QUALITY OF SERVICE-BASED SERVICE COMPOSITION OPTIMIZATION IN SMART AGRICULTURE

Thesis submitted in fulfilment of the requirements for the Degree of

DOCTOR OF PHILOSOPHY

by

SHALINI SHARMA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, SOLAN-173234, HIMACHAL PRADESH, INDIA

DECEMBER, 2024

DECLARATION BY THE SCHOLAR

I hereby declare that the work reported in the Ph.D. thesis entitled **"Quality of service-based service composition optimization in smart agriculture"** submitted at **Jaypee University of Information Technology, Waknaghat, India,** is an authentic record of my work carried out under the supervision of **Dr. Rajiv Kumar and Dr. Bhupendra Kumar Pathak**. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my Ph.D. thesis.

Shalini

Shalini Sharma

Enrolment Number: 216005

Department of Electronics and Communication Engineering

Jaypee University of Information Technology, Waknaghat, Solan (H.P.), India

Date: 16-05-2025

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
Established under H.P. Legislative Assembly Act No. 14 of 2002
and Approved by UGC under section 2(f)

SUPERVISOR'S CERTIFICATE

This is to certify that the work in the thesis entitled **"Quality of service-based service composition optimization in smart agriculture"** submitted by **Shalini Sharma at Jaypee University of Information Technology, Waknaghat, India,** is a bonafide record of her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Dr. Rajiv Kumar
Professor
Department of Electronics and Communication Engineering
Jaypee University of Information Technology, Waknaghat, Solan, India
Date: 16-05-2025

Bhupathak

Dr. Bhupendra Kumar Pathak
Assistant Professor (Senior Grade)
Department of Mathematics
Jaypee University of Information Technology, Waknaghat, Solan, India
Date: 16-05-2025

ACKNOWLEDGEMENT

With the divine gracae of “Almighty God”, the journey of Ph. D. comes to an end. I would like to take this opportunity to thank every person who has helped me to complete this journey.

It is an immense pleasure to express my profound gratitude towards my supervisors **Dr. Rajiv Kumar**, Professor and Head, Department of Electronics and Communication Engineering, and **Dr. Bhupendra Kumar Pathak**, Assistant Professor (Senior Grade), Department of Mathematics, who graciously allowed me to work under their guidance. I am thankful for their patience, continuous support, optimistic approach, and time-to-time guidance throughout this course. Their valuable motivation, help, suggestions, affirmative vision, magnificent supervision, and enormous confidence in my abilities made me face tough circumstances during the progress of the research work. It is a great honor to work under their supervision.

I would like to express my gratitude to our Honorable Vice Chancellor Prof. (Dr.) Rajendra Kumar Sharma, Dean (Academics and Research) Prof. (Dr.) Ashok Kumar Gupta to promote the research and facilitate resources in the institution. I would also like to thank my doctoral committee members for their valuable feedback and critical reviews during presentations and from time to time.

I am also grateful for the support received from JUIT, Waknaghat. In particular, I thank all staff members at the Department of Electronics and Communication Engineering, JUIT, Waknaghat who have been extremely helpful on numerous occasions. I would like to thank my parents for supporting me spiritually throughout my life. I would like to express my heartfelt gratitude to my husband for his unwavering support, encouragement, and understanding throughout this journey, which has been a source of strength and inspiration for me. Last but not least, I thank my fellow Ph.D. friends for their consistent help and valuable discussion.

Date: Shalini Sharma

ABSTRACT

Agriculture forms the cornerstone of human existence and serves as the fundamental basis for all production. It is the foundation upon which every nation's economy is built. As populations expand, the demand for food production rises correspondingly. This growth, however, is accompanied by climate change and a scarcity of natural resources necessary for agricultural activities. The agricultural sector is undergoing a transformation through the incorporation of Information and Communications Technology (ICT), ushering in a new agricultural era. This shift enhances crop yields, refines decision-making related to crop management, minimizes the environmental impact of farming practices by lowering chemical consumption, and cuts costs related to water, electricity, and fuel consumption. Smart agriculture technologies enable farmers to cultivate crops more systematically and accurately predict outcomes. Nearly every aspect of farming, from planting to harvesting, benefits from technological advancements. Consequently, farmers gain a comprehensive understanding of their land, leading to a more logical production process with fewer arbitrary elements. The term "agriculture field" encompasses a wide range of services. Meeting the needs of an expanding population using a single service is increasingly challenging due to growing complexity. Therefore, it is crucial to select services based on user requirements and quality of service (QoS) with similar functionality, rather than solely on the functionality of the services. The potential for substantial QoS with non-linear impacts on the service composition goal function makes this an NP-hard problem, which cannot be resolved using conventional optimization methods. For such intricate issues, meta-heuristics approaches offer the best substitute. These can be categorized as bio-inspired, physical, evolutionary, and swarm intelligence-based approaches. These methods provide solutions for both single and multi-objective optimization problems.

The primary aim of this thesis is to develop an optimized agricultural planning system tailored to meet farmer's needs, offering significant advantages such as remote farm management, efficient resource utilization, and streamlined processes, ultimately enhancing farmer's income. The study addresses the optimization of several integrated services in smart agriculture, with time and cost as dual objectives that must be minimized. In the first phase of the thesis, multi-objective service composition optimization is conducted using a straightforward approach that assumes a linear relationship between the cost and time objectives. This phase employs a set of optimization algorithms—namely, the multi-objective genetic algorithm (MOGA), non-

dominated sorting genetic algorithm (NSGA-II), and multi-objective gaining-sharing knowledge-based algorithm (MOGSK). However, real-world applications often involve significant non-linearities that cannot be adequately represented by a linear model. Therefore, in the second phase, the same service composition problem is reconsidered, this time incorporating a non-linear relationship between the competing objectives. Lagrange's interpolation-based algorithm is used to address these non-linearities, and optimization is performed using the MOGA, NSGA-II, and MOGSK algorithms. Agricultural data often contains uncertain factors that must be considered, as they can significantly impact outcomes—a primary challenge for modern farmers. To address this, in the third phase, a fuzzy inference system (FIS) is used to assess the impact of these uncertain factors on smart agriculture. In the final phase, a novel nature-inspired algorithm—the multi-objective electric eel foraging optimization (MO-EEFO) algorithm—is proposed to tackle real-world optimization challenges in smart agriculture, as well as in other applications. This thesis aims to provide a customizable agricultural plan for farmers, allowing them to prioritize either time or cost optimization based on their specific requirements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

3G	Third Generation
4G	Fourth Generation
AI	Artificial Intelligence
aKNC-GB	Adaptive K-nearest Centroid Neighbor Classifier – Gradient Boost
aKNCN	Adaptive K-nearest Centroid Neighbor Classifier
aKNC-RF	Adaptive K-nearest Centroid Neighbor Classifier – Random Forest
aKNC-SVM	Adaptive K-nearest Centroid Neighbor Classifier – Support Vector Machine
ALP-GA	Automated Land Portioning Genetic Algorithm
ANNs	Artificial Neural Networks
ARC	Anomaly-aware Robust Service Composition
BA	Bees Algorithm
BCO	Bee Colony Optimization
BOA	Butterfly Optimization Algorithm
CMF	Cauchy Fuzzy Membership Function
COG	Center of Gravity
DE	Differential Evolution

EC	Evolutionary Computational
EEFO	Electric Eel Foraging Optimization
ELM	Extreme Learning Machine
EVS	Explained Variance Score
Ex-GWO	Expanded Gray Wolf Optimization
FAO	Food and Agriculture Organization
FBCO	Fuzzy Bee Colony Optimization
FCS	Fuzzy Cuckoo Search Algorithm
FFA	Firefly Algorithm
FGSA	Fuzzy Gravitational Search Algorithm
FIS	Fuzzy Inference System
FL	Fuzzy Logic
FLS	Fuzzy Logic System
FRRWLX	Fuzzy Rough Set Roulette Wheel Selection with Laplace Crossover
FS	Farmer Skills
Fuzzy-La-NSGA-II	Fuzzy Lagrange's NSGA-II
Fuzzy-Li-NSGA-II	Fuzzy Linear NSGA-II
GA	Genetic Algorithm
GIS	Geographic Information System
GSA	Gravitational Search Algorithm

GSK	Gaining Sharing Knowledge-based Algorithm
GSM	Global System for Mobile Communication
HMM-ACO	Hidden Markov Model-Ant Colony Optimization
IaaS	Infrastructure as a Service
ICA	Imperialist Competitive Algorithm
ICT	Information and Communication Technologies
I-GWO	Incremental Gray Wolf Optimization
IoT	Internet of Things
IT2FLC	Interval Type-II Fuzzy Logic Controller
JGS	Junior Gaining Sharing
JGSK	Junior Gaining Sharing Knowledge Phase
KNN	K-Nearest Neighbor
La-MOGA	Lagrange's Multi-objective Genetic Algorithm
La-MOGSK	Lagrange's Multi-objective Gaining Sharing Knowledge-based Algorithm
La-NSGA-II	Lagrange's Multi-objective Non-Dominated Sorting Genetic Algorithm
LDR	Light Dependent Resistor
LED	Light Emitting Diode

Li-MOGA	Linear Multi-objective Genetic Algorithm
Li-MOGSK	Linear Multi-objective Gaining Sharing Knowledge-based Algorithm
Li-NSGA-II	Linear Multi-objective Non-Dominated Sorting Genetic Algorithm
LoRa	Long Range
LoW-PAN	Low-power Wireless Personal Area Network
MAE	Model Evaluation Metric
MAPE	Mean Absolute Percent Error
MedAE	Median Absolute Error
mhCPPmp	Multi-heterogeneous UAVs Coverage Path Planning with Moving Ground Platform
ML	Machine Learning
MO-EEFO	Multi-objective Electric Eel Foraging Optimization
MOGA	Multi-objective Genetic Algorithm
MOGSK	Multi-objective Gaining Sharing Knowledge based Algorithm
MOM	Mean of Maximum
MOPSO	Multi-objective Particle Swarm Optimization
MS	Management Skills
MSE	Mean Squared Error

MSLE	Mean Squared Logarithmic
NLP	Natural Language Processing
NP-Hard	Non-deterministic Polynomial-time Hard
NPK	Nitrogen, Phosphorus, and Potassium
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NSWOA	Non-dominated Sorting Whale Optimization Algorithm
PaaS	Platform as a Service
PDCA	Plando-check-act
pH sensors	Potential of Hydrogen sensors
QoS	Quality of Service
RCGA	Real Coded Genetic Algorithm
RFID	Radio Frequency Identification
RMSE	Root Mean Square Error
RSRWLC	Rough Set Real Coded based Genetic Algorithm with Roulette Wheel Selection and Laplace Crossover
RWFX	Roulette with Flat
RWLX	Roulette with Laplace
RWSX	Roulette with Simple
SA	Simulated Annealing
SaaS	Software as a Service

SBX	Simulate Binary Crossover
SC	Service Composition
SCO	Service Composition Optimization
SFAIS	Smart Farm Automatic Irrigation System
SGS	Senior Gaining Sharing
SGSK	Senior Gaining Sharing Knowledge Phase
SLA	Service Level Agreement
SMS	Short Message Service
SSA	Social Spider Algorithm
T1FGSA	Type-I Fuzzy Gravitational Search Algorithm
T1FLC	Type-I Fuzzy Logic Controller
TSFX	Tournament with Flat
TSLX	Tournament with Laplace
TSSX	Tournament with Simple
UAVs	Unmanned Aerial Vehicles
UML	Unified Modelling Language
UOD	Universe of Discourse
UV	Ultraviolet
WC	Weather Conditions
Wi-Fi	Wireless Fidelity

WiMax	Worldwide Interoperability for Microwave Access
WOA	Whale Optimization Algorithm
WSNs	Wireless Sensor Networks
WTMCS	Water Tank Monitoring and Control Subsystem
ZDT	Zitzler-Deb-Thiele

CHAPTER-1
INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1Chapter Overview

The QoS-based service composition optimization problem and its application in smart agriculture are thoroughly explained in this chapter. Depending on whether the problem is single-objective or multi-objective, it focuses on solving these challenges through optimization using different evolutionary algorithms that are influenced by nature and biology. The chapter also describes the two categories into which multi-objective problems fall: preference-based and ideal multi-objective. Scalarization techniques are employed for preference-based approaches and Pareto-based techniques are used for ideal solutions to solve these difficulties.

1.2 Motivation

By 2100, it is predicted that there will be 11.2 billion people on Earth. Large amounts of food are necessary for this group to survive. However, because of the high costs, labor requirements, and time required for food production, traditional agriculture will not be able to meet this level of demand for food in the future. Also, the wastage of resources is significantly increasing due to the lack of knowledge about efficiently utilizing the available resources. Thus, the concept of smart agriculture is introduced [1].

Over the last twenty years, smart agriculture has been continuously studied. Modern IoT technology has improved farming practices [2]. Researchers have focused on several applications in smart agriculture, such as tracking the food supply chain [3], employing image sensors for crop monitoring [4], greenhouse agriculture [5], and open-field agriculture [6]. A few control objectives, such as the use of fertilizers and pesticides, have also been put into practice [7]. Apart from that, other technologies such as Information and Communication Technologies (ICT), unmanned aerial vehicles (UAVs), machine learning (ML), cloud computing, and artificial intelligence (AI) techniques have also played a crucial role in providing solutions to these critical issues of inadequate chemical application, poor irrigation systems, and yield prediction [8].

One way to describe agriculture would be as a set of services used to get the intended result. It is now difficult for a single service to satisfy the degree of expectations made by users. This leads to service composition (SC) which can be characterized as a collection of basic services. New composite services are obtained by combining various atomic services. These services could have similar functionality but differ in terms of Quality of service (QoS) attributes [9]. Many times, several candidate services make it difficult to 2abell QoS constraints. Thus, the task is to identify the most suitable service to ensure the composite service satisfies the user's functional and non-functional requirements [10].

Solving these complex composite services is difficult as they are non-deterministic polynomial-time hard (NP-hard) and cannot be resolved in the polynomial time domain. Thus, one solution is to apply nature-inspired meta-heuristics algorithms. They are showing immense potential as an effective substitute for traditional methods based on mathematical and dynamic programming. In reality, conventional approaches (which promise to discover the best solution) are frequently only practical for small-scale instances of the problems and may involve a significant amount of computational effort due to the great complexity and difficulty of optimization problems. On the other hand, metaheuristic-based algorithms may typically find better and even optimal solutions in less time when applied to real-life applications [11]-[12]. Since various services are combined, one objective cannot purely satisfy the user's requirements, thus, multiple conflicting objectives are formulated as a multi-objective optimization problem.

The research work provides the optimization of various services involved in the field of smart agriculture by considering time and cost as multiple conflicting objectives that need to be minimized. The novelty of the work lies in the fact that no work in the literature has been focussed on service composition optimization (SCO) in smart agriculture. Also, there are potential benefits of this research work in the lives of farmers such as remote farm management, optimized resource utilization, increased yield production with professional management, and optimized processes thereby increasing their income along with a contribution to food security. It would be more beneficial to the farmers/landowners who are unavailable on-site due to various job commitments.

1.3 Introduction

Among the most important sectors of the global economy is agriculture. It contributes significantly to developing economies like India, where it makes up 15% of the country's GDP. According to the figures of World Bank, the global employment share of the agriculture sector exceeds 25%. The prominence of the agriculture industry in context of employment is higher in emerging economies like India, where over two-thirds of the population depends on agriculture as a monetary resource, either directly or indirectly. It is responsible for over 40% of employment creation [13]. According to Food and Agriculture Organization (FAO) predictions, the world's population is expected to reach 9.73 billion in 2050, indicating a surge in food demand [14].

Figure 1.1: General representation of smart agriculture

However, using this structure for smart agriculture is fraught with difficulties. The various primary barriers to integrating technology in smart agriculture are shown in Figure 1.2. To successfully integrate the new IoT technology and realize the notion of smart agriculture, all these obstacles must be minimized.

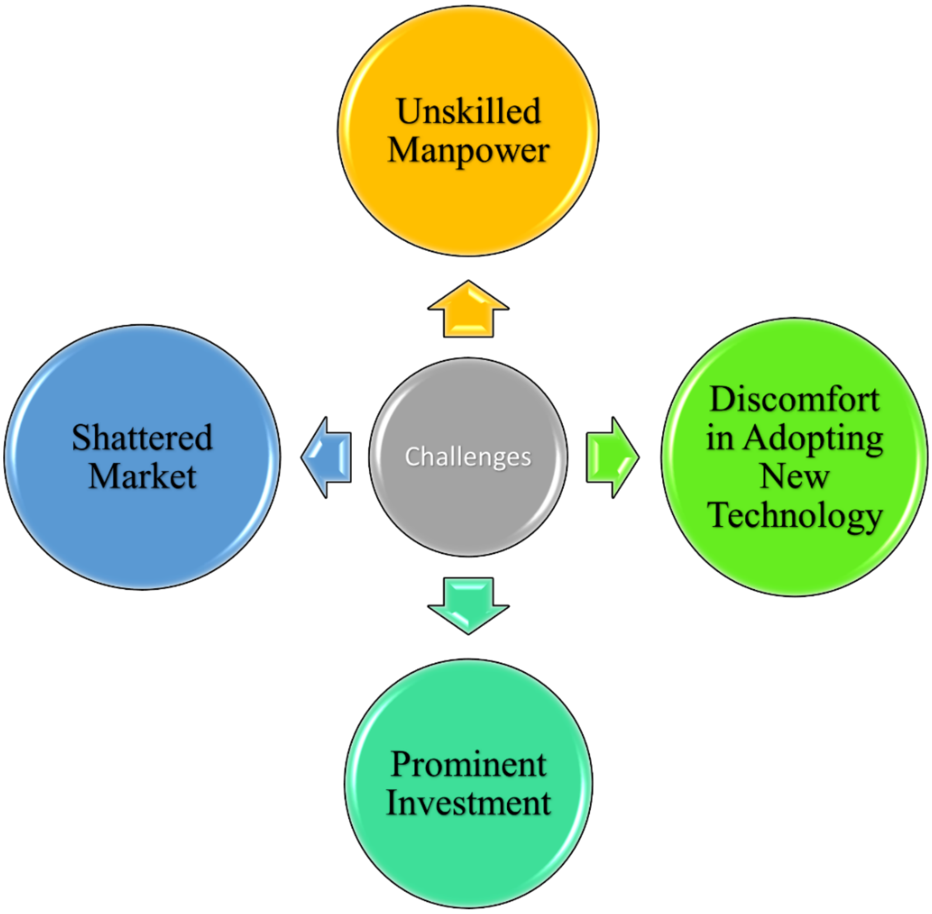


Figure 1.2: Obstacles in implementing smart agriculture

Agricultural logistics have been improved by the introduction of other technologies also such as Radio Frequency Identification (RFID), Wireless Sensor Networks (WSNs), Arduino UNO, Raspberry Pi (all involved in the physical layer of IoT), fog computing, big data, cloud computing, and artificial intelligence (all services in the service layer provided for application layer) [16]. Figure 1.3 illustrates a few of the applications of IoT in smart agriculture.

1.3.1 Service Composition

An IoT service is a decentralized structural unit that can be either atomic or composite. It functions as the digital representation of an object's actions. An atomic service is a selfcontained, well-defined behavioral unit that cannot be further subdivided into other services [17].

