with a CCHP system. The solar energy device comprises a solar thermal water heating system. They obtained a solar energy share, energy and exergy efficiency of 45.07%, 70.65%, and 26.59% respectively. When compared with a traditional methanol-based CCHP system, the proposed system yielded better performances in terms of the energy-saving ratio and CO_2 emission reduction rate.

Furthermore, a solar energy device based on the R134a refrigerant and incorporated into a tri-generator system was designed by Mohsenipour et al (2020) for application in agriculture. The proposed system was simulated with the engineering equation solver and the TRNSYS software. In comparison with a similar study, the authors achieved a better fuel and water-saving values.

A comparative study, between two developed solar-based CCHPs based on the organic rankine cycle, was conducted by Jafary et al. (2021). The first system was incorporated with an internal heat exchanger while the second system was provided with a mixture heater. With both systems utilizing the PTC for solar energy collection, the energy and exergy efficiencies of the first system surpassed that of the second system.

Aghaziarati & Aghdam (2021) developed a solar organic rankine cycle-based CCHP system for application in a hospital. A thermodynamic analysis of the system yielded energy and exergy efficiencies of about 89% and 8.7% respectively. Further analysis conducted showed that better system performance are obtained with a PDC than a linear fresnel reflector.

Haghghi et al. 2019 developed a novel solar-based CCHP system for application in an academic environment. The proposed system comprised a solar energy device driven by parabolic trough solar collectors, an organic Rankine cycle for power generation, heating sub-systems, and two single effect absorption chillers. A thermodynamic analysis was performed to ascertain the energy demand and fluctuating solar radiation for three distinct operational modes. A parametric analysis was carried out to understand the effects of the inlet pimp temperature, inlet pump pressure, and number of days on the system performance.

Peng et al. (2020) proposed a novel solar-energy integrated CCHP system that comprised a solid oxide fuel cell for electrical and thermal energy generation, a steam power turbine to produce further power, a PTC for capturing thermal energy, and a double effect absorption chiller for cooling purposes. High-temperature gases dissipated from the fuel cell were used to drive the absorption chiller and the PTC was structured as a heat exchanger to provide heating demand. A sensitivity analysis was conducted to investigate the impact of various system variables on the overall performance.

Alharthi et al. 2023 developed a solar-based CCHP system for the simultaneous generation of power, heating, and cooling demands employing an approach of generator temperature that boosts the efficiency of solar radiations. The authors also investigated the effect of some variables such as evaporator temperature and solar heat transfer fluid on the system performance. Among other research findings, the solar heat transfer fluid was seen to have a positive correlation with the electrical power and process heat but did not affect refrigeration.

2.7 Optimization

CCHP systems provide a sustainable way to enhance energy conservation by lowering greenhouse gas emissions, heat loss, and operating costs as well as increasing overall energy efficiency and guaranteeing the availability and dependability of several energy-generating options (Dincer & Zamfirescu, 2012.). Still, there is a constant search for more ideal thermodynamic performance measures and optimization can improve the performance of the CCHP system. Mirjalili, Jangir & Saremi (2017) defined optimization as the science of solving real problems by obtaining optimal design solutions using machines/software. With the development of optimization techniques in the 1960s and 1970s, a meta-heuristic method called evolutionary algorithms was introduced. Holland's genetic algorithm optimization is a well-known illustration of this methodology (Holland, 1975). Charles Darwin's theories of mutation, crossover, and survival of the fittest served as the basis for this strategy. The swarm intelligence algorithm, developed by Dorigo, Di Caro, &

Gambardella (1999), was another key metaheuristic technique that gained prominence in the 1990s. Kennedy & Eberhart (1995) suggested the ant colony optimization (ACO) and particle swarm optimization approaches, respectively. Since most real-world engineering problems have several objectives, it follows that there are generally multiple objective functions involved in the mathematical formulation (Mirjalili et al., 2016). Multi-objective functions were used by Zeng et al. (2015) and Song et al. (2020) to successfully increase the objectives of a CCHP system. They are addressed by randomly allocating weights in a weighted-sum problem formulation. However, the weight-based optimization or a priori method has the disadvantage of needing several runs and necessitating constant consultation with an expert or decision-maker (Kim, & De Weck, 2005). The posteriori technique, which keeps the multi-objective formulation intact and finds the Pareto optimal solutions in a single run, can also be used to solve them. These, however, require a lot of computing power.

The response surface method (RSM) (Hasanzadeh, et al. 2022), non-dominated sorting genetic algorithm-II (NSGA-II) (Ren et al. 2019), particle swarm optimization (PSO) (Azaza & Wallin, 2017), Harris hawk optimization (HHO) (Shehab et al 2022), grasshopper optimization (GOA) (Sharifian, & Abdi, 2022), ant-lion optimization (Ji et al. 2022), moth flame optimization (MFO) (Xu et al. 2023), and grey wolf optimization (GWO) (Chen, Huang & Shahabi 2021), among others, are some of the optimization techniques that have been documented in the literature. This study shows how a solar-assisted integrated cooling, heating, and power system problem might be formulated and solved using GWO. Figure 2.7 shows a breakdown of the optimization algorithms.

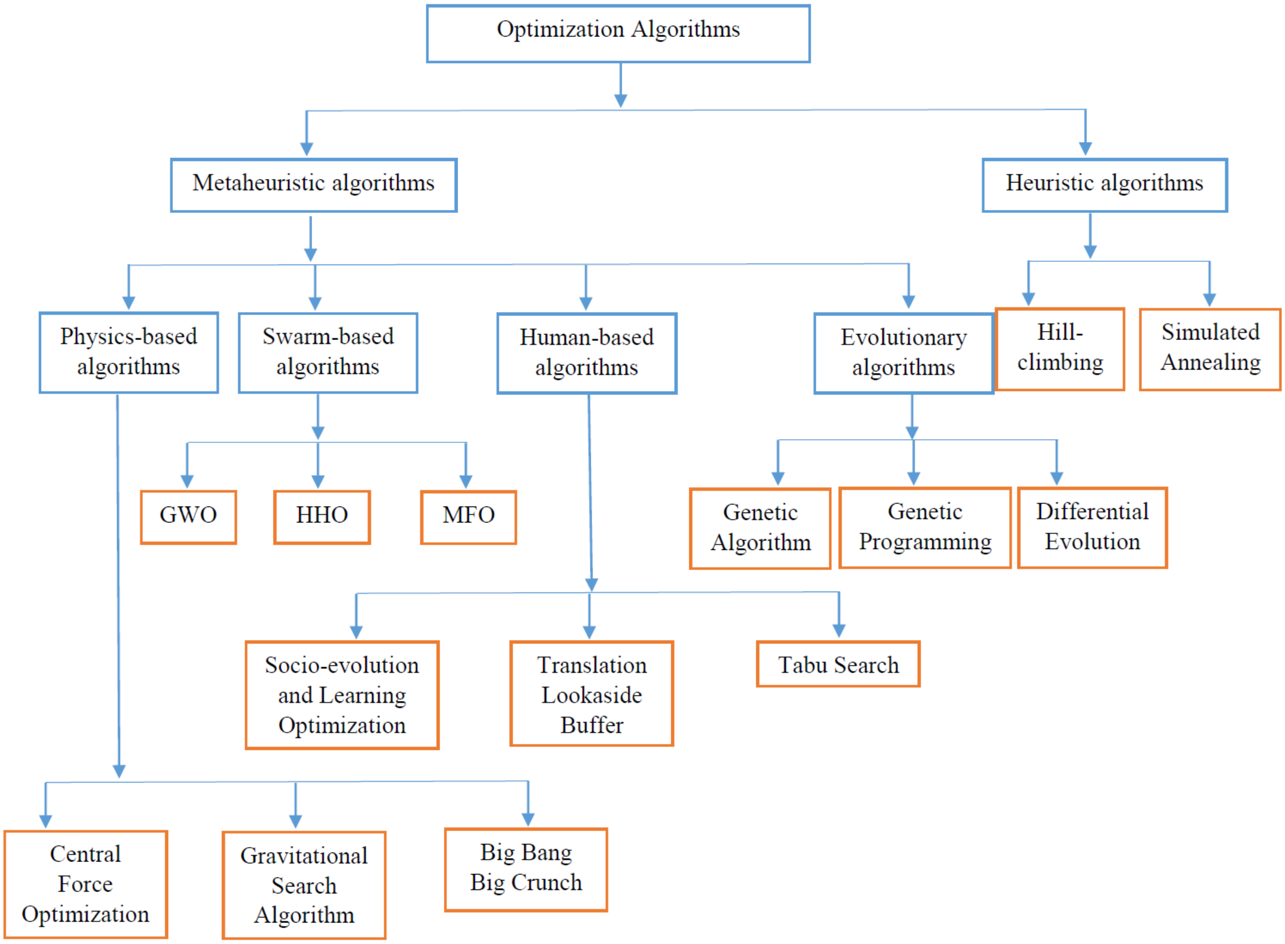


Fig 2.7: Optimization algorithms

Mathematically, optimization problems can be generally expressed as follows:

$$\underbrace{Maximize}_{a\in X}\ f_b(x), \qquad (b\ = 1{,}2, \dots K) \tag{2.1}$$

$$Subject\ to\colon\ h_c(x) = 0, \quad (\ c\ = 1{,}2, \dots L) \tag{2.2}$$

$$m_d(x) \leq 0, \qquad (d\ = 1{,}2, \dots N) \tag{2.3}$$

$$x\ =\ (x_1, x_2, \dots x_n)^T \tag{2.4}$$

Where,

$f_b(x)$ = objective functions,

$h_c(x), m_d(x)$ = constraints,

X= design/search space,

x_i= decision variables

2.8 Multi-objective-optimization of solar-based tri-generation systems

The ordeal to continuously improve CCHP systems with various optimization techniques is a progressive research trend in the domain of energy conservation/management. Therefore, this section will review relevant pieces of literature that optimized certain performance criteria of the solar-based CCHP system. An extensive review revealed that a greater number of researchers employed the genetic algorithm for optimization applications in solar-assisted CCHP systems.

Cao et al. (2021) proposed a modified solar-integrated CCHP system and optimized its electricity generated, exergy efficiency, and total cost per unit exergy via the genetic algorithm approach. They also carried out a parametric study to ascertain how their decision variables (oil mass ratio, Rankine inlet pressure, and temperature, etc.) affect the objective functions. The proposed approach provided improved results in terms of the above-mentioned performance criteria thus outperforming conventional methods.

The thermodynamic analysis and performance optimization of a solar energy and natural gas-integrated CCHP system was presented by Wang et al. (2016). They employed the genetic optimization algorithm to maximize the energetic and exergetic capacities of the CCHP system.

Furthermore, a multi-objective optimization model via the genetic algorithm was developed by Wang et al. (2020). They proposed an operational flexibility approach determined by the sizes of the photovoltaic (PV) solar panels and gas turbines to improve the CCHP system's energy saving, cost saving, CO_2 emission, and grid integration level. The results obtained illustrate that, although the operational flexibility as selected by the entropy weighting method, improved the system's ability to adjust to variable conditions, a corresponding decrease in grid interaction level, and exergetic, economic, and environmental factors were recorded.

Song, Liu & Lin (2020) employed the NSGA-II for the multi-objective optimization of a solar-based CCHP system modeled in three operational modes. Utilizing the gas turbine capacity, PV area, and solar collector area as decision variables, an optimal solution that maximized the cost-saving and energy-saving ratio was obtained. The study confirmed that the CCHP system was majorly affected by energy prices and efficiencies of the PV, solar collector, and gas turbine.

The NSGA-II optimization approach was presented by Yousefi, Ghodusinejad & Kasaeian (2017) to achieve the best microgrid capacities necessary to provide the needed tri-generation loads for a specified structure. They compared the results obtained from an internal combustion engine-based CCHP system and a solar energy-integrated CCHP system. This revealed that the latter had a better performance in terms of primary energy saving, and $C0_2$ emission although at the expense of a slightly increased net present cost.

Boyaghchi & Heidarnejad (2015) developed a medium–scale solar-based CCHP system comprising the organic rankine cycle for additional power generation and a thermal storage tank to regulate fluctuations in solar energy supply and demand. They also optimized the thermal efficiency, exergy efficiency, and total cost rate of the system using the genetic algorithm employing five decision variables. A sensitivity analysis was conducted to ascertain the effect of the inlet turbine pressure, back pressure, evaporator pressure, and heater outlet temperature on the objective functions. The results reported depict a general improvement in the system performance when compared with a conventional system.

Keshavarzzadeh & Ahmadi (2019) proposed a new solar-assisted CCHP system comprising a PTC, thermal storage tank, proton exchange membrane electrolyzer, organic Rankine cycle, and a single effect absorption chiller. In addition to the thermodynamic analysis performed, a multi-objective optimization was carried out using the NSGA-II technique which yielded an optimal exergy efficiency and cost rate of 79% and 0.058$/s respectively. The authors also utilized four other optimization techniques for the optimization of the proposed system and the indicator-based evolutionary algorithm (IBEA) surpassed the others. The IBEA technique was subsequently compared with the originally used NSGA-II and it yielded the best optimal solutions for exergy efficiency and cost.

Zarei et al. 2022, in a bid to improve the electrical and thermal efficiencies, coupled the PV thermal collector with a flat plate collector on a CCHP system. The authors carried out thermodynamic and economic analysis as well as multi-objective optimization using the genetic algorithm. An optimal energy efficiency, exergy efficiency, payback period, and internal rate of return of 90.15%. 12.46%, 4.025 years, and 26.67% were obtained respectively.

Finally, Wang et al. (2021) proposed the integration of solar energy and biomass into a CCHP system. The proposed system consisted of an internal combustion engine, absorption chiller, heat pump, thermal storage tank, etc. The authors employed the NSGA-II technique for the multi-objective optimization of the system performance and also used the ‘decision method’ to choose the best optimal solution from the set of Pareto solutions. From their analysis, it was deduced that it is more advantageous to use high-capacity PV thermal collectors than solar thermal collectors.

2.8.1 Multi-objective grey wolf optimization of Multi-generation systems

The MOGWO is one of the proposed optimization approaches in this research study hence an extensive review will be covered in this section. The application of the multi-objective greywolf technique has been employed for the optimization of various multi-generation systems.

Shakibi et al. (2023) proposed a new solar-assisted CCHP system utilizing the heliostat generation unit and employed the RSM and the grey wolf optimizer for the multi-objective optimization of exergy performance and unit cost via six selected decision variables. They utilized the three weight-based methods to determine the optimal exergy efficiency, unit cost, and performance coefficient.

Asgari et al. (2022) proposed a heliostat solar-based CCHP system incorporated with a phase change material to regulate the heat rate hence ensuring a constant temperature input to the gas turbine. They employed the multi-objective greywolf optimization in a bid to further increase the exergy efficiency and power generated while reducing the unit product cost. The optimization results showed an increase in exergy efficiency and exergoenvironmental impact index as well as a decrease in the unit cost and cooling loads when compared to a similar study.

Haghghi et al. (2023) employed the grey wolf multi-objective technique coupled with the ANN-based procedure for the optimization of a geothermal-operated poly-generation system. Based on the energy, exergy, and economic point of view, the study made use of four distinct approaches that involved the optimization of energy efficiency, investment cost, exergy efficiency, and levelized cost. The study achieved its optimization objective and it was discovered that the water inlet temperature had the most significant effect on the system performance.

Habibollahzade & Houshfar (2020) remodeled an ORC-based power generation system in a bid to reduce the emission of CO_2. This was achieved by incorporating a membrane separator to harness an appreciable amount of the CO_2 into a gasifier utilizing the greywolf optimizer, the proposed model yielded lower CO2 emission rates and higher exergy efficiency and cost when evaluated with a similar study.

Furthermore, Zhang & Sobhani (2022) proposed the analysis and multi-objective optimization of a power and freshwater generation system based on the geothermal and gas turbine cycles. The grey wolf optimizer was employed to maximize the net power, freshwater production, exergy efficiency, and total emission while minimizing the payback period. The sensitivity analysis conducted confirmed that the air-preheater effectiveness has the most effect on the system performance criteria.

A solar-based system that produces power, cooling capacity, freshwater, and hydrogen was presented by Azizi, Nedaei & Yari (2022). A thermodynamic analysis of the proposed model was carried out to ascertain the base conditions of the generated electricity, drinking water, cooling capacity, and hydrogen. Thereafter the greywolf optimizer was applied, using two different scenarios, to optimize the unit cost, exergy efficiency, and rate of freshwater production.

Chen Huang & Shahabi (2021) developed a hybrid CCHP system to reduce primary energy consumption, CO_2 emission, and cost. The study employed a modified version of the grey wolf optimizer that is based on the non-dominated sorting theory, variable detection, memory-based strategy selection, and fuzzy theory. The obtained optimization results were validated using the multi-objective particle swarm optimization technique.

In addition, Behzadi et al (2021) presented a methanol-fuelled co-generation system consisting of a solid-oxide fuel cell (SOFC), heat recovery unit, and absorption power cycle (APC). The grey wolf multi-objective technique was used to optimize the exergy efficiency and total cost implemented on three different systems which are the SOFC, SOFC-ORC, and SOFC-APC. The optimization results indicated better optimal results from the SOFC-APC and this was due to its non-thermal evaporator, condensation process and temperature glide matching. Zhang et al. (2022) investigated the feasibility of a biomass-based co-generation system. The investigations were carried out using fur biomass fuels and the best fuel being the municipal solid waste was subsequently the multi-objective optimization and parametric analysis of the system. Optimum results that maximized the total cost and minimized the CO_2 emission were generated.

Nedaei, Azizi & Farshi (2022) developed a heliostat solar-based multi-generation system comprising the Brayton cycle, absorption refrigeration cycle, humidification, dehumidification, etc. In addition to the thermodynamic exergetic and economic analysis conducted, the grey wolf technique was used to generate optimum values for the exergy efficiency, freshwater production rate, and unit product cost. Finally, Mahdavi, Mojaver & Khalilarya (2022) developed a new solar-based CCHP system and utilized the RSM for the multi-objective optimization of its net power, CO_2 emission, and exergy efficiency. In the developed system, waste heat between the compressors was harnessed by an intercooler to power an absorption chiller. By means of interaction effects between the four decision variables, six optimal solutions were obtained and the TOPSIS (technique for order preferences by similarity to ideal solution) method was used to determine the best solution. Optimal results were realized for the net power, CO_2 emission, and exergy efficiency respectively. Table 2.5 gives a summary of some of the reviewed pieces of literature.

Table 2.5: Summary of some related works

S/N	**References**	**Optimization technique**	**Integrated renewable energy type**	**Performance criteria to be optimized**	**System type**
1	Shakibi et al. (2023)	RSM+ MOGWO	Solar	Exergy efficiency, unit cost, and performance coefficient	Tri-generation
2	Asgari et al. (2022)	MOGWO	Solar	Exergy efficiency, net power, and unit product cost	Tri-generation
3	Haghghi et al. (2023)	ANN+MOGWO	Geothermal	Exergy efficiency, investment cost, energy, and levelized cost	Poly-generation
4	Habibollahzade & Houshfar (2020)	MOGWO	Not applicable	Emission, total specific cost, cost rate, and efficiency	Power generation
5	Zhang & Sobhani (2022)	MOGWO	Geothermal	Net power, freshwater production exergy	Co-generation

				efficiency, levelized total emission, and payback period	
6	Azizi, Nedaei & Yari (2022)	MOGWO	Solar	Exergy efficiency and unit cost	Poly-generation
7	Chen, Huang & Shahabi (2021)	MOGWO	Solar	Energy efficiency, energy cost, and CO_2 emission	Tri-generation
8	Behzadi et al. (2021)	MOGWO	Not applicable	Exergy efficiency, total cost rate	Co-generation
9	Nedaei, Azizi & Farshi (2022)	MOGWO	Solar	Exergy efficiency, freshwater production, and unit product cost	Multi-generation
10	Zhang et al. (2022)	MOGWO	Biomass	Exergy efficiency and total cost rate	Co-generation
11	Mahdavi, Mojaver & Khalilarya (2022)	RSM	Solar	Net power, exergy efficiency, and CO_2 emission	Tri-generation

2.8.2 Multi-objective Harris Hawks optimization for power generation systems

Yousri, Babu, and Fathy (2020) proposed the HHO to optimize the parameters of a proportional-integral controller contained in a renewable energy-based system. The system to be optimized for integral time absolute error and tie-line power comprised thermal plants, photovoltaic plants, wind turbines, etc. The results obtained were validated with the greywolf optimizer, multi-verse optimizer, and sine cosine algorithm.

Zhao et al. (2020) presented the optimization of a fuel cell-based CCHP system using the HHO modified by introducing the singer mechanism. The CCHP system was optimized for exergy performance, system performance, greenhouse reduction, and cost. The computed results were evaluated using the regular HHO and the NSGA-II. A lower exergy efficiency as well as a higher greenhouse gas reduction was recorded for the modified HHO technique.

Song, Tan, and Mizzi (2020) proposed a modified HHO for the optimization of a proton exchange membrane fuel cell. The HHO was modified using the quasi-oppositional concept and a logistic map mechanism to improve the speed and robustness of the algorithm. This was applied in three case studies and evaluated by comparison with the krill herd algorithm, grass fibrous root optimization, and a regular HHO algorithm. The aim of minimizing the squared deviation between the output voltage and estimated data was achieved as the proposed approach yielded satisfactory results.

Furthermore, Pandey & Jadoun (2023) examined the best scheduling practices for a virtual power plant that is made up of photovoltaic solar collectors, a wind energy source, a fuel cell, and a CCHP system. A modified HHO was employed to minimize the profit and emission. This optimization technique was improved by increasing the step length of the hawks and introducing a chaotic mapping mechanism for increased diversity of the algorithm. The obtained results were validated with results from the PSO and mixed linear programming.

Selim et al. (2020) suggested the application of the HHO to determine where a distribution generation system should be placed in a radial distribution network. The ‘rabbit location’ and ‘grey relation analysis’ in the single and multi-objective HHO were respectively introduced to minimize the power loss, and voltage deviation and increase the voltage stability index. The proposed approach was implemented on two types of radial distribution systems and compared with related works that employed the PSO and GA among other techniques. The results obtained proved the effectiveness of the proposed HHO algorithm.

Omar et al (2020) employed the HHO and flower pollinated algorithm (MOFPA) for the multi-objective optimization of a system comprising thermal, natural gas, and renewable energy sub-systems. The proposed approaches were implemented in three different case scenarios in a bid to minimize the fuel cost and emissions. The MOHHO with higher emission and low fuel cost was observed to generally outperform the MOFPA.

Mahalekshmi & Maruthupandi (2023) employed the HHO to solve the challenges of economic and emission dispatch in power systems. The proposed method applied a fuzzy-based technique to regulate the size of the Pareto optimal solutions and the overall system was optimized for emission dispatch and total cost. Also, the quality of the obtained results was evaluated using the set spacing, hypervolume, and CM and was compared with results obtained from the PSO technique.

Islam et al. (2019) employed the HHO to solve the issue of optimal flow. The presented approach was utilized to minimize fuel cost and power loss. Its result was seen to outperform those obtained using the bat, butterfly, ALO, whale, salp swarm, moth flame and glow warm optimization techniques.

2.8.3 Multi-objective ant lion optimization for power generation systems

Tung & Chakravorty (2016) proposed the ALO for application in the power dispatch problem as it concerns energy production planning. The proposed optimization algorithm was tested on two system units to minimize fuel cost. Its results were validated with other techniques such as the GWO, cuckoo search (CS), artificial bee colony (ABC), firefly algorithm (FA), etc.

Ali, Elazim, and Abdelaziz (2016) presented the ALO for the best placement and scaling of renewable energy-based distributed generation systems. This method was carried out by obtaining the loss sensitivity factor of the most probable buses followed by an inference of the position and sizes of the distributed generation using the ALO. The proposed algorithm was implemented in two radial distribution system types with the aim of minimizing power loss while maximizing net savings. This method was validated with other techniques such as the GA, CS, PS, etc.

Hadidian-Moghaddam et al. (2018) employed the ALO for proffering solutions to the optimal sizing and placement of distributed generation systems. The proposed approach was utilized to reduce the energy cost, application cost, distribution system losses, and voltage deviation while boosting the reliability improvement. It was implemented on two IEEE networks and its results were validated with the PSO and GA techniques.

Ali et al. (2018) proposed the ALO for the optimal sizing and placement of a solar energy-based distributed generation system. This was implemented on a radial distribution system in which five different weight factors were selected to depict how the objective function affects the size and location of the distributed system. The proposed study achieved its aim of minimizing the power loss and voltage deviation.

Wei and Xv (2022) developed an optimal scheduling model for a building integrated energy system based on the conventional CCHP system. The environmental cost, system operation, and maintenance cost were employed as objective functions to be optimized using a modified ALO. The modified version of the ALO utilized involves the introduction of differential evolution in both the single and multi-objective optimization problems. The proposed method was validated on a real-life building and a reduction in operation cost and CO_2 emission was highlighted.

Dubey, Pandit, and Panigrahi (2016) employed the ALO to solve the challenge involving the scheduling of wind-integrated hydrothermal power generation. Simulation tests using this technique were implemented on four systems. The proposed method yielded satisfactory power cost results when validated with related works of literature.

Abul'Wafa (2019) carried out a study on the optimal placement of distributed generators and condensers employing the ALO. Three case scenarios of the integrated distributed generator and condenser were implemented on a radial distribution generation system with active power loss, reactive power loss, and voltage stability as objective functions which were effectively optimized.

Kaveh and Eslamlou (2020) employed the ALO technique to select an optimal schedule of appliances in a residential building. The Shanon's entropy technique and the evidential reasoning approach were utilized to respectively obtain the weights of the objective functions and rank the obtained Pareto optimal solutions. The electricity cost, peak average ratio, and CO_2 emission were effectively optimized and the proposed method was recommended for smart home applications.

2.9 Summary of the chapter and Research gap

This chapter has provided an extensive overview of the fundamental theoretical concepts and research findings relevant to the optimization of solar-assisted CCHP system systems. The exploration of multi-generation systems, especially CCHP systems, elucidated their classification and subsystems, highlighting their importance in meeting the increasing demands of contemporary society. Furthermore, the integration of solar energy into CCHP systems was discussed, emphasizing its potential to further enhance system performance by reducing the usage of fossil fuels thus reducing GHG emission.

This chapter also reviewed works of literature that have demonstrated the effectiveness of solar-integrated CCHP systems. Also, the introduction of metaheuristic optimization techniques such as the Grey Wolf, Harris Hawks and Ant Lion have shown how a system's design configuration can be improved to maximize efficiency while reducing the adverse impact to the environment. In addition, this literature review highlighted the growing interest in employing these optimization algorithms in multi-generation systems such as the CCHP. It was also seen that, to the best of the author's knowledge, there was no implementation of the HHO and ALO on a solar-assisted CCHP system. Also, the optimization of the solar-assisted CCHP system under investigation using the response surface method yielded relatively lower exergy efficiency and higher CO_2 emission values. Hence, this research aims to build upon the insights gained from existing literature to employ the GWO, HHO and ALO for the optimization of a solar-assisted CCHP system.

Chapter Three – Methodology

3.1 Introduction

This chapter aimed to extensively discuss the methodology followed for the actualization of the research work. It began by describing the solar-assisted combined cooling, heating, and power (CCHP) system targeted for optimization, along with an explanation of its working principle. Additionally, the use of the response surface method, which was employed to develop regression models for obtaining net power, CO_2 emission, and exergy efficiency, was briefly explained. Furthermore, the chapter highlighted the regression models utilized in this study and provided insight into the mathematical formulation of both single and multi-objective optimization problems. Subsequently, the theoretical concepts, inspiration, and methodology for developing mathematical models for the Grey Wolf Optimization (GWO), Harris Hawks Optimization (HHO), and Antlion Optimization (ALO) techniques were comprehensively explained. The associated MATLAB codes for implementing the methodology were presented in Appendices A to F, to allow for replication and understanding of the optimization process.

3.2 System description

This research study will adopt the solar-integrated CCHP system presented by Mahdavi, Mojaver & Khalilarya (2022). The system, shown in Fig 3.1, comprises three gas turbines, two compressors, a Kalina cycle, an absorption chiller, solar collectors, a heat recovery steam generator unit (HSRG), and a hot water generator.

The system operated in such a way that air was compressed by two-stage compressors with an intercooler in between them. Waste heat from the intercooler was used to drive the absorption chiller via the transfer of heat in a desorber. Following that, the air leaving the second compressor obtained heat energy through the air preheater-1(APH-1) and air preheater-2 (APH-2) powered by solar energy and exhaust combustion gases from gas turbine-3(GT-3) respectively. Once the air was up to the required temperature and pressure, it reacted with combustion fuel (methane) to produce hot gases in the combustion chamber. The hot gases flowed into three consecutive gas turbines where the first two turbines supplied the power needed to drive the compressors while the third turbine was responsible for rotating the shaft of a generator to generate electrical power. Furthermore, exhaust gases from GT-3 were recovered back to the APH-2 and were used to provide heating power to the HSRG and Kalina cycle through the evaporator-1 (Eva-1) as well as the domestic water heat exchanger respectively before being released to the environment.

In the Kalina cycle, the pump-1 increased the pressure of its working fluid (ammonia water mixture) which was then passed into two successive heat exchangers (low-temperature recuperator (LTR) and high-temperature recuperator (HTR)) to improve its thermal energy hence reducing the energy input to Eva-1. The resulting two-phase mixture at Eva-1 was sent to the separator where separation into saturated vapor and saturated liquid occurred. The saturated vapor was supplied to the steam turbine to generate further work output while the saturated liquid was sent back to the HTR to recover some thermal energy before being passed to the expansion valve-1 (Ev-1) where its pressure was reduced. Subsequently, the low-pressure saturated liquid entered the mixer where it mixed with the output from the steam turbine. This mixture was passed through the LTR to dissipate its energy before being discharged to the atmosphere through the condenser (cond-1) hence completing the cycle.

A series of processes were employed in the single-effect absorption chiller to acquire energy at the desorber (provided by rejected gases from the intercooler) used to supply cooling capacity at the associated evaporator (Eva-2). The working fluid (saturated LiBr-H_2O liquid) was pumped in pump-2 before its entrance into the solution heat exchanger (SHE) where it gained thermal energy. In the water vapor state, the working fluid entered the condenser (cond-2) from the desorber and it condensed into saturated liquid before being throttled from the expansion valve (Ev-3) into Eva-2. At Eva-2, it gained energy to become a saturated vapor before being absorbed into the solution in Ev-2 where it transformed back to the saturated liquid specified at the start of the cycle.

The following assumptions were made - that the system operated in a steady state; kinetic and potential energy changes were negligible; heat losses from component systems, except the combustion chamber, were insignificant.

3.3 Response surface method

This is a mathematical and statistical approach that aims to determine the design factor settings to enhance the accurate implementation of a procedure (Costa et al. 2021). The regression models generated portray the relationship between a certain response variable and the associated design factors. The general procedures undertaken are:

- Design of experiments: This is carried out to establish the experimental conditions. It involves selecting the relevant input factors that would affect the response variable. This is followed by the determination of the constraints used to evaluate the design factors during the experiment.

- Experimental tests: Here, the necessary experiments are performed employing an already prepared experimental plan and the response variable data are collected according to the various fusion of the design factor levels. These tests are arbitrarily conducted to reduce the influence of unimportant design factors.
- Fitting the Regression models: The regression models are fitted employing the data obtained from the experiments using methods such as the least squares or the maximum likelihood estimation. The resulting regression models are evaluated for their goodness-of-fit to inspect for any discrepancies from the initial model presumptions.
- Validation of the regression model: After the model is successfully fitted, it is validated through prediction using further experimental tests with unused data.

Based on the described system in section 3.1, the response surface methodology was utilized by Mahdavi, Mojaver & Khalilarya (2022) to develop regression models capable of predicting its net power, CO_2 emission, and exergy efficiency. The decision variables, equations, and constraints are shown in table 3.1, equation 3.10-3.30, and table 3.2 respectively.

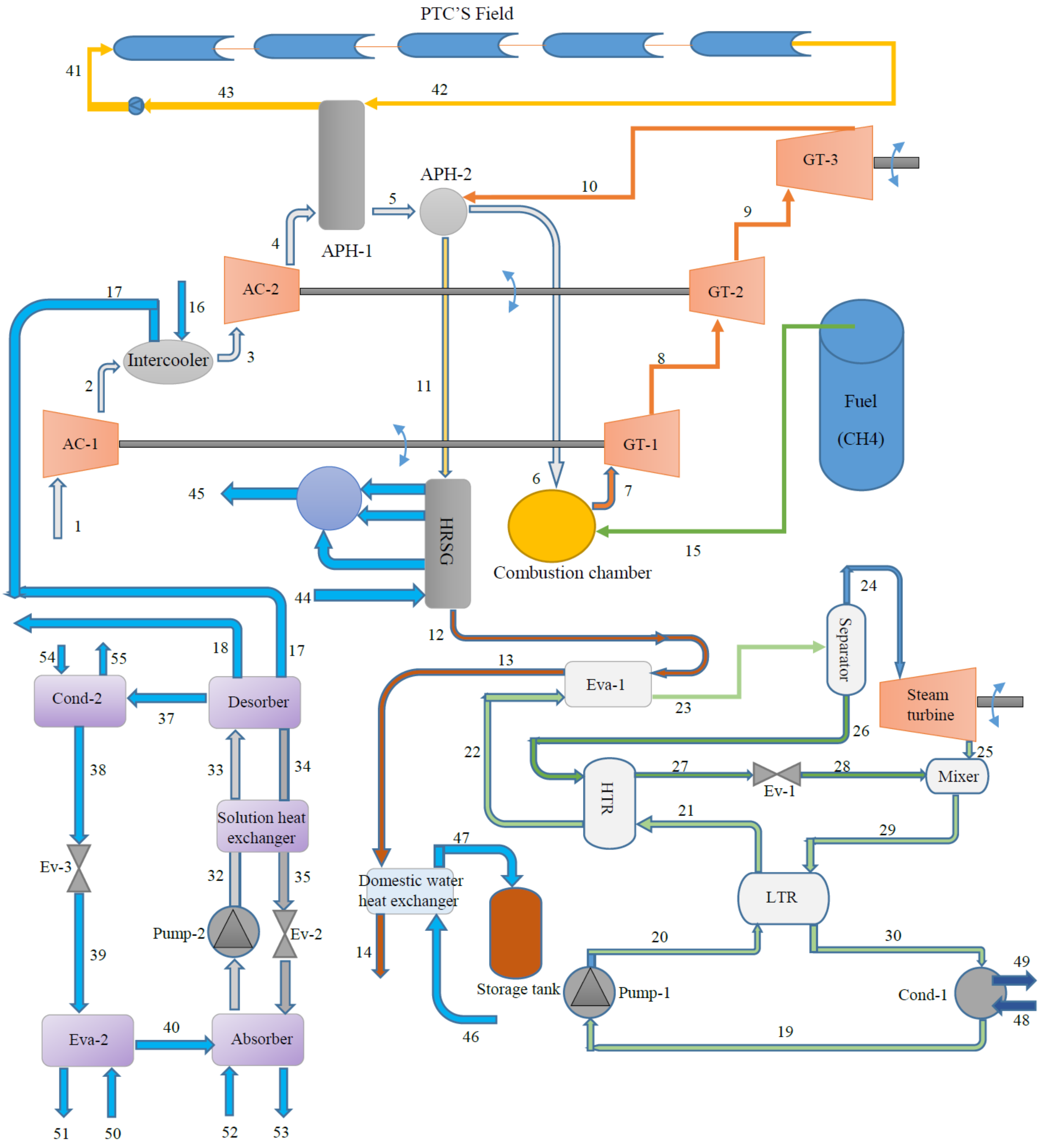


Fig 3.1: Diagram illustrating the solar-integrated CCHP system

This study considers three objective functions namely:

1. **Net Power Output**: The net power, which is a function of the energy analysis, is the summation of the work outputs from the gas turbine and Kalina cycle. Mathematically, it can be expressed as (Mahdavi, Mojaver & Khalilarya, 2022):

$$\dot{P}_{net} = \dot{P}_{net,GT} + \dot{P}_{net,KC} \tag{3.1}$$

$$\dot{P}_{net,GT} = \left(\dot{P}_{GT-1} + \dot{P}_{GT-2} + \dot{P}_{GT-3}\right) - (\dot{P}_{AC-1} - \dot{P}_{AC-2}) \tag{3.2}$$

$$\dot{P}_{net,KC} = \dot{P}_{ST} - \dot{P}_{pump-2} \tag{3.3}$$

$$\dot{P}_{GT-1} = \dot{m}_p(h_7 - h_8) = \dot{P}_{AC-1} \tag{3.4}$$

$$\dot{P}_{GT-2} = \dot{m}_p(h_8 - h_9) = \dot{P}_{AC-2} \tag{3.5}$$

$$\dot{P}_{GT-3} = \dot{m}_p(h_9 - h_{10}) \tag{3.6}$$

where,

$\dot{P}_{net}$= net power output

Kalina $\dot{P}_{net,GT}, \dot{P}_{net,KC}$= net power from the gas turbines and kalina cycle respectively,

$\dot{P}_{GT-1}, \dot{P}_{GT-2}, \dot{P}_{GT-3}$= net power from gas turbine 1, 2, and 3 respectively,

$\dot{P}_{AC-1}, \dot{P}_{AC-2}$= net power to compressor 1 and 2 respectively.