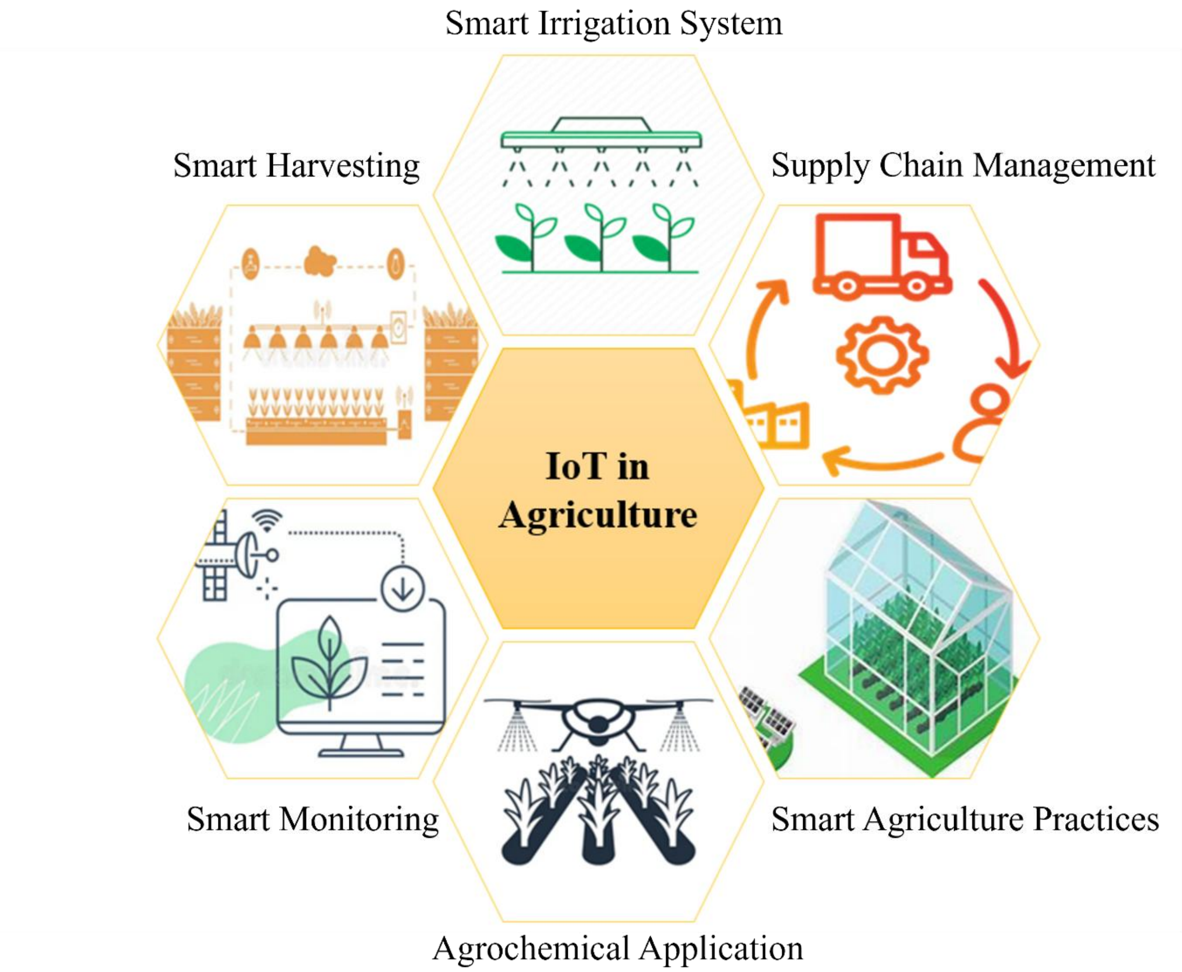


Figure 1.3: Applications of IoT in smart agriculture

On the other hand, a composite service is an advanced entity that combines numerous (atomic or composite) services to provide functionality and value. Can therefore readily handle the complex requirements of the user. For instance, an air conditioning composite can incorporate both a temperature and a humidity sensing service. All concrete/candidate services are interchangeable and functionally equivalent to each other [18]. These services can be combined and this process is called compositionality which is realized by a composition mechanism. Therefore, a things infrastructure, a concept of what a service is, and a choice of composing methods are required by an IoT system [19]. By taking into account two functional dimensions—control flow and data flow—the service composition method establishes a purposeful connection between services. Data flow describes how data is transferred between services whereas control flow describes the sequence in which communications take place [20].

A workflow, which can be hybrid, control-driven, or data-driven, is a set of distinct processes used to realize a computational activity. Tasks, actors, transitions, procedures, thorns, activities, and units are other names for phases in a control-driven workflow [18]. These steps can be carried out in branching, looping, sequencing, or parallelizing. When data becomes available, a data-driven workflow takes action without specifically defining any control flow components. Certain steps in a hybrid workflow are data-driven, and others are control-driven [21]. A generic workflow is shown in Figure 1.4 [18], which starts with task 1, decides whether to perform task 2 or task 3 based on branch conditioning, and then starts tasks 4 and 5 concurrently using parallel mode.

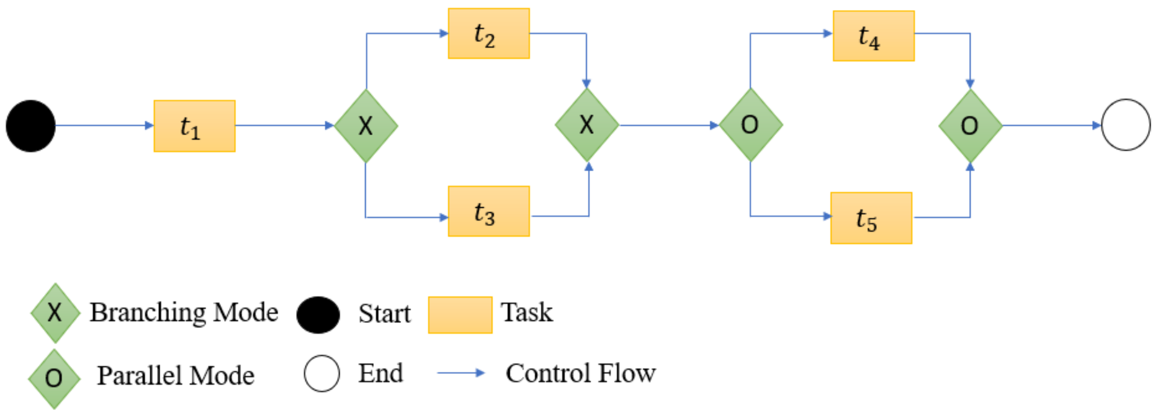


Figure 1.4: Generic workflow [18]

Workflows are crucial in systems because they blend services into intricate tasks that automate a particular context. For instance, in a smart home, a workflow that regulates a room's temperature in reaction to environmental changes can be automated. In the area of smart agriculture, a workflow can be set up concurrently to forecast diseases, assess data from harvest sensors, and take necessary action. In smart agriculture, this circumstance leads to the formation of the service composition problem.

In the context of smart agriculture, as a result of increased freedom and knowledge, farmers can now have some degree of control over their operations, including selecting crops that will produce the highest yields under the existing and anticipated climatic circumstances. The population's expectations have grown as a result of these breakthroughs in the use of artificial intelligence, leading to complex user demands in daily life. Meeting user's requirements can therefore frequently be challenging.

To satisfy user's those complex requirements, services are combined which is known as service composition. In other words, service composition can be defined as an aggregation of basic services. Service composition cannot be defined in a predetermined way. However, a range of non-functional attributes, sometimes known as QoS attributes, such as time, cost, availability, scalability, and dependability, are used to characterize those services. For instance, one might choose the fastest, least-priced service, or even the option that falls somewhere in the middle [22]. The QoS attributes are guaranteed by a contract between service providers and users, as indicated by the Service Level Agreement (SLA). To ascertain if a composite service can meet the SLA, consideration must be given to the 7abelled7es of the user's requests in atomic services [23].

Four steps are usually involved in creating QoS-based IoT services: plan composition, service discovery, QoS-based service selection, and service composition execution. An IoT application is generally composed of two stages. First, several action flows are used to combine the current classes, each of which contains a collection of atomic services, into a new service class. Second, the IoT application's components are selected from among the top candidate services from these classes. Both the data flow rules between candidate services and their order of invocation are shown in the composition plan. Following that, the service discovery phase chooses tasks from a group of services'with comparable functionality while taking QoS into account. The service selection step follows, during which the user selects the required services based on their needs. Services are finally composited by considering techniques that use global optimization or local selection. Figure 1.5 illustrates an instance of service discovery and service selection while considering time and cost as QoS factors [24]. Two services and the three candidate services that accompany them have been taken in this specific instance. The goal functions are assumed to be time and cost minimization. To achieve service composition, the candidate service with the lowest time and cost for both services concurrently was selected during the service selection step.

Any service composition problem's process can be defined using one of four possible architectural patterns: conditional, parallel, loop, and sequence. For every architectural pattern, a unique QoS composition rule is established as shown in Table 1.1. For example, the highest response time indicates the response time of a parallel composition consisting of more than one service. The total time it takes for all services to respond when they are called sequentially is known as the global response time. Each service is called with a probability p_i in the case of a conditional pattern, and the response time is an average depending on these probabilities.

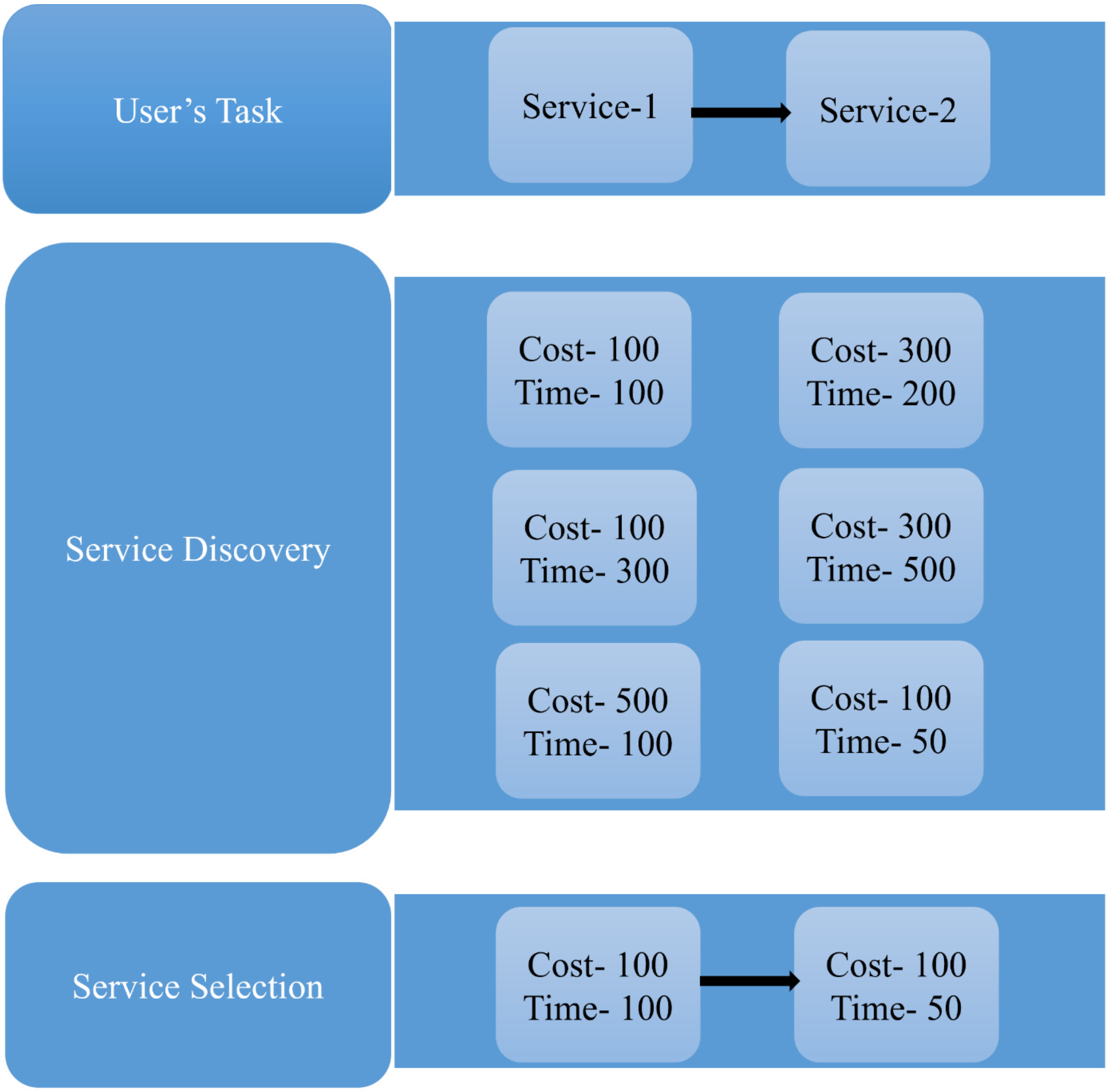


Figure 1.5: An instance of service discovery and service selection [24]

The response time will be multiplied by the number of loop cycles in a loop structure. A few of the QoS attributes have corresponding rules, tabulated in Table 1.1 [25] where t_i defines the response time, r_i is the reliability, a_i specifies the availability and c_i denotes the cost of i^{th}

service. The other factor k defines the number of loop cycles and p_i is the probability with which each service is called.

Table 1.1: QoS composition operators [25]

QoS attribute	**Sequence (m serial services)**	**Parallel (n parallel services)**	**Loop**	**Condition**
Response time	$\sum_{i=1}^{m} t_i$	$max\ \{t_i\}$	$k.t$	$\sum_{i=1}^{m} p_i \,.\, t_i$
Reliability	$\prod_{i=1}^{m} r_i$	$\prod_{i=1}^{n} r_i$	r^k	$\sum_{i=1}^{n} p_i \,.\, r_i$
Availability	$\prod_{i=1}^{m} a_i$	$\prod_{i=1}^{n} a_i$	a^k	$\sum_{i=1}^{n} p_i \,.\, a_i$
Cost	$\sum_{i=1}^{m} c_i$	$\sum_{i=1}^{n} c_i$	$k.c$	$\sum_{i=1}^{n} p_i \,.\, c_i$

It is imperative to distinguish between an atomic service and a candidate service. It is hypothesized that for every atomic service, multiple candidate services exist. For example, separate reservation services might be utilized for the same flight. Different QoS attributes are used to characterize each atomic service. Thus, it's critical to understand which candidate service is chosen to apply an atomic service. This optimization problem is thus, a combinatorial multi-objective optimization problem. Locating the optimal service composition is an NP-hard problem. This indicates that, except for the really basic situations (a small number of atomic and candidate services), an exhaustive search method is not feasible. Hence, evolutionary computational (EC) approaches or meta-heuristic algorithms are used to provide optimal or near-optimal solutions.

1.3.2 Multi-objective Optimization Problem

The service composition problem integrates several services acknowledging the user's preferences and different QoS criteria. A single objective is unable to satisfy the needs of several users at once due to the numerous services involved and the number of requests they have. Thus, it is possible to characterize this problem as a multi-objective optimization problem. This type of problem aims to find a set of optimal solutions that further provide a trade-off among multiple objectives. A multi-objective problem can be either a maximization or minimization problem, depending upon the user's requirement.

Commonly, a multi-objective problem consists of many objectives and several constraints that can be formulated as in equation 1.1 which is as follows:

$$f(x) = ((f_1(x), f_2(x), \dots, f_m(x))^T \tag{1.1}$$

where $m = 1, 2, 3, \dots, M$

subject to

$$h_l(x) \leq 0, \qquad l = 1, 2, 3, develop, \mathrm{L} \tag{1.2}$$

$$g_k(x) = 0, \qquad k = 1, 2, 3, develop, K \tag{1.3}$$

Equations 1.2 and 1.3 define the inequality and equality constraints, respectively.

Here, $f_m(x)$ is the m^{th} objective function

x is the decision variable representing the solution

k are the equality constraints

l are the inequality constraints

There are two categories of multi-objective problems: Preference-based and Ideal. While Pareto-based approaches are frequently utilized to solve ideal problems, scalarization-based approaches are typically used to solve preference-based problems. Figures 1.6 and 1.7 show the preference-based and ideal multi-objectives, respectively [106].

1.3.2.1 Scalarization-based Approach

A technique known as scalarization can be used to reduce a multi-objective problem to a single-objective problem. The global evaluation function, often known as the “fitness”, “utility”, or “objective function”, is a crucial component of this method. This function assigns a score to every solution, enabling the determination of which solution is superior to the others. Fitness functions can be organized into two categories: Weighted sum-based and fraction-based [25].

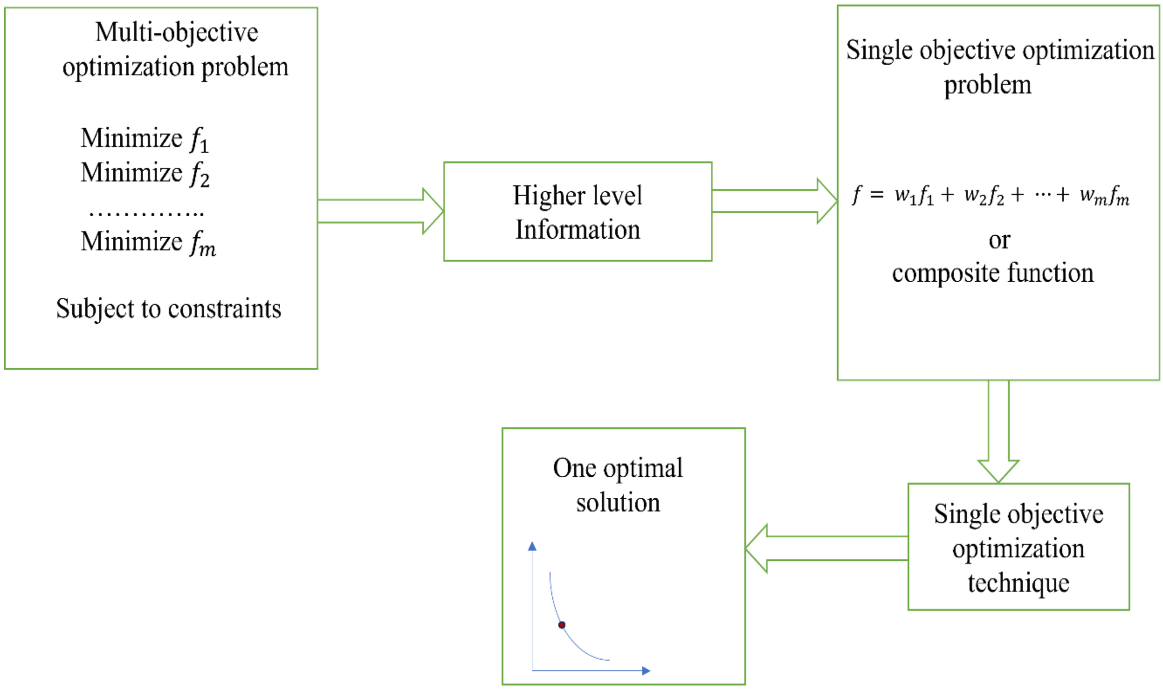


Figure 1.6: Preference-based multi-objective [106]

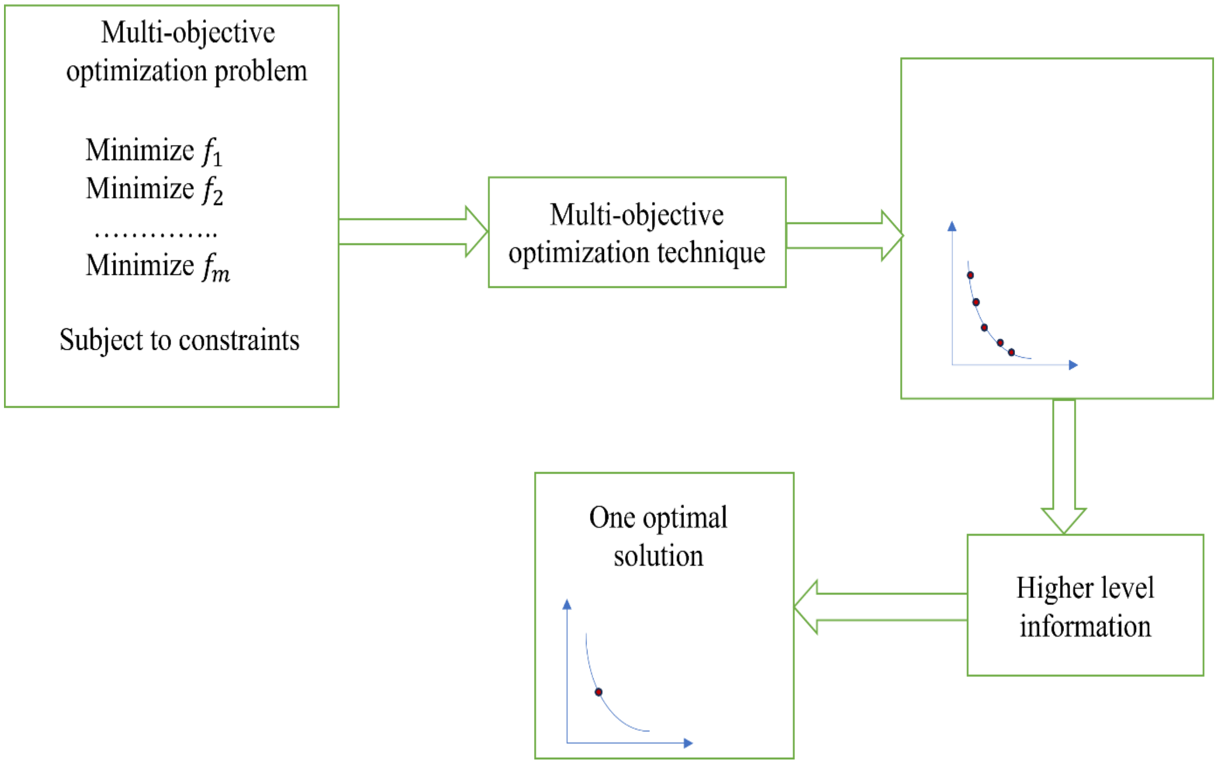


Figure 1.7: Ideal multi-objective [106]

For instance, in fraction-based, the fitness function can be outlined as in equation 1.4 given below

$$f(x) = \frac{w_1 * Cost\ (x)}{w_2 * Reliability\ (x) + w_3 * Availability\ (x)} \tag{1.4}$$

In weighted sum, the fitness function can be outlined as given in equation 1.5

$$f(x) = w_1 * Cost\ (x) + w_2 * Reliability\ (x) + w_3 * \ Availability\ (x) \tag{1.5}$$

In both equations 1.4 and 1.5, w_i defines weights associated with each attribute and $i = \{1, 2, 3\}$.