$\dot{P}_{ST}, \dot{P}_{pump-2}$= net power from steam turbine and pump2 of the Kalina cycle,

h_7, h_8, h_9, h_{10}= specific enthalpies at state 7, 8, 9, and 10,

$\dot{m}_p$= mass flow rate of the combustion gases from the combustion chamber

2. **Exergy Efficiency**: According to Kumar, (2017), it is a practical and effective criterion for determining the type, extent, and positions of irreversibilities in a thermodynamic system. Mathematically, it can be defined as the quotient obtained by dividing the output exergy by the input exergy.

$$\mathcal{E} = \frac{\dot{P}_{net} + \left(\dot{E}_{45} - \dot{E}_{44}\right) + \left(\dot{E}_{50} - \dot{E}_{51}\right) + (\dot{E}_{47} - \dot{E}_{46})}{\dot{E}_{in}} \tag{3.7}$$

$$\dot{E}_{in} = \dot{E}_{fuel} + \dot{E}_{coll} \tag{3.8}$$

$where,$

$\dot{E}_{44}, \dot{E}_{45,} \dot{E}_{46}, \dot{E}_{47}, \dot{E}_{50}, \dot{E}_{51}$ = Exergy at state 44, 45, 46, 47, 50 and 51,

$\dot{E}_{in}$ Input exergy

$\dot{E}_{fuel}$ = Exergy of fuel

$\dot{E}_{coll}$ = Exergy of solar collector

3. **CO_2 Emission**: The rejection of CO_2 into the atmosphere has detrimental effects on the environment and its continuous mitigation should be the goal in thermal energy systems. A measure of the production level of $C0_2$ is called Emission and is defined as the ratio of the mass flow rate of $C0_2$ to the total output energy (Mahdavi & Khalilarya, 2019).

$$Emission = \frac{\dot{m}_{CO_2}}{\dot{P}_{net} + \dot{Q}_{heating} + \dot{Q}_{cooling}} \tag{3.9}$$

$where,$

$\dot{Q}_{heating}$ and $\dot{Q}_{cooling}$ = Heating and cooling loads of the CCHP system

Table 3.1: Decision variables

Decision variable	**Symbol**
Compression ratio	Cr
Pinch point temperature difference	Pp
Inlet turbine temperature	Gt
Inlet combustion chamber temperature	Ct

$$\begin{aligned}\dot{P}_{net}\ (MW) &= 62.19 + 0.4573Cr + 0.0259Pp - 0.02421Gt + 0.03638Ct \\ &- 0.010867Cr \times Cr - 0.000029Pp \times Pp + 0.000005Gt \times Gt \\ &- 0.000009Ct \times Ct - 0.0003Cr \times Pp + 0.00022Cr \times Gt \\ &- 0.00042Cr \times Ct - 0.000005Pp \times Gt - 0.000005Pp \times Ct \\ &- 0.000003Gt \times Ct\end{aligned} \tag{3.10}$$

$$\begin{aligned}&Emission\ \left({}^{gr}/_{MJ}\right) \\ &= 13.1 + 3.722Cr + 0.2003Pp - 0.0122Gt + 0.0451Ct \\ &- 0.03047Cr \times Cr + 0.0000296Pp \times Pp + 0.00004Gt \times Gt \\ &+ 0.000052Ct \times Ct + 0.0049Cr \times Pp - 0.00294Cr \times Gt \\ &+ 0.00286Cr \times Ct - 0.000285Pp \times Gt + 0.000285Pp \times Ct \\ &- 0.000099Gt \times Ct\end{aligned} \tag{3.11}$$

$$\begin{aligned}\mathcal{E}\ (\%) &= -29 - 0.36Cr + 0.287Pp + 0.0659Gt + 0.0133Ct - 0.01807Cr \times Cr \\ &- 0.000029Pp \times Pp - 0.000011Gt \times Gt - 0.000009Ct \times Ct \\ &- 0.0125Cr \times Pp + 0.0003Cr \times Gt + 0.00086Cr \times Ct \\ &- 0.000205Pp \times Gt + 0.000195Pp \times Ct - 0.000009Gt \times Ct\end{aligned} \tag{3.12}$$

Table 3.2: Decision variables and constraints

Decision variable	Constraint
Compression ratio	10≤ Cr≤15
Pinch point temperature difference	10≤ Pp ≤30
Inlet turbine temperature	1420≤ Gt≤1520
Inlet combustion chamber temperature	850≤ Ct≤950

3.4 Mathematical Formulation

The optimization of net power can be expressed mathematically in the following way:

$$\text{Maximize } \mathcal{F} = \{P_{net}(Cr, Pp, Gt, Ct)\} \tag{3.13}$$

Similarly, the single-objective optimization of CO_2 emission can be formulated as follows:

$$\text{Maximize } \mathcal{F} = \{(-\text{emission (Cr, Pp, Gt, Ct)}\} \qquad (3.14)$$

The following formula can be used to formulate the single-objective optimization of exergy efficiency:

$$\text{Maximize } \mathcal{F} = \{\ \mathcal{E}\ \text{(Cr, Pp, Gt, Ct)}\} \qquad (3.15)$$

The formulation of the multi-objective optimization problem can be expressed as follows:

$$\text{Maximize } \mathcal{F} = \{P_{net}(\text{Cr, Pp, Gt, Ct}),\ \mathcal{E}(\text{Cr, Pp, Gt, Ct}),\ -\text{emission}(\text{Cr, Pp, Gt, Ct})\} \qquad (3.16)$$

Equations (4.1) – (4.4) are subject to variable restrictions:

$10 \leq \text{Cr} \leq 15$

$10 \leq \text{Pp} \leq 30$

$1420 \leq \text{Gt} \leq 1520$

$850 \leq \text{Ct} \leq 950$.

3.5 Greywolf optimization

The optimization technique employed in this study, called greywolf optimization (GWO), is a swarm intelligence algorithm proposed by Mirjalili, Mirjalili & Lewis (2014). It draws inspiration from the behavior, attacking method, social leadership, and encircling process of the wolf to determine the best solution for an optimization problem. The structuring of the GWO is such that the fittest solution is named the alpha (α) in order to reflect the social ranking of wolves. As a result, beta (β) and delta (δ) refer respectively to the next best solutions while the remaining solutions are called omega (ω) wolves. The α, β, and δ wolves pilot the hunting activity, with the ω wolves trailing them, in their search for the global optimum. The procedures for the grey wolf hunting activity according to Muro et al. (2011) are as follows:

- Tracking, pursuing, and getting close to the prey.
- Chasing, surrounding, and harassing the prey until its movement is halted.
- Launching an attack on the prey.

The following mathematical equations, presented in Mirjalili, Mirjalili & Lewis (2014), are used to initiate the encircling of a prey by the grey wolves during hunting,

$$\vec{M} = \left|\vec{Q} \cdot \vec{N}_p(t) - \vec{N}(t)\right| \qquad (3.17)$$

$$\vec{N}(t+1) = \vec{N}_p - \vec{H} \cdot \vec{M} \qquad (3.18)$$

where,

$t = current\ iteration$

$\vec{H}\ and\ \vec{Q} = coefficient\ vector$

$\vec{N} = position\ vector$

$\vec{N}_p = position\ vector\ vector\ of\ the\ prey$

Furthermore, the coefficient vectors, $\vec{H}\ and\ \vec{Q}$ are calculated as thus:

$$\vec{H} = 2 \cdot \vec{h} \cdot \vec{r}_1 - \vec{h} \tag{3.19}$$

$$\vec{Q} = 2 \cdot \vec{r}_2 \tag{3.20}$$

Where the components of $decline$ from 2 to 0, in a linear manner, across the iterations while $\vec{v}_1\ and\ \vec{v}_2$ are randomly selected vectors in [0, 1].

3.5.1 Single-objective optimization

The GWO algorithm commences optimization by producing a random solution set. The top three solutions generated are saved by the algorithm which requires the other search agents to adjust their locus about the optimum solutions. After the end condition has been met, the location and value of the alpha solution becomes the optimum solution. The following formula is continuously run, during the optimization process, for each search agent to initiate the hunting process and identify potential areas of the search space (Mirjalili, Mirjalili & Lewis, 2014). The flowchart and pseudo code for the GWO technique are given in Figure 3.2.

$$\vec{M}_\alpha = \left|\vec{Q}_1 \cdot \vec{N}_\alpha - \vec{N}\right| \tag{3.21}$$

$$\vec{M}_\beta = \left|\vec{Q}_1 \cdot \vec{N}_\beta - \vec{N}\right| \tag{3.22}$$

$$\vec{M}_\delta = \left|\vec{Q}_1 \cdot \vec{N}_\delta - \vec{N}\right| \tag{3.23}$$

$$\vec{N}_1 = \vec{N}_\alpha - \vec{H}_1 \cdot (\vec{M}_\alpha) \tag{3.24}$$

$$\vec{N}_2 = \vec{N}_\beta - \vec{H}_2 \cdot (\vec{M}_\beta) \tag{3.25}$$

$$\vec{N}_3 = \vec{N}_\delta - \vec{H}_1 \cdot (\vec{M}_\delta) \tag{3.26}$$

$$\vec{N}(t+1) = \frac{\vec{N_1} + \vec{N_2} + \vec{N_3}}{3} \qquad (3.27)$$

Equations 3.21 to 3.27 cannot be solved mathematically hence a mathematical programming model was developed in the MATLAB environment and was presented in Appendix A. The developed model comprised:

1. Initialization of solutions: The initial population of solutions termed 'search agents' were initialized at a value of 100. Also, an initialization function was created to ensure that the randomly initialized search agents do not exceed the set boundaries of the decision variables. The associated codes for this operation were reported in Appendix A.1.
2. Problem definition: The 'Get_functions_details.m' file was created to define the values of the upper and lower boundaries of the decision variables and the objective equations. The associated codes for this operation were reported in Appendix A.1.
3. Evaluation: The fitness/objective values of the search agents were computed by passing all probable solutions as arguments to the objective functions and the best three search agents were recorded as alpha, beta and delta. These three best search agents were updated in each iteration using equations 3.21 to 3.27 provided the termination criteria was not attained. The algorithm for this operation was presented Appendix A.2. When the maximum number of iterations was reached, the best search agent and the corresponding objective value were presented as the optimal solutions for the single objective optimization problem.

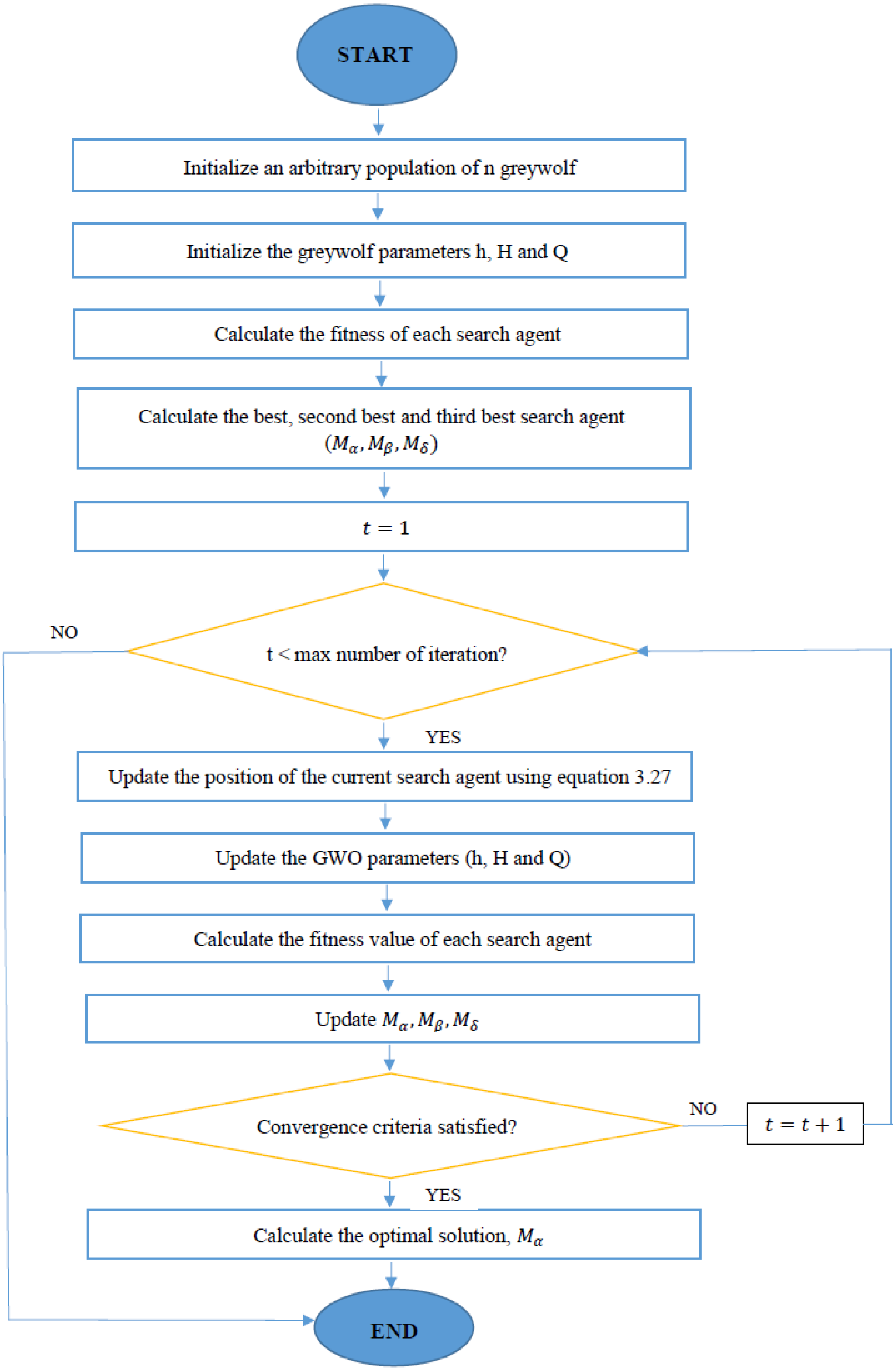

START
Initialize an arbitrary population of n greywolf
Initialize the greywolf parameters h, H and Q
Calculate the fitness of each search agent
Calculate the best, second best and third best search agent $(M_\alpha, M_\beta, M_\delta)$
$t = 1$
t < max number of iteration?
NO
YES
Update the position of the current search agent using equation 3.27
Update the GWO parameters (h, H and Q)
Calculate the fitness value of each search agent
Update $M_\alpha, M_\beta, M_\delta$
Convergence criteria satisfied?
NO
$t = t + 1$
YES
Calculate the optimal solution, M_α
END

Fig 3.2: Flowchart for single-objective grey wolf optimization

3.5.2 Multi-objective Greywolf Optimization

There are situations where objectives are conflicting in nature hence, we employ multi-objective optimization that involves simultaneously optimizing more than one objective function to generate a set of alternative solutions that are feasible with a compromise between the solutions known as Pareto optimal or non-dominated solutions. The flowchart for the multi-objective GWO technique is given in Figure 3.3.

Additional components were incorporated into the mathematical programming model of the single-objective optimization to yield the model for the multi-objective greywolf optimization. The developed model was shown in Appendix B, and it comprised the following:

1. Problem definition: All three objective functions were defined in the 'Ceco9.m' file. Also, the upper and lower boundaries of the decision variables were defined in the 'MOGWO.m' file. The grid inflation and leader selection parameters as well as the grids per dimension were also defined. This was presented in Appendix B.1.
2. Initialization: The initial population of solutions termed 'Greywolves_num' were initialized at a value of 100. An archive size of 100 was also initialized to store the non-dominated Pareto optimal solutions. A fundamental section of this archive was the archive controller that regulated the archive to decide whether a new solution would be allowed or not and ascertained when the archive was filled up. The associated codes for this operation were presented in Appendix B.2.
3. Iteration improvement: The leader selection approach and grid mechanism were employed to assist in the selection of the best three solutions (alpha, beta and delta), from the archive, as heads of the search activity and maintain diversity of the archive. In this approach as shown in the 'SelectLeader.m' and 'MOGWO.m' presented in Appendix B.2, the top three GWO solutions per iteration directed the remaining solutions in the archive to the 'hunting' space area to obtain results that are as close as possible to the global solution. The Pareto optimality concept hinders the comparability of the obtained solutions hence the leader selection approach managed the issue by selecting from sparsely populated regions of the search area offering with alpha, beta, or delta wolves as one of its non-dominated solutions.
4. Evaluation: The evaluation process was carried out such that during each iteration process, a new solution is added to the archive if it dominated one or more of the archive resident solutions or if neither the newly obtained solution nor the archive resident solutions

dominated each other. However, a newly obtained solution was not granted entry into the archive if it was dominated by at least one of the resident solutions in the archive. The algorithms to check for domination and update the archive are presented in Appendix B.2. For each non-dominated solution obtained using equations 3.21 to 3.27, the objective values were computed, and this was updated for each iteration until the maximum number of iterations was attained.

5. Selection: The roulette wheel approach was applied to select the leader of the GWO and the probability of choice for each hypercube is given by equation 3.28.

$$P_i = \frac{c}{A_i} \tag{3.28}$$

$where,$

$c = constant\ value$

$A_i = number\ of\ pareto\ optimal\ solutions\ in\ the\ ith\ segment$

The roulette wheel ensured that there was a low probability of selecting leaders from the most populated hypercubes hence introducing some element of randomness into the algorithm. This algorithm was presented in 'Roulettewheelselection.m' and 'CreateHypercubes.m' files shown in Appendix B.2.

6. Data presentation: When the maximum number of iterations was attained, the non-dominated solutions were used to obtain the corresponding objective values and the Pareto fronts were presented. The algorithm for this operation was presented in the 'MOGWO.m' file as shown in Appendix B.2.

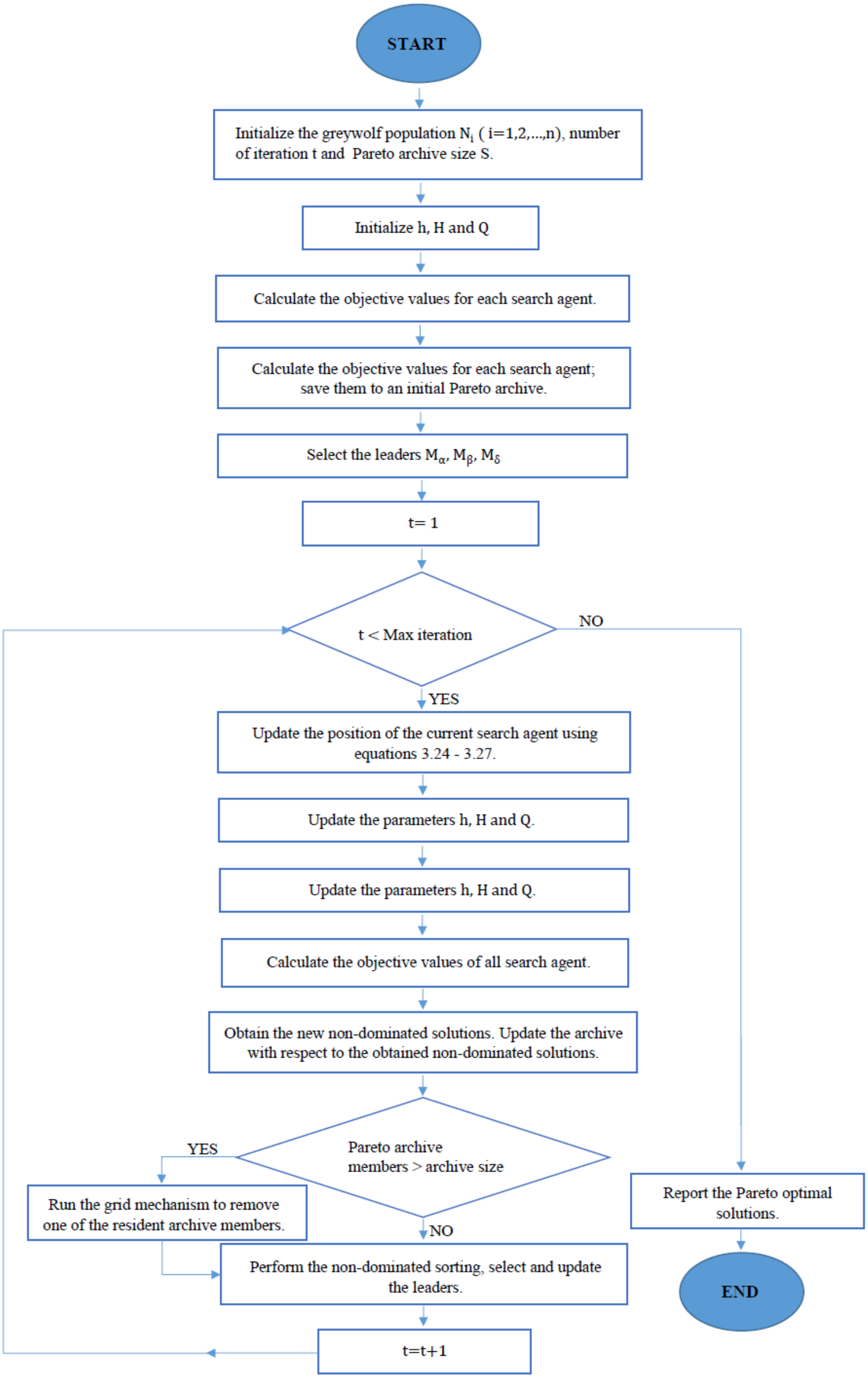

START
Initialize the greywolf population N_i (i=1,2,...,n), number of iteration t and Pareto archive size S.
Initialize h, H and Q
Calculate the objective values for each search agent.
Calculate the objective values for each search agent; save them to an initial Pareto archive.
Select the leaders M_α, M_β, M_δ
t= 1
t < Max iteration
NO
YES
Update the position of the current search agent using equations 3.24 - 3.27.
Update the parameters h, H and Q.
Update the parameters h, H and Q.
Calculate the objective values of all search agent.
Obtain the new non-dominated solutions. Update the archive with respect to the obtained non-dominated solutions.
Pareto archive members > archive size
YES
Run the grid mechanism to remove one of the resident archive members.
NO
Perform the non-dominated sorting, select and update the leaders.
t=t+1
Report the Pareto optimal solutions.
END

Fig 3.3: Flowchart for multi-objective grey wolf optimization

3.6 Harris Hawk Optimization

This optimization technique is inspired by one of the smartest birds, usually located in South Arizona USA, known as Harris' hawk (Bednarz, 1988). These hawks have a peculiar behavior of carrying out coordinated exploration, attacking, and feeding off the prey while in groups. They mainly employ the 'surprise pounce' or 'seven kills' tactics to catch their prey. This involves a situation where a group of hawks, emerging from different angles, simultaneously converge on the fleeing prey. The time taken to capture the prey ultimately depends on its evasion ability and actions. During this time interval, the hawk displays several attacking styles that may comprise multiple, brief, and swift dives close to the prey over several minutes. The fundamental benefit of these tactics is to exhaust the prey thereby making it more vulnerable as well as confuse it so that it is unable to regain its defensive ability. Subsequently, the hawk with the most experience and strength finally captures the worn-out prey which is then distributed among the other group members (Heidari et al. 2019).

The harris hawk optimization (HHO), proposed by Heidari et al. (2019), is a common nature-inspired meta-heuristic algorithm that mimics the exploration and attacking behavior of the harris hawk. Figure 3.4 shows the stages of the HHO in capturing the prey and can be modeled using the following equations.

$A = 2A_0(1 - \frac{r}{R})$ (3.29)

$A_0 = 2t_1 - 1$ (3.30)

Where,

A= flight energy of the prey

r= number of initial repetitions

R= number of maximum repetitions

A_0= initial flight energy of the prey and is a random number in [-1, 1]

t_1= random value in [0, 1]

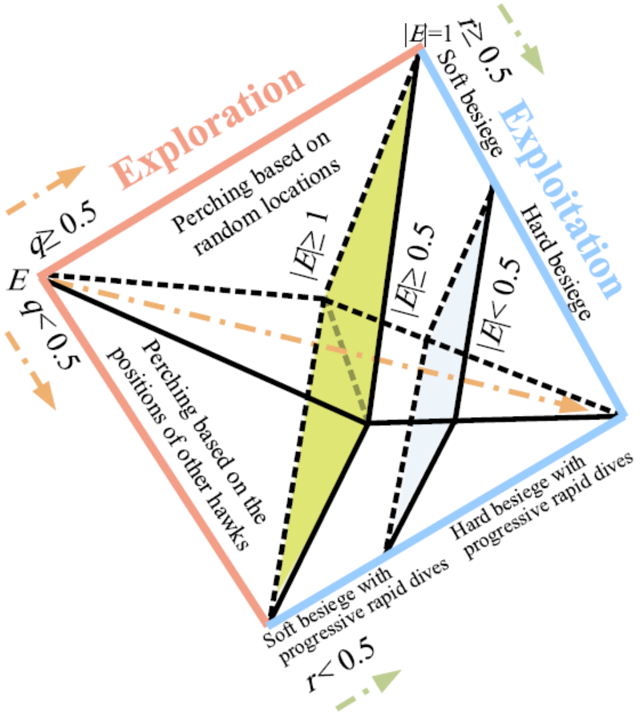

|E|=1
r≥0.5
Soft besiege
Exploitation
Hard besiege
Exploration
Perching based on random locations
q≥0.5
E
q<0.5
Perching based on the positions of other hawks
|E|≥1
|E|≥0.5
|E|<0.5
Soft besiege with progressive rapid dives
Hard besiege with progressive rapid dives
r<0.5

Fig 3.4: Phases of the HHO (adapted from Heidari et al. 2019)

The value of A steadily reduces with each iteration since it is correlated with the prey's flight energy. The prey is said to be physically exhausted when A_0 reduces from 0 to -1 and strengthened when A_0 is increased from 0 to 1. Hence, when $A \geq 1$, the hawks are still in the process of searching in various areas for the prey's whereabouts and this is called the exploration phase. This process is transitioned to the exploitation phase when $A < 1$ indicating that the hawks are prepared to capture the prey. These stages will be further explained in the following section.

3.6.1 Exploration stage

This is the first stage of the hunting process whereby the hawks (also known as candidate solutions) monitor, track, and locate a prey (also known as target) using their powerful eyes. In reality, it may take quite several hours to discover a prey. The likelihood of discovering the target solution is dependent on the candidate solution and the best solution in each step is regarded as the targeted prey or estimated optimum. The Harris' hawks wait for their prey in two ways. They either wait in places where their group members will be nearby at the time of the capture or wait in arbitrary places such as tall trees. If an equal likelihood, t for each waiting strategy is considered, $m < 0.5$

depicts the condition for the first waiting method while $m \geq 0.5$ depicts the condition for the second waiting method. Both methods are modeled in equation 3.31.

$$Y(r+1) = \begin{cases} Yrand(r) - t_2|Yrand(r) - 2t_3Y(r)| & m \geq 0.5 \\ \left(Y_{prey}(r) - Y_n(r)\right) - t_4\left(LB + t_5(UB - LB)\right) & m < 0.5 \end{cases} \tag{3.31}$$

Where,

$Y(r+1)$= new position of the hawks based on r repetitions

Y_{prey}= position of the prey

Y_n= average position of the current hawk's population

$Yrand(r)$= randomly chosen hawk from the hawk's population

$Y(r)$= current position of the hawk

t_2, t_3, t_4, t_5, m= random values in (0, 1) updated in each repetition, r.

LB= lower bund of the variable

UB= upper bound of the variable

The average location of the hawk is obtained using equation 3.32 as follows:

$$Y_n(r) = \frac{1}{N}\sum_{i=1}^{N} Y_i(r) \tag{3.32}$$

Where,

$Y_i(r)$= position of each hawk in repetition r.

N= total number of hawks.

3.6.2 Exploitation stage

This is the stage that follows after the exploration where the hawks carry out the 'surprise pounce' tactics on the detected prey. In nature, the prey tries to flee and this prompts several chasing styles by the predator. Based on the evasion behavior of the prey and the attacking method of the Harris' hawks, four potential strategies are being modeled by the HHO during this exploitation stage. r is assumed to be the likelihood of the prey fleeing successfully ($t < 0.5$) or not fleeing successfully ($t \geq 0.5$). Irrespective of the escape mechanism of the prey, the hawk would carry out a hard or soft besiege to capture the prey. This implies that they would surround the prey from several angles softly or hard according to the prey's energy, A. Modelling this with the HHO, when$|A| \geq 0.5$, a soft besiege occurs while a hard besiege takes place when$|A| < 0.5$.

3.6.2.1 Soft besiege

In this situation when $t \geq 0.5$ and$|A| \geq 0.5$, the prey still possesses sufficient energy and attempts to flee through arbitrary confusing leaps but ultimately fails. This is because the harris' hawks softly surround the prey, facilitating its vulnerability, before performing the surprise pounce. This strategy is modeled in equation 3.33 as follows:

$$Y(r+1) = \Delta Y(r) - A\left|\mathcal{J}Y_{prey}(r) - Y(r)\right| \qquad (3.33)$$

$$\Delta Y(r) = Y_{prey}(r) - Y(r) \qquad (3.34)$$

$$\mathcal{J} = 2(1 - t_5) \qquad (3.35)$$

Where,

$\Delta Y(r)$=difference between the prey’s position and the current location in repetition, r

t_5= random value in (0, 1)

$\mathcal{J}$= prey’s random leap strength which changes arbitrarily to mimic the prey’s movement

3.6.2.2 Hard besiege

In this situation when $t \geq 0.5$and$|A| < 0.5$, the prey is extremely tired and possesses little escape energy. Also, the Harris’ hawks rarely surround the targeted prey to carry out the surprise pounce. For this strategy, equation 3.36 shows the updated current positions.

$$Y(r+1) = Y_{prey}(r) - A|\Delta Y(r)| \qquad (3.36)$$

3.6.2.3 Soft besiege with progressive rapid dives

This presents a tricky situation when $|A| \geq 0.5$and $t < 0.5$ which indicates that the prey still possesses enough energy to escape, and a soft besiege is performed by the harris’ hawks. In addition, the hawks perform quick dives in an attempt to distract and cause the prey to change its path. Akin to a hawk’s natural behavior, the levy flight (LF) method employed in the HHO technique can be used to determine the best probable dive toward the prey when they intend to capture it in challenging situations. The rule in equation 3.37 is implemented by the Hawks to decide their next move during the soft besiege.

$$W = Y_{prey} - A\left|\mathcal{J}Y_{prey}(r) - Y(r)\right| \qquad (3.37)$$

The probable outcome from such a movement is compared with the previous dive to determine whether it will be a good dive or not. If the hawks observe that the prey performs deceptive movements and is about to escape, they increase the irregularity and rapidness of their dives. These new dives would be based on the LF method as shown in equation 3.38.

$$Z = W + Q \times LF(G) \qquad (3.38)$$

Where,

G= dimension of the problem

Q= arbitrary vector of size $1 \times D$

LF= levy flight function

LF is obtained using equation 3.39 as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}} \tag{3.39}$$

$$\sigma = (\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2(\frac{\beta-1}{2})})^{1/\beta} \tag{3.40}$$

Where,

u, v= random numbers in (0, 1)

β= fixed parameters set to 1.5

Therefore, the mathematical model for updating the position of the Harris' hawk in the soft besiege stage is illustrated in equation 3.41 as follows:

$$Y(r+1) = \begin{cases} W & if\ F(W) < F(Y(r)) \\ Z & if\ F(Z) < F(Y(r)) \end{cases} \tag{3.41}$$

Where W & $Z$are gotten from equations 3.37 and 3.38

3.6.2.5 Hard besiege with progressive rapid dives

This situation occurs when $t < 0.5$ and $|A| < 0.5$ whereby the prey does not have enough energy to escape, and a hard encircling is performed by the hawk, before the surprise pounce, to capture the prey. Unlike the soft besiege, the hawks minimize the distance between their average position and the fleeing prey. Equation 3.42 depicts the use of hard besiege to update the hawk's position.

$$Y(r+1) = \begin{cases} W & if\ F(W) < F(Y(r)) \\ Z & if\ F(Z) < F(Y(r)) \end{cases} \tag{3.42}$$

Where $Y_n(r)$, W & $Z$are gotten from equations 3.32, 3.37, and 3.38

3.6.3 Single-objective optimization

Based on the above-mentioned equations, a mathematical model was developed, and it comprised the following components.

1. Problem definition: The 'Get_functions_details.m' file was created as shown in Appendix C.1 to define the upper and lower and boundaries of the decision variables and the objective equations.
2. Initialization: The initial population of solutions were initialized to a size of 100. The initialization function was created to ensure that the initialized values do not exceed the

defined boundaries of the decision variables. These algorithms are presented in Appendix C.2.

3. Evaluation: The prey's energy calculated using equations 3.29 and 3.30 determined if exploration (equation 3.31) or exploitation (equations 3.33 to 3.42) was employed by the HHO algorithm. This was modelled using the codes presented in Appendix C.3. For any of the two methods employed, the prey's energy was employed to calculate the fitness value of the hawks (objective value) while the prey's location was regarded as the solution for the decision variable. The prey's location and hawks' fitness value were updated at each iteration until the maximum number of iterations was attained. The prey's location at the final iteration became the optimal decision variable solution returned by the algorithm and the corresponding objective value was obtained.

The flowchart for the HHO is shown in Fig 3.5

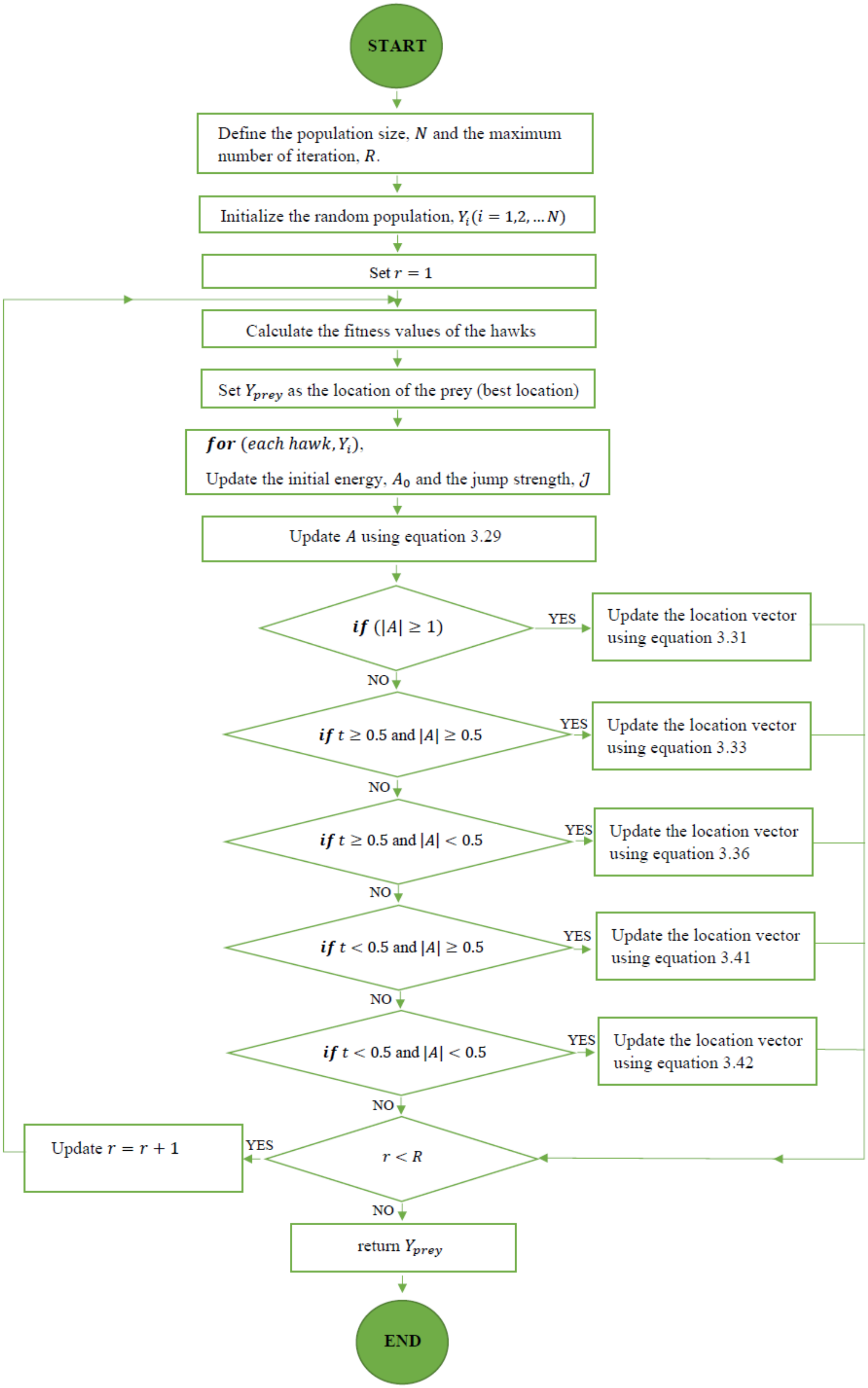

START
Define the population size, N and the maximum number of iteration, R.
Initialize the random population, $Y_i (i = 1,2, \dots N)$
Set $r = 1$
Calculate the fitness values of the hawks
Set Y_{prey} as the location of the prey (best location)
$\boldsymbol{for}$ $(each\ hawk, Y_i)$,
Update the initial energy, A_0 and the jump strength, $\mathcal{J}$
Update A using equation 3.29
$\boldsymbol{if}$ $(|A| \geq 1)$
YES
Update the location vector using equation 3.31
NO
$\boldsymbol{if}$ $t \geq 0.5$ and $|A| \geq 0.5$
YES
Update the location vector using equation 3.33
NO
$\boldsymbol{if}$ $t \geq 0.5$ and $|A| < 0.5$
YES
Update the location vector using equation 3.36
NO
$\boldsymbol{if}$ $t < 0.5$ and $|A| \geq 0.5$
YES
Update the location vector using equation 3.41
NO
$\boldsymbol{if}$ $t < 0.5$ and $|A| < 0.5$
YES
Update the location vector using equation 3.42
NO
$r < R$
YES
Update $r = r + 1$
NO
return Y_{prey}
END

Fig 3.5: Flowchart for the single-objective Harris hawks optimization.