Scalarization techniques establish a relation between possible solutions, calculating the convex fusion of objective functions. Scalarization-based approaches have a drawback in the form of the aggregation function, as weighted sums do not guarantee user priorities and there is no standard way for calculating weights. Additionally, it lacks a criterion to verify non-dominance in the final solution produced by the single-objective algorithm. The weighted sum approach has several drawbacks, including subjectivity, Pareto Front convexity [26], differences between objective function shapes, and the number of solutions. Weighted sum aggregation is only appropriate for convex problems, while Pareto-based approaches can approximate the Pareto Front for both non-convex and convex problems. Additionally, weighted sum aggregation is inappropriate for functions with different shapes, and scalarization approaches returns only one solution per run. Ultimately, the diversity of solutions is lost when using a scalarization-based method.

1.3.2.2 Pareto-based Approach

Multiple objective functions are simultaneously optimized in the majority of real-world issues. These roles typically include competing and in conflict goals. When there are conflicting objective functions in multi-objective optimization, there exist several optimal solutions rather than just one. In this case, no approach can be deemed superior to any other in terms of achieving every goal. Pareto-optimal solutions are those that are the best available. Let us suppose a multi-objective optimization problem has two solutions, $x_1$and, $x_2$which can either dominate or not. In minimization problem, a solution x_1 dominates x_2 if certain conditions are met. Non-dominated solutions within the search space are called Pareto-optimal and form the

Pareto-optimal set or Pareto front. These solutions cannot be improved without worsening another objective. Thus, the set of viable non-dominated solutions is known as the Pareto-optimal set [27].

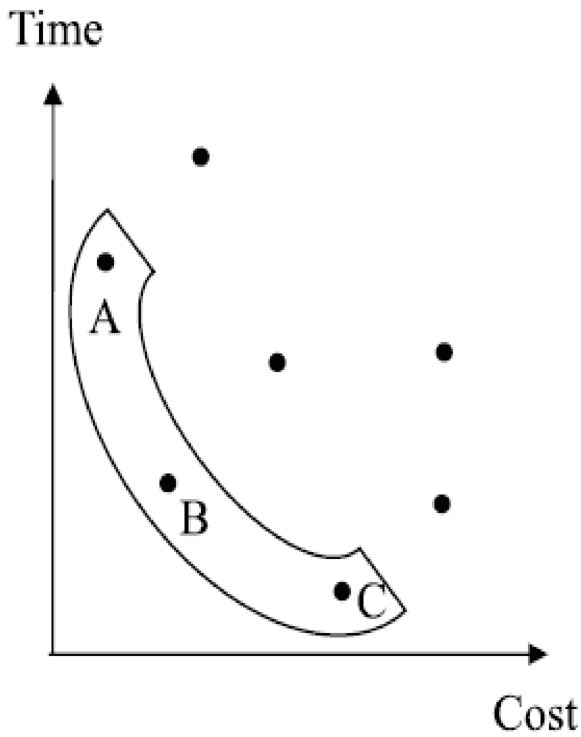


Figure 1.8: An instance of the Pareto front obtained in a multi-objective optimization problem [28]

As an illustration, Figure 1.8 displays candidates in a two-dimensional objective space while taking time and cost into account as QoS attributes [28]. The possibilities A, B, and C that are clustered together in the set stand for the non-dominated, or Pareto front, trade-off solutions [28].

1.4 Optimization using Meta-heuristic Algorithms

The modern age of information technology is causing numerous optimization problems in fields like bioinformatics, computer vision, big data analytics, and IoT. However, most problems are NP-hard and cannot be handled in a polynomial time domain. Therefore, precise mathematical methods can only be used in small-scale instances. Instead of losing up, the researchers considered using potential approximation techniques that could identify a workable solution in the allotted amount of time. Based on the randomization method, these algorithms can be classified into heuristics and meta-heuristics. Heuristic algorithms and meta-heuristics differ significantly in that the former is more dependent on the specific task at hand. These algorithms are limited to solving certain particular problems. By contrast, meta-heuristic algorithms apply to nearly all optimization problems since they employ the so-called "black box" optimizer [29]. A meta-heuristic is a process for locating, creating, or choosing an imperfect search algorithm to offer a sufficiently excellent solution to an optimization problem, especially when the knowledge is insufficient. These algorithms ensure optimal results since they explore the whole search space through successive generations of advancement. They offer intriguing benefits over standard methods, such as locating good solutions with less computing work and progressing swiftly toward extremely good solutions. As a result, they provide an incredibly effective means of handling complex, large-scale problems [30]. On the whole, meta-heuristics can be viewed as a category of cognitive self-learning algorithms that imitate intelligent processes and behaviors found in thinking, sociology, nature, and other fields to find close to optimal solutions to challenging optimization problems. These nature-inspired meta-heuristic algorithms can be classified into various groups naming evolutionary-based algorithms, bio-inspired algorithms, swarm intelligence-based algorithms, physics-based algorithms human-inspired algorithms, and miscellaneous algorithms, and are illustrated in Figure 1.9 [31].

a) Evolutionary Algorithms – The ideas of Darwin's theory of natural selection, which is predicated on the survival of the fittest in a particular environment, serve as the basis for evolution-based algorithms. These algorithms begin with an initial collection of populations, and as a result, a search process is carried out across a number of iterations until the finest practical answer is found. Examples are genetic algorithm (GA) [32], granular agent algorithm [33], bio-geography-based algorithm [34] etc.
b) Bio-inspired Algorithms – These algorithms are focused on distributed, decentralized, self-organizing, and flexible intelligence observed in biological systems. Examples are bacteria foraging optimization [35], artificial immune system optimization [36], artificial humming bird [37] etc.
c) Swarm intelligence-based Algorithms –Social insect or animal behavior are the sources of inspiration for swarm intelligence approaches. In it, each person possesses its behavior and intelligence, but the combination of individuals are given greater authority to tackle challenging issues. Examples are fish

swarm optimization [38], artificial bee colony optimization [39], dragonfly optimization [40] etc.

d) Physics-based Algorithms – These algorithms are based on physics and motivated by the laws regulating a natural phenomenon such as the law of gravity, thermodynamics, electromagnetism etc. Examples are simulated annealing [41], sine cosine algorithm [42], water cycle algorithm [43] etc.

e) Human-inspired Algorithms – These algorithms take inspiration from humans. Every person engages in non-physical activities like mind activities and physical activities that impact his performance which forms the basis of these algorithms. Examples are teaching-learning optimization [44], brain storm optimization [45], league championship optimization [46] etc.

f) Miscellaneous Algorithms – Those algorithms which cannot be classified in a particular group are put together in miscellaneous algorithms. For example, queuing search optimization [47], chemical reaction-inspired optimization [48] etc.

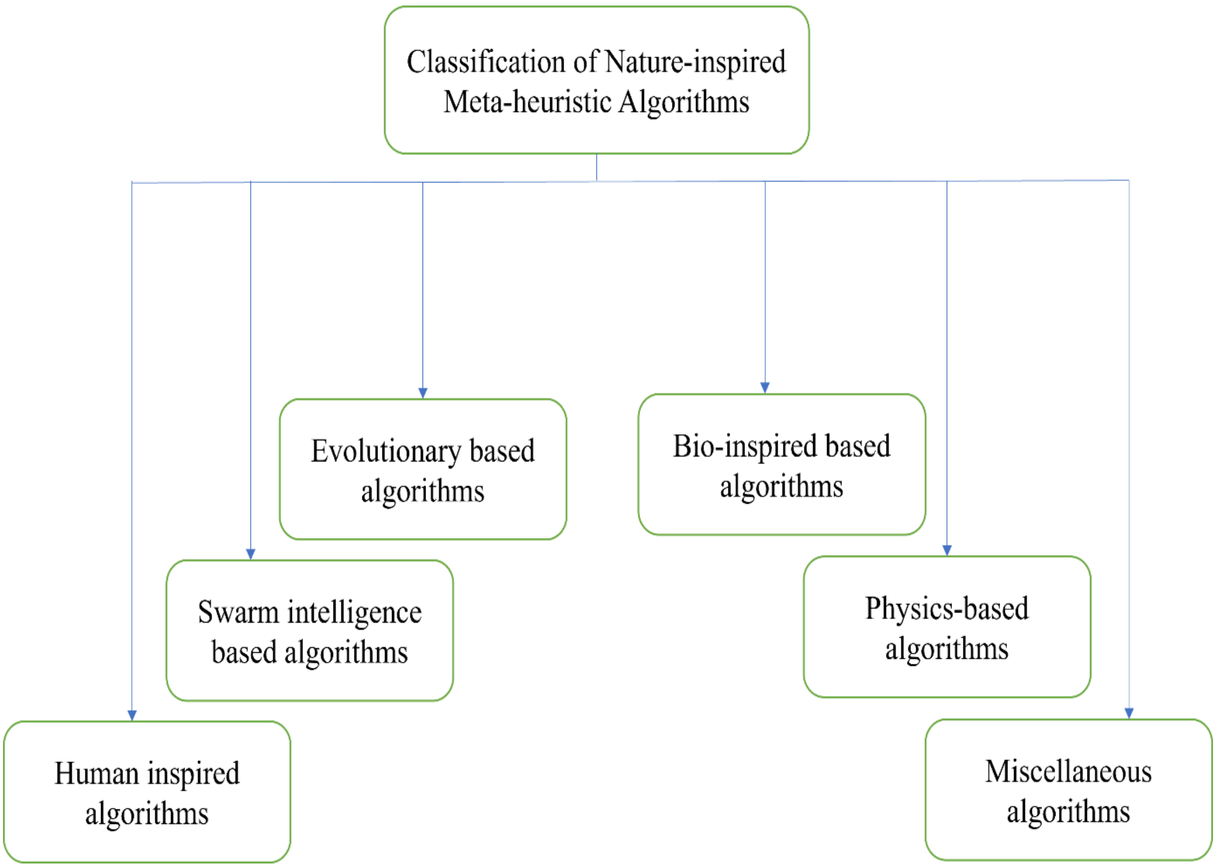


Figure 1.9: Classification of nature-inspired meta-heuristic algorithms [31]

1.5 Research Gaps

The following research gaps have been found after a careful examination of the literature.

a) Lack of research on service composition optimization in smart agriculture, despite its application in other fields.
b) The concept of ideal multi-objective optimization is barely used for service composition optimization problems in distinct applications.

1.6 Proposed IoT-based Framework

To enforce the service composition optimization in smart agriculture, Figure 1.10 illustrates the proposed IoT-based framework for the same. The sensor, network, cloud, service composition, and application (user interface) layers are its five layers.

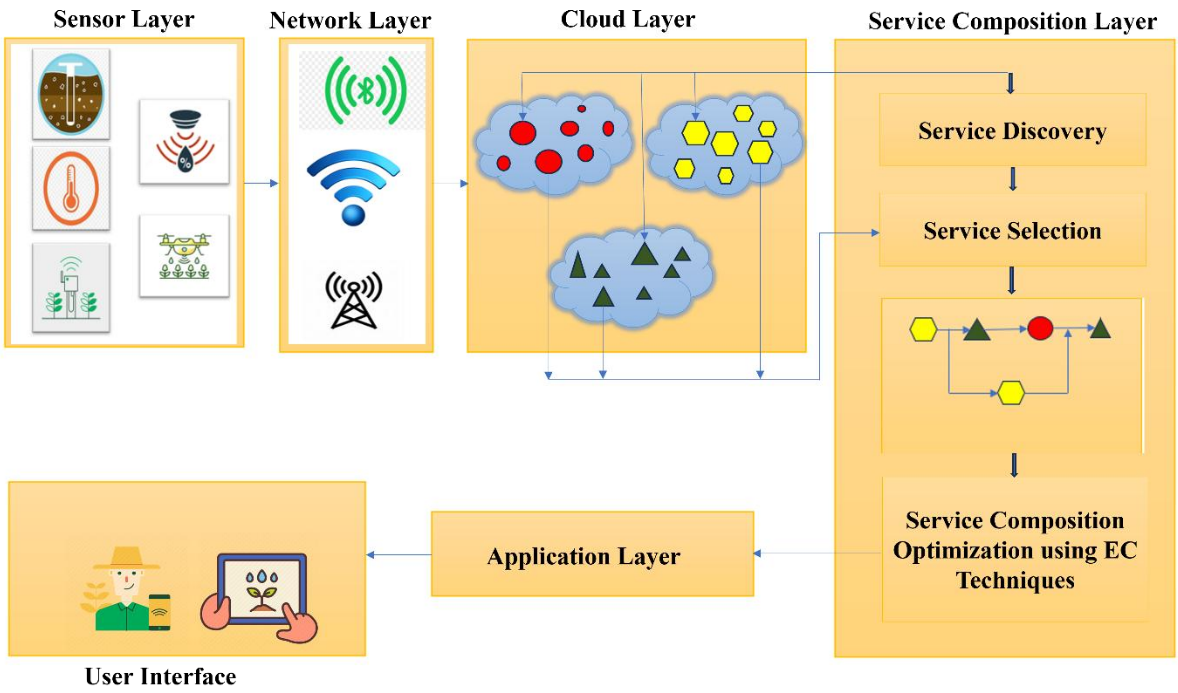


Figure 1.10: Proposed IoT-based framework for service composition optimization

a) Sensor Layer: This layer is in charge of gathering information from a variety of IoT sensors, including cameras, motion sensors, temperature sensors, and moisture sensors in the soil.

b) Network Layer: This layer establishes a communication channel between the servers and the data gathered from sensors. For instance, Wi-Fi (Wireless Fidelity), Bluetooth, Zigbee, LoRa (Long Range), and LoW-PAN (Low-power Wireless Personal Area Network).

c) Cloud Layer: This layer provides a range of sub-services across several private, public, or hybrid clouds and acts as virtual storage. There are three options: Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS). In our study, a sequential workflow for fourteen services relevant to apple orchard establishment is considered which can be taken from the cloud layer as it stores the data.

d) Service Composition Layer: The fourth and most crucial layer of the framework is the service composition layer. To satisfy the user's complex requirements, it is divided into multiple sub-services. First, cloud services are identified, next the necessary services are chosen among the available cloud options, and lastly, services are composed. To optimize the composite services according to user demands, this layer is linked with optimization algorithms in our work.

e) Application Layer: The application layer is necessary to provide end users with access To the services that were developed in the preceding step.

Thus, the research work in this thesis explores the service composition layer in smart agriculture, by first analyzing data, then combining relevant services, and optimizing them using EC techniques to get the optimal composition plan for its users.

1.7 Objectives of Research Work

Optimization of service composition has been the subject of extensive investigation. However, its application in smart agriculture remains unexplored. Therefore, this work employs distinct meta-heuristic approaches to achieve optimization in this domain. Motivated by this gap, the following objectives have been framed for this research.

a) Linear multi-objective service composition optimization in smart agriculture using EC techniques such as

- Multi-objective Genetic algorithm (MOGA)
- Non-dominated Sorting Genetic Algorithm II (NSGA-II)

- ➢ Multi-objective Gaining Sharing Knowledge based algorithm (MOGSK)

b) Non-linear multi-objective service composition optimization in smart agriculture using EC techniques such as

- ➢ Multi-objective Genetic algorithm (MOGA)
- ➢ Non-dominated Sorting Genetic Algorithm II (NSGA-II)
- ➢ Multi-objective Gaining Sharing Knowledge based algorithm (MOGSK)

c) Impact of uncertainties on both linear and non-linear service composition optimization in smart agriculture using fuzzy inference system (FIS).

d) To develop a novel Multi-objective Electric Eel Foraging Optimization (MO-EEFO) algorithm for real-world optimization problems.

1.8 Organization of Thesis

The thesis is classified into seven chapters. The detailed description is given below.

Chapter 1 describes the service composition problem and how it can be solved using various meta-heuristic algorithms.

Chapter 2 covers insights into the work done in the field of smart agriculture. It also includes various single and multi-objective optimizations done in smart agriculture using distinct meta-heuristic algorithms.

Chapter 3 discusses the first objective of the thesis i.e. multi-objective optimization of composited services by establishing a linear relationship between the two objectives by using MOGA, NSGA-II, and MOGSK algorithms.

Chapter 4 addresses the second objective i.e. multi-objective optimization of composited services by establishing a non-linear relationship between the two objectives by using MOGA, NSGA-II, and MOGSK algorithms.

Chapter 5 deals with analyzing the impact of uncertainties on both linear and non-linear multi-objective service composition optimization by using a FIS.

Chapter 6 proposed a novel nature-inspired MO-EEFO algorithm for solving real-world applications.

Chapter 7 concludes the thesis along with the clarification of future work.

1.9 Summary

This chapter provides an extensive understanding of the service composition problem and how it relates to real-world smart agricultural challenges. It illustrates the successful implementation of several well-established EC approaches that are available to address these challenges. Various methods for tackling multi-objective optimization problems, such as Pareto and scalarization-based methods, are discussed as the problem can be formulated as one. Furthermore, a detailed discussion of the identified research gaps and proposed IoT-based framework along with the dataset as well as the study's objectives is provided to conclude the chapter.