3.6.4 Multi-objective Harris Hawk Optimization

The multi-objective Harris hawks optimization (MOHHO) is inspired by the multi-objective grey wolf technique to perform organized hunting by the Harris' hawks. The archive repository and roulette wheels method are incorporated into the MOHHO. The non-dominated Pareto solutions are stored in the archive. During each iteration of the process, the non-dominated solutions obtained so far are compared with the resident solutions before the archive is being updated. When the archive becomes full, the optimal solutions having the most crowded region are eliminated to create space for the storage of the new solutions. The likelihood of eliminating a solution from the archive is given by:

$$P_i = \frac{N_i}{c}, \ c > 1 \qquad (3.43)$$

Where,

c= constant value,

N_i= number of solutions around the i^{th} solutions

To develop the mathematical model for the multi-objective HHO, additional components were integrated into the model for single-objective HHO. The model comprised the following:

1. Problem definition: All three objective functions were defined in the 'CostFunction.m' file. Also, the upper and lower boundaries of the decision variables were defined in the 'main_MOHHO.m' file. Additionally, parameters such as the inertia weight, global learning coefficient, inflation rate, leader selection pressure, mutation rate, etc., were defined. The associated codes were presented in Appendix D.1.
2. Initialization: The initial population size and number of iterations were both initialized at a value of 100. An archive/repository size of 100 was also initialized to store the non-dominated solutions. The algorithm for this operation was presented in Appendix D.1.
3. Iteration improvement: The leader selection approach was employed in this algorithm to maintain the diversity of the repository as depicted in the 'leader' function presented in Appendix D.2. The leader selection pressure parameter (beta) and the non-dominated solution were passed as arguments to the 'leader' function and the yielded the fittest non-dominated solution as output. This output was subsequently employed to guide the algorithm in performing exploration and exploitation. The roulette wheel was employed to

select the leader of the MOHHO where each solution's probability of being selected as a leader depended on their fitness value.

4. Evaluation: During the evaluation process, the Pareto dominance was used to determine the domination of the solutions. For each iteration, a new solution was added to the archive if it dominated one or more of the archive resident solutions and if neither the new solution of the archive resident solutions dominated each other. However, a newly obtained solution was not granted entry into the archive if it was dominated by at least one of the resident solutions. When a solution was added to the archive, a resident solution was deleted to maintain the pre-defined archive size. The algorithm employed to check for domination and delete an existing archive solution was presented in Appendix D.3. Within each iteration, non-dominated solutions and objective values were obtained using MATLAB codes that modelled equations 3.31 to 3.42 and these were updated until the maximum m number of iteration was attained as presented in Appendix D.3. The optimal solutions and their corresponding objective values were obtained and represented with Pareto fronts.

The flowchart for the MOHHO is shown in Fig 3.6

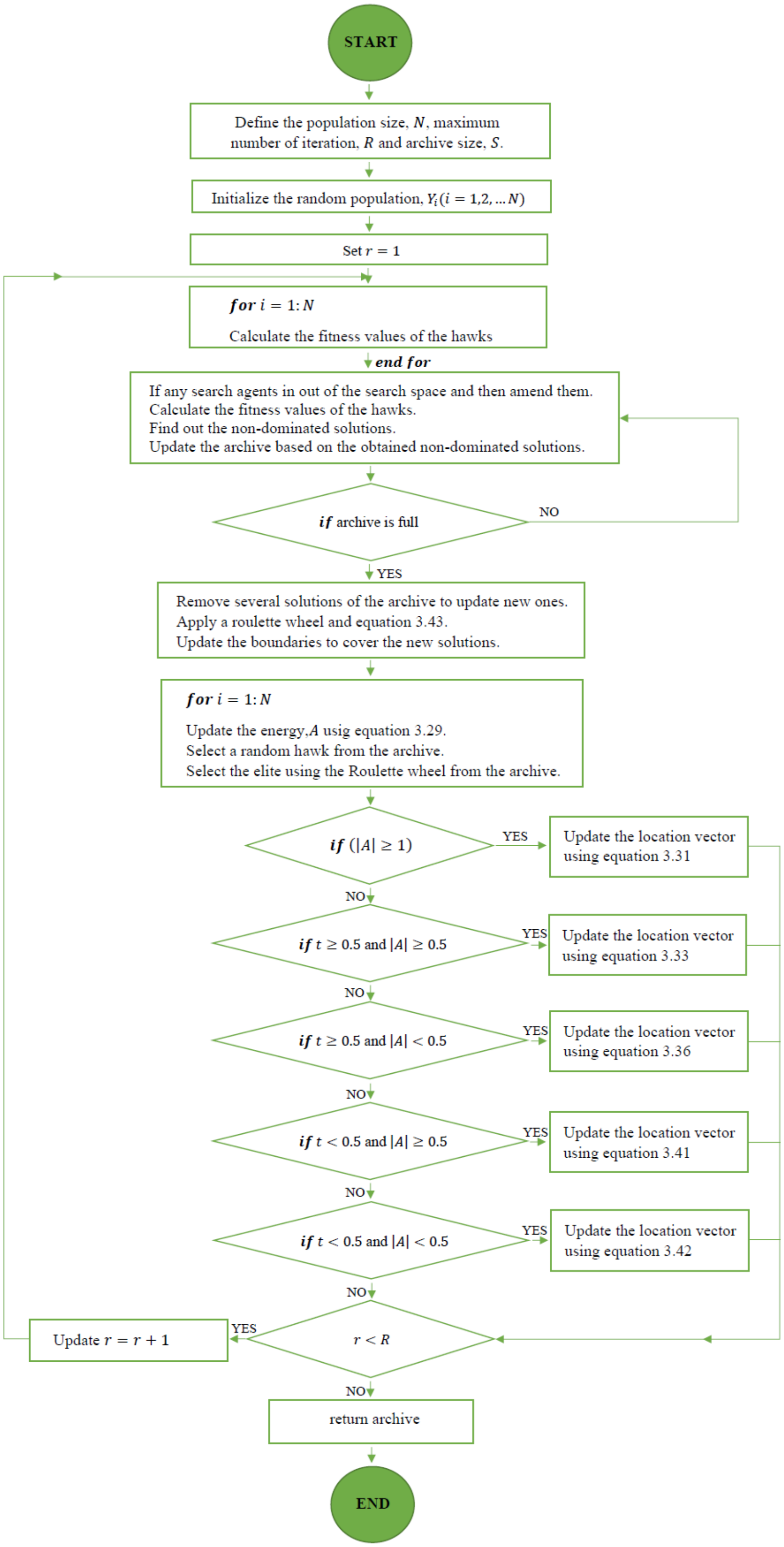

START
Define the population size, N, maximum number of iteration, R and archive size, S.
Initialize the random population, $Y_i (i = 1,2, \dots N)$
Set $r = 1$
$\boldsymbol{for}\ i = 1{:}\,N$
Calculate the fitness values of the hawks
$\boldsymbol{end\ for}$
If any search agents in out of the search space and then amend them.
Calculate the fitness values of the hawks.
Find out the non-dominated solutions.
Update the archive based on the obtained non-dominated solutions.
$\boldsymbol{if}$ archive is full
NO
YES
Remove several solutions of the archive to update new ones.
Apply a roulette wheel and equation 3.43.
Update the boundaries to cover the new solutions.
$\boldsymbol{for}\ i = 1{:}\,N$
Update the energy,A usig equation 3.29.
Select a random hawk from the archive.
Select the elite using the Roulette wheel from the archive.
$\boldsymbol{if}\ (|A| \geq 1)$
YES
Update the location vector using equation 3.31
NO
$\boldsymbol{if}\ t \geq 0.5$ and $|A| \geq 0.5$
YES
Update the location vector using equation 3.33
NO
$\boldsymbol{if}\ t \geq 0.5$ and $|A| < 0.5$
YES
Update the location vector using equation 3.36
NO
$\boldsymbol{if}\ t < 0.5$ and $|A| \geq 0.5$
YES
Update the location vector using equation 3.41
NO
$\boldsymbol{if}\ t < 0.5$ and $|A| < 0.5$
YES
Update the location vector using equation 3.42
NO
Update $r = r + 1$
YES
$r < R$
NO
return archive
END

Fig 3.6: Flowchart for the multi-objective Harris hawks optimization.

3.7 Ant lion optimization

Ant lions are net-winged insects whose lifestyle consists of larvae and adults. They exist for about three years and spend most of their lives in the larvae stage during which the hunting activities take place. The remaining 3-5 weeks of their life span are for adulthood mainly for reproduction activities.

The Ant lion optimization (ALO) algorithm is inspired and modeled according to the foraging characteristics of the ant-lion's larvae. Its hunting behavior is such that cone-shaped holes are excavated in the sand by the ant-lion's larvae using its powerful jaws (Scharf, Subach & Ovadia 2008). The cone-shaped holes, formed by expelling sand as the larvae move in a circular path, are of varying sizes. According to Goodenough, McGuire & Jakob, (2009), the sizes of these holes depend on the extent of hunger and the moon's shape. Hence, bigger holes are dug out when the ant lions are hungrier and/or when there is an occurrence of a full moon. The ant lion, after the digging process, waits for the emergence of the prey which are usually insects. The edge of the cone is made razor-sharp so that it is easy for the prey to fall to the base of the pit. The ant lion captures the insect once it enters the trap and reasonably tosses more sand on the prey if it tries to escape from the pit. The captured prey is dragged into the soil and eaten up with its remnant thrown out of the pit. Subsequently, the pit is modified in preparation for the upcoming hunting process.

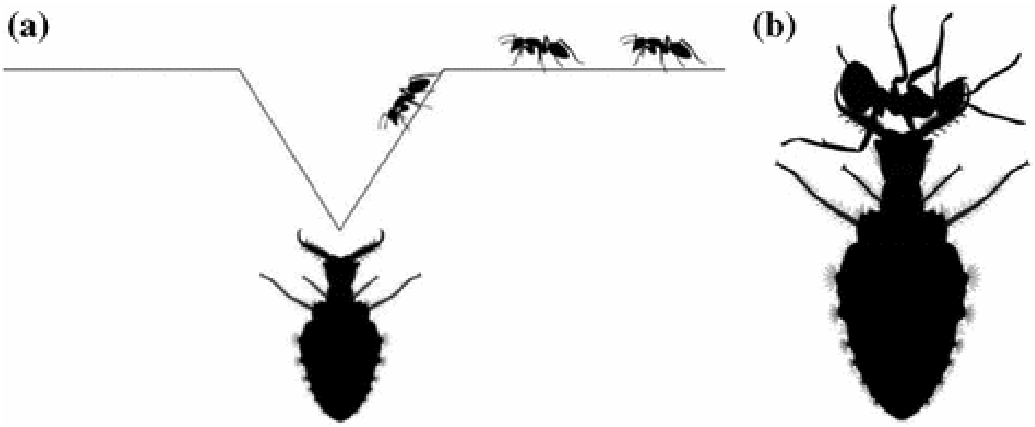


Fig 3.7: Cone-shaped traps and ant lion's hunting behavior (adapted from Mani, Bozorg-Haddad & Chu 2018)

The ALO algorithm emulates the relation between the ant lion and the insect prey in the cone-shaped trap. In nature, the insects move randomly over a certain area in search of food while the ant lion hunts for the insect with the aid of its trap. The random movement of the prey can be modeled as follows:

$$Y(t) = [0, cumsum(2t(r_1) - 1), cumsum(2t(r_2) - 1), \ldots cumsum(2t(r_n) - 1) \qquad (3.44)$$

Where,

$cumsum$= cumulative sum,

n= maximum number of epochs,

r= an iteration of random walk by the prey,

$t(r)$ is a randomly determined function defined by:

$$t(r) = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & if\ rand \leq 0.5 \end{cases} \qquad (3.44)$$

Where,

r= an iteration of random walk,

$rand$= a random value in [0, 1]

The location of the prey is committed to memory and employed for optimization using the following matrix format:

$$M_{Ant} = \begin{array}{ccccc} A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,d} \\ A_{2,1} & A_{2,2} & \cdots & \cdots & A_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & \cdots & A_{n,d} \end{array} \qquad (3.46)$$

Where,

M_{Ant}= matrix for saving the location of each ant/prey,

$A_{i,j}$= value of the j^{th} variable of the i^{th} ant,

n= number of ants,

d= number of variables.

An objective function is employed for the evaluation of each ant during the optimization process and the following matrix is used to store the fitness value of all prey.

$$M_{OA} = \begin{array}{ccccc} F([A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,d}]) \\ F([A_{2,1} & A_{2,2} & \cdots & \cdots & A_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F([A_{n,1} & A_{n,2} & \cdots & \cdots & A_{n,d}]) \end{array} \qquad (3.47)$$

Where,

M_{OA}= matrix for storing the fitness value of each ant/prey

$A_{i,j}$= value of the j^{th} dimension of the i^{th} ant,

n= number of ants.

Also, the location of the ant lions is saved, hence their position and fitness matrices are given as follows:

$$M_{Antlion} = \begin{matrix} AL_{1,1} & AL_{1,2} & \cdots & \cdots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \cdots & \cdots & AL_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ AL_{n,1} & AL_{n,2} & \cdots & \cdots & AL_{n,d} \end{matrix} \tag{3.48}$$

$$M_{OAL} = \begin{matrix} F([AL_{1,1} & AL_{1,2} & \cdots & \cdots & AL_{1,d}]) \\ F([AL_{2,1} & AL_{2,2} & \cdots & \cdots & AL_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F([AL_{n,1} & AL_{n,2} & \cdots & \cdots & AL_{n,d}]) \end{matrix} \tag{3.49}$$

Where,

$M_{Antlion}$= matrix for saving the location of each ant lion,

M_{OAL}= matrix for storing the fitness value of each ant lion,

$AL_{i,j}$= value of the j^{th} dimension of the i^{th} ant lion,

n= number of ant lions,

F= objective functions

The following conditions are employed during the optimization process:

1. The ants navigate the search area using various random movements.
2. All dimensions of the ants are subject to the application of random walks.
3. The traps set by the ant lions have effects on the random walks of the ants.
4. The pits built by the ant lions is directly proportional to their fitness.
5. Ant lions that possess bigger pits are more likely to capture ants.
6. An ant lion in each iteration and the fittest ant lion are capable of capturing any ant.
7. The sliding of the ants towards the ant lion is simulated by a decrease in the range of the random walk.
8. An ant is dragged into the sand by an ant lion once it becomes fitter than the ant lion.
9. After each hunt, the ant lion moves itself closer to the most recent prey it has caught and digs a pit to increase its chances of capturing more prey.

3.7.1 Random walks of ants

The random walks of the ants are based on equation 3.44. During each step of the optimization process, the position of the ants is updated. In order to account for the boundary of the search area, equation 3.44 is normalized to produce the following equation.

$$Y_i^t = \frac{(Y_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \qquad (3.50)$$

Where,

a_i= minimum random walk of the i^{th} variable,

d_i= maximum random walk in the i^{th} variable,

c_i^t= minimum i^{th} variable at the t^{th} iterations,

d_i^t= maximum i^{th} variable at the t^{th} iterations,

Therefore, to ensure the presence of random walks in the search area, equation 3.46 must be used in each iteration.

3.7.2 Trapping in ant lion's pit

As stated in the above conditions for the optimization process, the traps of the ant lions affect the random walks of the ants and this assumption can be mathematically modeled in equation 3.51 and 3.52 as follows:

$$c_i^t = Antlion_j^t + c^t \qquad (3.51)$$

$$d_i^t = Antlion_j^t + d^t \qquad (3.52)$$

Where,

c^t= minimum of all variables at t^{th} iteration,

d^t = maximum of all variables at t^{th} iteration,

c_i^t= minimum of all variables at i^{th} ant,

d_i^t= maximum of all variables at i^{th} ant,

$Antlion_j^t$= position of the selected j^{th} ant lion at the i^{th} iteration.

Figure 3.8 illustrates a two-dimensional hypersphere around the chosen ant ion within which the ants are allowed to move.

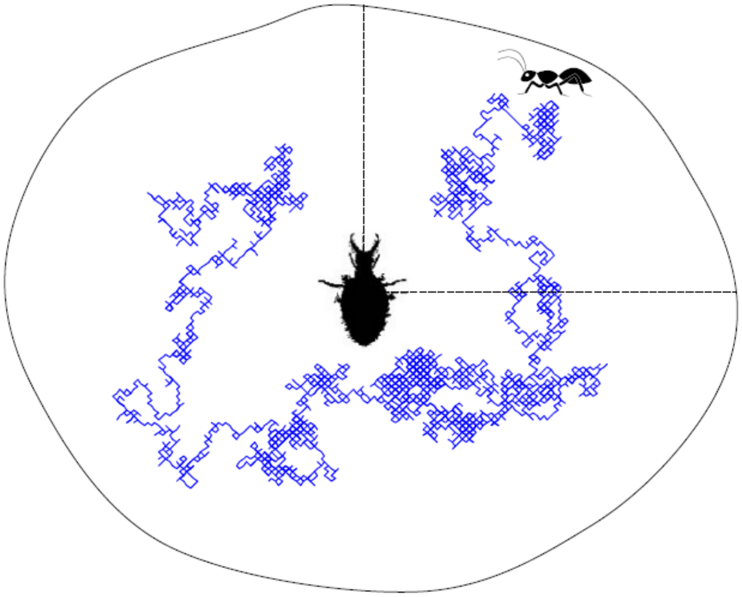


Fig 3.8: An ant randomly walking through an ant lion's trap (adapted from Heidari et al. 2019)

3.7.3 Building the Trap

The ability of the ant lion to hunt is modeled using a roulette wheel. During optimization, the roulette wheel operator is utilized in the ALO algorithm to select an ant lion according to its fitness. Hence, the fitter ant lion has a greater probability of catching the ants courtesy of this roulette technique.

3.7.4 Sliding ants towards the ant lion

Using the same roulette technique, the ant lions can construct traps in proportion to their fitness, and the ants must move arbitrarily. However, the ant lion throws sand towards the trapped prey as it tries to escape hence facilitating its sliding down towards the ant lion. To mathematically model this behavior, the radius of the ant's random walks hypersphere is reduced. Following this, equations 3.53 and 3.54 are given as follows.

$$c^t = \frac{c^t}{I} \qquad (3.53)$$

$$d^t = \frac{d^t}{I} \qquad (3.54)$$

Where,

I= ratio,

c^t= minimum of all variables at t^{th} iteration,

d^t = maximum of all variables at t^{th} iteration,

$$I = 10^w \frac{t}{T} \qquad (3.55)$$

Where,

t= current iteration,

T= maximum number of iterations,

w= constant defined based on the current iteration ($w = 1\ when\ t > 0.1T; w = 3\ when\ t > 0.5T; w = 4\ when\ t > 0.75T; w = 5\ when\ t > 0.9T\ and\ w = 6\ when\ t > 0.95T$)

3.7.5 Catching the prey and rebuilding the pit.

This is the last stage of the ant lion's hunt whereby the ant enters the bottom of the pit, is captured by the ant lion's jaw into the sand before being eaten up. As stated in the conditions for the optimization process, the prey is caught when it becomes fitter than the ant lion. Hence, the ant lion updates its position to the current position of the captured ant in order to improve its probability of capturing a new prey. Following from this, equation 3.56 is given as follows:

$$Antlion_j^t = Ant_i^t \quad if\ F(Ant_i^t) > F\left(Antlion_j^t\right) \qquad (3.56)$$

Where,

t= current iteration,

$Antlion_j^t$= position of the selected j^{th} ant lion at the t^{th} iteration,

Ant_i^t = position of the i^{th} ant at the t^{th} iteration,

3.7.6 Elitism

This is a fundamental feature in evolutionary algorithms where the best solution(s) found at a step of the optimization process is kept. Hence, for ALO, the best ant lion obtained so far is committed to memory and referred to as an elite. Since the elite is the fittest ant lion, it ought to have an impact on how the ants move during the iterations. Therefore, it is supposed that each ant arbitrarily moves near an ant lion chosen by the roulette wheel and elite concurrently sown in equation 3.57 as follows:

$$Ant_i^t = \frac{P_A^t + P_E^t}{2} \qquad (3.57)$$

Where,

P_A^t= random walk around the ant lion chosen by the roulette wheel at the t^{th} iteration,

P_E^t= random walk around the elite at the t^{th} iteration,

Ant_i^t= position of the i^{th} ant at the t^{th} iteration.

3.7.7 Single-objective optimization

Based on the above-mentioned equations, a mathematical model was developed, and it comprised the following components.

1. Problem definition: The ‘Get_Functions_details.m’ file was created as shown in Appendix E.1 to define the upper and lower limits of the decision variable as well as the objective functions to be optimized.
2. Initialization: The initial population of solutions were initialized to a size of 100. An initialization function was created to ensure that the randomly initialized values did not exceed the defined decision variable boundaries. These algorithm are presented in Appendix E.1.
3. Evaluation: The random walk of the ant around the antlion as depicted in equation 3.44 to 3.50 is modelled in the ‘random_walk_around_antlion.m’ file presented in Appendix E.2. This encouraged diversity and hence exploration of the algorithm. The position of the ants during the random walk were updated and their fitness were calculated as shown in ‘ALO.m’ file (Appendix E.2). The position and fitness of the antlion were also computed. For this algorithm, the concept of elitism was employed and hence codes were written to obtain the ant and antlion positions per iteration and the elite ant and antlion was retained until the maximum number of iterations was attained. The ‘ALO.m’ then returns the final position of the fitness of the elite antlion (‘elite_antlion_position’ and

‘elite_antlion_fitness’) as the optimal decision variable and objective value respectively. The associated codes were presented in Appendix E.2.

The flowchart and pseudo code for the ALO is shown in Fig 3.9

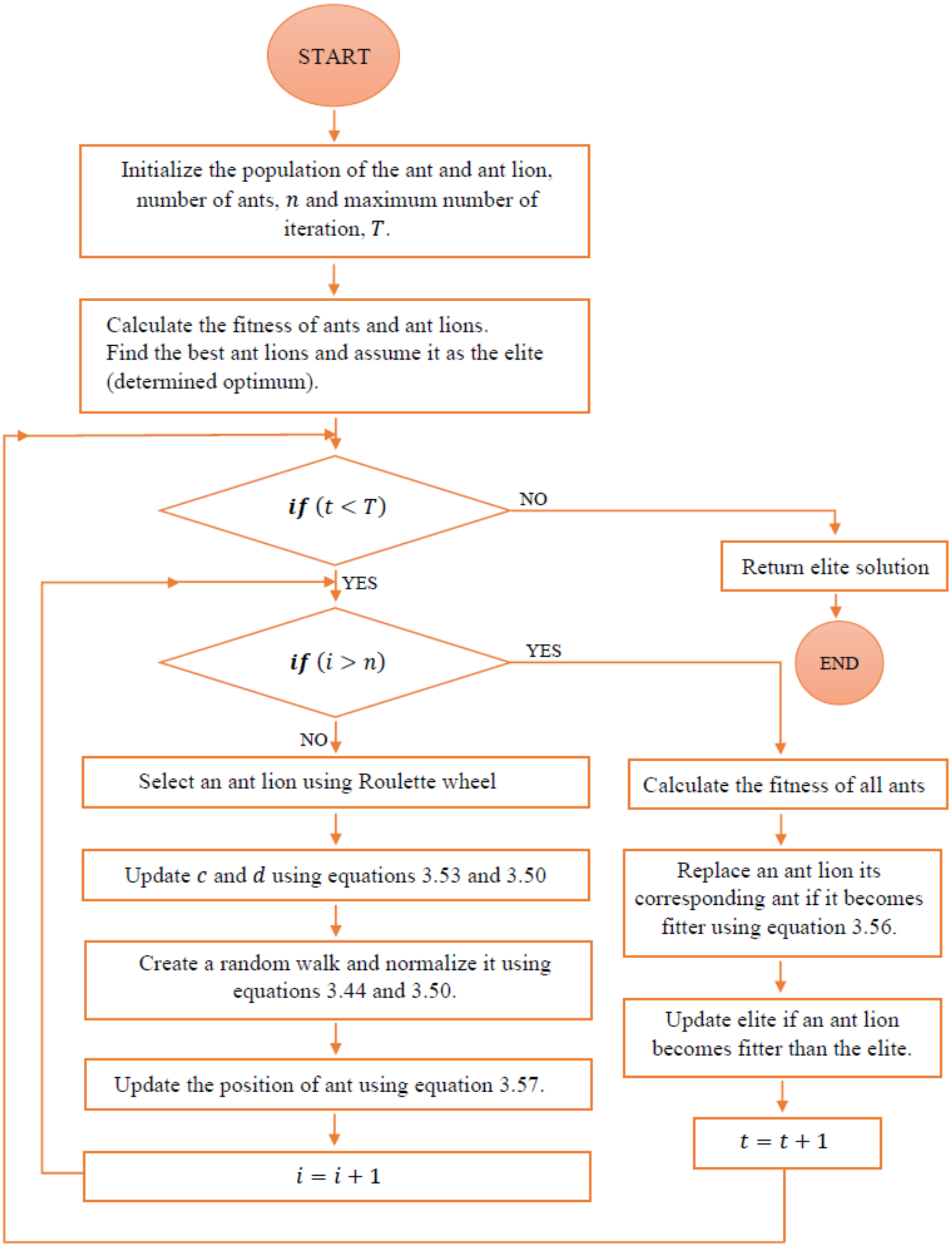

START
Initialize the population of the ant and ant lion, number of ants, n and maximum number of iteration, T.
Calculate the fitness of ants and ant lions. Find the best ant lions and assume it as the elite (determined optimum).
$\boldsymbol{if}\ (t < T)$
NO
Return elite solution
END
YES
$\boldsymbol{if}\ (i > n)$
YES
NO
Select an ant lion using Roulette wheel
Update c and d using equations 3.53 and 3.50
Create a random walk and normalize it using equations 3.44 and 3.50.
Update the position of ant using equation 3.57.
$i = i + 1$
Calculate the fitness of all ants
Replace an ant lion its corresponding ant if it becomes fitter using equation 3.56.
Update elite if an ant lion becomes fitter than the elite.
$t = t + 1$

Fig 3.9: Flowchart for the single-objective Antlion optimization

3.7.8 Multi-objective optimization

The multi-objective ant lion optimization (MOALO), inspired by the MOPSO, incorporates the leader selection approach and archive maintenance. The size of the archive is pre-determined, and solutions are selected from the archive so as to enhance the distribution. Niching is employed to, ensure the distribution of the solution in the archive such that a pre-determined radius is used to study the area around each solution. The number of nearby solutions is then tallied and taken into account as the distributional measure. In order to increase the distribution of the solution in the archive, two strategies are considered similar to those in MOPSO. First, the ant lion is chosen from the solutions having the least crowded neighborhood. The chances of electing a solution from the archive are determined using equation 3.58 (Mirjalili, Jangir & Saremi 2017).

$$P_i = \frac{c}{N_i} \qquad (3.58)$$

Where,

c= constant value greater than 1,

N_i= number of solutions in the neighborhood of the i^{th} solution.

For the second strategy, when the archive is filled up, the solutions having the most crowded neighborhood are deleted to make room for new solutions. To determine the likelihood of deleting a solution from the archive, the following equation is used.

$$P_i = \frac{N_i}{c} \qquad (3.59)$$

Finally, it is necessary that equation 3.52 should be modified in order to solve the multi-objective problem. Also, equation 3.53 is modified for the selection of the arbitrary ant lions and elite such that it is employed to choose a non-dominated solution from the archive.

The developed mathematical model for multi-objective ALO comprises the following components:

1. Problem definition: All three objective functions were defined in the 'ZDT1.m' file as. Also, the upper and lower boundaries of the decision variables were defined in the 'MALO.m' file. The content of these files was presented in Appendix F.1.

2. Initialization: The initial population size and number of iterations were initialized. Additionally, positions and fitness of the ant and antlion were initialized and an archive with a size of 100 was also initialized to store the non-dominated solutions. The initialization function was created to ensure that the initialized values for ant and antlion positions did not exceed the decision variables constraints. The associated codes were shown in Appendix F.2.
3. Iteration improvement: The elitism approach was employed in this algorithm as shown in the 'MALO.m' to ensure that the best ant lion position and fitness value per iteration was returned. The elite positions were determined based on the roulette wheel selection modelled in 'RouletteWheelSelection.m' as shown in Appendix F.2.
4. Evaluation: Within each iteration, the fitness value for the antlion were calculated based on the defined objective function. The archive was updated with non-dominated solutions using the 'UpdateArchive' function (Appendix F.3). This function ensured that the number of non-dominated solutions did not exceed the predefined archive size. From the stored non-dominated solutions, an elite ant lion was selected based on its rank using the roulette wheel selection. The random walk function (depicted in equations 3.44 to 3.50) was used to model the random walk of the ant around the ant lion. The ant's position was then updated by taking the average of the random walk. This was carried out until the maximum number of iterations was attained as shown in Appendix F.3. Finally, the obtained ant and ant lion position became the optimal decision variable and objective values respectively and they were employed to present the pareto fronts.

The flowchart for the ALO is shown in Fig 3.10

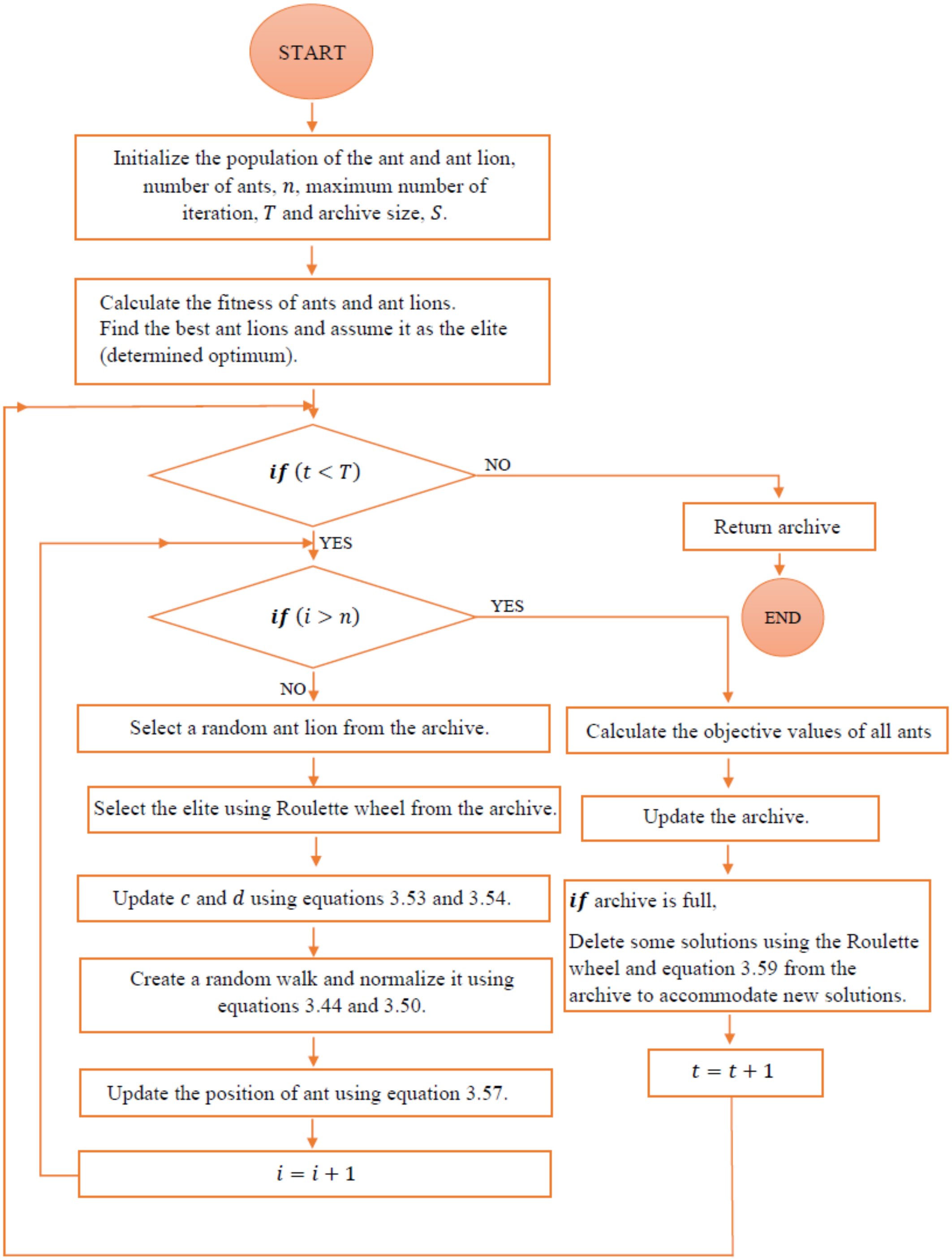

START
Initialize the population of the ant and ant lion, number of ants, n, maximum number of iteration, T and archive size, S.
Calculate the fitness of ants and ant lions.
Find the best ant lions and assume it as the elite (determined optimum).
$\boldsymbol{if}\ (t < T)$
NO
Return archive
END
YES
$\boldsymbol{if}\ (i > n)$
YES
NO
Select a random ant lion from the archive.
Select the elite using Roulette wheel from the archive.
Update c and d using equations 3.53 and 3.54.
Create a random walk and normalize it using equations 3.44 and 3.50.
Update the position of ant using equation 3.57.
$i = i + 1$
Calculate the objective values of all ants
Update the archive.
$\boldsymbol{if}$ archive is full,
Delete some solutions using the Roulette wheel and equation 3.59 from the archive to accommodate new solutions.
$t = t + 1$

Fig 3.10: Flowchart for the multi-objective Antlion optimization

3.8 Chapter summary

This chapter extensively illustrated the methodology employed in this research work. It started by describing the CCHP system to be optimized as well as an introduction into the RSM method used for the derivation of the objective function. The concepts of the GWO, HHO and ALO techniques were elucidated and a detailed explanation of how each mathematical programming model for the single and multi-objective optimization was discussed.

Chapter Four – Results

This chapter presents and analyses the solutions to the solar-assisted CCHP optimization problem obtained using the GWO, HHO, and ALO. Other analysis such as the sensitivity and convergence analysis will also be reported.

4.1 Greywolf optimization

4.1.1 Single-Objective Optimization

The single-objective grey wolf optimization was carried out on each of the objective functions with several search agents and iterations equally set to a value of 100. It was then executed on the MATLAB platform using an 8GB RAM Intel(R) Core (TM) i3-5005U CPU @ 2.00GHz laptop from the mathematical programming model developed in this study as shown in Appendix A..

4.1.1.1 Net Power Optimization

The solution of Equation (3.13) involved using the decision variables and their corresponding values outlined in Tables 3.1 and 3.2 on the GWO algorithm. The outcomes, as presented in Table 4.1, illustrate the necessary actions to maximize net power. The following results indicate the steps required for optimizing net power:

- Minimize the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature.
- Maximize the inlet turbine temperature.

Table 4.1. Optimal solutions maximizing the net power (GWO).

Cr	Pp	Gt	Ct	Maximum Net Power
10	10	1520	850	61.8462

4.1.1.2 CO_2 Emission Optimization

The GWO algorithm was used to solve Equation (3.14) by utilizing the decision variables and their corresponding values listed in Tables 3.1 and 3.2. The results of this optimization process are shown in Table 4.2, which presents the required actions to minimize CO_2 emissions. The following outcomes indicate the necessary steps that need to be taken to optimize and reduce the levels of CO_2 emissions:

- Minimize the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature.
- Maximize the inlet turbine temperature.

Table 4.2. Optimal solutions maximizing the CO_2 emission (GWO).

Cr	Pp	Gt	Ct	Minimum CO_2 Emission
10	10	1520	850	50.4771

4.1.1.3 Exergy Efficiency Optimization

The efficiency of exergy was calculated using Equation (3.15), with the GWO algorithm utilizing the decision variables and values presented in Tables 3.1 and 3.2. The following outcomes indicate the necessary steps that need to be taken to maximize the exergy efficiency:

- Minimize the compression ratio, pinch point temperature difference, and inlet turbine temperature.
- Maximize the inlet combustion chamber temperature.

Table 4.3. Optimal solutions maximizing the exergy efficiency (GWO).

Cr	Pp	Gt	Ct	Maximum Exergy Efficiency
10	10	1420	950	42.3507

4.1.1.4 Analysis of the Single-Objective Optimization Results

The results obtained in Tables 4.1 to 4.3 have been summarized in Table 4.4, which shows the behavior of the optimal solutions. As shown in table 4.4, an upward trend is represented by ↑, a downward trend is represented by ↓ while a conflict between the parameters is indicated by ≠. It could be seen that for decision variables Cr and Pp, all three objective functions are in harmony. However, for the Gt and Ct decision variables, the exergy efficiency conflicts with the net power and CO_2 emission. Thus, simultaneous optimization of the three objectives is required to determine the Pareto optimal solutions.

Table 4.4. Illustration of the trend of the objective functions (GWO).

Decision Variables	Net Power	CO_2 Emission	Exergy Efficiency
Cr	↓	↓	↓
Pp	↓	↓	↓
Gt	↑	↑	↓≠
Ct	↓	↓	↑≠

4.1.2 Multi-Objective Optimization

The objective functions were optimized using the multi-objective grey wolf optimization from the developed mathematical model after adopting 100 iterations. The codes were implemented on MATLAB using an Intel(R) Core (TM) i3-5005U CPU @ 2.00GHz laptop with 8GB RAM. Table 4.5 provides the hyperparameters' set values and the associated codes are shown in Appendix B.

Table 4.5. Set values of the hyperparameters (MOGWO).

Hyperparameters	Value
Archive size	100
Number of variables	4
Greywolf number	100
Grid inflation parameter (alpha)	0.1
Number of grids per dimension (nGrid)	4
Leader selection pressure parameter (beta)	4
Gamma	2

The optimization algorithm produced one hundred sets of solutions that were not dominated by any other solutions. The sets of optimal solutions obtained provide the best trade-off solutions despite the conflicting nature of the objective functions. All the Pareto-front options are potentially optimum solutions as no one solution is better than the other; the decision-maker can choose the best optimal answer at their discretion.

Three scenarios describing the relationship between CO_2 emission and net power output, exergy efficiency and net power output, and exergy efficiency and CO_2 emission were taken into consideration to get further insight into the relationship between each objective function. The Pareto front of the three objective functions is represented by the global results, which are displayed in Figure 4.1. To highlight certain key points, this will be reduced to three two-dimensional Pareto fronts.

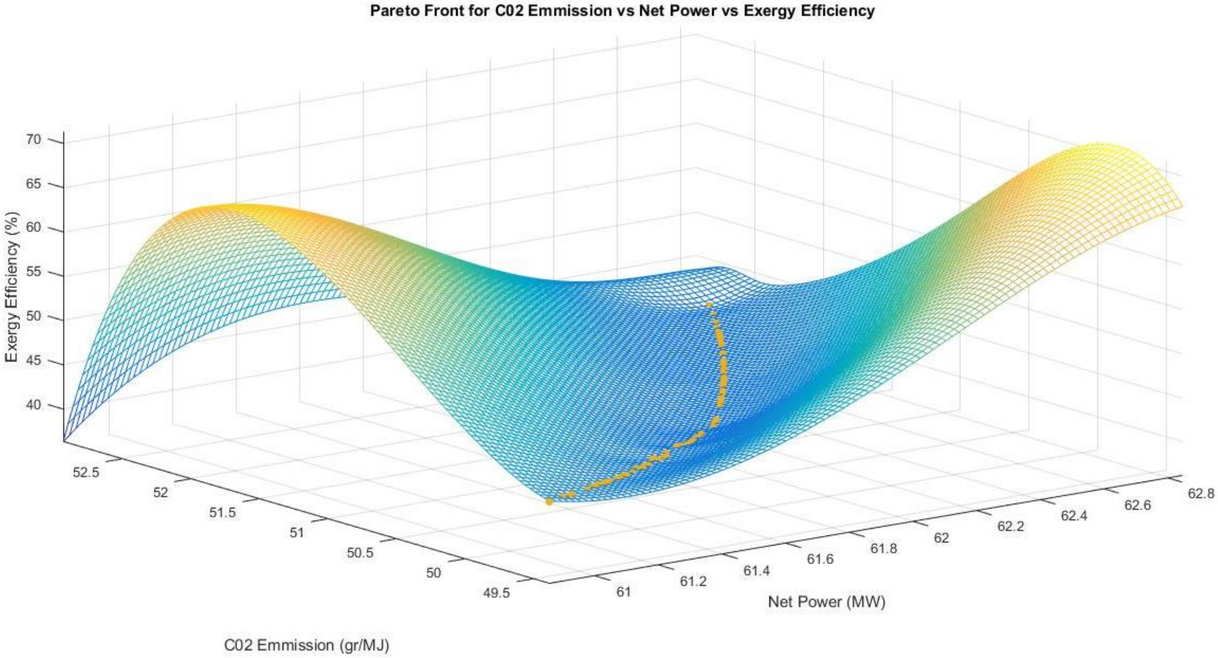


Fig 4.1: Pareto front of CO_2 emission vs net power vs exergy efficiency (MOGWO).

Figure 4.2's Pareto front effectively illustrates a conflict in the decision-making process since maximizing net power will inevitably lead to higher CO_2 emissions, which will have a detrimental impact on the environment. In addition to the additional power generated by the Kalina cycle, the power output from the GT-3 has a substantial impact on the net power of the CCHP system under investigation. Considering this, the system could be designed so that the temperature of the combustion gases from APH-2 entering the unfired HRSG is increased more than those entering the combustion chamber. In an effort to reduce CO_2 emissions, the photovoltaic thermal collectors' heating power may also be increased, and the fuel consumption rate may be lowered. As a result, the power production from the Kalina cycle—which produces electricity in smaller amounts but is "cleaner"—will increase while the GT-3, which produces the larger net power as a "top" system, will create less net power. Even while this will eventually result in the system producing less net electricity, it will optimize the system and minimize CO_2 emissions.

As a lower amount of CO_2 emission corresponds with a better exergy efficiency, the Pareto graph in Figure 4.3 implies that the exergy efficiency and the CO_2 emission are compatible. Additionally, it shows that a considerable gain in exergy efficiency requires minimizing CO_2 emissions up to a certain point. Subsequent investigation shows that reducing fuel usage would also reduce fuel input exergy. With reduced input exergy and enhanced exergy outputs/useful work from the Kalina cycle, the water heat exchanger and absorption chiller may greatly boost exergy efficiency, which inevitably results in low CO_2 emissions.

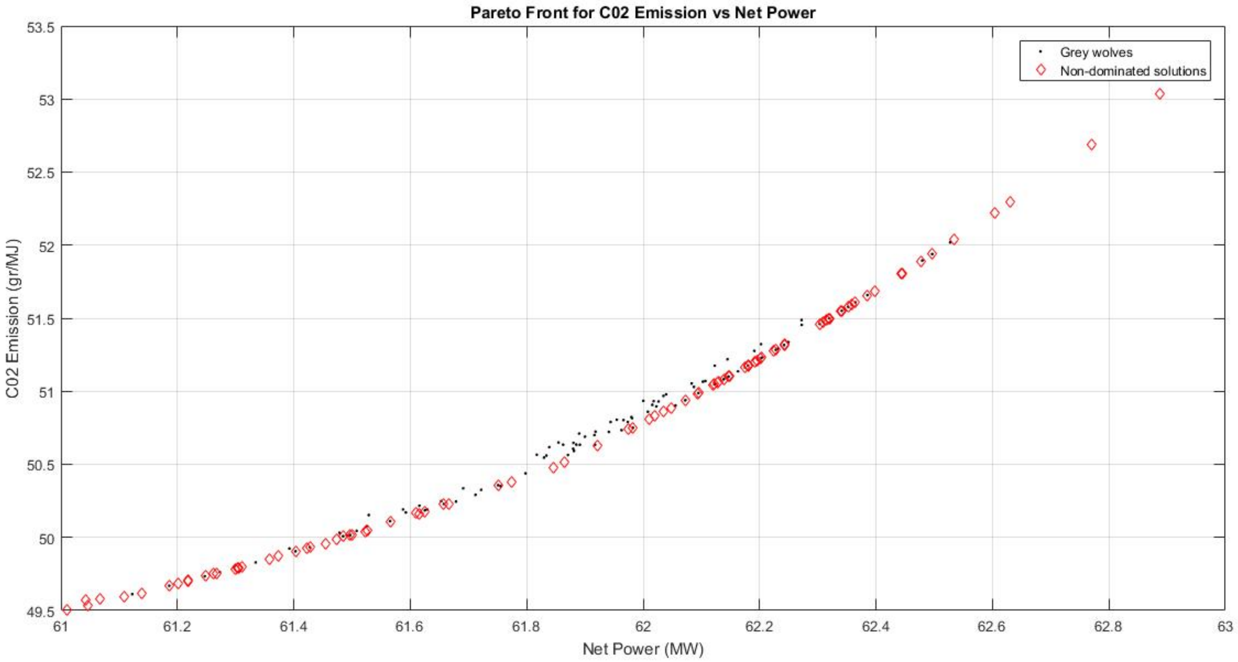


Fig 4.2: Pareto front of CO_2 emission vs net power (MOGWO).

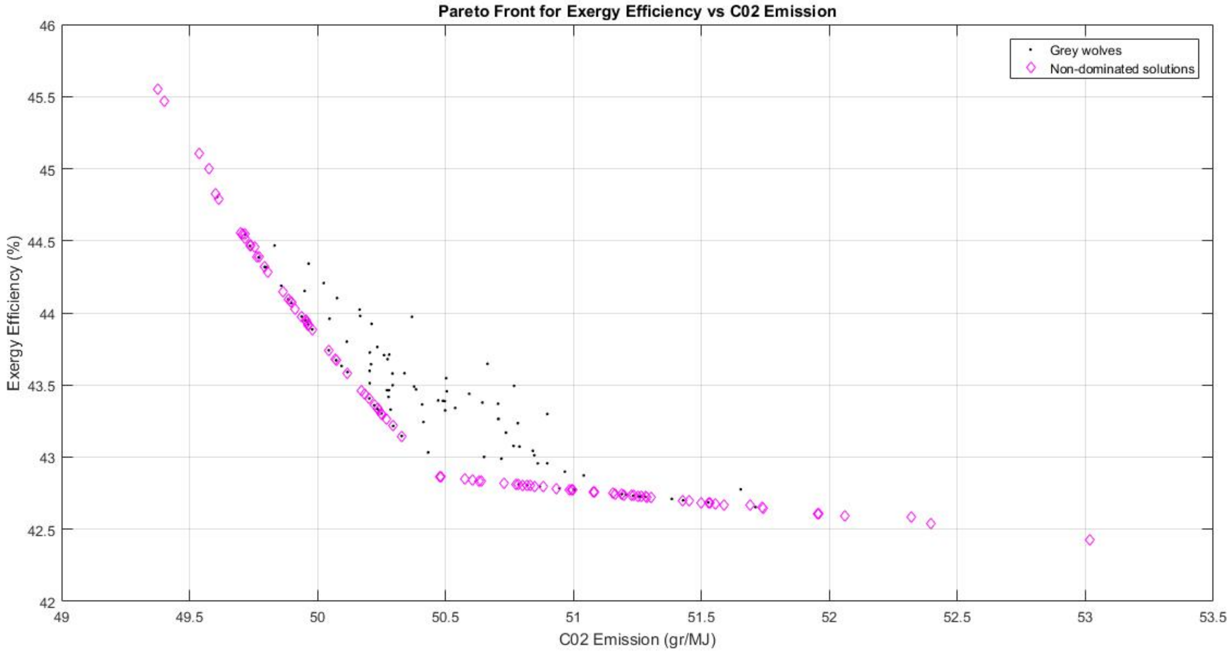


Fig 4.3: Pareto front of exergy efficiency vs CO2 emission (MOGWO).

Additionally, the relationship between the exergy efficiency and net power is inverse, as illustrated in Figure 4.4. This implies that a system may not always be highly efficient just because it produces a high net power output. Nonetheless, a system's maximum energy efficiency is determined by how well the power produced within the CCHP system is utilized. This implies that bottom systems (such as the absorption chiller and Kalina cycle) should be built and operated to maximize the use of waste heat and minimize energy losses.

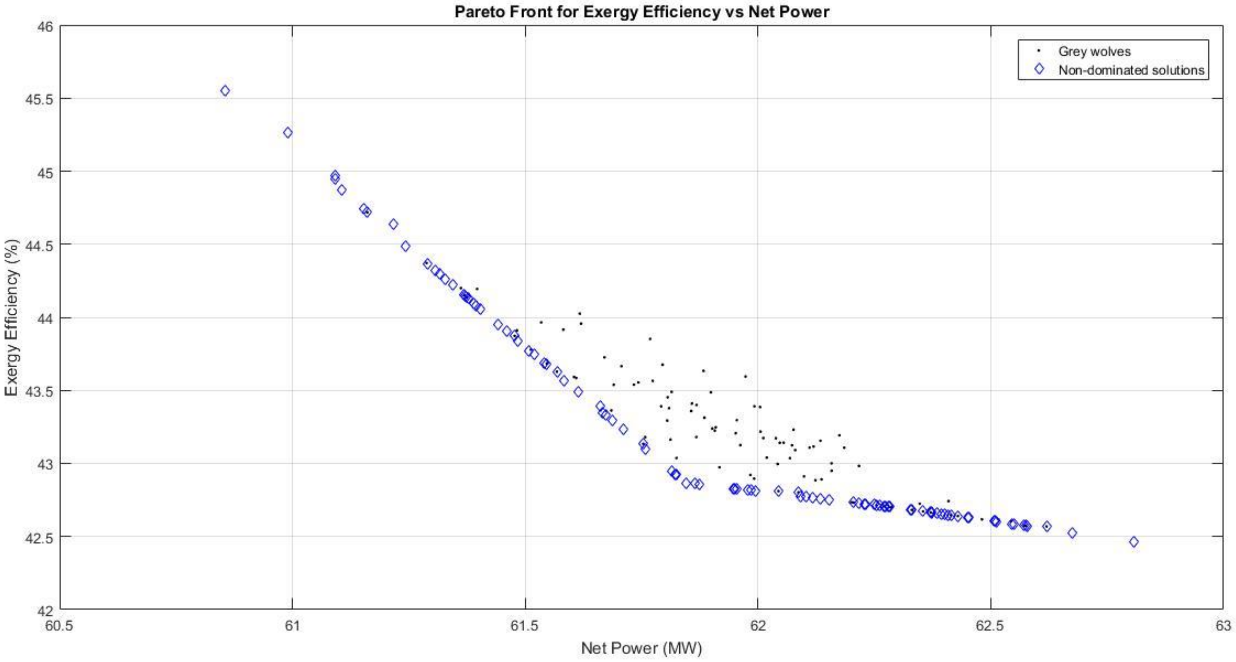


Fig 4.4: Pareto front of exergy efficiency vs net power (MOGWO).

The MOGWO was designed to produce six optimal solutions, then compared to those from a related study for validation purposes. Table 4.6 displays the obtained data, which indicates that the MOGWO generated an optimized system with reduced CO_2 emissions and increased energy efficiency values although this came at the expense of a lower net power.

Table 4.6: Comparison of optimization results with a related study.

	Optimal Decision Variables				Optimal Objective Functions		
	Cr	**Pp**	**Gt**	**Ct**	**Net Power**	**CO_2 Emission**	**Exergy Efficiency**
Present study	10.00	10.86	1520	913.11	61.60	50.57	45.21
	10.00	10.56	1520	901.58	61.47	50.31	45.28
	10.00	10.46	1520	895.34	61.40	50.18	45.31
	10.01	10.46	1520	893.99	61.38	50.16	45.32
	10.00	10.68	1520	908.14	61.55	50.45	45.24
	10.00	10.68	1520	907.69	61.54	50.44	45.24

A similar study (Mahdavi, Mojaver & Khalilarya, 2022)	11.66	11.96	1470	900	61.73	52.87	44.22
	11.11	20.00	1470	900	61.73	52.99	44.09
	11.98	20.00	1470	890	61.75	53.84	44.12
	12.50	16.10	1484	900	61.75	54.07	44.58
	12.50	15.72	1470	891	61.79	54.07	44.10
	12.50	20.00	1468	882	61.75	54.28	44.30

4.1.3 Sensitivity analysis

The sensitivity analysis was carried out to better understand the dynamics around the multi-objective optimization outcomes. It also aids in determining the effect observed on the two-dimensional Pareto front of net power, exergy efficiency, and CO_2 emission by varying each decision variable. The parameters include the compression ratio, pinch point temperature difference, inlet turbine temperature, and inlet combustion chamber temperature. The results obtained from each analysis are in line with the study conducted by Mahdavi, Mojaver & Khalilarya (2022), who used the RSM to model and optimize a solar-based CCHP system.

4.1.3.1. Analysis of the Compression Ratio

To conduct this analysis, the other decision variables (10–30 for Pp, 1420–1520 for Gt, and 850–950 for Ct) were allowed to fluctuate within their respective ranges, while the compression ratio was kept constant at values of 10, 12, and 15. As illustrated in Figure 4.5, the highest values of net power and energy efficiency are achieved at Cr = 15, albeit at the expense of elevated CO_2 emissions. Additionally, it was evident that lowering the value of Cr from 15 to 10 would result in a greater decrease in CO_2 emissions than in net power and energy efficiency. According to this, using a lower compression ratio will significantly reduce CO_2 emissions while having little or no impact on the optimal values of net power and energy efficiency. Interestingly, the Cr creates the most change/effect in the CO_2 emission and yields the lowest result when compared to other decision variables.

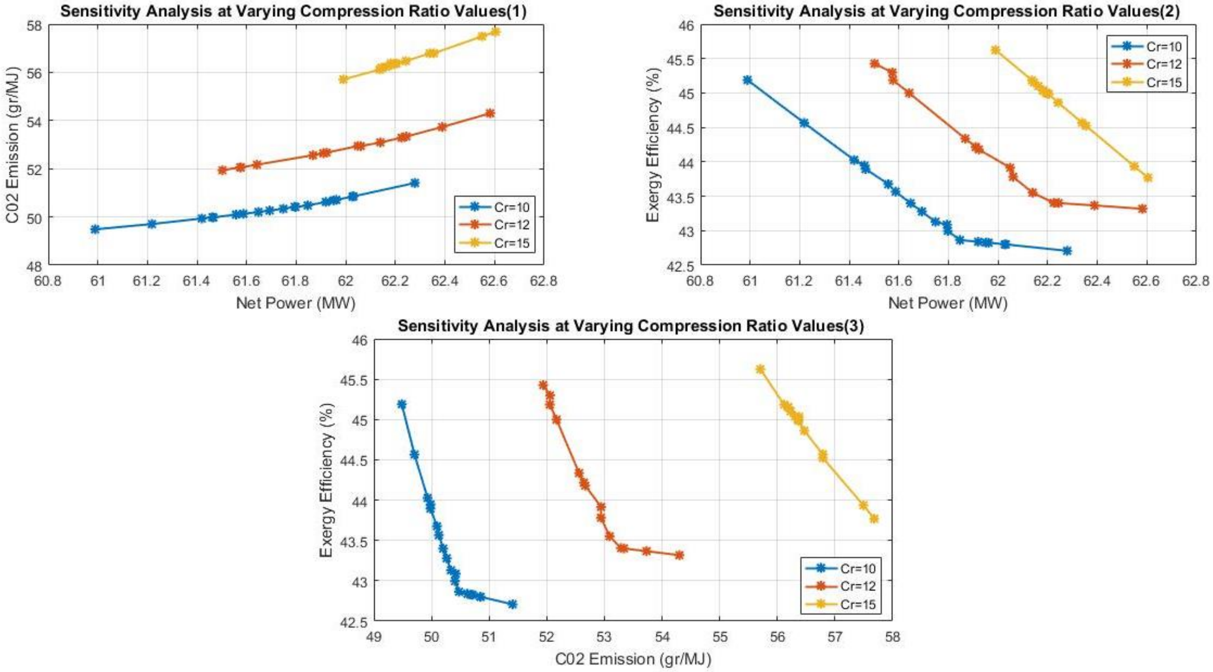


Figure 4.5. Effects of compression ratio on the optimal objective functions.

4.1.3.2 Analysis of the Pinch Point Temperature Difference

The air preheater 1 pinch point temperature was maintained at 10, 20, and 30 degrees Celsius, and the other variables were allowed to fluctuate within their respective ranges (10–15 degrees for Cr, 1420–1520 degrees for Gt, and 850–950 degrees for Ct). Figure 4.6 demonstrates that maximum values for CO_2 emissions, energy efficiency, and net power were attained at Pp = 30. Additionally, it shows that a decrease in Pp from 30 to 20 results in a higher variation in CO_2 emissions (in comparison to a decrease from 20 to 10) along with a corresponding decrease in net power and energy efficiency. This implies that Pp would only need to be around average to yield positive outcomes. The Pp affects the objective functions the least when compared to other decision variables.

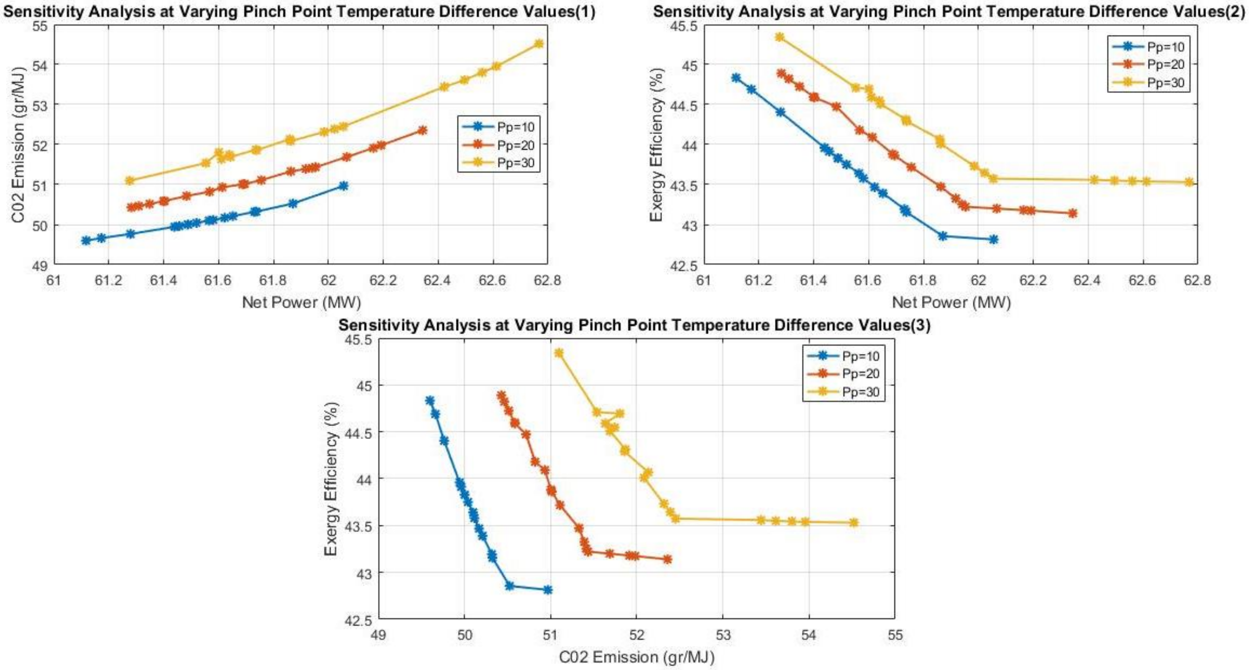


Figure 4.6: Effects of pinch point temperature difference on the optimal objective functions.

4.1.3.3 Analysis of the Inlet Turbine Temperature

The temperature at the inlet of the turbine was maintained at 1420, 1470, and 1520 while the other variables were allowed to fluctuate within their respective ranges (10–15 for Cr, 10–30 for Pp, and 850–950 for Ct). Figure 4.7 illustrates that higher exergy efficiency values, a minimal net power, and CO_2 emission are reached at higher values of Gt (Gt = 1520). This indicates that higher Gt values result in a system that is very efficient and emits very little CO_2 but at the cost of a lower net power. When compared to other decision variables, the Gt has the biggest impact on the efficiency of exergy.

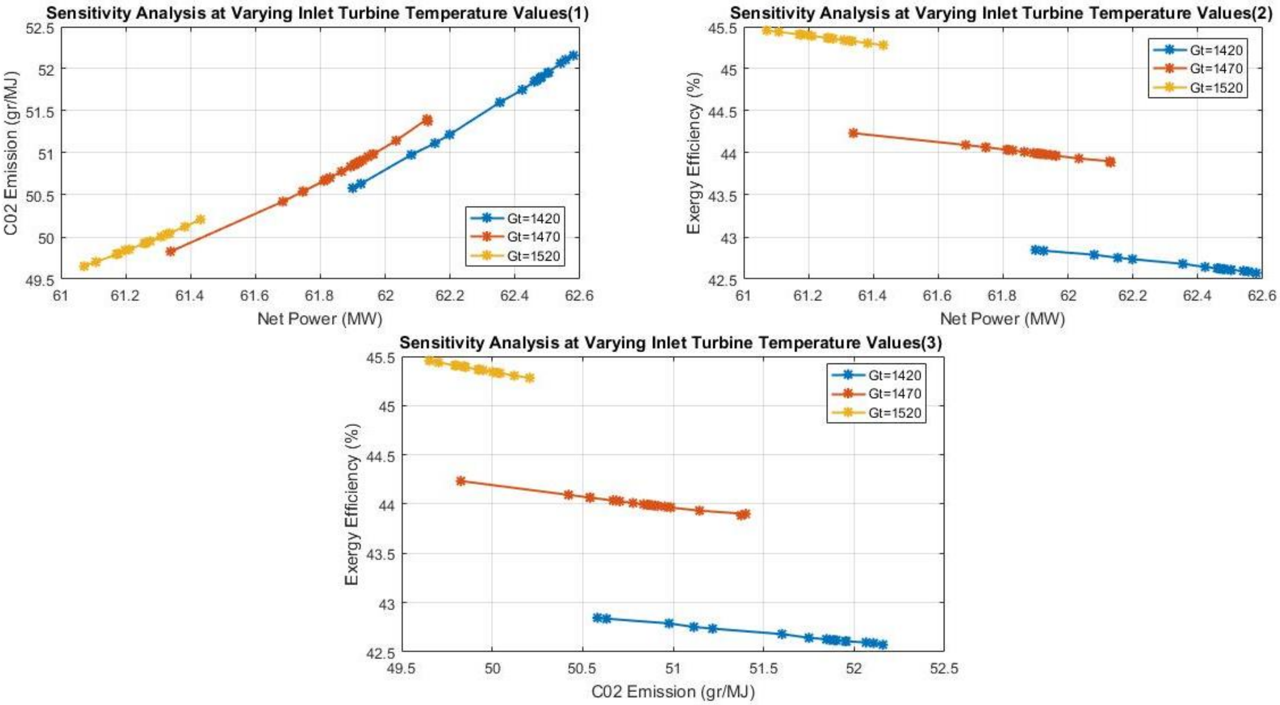


Figure 4.7: Effects of turbine inlet temperature on the optimal objective functions.

4.1.3.4 Analysis of the Inlet Combustion Chamber Temperature

The other variables were allowed to fluctuate within their respective ranges (10–15 for Cr, 10–30 for Pp, and 1420–1520 for Gt), but the combustion chamber inlet temperature was maintained at values of 850, 900, and 950. A Ct value of 950 produced the largest net power and CO_2 emission as well as the minimum exergy efficiency, as shown in Figure 4.8. This implies that in order to achieve lower CO_2 emission values, maximum energy efficiency, and minimum net power, lower values of Ct are needed.

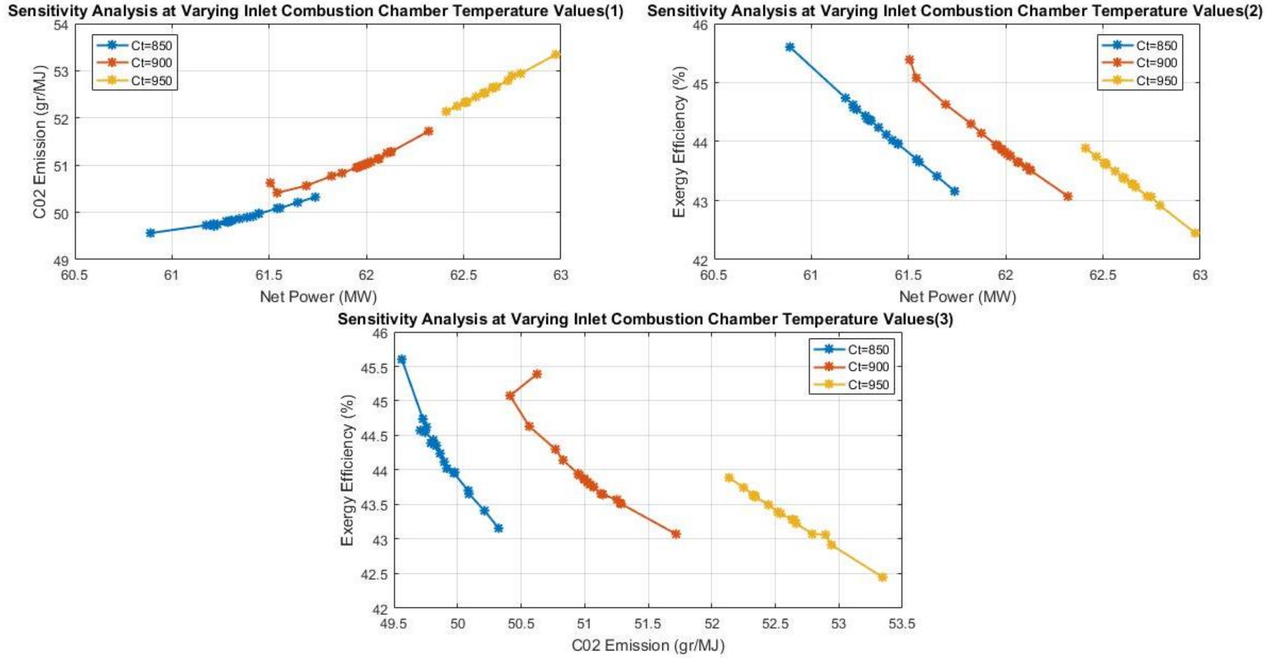


Figure 4.8: Effects of combustion chamber inlet temperature on the optimal objective functions

4.2 Harris Hawks Optimization

4.2.1 Single-objective optimization

The single-objective HHO optimization was performed on each of the objective functions for 100 iterations using a search agent number of 100. The MATLAB code was executed on a laptop equipped with an Intel Core i3-5005U CPU @ 2.00GHz and 8GB of RAM. The associated codes have been reported in Appendix C.

4.2.1.1 Net Power Optimization

The solution of Equation (3.13) involved using the decision variables and their corresponding values outlined in Tables 3.1 and 3.2. The HHO (Harris Hawks Optimization) was employed for this purpose as shown in appendix C. The outcomes, as presented in Table 4.7, illustrate the necessary actions to maximize net power. The following results indicate the steps required for optimizing net power:

- Minimize the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature.

- Maximize the inlet turbine temperature.

Table 4.7: Optimal solutions maximizing the net power (HHO).

Algorithm	Cr	Pp	Gt	Ct	Maximum power
HHO	10	10	1520	850	60.8552

4.2.1.2 CO_2 Emission Optimization

The HHO algorithm was used to solve Equation (3.14) by utilizing the decision variables and their corresponding values listed in Tables 3.1 and 3.2. The results of this optimization process are shown in Table 4.8, which presents the required actions to minimize CO_2 emissions. The following outcomes indicate the necessary steps that need to be taken to optimize and reduce the levels of CO_2 emissions:

- Minimize the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature.
- Maximize the inlet turbine temperature.

Table 4.8: Optimal solutions minimizing CO_2 emission (HHO).

Algorithm	Cr	Pp	Gt	Ct	Minimum CO_2
HHO	10	10	1520	850	49.3771

4.2.1.3 Exergy Efficiency Optimization

The efficiency of exergy was calculated using Equation (3.15), with the HHO algorithm utilizing the decision variables and values presented in Tables 3.1 and 3.2. The following outcomes indicate the necessary steps that need to be taken to maximize the exergy efficiency:

- Minimize the compression ratio, pinch point temperature difference, and inlet turbine temperature.
- Maximize the inlet combustion chamber temperature.

Table 4.9: Optimal solutions maximizing the exergy efficiency (HHO).

Algorithm	Cr	Pp	Gt	Ct	Minimum CO_2
HHO	10	10	1420	950	42.3507

4.2.1.4 Analysis of the Single-Objective Optimization Results

The results obtained in Tables 4.7 to table 4.9 have been summarized in Table 4.10, which shows the behavior of the optimal solutions. As shown in table 4.10, an upward trend is represented by ↑, a downward trend is represented by ↓ while a conflict between the parameters is indicated by ≠. It could be seen that for decision variables Cr and Pp, all three objective functions are in harmony. However, for the Gt and Ct decision variables, the exergy efficiency conflicts with the net power and CO_2 emission. Thus, a simultaneous optimization of the three objectives is required to determine the Pareto optimal solutions.

Table 4.10: Illustration of the trend of the objective functions (HHO).

Decision variable	Net power	CO_2 emission	Exergy efficiency
Cr	↓	↓	↓
Pp	↓	↓	↓
Gt	↑	↑	↓≠
Ct	↓	↓	↑≠

4.2.2 Multi-objective optimization

The optimization algorithm has produced a hundred non-dominated solution sets, as depicted in Figure 4.9 by the Pareto front of all three objective functions. The ensuing optimal solution sets provide the best feasible trade-offs in spite of the inherent conflicts between these objectives. Every solution on the Pareto front is potentially optimal, and the final decision rests at the discretion of the decision maker. The associated codes have been presented in appendix D. Table 4.11 shows the MOHHO parameters employed.

Table 4.11: Parameters used by the MOHHO algorithm

S/N	Parameter	Value
1	Maximum Number of Iterations (MaxIt)	100

2	Population Size (nPop)	100
3	Repository Size (nRep)	100
4	Inertia Weight	0.5
5	Inertia Weight Damping Rate	0.99
6	Personal Learning Coefficient	1
7	Global Learning Coefficient	2
8	Number of Grids per Dimension (nGrid)	4
9	Inflation Rate (alpha)	0.1
10	Leader Selection Pressure (beta)	2
11	Deletion Selection Pressure(gamma)	2
12	Mutation Rate (mu)	0.1

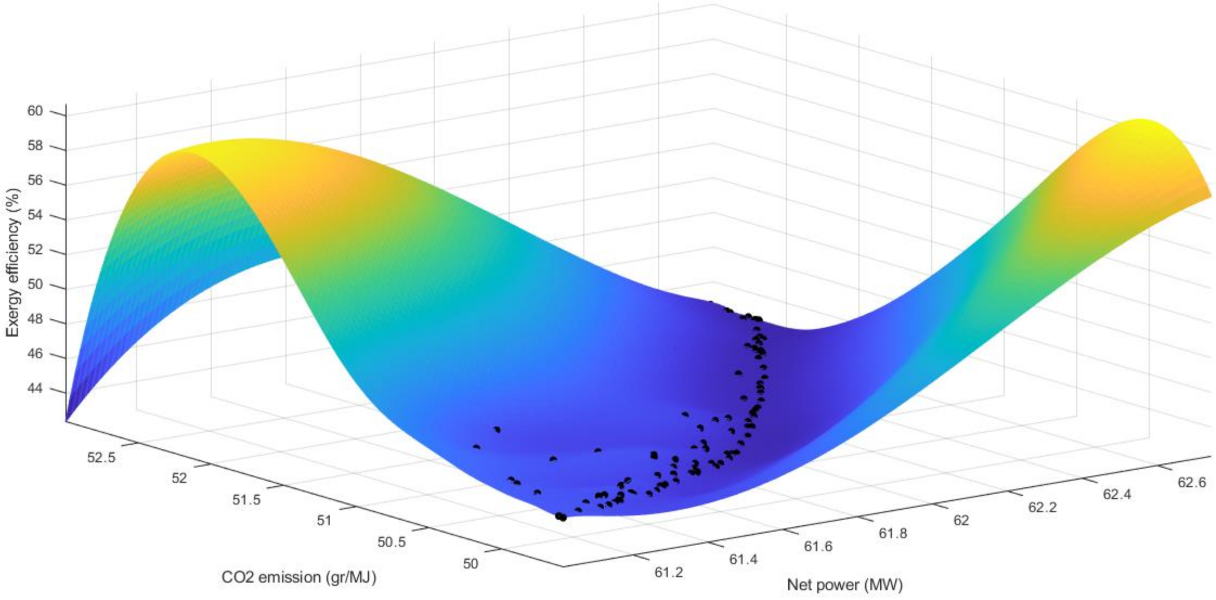


Fig 4.9: Pareto front of CO_2 emission vs net power vs exergy efficiency (MOHHO).