CHAPTER-2
LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1 Chapter Overview

QoS-based service composition optimization plays a crucial role in satisfying the user's complex needs when multiple services with comparable capabilities exist but have distinct QoS metrics. Given that the problem is NP-hard, meta-heuristics are frequently helpful in identifying the optimal solution while adhering to the imposed global constraints, which satisfies the complicated needs of the user. Various researchers have focussed on this idea in a variety of domains using IoT, artificial neural networks (ANNs), cloud computing, and ML. Applications covered in the literature include traveling salesman problems, smart healthcare, supply chain management, and many more.

This chapter initially covers all the reviews and surveys done in the field of smart agriculture, followed by diversified optimization methods used for handling various agricultural-related issues. Eventually, a literature table is provided that compares existing literature on service composition with the research gap found for the study of this thesis.

2.2 Review Papers on QoS-based Optimization in Smart Agriculture

P.P. Ray et al. [49] have provided a review of the various IoT-based agricultural applications that offer guidance for further agricultural research in agriculture. They have given a thorough overview of the various communication technologies used in agriculture, including Bluetooth, LoRa, Arduino modules, WiMax (Worldwide Interoperability for Microwave Access), and 802.11 (Wi-fi). The kinds of cloud services offered by a few IoT-based cloud platforms, as well as their costs, times, data visualization capabilities, and real-time data collection, were also compared. In addition, the sensors, cloud support, and application types of the most popular IoT sensory systems were compared. The authors came to several important conclusions about topics for further study, including fish farming, data analytics cost optimization, and smart irrigation systems. However, the comprehensive literature review of IoT in advanced agriculture was not the main focus of this paper.

After conducting a thorough literature study, A. Khanna et al. [50] have characterized IoT as an actual paradigm shift in precision agriculture. It has described every communication technology that might be utilized with the IoT as well as the different IoT applications that were especially related to precision farming. The barriers in this area are described as data privacy, interoperability, scalability, virtualization, reliability, mobility, and availability. They have also thought that the main concerns for further study should be the right deployment of sensors, service composition, cost, and discovery.

Another comprehensive evaluation of IoT applications in smart agriculture can be found in Wen Tao et al. [51]. The challenges encountered along with the usage of IoT sensors and other communication methods in agriculture are analyzed in-depth. The authors have summarized that three main issues that need to be addressed are cost, data reliability, and IoT device standardization.

A review, by A. Srivastava et al. [52], has explored how DevelopoT technology develops helping farmers overcome many of their challenges in the agricultural sector. However, it also explains that to effectively apply technology to improve agriculture, problems like equipment cost, data security, IoT node power savings, fault tolerance, and data privacy must be resolved.

V.P. Kour et al. [53] have given a summary on the expansion of the growth of IoT in smart agriculture and conclude that building solutions that are both power – and cost – optimized presents substantial problems that need to be overcome.

An overview of the application and effects of IoT based on cloud in climate-smart agriculture is provided by E.G. Symeonaki et al. [54]. A few applications are described in detail, such as cloud agro-systems and cloud services based on the PDCA (plando-check-act) cycle of agriculture. The authors discovered that although these technologies have many benefits, there is still a lack of integration in the experimental phase. The main issues that need to be resolved include farmer training centers, inexpensive network coverage, user-friendliness, and appropriate standardization for IoT devices.

B. Sinha et al. [55] have provided a review on how to work with IoT to elevate productivity and optimization of costs in smart agriculture. Precision farming, livestock monitoring, crop management, irrigation management, etc. are the important aspects of IoT in smart agriculture. They have also provided a comprehensive description of sensors like temperature sensors, soil moisture sensors, potential of hydrogen (pH) sensors, ultraviolet (UV) sensors, etc. The authors concluded their work by considering security, scalability, dependability, and resource optimization as the biggest issues that need to be tackled in the future.

In their review, Saiz-Rubio et al. [56] have discussed how data-driven management, sometimes known as “Agriculture 5.0,” might be applied to sustainable agriculture to save costs while protecting the environment. The authors have talked about the idea of “Agriculture 5.0,” which is essentially the application of robotics and artificial intelligence combined with unmanned machinery and autonomous decision-making systems.

A thorough analysis of bio-inspired algorithms for agriculture has been given by C. Maraveas et al. [57], who divided them into four categories: multi-objective, evolutionary, ecology, and swarm intelligence-based techniques. The finest algorithms for agricultural yield, land planning, pest management, and fertilizer optimization, according to their description, are GA, ant colony optimization (ACO), firefly, and cuckoo algorithms. Particle swarm optimization (PSO) is the most appropriate algorithm for predicting irrigation, though. It has also been noted that compared to single-objective approaches, multi-objective approaches yield a greater number of nearly optimum solutions. The paper concludes that while hybrid strategies have received limited attention, bio-inspired artificial neural networks outperform other algorithms in the field. No algorithm can perform every type of function.

Using meta-heuristics, Masdari et al. [58] have presented a thorough review of the literature on QoS-based service composition. To tackle the web service creation challenge, they categorized the literature into seventeen different meta-heuristics and compared each one with certain meta-heuristic qualities. The authors conclude that, after PSO, GA is the most frequently utilized technique for solving service composition problems. The majority of the evaluation was covered by fitness value parameters, then time-related parameters. Numerous articles use the QWS dataset, followed by random datasets for web service composition.

As an application of AI in agriculture, M. Pathan et al. [59] have covered precision agriculture, crop phenotyping, and disease identification utilizing deep learning, ML, ANNs, WSNs, IoT, fuzzy logic, and GA. They concluded that it can produce high productivity at low labor and cost costs and lower environmental risk.

S. Qazi et al. have contributed to an overview of the use of AI and IoT technology in smart agriculture, accompanied by a few predictions for subsequent generations [60]. They give instances of a few smart irrigation methods based on IoT, such as the usage of neural networks, fuzzy logic (FL), UAVs, and soil-based methods like drip irrigation and aeroponics. The authors also discuss pest-weed identification, phenotyping, and plant disease prognosis using deep learning. The authors conclude by listing a few challenges that still need to be met, such as the international consortium for the development of coherent wireless sensing systems, cyberattacks, and the ever-increasing cost of technology.

A. De et al. [61] have emphasized in their study of fuzzy implementations in the agri-supply chain how important it is to focus on the entire agri-supply chain as opposed to just enhancing agri-production. The eight primary challenges that are recognized include land appropriateness, irrigation, production practices, transportation, insufficient cold storage, drought management, waste management, environmental concerns, and sustainability. It is mentioned that the problems that have not yet gotten enough attention are waste management, transportation, inadequate cold storage, and drought management. Furthermore, real-time applications require the study of geographic information systems (GIS) and big data.

F. Valdez et al. [62] have given a survey on the use of FL with nature-inspired approaches to solving difficult optimization issues. This article covers the three most crucial methods: gravitational search algorithm (GSA), PSO, and ACO. According to the authors, using optimization techniques in conjunction with FL yielded better results than using optimization algorithms alone.

Smart farm management applications of ML are demonstrated by A. Sharma et al. [63]. They have clarified that while regression techniques are better for predicting the weather, crop production, and soil qualities, deep learning algorithms including decision trees, random forests, convolutional neural networks, and support vector machines are good for identifying plant diseases. Drones, robotics, intelligent harvesting, and irrigation systems are all essential for reducing the need for human labor. To make this industry more sustainable, they have mentioned chatbots based on natural language processing (NLP) and hybrid algorithms in their paper analysis conclusion.

The power and promise of computer technologies employed in agriculture, namely ML and IoT data interpretation, have been shown by R. Akhter et al. [64]. A prognostic model for the Scab apple disease has also been suggested for apple farms in the valley of Kashmir region. They asked farmers about the newest agricultural technologies and how they affected yield output to further elucidate the survey.

A survey of big data applications in smart farming is given by S. Wolfert et al. [65]. They claimed that its reach is impacting every link in the food supply chain and offering farming predictions. On top of that, the significant growth in IoT gadgets is producing a large amount of diverse data that can be captured, examined, and utilized in decision-making processes through the implementation of big data. The authors draw a continuum between two extreme scenarios for the future of smart farming: closed proprietary systems and open collaborative systems. Some other topics, like security, openness of platforms, privacy of data, and intelligent analytics, have also been covered.

2.3 QoS-based Optimization in Smart Agriculture

Ocampo et al. [66] have provided a study that uses GA to reduce the energy cost of two motor pumps in a smart farm, with the requirement that there be enough energy available for both pumps. Moreover, restrictions were implemented. Each solution is viewed as a set of weights that need to be multiplied by the sensor readings that correspond to it. Three mutation operators (Uniform, adaptive feasibility, gaussian), six crossover operators (Scattered, single point, two-point, intermediate, heuristic, and arithmetic), tournament selection, crossover probability = 50%, and population size variation between 50 and 500 with a spacing of ten are all included. After testing several settings, the authors conclude that several simulations are needed to find the optimal solution. The paper's conclusion is ambiguous because neither trade-off points nor a specific optimal solution have been taken into account.

Hakli et al. [67] have presented a novel GA-based method for autonomous land partitioning. The goal function is defined as the product of three competing parameters: the location of cadastral parcels, the degree of cadastral parcels, and the fixed facilities multiplied by two. The block's unique number is utilized to start the random population. The simulation operators in the suggested model—population size = 20, number of generations = 50, roulette wheel selection method, single point crossover, swapping mutation, mutation probability = 0.1, and crossover probability = 0.8—are applied to a completed project of Alanozu by the authors. A comparison is made with another study in which the model took 4.8 hours to optimize a 3-

hectare block with six parcels, whereas the suggested technique takes just eight hours to optimize a 109-hectare block with eighteen blocks and thirty-three parcels. The authors demonstrate their accomplishment by contrasting the target function results with the identical land portioning carried out by the designer. They discovered that the suggested Automated land portioning genetic algorithm (ALP-GA) is significantly better.

Roy et al. [68] have presented a design for terrace gardening and outdoor spaces that uses GA to forecast rainfall based on actual data from Kolkata, West Bengal, India. If rainfall is not expected, a system based on sensors in terrace gardening determines whether soil moisture is below a predetermined point. If so, an Arduino UNO relay module and global system for mobile communications (GSM) module receive the signal, which activates the water pump until the soil sensor's threshold value is reached. In outdoor regions, the moisture sensor's signal is transmitted to a mobile device via an ESP8266 Wi-Fi module, which directs the UAV to disperse water where it is desired. Although the roulette wheel is selected, no information regarding crossover and mutation is given.

A GA-based UAV path planning method is proposed by Shivgan et al. [69] to minimize energy consumption by limiting the number of turns while covering a region. They run the experiment with waypoints = 10, 25, 50, and 100. The parameters are swapping mutation, two-point crossover, and tournament selection. The authors compare the optimal solutions with a greedy technique to assess the outcomes. According to the authors, the suggested GA uses two to five times less energy than the greedy method.

Through the optimization of the path coverage of 40 sensor nodes connected to greenhouses using the hop-to-hop delivery technique, Gaofeng [70] have illustrated the use of evolutionary algorithms for cost optimization. Thirty iterations in all were conducted, with the twentieth iteration yielding the best value of 3838 for the optimal path determination.

Use of meta-heuristics along with artificial intelligence like machine learning, deep learning is also taking smart agriculture to the next level.

Acharjya et al. [71] have presented a model for crop identification based on regression, the K-nearest neighbor (KNN) method, real coded genetic algorithm (RCGA), and hybridization of fuzzy rough sets. Using a fuzzy real set, redundant attributes are eliminated in the first step, after which the data is split into training, testing, and validation sections. Regression, KNN, and

RCGA are used in the analysis of training data. Six combinations are possible for this: Tournament with Laplace (TSLX), Roulette with Laplace (RWLX), Tournament with Simple (TSSX), Roulette with Simple (RWSX), Roulette with flat (RWFX), and Tournament with flat (TSFX). These combinations can be made using simple crossover, flat crossover, Laplace crossover, roulette wheel selection, and tournament selection. Using data from Tamil Nadu's Tiruvannamalai district's Krishi Vigyan Kendra, all of these combinations are compared for success rate, accuracy, and execution time with the goal function being the lowest mean squared error. The optimal combination among them is found to be fuzzy rough set roulette wheel selection with Laplace crossover which can be abbreviated as FRRWLX. For a variety of crops grown in the Tiruvannamalai district, the authors also compare their findings with five other methodologies and a rough set real coded based genetic algorithm with roulette wheel selection and Laplace crossover (RSRWLC). The conclusion of the paper states that the FRRWLX technique is the best of all of the others.

R.I. Mukhamediev et. al [72] have developnvestigated the use of flight planning for heterogeneous UAVs in monitoring and agrotechnical measure implementation to address coverage challenges. For multi-heterogeneous UAV coverage path planning, an approach based on GA called multi-heterogeneous UAVs coverage path planning with moving ground platform (mhCPPmp) is suggested. It offers flyby calculations, optimal UAVs subset selection, and a 10% cost savings over algorithms that do not take into account heterogeneous UAVs.

Farzad Kiani et. al [73] have suggested two evolutionary computational algorithms: Expanded Gray Wolf Optimization (Ex-GWO) and Incremental Gray Wolf Optimization (I-GWO) for 3D robot path planning. With a 55.56% success rate utilizing the Ex-GWO algorithm, the suggested methods effectively locate collision-free pathways for robots in large-scale farmlands while minimizing resource consumption and process costs.

For IoT-based smart agriculture applications, S. P. Singh et al. [74] have suggested a novel fitness function termed service cost that takes into account localization rate, lifetime, coverage rate, energy consumption, and delays utilizing IoT-based wireless sensor networks. When the results of the proposed extended differential evolution (DE) algorithm are compared to those of the whale optimization algorithm (WOA), PSO, GA, and firefly algorithm (FFA), it is discovered that the proposed algorithm produces better results.

H. Babazadeh et. al [75] have focused on maximizing agriculture output and water productivity in arid and semi-arid regions. They employ a simulated annealing method (SA) and MOGA based on experimental data from two conductive agricultural seasons in 2010 and 2011. The results demonstrate that MOGA is more capable of optimizing grain yield and water productivity at the same time.

To optimize the benefit-cost ratio and output energy for watermelon growing in Iran while limiting greenhouse gas emissions, S. Shamshirband et al. [76] employed MOGA. The findings indicate a simultaneous average drop of 33% in greenhouse gas emissions and 28% in energy intake.

Using data for the Tamil Nadu region of Coimbatore, N. Sivakumar et al. [77] have presented a model for minimizing the use and cost of fertilizers by utilizing the FFA. To ensure that crops meet the NPK (nitrogen, phosphorus, and potassium) requirement, they have applied two different types of fertilizers—Complex (STD-10 and STD-3) and Simple (Urea and SSP)—to eleven distinct regional crops.

In the Coimbatore, Tamil Nadu, area, N. Thilagavathi et al. [78] have worked on the optimized use of agricultural land utilizing social spider algorithm (SSA), ACO, and LINGO global server. They have taken into account that the goal function is to cultivate the right crops and crop combinations to minimize the need for water and optimize overall returns, or profit. Four situations are chosen. Every major crop (sugarcane, maize, cholam, three varieties of gingelly, paddy, cotton, and groundnut) in Scenarios 1 and 2 has a small – to medium-sized land area (twenty thousand to forty thousand sq. m) and a medium-sized land area (forty thousand to one lakh sq. m).

Bahram Saeidian et al. [79] have proposed an imperialist competitive algorithm (ICA) to maximize overall income for all lands by optimizing water allocation at the farm level utilizing temporal agriculture data. Compared to other algorithms such as PSO, bees algorithm (BA), and GA, the proposed algorithm was found to provide superior income.

Another smart agriculture system based on IoT, created by G. Sushanth et al. [80], makes decisions about plant watering based on temperature, moisture, and humidity readings. Moreover, a motion detector sensor employs an Arduino board to monitor animal activity in the field. The farmer receives updates via short message service (SMS) via Wi-Fi, third generation (3G), and fourth generation (4G). This work lacks the use of any optimization technique.

A wireless sensor-based system for crop irrigation has been proposed by J. Muangprathub et al. [81]. The three main components of this framework are mobile applications, web-based applications, and hardware. A hardware module is used to collect data from soil moisture sensors. A web-based application is then developed to modify the data obtained using data mining, and a mobile app is used to water the field manually or automatically. The actual experiment used vegetables grown at home and lime as the crops to be assessed. It was conducted in three different villages in the Makhamtia region of Thailand. The study showed that 72–81% and 29–32 degrees, respectively, are the ideal temperatures for producing a decent crop of homegrown veggies and limes, respectively. However, no concept of optimization was used.

A SmartFarmNet platform that is based on the IoT has been presented by Jayaraman et al. [82] for automated data collection from gadgets such as mobile phones, cameras, weather stations, and WSNs. This data is then correlated to verify crop performance and forecasts for any farm. The data has been stored in the cloud for later processing and outputs.

A hybrid model of machine learning incorporating a Butterfly optimization algorithm (BOA) with IoT has been presented by A. Gupta et al. [83] for crop yield optimization. The study has been broken down into three stages by the authors: pre-processing, feature selection (using the Variance Inflation Factor algorithm and correlation-based feature selection), and classification. A dual-layer model for classification is demonstrated: an extreme learning machine (ELM) method for crop yield prediction, and an adaptive K-nearest centroid neighbor classifier (aKNCN) model for estimating soil quality and subsequently classifying them into various classes. Metrics such as Mean Absolute Percent Error (MAPE), Mean Squared Logarithmic Error (MSLE), Accuracy, Mean Squared Error (MSE), Median Absolute Error (MedAE), EVS (Explained Variance Score), Root Mean Square Error (RMSE), Model Evaluation metric (MAE), and their contrast analysis with aKNC-GB (adaptive K-nearest centroid neighbor classifier – Gradient boost), aKNCN-ELM-BOA, aKNC-ANN, aKNC-RF (adaptive K-nearest centroid neighbor classifier – Random forest), aKNCN-ELM, and aKNC-SVM (adaptive K-nearest centroid neighbor classifier – Support vector machine) are taken into account when evaluating performance. The suggested approach, according to the authors, outperforms the others in every comparison of metrics. Nevertheless, its complexity, requirement for constant internet access, and vast training data set are its drawbacks.

2.4 Dealing Uncertainties in Smart Agriculture

Fuzzy bee colony optimization (FBCO), as developed by O. Castillo [84], is a widespread type-II fuzzy logic technique for adapting dynamic parameters of the Bee colony optimization (BCO) method for the optimum performance of water tank controller and mathematical functions. Nine fuzzy inference rules for FBCO have been taken into consideration for the Mamdani fuzzy system with a trapezoidal membership function. The two input variables are $diversity$, and $iteration$, and the two output variables are $alpha$ (α) and $beta$ (β), which have respective ranges of 0–1 and 2–5. Level $(high, okay, low)$ and rate $(positive, none, negative)$ are the input variables for the water tank controller in case of the primary benchmark problem, while the output variable is a valve with five membership functions of the triangle type $(openfast, openslow, nochange, closeslow, closefast)$. Then, fifteen experiments have been conducted for each of the ten mathematical functions. The Type-I fuzzy logic controller (T1FLC), original BCO, and an Interval Type-II fuzzy logic controller (IT2FLC) were compared with FBCO. The findings highlighted that FBCO executes better than the others in relation to convergence rate, and stability.