Three scenarios were looked at in order to obtain a better understanding of the relationship between each objective function. The relationships between CO_2 emissions and net power output, exergy efficiency and net power output, and exergy efficiency and CO_2 emissions were investigated in these scenarios. For clarity, the overall outcomes have been condensed into three two-dimensional Pareto fronts that emphasize crucial aspects. The Pareto front, which shows a conflict in the decision-making process, is shown in Figure 4.10. Increased CO_2 emissions will unavoidably result from the goal of maximizing net power output, which may have detrimental consequences on the environment. In the CCHP system under study, the net power output is significantly influenced by the power generated by the GT-3 and then the Kalina cycle. As a result, the system can be designed to make the combustion gases entering the unfired HRSG from APH-2 hotter than they are when they enter the combustion chamber. Additionally, it is possible to decrease the fuel consumption rate and enhance the power produced by the photovoltaic thermal collectors at the same time. The principal power source of the system, the GT-3, produces less net power as a result of these modifications, which are intended to minimize CO_2 emissions. However, there will be an increase in the cleaner, non-combustible power output from the Kalina cycle. Even if in the end this method lowers the net power generated, it is optimized to minimize CO_2 emissions.

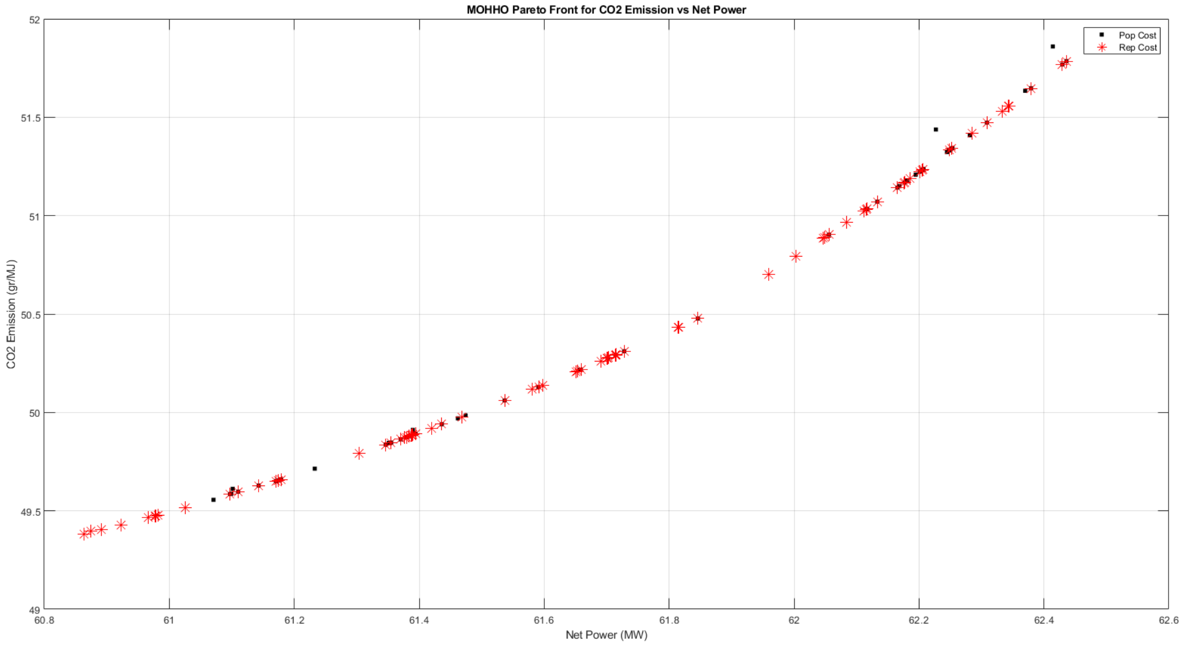


Fig 4.10: Pareto front of CO_2 emission vs net power (MOHHO).

According to the Pareto graph shown in Figure 4.11, there is a compatibility between exergy efficiency and CO_2 emission. An increased energy efficiency is correlated with a lower CO_2 emission value. The graph also shows that in order to significantly boost energy efficiency, CO_2 emissions must be reduced to a certain extent. Further analysis indicates that reducing fuel consumption would result in a decrease in the fuel's input exergy. This, in turn, would improve the exergy outputs and useful work from the Kalina cycle, water heat exchanger, and absorption chiller, thus significantly maximizing the exergy efficiency. Consequently, there would be very little CO_2 emissions.

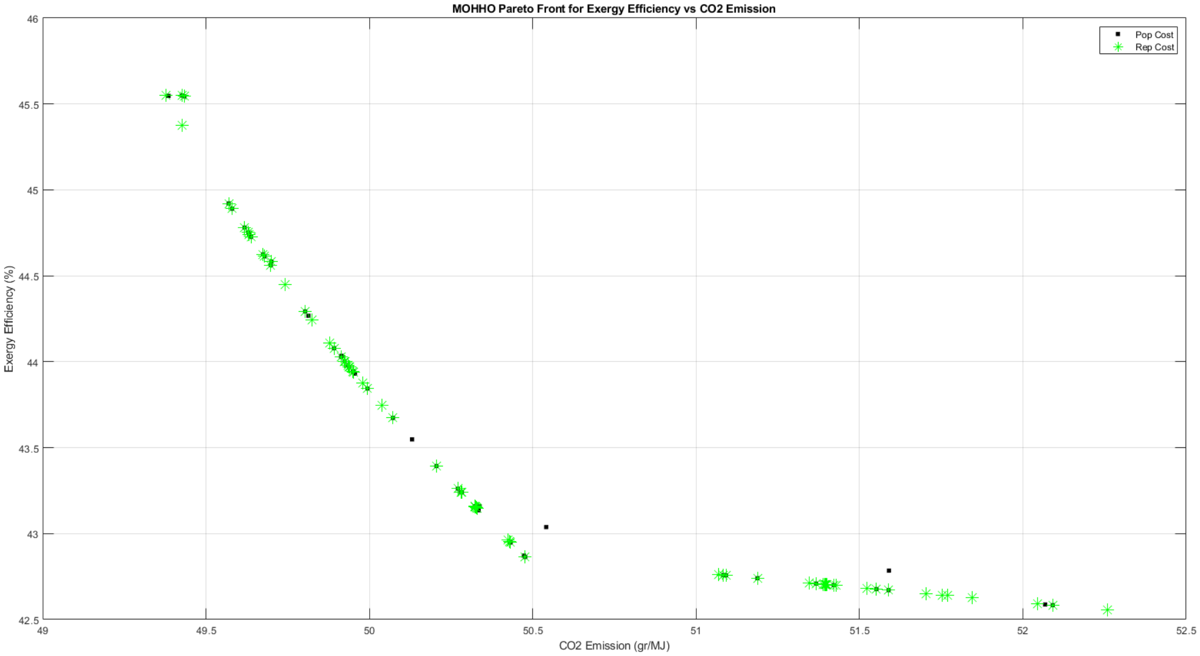


Fig 4.11: Pareto front of exergy efficiency vs CO_2 emission (MOHHO).

In addition, as depicted in Figure 4.12, there is an inverse correlation between the exergy efficiency and net power. This suggests that a system may not be highly efficient even though it produces a lot of net power. Rather, maximum exergy efficiency is instead determined by how well the power generated by the CCHP system is used. Consequently, it is advisable to operate and design bottom systems (such as the Kalina cycle and absorption chiller) in a manner that ensures maximum waste heat utilization and minimized exergy losses. Figure 4.12 illustrates the Pareto front for exergy efficiency versus CO_2 emission.

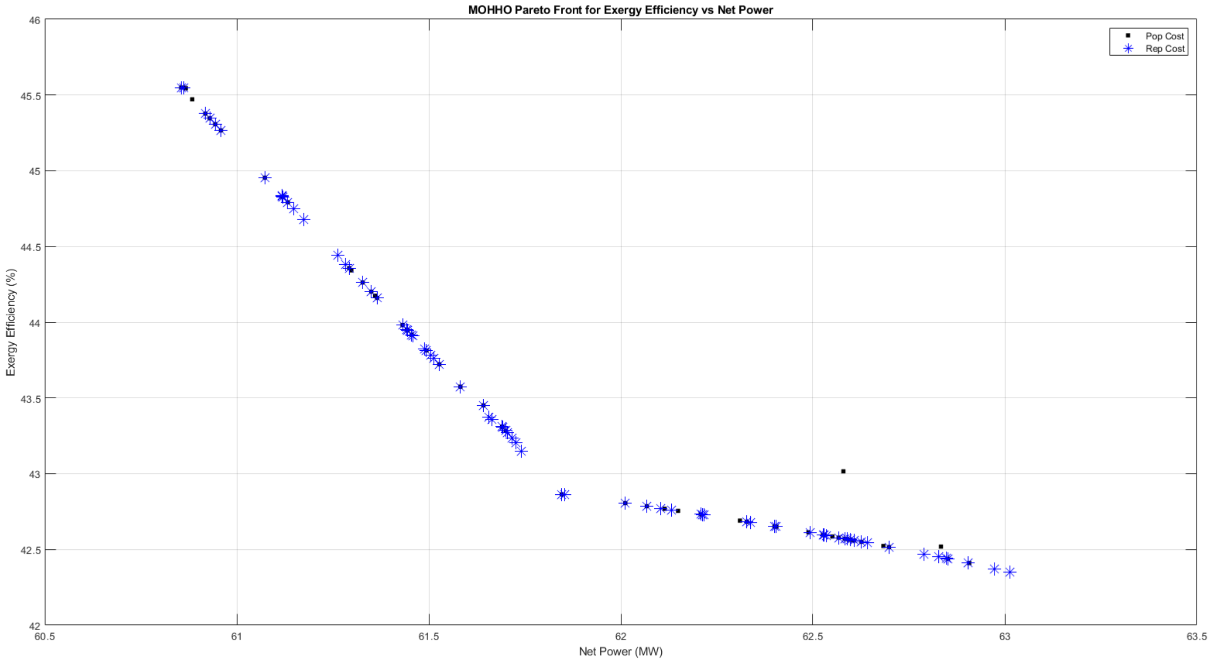


Fig 4.12: Pareto front of exergy efficiency vs net power (MOHHO).

The constraints placed on the HHO optimization techniques, and the convergence of each choice variable are shown in Figure 4.13's scatter plot distribution. The results show that lower compression ratios and pinch point temperatures are preferred by the suggested HHO optimization strategy. Values in the range between the minimum and the midpoint were selected, by the optimization method, for the combustion chamber inlet temperature. The scatter plot for the gas turbine inlet temperature is distributed between the constraint's upper and lower bounds.

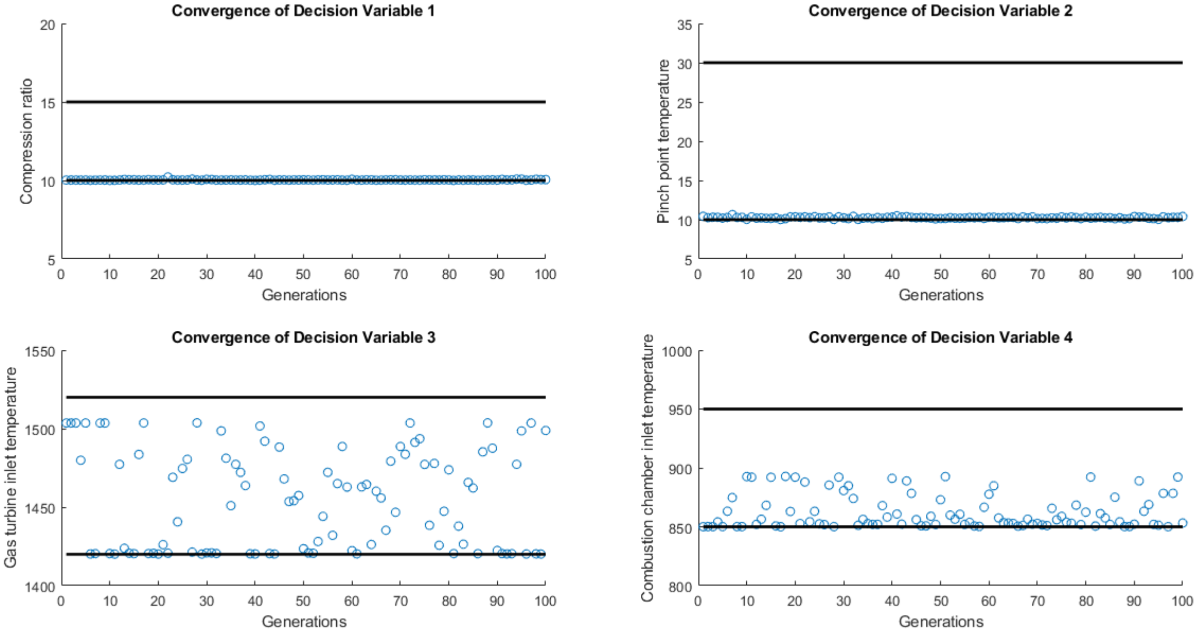


Fig 4.13: Scatter distribution of the decision variables (MOHHO).

The MOHHO was designed to produce six ideal solutions, which were then compared with the outcomes of other research of a similar nature. The data gathered, detailed in Table 4.12, indicates the MOHHO, and MOGWO produced CO_2 emissions ranging from 50.16 gr/MJ to 50.66 gr/MJ, which is significantly lower than the higher values produced by the RSM. The MOGWO was found to produce the least CO_2 emission of 50.16 gr/MJ. The net power yielded by the three methods was within a small range of 61.32 MW to 61.93 MW with the MOHHO providing the configuration that yielded the highest net power of 61.91 MW. In terms of the exergy efficiency, each of the three method configurations with their best efficiency values ranging from 44.12 % to 45.32% with the MOGWO yielded the highest exergy efficiency value.

Table 4.12: Comparison of optimization results with a related study.

	Optimal Decision Variables				**Optimal Objective Functions**		
	Cr	**Pp**	**Gt**	**Ct**	**Net Power**	**CO_2 Emission**	**Exergy Efficiency**
MOHHO	10	10	1420	855.34	**61.91**	50.61	42.84
	10	10	1422.67	850	61.82	50.44	42.94
	10.28	11.14	1514.57	896.17	61.52	50.66	45.28

	10	10	1426.26	850	61.78	50.38	43.04
	10.01	10.77	1469.76	870.77	61.60	50.33	44.15
	10.28	11.14	1514.31	896.01	61.52	50.66	45.28
MOGWO result (table 4.6)	10.00	10.86	1520	913.11	61.60	50.57	45.21
	10.00	10.56	1520	901.58	61.47	50.31	45.28
	10.00	10.46	1520	895.34	61.40	50.18	45.31
	10.01	10.46	1520	893.99	61.38	**50.16**	**45.32**
	10.00	10.68	1520	908.14	61.55	50.45	45.24
	10.00	10.68	1520	907.69	61.54	50.44	45.24
A similar study with RSM (Mahdavi, Mojaver & Khalilarya, 2022)	11.66	11.96	1470	900	61.73	52.87	44.22
	11.11	20.00	1470	900	61.73	52.99	44.09
	11.98	20.00	1470	890	61.75	53.84	44.12
	12.50	16.10	1484	900	61.75	54.07	44.58
	12.50	15.72	1470	891	61.79	54.07	44.10
	12.50	20.00	1468	882	61.75	54.28	44.30

4.3 Ant Lion Optimization

4.3.1 Single-objective optimization

For 100 iterations, a search agent number of 100 was used to carry out the single-objective ALO optimization on each of the objective functions. A laptop with an Intel Core i3-5005U CPU running at 2.00GHz and 8GB of RAM was used to run the MATLAB code. The associated codes are presented in Appendix E.

4.3.1.1 Net Power Optimization

Equation 3.13 had to be solved by applying the ALO method to the decision variables and their associated values listed in Tables 3.1 and 3.2. The results, which are shown in Table 4.13, show what should be done in order to optimize net power. The subsequent findings delineate the necessary measures for optimizing net power.:

- Minimize the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature.
- Maximize the inlet turbine temperature.

Table 4.13: Optimal solutions maximizing the net power (ALO).

Algorithm	Cr	Pp	Gt	Ct	Maximum power
ALO	10	10	1520	850	60.8552

4.3.1.2 CO_2 Emission Optimization

Equation 3.14 was solved by applying the ALO method with the decision variables and values specified in Tables 3.1 and 3.2. Table 4.14, which lists the necessary steps to reduce CO_2 emissions, displays the outcomes of this optimization process. The ensuing results point to the actions that must be performed in order to maximize and minimize CO_2 emissions.:

- Minimize the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature.
- Maximize the inlet turbine temperature.

Table 4.14: Optimal solutions minimizing CO_2 emission (ALO).

Algorithm	Cr	Pp	Gt	Ct	Minimum CO_2
ALO	10	10	1520	850	49.3771

4.3.1.3 Exergy Efficiency Optimization

Equation 3.15 was used to determine the efficiency of exergy, and the ALO algorithm made use of the values and decision variables listed in Tables 3.1 and 3.2. The subsequent results delineate the essential measures that must be implemented in order to optimize energy efficiency.:

- Minimize the compression ratio, pinch point temperature difference, and inlet turbine temperature.
- Maximize the inlet combustion chamber temperature.

Table 4.15: Optimal solutions maximizing the exergy efficiency (ALO).

Algorithm	Cr	Pp	Gt	Ct	Minimum CO_2
ALO	10	10	1420	950	42.3507

4.3.1.4 Analysis of the Single-Objective Optimization Results

Table 4.16 summarizes the data found in Tables 4.13 through 4.15 and displays how the optimal solutions behave. Table 4.16 illustrates this with the following trends: ↑ indicates an increasing trend, ↓ indicates a downward trend, and ≠ indicates a disagreement between the parameters. It is evident that all three objective functions are in harmony for the decision variables Cr and Pp. Nonetheless, the exergy efficiency conflicts with the net power and CO_2 emission for the Gt and Ct decision factors. To find the Pareto optimal solutions, the three objectives must be optimized simultaneously.

Table 4.16: Illustration of the trend of the objective functions (ALO).

Decision variable	Net power	CO_2 emission	Exergy efficiency
Cr	↓	↓	↓
Pp	↓	↓	↓
Gt	↑	↑	↓≠
Ct	↓	↓	↑≠

4.3.2 Multi-objective optimization

The Pareto front of each of the three objective functions in Figure 4.14 shows that the optimization process has generated one hundred non-dominated solution sets. Despite the inherent conflicts between these aims, the resulting optimal solution sets offer the best possible trade-offs. Each Pareto-front solution has the potential to be the best one, and the choice of which to choose ultimately belongs to the decision-maker. The used MOALO parameters are displayed in Table 4.17.

Table 4.17: Parameters used by the MOALO algorithm.

S/N	Parameter	Value

1	Maximum Number of Iterations (max_iter)	100
2	Population Size (N)	100
3	Archive Size	100

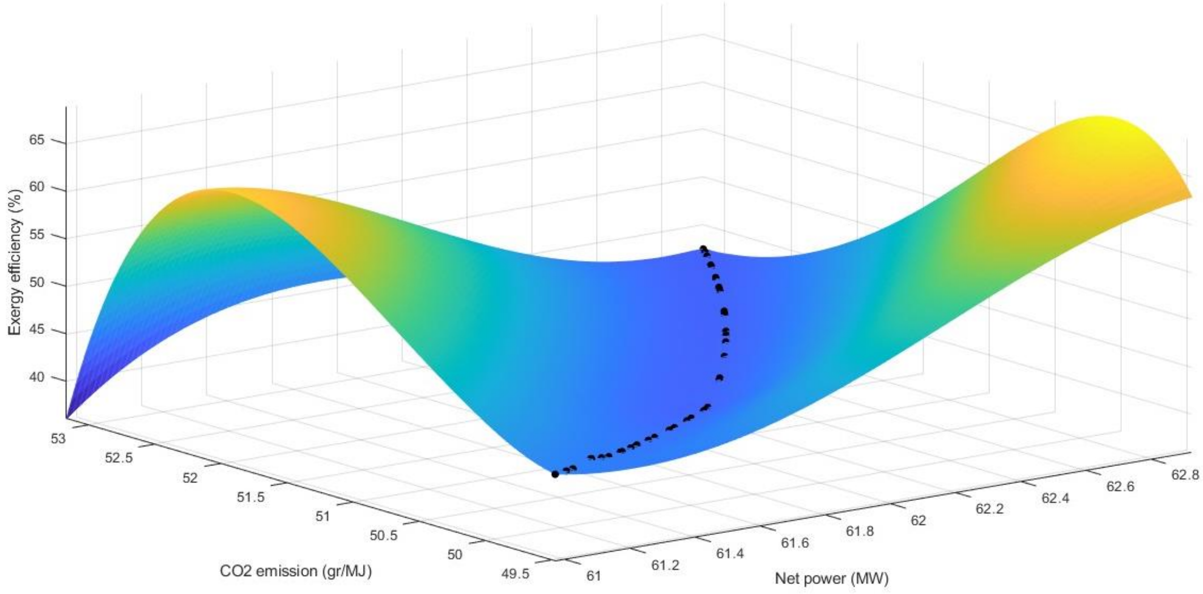


Fig 4.14: Pareto front of CO_2 emission vs net power vs exergy efficiency (MOALO).

To have a better understanding of the relationship between each objective function, three situations were examined. In these situations, the correlations between exergy efficiency and net power output, exergy efficiency and CO_2 emissions, and exergy efficiency and net power output were examined. To make things more understandable, the total results have been reduced to three two-dimensional Pareto fronts that highlight important features. Figure 4.15 displays the Pareto front, which illustrates a conflict in the decision-making process between net power and CO_2 emission. The objective of optimizing net power output will inevitably lead to higher CO_2 emissions, which could have negative environmental effects. In the CCHP system under investigation, the power produced by the GT-3 and subsequently the Kalina cycle have a major impact on the net power output.

Consequently, the combustion gases entering the unfired HRSG from APH-2 can be made hotter in the system design than they are upon entering the combustion chamber. Furthermore, it is feasible to simultaneously increase the power generated by the photovoltaic thermal collectors and decrease their rate of fuel consumption. These changes, which are meant to reduce CO_2 emissions, result in less net power being produced by the system's main power source, the GT-3. On the other hand, the Kalina cycle will produce more power that is cleaner and non-combustible. Despite the fact that this approach ultimately reduces net electricity output, it is tailored to reduce CO_2 emissions.

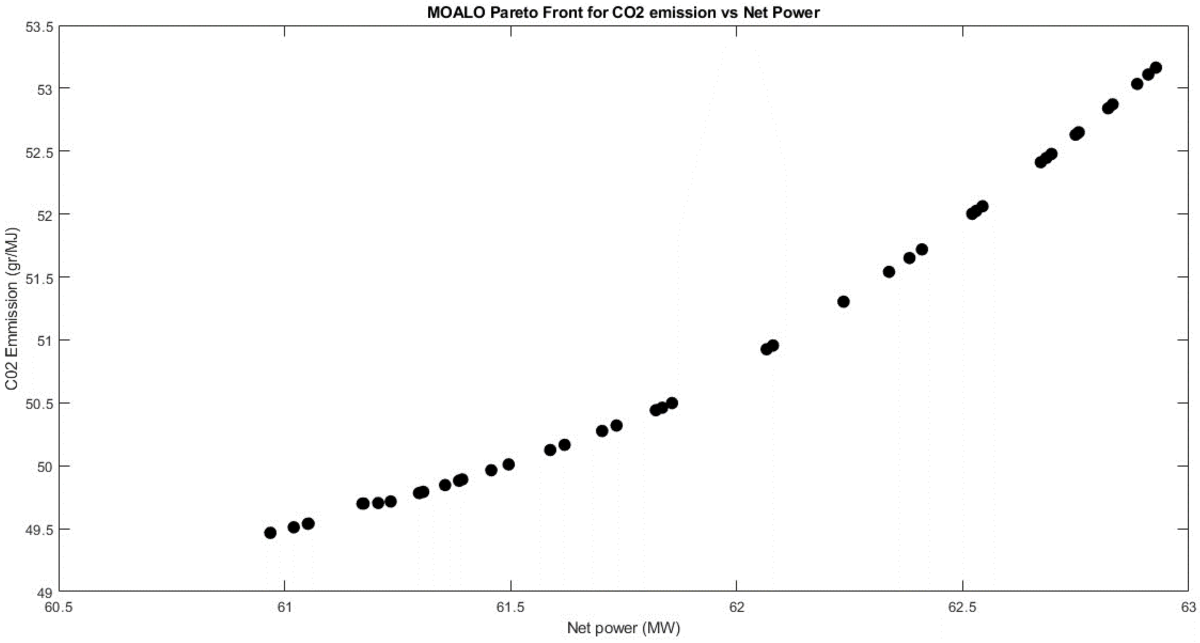


Fig 4.15: Pareto front of CO_2 emission vs net power (MOALO).

The Pareto graph in Figure 4.16 indicates that energy efficiency and CO_2 emissions are compatible. A lower value of CO_2 emissions relates to higher exergy efficiency. The graph also demonstrates that CO_2 emissions must be reduced to a certain extent in order to greatly increase exergy efficiency. Additional investigation shows that cutting fuel use would lower fuel's input exergy. This will greatly maximize the exergy efficiency by enhancing the exergy outputs and useful work from the absorption chiller, water heat exchanger, and Kalina cycle. As a result, relatively little CO_2 would be released.

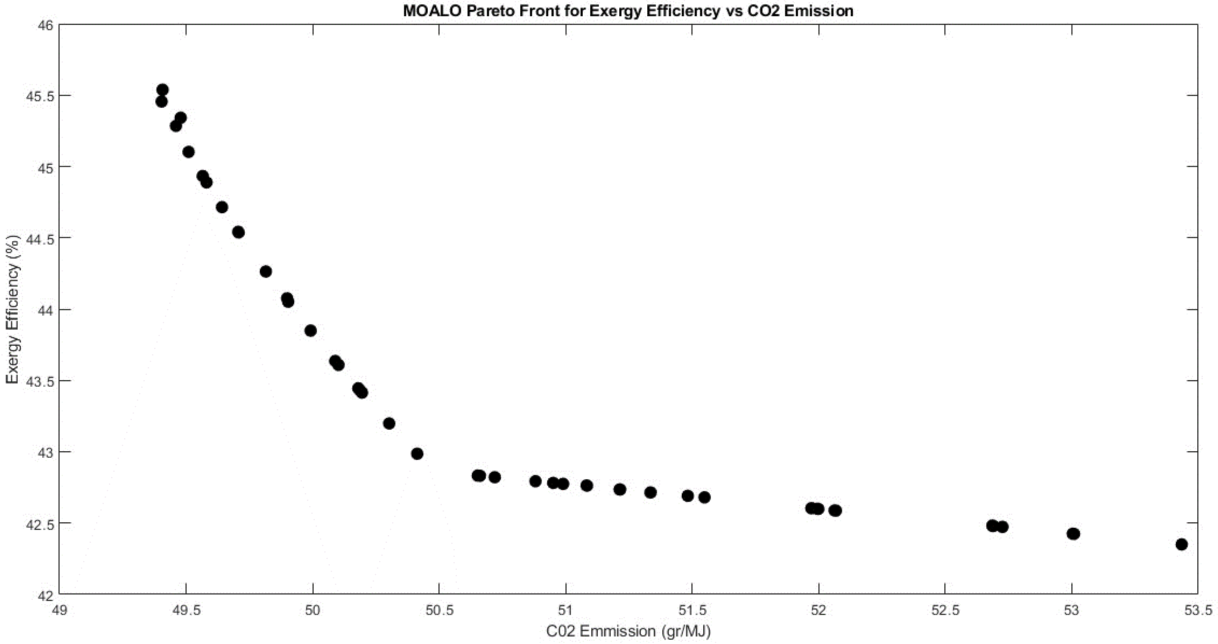


Fig 4.16: Pareto front of exergy efficiency vs CO_2 emission (MOALO).

Furthermore, there is an inverse relationship between exergy efficiency and net power, as seen in Figure 4.17. This implies that even while a system generates a lot of net power, it may not be very efficient. Maximum exergy efficiency is instead determined by how effectively the CCHP system's power is used. Thus, bottom systems (such the Kalina cycle and absorption chiller) should be designed and operated to maximize waste heat utilization and reduce energy losses. The Pareto front for exergy efficiency versus CO_2 emissions is shown in Figure 4.17.

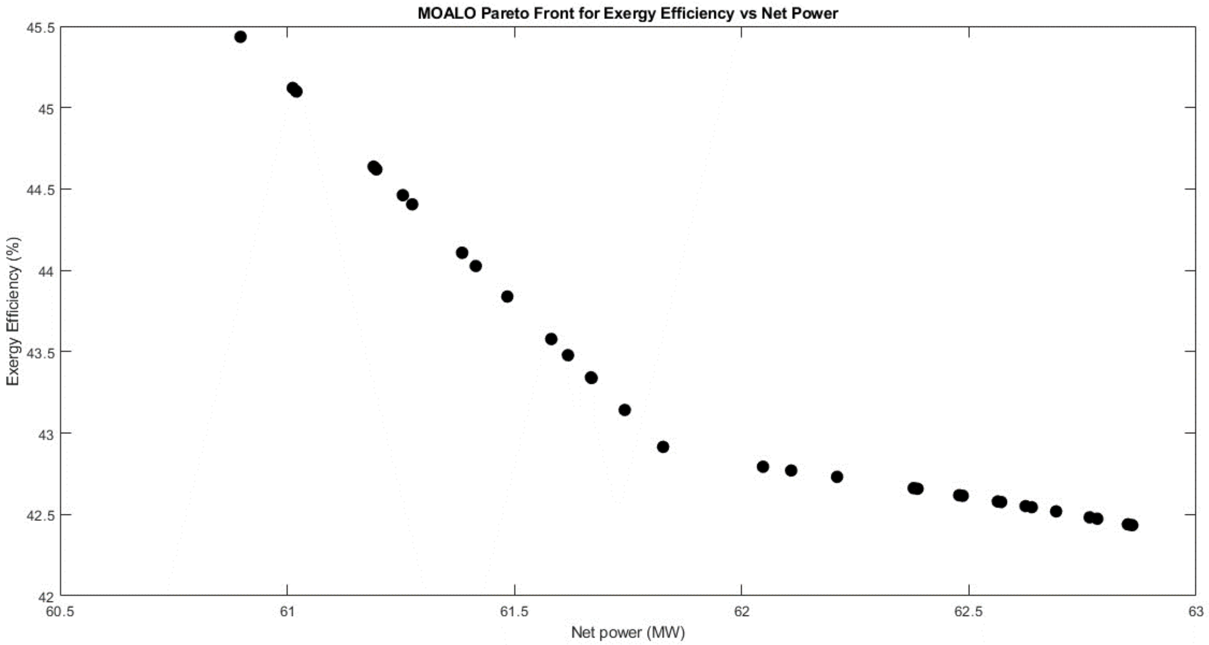


Fig 4.17: Pareto front of exergy efficiency vs net power (MOALO).

The constraints placed on the MOALO techniques, and the convergence of each decision variable are shown in Figure 4.18's scatter plot distribution. The results show that lower compression ratios, pinch point temperatures and combustion chamber inlet temperatures are preferred by the suggested MOALO strategy (10, 10 and about 850 respectively). Values close to the maximum (around 1520) were selected, by the optimization method, for the gas turbine temperature.

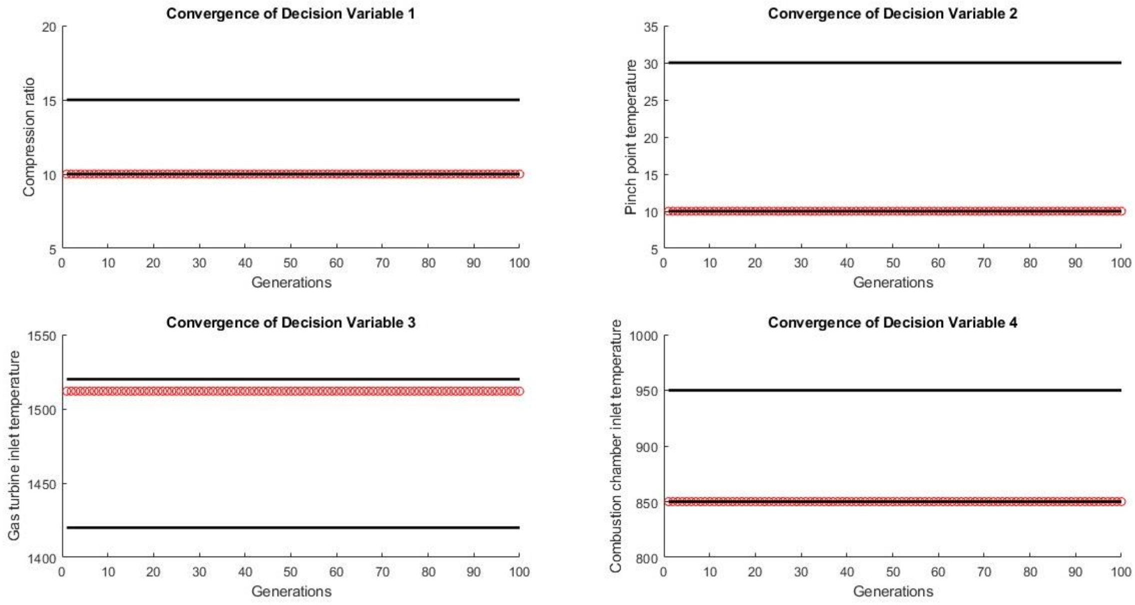


Fig 4.18: Scatter distribution of the decision variables (MOALO).

The MOALO was designed to produce six optimal solutions, which were then evaluated with the outcomes of other research of a similar nature. The data gathered, detailed in Table 4.18, indicates the MOALO, MOHHO, and MOGWO produced CO_2 emissions ranging from 49.40 gr/MJ to 50.66 gr/MJ, which is significantly lower than the higher values produced by the RSM. The MOALO was found to produce the least CO_2 emission of 49.40 gr/MJ. The net power yielded by the four methods was within a range of 60.90 MW to 61.91 MW with the MOHHO providing the configuration that yielded the highest net power of 61.91 MW. In terms of the exergy efficiency, each of the four method configurations with their best efficiency values ranging from 44.12 % to 45.45% with the MOALO yielded the highest exergy efficiency value. In summary, the MOALO produced the best CO_2 emission and exergy efficiency values and the least net power output when compared with the other four methods.

Table 4.18: Comparison of optimization results with a related study.

	Optimal Decision Variables				**Optimal Objective Functions**		
	Cr	**Pp**	**Gt**	**Ct**	**Net Power**	**CO_2 Emission**	**Exergy Efficiency**
MOALO	10.00	10.01	1513.82	850.00	60.93	49.43	45.36
	10.00	10.01	1513.82	850	60.94	49.44	45.32
	10.00	10.01	1513.83	850.00	60.90	49.41	45.43
	10	10.01	1513.82	850	60.90	49.41	45.43
	10	10.01	1513.82	850.00	60.90	49.41	45.43
	10.00	10.01	1513.83	850	60.89	**49.40**	**45.45**
MOHHO result (Table 4.12)	10	10	1420	855.34	**61.91**	50.61	42.84
	10	10	1422.67	850	61.82	50.44	42.94
	10.28	11.14	1514.57	896.17	61.52	50.66	45.28
	10	10	1426.26	850	61.78	50.38	43.04
	10.01	10.77	1469.76	870.77	61.60	50.33	44.15
	10.28	11.14	1514.31	896.01	61.52	50.66	45.28
MOGWO result (Table 4.6)	10.00	10.86	1520	913.11	61.60	50.57	45.21
	10.00	10.56	1520	901.58	61.47	50.31	45.28

	10.00	10.46	1520	895.34	61.40	50.18	45.31
	10.01	10.46	1520	893.99	61.38	50.16	45.32
	10.00	10.68	1520	908.14	61.55	50.45	45.24
	10.00	10.68	1520	907.69	61.54	50.44	45.24
A similar study with RSM (Mahdavi, Mojaver & Khalilarya, 2022)	11.66	11.96	1470	900	61.73	52.87	44.22
	11.11	20.00	1470	900	61.73	52.99	44.09
	11.98	20.00	1470	890	61.75	53.84	44.12
	12.50	16.10	1484	900	61.75	54.07	44.58
	12.50	15.72	1470	891	61.79	54.07	44.10
	12.50	20.00	1468	882	61.75	54.28	44.30

4.6 Findings and Conclusion

The objective of this research study was to optimize a solar-assisted CCHP system multi-objectively using the greywolf, harris hawks and ant lion approaches. By optimizing the net power and energy efficiency of the CCHP system, its overall efficiency was improved. Additionally, it aimed to minimize greenhouse gas emissions, which are known to have detrimental effects on human health and contribute to global warming. Given that a solar-assisted CCHP was optimized, this research lends credence to the notion that the rate at which fossil fuels are depleting must be slowed down. This work suggests a new optimization approach for the multi-objective optimization of a solar-based CCHP system, as related studies have used other optimization approaches. The following is an overview of the research's findings and outcomes:

- The optimal set of parameters characterizing the thermodynamic configuration of the solar-powered CCHP system is found by applying a multi-objective optimization technique. These parameters include the compression ratio, pinch point temperature difference, intake turbine temperature, and inlet combustion chamber temperature.
- The net power, CO_2 emission, and energy efficiency are used as objective functions to evaluate the effectiveness of the CCHP system by gauging how well each set of decision variables complies with all the constraints.

- The multi-objective greywolf, harris hawks and ant lion optimization methods are used to produce non-dominated Pareto optimum solutions for multi-objective optimization.
- This study computes a set of Pareto optimum solutions. In order to let the decision-maker select a preferred choice based on their discretion and enhance the functionality of the CCHP system, the optimal solutions are presented as possibilities. Conducting a sensitivity analysis suggests a roadmap to help with this decision-making process.
- The interdependency of the four decision variables is an intriguing finding. This implies that when one of the decision variables changes, the other three variables also change accordingly. Therefore, it is relevant and beneficial to use a multi-objective optimization technique for assessing the CCHP system's performance.
- According to this study, a maximum net power corresponds with an undesirable maximum CO_2 emission, hence, there is a conflict in the decision-making processes between net power and CO_2 emission. The compatibility between exergy efficiency and CO_2 emission, which shows that a system with low CO_2 emissions is extremely efficient, is another significant finding. Because of the inverse relationship found between net power and energy efficiency, this study has demonstrated that a system's high net power production is not a guarantee of its high efficiency.
- The results of the sensitivity research indicate that, while having minimal effect on the ideal net power and energy efficiency values, a lower compression ratio will drastically reduce CO_2 emissions. Additionally, it was discovered that although a system with higher turbine inlet temperature values may be more efficient and release less CO_2, the system will have a lower net power. This suggests that in order to obtain the lowest possible CO_2 emission values and the highest possible energy efficiency, which corresponds to the lowest possible net power, lower values of the combustion chamber inlet temperature are required.
- The results obtained from analyzing the convergence of decision variables indicate that minimizing the compression ratio, pinch point temperature difference, and inlet combustion chamber temperature leads to optimized objective values. However, it's noteworthy that the ranges for the gas turbine temperature are spread across the upper and lower bounds of the constraints. This dispersion suggests challenges in achieving convergence for this variable, hinting at potential difficulties in its optimization.

Consequently, additional techniques may be necessary to address these challenges and achieve convergence for the gas turbine temperature within the defined constraints.

- The study supports the findings of Mahdavi, Mojaver & Khalilarya (2022), who discovered that, due to its largest incremental change, the compression ratio had the greatest impact on CO_2 emissions. Similarly, the inlet temperature of the combustion chamber had the greatest influence on net power and the inlet turbine temperature had the greatest impact on exergy efficiency.

Chapter Five – Conclusion and Recommendation

7.1 Conclusion

The research study succeeded in employing three meta-heuristic optimization methods for the optimization of a solar-assisted combined cooling, heating, and power system. It aimed to improve the overall efficiency of the solar-assisted CCHP system by maximizing net power output and exergy efficiency while minimizing CO_2 emission by considering decision variables such as compression ratio, pinch point temperature difference, gas turbine temperature, and combustion chamber inlet temperature. It attempted to ascertain the effects of each decision variable on the objective functions.

The study commenced by emphasizing the fundamental role of energy generation while also pointing out the need for renewable energy sources. It highlighted the need for a CCHP system due to its benefit of fuel efficiency and the fact that it can generate more than one form of energy among others. It also introduced the concept of optimization to improve the objectives of the system. Furthermore, an extensive review of the literature was conducted by explaining the theoretical concepts of a multi-generation system, the CCHP system as well as its subsystems, classification, and theory underlying the use of solar energy in CCHP systems. Relevant research works that utilized MOGWO, MOHHO, and MOALO were reviewed to conclude the literature review section. The methodology followed through with the description of each optimization technique (MOGWO, MOHHO, and MOALO) in addition to the description of the solar-assisted CCHP system to be optimized, objective function equations, constraints, and mathematical formulation of the optimization problem. Subsequently, the application of each optimization technique on the solar-assisted CCHP system was reported.

All three optimization techniques had similar pareto fronts as they reported conflicting objectives hence the need for multi-objective optimization. Also, there was interdependency among all four decision variables which implied that a change in one of them affected the other three. Furthermore, there was a conflict in the decision-making process between net power and CO_2 emission by virtue of the fact that a maximum net power corresponds to an undesirable maximum CO_2 emission. Based on this finding, the research study suggested increasing the temperature of the combustion gases entering the unfired HRSG compared to that entering the combustion chamber and increasing the heating power provided by the solar thermal collectors. In the same vein, there was an observed conflict between the net power output and the exergy efficiency which suggested that maximum power production does not equate to a highly efficient system. Following this, this study put forward that bottom systems such as absorption chillers, Kalina cycle, and HRSG systems should be designed and operated in such a way as to maximize the use of waste heat thus reducing energy losses to the surroundings. Furthermore, there was a noticeable compatibility between exergy efficiency and CO_2 emission in the sense that a system with minimum CO_2 emission relates to a highly efficient system. Sequel to the foregoing, it was suggested to reduce fuel usage which will in turn reduce the input exergy and improve exergy output from the bottom systems, all geared towards boosting the exergy efficiency and yielding a relatively low CO_2 emission.

The sensitivity analysis conducted revealed that a low compression ratio will significantly reduce the CO_2 emission and therefore it has the most effect on the CO_2 emission compared to other decision variables. In addition, the pinch point temperature difference had the least effect on all objective functions. Furthermore, it was observed that a CCHP system with maximum gas turbine temperature emits the least CO_2 and this translates to a high exergy efficiency. Therefore, it was revealed that the gas turbine temperature has the greatest influence on the exergy efficiency of the CCHP system. Also, this analysis revealed that a minimum inlet combustion chamber temperature is required to achieve a minimum CO_2 emission and maximum exergy efficiency as well as the fact that the inlet combustion chamber temperature has the greatest effect on the net power output.

Furthermore, an analysis conducted on the convergence of each decision variable across the number of iterations revealed that the Antlion optimization technique aligns with the findings from the sensitivity analysis. Full convergence by the algorithm was reported as the inlet gas turbine temperature converged to a maximum value while the compression ratio, pinch point temperature, and inlet combustion chamber converged to their minimum values. The grey wolf optimization technique agrees with the notion of converging a maximum gas turbine temperature as well as a minimum compression ratio and pinch point temperature difference temperature. However, its inlet combustion chamber temperature values did not achieve full convergence as it ranged from 850 to 950. Similarly, the Harris hawks algorithm did not achieve full convergence for the gas turbine temperature values as they ranged between 1420 and 1520. However, it agrees with the minimum values for compression ratio, pinch point temperature difference, and inlet combustion chamber temperature. Hence, it is recommended in this study that an additional technique should be employed on the greywolf and Harris Hawks methods to achieve convergence of the inlet combustion chamber and gas turbine temperatures respectively within the specified constraints.

Comparing all three meta-heuristic optimization techniques, the Antlion produced the decision variable configuration that gave the best CO_2 emission and exergy efficiency of 49.40 gr/MJ and 45.45% respectively. Although it also had the best runtime value of 3.02 seconds, it reported the least net power output (60.89MW). The Harris hawks optimization technique produced the decision variable configuration that yielded the best net power output of 61.91 MW but this was at the cost of having the highest CO_2 emission and least exergy efficiency of 50.61 gr/MJ and 42.84% respectively compared to the other two methods. The grey wolf optimization technique attempted to strike a balance between net power output and exergy efficiency which are incompatible objective functions. This is because it surprisingly produced a high exergy efficiency and a high net power simultaneously which was not the case in the other two optimization methods. Notwithstanding, it reported the highest runtime of 145.74 seconds. In comparison with the response surface method used to evaluate the meta-heuristic techniques, it could be seen that the latter generally produced lower CO_2 emission and higher exergy efficiency as well as a relatively smaller net power output (average standard deviation of 0.39). Hence, the above research finding would suffice to conclude that the meta-heuristic techniques generally outperformed the RSM method (used in a contemporary study) in carrying out the maximization of net power and exergy efficiency and minimization of CO_2 emission.

Finally, the optimization techniques employed in this study provided sets of 100 Pareto solutions and it is believed that in addition to other analyses conducted, the decision will be fully informed in proffering solutions required to enhance the performance of the solar-assisted CCHP system.

7.2 Recommendations

As a recommendation, other meta-heuristic optimization techniques such as the genetic algorithm, particle swarm optimization may be explored to provide improved performance of the solar-assisted-CCHP system. Predictive control methods such as the model predictive control approach, and fuzzy logic control can be integrated with solutions from computational techniques such as meta-heuristic methods, to dynamically adjust the CCHP system's configuration to changing environmental conditions and user/decision maker's expectations. Furthermore, validating the obtained optimal results from meta-heuristic methods with experimental results would give more credence to the effectiveness of the proposed solution in a real-world setting. Finally, as a future direction, incorporating other renewable energy sources such as wind and biomass in addition to the integration of renewable energy storage systems would greatly reduce the issue of variability in energy supply commonly experienced with relying on solar energy as the only renewable energy source.

References

1. Abualigah, L. 2020 Multi-verse optimizer algorithm: A comprehensive survey of its results, variants, and applications. *Neural Comput*. *Appl.*, 32, 12381–12401
2. Abul'Wafa, A.R., 2019. Ant-lion optimizer-based multi-objective optimal simultaneous allocation of distributed generations and synchronous condensers in distribution networks. *International Transactions on Electrical Energy Systems*, *29*(3), p.e2755.
3. Aghaziarati, Z. and Aghdam, A.H., 2021. Thermoeconomic analysis of a novel combined cooling, heating and power system based on solar organic Rankine cycle and cascade refrigeration cycle. *Renewable Energy*, *164*, pp.1267-1283.
4. Alanne, K. and Saari, A., 2004. Sustainable small-scale CHP technologies for buildings: the basis for multi-perspective decision-making. *Renewable and Sustainable Energy Reviews*, *8*(5), pp.401-431.
5. Alharthi, M.A., Khaliq, A., Alqaed, S. and Almehmadi, F., 2023. Investigation of new combined cooling, heating and power system based on solar thermal power and single-double-effect refrigeration cycle. *Energy Reports*, *9*, pp.289-309.
6. Ali, A.H., Youssef, A.R., George, T. and Kamel, S., 2018, February. Optimal DG allocation in distribution systems using Ant lion optimizer. In *2018 International Conference on Innovative Trends in Computer Engineering (ITCE)* (pp. 324-331). IEEE.
7. Ali, E.S., Abd Elazim, S.M. and Abdelaziz, A.Y., 2016. Ant lion optimization algorithm for renewable distributed generations. *Energy*, *116*, pp.445-458.
8. Al-Sulaiman, F.A., Dincer, I. and Hamdullahpur, F., 2012. Energy and exergy analyses of a biomass trigeneration system using an organic Rankine cycle. *Energy*, *45*(1), pp.975-985.
9. American Society of Heating, Refrigerating and Air-Conditioning Engineers (**ASHRAE**) (**2000**). 2000 ASHRAE handbook: Heating, ventilating, and air-conditioning systems and equipment. Atlanta, GA: ASHRAE.
10. Asgari, A., Yari, M., Mahmoudi, S. M. S., & Desideri, U. 2022. Multi-objective grey wolf optimization and parametric study of a continuous solar-based tri-generation system using a phase change material storage unit. *Journal of Energy Storage*, *55*, 105783.
11. Azaza, M. and Wallin, F., 2017. Multi objective particle swarm optimization of hybrid micro-grid system: A case study in Sweden. *Energy*, *123*, pp.108-118.

12. Azizi, S., Nedaei, N., & Yari, M. (2022). Proposal and evaluation of a solar-based polygeneration system: Development, exergoeconomic analysis, and multi-objective optimization. *International Journal of Energy Research*, *46*(10), 13627-13656.
13. Bai, W. and Xu, X., 2018. Comparative analyses of two improved CO2 combinated cooling, heating, and power systems driven by solar energy. *Thermal Science*, *22*(Suppl. 2), pp.693-700.
14. Barlev, D., Vidu, R. and Stroeve, P., 2011. Innovation in concentrated solar power. *Solar energy materials and solar cells*, *95*(10), pp.2703-2725.
15. Bednarz, J.C., 1988. Cooperative hunting Harris' hawks (Parabuteo unicinctus). *Science*, *239*(4847), pp.1525-1527.
16. Behbahani-Nia, A., Bagheri, M. and Bahrampoury, R., 2010. Optimization of fire tube heat recovery steam generators for cogeneration plants through genetic algorithm. *Applied Thermal Engineering*, *30*(16), pp.2378-2385.
17. Behzadi, A., Habibollahzade, A., Arabkoohsar, A., Shabani, B., Fakhari, I., & Vojdani, M. (2021). 4E analysis of efficient waste heat recovery from SOFC using APC: An effort to reach maximum efficiency and minimum emission through an application of grey wolf optimization. *International Journal of Hydrogen Energy*, *46*(46), 23879-23897.
18. Boyaghchi, F.A. and Heidarnejad, P., 2015. Thermoeconomic assessment and multi objective optimization of a solar micro CCHP based on Organic Rankine Cycle for domestic application. *Energy conversion and Management*, *97*, pp.224-234.
19. BP Energy Outlook (2023). Available at:https://www.bp.com/en/global/corporate/energy-economics/energy-outlook.html (accessed on 31st May 2023).
20. Cao, Y., Dhahad, H. A., Togun, H., Haghghi, M. A., Athari, H., & Mohamed, A. M. 2021. Exergetic and economic assessments and multi-objective optimization of a modified solar-powered CCHP system with thermal energy storage. *Journal of Building Engineering*, *43*, 102702.
21. Chen, J., Huang, S. and Shahabi, L., 2021. Economic and environmental operation of power systems including combined cooling, heating, power and energy storage resources using developed multi-objective grey wolf algorithm. *Applied Energy*, *298*, p.117257.

22. Chen, Q., Han, W., Zheng, J.J., Sui, J. and Jin, H.G., 2014. The exergy and energy level analysis of a combined cooling, heating and power system driven by a small scale gas turbine at off design condition. *Applied Thermal Engineering*, *66*(1-2), pp.590-602.
23. Costa, E., Almeida, M. F., Alvim-Ferraz, C., & Dias, J. M. (2021). Otimization of Crambe abyssinica enzymatic transesterification using response surface methodology. *Renewable Energy*, *174*, 444-452.
24. Critoph, R.E. and Zhong, Y., 2005. Review of trends in solid sorption refrigeration and heat pumping technology. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, *219*(3), pp.285-300.
25. Darrow, K., Tidball, R., Wang, J. and Hampson, A., 2015. Catalog of CHP technologies. *US Environmental Protection Agency Combined Heat and Power Partnership*, pp.2015-07.
26. Díaz, P.R., Benito, Y.R. and Parise, J.A.R., 2010. Thermoeconomic assessment of a multi-engine, multi-heat-pump CCHP (combined cooling, heating and power generation) system–a case study. *Energy*, *35*(9), pp.3540-3550.
27. Dincer, I. and Zamfirescu, C., 2012. Renewable‐energy‐based multigeneration systems. *International Journal of Energy Research*, *36*(15), pp.1403-1415.
28. DOE, C., 2000. heat and power: a federal manager's resource guide (Final Report). *US Department of Energy, Federal Energy Management Program*.
29. Dorigo, M., Di Caro, G., & Gambardella, L. M. 1999. Ant algorithms for discrete optimization. *Artificial life*, *5*(2), 137-172.
30. Dubey, H.M., Pandit, M. and Panigrahi, B.K., 2016. Ant lion optimization for short-term wind integrated hydrothermal power generation scheduling. *International Journal of Electrical Power & Energy Systems*, *83*, pp.158-174.
31. Ebrahimi, M. and Keshavarz, A., 2012. Climate impact on the prime mover size and design of a CCHP system for the residential building. *Energy and Buildings*, *54*, pp.283-289.
32. Eshkevari M. (2023). Multi-objective Harris Hawks Optimization (MOHHO) Available at: https://github.com/miladesh1212/Multi-objective-Harris-hawks-optimization-MOHHO-, GitHub. (accessed on 9th May 2023)
33. Fanchi, J.R., 2004. *Energy technology and directions for the future*. Academic Press.

34. Ghaebi, H., Amidpour, M., Karimkashi, S. and Rezayan, O., 2011. Energy, exergy and thermoeconomic analysis of a combined cooling, heating and power (CCHP) system with gas turbine prime mover. *International Journal of Energy Research*, *35*(8), pp.697-709.

35. Goodenough, J., McGuire, B. and Jakob, E., 2009. *Perspectives on animal behavior*. John Wiley & Sons.

36. Gosney WB. 1982. Principle of refrigeration. Cambridge, United Kingdom: Cam- bridge University Press.

37. Habibollahzade, A., & Houshfar, E. (2020). Improved performance and environmental indicators of a municipal solid waste fired plant through CO2 recycling: Exergoeconomic assessment and multi-criteria grey wolf optimisation. *Energy Conversion and Management*, *225*, 113451.

38. Hadidian-Moghaddam, M.J., Arabi-Nowdeh, S., Bigdeli, M. and Azizian, D., 2018. A multi-objective optimal sizing and siting of distributed generation using ant lion optimization technique. *Ain shams engineering journal*, *9*(4), pp.2101-2109.

39. Haghghi, M. A., Mohammadi, Z., Delpisheh, M., Nadimi, E., & Athari, H. (2023). Multi-variable study/optimization of a novel geothermal-driven poly-generation system: Application of a soft-computing intelligent procedure and MOGWO. *Process Safety and Environmental Protection*, *171*, 507-531.

40. Haghghi, M.A., Pesteei, S.M., Chitsaz, A. and Hosseinpour, J., 2019. Thermodynamic investigation of a new combined cooling, heating, and power (CCHP) system driven by parabolic trough solar collectors (PTSCs): a case study. *Applied Thermal Engineering*, *163*, p.114329.

41. Harrison J. 2002. Micro combined heat and power. Technical report. EA Technology;

42. Hasanzadeh, R., Mojaver, M., Azdast, T. and Park, C.B., 2022. A novel systematic multi-objective optimization to achieve high-efficiency and low-emission waste polymeric foam gasification using response surface methodology and TOPSIS method. *Chemical Engineering Journal*, *430*, p.132958.

43. Hansen, N., Müller, S.D. and Koumoutsakos, P., 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, *11*(1), pp.1-18.

44. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., 2019. Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, *97*, pp.849-872.

45. Hernández-Santoyo, J. and Sánchez-Cifuentes, A., 2003. Trigeneration: an alternative for energy savings. *Applied Energy*, *76*(1-3), pp.219-227.

46. Holland J.H. (1975). Adaptation in natural and artificial systems. University of Michigan Press, *Ann Arbor*.

47. Huicochea, A., Rivera, W., Gutiérrez-Urueta, G., Bruno, J.C. and Coronas, A., 2011. Thermodynamic analysis of a trigeneration system consisting of a micro gas turbine and a double effect absorption chiller. *Applied Thermal Engineering*, *31*(16), pp.3347-3353.

48. IEA (2023). C02 Emissions in 2022. Available at: https://www.iea.org/reports/co2-emissions-in-2022 (accessed on 9th May 2023)

49. Islam, M.Z., Wahab, N.I.A., Veerasamy, V., Hizam, H., Mailah, N.F., Khan, A. and Sabo, A., 2019, November. Optimal power flow using a novel Harris hawk optimization algorithm to minimize fuel cost and power loss. In *2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET)* (pp. 246-250). IEEE.

50. Jafary, S., Khalilarya, S., Shawabkeh, A., Wae-hayee, M. and Hashemian, M., 2021. A complete energetic and exergetic analysis of a solar powered trigeneration system with two novel organic Rankine cycle (ORC) configurations. *Journal of Cleaner Production*, *281*, p.124552.

51. Ji, J., Wang, F., Zhou, M., Guo, R., Ji, R., Huang, H., Zhang, J., Nazir, M.S., Peng, T., Zhang, C. and Huang, J., 2022. Evaluation Study on a Novel Structure CCHP System with a New Comprehensive Index Using Improved ALO Algorithm. *Sustainability*, *14*(22), p.15419.

52. Jurgen, R.K., 1986. The promise of the Kalina cycle: Using an ammonia-water mixture, the Kalina steam cycle may permit thermal-mechanical-electrical energy conversion efficiencies of 45 percent. *IEEE Spectrum*, *23*(4), pp.68-70.

53. Kalina, A.I., 1984. Combined-cycle system with novel bottoming cycle.

54. Kalogirou, S.A., 2004. Solar thermal collectors and applications. *Progress in energy and combustion science*, *30*(3), pp.231-295.

55. Karabulut, H., Yucesu, H.S., Cinar, C. and Aksoy, F., 2009. Construction and testing of a dish/Stirling solar energy unit. *Journal of the Energy Institute*, *82*(4), pp.228-232.
56. Karellas, S. and Schuster, A., 2008. Supercritical fluid parameters in organic Rankine cycle applications. *International journal of Thermodynamics*, *11*(3), pp.101-108.
57. Kaveh, A., Dadras Eslamlou, 2020. Multi-objective Electrical Energy Scheduling in Smart Homes Using Ant Lion Optimizer and Evidential Reasoning. *Metaheuristic Optimization Algorithms in Civil Engineering: New Applications*, pp.331-374.
58. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
59. Keshavarzzadeh, A.H. and Ahmadi, P., 2019. Multi-objective techno-economic optimization of a solar based integrated energy system using various optimization methods. *Energy conversion and management*, *196*, pp.196-210.
60. Kim, I. Y., & De Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, *29*, 149-158.
61. Kitto, J.B., 2005. Steam: its generation and use. 1. print. *Barberton, Ohio: Babcock & Wilcox*.
62. Knight, I., Ugursal, I. and Beausoleil-Morrison, I., 2005. Residential cogeneration systems: a review of the current technologies [A report of Subtask A of FC+ COGEN-SIM (the simulation of building-integrated fuel cell and other cogeneration systems): Annex 42 of the International Energy Agency, Energy Conservation in Buildings and Community Systems Programme].
63. Kuhn, V., Klemeš, J. and Bulatov, I., 2008. MicroCHP: Overview of selected technologies, products and field test results. *Applied Thermal Engineering*, *28*(16), pp.2039-2048.
64. Kumar, R., 2017. A critical review on energy, exergy, exergoeconomic and economic (4-E) analysis of thermal power plants. *Engineering Science and Technology, an International Journal*, *20*(1), pp.283-292.
65. Lahoud, C., Al Asmar, J. and Brouche, M., 2018. Review of cogeneration and trigeneration systems. *African Journal of Engineering Research*, *6*(3), pp.39-54.

66. Lévy, C., 1996. Cogeneration techniques, Engineering techniques. Energy Engineering, 6, pp.8910-8911.

67. Li, H., Fu, L., Geng, K. and Jiang, Y., 2006. Energy utilization evaluation of CCHP systems. *Energy and buildings*, *38*(3), pp.253-257.

68. Li, Z., Chen, H., Xu, Y. and Ooi, K.T., 2020. Comprehensive evaluation of low-grade solar trigeneration system by photovoltaic-thermal collectors. *Energy Conversion and Management*, *215*, p.112895.

69. Liu, M., Shi, Y. and Fang, F., 2014. Combined cooling, heating and power systems: A survey. *Renewable and Sustainable Energy Reviews*, *35*, pp.1-22.

70. Lozano, M.A., Carvalho, M. and Serra, L.M., 2011. Allocation of economic costs in trigeneration systems at variable load conditions. *Energy and Buildings*, *43*(10), pp.2869-2881.

71. Macriss, R.A., Gutraj, J.M. and Zawacki, T.S., 1988. *Absorption fluids data survey: final report on worldwide data* (No. ORNL/Sub-84-47989/3). Oak Ridge National Lab., TN (USA); Institute of Gas Technology, Chicago, IL (USA).

72. Mahalekshmi, T. and Maruthupandi, P., 2023. Multiobjective Economic/Environmental Dispatch Using Harris Hawks Optimization Algorithm. *Intelligent Automation and Soft Computing*, *36*(1), pp.445-460.

73. Mahdavi, N., & Khalilarya, S. 2019. Comprehensive thermodynamic investigation of three cogeneration systems including GT-HRSG/RORC as the base system, intermediate system and solar hybridized system. *Energy*, *181*, 1252-1272.

74. Mahdavi, N., Mojaver, P. and Khalilarya, S., 2022. Multi-objective optimization of power, CO2 emission and exergy efficiency of a novel solar-assisted CCHP system using RSM and TOPSIS coupled method. *Renewable Energy*, *185*, pp.506-524.

75. Mani, M., Bozorg-Haddad, O. and Chu, X., 2018. Ant lion optimizer (ALO) algorithm. *Advanced optimization by nature-inspired algorithms*, pp.105-116.

76. Mirjalili S. (2023a). Grey Wolf Optimizer (GWO) Available at: https://www.mathworks.com/matlabcentral/fileexchange/44974-grey-wolf-optimizer-gwo , MATLAB Central File Exchange. (Accessed on 9th May 2023).

77. Mirjalili S. (2023c). Ant Lion Optimizer (ALO) Available at: https://www.mathworks.com/matlabcentral/fileexchange/49920-ant-lion-optimizer-alo , MATLAB Central File Exchange. (Accessed on 9th May 2023).

78. Mirjalili S. (2023d). Multi-objective Ant Lion Optimizer (MOALO) Available at: https://www.mathworks.com/matlabcentral/fileexchange/58460-multi-objective-ant-lion-optimizer-moalo , MATLAB Central File Exchange. (Accessed on 9th May 2023).

79. Mirjalili S.(2023b). Multi-Objective Grey Wolf Optimizer (MOGWO) Available at: https://www.mathworks.com/matlabcentral/fileexchange/55979-multi-objective-grey-wolf-optimizer-mogwo , MATLAB Central File Exchange. (Accessed on 9th May 2023).

80. Mirjalili, S., Jangir, P. and Saremi, S., 2017. Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence*, *46*, pp.79-95.

81. Mirjalili, S., Mirjalili, S. M., & Lewis, A. 2014. Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.

82. Mirjalili, S., Saremi, S., Mirjalili, S.M. and Coelho, L.D.S., 2016. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, *47*, pp.106-119.

83. Mirolli, M.D., 2007, April. Ammonia-water based thermal conversion technology: Applications in waste heat recovery for the cement industry. In *2007 IEEE Cement Industry Technical Conference Record* (pp. 234-241). IEEE.

84. MITCO, P., 2006. Mitigation of industrial CO2 emissions through the use of heat, cooling, and power networks in industrial parks.

85. Mohsenipour, M., Ebadollahi, M., Rostamzadeh, H. and Amidpour, M., 2020. Design and evaluation of a solar-based trigeneration system for a nearly zero energy greenhouse in arid region. *Journal of cleaner production*, *254*, p.119990.

86. Muro, C., Escobedo, R., Spector, L. and Coppinger, R.P., 2011. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes*, *88*(3), pp.192-197.

87. N. Hansen, S.D. Muller, P. Koumoutsakos, 2003 Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evol. Comput.* 11 (1) 1–18.

88. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Zamani, H.; Bahreininejad, A. 2022 GGWO: Gaze cues learning-based grey wolf optimizer and its applications for solving engineering problems. J. *Comput. Sci.*, 61, 101636.
89. Nasrin, R., Rahim, N.A., Fayaz, H. and Hasanuzzaman, M., 2018. Water/MWCNT nanofluid based cooling system of PVT: Experimental and numerical research. *Renewable Energy*, *121*, pp.286-300.
90. Nedaei, N., Azizi, S., & Farshi, L. G. (2022). Performance assessment and multi-objective optimization of a multi-generation system based on solar tower power: A case study in Dubai, UAE. *Process Safety and Environmental Protection*, *161*, 295-315.
91. Ogriseck, S., 2009. Integration of Kalina cycle in a combined heat and power plant, a case study. *Applied Thermal Engineering*, *29*(14-15), pp.2843-2848.
92. Oland, C.B., 2002. Guide to Low-Emission Boiler and Combustion Equipment Selection. TN, USA: Oak Ridge National Laboratory.
93. Oland, C.B., 2004. *Guide to combined heat and power systems for boiler owners and operators*. United States. Department of Energy.
94. Omar, A.I., Ali, Z.M., Al-Gabalawy, M., Abdel Aleem, S.H. and Al-Dhaifallah, M., 2020. Multi-objective environmental economic dispatch of an electricity system considering integrated natural gas units and variable renewable energy sources. *Mathematics*, *8*(7), p.1100.
95. Onovwiona, H.I. and Ugursal, V.I., 2006. Residential cogeneration systems: review of the current technology. *Renewable and sustainable energy reviews*, *10*(5), pp.389-431.
96. Ortiga, J., Bruno, J.C. and Coronas, A., 2013. Operational optimisation of a complex trigeneration system connected to a district heating and cooling network. *Applied Thermal Engineering*, *50*(2), pp.1536-1542.
97. Pandey, A.K. and Jadoun, V.K., 2023. Real-time and day-ahead risk averse multi-objective operational scheduling of virtual power plant using modified Harris Hawk's optimization. *Electric Power Systems Research*, *220*, p.109285.
98. Parikhani, T., Azariyan, H., Behrad, R., Ghaebi, H. and Jannatkhah, J., 2020. Thermodynamic and thermoeconomic analysis of a novel ammonia-water mixture combined cooling, heating, and power (CCHP) cycle. *Renewable Energy*, *145*, pp.1158-1175.

99. Peng, M.Y.P., Chen, C., Peng, X. and Marefati, M., 2020. Energy and exergy analysis of a new combined concentrating solar collector, solid oxide fuel cell, and steam turbine CCHP system. *Sustainable Energy Technologies and Assessments*, *39*, p.100713.

100. Preet, S., 2021. A review on the outlook of thermal management of photovoltaic panel using phase change material. *Energy and Climate Change*, *2*, p.100033.

101. Petchers, N., 2020. *Combined heating, cooling & power handbook: technologies & applications*. River Publishers.

102. Ren, F., Wang, J., Zhu, S. and Chen, Y., 2019. Multi-objective optimization of combined cooling, heating and power system integrated with solar and geothermal energies. *Energy conversion and management*, *197*, p.111866.

103. Ren, F., Wang, J., Zhu, S. and Chen, Y., 2019. Multi-objective optimization of combined cooling, heating and power system integrated with solar and geothermal energies. *Energy conversion and management*, *197*, p.111866.

104. Ren, F., Wei, Z. and Zhai, X., 2021. Multi-objective optimization and evaluation of hybrid CCHP systems for different building types. *Energy*, *215*, p.119096.

105. Rosato, A., Sibilio, S. and Ciampi, G., 2013. Energy, environmental and economic dynamic performance assessment of different micro-cogeneration systems in a residential application. *Applied Thermal Engineering*, *59*(1-2), pp.599-617.

106. Saini, P., Singh, J. and Sarkar, J., 2020. Thermodynamic, economic and environmental analyses of a novel solar energy driven small-scale combined cooling, heating and power system. *Energy Conversion and Management*, *226*, p.113542.

107. Scharf, I., Subach, A. and Ovadia, O., 2008. Foraging behaviour and habitat selection in pit-building antlion larvae in constant light or dark conditions. *Animal Behaviour*, *76*(6), pp.2049-2057.

108. Selim, A., Kamel, S., Alghamdi, A.S. and Jurado, F., 2020. Optimal placement of DGs in distribution system using an improved harris hawks optimizer based on single-and multi-objective approaches. *IEEE Access*, *8*, pp.52815-52829.

109. Shahverdian, M.H., Sohani, A., Sayyaadi, H., Samiezadeh, S., Doranehgard, M.H., Karimi, N. and Li, L.K., 2021. A dynamic multi-objective optimization procedure for water cooling of a photovoltaic module. *Sustainable Energy Technologies and Assessments*, *45*, p.101111.

110. Shakibi, H., Nedaei, N., Farajollahi, A. H., & Chitsaz, A. 2023. Exergoeconomic appraisement, sensitivity analysis, and multi-objective optimization of a solar-driven generation plant for yielding electricity and cooling load. *Process Safety and Environmental Protection*, *170*, 89-111.

111. Sharifian, Y. and Abdi, H., 2022. Solving multi-zone combined heat and power economic emission dispatch problem considering wind uncertainty by applying grasshopper optimization algorithm. *Sustainable Energy Technologies and Assessments*, *53*, p.102512.

112. Sharma, M.K. and Bhattacharya, J., 2022. Finding optimal operating point for advection-cooled concentrated photovoltaic system. *Sustainable Energy Technologies and Assessments*, *49*, p.101769.

113. Shehab, M., Mashal, I., Momani, Z., Shambour, M.K.Y., AL-Badareen, A., Al-Dabet, S., Bataina, N., Alsoud, A.R. and Abualigah, L., 2022. Harris hawks optimization algorithm: variants and applications. *Archives of Computational Methods in Engineering*, *29*(7), pp.5579-5603.

114. Song, Y., Tan, X. and Mizzi, S., 2020. Optimal parameter extraction of the proton exchange membrane fuel cells based on a new Harris Hawks Optimization algorithm. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, pp.1-18.

115. Song, Z., Liu, T., & Lin, Q. (2020). Multi-objective optimization of a solar hybrid CCHP system based on different operation modes. *Energy*, *206*, 118125.

116. Song, Z., Liu, T., Liu, Y., Jiang, X. and Lin, Q., 2020. Study on the optimization and sensitivity analysis of CCHP systems for industrial park facilities. *International Journal of Electrical Power & Energy Systems*, *120*, p.105984.

117. Spahni, R.; Chappellaz, J.; Stocker, T.F.; Loulergue, L.; Hausammann, G.; Kawamura, K.; Fluckiger, J.; Schwander, J.; Raynaud, D.; Masson-Delmotte, V.; et al. 2005 Atmospheric methane and nitrous oxide of the late Pleistocene from Antarctic ice cores. *Science*, *310*, 1317–1321.

118. Tung, N.S. and Chakravorty, S., 2016. Ant lion optimizer based approach for optimal scheduling of thermal units for small scale electrical economic power dispatch problem. *International Journal of Grid and Distributed Computing*, *9*(7), pp.211-224.

119. Wang, C. and Nehrir, M.H., 2006, March. Distributed generation applications of fuel cells. In *2006 Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources* (pp. 244-248). IEEE.

120. Wang, J., Dong, F., Ma, Z., Chen, H. and Yan, R., 2021. Multi-objective optimization with thermodynamic analysis of an integrated energy system based on biomass and solar energies. *Journal of Cleaner Production*, *324*, p.129257.

121. Wang, J., Han, Z. and Guan, Z., 2020. Hybrid solar-assisted combined cooling, heating, and power systems: A review. *Renewable and Sustainable Energy Reviews*, *133*, p.110256.

122. Wang, J., Han, Z., Liu, Y., Zhang, X. and Cui, Z., 2021. Thermodynamic analysis of a combined cooling, heating, and power system integrated with full-spectrum hybrid solar energy device. *Energy Conversion and Management*, *228*, p.113596.

123. Wang, J., Liu, Y., Ren, F., & Lu, S. 2020. Multi-objective optimization and selection of hybrid combined cooling, heating and power systems considering operational flexibility. *Energy*, *197*, 117313.

124. Wang, J., Lu, Y., Yang, Y., & Mao, T. 2016. Thermodynamic performance analysis and optimization of a solar-assisted combined cooling, heating and power system. *Energy*, *115*, 49-59.

125. Wang, L., Lu, J., Wang, W. and Ding, J., 2016. Energy, environmental and economic evaluation of the CCHP systems for a remote island in south of China. *Applied Energy*, *183*, pp.874-883.

126. Wang, N.; Wang, D.; Xing, Y.; Shao, L.; Afzal, S. 2020 Application of co-evolution RNA genetic algorithm for obtaining optimal parameters of SOFC model. *Renew. Energy*, *150*, 221–233. https://doi.org/10.1016/j.renene.2019.12.105.

127. Wei, L. and Xv, S., 2022. Optimal scheduling of building integrated energy systems using an improved ant lion algorithm. *International Journal of Low-Carbon Technologies*, *17*, pp.720-729.