Drawing from the plant's innate defense mechanism, C. Caraveo et al. [85] have created a modified predatory pray optimization approach that uses Type-II fuzzy logic to preserve balance. By dynamically altering the variables, the autonomous robot's travel path has been modified to reduce errors. The Mamdani kind of fuzzy controller has been used when the input variables are angular velocity and linear velocity, and the output variables are left and right torques. Together with nine fuzzy inference rules, two different membership function types are used: trapezoidal for positive and negative terms and triangular for zero terms. By contrasting it with FBCO, its viability has been examined. Based on statistical analyses, the author's optimization approach and fuzzy logic system (FLS) have significantly improved performance and stability.

M. Guerrero et al. [86] created a "uzzy'control system that would continuously change the parameters—the probability of discovering host bird (P_a) and scale factor (β) to improve the convergence rate. This system is known as the fuzzy cuckoo search algorithm (FCS). Five benchmark functions with various dimensions ranging from eight to one hundred and ttwenty-eight—Griewank, Rastringin, Ackley, Spherical, Rosenbrock—have been used to test the suggested technique. The Mamdani fuzzy system type, comprising three fuzzy rules and triangular membership functions 30abelled as $high, medium\ and\ low$, has been applied to a single input (iterations) and output (P_a or β). The research concludes with a comparison between FCS (P_a) and FCS (β), and cuckoo search, showing that FCS (β) exceeds the performance when compared with other two algorithms for four out of five functions when the number of dimensions increases.

A unique method for dynamically modifying parameters ($\alpha\ and\ kbest$) in Fuzzy gravitational search algorithm (FGSA), which is based on interval Type-II fuzzy logic, has been presented by F. Olivas et al. [87]. To test it, they first optimized fifteen key mathematical benchmark functions, and then they worked on a fuzzy controller that regulates the temperature of hot and cold water. In the process of optimizing mathematical functions, $iterations$, which range from 0 to 1, and $diversity$ ($high, medium\ and\ low$) are input variables. α, which spans from 0 to 100, and $kbest$, which spans from 0 to 1, are taken into account as output variables. There are nine fuzzy inference rules for a fuzzy controller; its inputs are $flow\ and\ temperature$, and its outputs are $hot\ and\ cold$. Nine fuzzy inference rules make up the fuzzy controller's inputs ($temperature\ and\ flow$) and outputs ($hot\ and\ cold$). Its efficacy is further confirmed by comparison with the Type-I Fuzzy GSA for altered parameters (T1FGSA) and the original GSA. The suggested algorithm, according to the authors, performs better for local or global searches than the other two nearby algorithms.

To ensure that farms only utilize the appropriate number of fertilizers, G. Lavanya et al. [88] have developed a revolutionary NPK sensor that is outfitted with an LED (light emitting diode) and an LDR (light dependent resistor). This sensor allows for thorough monitoring of the nutrients present in the soil. IoT is utilized to transmit data to Google Cloud for speedy information retrieval, while fuzzy system is used to apply the Mamdani inference model to identify vitamin deficiencies in sensed data. When defining IF – THEN rules, output levels are categorized as $very\ high, high, medium, low\ and\ very\ low$by using ranges of 0.8-1, 0.5-0.8, 0.3-0.5, 0.1-0.3, and 0-0.1, respectively. Its efficacy is evaluated with a software and hardware model. Three test samples of red, mountain, and desert soil were collected for the hardware testing. Data is sent from NPK sensors to the cloud servers for software simulations while taking metrics like jitter, throughput, and end-to-end delay into account. The authors stated in conclusion that their approach, when used with a smart, low-cost, and accurate IoT system, produces high crop production.

In order to maximize water resources, Cruz et al. [89] have suggested using a fuzzy logic-based decision assistant tool for the water tank monitoring and control subsystem (WTMCS) in a smart farm automatic irrigation system (SFAIS). The water tank's state determines how much priority the power management system has when it comes to turning on the pump. Priority levels have been determined by keeping an eye on the $water\ level\ (L)$ and its $variations\ in\ water\ level\ (DL)$. While values of L are fuzzified as $full\ (F), normal\ (N)\ and\ empty$, values of (DL) are defined as $high\ (HP), medium\ (MP) and\ low\ (LP)$. The three priority levels are defuzzed. With the defuzzification method of center of gravity, they have defined nine fuzzy IF – THEN rules for making decisions in order to establish the relationship between input and output variables. The authors conclude that WTMCS is more likely to supply the farm with the best possible distribution of power and water resources.

A method based on fuzzy logic has been developd by R.P. Sharma et al. [90] to prevent pests in a millet and rice field by persistently monitoring the expansion of pests. Temperature, humidity, and rainfall data samples were collected in real-time by the suggested system using an IoT monitoring mechanism, which produced a data collection. GA has utilized this data as training to refine the fuzzy-based prediction system's rules. GA has found a correlation between meteorological variables and insect breeding requirements using conditioned data from the cloud. The linguistic parameters of the Cauchy fuzzy membership function (CMF), which include $very\ high\ (VH), high\ (HI), moderate\ (MOD), low\ (LO)\ and\ very\ low\ (VL)$, have been derived from this correlation. The suggested approach has been tested in the Madhya Pradesh region of Gwalior, where the right environment is present for pests to flourish in rice and millets. The authors have determined that there are high and high incidences of pests, and this technique will assist farmers in taking preventive action in advance.

A fuzzy-based zoning smart irrigation system has been presented by H. Benyezza et al. [91] with the aim of optimizing greenhouse water and energy use. To do this, they have separated the greenhouse into various zones, used a node equipped with a soil moisture sensor in each zone, transferred data to a fuzzy system for best decision-taking, and utilized the cloud layer to store data for remote access. A real six-square-meter field has been divided into two zones and irrigated with tomato water for eight days to test its efficacy. After doing a comparative analysis with three other approaches suggested in the literature, it was discovered that the suggested algorithm outperformed other state-of-the-art for the identical trial area regarding energy consumption and water usage, by 65.22% and 26.41%, respectively.

Table 2.1 presents an overview of the current literature on service composition and elucidates how the research presented in this thesis differs from other investigations.

Table 2.1: A literature review on service composition optimization

Article [Ref.]	Description	Parameters	Types of objectives	Applications
N. Kashyap et al. [92]	Minimized time and cost & maximized reliability using a Hyper-heuristic approach	Population size = 100 No. of services = 10 -50 No. of candidate services = 10-50/service	Preference-based multi-objective	No application taken
P. Asghari et al. [93]	Proposed a model for predicting disease using techniques of data mining and provided composited medical prescriptions. Location, cost, and time as QoS metrics. Not providing optimal solutions using any EC	No. of services = 8 No. of candidate services = 6/service	Preference-based multi-objective	Smart Healthcare

	technique is a limitation			
N. Kashyap et al. [94]	Minimized time and cost and maximized reliability using GA and PSO. GA performed better than PSO	Population size = 100 No. of services = 10-50 No. of candidate services = 10-50/service	Preference-based multi-objective	No application taken
N. Kashyap et al. [95]	Minimized time and maximized reliability using NSGA-II algorithm in IoT	Population size = 100 No. of services = 10 No. of candidate services = 10, 30 and 50/service	Ideal multi-objective	No application taken
M. Razian et al. [96]	Proposed a new Anomaly-aware Robust service Composition (ARC) algorithm to address the issue of QoS value uncertainty in an IoT context that is always changing. Cost is minimized	Conducted a series of experiments.	Preference-based multi-objective	Smart healthcare as motivation scenario

S. Sefati et al. [97]	Five QoS parameters optimized using hidden Markov model, and ACO (HMM-ACO)	No. of services = 63 No. of candidate services = 1000-10000/service	Preference-based multi-objective	No application taken
P. Kumar et al. [98]	Seven QoS parameters have been optimized using a decision tree and GA	No. of services = 2,4,6,8,10 No. of candidates = 5-200/service	Preference-based multi-objective	No application taken
R. Boucetti et al. [99]	Nine QoS parameters optimized using neural network and GA	Population size = 20 No. of services = 2 No. of candidate services = 9/service	Preference-based objective	No application taken

After a comprehensive review of the existing literature on QoS-based optimization in smart agriculture, it is evident that service composition optimization in smart agriculture has not yet been investigated, and the idea of using ideal multi-objective optimization is still barely implemented. Moreover, the literature reveals that real-world smart agriculture systems involve numerous uncertainties that are often overlooked. Addressing these uncertainties is crucial for developing practical and robust optimization solutions. Overall, there is a significant research gap in applying multi-objective service composition optimization and checking the impact of uncertain conditions in smart agriculture.

2.5 Summary

This chapter offers a few insights from related work in the literature to understand the research gaps in the area of QoS-based service composition optimization. After carefully examining the literature, it has been discovered that the service composition problem has not yet been investigated in the context of smart agriculture, and the ideal multi-objective is barely used in this field. Another finding is that multi-objective optimization is useful in smart agriculture because it may resolve conflicting objectives with ease, as only multiple objectives can satisfy the user's complicated requirements instead of a single objective optimization. Furthermore, there are a lot of uncertain factors to consider while solving smart agriculture problems in the real world. Therefore, the multi-objective QoS-based service composition optimization in smart agriculture applications is the overarching focus of this thesis's study.

CHAPTER-3
LINEAR MULTI-OBJECTIVE SERVICE COMPOSITION OPTIMIZATION IN SMART AGRICULTURE USING EVOLUTIONARY COMPUTATIONAL TECHNIQUES

CHAPTER 3

LINEAR MULTI-OBJECTIVE SERVICE COMPOSITION OPTIMIZATION IN SMART AGRICULTURE USING EVOLUTIONARY COMPUTATIONAL TECHNIQUES

3.1 Chapter Overview

QoS-based service composition optimization is crucial for fulfilling the user's complex requirements. Local service selection and global composite service optimization are two approaches for this. For dispersed systems where centralized management is impractical, local selection works well whereas global optimization involves selecting the best candidate service for all atomic services in a workflow, aiming to achieve the top-quality composite service within the constraints set by the users. Thus, population-based meta-heuristic approaches have been widely used to tackle the issue of service composition optimization.

This chapter examines the idea of service composition in real-world smart agriculture applications by focusing on minimizing two important QoS-based metrics—cost and time. Additionally, it looks at how these goals are linearly related and discusses how to optimize composite services using three different EC techniques.

3.2 Linear Service Composition Model

Service composition is a combination of multiple web services, defined by QoS characteristics like time, scalability, cost, availability, and throughput. A service pipeline is used to route user requests, producing candidate service lists with distinct QoS requirements. The objective of the study is to offer the optimal solution for the apple orchard establishment and management in the Kullu and Shimla areas of Himachal Pradesh (a state in India) to address the multi-objective problem of associated time and cost in the growing surroundings. Let us suppose that there is total "s" services that are involved in the cultivation of apple harvests; these services are all regarded as atomic services with distinct QoS metrics. Out of which, each service "i" can have different candidate services or options based on QoS metrices which are time and cost in this case. It is assumed that each service i has a minimum completion time denoted by min $\_time$ and maximum completion time denoted by max $\_time$ along with c_min and c_max as the minimum and maximum cost for completion of that particular service. This complete concept can be mathematically expressed using equations 3.1 to 3.11 where equation 3.1 shows how atomic services (AS_i) can be described using candidate services (CS_{ij}) while equation 3.2 shows how these candidate services CS_{ij} are reliant on QoS factors [98].

$$AS_i = \{CS_{i1}, CS_{i2}, CS_{i3}, \dots CS_{ij}, \dots, CS_{ik}\} \tag{3.1}$$

$$CS_{ij} = \{QoS(CS_{ij})\} \qquad where, 1 \leq i \ \leq s \text{ and } 1 \leq j \leq k \tag{3.2}$$

Equation 3.3 below can be used to define the service composition once the QoS-based appropriate candidate service has been chosen.

$$C = \{CS_{1j}^{*}, CS_{2j}^{*}, CS_{3j}^{*}, \dots, CS_{sj}^{*}\} \tag{3.3}$$

Further, since this work considers minimizing the time and cost associated with the various atomic services as the objective function so the related time and cost with each service can be described using equations 3.4 and 3.5, respectively.

$$T = \{t_1, t_2, t_3, \dots, t_i, \dots, t_s\} \tag{3.4}$$

$$C = \{c_1, c_2, c_3, \dots, c_i, \dots, c_s\} \tag{3.5}$$

Where, t_i and c_i are the time and cost of i^{th} service, respectively.

The mathematical description of the objective function is given in equation 3.6 whereas "T" and "C" defines total time and total cost associated with all services given in equations 3.7 and 3.8, respectively.

$$Minimize\ (T, C) \tag{3.6}$$

$$T = \sum_{i=1}^{s} t_i \tag{3.7}$$

$$C = \sum_{i=1}^{s} c_i \tag{3.8}$$

For cost objective (c_i), it can be defined as the linear function of t_i by using the slope-intercept form shown in equation 3.9.

$$c_i = m_i t_i + \alpha_i \tag{3.9}$$

Where, $m_i = \frac{(max\ \_cost)_i - (min\ \_cost)_i}{(min\_time)_i - (max\ \_time)_i}$ (3.10)

and $\alpha_i = (max\ \_cost)_i - (min\ \_cost)_i$ (3.11)

Here, m_i is the slope of i^{th} service, indicating the rate of change in cost with respect to time and α_i is y-intercept.

The concept of the linear relationship between time and cost used in this work is illustrated in Figure 3.1 [119].

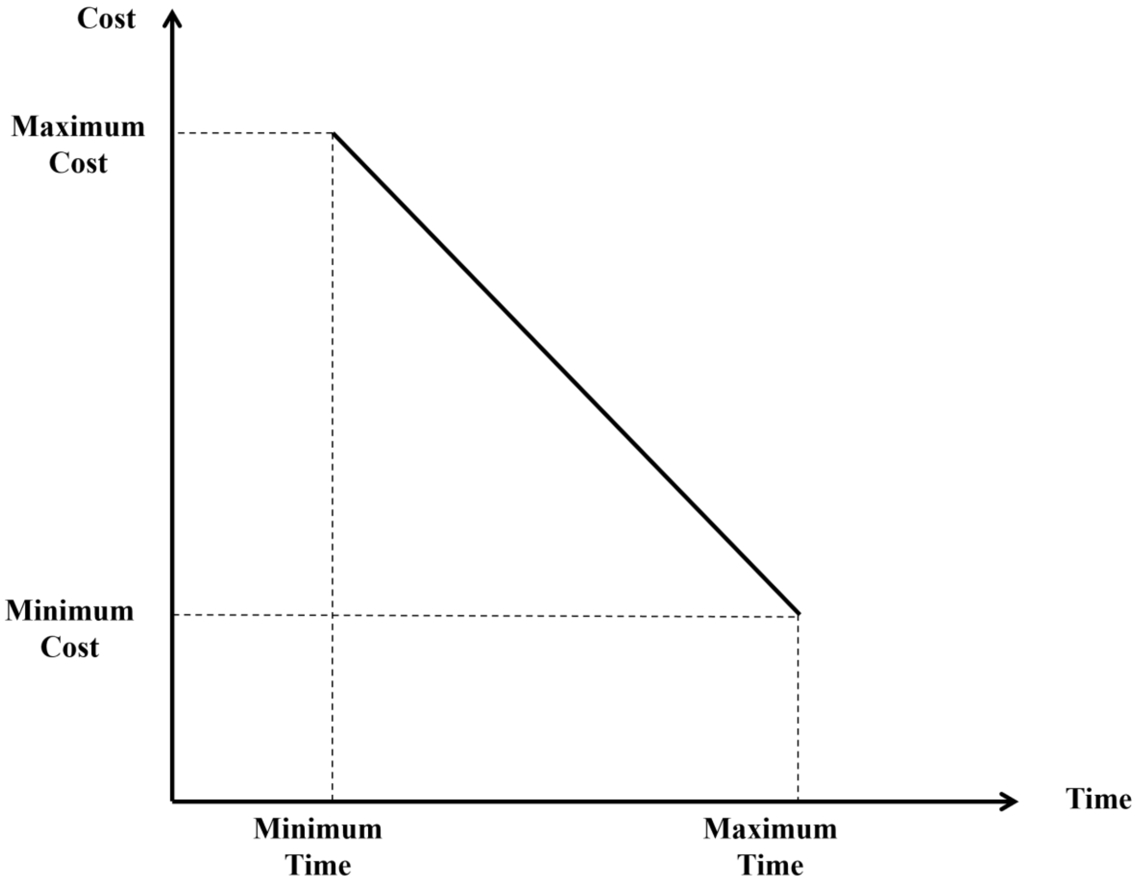


Figure 3.1: Linear time-cost trade-off of services using slope-intercept form [119]

Figure 3.2 offers a general view of the QoS-based service composition strategy for better understanding. In Figure 3.2, a service composition plan is portrayed, comprising three atomic services. A variety of cloud-based services must be chosen from a pool of candidate services in order to carry out this approach. For every atomic service, let's say there are four candidate services. So, the key question is: which candidate service ought to be picked? This choice is made in the service selection phase when the relevant candidate services are picked in accordance with the defined QoS metrics. The service composition plan is then finally carried out after the best candidate services have been chosen.

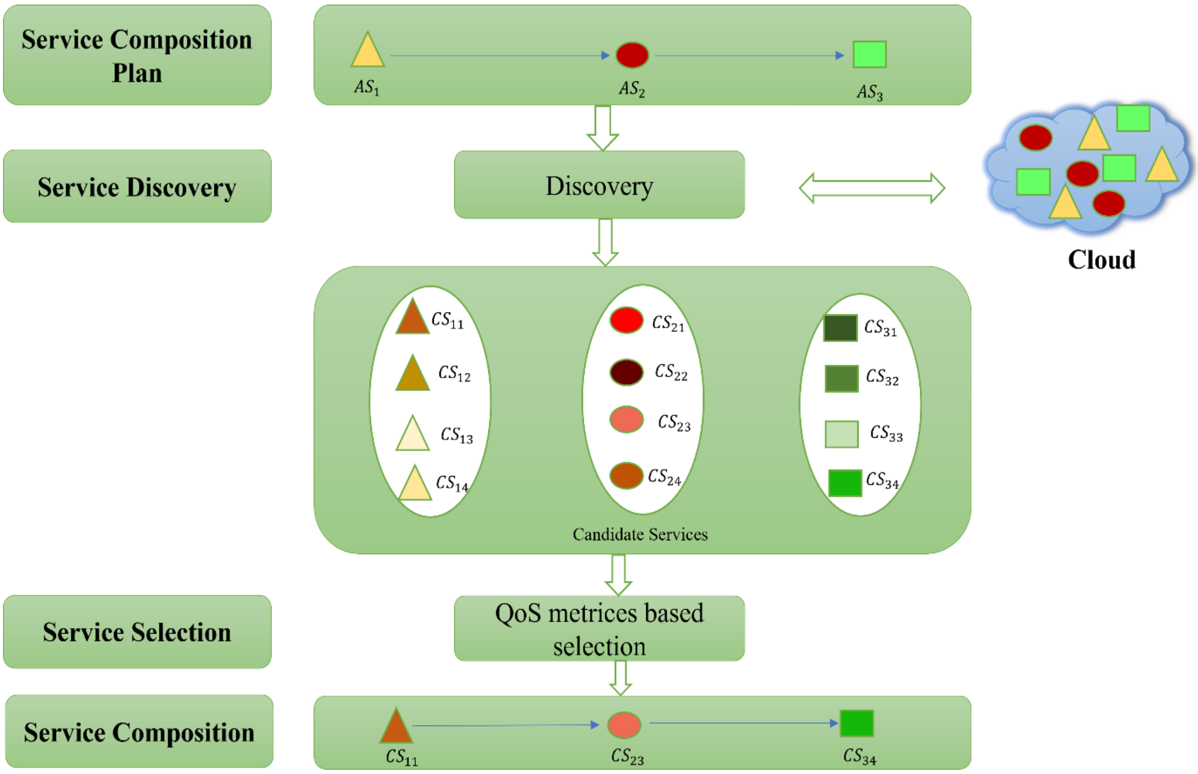


Figure 3.2: Understanding of QoS-based service composition

3.3 Case Study

Most researchers have focused on reducing fertilizer usage, improving irrigation management systems, and increasing crop productivity and profitability; however, the integration of these diverse services and their optimization to achieve multiple objectives simultaneously has not yet been investigated. This optimization of integrated services can help provide customized optimal plans to the farmers and users. To understand this concept, an illustrative scenario of service composition in smart agriculture is explained. Consider a scenario where "Company A" creates an agricultural plan for its customers/users, offering the following atomic services related to apple tree cultivation and management.

- Soil Testing and Analysis
- Apple Variety Selection
- Orchard Establishment
- Tree Planting

- Fertilizer Application
- Irrigation System Installation
- Pest and Disease Control
- Pruning and Training
- Crop Monitoring and Management
- Harvesting
- Packaging and Labelling
- Sorting and Grading
- Storage and Cold Chain Management
- Marketing and Distribution

The complete service composition process is illustrated in Figure 3.3 using a unified modelling language (UML) diagram [96]. In this scenario, customers will approach the company with specific service requests. The company will then create a tailored plan that incorporates only the services desired by the customers. This customized plan will be based solely on the services explicitly requested by the users.

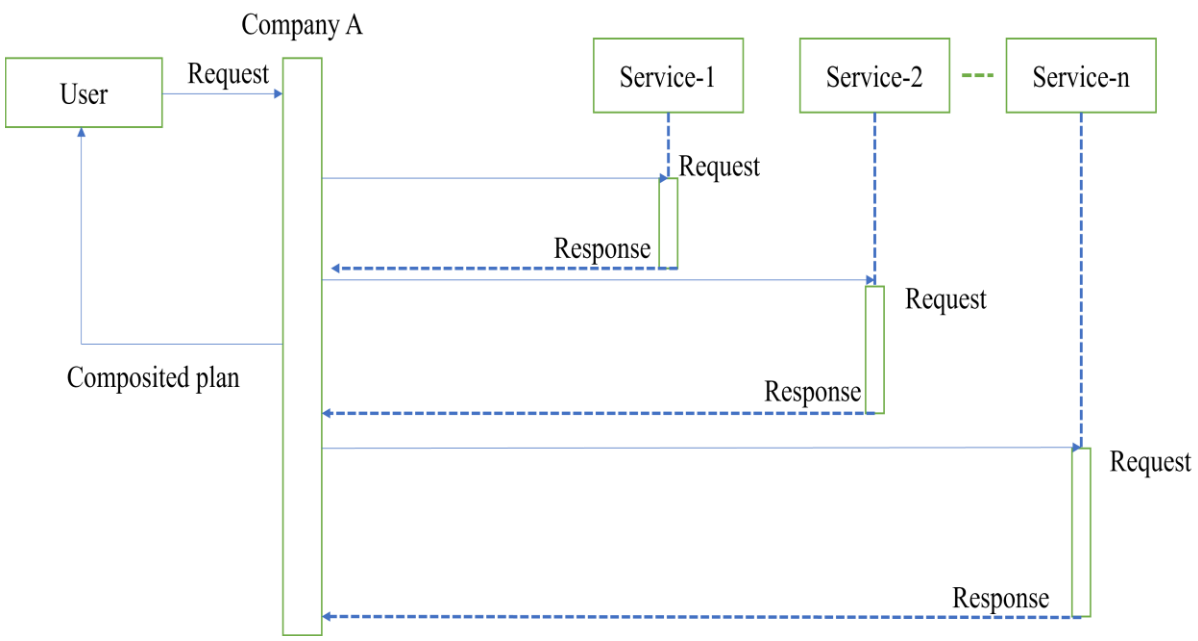


Figure 3.3: Sequence diagram showing the flow of service composition [96]

3.3.1 Proposed Dataset

To enforce the service composition optimization in agriculture, a survey on establishing and managing apple orchards has been conducted on the fifty-three farmers of the Shimla and Kullu regions of Himachal Pradesh (a state in India). Based on their responses, a dataset has been created that includes the basic fourteen services starting from soil testing to marketing and distribution, required to establish and manage the apple orchard within one acre of an area.

The criteria for including and excluding the responses are as follows:

a) Inclusion criteria
   - People who respond to the minimum 70% of questions.

b) Exclusion criteria
   - People with no experience of apple orchards.
   - People who are unwilling to respond to less than 70% of questions.

Those fourteen services along with corresponding cost and time metrices are cataloged in Table 3.1.

Table 3.1: Dataset showcasing atomic services in smart agriculture

Service Number	Atomic Services	Cost (in rupees)	Time (in days)
1	Soil Testing and Analysis	10000	7
		5000	14
2	Apple Variety Selection	4000	1
		2000	3
3	Orchard Establishment	200000	30
		50000	90
4	Tree Planting	10000	2
		7000	6

5	Irrigation System Installation	150000	7
		50000	14
6	Fertilizer Application	100000	14
		50000	28
7	Pruning and Training	30000	7
		15000	21
8	Pest and Disease Control	100000	14
		70000	28
9	Crop Monitoring and Management	50000	60
		20000	120
10	Harvesting	70000	14
		35000	28
11	Sorting and Grading	30000	7
		15000	14
12	Packaging and Labelling	90000	14
		60000	28
13	Storage and Cold Chain Management	50000	60
		25000	120
14	Marketing and Distribution	80000	90
		40000	180

The two primary QoS metrics in this study that must be simultaneously minimized to give the user an optimal plan are cost and time. Take the service of soil analysis and testing, for instance. For this service, there are two options: one that costs 10,000 rupees and takes seven days, and another that costs 5,000 rupees and takes fourteen days. There is a possibility of having other options that fall between these cost and time frames, offering a wide variety of choices.

An ideal solution based on the user's specific preferences is needed to identify the best option. For example, a user may select the second option if he/she is more concerned about the cost as this option increases the time taken but is cheaper. However, if the priority of the user is time over cost, he/she may opt for faster service. Whether the user wants to save time, cut costs, or strike a compromise between the two, the objective is to choose the service option that best suits his/her priorities. However, if the user values time over cost, they may opt for the faster service, even if it costs more. Similarly for the second service which is apple variety selection, speaking with specialists or researching several apple varieties that are appropriate for the soil and climate in the area is a must. So, it can take either one day with a cost of 4,000 rupees or three days with a cost of 2,000 rupees or in between. The same will happen for other services. Thus, this work provides an optimal service composition plan for the farmers/users for an entire agricultural process, ensuring that farmers achieve the best possible outcomes for their field.

3.4 Methodology for Linear Service Composition Optimization

To get the optimal responses for service composition, various distinct meta-heuristics can be used. There are two basic stages to these meta-heuristics that are included in each algorithm used in this work.

3.4.1 Population Initialization

Initializing the population is a foremost and crucial stage in any meta-heuristic algorithm. It involves representing a possible solution in a manner that the algorithm can understand. The population initially consists of "N" solutions equivalent to population size, each solution is represented with a string $[t_1, t_2, t_3, \dots, t_i, \dots, t_t]$ where min $\_time \leq t_i \leq$ max $\_time$. The size of the string equals the aggregate number of services considered, with indices denoting the corresponding number and contents indicating the specific candidate for each service. Figure 3.4 illustrates the solution representation process, using time as the objective measure [119].

This work in this thesis consists of fourteen atomic services involved in apple orchard establishment and management.

t_1	t_2	t_3	------------------------------	t_t

A n-tuple string

Figure 3.4: Solution representation for atomic services by taking time as an objective function [119]

3.4.2 Evaluation of Objectives

After initializing the population using time as an input variable, the next step includes evaluating cost using slope intercept form as both have a linear relationship between them and already shown in equations 3.9 to 3.11. The pictorial representation of solution after calculating both objectives for fourteen atomic services is portrayed in Figure 3.5.

S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	Time	Cost
t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	T	C

Figure 3.5: Solution representation for atomic services including time and cost objective functions

Thus, this step generates a potential population of solutions by using linear slope intercept form to estimate the cost for composite services corresponding to the service time.

After these two steps, the further mechanism for generating Pareto optimal solutions for multi-objective problem is followed as per the pseudocode of specified meta-heuristic algorithm.

3.5 Linear Service Composition Optimization using MOGA

This section describes how the composited services with a linear relationship between time and cost objectives are optimized using MOGA.

3.5.1 Optimization Algorithm: MOGA

A population-based optimization method inspired by nature that imitates the behavior of genetic processes is called a genetic algorithm. John Holland initially suggested this automated and computerized search method in 1990 [32]. Unlike traditional searching algorithms, the GA begins the search from an arbitrarily created primary collection called the population. A single chromosome comprises every member of the population. A chromosome is a binary code-like sequence of characters for a binary-coded genetic algorithm. In every generation, the fitness function is computed to determine how effectively the current set of chromosomes is working. It's a quality that, whether it's maximization or minimization, must always be at its peak. The next step in GA is parent selection, which is important since the fitness of the next generation directly affects how optimizations turn out. After that, the chosen parents experience crossover procedure, and mutation procedure to produce the offspring, which are new chromosomes. Only the fittest chromosomes will survive in the newly generated population since the chromosomes are picked as per their fitness function, eliminating any unwanted chromosomes. Pareto optimum solutions are the chromosomes on which the population converges after several repeats [100].

The procedures listed below must be completed in order to use GA to achieve a globally optimized Pareto optimal solution for multi-objective problems.

a) Initialization of the Population and Encoding

The population is the total number of possible ways to solve a particular problem. A gene is an element's index, whereas a chromosome refers to a single solution. Therefore, a chromosome is made up of genes, and a population is made up of several chromosomes. This work depicts the chromosome by using a string with gene number equal to the number of atomic services taken as mentioned in subsection 3.4.1.

b) Fitness Function

To determine the fitness value for each chromosome, the fitness function must be defined after the population has been initialized. It takes the value of the chromosome that fits the best out of all those compared at each iteration. The fitness function can be set to maximize or minimize based on the needs of the user. The fitness functions that this work has adopted are cost and time minimization.

c) Selection Mechanism

The population's average quality is greatly increased via selection, which transfers the better-quality chromosomes to the following generation. Every iteration generates a "N" number of new individual offspring from “N” number of pre-existing individual parents. Parents and children have to compete with each other to make it into the next iteration. This study makes use of a tournament selection approach in which a tournament is created by selecting "p" random chromosomes from the population. The chromosome with the best fitness among them is selected as the tournament winner and advances to the following round. It continues until the number of parents becomes equal to the population size [101].

d) Crossover Mechanism

The first genetic change introduced to a mating pool’s chromosomes is called a crossover. Establishing a communication channel between two chromosomes is the goal of crossover. By exploring new offspring, the algorithm aims to identify superior offspring based on the discovered fitness value. This work employs simulated binary crossover (SBX) with a probability equal to σ in our work. There are two different coefficients (β) for the SBX operator to assess based on the values of the $rand$ function, which has random values between 0 and 1. Equation 3.12 is defined in the following two cases:

$$\beta = \begin{cases} (2 * rand)^{1/3} & if\ rand < \sigma \\ \frac{1}{(2*(1-rand))^{1/3}} & otherwise \end{cases} \tag{3.12}$$

Furthermore, SBX generates two offspring from a pair of randomly selected parental solutions drawn from the existing population. Ultimately, one of the children is retained based on equal likelihood [102]. The primary contribution of SBX to the whole algorithm is its ability to expedite the Pareto Front blending process by recombining different solutions.

e) Mutation Mechanism

A mutation operation is performed on the new offspring chromosome to change one or more genes in order to establish the new chromosome. This mechanism uses a polynomial distribution index parameter η_m that determines how much the solution can be disrupted by controlling the magnitude of variations. A random number u between 0 and 1 is chosen by the operator. Based on this random number, the mutated parent p' is created for a given parent "p" shown in equation 3.13 given below.

$$p' = \begin{cases} p + \overline{\delta_L}\left(p - x_i^{(L)}\right) & if\ u \leq 0.5 \\ p + \overline{\delta_R}\left(x_i^{(U)} - p\right) & if\ u > 0.5 \end{cases} \quad (3.13)$$

Next, the following formulas in equations 3.14 and 3.15 are used to determine one of the two parameters $\overline{\delta_L}$ and $\overline{\delta_R}$.

$$\overline{\delta_L} = (2u)^{\frac{1}{(1+\eta m)}} - 1 \qquad if\ u \leq 0.5 \quad (3.14)$$

$$\overline{\delta_R} = 1 - (2(1-u))^{\frac{1}{(1+\eta m)}} \qquad if\ u \leq 0.5 \quad (3.15)$$

Here, $x_i^{(L)}$ and $x_i^{(R)}$ defines the lower and upper bounds of the i^{th} variable of the solution. $\overline{\delta_L}$ and $\overline{\delta_R}$ regulates the extent to which the mutation pushes the solution in the direction of the lower and upper bounds, respectively. Thus, early convergence and population diversity are preserved by the mutation [103]. The polynomial mutation is utilized in this work to replace genes.

Below is presented the pseudocode for MOGA in Figure 3.6.

Algorithm: MOGA

Begin

    Solution Representation, $t \coloneqq 1$, Maximum allowed generation = T;
    Initialize random population $P(t)$;
    Evaluate $P(t)$ and assign rank using non-dominated sorting
    while t < T do
        $M(t) \coloneqq Selection\ (P(t))$;   %Selection%
        $Q(t) \coloneqq variation\ (M(t))$;   % Crossover and Mutation%
        Evaluate $Q(t)$;   % Offspring%
        $P(t+1) \coloneqq$ Q(t);
        $t \coloneqq t + 1$;
    end while

End

Figure 3.6: Pseudocode of MOGA

To elucidate the concept of MOGA, a flow chart is presented in Figure 3.7 below.

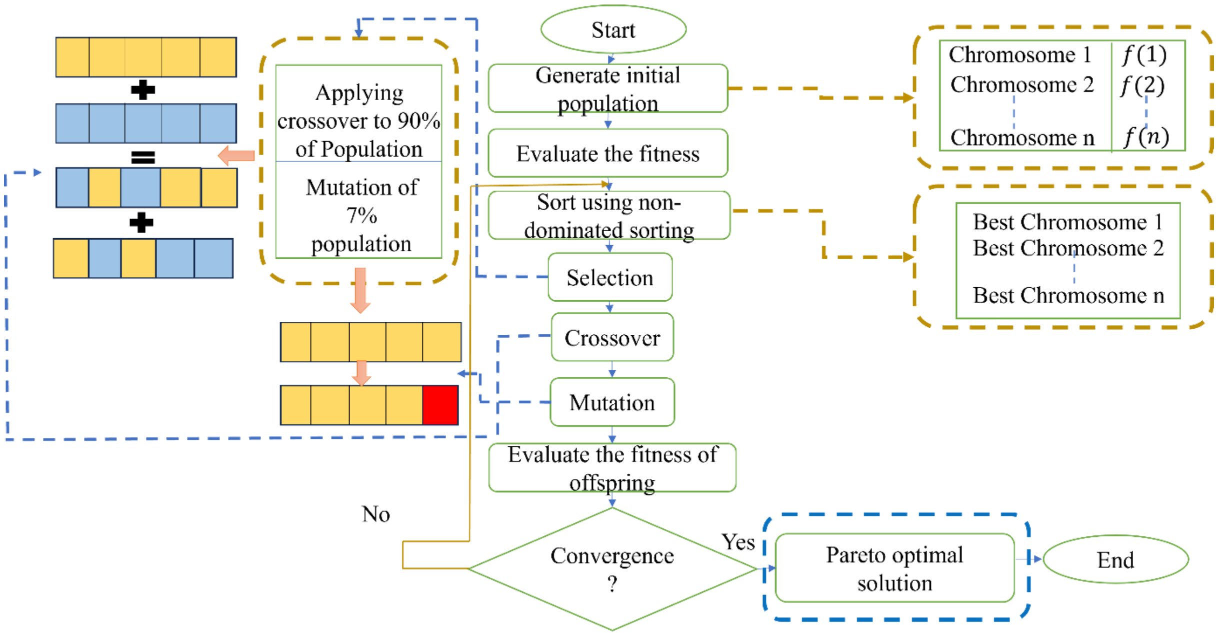


Figure 3.7: Illustration of MOGA using flow chart

3.5.2 Proposed Framework

This framework operates across various tiers of IoT infrastructures. IoT sensor data is stored in cloud-based services. Numerous services provide comparable functionalities but with differing QoS characteristics. Initially, services with similar functions are identified during the discovery phase. Subsequently, services are chosen from the available options to meet user requirements, based on QoS criteria. Complex user requests typically require multiple services, necessitating a service composition phase. The composited services are then optimized using MOGA to provide a series of Pareto solutions. The whole framework is portrayed in Figure 3.8.

3.5.3 Simulation Setup

The proposed approach is run on a desktop computer equipped with 16 GB RAM, and MATLAB R2013a software. The various parameters required to be set while executing MOGA are structured in Table 3.2. When trade-off points hold steady for three subsequent iterations—achieved in 1000 iterations—the search for optimal solutions is terminated.

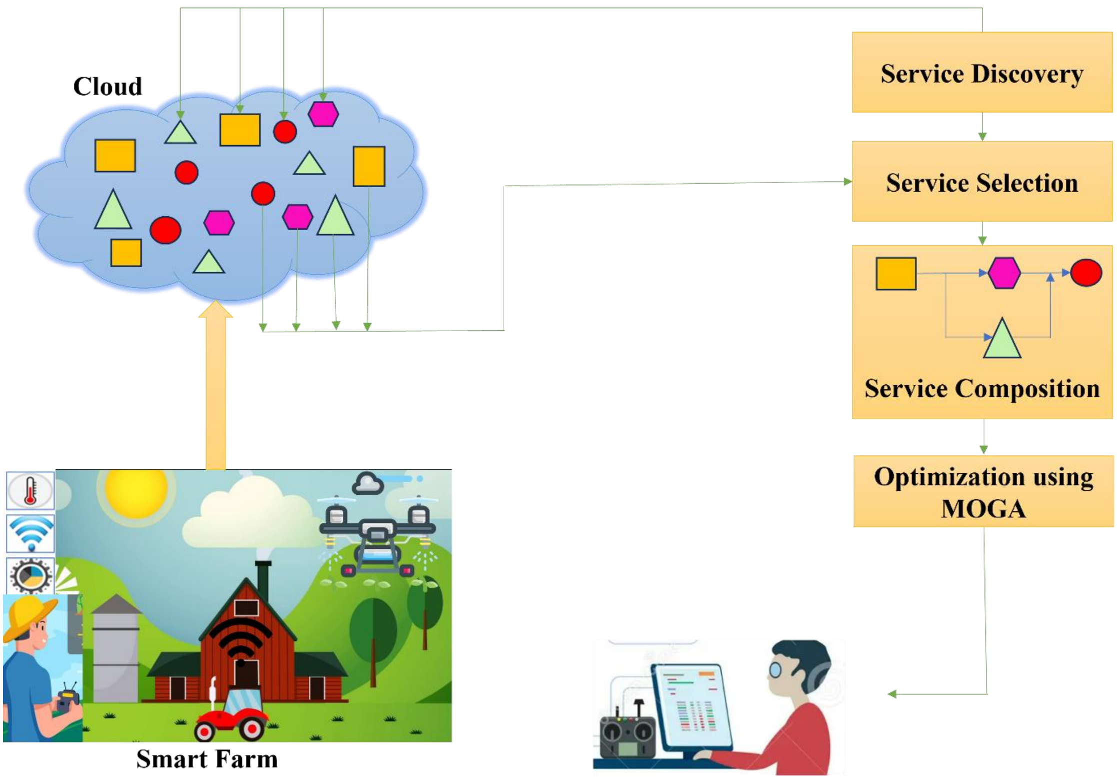


Figure 3.8: Proposed framework for service composition optimization

Table 3.2: Genetic operators for MOGA

Parameters	**Values**
Population Size	200
Selection Mechanism	Tournament Selection
Crossover Operator	SBX
Mutation Operator	Polynomial Mutation
Crossover Probability	0.9
Mutation Probability	0.07
No. of iterations	1000

3.5.4 Results and Discussions

Figure 3.9 displays the simulation results for the service composition optimization problems, wherein after a predetermined number of iterations, the Pareto optimal solutions are found. The results show that MOGA generates trade-off points between time and cost parameters in the realm of smart agriculture by offering diverse Pareto optimal solutions for multi-objective optimization problems. The solutions offered show the range of choices farmers can make in response to their complicated and varied needs.