128. Wu, D. and Wang, R., 2006. Combined cooling, heating and power: A review. *progress in energy and combustion science*, *32*(5-6), pp.459-495.

129. Wu, D.; Wang, R. 2006 Combined cooling, heating and power: A review. *Prog. Energy Combust. Sci.*, *32*, 459–495.

130. Xu, L., Luo, X., Wen, Y., Wu, T., Wang, X. and Guan, X., 2023. Energy Management of Hybrid Power Ship System Using Adaptive Moth Flame Optimization Based on Multi-Populations. *IEEE Transactions on Power Systems*.

131. Yan, R., Wang, J., Cheng, Y., Ma, C. and Yu, T., 2020. Thermodynamic analysis of fuel cell combined cooling heating and power system integrated with solar reforming of natural gas. *Solar Energy*, *206*, pp.396-412.

132. Yong, L. and Wang, R.Z., 2007. Adsorption refrigeration: a survey of novel technologies. *Recent Patents on Engineering*, *1*(1), pp.1-21.

133. Yousefi, H.; Ghodusinejad, M.H.; Kasaeian, 2017. A. Multi-objective optimal component sizing of a hybrid ICE + PV/T driven CCHP microgrid. *Appl. Therm. Eng.*, *122*, 126–138. https://doi.org/10.1016/j.applthermaleng.2017.05.017.

134. Yousri, D., Babu, T.S. and Fathy, A., 2020. Recent methodology based Harris Hawks optimizer for designing load frequency control incorporated in multi-interconnected renewable energy plants. *Sustainable Energy, Grids and Networks*, *22*, p.100352.

135. Zarei, A., Akhavan, S., Ghodrat, M. and Behnia, M., 2022. Thermodynamic analysis and multi-objective optimization of a modified solar trigeneration system for cooling, heating and power using photovoltaic-thermal and flat plate collectors. *International Communications in Heat and Mass Transfer*, *137*, p.106261.

136. Zeng, R., Li, H., Liu, L., Zhang, X. and Zhang, G., 2015. A novel method based on multi-population genetic algorithm for CCHP–GSHP coupling system optimization. *Energy Conversion and Management*, *105*, pp.1138-1148.

137. Zhang, G., Li, H., Xiao, C., & Sobhani, B. (2022). Multi-aspect analysis and multi-objective optimization of a novel biomass-driven heat and power cogeneration system; utilization of grey wolf optimizer. *Journal of Cleaner Production, 355*, 131442.

138. Zhang, L., & Sobhani, B. (2022). Comprehensive economic analysis and multi-objective optimization of an integrated power and freshwater generation cycle based on flash-binary geothermal and gas turbine cycles. *Journal of Cleaner Production*, *364*, 132644.

139. Zhang, N., Wang, Z., Lior, N. and Han, W., 2018. Advancement of distributed energy methods by a novel high efficiency solar-assisted combined cooling, heating and power system. *Applied Energy*, *219*, pp.179-186.

140. Zhang, X., He, M. and Zhang, Y., 2012. A review of research on the Kalina cycle. *Renewable and sustainable energy reviews*, *16*(7), pp.5309-5318.

141. Zhao, L., Li, Z., Chen, H., Li, J., Xiao, J. and Yousefi, N., 2020. A multi-criteria optimization for a CCHP with the fuel cell as primary mover using modified Harris Hawks optimization. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, pp.1-16.

APPENDICES

APPENDIX A: MATLAB codes for single-objective greywolf optimization

APPENDIX A.1: Initialization and problem definition

```
%Initialization.m – Ensuring the initialized values do not exceed the upper and lower
boundaries of the decision variables.


% This function initialize the first population of search agents
function Positions=initialization(SearchAgents_no,dim,ub,lb)
Boundary_no= size(ub,2); % numnber of boundaries
% If the boundaries of all variables are equal and user enter a signle
% number for both ub and lb
if Boundary_no==1
    Positions=rand(SearchAgents_no,dim).*(ub-lb)+lb;
end
% If each variable has a different lb and ub
if Boundary_no>1
    for i=1:dim
        ub_i=ub(i);
        lb_i=lb(i);
        Positions(:,i)=rand(SearchAgents_no,1).*(ub_i-lb_i)+lb_i;
    end
end


% Get_functions_details.m - In the Get_functions_details.m file, the lower and upper
%boundary values of the constraints were defined. Also, the objective function to be
%maximized or minimized was  correctly stated.

function [lb,ub,dim,fobj] = Get_Functions_details(F)
switch F
    case 'F1'
        fobj = @F1;
        lb=[10 10 1420 850];
        ub=[15 30 1520 950];
        dim=4;
end
end
% F1
function o = F1(x)
%o=62.19 + 0.4573.*x(1) + 0.0259 .*x(2) - 0.02421 .*x(3) + 0.03638 .*x(4) - 0.010867
.*x(1).^2 ...
%- 0.000029 .*x(2).^2 + 0.000005 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0003 .*x(1).*x(2)
+ 0.00022 .*x(1).*x(3) ...
%- 0.00042 .*x(1).*x(4) - 0.000005 .*x(2).*x(3) - 0.000005 .*x(2).*x(4) - 0.000003
.*x(3).*x(4);

%o= 13.1 + 3.722.*x(1) + 0.2003 .*x(2) - 0.0122 .*x(3) + 0.0451 .*x(4) - 0.03047
.*x(1).^2 ...
%+ 0.000296 .*x(2).^2 + 0.00004 .*x(3).^2 + 0.000052 .*x(4).^2 + 0.0049 .*x(1).*x(2)
- 0.002940 .*x(1).*x(3) ...
```

```
%+ 0.00286 .*x(1).*x(4) - 0.000285 .*x(2).*x(3) + 0.000285 .*x(2).*x(4) - 0.000099
.*x(3).*x(4);

o= -29.0 - 0.36.*x(1) + 0.287 .*x(2) + 0.0659 .*x(3) + 0.0133 .*x(4) - 0.01807
.*x(1).^2 ...
- 0.000029 .*x(2).^2 - 0.000011 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0125 .*x(1).*x(2)
+ 0.0003 .*x(1).*x(3) ...
+ 0.00086 .*x(1).*x(4) - 0.000205 .*x(2).*x(3) + 0.000195 .*x(2).*x(4) - 0.000009
.*x(3).*x(4);
%o=1.27686-0.000897964*x(1)+0.0008937*x(2)+0.036303*x(3)+0.000000169203*(x(1)^2);
end
```

APPENDIX A.2: Evaluation

```
%GWO.m file -  All equations (3.19 to 3.27) were defined here to obtain the values of
%the objective function and also the search agent/best three solutions as well as
%carry out updates per iteration.GWO.m file -  All equations (3.19 to 3.27) were
%defined here to obtain the values of the objective function and also the search
%agent/best three solutions as well as carry out updates per iteration.% Grey Wolf
Optimizer
function
[Alpha_score,Alpha_pos,Convergence_curve]=GWO(SearchAgents_no,Max_iter,lb,ub,dim,fobj
)
% initialize alpha, beta, and delta_pos
Alpha_pos=zeros(1,dim);
Alpha_score=inf; %change this to -inf for maximization problems
Beta_pos=zeros(1,dim);
Beta_score=inf; %change this to -inf for maximization problems
Delta_pos=zeros(1,dim);
Delta_score=inf; %change this to -inf for maximization problems
%Initialize the positions of search agents
Positions=initialization(SearchAgents_no,dim,ub,lb);
Convergence_curve=zeros(1,Max_iter);
l=0;% Loop counter
% Main loop
while l<Max_iter
    for i=1:size(Positions,1)

       % Return back the search agents that go beyond the boundaries of the search
space
        Flag4ub=Positions(i,:)>ub;
        Flag4lb=Positions(i,:)<lb;

Positions(i,:)=(Positions(i,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag4lb;

        % Calculate objective function for each search agent
        fitness=fobj(Positions(i,:));

        % Update Alpha, Beta, and Delta
        if fitness<Alpha_score
            Alpha_score=fitness; % Update alpha
            Alpha_pos=Positions(i,:);
        end
        if fitness>Alpha_score && fitness<Beta_score
            Beta_score=fitness; % Update beta
            Beta_pos=Positions(i,:);
        end

        if fitness>Alpha_score && fitness>Beta_score && fitness<Delta_score
            Delta_score=fitness; % Update delta
            Delta_pos=Positions(i,:);
        end
    end


    a=2-l*((2)/Max_iter); % a decreases linearly fron 2 to 0

    % Update the Position of search agents including omegas
    for i=1:size(Positions,1)
        for j=1:size(Positions,2)

            r1=rand(); % r1 is a random number in [0,1]
            r2=rand(); % r2 is a random number in [0,1]

            A1=2*a*r1-a; % Equation (3.19)
            C1=2*r2; % Equation (3.20)

            D_alpha=abs(C1*Alpha_pos(j)-Positions(i,j)); % Equation (3.21)-
            X1=Alpha_pos(j)-A1*D_alpha; % Equation (3.24)

            r1=rand();
            r2=rand();

            A2=2*a*r1-a; % Equation (3.19)
            C2=2*r2; % Equation (3.20)

            D_beta=abs(C2*Beta_pos(j)-Positions(i,j)); % Equation (3.22)
            X2=Beta_pos(j)-A2*D_beta; % Equation (3.25)

            r1=rand();
            r2=rand();

            A3=2*a*r1-a; % Equation (3.19)
            C3=2*r2; % Equation (3.20)

            D_delta=abs(C3*Delta_pos(j)-Positions(i,j)); % Equation (3.23)
            X3=Delta_pos(j)-A3*D_delta; % Equation (3.26)

            Positions(i,j)=(X1+X2+X3)/3;% Equation (3.27)

        end
    end
    l=l+1;
    Convergence_curve(l)=Alpha_score;
end
```

```
% Main. M file – This houses all other functions used for this mathematical model. In
the main.m file.
% the number of search agents and number of iterations were defined.
%Thereafter, this file was made to run after adding the folder to the path.clear all
clc
SearchAgents_no=100; % Number of search agents
Function_name='F1'; % Name of the test function that can be from F1 to F23 (Table
1,2,3 in the paper)
Max_iteration=100; % Maximum numbef of iterations
% Load details of the selected benchmark function
[lb,ub,dim,fobj]=Get_Functions_details(Function_name);
[Best_score,Best_pos,GWO_cg_curve]=GWO(SearchAgents_no,Max_iteration,lb,ub,dim,fobj);
figure('Position',[500 500 660 290])
%Draw search space
subplot(1,2,1);
func_plot(Function_name);
title('Parameter space')
xlabel('x_1');
ylabel('x_2');
zlabel([Function_name,'( x_1 , x_2 , x_3, x_4 )'])
%Draw objective space
subplot(1,2,2);
semilogy(GWO_cg_curve,'Color','r')
title('Objective space')
xlabel('Iteration');
ylabel('Best score obtained so far');
axis tight
grid on
box on
legend('GWO')
display(['The best solution obtained by GWO is : ', num2str(Best_pos)]);
display(['The best optimal value of the objective funciton found by GWO is : ',
num2str(Best_score)]);
```

APPENDIX B: MATLAB codes for multi-objective greywolf optimization

APPENDIX B.1: Problem definition

```
% cec09.m - In the cec09.m file, the objective functions to be optimized are
correctly stated.
% The Matlab version of the test instances for CEC 2009 Multiobjective
%   Optimization Competition.
%
% Usage: fobj = cec09(problem_name), the handle of the function will be
%   with fobj
%
% Please refer to the report for correct one if the source codes are not
%   consist with the report.
% History:
%   v1 Sept.08 2008
%   v2 Nov.18  2008
%   v3 Nov.26  2008
function fobj = cec09(name)
    switch name
        case 'UF1'
            fobj = @UF1;
        case 'UF2'
            fobj = @UF2;
        case 'UF3'
            fobj = @UF3;
        case 'UF4'
            fobj = @UF4;
    end
end
function y = UF1(x)
y(1)=62.19 + 0.4573.*x(1) + 0.0259 .*x(2) - 0.02421 .*x(3) + 0.03638 .*x(4) -
0.010867 .*x(1).^2 ...
     - 0.000029 .*x(2).^2 + 0.000005 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0003
.*x(1).*x(2) + 0.00022 .*x(1).*x(3) ...
     - 0.00042 .*x(1).*x(4) - 0.000005 .*x(2).*x(3) - 0.000005 .*x(2).*x(4) -
0.000003 .*x(3).*x(4);

     y(2)= 13.1 + 3.722.*x(1) + 0.2003 .*x(2) - 0.0122 .*x(3) + 0.0451 .*x(4) -
0.03047 .*x(1).^2 ...
     + 0.000296 .*x(2).^2 + 0.00004 .*x(3).^2 + 0.000052 .*x(4).^2 + 0.0049
.*x(1).*x(2) - 0.002940 .*x(1).*x(3) ...
     + 0.00286 .*x(1).*x(4) - 0.000285 .*x(2).*x(3) + 0.000285 .*x(2).*x(4) -
0.000099 .*x(3).*x(4);

     y(3) = -29.0 - 0.36.*x(1) + 0.287 .*x(2) + 0.0659 .*x(3) + 0.0133 .*x(4) -
0.01807 .*x(1).^2 ...
     - 0.000029 .*x(2).^2 - 0.000011 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0125
.*x(1).*x(2) + 0.0003 .*x(1).*x(3) ...
     + 0.00086 .*x(1).*x(4) - 0.000205 .*x(2).*x(3) + 0.000195 .*x(2).*x(4) -
0.000009 .*x(3).*x(4);
end
```

APPENDIX B.2: Initialization, Evaluation, Selection and Iteration Improvement

```
%MOGWO.m - Here the initial population, archive number, number of iteration
%and other parameters are initialized. Also, the evaluation process and plotting of
the pareto graph takes
%place.
%The mogwo.m file is made to run after adding the folder to the path. All files must
be saved in the same folder.

% Start the timer
tic;

drawing_flag = 1;
TestProblem='UF1';
nVar=4;
fobj = cec09(TestProblem);
xrange = xboundary(TestProblem, nVar);
% Lower bound and upper bound
%lb=xrange(:,1)';
%ub=xrange(:,2)';
%lb=10*ones(1,nVar);
%ub=1520*ones(1,nVar);
Alb = [10];
Aub = [15];
Blb = [10];
Bub = [30];
Clb = [1520];
Cub = [1520];
Dlb = [850];
Dub = [950];

lb=[Alb Blb Clb Dlb];
ub=[Aub Bub Cub Dub];
VarSize=[1 nVar];
GreyWolves_num=100;
MaxIt=100;  % Maximum Number of Iterations
Archive_size=100;   % Repository Size
%alpha=0.1;  % Grid Inflation Parameter
alpha=0.1;
nGrid=4;   % Number of Grids per each Dimension
beta=4; %=4;    % Leader Selection Pressure Parameter
gamma=2;
% Initialization
GreyWolves=CreateEmptyParticle(GreyWolves_num);
for i=1:GreyWolves_num
    GreyWolves(i).Velocity=0;
    GreyWolves(i).Position=zeros(1,nVar);
    for j=1:nVar
        GreyWolves(i).Position(1,j)=unifrnd(lb(j),ub(j),1);
    end
    GreyWolves(i).Cost=fobj(GreyWolves(i).Position')';
    GreyWolves(i).Best.Position=GreyWolves(i).Position;
    GreyWolves(i).Best.Cost=GreyWolves(i).Cost;
end
GreyWolves=DetermineDomination(GreyWolves);
Archive=GetNonDominatedParticles(GreyWolves);
```

```
Archive_costs=GetCosts(Archive);
G=CreateHypercubes(Archive_costs,nGrid,alpha);
for i=1:numel(Archive)
    [Archive(i).GridIndex Archive(i).GridSubIndex]=GetGridIndex(Archive(i),G);
end
% MOGWO main loop
for it=1:MaxIt
    a=2-it*((2)/MaxIt);
    for i=1:GreyWolves_num

        clear rep2
        clear rep3

        % Choose the alpha, beta, and delta grey wolves
        Delta=SelectLeader(Archive,beta);
        Beta=SelectLeader(Archive,beta);
        Alpha=SelectLeader(Archive,beta);

        % If there are less than three solutions in the least crowded
        % hypercube, the second least crowded hypercube is also found
        % to choose other leaders from.
        if size(Archive,1)>1
            counter=0;
            for newi=1:size(Archive,1)
                if sum(Delta.Position~=Archive(newi).Position)~=0
                    counter=counter+1;
                    rep2(counter,1)=Archive(newi);
                end
            end
            Beta=SelectLeader(rep2,beta);
        end

        % This scenario is the same if the second least crowded hypercube
        % has one solution, so the delta leader should be chosen from the
        % third least crowded hypercube.
        if size(Archive,1)>2
            counter=0;
            for newi=1:size(rep2,1)
                if sum(Beta.Position~=rep2(newi).Position)~=0
                    counter=counter+1;
                    rep3(counter,1)=rep2(newi);
                end
            end
            Alpha=SelectLeader(rep3,beta);
        end

        % Eq.(3.20) in the paper
        c=2.*rand(1, nVar);
        % Eq.(3.21) in the paper
        D=abs(c.*Delta.Position-GreyWolves(i).Position);
        % Eq.(3.19) in the paper
        A=2.*a.*rand(1, nVar)-a;
        % Eq.(3.24) in the paper
        X1=Delta.Position-A.*abs(D);
        % Eq.(3.20) in the paper
        c=2.*rand(1, nVar);
        % Eq.(3.22) in the paper
        D=abs(c.*Beta.Position-GreyWolves(i).Position);
        % Eq.(3.19) in the paper
        A=2.*a.*rand()-a;
        % Eq.(3.25) in the paper
        X2=Beta.Position-A.*abs(D);


        % Eq.(3.20) in the paper
        c=2.*rand(1, nVar);
        % Eq.(3.23) in the paper
        D=abs(c.*Alpha.Position-GreyWolves(i).Position);
        % Eq.(3.19) in the paper
        A=2.*a.*rand()-a;
        % Eq.(3.26) in the paper
        X3=Alpha.Position-A.*abs(D);

        % Eq.(3.27) in the paper
        GreyWolves(i).Position=(X1+X2+X3)./3;

        % Boundary checking
        GreyWolves(i).Position=min(max(GreyWolves(i).Position,lb),ub);

        GreyWolves(i).Cost=fobj(GreyWolves(i).Position')';
    end

    GreyWolves=DetermineDomination(GreyWolves);
    non_dominated_wolves=GetNonDominatedParticles(GreyWolves);

    Archive=[Archive
        non_dominated_wolves];

    Archive=DetermineDomination(Archive);
    Archive=GetNonDominatedParticles(Archive);

    for i=1:numel(Archive)
        [Archive(i).GridIndex Archive(i).GridSubIndex]=GetGridIndex(Archive(i),G);
    end

    if numel(Archive)>Archive_size
        EXTRA=numel(Archive)-Archive_size;
        Archive=DeleteFromRep(Archive,EXTRA,gamma);

        Archive_costs=GetCosts(Archive);
        G=CreateHypercubes(Archive_costs,nGrid,alpha);

    end

    disp(['In iteration ' num2str(it) ': Number of solutions in the archive = '
num2str(numel(Archive))]);
    save results
    % Results

    costs=GetCosts(GreyWolves);
    Archive_costs=GetCosts(Archive);

    if drawing_flag==1
        hold off

        plot(costs(1,:),costs(3,:),'k.');
        hold on
        plot(Archive_costs(1,:),Archive_costs(3,:),'bd');
        legend('Grey wolves','Non-dominated solutions');
        title('Pareto Front for Exergy Efficiency vs Net Power')
        xlabel('Net Power (MW)');
        ylabel('Exergy Efficiency (%)');
        grid on


       drawnow
    end

end

% Stop the timer
elapsedTime = toc;

% Display the elapsed time
disp(['Elapsed time: ', num2str(elapsedTime), ' seconds']);


function pop=DetermineDomination(pop)
    npop=numel(pop);

    for i=1:npop
        pop(i).Dominated=false;
        for j=1:i-1
            if ~pop(j).Dominated
                if Dominates(pop(i),pop(j))
                    pop(j).Dominated=true;
                elseif Dominates(pop(j),pop(i))
                    pop(i).Dominated=true;
                    break;
                end
            end
        end
    end
end


function dom=Dominates(x,y)
    if isstruct(x)
        x=x.Cost;
    end
    if isstruct(y)
        y=y.Cost;
    end

    dom=all(x<=y) && any(x<y);
end


function nd_pop=GetNonDominatedParticles(pop)
    ND=~[pop.Dominated];

    nd_pop=pop(ND);
end


function G=CreateHypercubes(costs,ngrid,alpha)
    nobj=size(costs,1);

    empty_grid.Lower=[];
    empty_grid.Upper=[];
    G=repmat(empty_grid,nobj,1);

    for j=1:nobj

        min_cj=min(costs(j,:));
        max_cj=max(costs(j,:));

        dcj=alpha*(max_cj-min_cj);

        min_cj=min_cj-dcj;
        max_cj=max_cj+dcj;

        gx=linspace(min_cj,max_cj,ngrid-1);

        G(j).Lower=[-inf gx];
        G(j).Upper=[gx inf];

    end


function i=RouletteWheelSelection(p)
    r=rand;
    c=cumsum(p);
    i=find(r<=c,1,'first');
end


function rep_h=SelectLeader(rep,beta)
    if nargin<2
        beta=1;
    end
    [occ_cell_index occ_cell_member_count]=GetOccupiedCells(rep);

    p=occ_cell_member_count.^(-beta);
    p=p/sum(p);

    selected_cell_index=occ_cell_index(RouletteWheelSelection(p));
    GridIndices=[rep.GridIndex];

    selected_cell_members=find(GridIndices==selected_cell_index);

    n=numel(selected_cell_members);

    selected_memebr_index=randi([1 n]);

    h=selected_cell_members(selected_memebr_index);

    rep_h=rep(h);
end


function costs=GetCosts(pop)
    nobj=numel(pop(1).Cost);
    costs=reshape([pop.Cost],nobj,[]);
end


function [Index SubIndex]=GetGridIndex(particle,G)
    c=particle.Cost;

    nobj=numel(c);
    ngrid=numel(G(1).Upper);

    str=['sub2ind(' mat2str(ones(1,nobj)*ngrid)];
    SubIndex=zeros(1,nobj);
    for j=1:nobj

        U=G(j).Upper;

        i=find(c(j)<U,1,'first');

        SubIndex(j)= i;

        str=[str ',' num2str(i)];
    end

    str=[str ');'];

    Index=eval(str);

end


function [occ_cell_index occ_cell_member_count]=GetOccupiedCells(pop)
    GridIndices=[pop.GridIndex];

    occ_cell_index=unique(GridIndices);

    occ_cell_member_count=zeros(size(occ_cell_index));
    m=numel(occ_cell_index);
    for k=1:m
```

```
        occ_cell_member_count(k)=sum(GridIndices==occ_cell_index(k));
    end

end




function rep=DeleteFromRep(rep,EXTRA,gamma)
    if nargin<3
        gamma=1;
    end
    for k=1:EXTRA
        [occ_cell_index occ_cell_member_count]=GetOccupiedCells(rep);
        p=occ_cell_member_count.^gamma;
        p=p/sum(p);
        selected_cell_index=occ_cell_index(RouletteWheelSelection(p));
        GridIndices=[rep.GridIndex];
        selected_cell_members=find(GridIndices==selected_cell_index);
        n=numel(selected_cell_members);
        selected_memebr_index=randi([1 n]);
        j=selected_cell_members(selected_memebr_index);

        rep=[rep(1:j-1); rep(j+1:end)];
    end

end


function particle=CreateEmptyParticle(n)

    if nargin<1
        n=1;
    end
    empty_particle.Position=[];
    empty_particle.Velocity=[];
    empty_particle.Cost=[];
    empty_particle.Dominated=false;
    empty_particle.Best.Position=[];
    empty_particle.Best.Cost=[];
    empty_particle.GridIndex=[];
    empty_particle.GridSubIndex=[];

    particle=repmat(empty_particle,n,1);

end
```

APPENDIX C: MATLAB codes for single-objective harris hawks optimization

APPENDIX C.1:Problem definition and Initialization

```
% Get_functions_details.m - In the Get_functions_details.m file, the lower and upper
%boundary values of the constraints were defined. Also, the objective function to be
%maximized or minimized was  correctly stated.


function [lb,ub,dim,fobj] = Get_Functions_details(F)
switch F
    case 'F1'
        fobj = @F1;
        lb=[10 10 1420 850];
        ub=[15 30 1520 950];
        dim=4;

    case 'F2'
        fobj = @F2;
        lb=-10;
        ub=10;
        dim=30;

    case 'F3'
        fobj = @F3;
        lb=-100;
        ub=100;
        dim=30;

end
end
% F1
function o = F1(x)
% o=62.19 + 0.4573.*x(1) + 0.0259 .*x(2) - 0.02421 .*x(3) + 0.03638 .*x(4) - 0.010867
.*x(1).^2 ...
%     - 0.000029 .*x(2).^2 + 0.000005 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0003
.*x(1).*x(2) + 0.00022 .*x(1).*x(3) ...
%     - 0.00042 .*x(1).*x(4) - 0.000005 .*x(2).*x(3) - 0.000005 .*x(2).*x(4) -
0.000003 .*x(3).*x(4);

o= 13.1 + 3.722.*x(1) + 0.2003 .*x(2) - 0.0122 .*x(3) + 0.0451 .*x(4) - 0.03047
.*x(1).^2 ...
    + 0.000296 .*x(2).^2 + 0.00004 .*x(3).^2 + 0.000052 .*x(4).^2 + 0.0049
.*x(1).*x(2) - 0.002940 .*x(1).*x(3) ...
    + 0.00286 .*x(1).*x(4) - 0.000285 .*x(2).*x(3) + 0.000285 .*x(2).*x(4) - 0.000099
.*x(3).*x(4);

% o = -29.0 - 0.36.*x(1) + 0.287 .*x(2) + 0.0659 .*x(3) + 0.0133 .*x(4) - 0.01807
.*x(1).^2 ...
%     - 0.000029 .*x(2).^2 - 0.000011 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0125
.*x(1).*x(2) + 0.0003 .*x(1).*x(3)  ...
%     + 0.00086 .*x(1).*x(4) - 0.000205 .*x(2).*x(3) + 0.000195 .*x(2).*x(4) -
0.000009 .*x(3).*x(4);

end
```

```
%Initialization.m – Ensuring the initialized values do not exceed the upper and lower
boundaries of the decision variables.

function [X]=initialization(N,dim,up,down)
if size(up,1)==1
    X=rand(N,dim).*(up-down)+down;
end
if size(up,1)>1
    for i=1:dim
        high=up(i);low=down(i);
        X(:,i)=rand(1,N).*(high-low)+low;
    end
end
end
```

APPENDIX C.2:Evaluation

```
%HHO.m file - Equations 3.29 and 3.30 were used to obtain the Prey’s energy. Equation
3.31 was used for exploration while equations 3.33 to 3.42 were used for
exploitation. These were used to obtain the optimal decision variable and objective
value.


function [Rabbit_Energy,Rabbit_Location,CNVG]=HHO(N,T,lb,ub,dim,fobj)
disp('HHO is now tackling your problem')
tic
% initialize the location and Energy of the rabbit
Rabbit_Location=zeros(1,dim);
Rabbit_Energy=inf;
%Initialize the locations of Harris' hawks
X=initialization(N,dim,ub,lb);
CNVG=zeros(1,T);
t=0; % Loop counter
while t<T
    for i=1:size(X,1)
        % Check boundries
        FU=X(i,:)>ub;FL=X(i,:)<lb;X(i,:)=(X(i,:).*(~(FU+FL)))+ub.*FU+lb.*FL;
        % fitness of locations
        fitness=fobj(X(i,:));
        % Update the location of Rabbit
        if fitness<Rabbit_Energy
            Rabbit_Energy=fitness;
            Rabbit_Location=X(i,:);
        end
    end

    E1=2*(1-(t/T)); % factor to show the decreaing energy of rabbit (Equation 3.29)
    % Update the location of Harris' hawks
    for i=1:size(X,1)
        E0=2*rand()-1; %-1<E0<1 (Equation 3.30)
        Escaping_Energy=E1*(E0);  % escaping energy of rabbit
        if abs(Escaping_Energy)>=1
            %% Exploration:
            % Harris' hawks perch randomly based on 2 strategy:
            %Equation 3.27
            q=rand();
            rand_Hawk_index = floor(N*rand()+1);
            X_rand = X(rand_Hawk_index, :);
            if q<0.5
                % perch based on other family members
                X(i,:)=X_rand-rand()*abs(X_rand-2*rand()*X(i,:));
            elseif q>=0.5
                % perch on a random tall tree (random site inside group's home range)
                X(i,:)=(Rabbit_Location(1,:)-mean(X))-rand()*((ub-lb)*rand+lb);
            end

        elseif abs(Escaping_Energy)<1
            %% Exploitation:(Equation 3.33 to 3.42)
            % Attacking the rabbit using 4 strategies regarding the behavior of the
rabbit

            %% phase 1: surprise pounce (seven kills)
            % surprise pounce (seven kills): multiple, short rapid dives by different
hawks

            r=rand(); % probablity of each event

            if r>=0.5 && abs(Escaping_Energy)<0.5 % Hard besiege
                X(i,:)=(Rabbit_Location)-Escaping_Energy*abs(Rabbit_Location-X(i,:));
            end

            if r>=0.5 && abs(Escaping_Energy)>=0.5  % Soft besiege
                Jump_strength=2*(1-rand()); % random jump strength of the rabbit
                X(i,:)=(Rabbit_Location-X(i,:))-
Escaping_Energy*abs(Jump_strength*Rabbit_Location-X(i,:));
            end

            %% phase 2: performing team rapid dives (leapfrog movements)
            if r<0.5 && abs(Escaping_Energy)>=0.5, % Soft besiege % rabbit try to
escape by many zigzag deceptive motions

                Jump_strength=2*(1-rand());
                X1=Rabbit_Location-Escaping_Energy*abs(Jump_strength*Rabbit_Location-
X(i,:));

                if fobj(X1)<fobj(X(i,:)) % improved move?
                    X(i,:)=X1;
                else % hawks perform levy-based short rapid dives around the rabbit
                    X2=Rabbit_Location-
Escaping_Energy*abs(Jump_strength*Rabbit_Location-X(i,:))+rand(1,dim).*Levy(dim);
                    if (fobj(X2)<fobj(X(i,:))), % improved move?
                        X(i,:)=X2;
                    end
                end
            end
            if r<0.5 && abs(Escaping_Energy)<0.5, % Hard besiege % rabbit try to
escape by many zigzag deceptive motions
                % hawks try to decrease their average location with the rabbit
                Jump_strength=2*(1-rand());
                X1=Rabbit_Location-Escaping_Energy*abs(Jump_strength*Rabbit_Location-
mean(X));

                if fobj(X1)<fobj(X(i,:)) % improved move?
                    X(i,:)=X1;
                else % Perform levy-based short rapid dives around the rabbit
                    X2=Rabbit_Location-
Escaping_Energy*abs(Jump_strength*Rabbit_Location-mean(X))+rand(1,dim).*Levy(dim);
                    if (fobj(X2)<fobj(X(i,:))), % improved move?
                        X(i,:)=X2;
                    end
                end
            end
            %%
        end
    end
    t=t+1;
    CNVG(t)=Rabbit_Energy;
%    Print the progress every 100 iterations
%    if mod(t,100)==0
%        display(['At iteration ', num2str(t), ' the best fitness is ',
num2str(Rabbit_Energy)]);
%    end
end
toc
end
% ___________________________________
function o=Levy(d)
beta=1.5;
sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^((beta-
1)/2)))^(1/beta);
u=randn(1,d)*sigma;v=randn(1,d);step=u./abs(v).^(1/beta);
o=step;
end


% Main.m file - This houses all other functions used for this mathematical model. In
the main.m file,the number of search agents and number of iterations were defined.
%Thereafter, this file was made to run after adding the folder to the path.

clear all %#ok<CLALL>
close all
clc
N=100; % Number of search agents
Function_name='F1'; % Name of the test function
T=500; % Maximum number of iterations
% Load details of the selected benchmark function
[lb,ub,dim,fobj]=Get_Functions_details(Function_name);
[Rabbit_Energy,Rabbit_Location,CNVG]=HHO(N,T,lb,ub,dim,fobj);
%Draw objective space
figure,
hold on
semilogy(CNVG,'Color','b','LineWidth',4);
title('Convergence curve')
xlabel('Iteration');
ylabel('Best fitness obtained so far');
axis tight
grid off
box on
legend('HHO')
display(['The best location of HHO is: ', num2str(Rabbit_Location)]);
display(['The best fitness of HHO is: ', num2str(Rabbit_Energy)]);
```

APPENDIX D: MATLAB codes for multi-objective harris hawks optimization

APPENDIX D.1: Problem definition and initialization

```
%CostFunction.m -  Here the objective functon is defined
function [z, Sol]=CostFunction(x)
    %n=numel(x);
    %f1=x(1);

    %g=1+9/(n-1)*sum(x(2:end));

    %h=1-sqrt(f1/g);

    %f2=g*h;

    %f3=1/(f2+f1);


    f1=62.19 + 0.4573.*x(1) + 0.0259 .*x(2) - 0.02421 .*x(3) + 0.03638 .*x(4) - 
0.010867 .*x(1).^2 ...
    - 0.000029 .*x(2).^2 + 0.000005 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0003 
.*x(1).*x(2) + 0.00022 .*x(1).*x(3) ...
    - 0.00042 .*x(1).*x(4) - 0.000005 .*x(2).*x(3) - 0.000005 .*x(2).*x(4) - 0.000003 
.*x(3).*x(4);

    f2= 13.1 + 3.722.*x(1) + 0.2003 .*x(2) - 0.0122 .*x(3) + 0.0451 .*x(4) - 0.03047 
.*x(1).^2 ...
    + 0.000296 .*x(2).^2 + 0.00004 .*x(3).^2 + 0.000052 .*x(4).^2 + 0.0049 
.*x(1).*x(2) - 0.002940 .*x(1).*x(3) ...
    + 0.00286 .*x(1).*x(4) - 0.000285 .*x(2).*x(3) + 0.000285 .*x(2).*x(4) - 0.000099 
.*x(3).*x(4);

    f3 = -29.0 - 0.36.*x(1) + 0.287 .*x(2) + 0.0659 .*x(3) + 0.0133 .*x(4) - 0.01807 
.*x(1).^2 ...
    - 0.000029 .*x(2).^2 - 0.000011 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0125 
.*x(1).*x(2) + 0.0003 .*x(1).*x(3) ...
    + 0.00086 .*x(1).*x(4) - 0.000205 .*x(2).*x(3) + 0.000195 .*x(2).*x(4) - 0.000009 
.*x(3).*x(4);



    z=[f1
       f2
       f3];
   Sol.f1=f1;
   Sol.f2=f2;
   Sol.f3=f3;
   Sol.x=x;
end
```

APPENDIX D.2. Iteration improvement

```
%SelectLeader.m

function leader=SelectLeader(rep,beta)
    % Grid Index of All Repository Members
    GI=[rep.GridIndex];

    % Occupied Cells
    OC=unique(GI);

    % Number of Particles in Occupied Cells
    N=zeros(size(OC));
    for k=1:numel(OC)
        N(k)=numel(find(GI==OC(k)));
    end

    % Selection Probabilities
    P=exp(-beta*N);
    P=P/sum(P);

    % Selected Cell Index
    sci=RouletteWheelSelection(P);

    % Selected Cell
    sc=OC(sci);

    % Selected Cell Members
    SCM=find(GI==sc);

    % Selected Member Index
    smi=randi([1 numel(SCM)]);

    % Selected Member
    sm=SCM(smi);

    % Leader
    leader=rep(sm);
end


    %CreateGrid.m
function Grid=CreateGrid(pop,nGrid,alpha)
    c=[pop.Cost];

    cmin=min(c,[],2);
    cmax=max(c,[],2);

    dc=cmax-cmin;
    cmin=cmin-alpha*dc;
    cmax=cmax+alpha*dc;

    nObj=size(c,1);

    empty_grid.LB=[];
    empty_grid.UB=[];
    Grid=repmat(empty_grid,nObj,1);

    for j=1:nObj

        cj=linspace(cmin(j),cmax(j),nGrid+1);
        Grid(j).LB=[-inf cj];
        Grid(j).UB=[cj +inf];

    end
end

%FindGridIndex.m

function particle=FindGridIndex(particle,Grid)
    nObj=numel(particle.Cost);

    nGrid=numel(Grid(1).LB);

    particle.GridSubIndex=zeros(1,nObj);

    for j=1:nObj

        particle.GridSubIndex(j)=...
            find(particle.Cost(j)<Grid(j).UB,1,'first');

    end
    particle.GridIndex=particle.GridSubIndex(1);
    for j=2:nObj
        particle.GridIndex=particle.GridIndex-1;
        particle.GridIndex=nGrid*particle.GridIndex;
        particle.GridIndex=particle.GridIndex+particle.GridSubIndex(j);
    end

end

%Roulette wheel
function i=RouletteWheelSelection(P)
    r=rand;

    C=cumsum(P);

    i=find(r<=C,1,'first');
end
```