Table 3.3 provides a statistical analysis of the simulation outputs for a more in-depth look at the data.

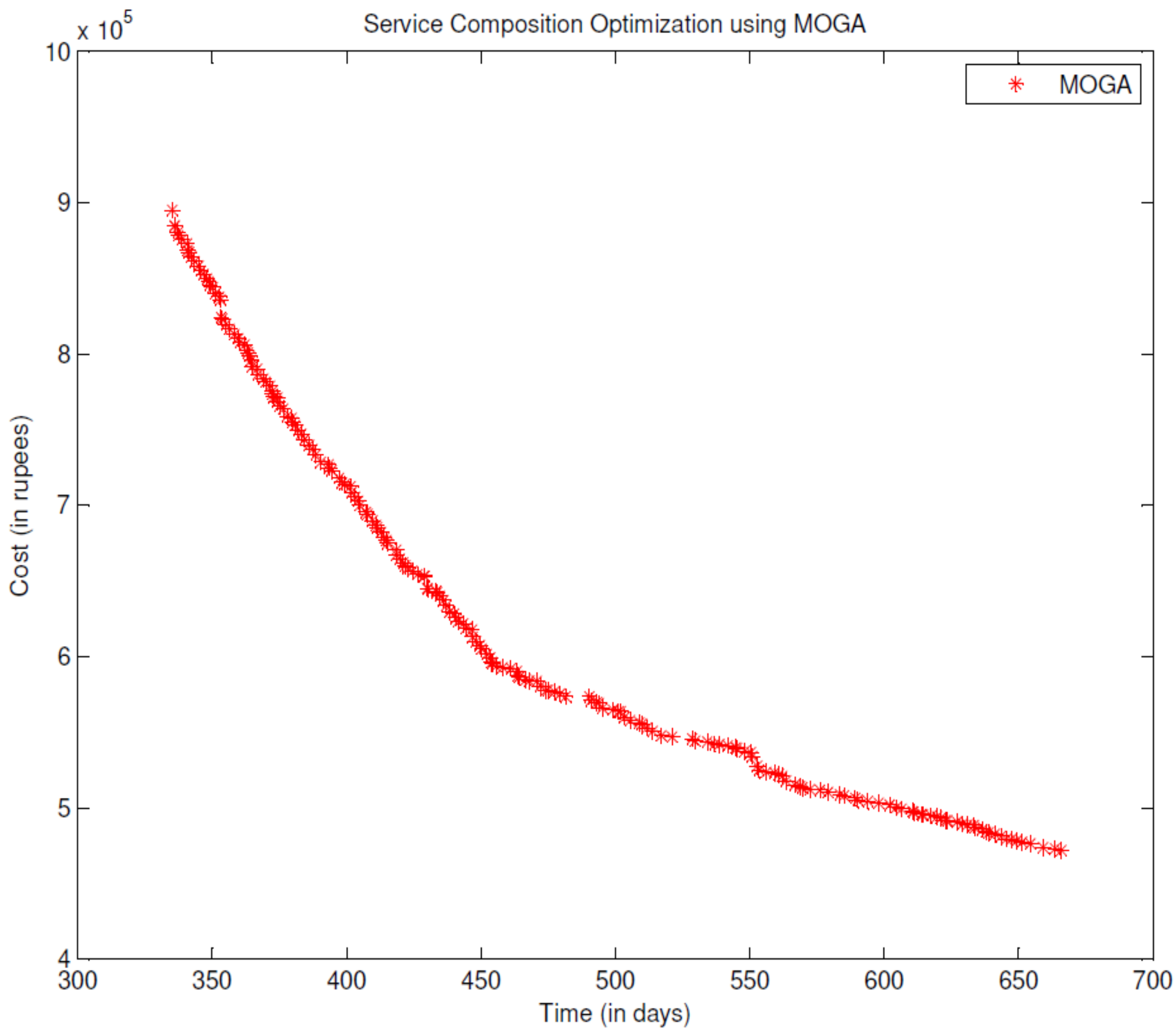


Figure 3.9: Pareto optimal solutions obtained using MOGA

Table 3.3: Statistical analysis

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
MOGA	Time	666	335.2	97.48	467.7	445.7	335.2	330.8
	Cost	8.951e+05	4.719e+05	1.25e+05	6.424 e+05	6.183 e+05	4.719e+05	4.232 e+05

3.6 Linear Service Composition Optimization using NSGA-II

This chapter section explains how NSGA-II is used as an optimization algorithm to serve service composition optimization in smart agriculture.

3.6.1 Optimization Algorithm: NSGA-II

NSGA-II is an enhanced version of the NSGA algorithm, which was introduced by N. Srinivas and K. Deb in 1995 [104]. Among the many shortcomings of the original method were its excessive computing complexity, lack of a distribution parameter, and inadequacy of elitism. To address these issues, Deb proposed a multi-objective evolutionary algorithm called NSGA-II in 2002 [105]. This improved algorithm employs non-dominated sorting and crowding distance techniques to discover a well-distributed set of solutions and enhance diversity for various multi-objective problems.

The basic foundation of the NSGA-II algorithm is defined below.

a) Non-dominated Sorting

This method involves ranking population members based on Pareto dominance. The process of non-dominated sorting commences by assigning the highest rank to non-dominated individuals in the initial population. These top-ranked members are then moved to the first front and excluded from the original population. Subsequently, the remaining population undergoes non-dominated sorting. The non-dominated individuals from this subset are given the second rank and placed in the second front. This ranking and sorting continue until every member of the population is dispersed over different fronts in accordance with their designated ranks, as illustrated in Figure 3.10 [106].

b) Elitism-preserving operator

The elitism-preserving strategy is a method that maintains the best solutions within a population by directly moving them to the subsequent generation. This approach ensures that the most effective, non-dominated solutions discovered in each generation continue to exist in future generations until they are surpassed by superior solutions.

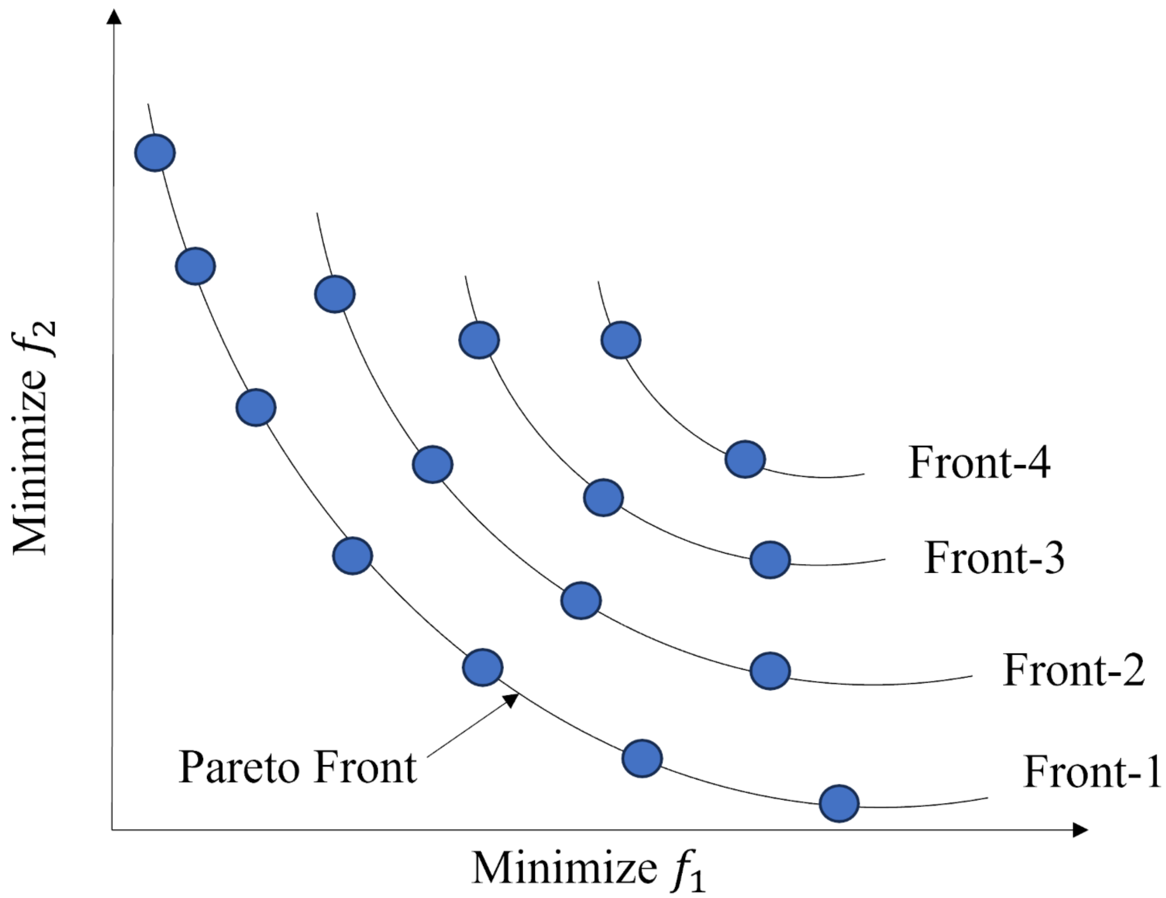


Figure 3.10: Concept of non-dominated sorting [106]

c) Crowding Distance Concept

The concentration distribution of solutions enclosing a particular solution is estimated using crowding distance. This measure is calculated by taking the average distance between two solutions on either side of the solution for each objective. The solution with the greater crowding distance is seen as being in a less crowded region when two solutions with distinct crowding distances are compared. As illustrated in Figure 3.11 [106], the crowding distance of the i^{th} solution is represented by the average side length of the cuboid. Let f_k^i denote the k^{th} value of any objective function for the i^{th} individual, and f_k^{max} and f_k^{min} represent the maximum and minimum values of the k^{th} objective function across all individuals, respectively. The crowding distance of the i^{th} individual is then outlined as the average distance between the two closest solutions on either side, as expressed in equation 3.16.

$$CD(i) = \sum_{i=1}^{j} \frac{f_k^{i+1} - f_k^{i-1}}{f_k^{max} - f_k^{min}} \tag{3.16}$$

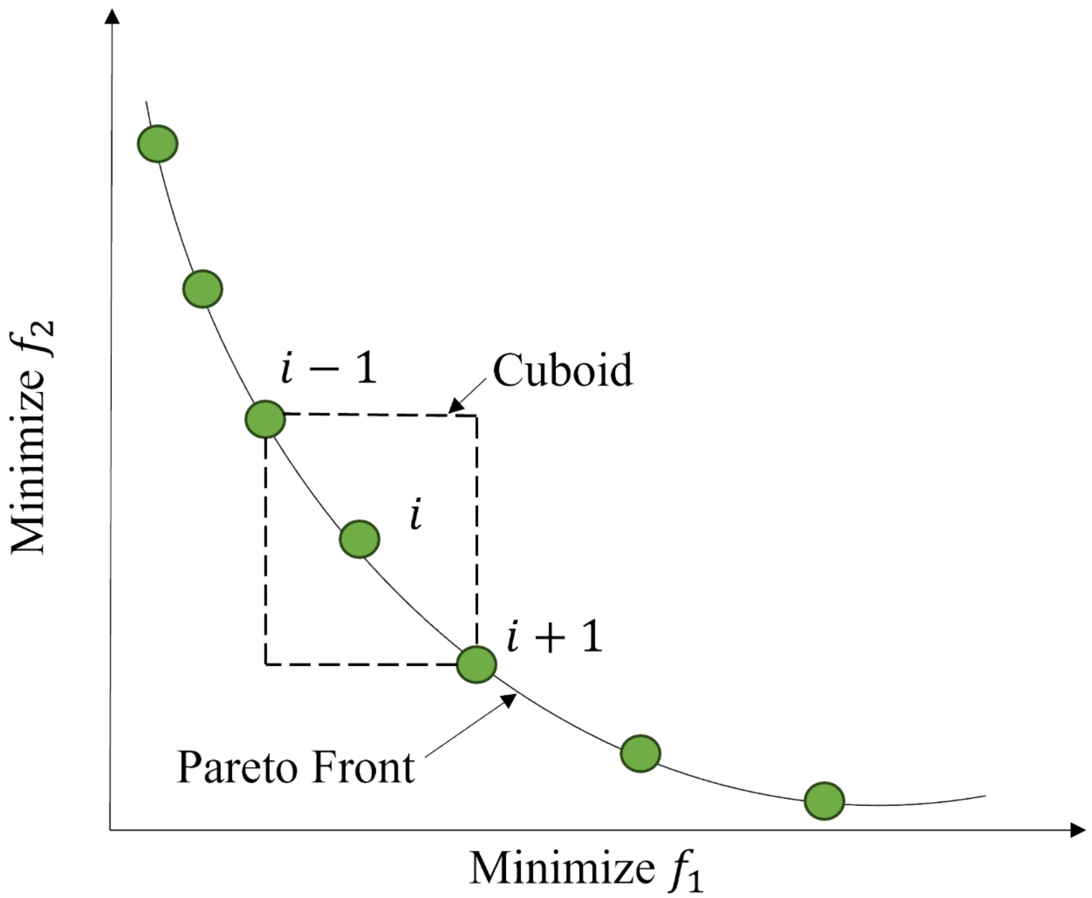


Figure 3.11: Concept of crowding distance [106]

d) Selection Mechanism

The selection of individuals for the subsequent generation employs a crowded tournament selection mechanism, which considers both the ranking and crowding distances of population members. The process for choosing between two individuals for the next generation follows these rules:

- When the two individuals have distinct ranks, the individual with the superior rank is chosen to advance to the following generation.
- In cases where both individuals share an identical rank, the one with the larger crowding distance is selected for the following generation [106].

e) Genetic Operators

Crossover and mutation operators are essential genetic operators that produce offspring populations from the parent population, guarantee diversity, and efficiently explore the search space. The process of combining two parent solutions to create one or more offspring is called crossover. The SBX [107] is utilized as a crossover operator in NSGA-II. Although it operates on real-coded individuals, it resembles the single-point crossover behavior of binary-coded genetic algorithms. To create offspring that are identical to their parents but introduce variability, the SBX operator exchanges components of two parent solutions to produce fresh individuals. The crossover operator creates the offspring, while the mutation operator adds tiny, random alterations to it. Preventing premature convergence to local optima and preserving genetic variety in the population depends on this. The Polynomial Mutation [108] is frequently applied to real-coded individuals in NSGA-II. This operator introduces variances into its offspring by slightly altering the choice variables according to a probability distribution.

3.6.2 Procedure of NSGA-II

The algorithm's process starts by creating an initial population P_t of "N" members. A new population Q_t is then formed through crossover and mutation of P_t . These two populations are combined to create R_t which is then evaluated via non-dominated sorting. The members of R_t are then categorized into several distinct fronts based on their degree of non-domination.

The subsequent step involves selecting "N" members from R_t to form the subsequent population P_{t+1}. If the first front contains "N" or more members, "N" individuals are chosen from its least crowded areas to create P_{t+1}. However, if the first front has fewer than "N" members, all of them are progressed directly to the subsequent generation. The unused slots are filled with members from the smallest crowded areas of the second front. This process continues with subsequent fronts until P_{t+1} reaches "N" members. The same method is used to generate future populations (P_{t+2}, P_{t+3}, etc.) until the stopping criteria are met [105]. Figure 3.12 illustrates the procedure of NSGA-II [105]. Figure 3.13 presents a simplified and easily comprehensible outline of the steps involved in the NSGA-II with the help of pseudocode. Further, the flow chart for the same is displayed in Figure 3.14.

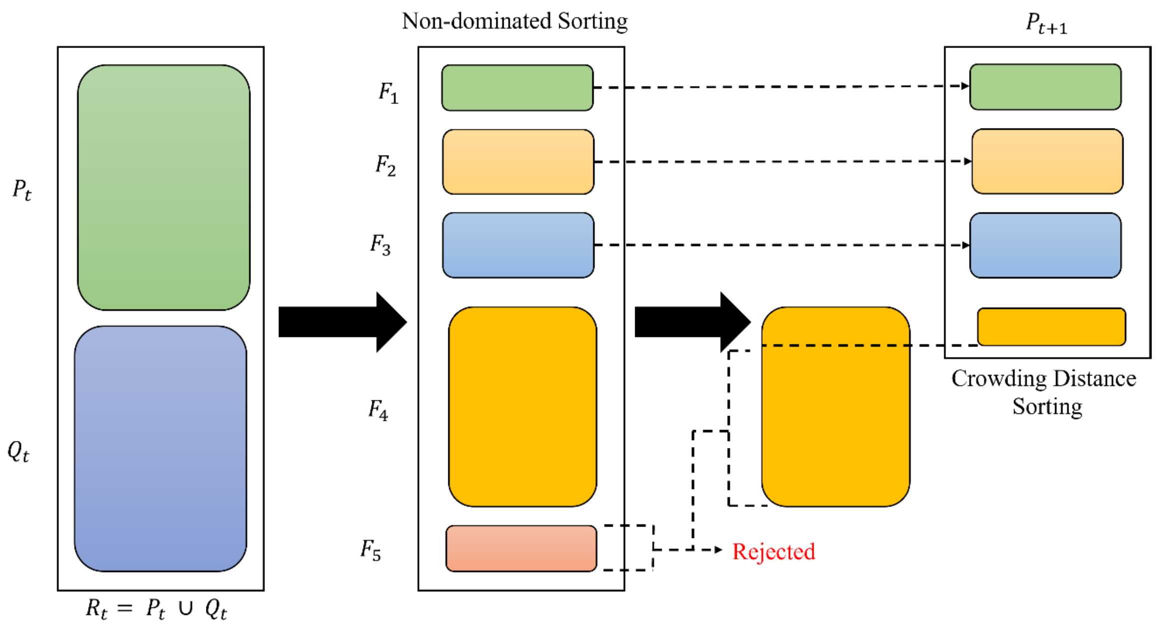


Figure 3.12: Procedure for NSGA-II [105]

Algorithm: NSGA-II

Begin
- Solution Representation, $t := 1$, Maximum allowed generation = T;
- Initialize random population $P(t)$;
- Evaluate $P(t)$ and assign rank using dominance depth method and diversity using crowding distance method to P(t);
- while t < T do
  - $M(t) := Selection\ (P(t))$; %Crowded Binary Tournament Selection%
  - $Q(t) := variation\ (M(t))$; % Crossover and Mutation%
  - Evaluate $Q(t)$; % Offspring%
  - Merge population $\hat{P}(t) = (P(t) \cup Q(t))$;
  - Assign Rank using dominance depth method and diversity using Crowding distance operator to $\hat{P}(t)$;
  - $P(t+1) := Survivor\ (\hat{P}(t))$;
  - $t := t + 1$;
- end while