APPENDIX D.3. Evaluation

```
%MainMOHHO.m – the constraints are defined here. It calls all functions. All
files/functions must be in same folder as the MainMOHHO.m.

clc;
clear;
close all;
%% Problem Definition
CostFunction=@(x) CostFunction(x);      % Cost Function
nVar=4;              % Number of Decision Variables
VarSize=[1 nVar];    % Size of Decision Variables Matrix
Alb = [10];
Aub = [15];
```

```
Blb = [10];
Bub = [30];
Clb = [1420];
Cub = [1520];
Dlb = [850];
Dub = [950];
%lb=[Alb Blb Clb Dlb];
%ub=[Aub Bub Cub Dub];
VarMin=[Alb Blb Clb Dlb];          % Lower Bound of Variables
VarMax=[Aub Bub Cub Dub];
%VarMin=10;          % Lower Bound of Variables
%VarMax=1520;          % Upper Bound of Variables
%% MOPSO Parameters
MaxIt=100;           % Maximum Number of Iterations
nPop=100;            % Population Size
nRep=100;            % Repository Size
w=0.5;              % Inertia Weight
wdamp=0.99;         % Intertia Weight Damping Rate
c1=1;               % Personal Learning Coefficient
c2=2;               % Global Learning Coefficient
nGrid=4;            % Number of Grids per Dimension
alpha=0.1;          % Inflation Rate
beta=2;             % Leader Selection Pressure
gamma=2;            % Deletion Selection Pressure
mu=0.1;             % Mutation Rate
%% Initialization
empty_Rabbit.Location=[];
empty_Rabbit.Cost=[];
empty_Rabbit.Sol=[];
empty_Rabbit.IsDominated=[];
empty_Rabbit.GridIndex=[];
empty_Rabbit.GridSubIndex=[];
Rabbits=repmat(empty_Rabbit,nPop,1);
X = zeros(nPop, nVar);
Rabbit_Location=zeros(VarSize);
Rabbit_Energy=inf;
for i=1:nPop

    Rabbits(i).Location = rand(VarSize).*(VarMax-VarMin)+VarMin;
    X(i,:) = rand(VarSize).*(VarMax-VarMin)+VarMin;
    [Rabbits(i).Cost, Rabbits(i).Sol] = CostFunction(Rabbits(i).Location);

end
% Determine Domination
Rabbits=DetermineDomination(Rabbits);
rep=Rabbits(~[Rabbits.IsDominated]);
Grid=CreateGrid(rep,nGrid,alpha);
for i=1:numel(rep)
    rep(i)=FindGridIndex(rep(i),Grid);
end
%% MOPSO Main Loop
for it=1:MaxIt
    E1=2*(1-(it/MaxIt)); % factor to show the decreaing energy of rabbit
    for i=1:nPop
```

```
        leader=SelectLeader(rep,beta);


        E0=2*rand()-1; %-1<E0<1
        Escaping_Energy=E1*(E0);  % escaping energy of rabbit

        if abs(Escaping_Energy)>=1
            %% Exploration:
            % Harris' hawks perch randomly based on 2 strategy:

            q=rand();
            rand_Hawk_index = floor(nPop*rand()+1);
            X_rand = Rabbits(rand_Hawk_index);
            if q<0.5
                % perch based on other family members
                Rabbits(i).Location=X_rand.Location-rand()*abs(X_rand.Location-
2*rand()*Rabbits(i).Location);
                X(i,:)=X_rand.Location-rand()*abs(X_rand.Location-
2*rand()*Rabbits(i).Location);
            elseif q>=0.5
                % perch on a random tall tree (random site inside group's home range)
                Rabbits(i).Location=(leader.Location-mean(X))-rand()*((VarMax-
VarMin)*rand+VarMin);
                X(i,:)=(leader.Location-mean(X))-rand()*((VarMax-
VarMin)*rand+VarMin);
            end

        elseif abs(Escaping_Energy)<1
            %% Exploitation:
            % Attacking the rabbit using 4 strategies regarding the behavior of the
rabbit

            %% phase 1: surprise pounce (seven kills)
            % surprise pounce (seven kills): multiple, short rapid dives by different
hawks

            r=rand(); % probablity of each event

            if r>=0.5 && abs(Escaping_Energy)<0.5 % Hard besiege
                Rabbits(i).Location=(leader.Location)-
Escaping_Energy*abs(leader.Location-Rabbits(i).Location);
                X(i,:)=(leader.Location)-Escaping_Energy*abs(leader.Location-X(i,:));
            end

            if r>=0.5 && abs(Escaping_Energy)>=0.5  % Soft besiege
                Jump_strength=2*(1-rand()); % random jump strength of the rabbit
                X(i,:)=(leader.Location-X(i,:))-
Escaping_Energy*abs(Jump_strength*Rabbit_Location-X(i,:));
                Rabbits(i).Location=(leader.Location-Rabbits(i).Location)-
Escaping_Energy*abs(Jump_strength*Rabbit_Location-Rabbits(i).Location);
            end

            %% phase 2: performing team rapid dives (leapfrog movements)
            if r<0.5 && abs(Escaping_Energy)>=0.5 % Soft besiege % rabbit try to
escape by many zigzag deceptive motions
```

```
                Jump_strength=2*(1-rand());
                X1.Location=leader.Location-
Escaping_Energy*abs(Jump_strength*leader.Location-X(i,:));
                [X1.Cost, X1.Sol] = CostFunction(X1.Location);
                if Dominates(X1,Rabbits(i))
                    Rabbits(i).Location=X1.Location;
                    Rabbits(i).Cost=X1.Cost;
                    Rabbits(i).Cost=X1.Cost;
                elseif Dominates(Rabbits(i),X1)
                    X2.Location=leader.Location-
Escaping_Energy*abs(Jump_strength*leader.Location-X(i,:))+rand(1,nVar).*Levy(nVar);
                    [X2.Cost, X2.Sol] = CostFunction(X2.Location);
                    if Dominates(X2,Rabbits(i))
                        Rabbits(i).Location=X2.Location;
                        Rabbits(i).Cost=X2.Cost;
                        Rabbits(i).Cost=X2.Cost;
                    end
                else
                    if rand<0.5
                        Rabbits(i).Location=X1.Location;
                        Rabbits(i).Cost=X1.Cost;
                        Rabbits(i).Cost=X1.Cost;
                        Rabbits(i).Sol=X1.Sol;
                    end
                end
            end

            if r<0.5 && abs(Escaping_Energy)<0.5 % Hard besiege % rabbit try to
escape by many zigzag deceptive motions
                % hawks try to decrease their average location with the rabbit
                Jump_strength=2*(1-rand());
                X1.Location=leader.Location-
Escaping_Energy*abs(Jump_strength*leader.Location-mean(X));
                [X1.Cost, X1.Sol] = CostFunction(X1.Location);
                if Dominates(X1,Rabbits(i))
                    Rabbits(i).Location=X1.Location;
                    Rabbits(i).Cost=X1.Cost;
                    Rabbits(i).Cost=X1.Cost;
                elseif Dominates(Rabbits(i),X1)
                    X2.Location=leader.Location-
Escaping_Energy*abs(Jump_strength*leader.Location-mean(X))+rand(1,nVar).*Levy(nVar);
                    [X2.Cost, X2.Sol] = CostFunction(X2.Location);
                    if Dominates(X2,Rabbits(i))
                        Rabbits(i).Location=X2.Location;
                        Rabbits(i).Cost=X2.Cost;
                        Rabbits(i).Cost=X2.Cost;
                    end
                else
                    if rand<0.5
                        Rabbits(i).Location=X1.Location;
                        Rabbits(i).Cost=X1.Cost;
                        Rabbits(i).Cost=X1.Cost;
                        Rabbits(i).Sol=X1.Sol;
                    end
                end
            end
        end

        Rabbits(i).Location = max(Rabbits(i).Location, VarMin);
        Rabbits(i).Location = min(Rabbits(i).Location, VarMax);

        [Rabbits(i).Cost, Rabbits(i).Sol] = CostFunction(Rabbits(i).Location);

        % Apply Mutation
        pm=(1-(it-1)/(MaxIt-1))^(1/mu);
        if rand<pm
            NewSol.Location=Mutate(Rabbits(i).Location,pm,VarMin,VarMax);
            [NewSol.Cost, NewSol.Sol]=CostFunction(NewSol.Location);
            if Dominates(NewSol,Rabbits(i))
                Rabbits(i).Location=NewSol.Location;
                Rabbits(i).Cost=NewSol.Cost;
                Rabbits(i).Sol=NewSol.Sol;
            elseif Dominates(Rabbits(i),NewSol)
                % Do Nothing
            else
                if rand<0.5
                    Rabbits(i).Location=NewSol.Location;
                    Rabbits(i).Cost=NewSol.Cost;
                    Rabbits(i).Sol=NewSol.Sol;
                end
            end
        end

    end

    % Add Non-Dominated Particles to REPOSITORY
    rep=[rep
         Rabbits(~[Rabbits.IsDominated])]; %#ok

    % Determine Domination of New Resository Members
    rep=DetermineDomination(rep);

    % Keep only Non-Dminated Memebrs in the Repository
    rep=rep(~[rep.IsDominated]);

    % Update Grid
    Grid=CreateGrid(rep,nGrid,alpha);
    % Update Grid Indices
    for i=1:numel(rep)
        rep(i)=FindGridIndex(rep(i),Grid);
    end

    % Check if Repository is Full
    if numel(rep)>nRep

        Extra=numel(rep)-nRep;
        for e=1:Extra
            rep=DeleteOneRepMemebr(rep,gamma);
        end
    end

    % Plot Costs
    figure(1);
    PlotCosts(Rabbits,rep);
    pause(0.01);

    % Show Iteration Information
    disp(['Iteration ' num2str(it) ': Number of Rep Members = '
num2str(numel(rep))]);

    % Damping Inertia Weight
    w=w*wdamp;
end
%% Results
solutions = [];
costs = [];
%Sols = [];
for i=1:numel(rep)
    %disp(['solutions = ' num2str(rep.Location(i))]);
    %disp(['costs = ' num2str(rep.Cost(i))]);
    %disp(['Sols = ' num2str(rep.Sol(i))]);
%solutions = rep.Location;
%costs = rep.Cost ;
%Sols = rep.Sol;
solutions = cat(1, solutions, rep.Location);
costs = cat(1, costs, rep.Cost);
%Sols = cat(1, Sols, rep.Sol);
End


%DetermineDomination.m

function pop=DetermineDomination(pop)
    nPop=numel(pop);

    for i=1:nPop
        pop(i).IsDominated=false;
    end

    for i=1:nPop-1
        for j=i+1:nPop

            if Dominates(pop(i),pop(j))
               pop(j).IsDominated=true;
            end

            if Dominates(pop(j),pop(i))
               pop(i).IsDominated=true;
            end

        end
    end
end
```

```
%Dominates.m
function b=Dominates(x,y)
    if isstruct(x)
        x=x.Cost;
    end

    if isstruct(y)
        y=y.Cost;
    end
    b=all(x<=y) && any(x<y);
end


%DeletOneRepMember.m

function rep=DeleteOneRepMemebr(rep,gamma)
    % Grid Index of All Repository Members
    GI=[rep.GridIndex];

    % Occupied Cells
    OC=unique(GI);

    % Number of Particles in Occupied Cells
    N=zeros(size(OC));
    for k=1:numel(OC)
        N(k)=numel(find(GI==OC(k)));
    end

    % Selection Probabilities
    P=exp(gamma*N);
    P=P/sum(P);

    % Selected Cell Index
    sci=RouletteWheelSelection(P);

    % Selected Cell
    sc=OC(sci);

    % Selected Cell Members
    SCM=find(GI==sc);

    % Selected Member Index
    smi=randi([1 numel(SCM)]);

    % Selected Member
    sm=SCM(smi);

    % Delete Selected Member
    rep(sm)=[];
end
%PlotCosts.m

function PlotCosts(pop,rep)
    % pop_costs=[pop.Cost];
    % %plot3(pop_costs(1,:),pop_costs(2,:), pop_costs(3,:),'k.', 'markersize', 12);
    % plot(pop_costs(1,:),pop_costs(2,:),'k.', 'markersize', 12);
    % hold on;
    %
    % rep_costs=[rep.Cost];
    % %plot3(rep_costs(1,:),rep_costs(2,:), rep_costs(3,:),'r*', 'markersize', 12);
    % plot(rep_costs(1,:),rep_costs(2,:),'r*', 'markersize', 12);
    %
    % xlabel('Net Power (MW)');
    % ylabel('CO2 Emission (gr/MJ)');
    % %zlabel('Exergy Efficincy (%)');
    % legend('Pop Cost','Rep Cost');
    % title('MOHHO Pareto Front for CO2 Emission vs Net Power');
    %
    % hold off;

    % pop_costs=[pop.Cost];
    % plot(pop_costs(1,:),pop_costs(3,:),'k.', 'markersize', 12);
    % hold on;
    %
    % rep_costs=[rep.Cost];
    % %plot3(rep_costs(1,:),rep_costs(2,:), rep_costs(3,:),'r*', 'markersize', 12);
    % plot(rep_costs(1,:),rep_costs(3,:),'b*', 'markersize', 12);
    %
    % xlabel('Net Power (MW)');
    % %ylabel('CO2 Emission (gr/MJ)');
    % ylabel('Exergy Efficiency (%)');
    % legend('Pop Cost','Rep Cost');
    % title('MOHHO Pareto Front for Exergy Efficiency vs Net Power');

    % grid on;
    %
    % hold off;
    pop_costs=[pop.Cost];
    plot(pop_costs(2,:),pop_costs(3,:),'k.', 'markersize', 12);
    hold on;

    rep_costs=[rep.Cost];
    %plot3(rep_costs(1,:),rep_costs(2,:), rep_costs(3,:),'r*', 'markersize', 12);
    plot(rep_costs(2,:),rep_costs(3,:),'g*', 'markersize', 12);

    %xlabel('Net Power (MW)');
    xlabel('CO2 Emission (gr/MJ)');
    ylabel('Exergy Efficiency (%)');
    legend('Pop Cost','Rep Cost');
    title('MOHHO Pareto Front for Exergy Efficiency vs CO2 Emission');
    grid on;
    %
    hold off;

end
```

APPENDIX E: MATLAB codes for single-objective antlion optimization

APPENDIX E.1: Problem definition and initialization

```
% Get_Function_details.m – for defining the constraints and objective function
function [lb,ub,dim,fobj] = Get_Functions_details(F)
switch F
case 'F1'
fobj = @F1;
lb=[10 10 1420 850];
ub=[15 30 1520 950];
dim=4;
end
end
% F1
function o = F1(x)
% o=62.19 + 0.4573.*x(1) + 0.0259 .*x(2) - 0.02421 .*x(3) + 0.03638 .*x(4) - 0.010867
.*x(1).^2 ...
% - 0.000029 .*x(2).^2 + 0.000005 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0003
.*x(1).*x(2) + 0.00022 .*x(1).*x(3) ...
% - 0.00042 .*x(1).*x(4) - 0.000005 .*x(2).*x(3) - 0.000005 .*x(2).*x(4) - 0.000003
.*x(3).*x(4);
% o= 13.1 + 3.722.*x(1) + 0.2003 .*x(2) - 0.0122 .*x(3) + 0.0451 .*x(4) - 0.03047
.*x(1).^2 ...
% + 0.000296 .*x(2).^2 + 0.00004 .*x(3).^2 + 0.000052 .*x(4).^2 + 0.0049 .*x(1).*x(2)
- 0.002940 .*x(1).*x(3) ...
% + 0.00286 .*x(1).*x(4) - 0.000285 .*x(2).*x(3) + 0.000285 .*x(2).*x(4) - 0.000099
.*x(3).*x(4);
o= -29.0 - 0.36.*x(1) + 0.287 .*x(2) + 0.0659 .*x(3) + 0.0133 .*x(4) - 0.01807
.*x(1).^2 ...
- 0.000029 .*x(2).^2 - 0.000011 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0125 .*x(1).*x(2)
+ 0.0003 .*x(1).*x(3) ...
+ 0.00086 .*x(1).*x(4) - 0.000205 .*x(2).*x(3) + 0.000195 .*x(2).*x(4) - 0.000009
.*x(3).*x(4);
end


% Initialization.m – ensures the initialized values do not exceed the defined
constraints.
%This function creates the first random population

function X=initialization(SearchAgents_no,dim,ub,lb)

Boundary_no= size(ub,2); % numnber of boundaries

% If the boundaries of all variables are equal and user enter a signle
% number for both ub and lb
if Boundary_no==1
    X=rand(SearchAgents_no,dim).*(ub-lb)+lb;
end

% If each variable has a different lb and ub
if Boundary_no>1
    for i=1:dim
```

```
        ub_i=ub(i);
        lb_i=lb(i);
        X(:,i)=rand(SearchAgents_no,1).*(ub_i-lb_i)+lb_i;
    end
end
```

APPENDIX E.2: Evaluation

```
%ALO.m -  Here, the random walk function is called to obtain the elite ant
%and antlion positions. Updates are carried out and the optimal solution is
%returned.


function
[Elite_antlion_fitness,Elite_antlion_position,Convergence_curve]=ALO(N,Max_iter,lb,ub
,dim,fobj)

% Initialize the positions of antlions and ants
antlion_position=initialization(N,dim,ub,lb);
ant_position=initialization(N,dim,ub,lb);

% Initialize variables to save the position of elite, sorted antlions,
% convergence curve, antlions fitness, and ants fitness
Sorted_antlions=zeros(N,dim);
Elite_antlion_position=zeros(1,dim);
Elite_antlion_fitness=inf;
Convergence_curve=zeros(1,Max_iter);
antlions_fitness=zeros(1,N);
ants_fitness=zeros(1,N);

% Calculate the fitness of initial antlions and sort them
for i=1:size(antlion_position,1)
    antlions_fitness(1,i)=fobj(antlion_position(i,:));
end

[sorted_antlion_fitness,sorted_indexes]=sort(antlions_fitness);

for newindex=1:N
     Sorted_antlions(newindex,:)=antlion_position(sorted_indexes(newindex),:);
end

Elite_antlion_position=Sorted_antlions(1,:);
Elite_antlion_fitness=sorted_antlion_fitness(1);

% Main loop start from the second iteration since the first iteration
% was dedicated to calculating the fitness of antlions
Current_iter=2;
while Current_iter<Max_iter+1

    % This for loop simulate random walks
    for i=1:size(ant_position,1)
        % Select ant lions based on their fitness (the better anlion the higher
chance of catching ant)
```

```
        Rolette_index=RouletteWheelSelection(1./sorted_antlion_fitness);
        if Rolette_index==-1
            Rolette_index=1;
        end

        % RA is the random walk around the selected antlion by rolette wheel
        RA=Random_walk_around_antlion(dim,Max_iter,lb,ub,
Sorted_antlions(Rolette_index,:),Current_iter);

        % RA is the random walk around the elite (best antlion so far)
        [RE]=Random_walk_around_antlion(dim,Max_iter,lb,ub,
Elite_antlion_position(1,:),Current_iter);

        ant_position(i,:)= (RA(Current_iter,:)+RE(Current_iter,:))/2; %
    end

    for i=1:size(ant_position,1)

        % Boundar checking (bring back the antlions of ants inside search
        % space if they go beyoud the boundaries
        Flag4ub=ant_position(i,:)>ub;
        Flag4lb=ant_position(i,:)<lb;

ant_position(i,:)=(ant_position(i,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag4lb;

        ants_fitness(1,i)=fobj(ant_position(i,:));

    end

    % Update antlion positions and fitnesses based of the ants (if an ant
    % becomes fitter than an antlion we assume it was cought by the antlion
    % and the antlion update goes to its position to build the trap)
    double_population=[Sorted_antlions;ant_position];
    double_fitness=[sorted_antlion_fitness ants_fitness];

    [double_fitness_sorted I]=sort(double_fitness);
    double_sorted_population=double_population(I,:);

    antlions_fitness=double_fitness_sorted(1:N);
    Sorted_antlions=double_sorted_population(1:N,:);

    % Update the position of elite if any antlinons becomes fitter than it
    if antlions_fitness(1)<Elite_antlion_fitness
        Elite_antlion_position=Sorted_antlions(1,:);
        Elite_antlion_fitness=antlions_fitness(1);
    end

    % Keep the elite in the population
    Sorted_antlions(1,:)=Elite_antlion_position;
    antlions_fitness(1)=Elite_antlion_fitness;

    % Update the convergence curve
    Convergence_curve(Current_iter)=Elite_antlion_fitness;

    % Display the iteration and best optimum obtained so far
    if mod(Current_iter,50)==0
        display(['At iteration ', num2str(Current_iter), ' the elite fitness is ',
num2str(Elite_antlion_fitness)]);
    end

    Current_iter=Current_iter+1;
end



%Random_walk_around_antlion.m - Random walk of the ant around the antlion used in
obtaining the elite ant and antlion positions
% This function creates random walks

function [RWs]=Random_walk_around_antlion(Dim,max_iter,lb, ub,antlion,current_iter)
if size(lb,1) ==1 && size(lb,2)==1 %Check if the bounds are scalar
    lb=ones(1,Dim)*lb;
    ub=ones(1,Dim)*ub;
end

if size(lb,1) > size(lb,2) %Check if boundary vectors are horizontal or vertical
    lb=lb';
    ub=ub';
end

I=1; % I is the ratio in Equations (3.53) and (3.54)

if current_iter>max_iter/10
    I=1+100*(current_iter/max_iter);
end

if current_iter>max_iter/2
    I=1+1000*(current_iter/max_iter);
end

if current_iter>max_iter*(3/4)
    I=1+10000*(current_iter/max_iter);
end

if current_iter>max_iter*(0.9)
    I=1+100000*(current_iter/max_iter);
end

if current_iter>max_iter*(0.95)
    I=1+1000000*(current_iter/max_iter);
end


% Decrease boundaries to converge towards antlion
lb=lb/(I); % Equation (3.53) in the thesis
ub=ub/(I); % Equation (3.54) in the thesis

% Move the interval of [lb ub] around the antlion [lb+anlion ub+antlion]
if rand<0.5
    lb=lb+antlion; % Equation (3.51) in the thesis
```

```
else
    lb=-lb+antlion;
end

if rand>=0.5
    ub=ub+antlion; % Equation (3.52) in the thesis
else
    ub=-ub+antlion;
end

% This function creates n random walks and normalize according to lb and ub
% vectors
for i=1:Dim
    X = [0 cumsum(2*(rand(max_iter,1)>0.5)-1)']; % Equation (3.44) in the thesis
    %[a b]--->[c d]
    a=min(X);
    b=max(X);
    c=lb(i);
    d=ub(i);
    X_norm=((X-a).*(d-c))./(b-a)+c; % Equation (3.50) in the thesis
    RWs(:,i)=X_norm;
end

% RouletteWheelSelection.m – defines how the optimal solutions are selected.
function choice = RouletteWheelSelection(weights)
  accumulation = cumsum(weights);
  p = rand() * accumulation(end);
  chosen_index = -1;
  for index = 1 : length(accumulation)
    if (accumulation(index) > p)
      chosen_index = index;
      break;
    end
  end
  choice = chosen_index;


% main.m – calls the ALO and Get_Functions_details to obtain the optimal solution.
All functions must be same folder as the main.m file


clear all
clc
SearchAgents_no=40; % Number of search agents
Function_name='F1'; % Name of the test function that can be from F1 to F23 (Table
1,2,3 in the paper)
Max_iteration=100; % Maximum number of iterations
% Load details of the selected benchmark function
[lb,ub,dim,fobj]=Get_Functions_details(Function_name);
[Best_score,Best_pos,cg_curve]=ALO(SearchAgents_no,Max_iteration,lb,ub,dim,fobj);
figure('Position',[500 500 660 290])
%Draw search space
subplot(1,2,1);
func_plot(Function_name);
title('Test function')
```

```
xlabel('x_1');
ylabel('x_2');
zlabel([Function_name,'( x_1 , x_2 )'])
grid off
%Draw objective space
subplot(1,2,2);
semilogy(cg_curve,'Color','r')
title('Convergence curve')
xlabel('Iteration');
ylabel('Best score obtained so far');
axis tight
grid off
box on
legend('ALO')
display(['The best solution obtained by ALO is : ', num2str(Best_pos)]);
display(['The best optimal value of the objective funciton found by ALO is : ',
num2str(Best_score)]);
```

APPENDIX F: MATLAB codes for multi-objective antlion optimization

APPENDIX F.1: Problem definition

```
% Modify this file with respect to your objective function
function o = ZDT1(x)

%o = [0, 0];
%dim = length(x);
%g = 1 + 9*sum(x(2:dim))/(dim-1);

%o(1) = x(1);
%o(2) = g*(1-sqrt(x(1)/g));
o(1)=62.19 + 0.4573.*x(1) + 0.0259 .*x(2) - 0.02421 .*x(3) + 0.03638 .*x(4) -
0.010867 .*x(1).^2 ...
- 0.000029 .*x(2).^2 + 0.000005 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0003 .*x(1).*x(2)
+ 0.00022 .*x(1).*x(3) ...
- 0.00042 .*x(1).*x(4) - 0.000005 .*x(2).*x(3) - 0.000005 .*x(2).*x(4) - 0.000003
.*x(3).*x(4);

o(2)= 13.1 + 3.722.*x(1) + 0.2003 .*x(2) - 0.0122 .*x(3) + 0.0451 .*x(4) - 0.03047
.*x(1).^2 ...
+ 0.000296 .*x(2).^2 + 0.00004 .*x(3).^2 + 0.000052 .*x(4).^2 + 0.0049 .*x(1).*x(2) -
0.002940 .*x(1).*x(3) ...
+ 0.00286 .*x(1).*x(4) - 0.000285 .*x(2).*x(3) + 0.000285 .*x(2).*x(4) - 0.000099
.*x(3).*x(4);

o(3) = -29.0 - 0.36.*x(1) + 0.287 .*x(2) + 0.0659 .*x(3) + 0.0133 .*x(4) - 0.01807
.*x(1).^2 ...
- 0.000029 .*x(2).^2 - 0.000011 .*x(3).^2 - 0.000009 .*x(4).^2 - 0.0125 .*x(1).*x(2)
+ 0.0003 .*x(1).*x(3) ...
+ 0.00086 .*x(1).*x(4) - 0.000205 .*x(2).*x(3) + 0.000195 .*x(2).*x(4) - 0.000009
.*x(3).*x(4);
```

APPENDIX F.2: Initialization and iteration improvement

```
%Initialization.m

% This function initialize the first population of search agents
function Positions=initialization(SearchAgents_no,dim,ub,lb)

Boundary_no= size(ub,2); % numnber of boundaries

% If the boundaries of all variables are equal and user enter a single
% number for both ub and lb
if Boundary_no==1
    ub_new=ones(1,dim)*ub;
    lb_new=ones(1,dim)*lb;
else
     ub_new=ub;
     lb_new=lb;
end

% If each variable has a different lb and ub
    for i=1:dim
        ub_i=ub_new(i);
        lb_i=lb_new(i);
        Positions(:,i)=rand(SearchAgents_no,1).*(ub_i-lb_i)+lb_i;
    end

Positions=Positions';

% Roulette Wheel Selection Algorithm.m

% ---------------------------------------------------------
% Roulette Wheel Selection Algorithm. A set of weights
% represents the probability of selection of each
% individual in a group of choices. It returns the index
% of the chosen individual.
% Usage example:
% fortune_wheel ([1 5 3 15 8 1])
%    most probable result is 4 (weights 15)
% ---------------------------------------------------------

function choice = RouletteWheelSelection(weights)
  accumulation = cumsum(weights);
  p = rand() * accumulation(end);
  chosen_index = -1;
  for index = 1 : length(accumulation)
    if (accumulation(index) > p)
      chosen_index = index;
      break;
    end
  end
  choice = chosen_index;


function ranks=RankingProcess(Archive_F, ArchiveMaxSize, obj_no)

global my_min;
global my_max;

%if all(my_min>min(Archive_F))
if size(Archive_F,1) == 1 && size(Archive_F,2) == 2
    my_min = Archive_F;
    my_max = Archive_F;
else
    my_min=min(Archive_F);
    %end
```

APPENDIX F.3: Evaluation

```
%Random_walk_around_antlion.m - Random walk of the ant around the antlion used in
obtaining the elite ant and antlion positions
% This function creates random walks


% This function creates random walks
```

```
function [RWs]=Random_walk_around_antlion(Dim,max_iter,lb, ub,antlion,current_iter)
if size(lb,1) ==1 && size(lb,2)==1 %Check if the bounds are scalar
    lb=ones(1,Dim)*lb;
    ub=ones(1,Dim)*ub;
end

if size(lb,1) > size(lb,2) %Check if boundary vectors are horizontal or vertical
    lb=lb';
    ub=ub';
end

I=1; % I is the ratio in Equations (3.53) and (3.54)

if current_iter>max_iter/10
    I=1+100*(current_iter/max_iter);
end

if current_iter>max_iter/2
    I=1+1000*(current_iter/max_iter);
end

if current_iter>max_iter*(3/4)
    I=1+10000*(current_iter/max_iter);
end

if current_iter>max_iter*(0.9)
    I=1+100000*(current_iter/max_iter);
end

if current_iter>max_iter*(0.95)
    I=1+1000000*(current_iter/max_iter);
end


% Dicrease boundaries to converge towards antlion
lb=lb/(I); % Equation (3.53) in the thesis
ub=ub/(I); % Equation (3.54) in the thesis

% Move the interval of [lb ub] around the antlion [lb+anlion ub+antlion]
if rand<0.5
    lb=lb+antlion; % Equation (3.51) in the thesis
else
    lb=-lb+antlion;
end

if rand>=0.5
    ub=ub+antlion; % Equation (3.52) in the thesis
else
    ub=-ub+antlion;
end

% This function creates n random walks and normalize accroding to lb and ub
% vectors
for i=1:Dim
    X = [0 cumsum(2*(rand(max_iter,1)>0.5)-1)']; % Equation (3.44) in the thesis
    %[a b]--->[c d]
    a=min(X);
    b=max(X);
    c=lb(i);
    d=ub(i);
    X_norm=((X-a).*(d-c))./(b-a)+c; % Equation (3.50) in the thesis
    RWs(:,i)=X_norm;
End

%HandleFullArchive

function [Archive_X_Chopped, Archive_F_Chopped, Archive_mem_ranks_updated,
Archive_member_no]=HandleFullArchive(Archive_X, Archive_F, Archive_member_no,
Archive_mem_ranks, ArchiveMaxSize)


for i=1:size(Archive_F,1)-ArchiveMaxSize
    index=RouletteWheelSelection(Archive_mem_ranks);
    %[value index]=min(Archive_mem_ranks);

    Archive_X=[Archive_X(1:index-1,:) ; Archive_X(index+1:Archive_member_no,:)];
    Archive_F=[Archive_F(1:index-1,:) ; Archive_F(index+1:Archive_member_no,:)];
    Archive_mem_ranks=[Archive_mem_ranks(1:index-1)
Archive_mem_ranks(index+1:Archive_member_no)];
    Archive_member_no=Archive_member_no-1;
end

Archive_X_Chopped=Archive_X;
Archive_F_Chopped=Archive_F;
Archive_mem_ranks_updated=Archive_mem_ranks;

%UpdateArchive

function [Archive_X_updated, Archive_F_updated,
Archive_member_no]=UpdateArchive(Archive_X, Archive_F, Particles_X, Particles_F,
Archive_member_no)
Archive_X_temp=[Archive_X ; Particles_X'];
Archive_F_temp=[Archive_F ; Particles_F];

o=zeros(1,size(Archive_F_temp,1));

for i=1:size(Archive_F_temp,1)
    o(i)=0;
    for j=1:i-1
        if any(Archive_F_temp(i,:) ~= Archive_F_temp(j,:))
            if dominates(Archive_F_temp(i,:),Archive_F_temp(j,:))
                o(j)=1;
            elseif dominates(Archive_F_temp(j,:),Archive_F_temp(i,:))
                o(i)=1;
                break;
            end
        else
            o(j)=1;
            o(i)=1;
        end
    end
end

Archive_member_no=0;
index=0;
for i=1:size(Archive_X_temp,1)
    if o(i)==0
        Archive_member_no=Archive_member_no+1;
        Archive_X_updated(Archive_member_no,:)=Archive_X_temp(i,:);
        Archive_F_updated(Archive_member_no,:)=Archive_F_temp(i,:);
    else
        index=index+1;
        %         dominated_X(index,:)=Archive_X_temp(i,:);
        %         dominated_F(index,:)=Archive_F_temp(i,:);
    end
end

end

function o=dominates(x,y)

    o=all(x<=y) && any(x<y);

end



%MALO.m - Here, the constraints were defined. All other functions are called to yield
the Pareto optimal solutions.
clc;
clear;
close all;

% Change these details with respect to your problem%%%%%%%%%%%%%%
ObjectiveFunction=@ZDT1;
dim=4;
%Alb = [10];
%Aub = [15];
%Blb = [10];
%Bub = [30];
%Clb = [1420];
%Cub = [1520];
%Dlb = [850];
%Dub = [950];
%lb=[Alb Blb Clb Dlb];
%ub=[Aub Bub Cub Dub];
lb= [10 10 1420 850];
ub= [15 30 1520 950];
%lb=0;
%ub=1;
obj_no=3;

if size(ub,2)==1
    ub=ones(1,dim).*ub;
    lb=ones(1,dim).*lb;
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initial parameters of the MODA algorithm
max_iter=100;
N=100;
ArchiveMaxSize=6;

Archive_X=zeros(100,dim);
Archive_F=ones(100,obj_no)*inf;

Archive_member_no=0;

r=(ub-lb)/2;
V_max=(ub(1)-lb(1))/10;

Elite_fitness=inf*ones(1,obj_no);
Elite_position=zeros(dim,1);

Ant_Position=initialization(N,dim,ub,lb);
fitness=zeros(N,2);

V=initialization(N,dim,ub,lb);
iter=0;

position_history=zeros(N,max_iter,dim);

for iter=1:max_iter

    for i=1:N %Calculate all the objective values first
        Particles_F(i,:)=ObjectiveFunction(Ant_Position(:,i)');
        if dominates(Particles_F(i,:),Elite_fitness)
            Elite_fitness=Particles_F(i,:);
            Elite_position=Ant_Position(:,i);
        end
    end

    [Archive_X, Archive_F, Archive_member_no]=UpdateArchive(Archive_X, Archive_F, 
Ant_Position, Particles_F, Archive_member_no);

    if Archive_member_no>ArchiveMaxSize
        Archive_mem_ranks=RankingProcess(Archive_F, ArchiveMaxSize, obj_no);
        [Archive_X, Archive_F, Archive_mem_ranks, 
Archive_member_no]=HandleFullArchive(Archive_X, Archive_F, Archive_member_no, 
Archive_mem_ranks, ArchiveMaxSize);
    else
        Archive_mem_ranks=RankingProcess(Archive_F, ArchiveMaxSize, obj_no);
    end

    Archive_mem_ranks=RankingProcess(Archive_F, ArchiveMaxSize, obj_no);

    % Chose the archive member in the least population area as arrtactor
    % to improve coverage
    index=RouletteWheelSelection(1./Archive_mem_ranks);
    if index==-1
```

```
        index=1;
    end
    Elite_fitness=Archive_F(index,:);
    Elite_position=Archive_X(index,:)';

    Random_antlion_fitness=Archive_F(1,:);
    Random_antlion_position=Archive_X(1,:)';

    for i=1:N

        index=0;
        neighbours_no=0;

        RA=Random_walk_around_antlion(dim,max_iter,lb,ub,
Random_antlion_position',iter);

        [RE]=Random_walk_around_antlion(dim,max_iter,lb,ub, Elite_position',iter);

        Ant_Position(:,i)=(RE(iter,:)'+RA(iter,:)')/2;



        Flag4ub=Ant_Position(:,i)>ub';
        Flag4lb=Ant_Position(:,i)<lb';

Ant_Position(:,i)=(Ant_Position(:,i).*(~(Flag4ub+Flag4lb)))+ub'.*Flag4ub+lb'.*Flag4lb
;

    end
    display(['At the iteration ', num2str(iter), ' there are ',
num2str(Archive_member_no), ' non-dominated solutions in the archive']);
end

figure

%Draw_ZDT1();

%hold on

plot(Archive_F(:,2),Archive_F(:,3),'ko','MarkerSize',8,'markerfacecolor','k');

%legend('True PF','Obtained PF');
%xlabel('Net power (MW)')
ylabel('Exergy Efficiency (%)')
xlabel('C02 Emmission (gr/MJ)')
title('MOALO Pareto Front for Exergy Efficiency vs CO2 Emission');

set(gcf, 'pos', [403   466   230   200])
```