End

Figure 3.13: Pseudocode for NSGA-II algorithm

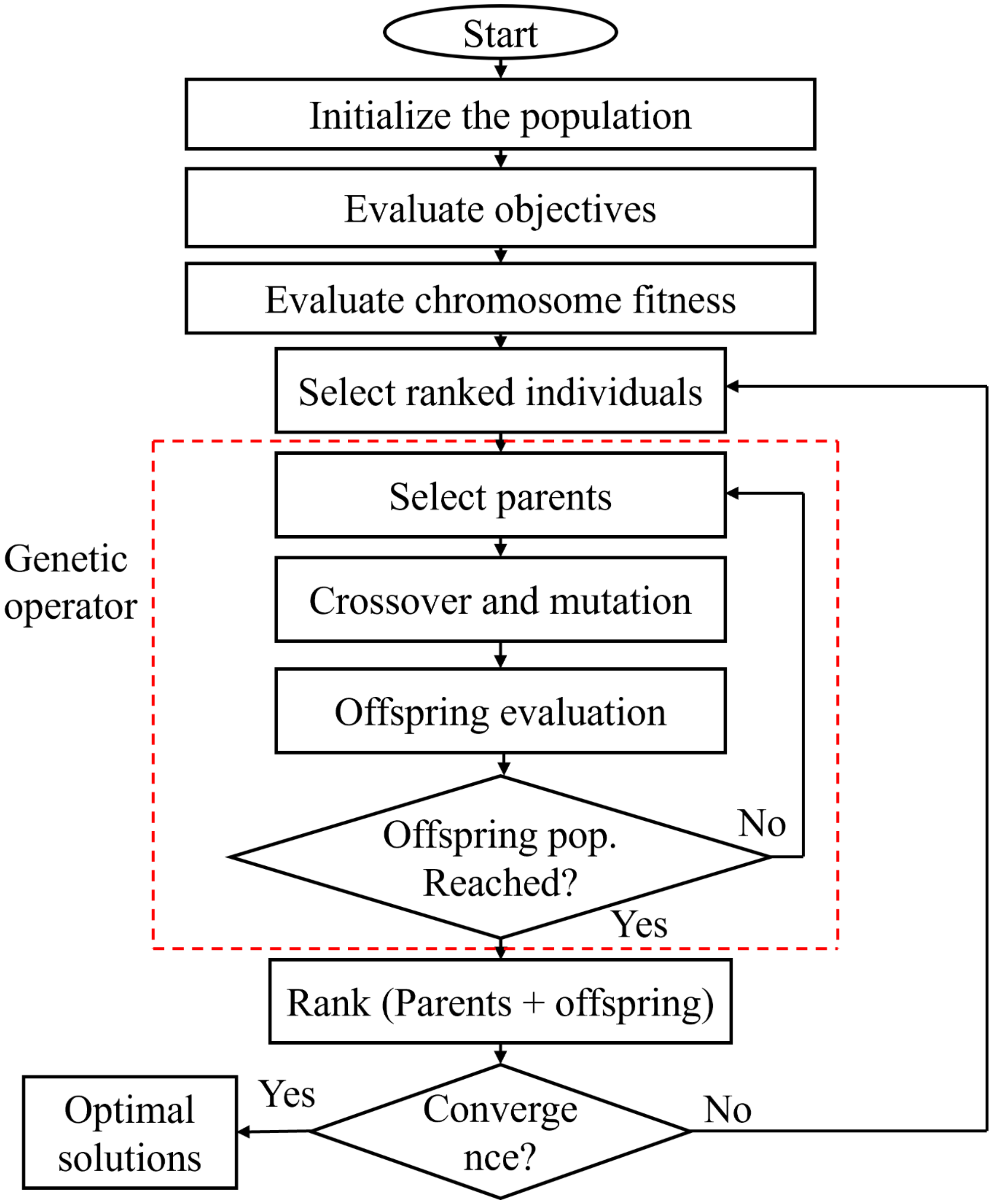


Figure 3.14: Flow chart illustration of NSGA-II

3.6.3 Proposed Framework

This framework functions on different IoT infrastructure layers. Cloud-based services are used to store data from IoT sensors. Many services offer similar features, yet they have different QoS attributes. During the discovery phase, services with comparable functions are first found.

Based on QoS criteria, services are then selected among the available possibilities to satisfy user requirements. A service composition step is required because complex user queries usually ask for many services. After that, NSGA-II is used to optimize the composited services, yielding a collection of Pareto optimal solutions. Figure 3.15 illustrates the entire framework.

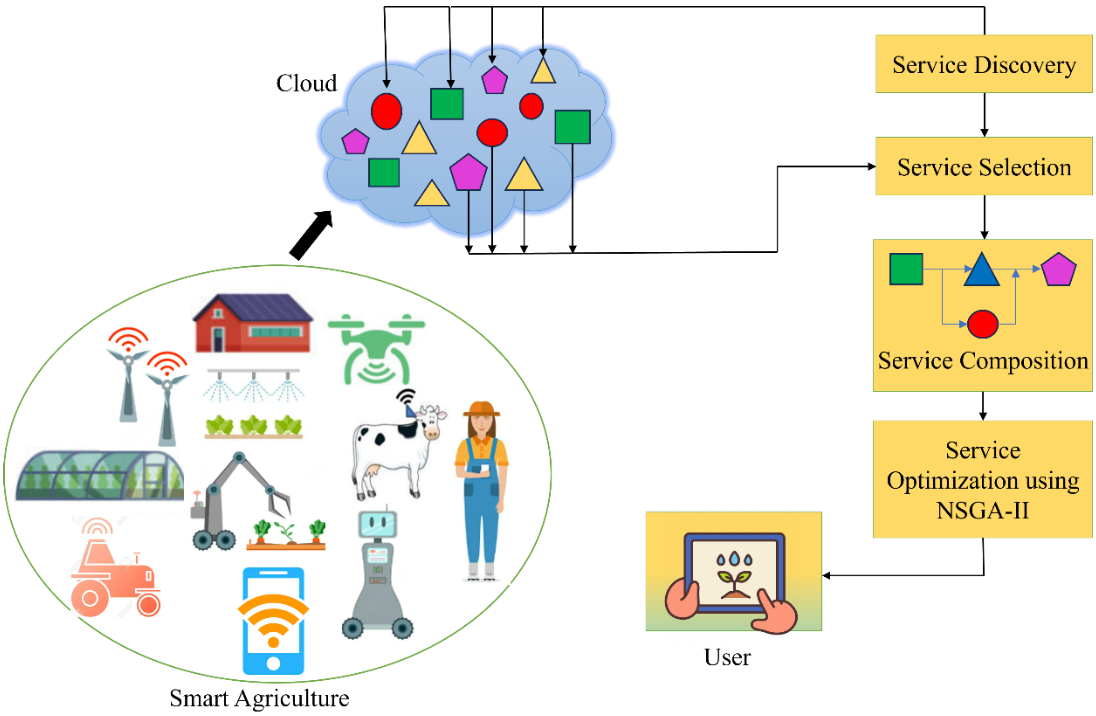


Figure 3.15: Proposed framework for service composition optimization using NSGA-II

3.6.4 Simulation Setup

The main parameters utilized to validate the algorithmic performance are listed in Table 3.4. Time and cost minimization are the main goals of the optimization process, and the fitness function is made to balance these goals. When the trade-off between the two goals is maintained for three consecutive iterations—usually within 1000 generations—the search process comes to an end.

3.6.5 Results and Discussions

Following the simulation, the Pareto optimum solutions show a distinct movement toward the coordinate axes, as seen in Figure 3.16.

Table 3.4: Simulation operators of NSGA-II

Parameters	**Values**
No. of iterations	1000
Population Size	200
Mutation Probability (P_m)	0.07
Crossover Probability (P_c)	0.9

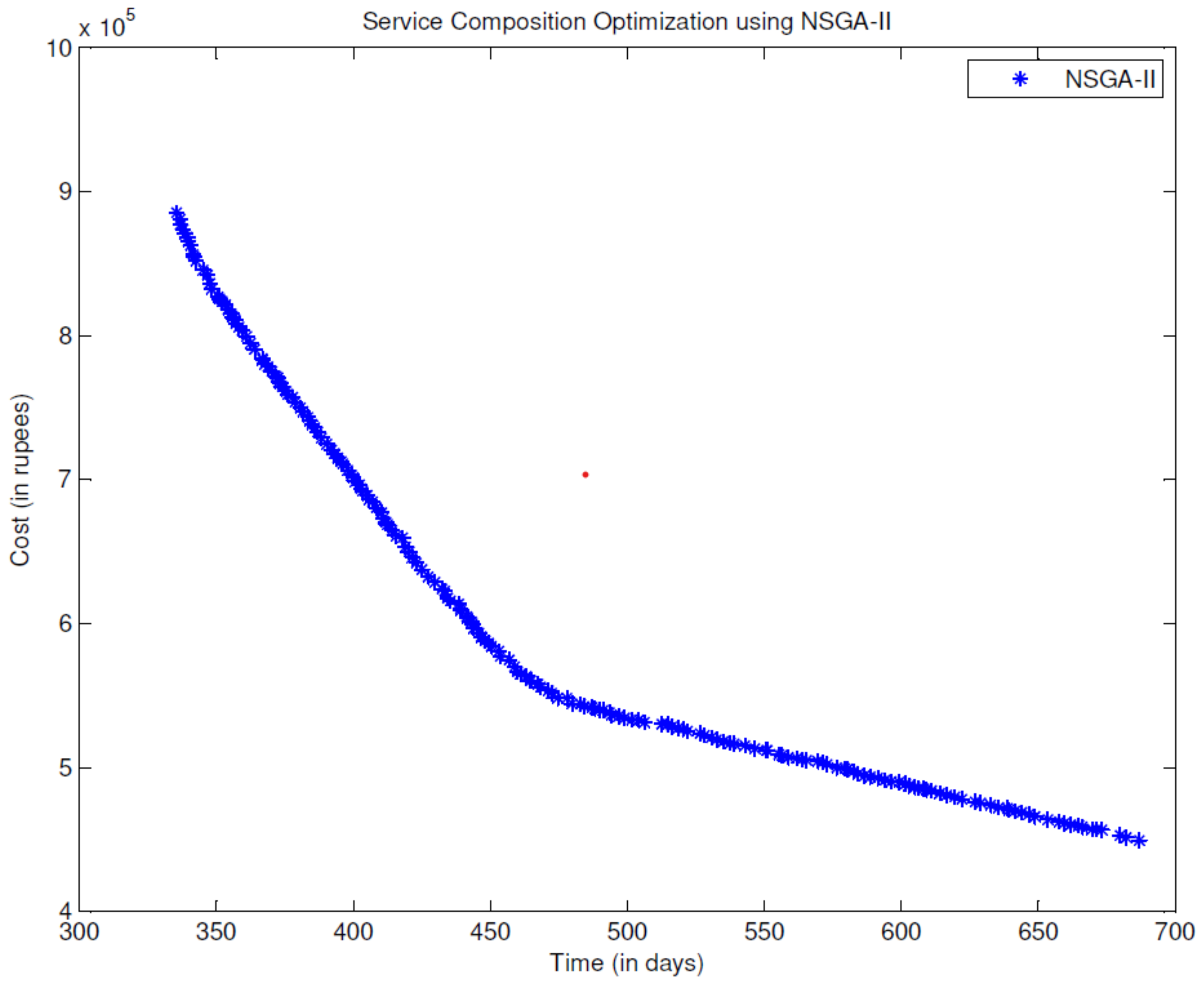


Figure 3.16: Pareto optimal solutions obtained using NSGA-II

This movement demonstrates the effectiveness of the NSGA-II algorithm by effectively minimizing both cost and time. The graph shows a successful balance between the competing goals with an equitably dispersed set of trade-off points along the Pareto front. The algorithm's supremacy in resolving multi-objective optimization problems is confirmed by the solution's closeness to the origin, which shows that it consistently finds optimal configurations.

A thorough statistical analysis is included in Table 3.5 to support this graphical representation and provide additional insight into the algorithm's robustness and performance.

Table 3.5: Statistical analysis

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
NSGA-II	Time	686.9	335.6	101.6	474.5	449.9	335.6	351.3
	Cost	8.854e+05	4.494e+05	1.301e+05	6.206 e+05	5.853 e+05	4.494e+05	4.361 e+05

3.7 Linear Service Composition Optimization using MOGSK

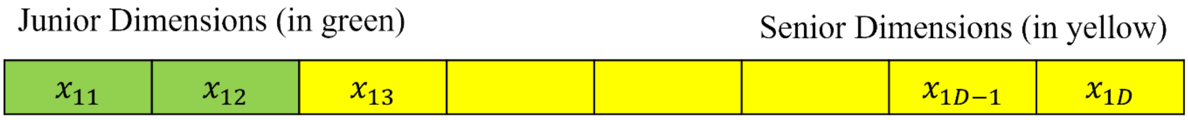
This part covers a human-inspired evolutionary computational algorithm known as the Gaining sharing knowledge-based algorithm (GSK) for optimizing the composed services of smart agriculture applications.

3.7.1 Optimization Algorithm: MOGSK

GSK is a revolutionary optimization approach inspired by human strategies, has been created recently. It adheres to the concept of acquiring and disseminating information globally to a human being. GSK mostly depends on two crucial phases: Junior-gaining-sharing knowledge (JGSK) phase and Senior-gaining-sharing knowledge (SGSK) phase. Everyone acquires knowledge and then imparts it to others along with their own opinions. Early on in life, humans learn from their small social networks of friends, neighbors, and family. Out of a natural curiosity to learn more about other people in the population, they try to share what they have learned and their opinions with others who may not be from their social networks. However, they may lack the knowledge or expertise to categorize the citizens of their area. In line with the same idea, people in their middle or subsequent years attempt to learn more by interacting with a larger network, including social media acquaintances, coworkers, and friends, and seek out ways to share their thoughts and opinions with those who can use it the best. Those beings possess the requisite expertise to categorize and swiftly rate individuals as being either good or wicked [109]. The earlier mentioned process can be explained mathematically step-wise as follows-

Step 1: Initially, population size is defined (Here, assumed to be N_P) and it is randomly initialized. Let x_i $where\ i = \{1, 2, 3, \dots \dots, N_P\}$ be the population's individuals. Each individual x_i can be defined as $x_{ij} = \{x_{i1}, x_{i2}, x_{i3}, x_{i,4}, \dots \dots, , x_{iD}\}$, where D is the domain of knowledge that an individual is provided with, defining its dimensions. Furthermore, the corresponding fitness values of individuals are defined by $f_i, where\ i = \{1, 2, 3, \dots \dots, N_P\}$. All concepts of junior gaining sharing (JGS) and senior gaining sharing (SGS) are illustrated in Figure 3.17 (a) and (b), respectively using a vector x_{ij} [111].

(a)



(b)

Figure 3.17: (a) Vector x_{ij} $for\ i = 1$ during JGSK phase (b) Vector x_{ij} $for\ i = 1$ during SGSK phase [111]

Two important conclusions have been drawn from Figure 3.17. First, the number of updated dimensions utilizing the JGS strategy throughout the JGSK phase is larger than the number of updated dimensions utilizing the SGS strategy. Second, the number of updated dimensions for each vector during the senior phase using the SGS strategy is larger than the number of updated dimensions using the JGS strategy. Additionally, the magnitude of the knowledge rate (k), which must also be considered when calculating the necessary number of dimensions that will be substituted using both phases, will control the amount of knowledge that will be passed down through generations using JGS and SGS strategies. Another parameter is the knowledge factor (k_f) (any real number > 0) that controls the entire acquired and shared knowledge to be incorporated to the current generation of individuals over the course of generations and knowledge ratio (k_r) (any number between 0 and 1 including them) that controls the entire gained shared knowledge to be passed down over generations [110].

Step 2: Then, the dimensions of each phase are calculated using the formula in Equations 3.17 and 3.18 given below-

$$D_{junior} = (problemsize) * \left(\frac{Gen-G}{Gen}\right)^{k} \tag{3.17}$$

$$D_{senior} = (problemsize) - D_{junior} \tag{3.18}$$

Here, G is the ongoing generation.

Gen is describing the total number of generations.

D_{senior} and D_{junior} are the dimensions of the senior and junior phases, respectively.

Step 3: JGSK Phase

Because of curiosity and a desire to learn about others, each person tries to learn from the closest and most reliable individuals who are part of small groups while also attempting to provide knowledge to someone who does not belong to or is not a member of any group.

At this phase, each person tries to learn from the most reliable and closest people who are part of small groups while simultaneously trying to impart knowledge to someone who is not connected to or is not a part of any group out of eagerness and a desire to learn about others. Accordingly, utilizing the junior strategy, upgrading each individual can be calculated as follows:

a) Sort each person in descending order by their objective function value:

$$x_{best}, \dots \dots x_{i-1}, x_i, x_{i+1}, \dots \dots, x_{worst}$$

b) Next, choose two other individuals (the closest individuals) who are better (x_{i-1}) or worse (x_{i+1}) than the existing individual to establish the knowledge-gaining source. Additionally, choose another person at random (x_r) to serve as a knowledge-sharing source [111].

Step-4: SGSK Phase

Utilizing the information that is already available and relevant expertise from the best, better, and worst individuals within a given community are the main goals of this phase. Utilization refers to the influence and result of others—both good and bad—on an individual. Thus, by using the senior strategy updating each individual can be calculated as follows:

a) All people are ranked based on objective function in ascending order, and then they are separated into categories: best individual, better individual, and worst individual.

b) Then to form the gaining part, two vectors are randomly chosen from the top and bottom $200p\%$ individuals of the present population and for sharing part, the third vector is chosen from the middle $N_P - (2 * 200p\%)$. This process is repeated for each individual, x_i. The pseudocodes for both JGSK and SGSK phases are shown in Figures 3.18 and 3.19, respectively [112].

Algorithm: Junior Gaining Sharing Knowledge Phase

```
Begin
  For i = 1: N_P
  For j = 1:D
      If rand ≤ k_r
        If f(x_i) < f(x_r)
          x_ij^new = x_i + k_f * [(x_{i-1} − x_{i+1}) + (x_r − x_i)]
        else
          x_ij^new = x_i + k_f * [(x_{i-1} − x_{i+1}) + (x_i − x_r)]
        End (if)
    Else x_ij^new = x_ij^old
    End (If)
End
End
```

Figure 3.18: Pseudocode for junior gaining sharing phase

Algorithm: Senior Gaining Sharing Knowledge Phase

Begin
  For i = 1: N_P
  For j = 1:D
    If $rand \leq k_r$
      If $f(x_i) < f(x_m)$
        $x_{ij}^{new} = x_i + k_f * \left[\left(x_{p-best} - x_{p-worst}\right) + (x_m - x_i)\right]$
      else
        $x_{ij}^{new} = x_i + k_f * \left[\left(x_{p-best} - x_{p-worst}\right) + (x_i - x_m)\right]$
      End (if)
    Else $x_{ij}^{new} = x_{ij}^{old}$
    End (If)
End
End

Figure 3.19: Pseudocode for senior gaining sharing phase

In the flowchart illustration of Figure 3.20, the proposed MOGSK's entire process is depicted.

3.7.2 Proposed Framework

In this multi-objective optimization, fast nondominated sorting, crowding distance, and the Pareto dominance relation are used to generate those nondominated solutions, which promote diversity, enhance exploitation and exploration, help to increase coverage, and hasten convergence to the Pareto solutions. The proposed framework is displayed in Figure 3.21.

Its working involves the initialization of parameters like population size, number of generations, knowledge rate, knowledge ratio, and knowledge factor. The entire population is then randomly initialized, and the evaluation of each individual's fitness value follows. On the initial population, fast nondominated sorting is employed to obtain the non-dominated plus sorted solutions according to distinct fronts and crowding distance. MOGSK then upgrades the junior/senior population state just like GSK. Until the final requirement of the maximum number of iterations is met, these procedures are carried out repeatedly.

These steps are continued until the end condition of the maximum number of iterations is settled.

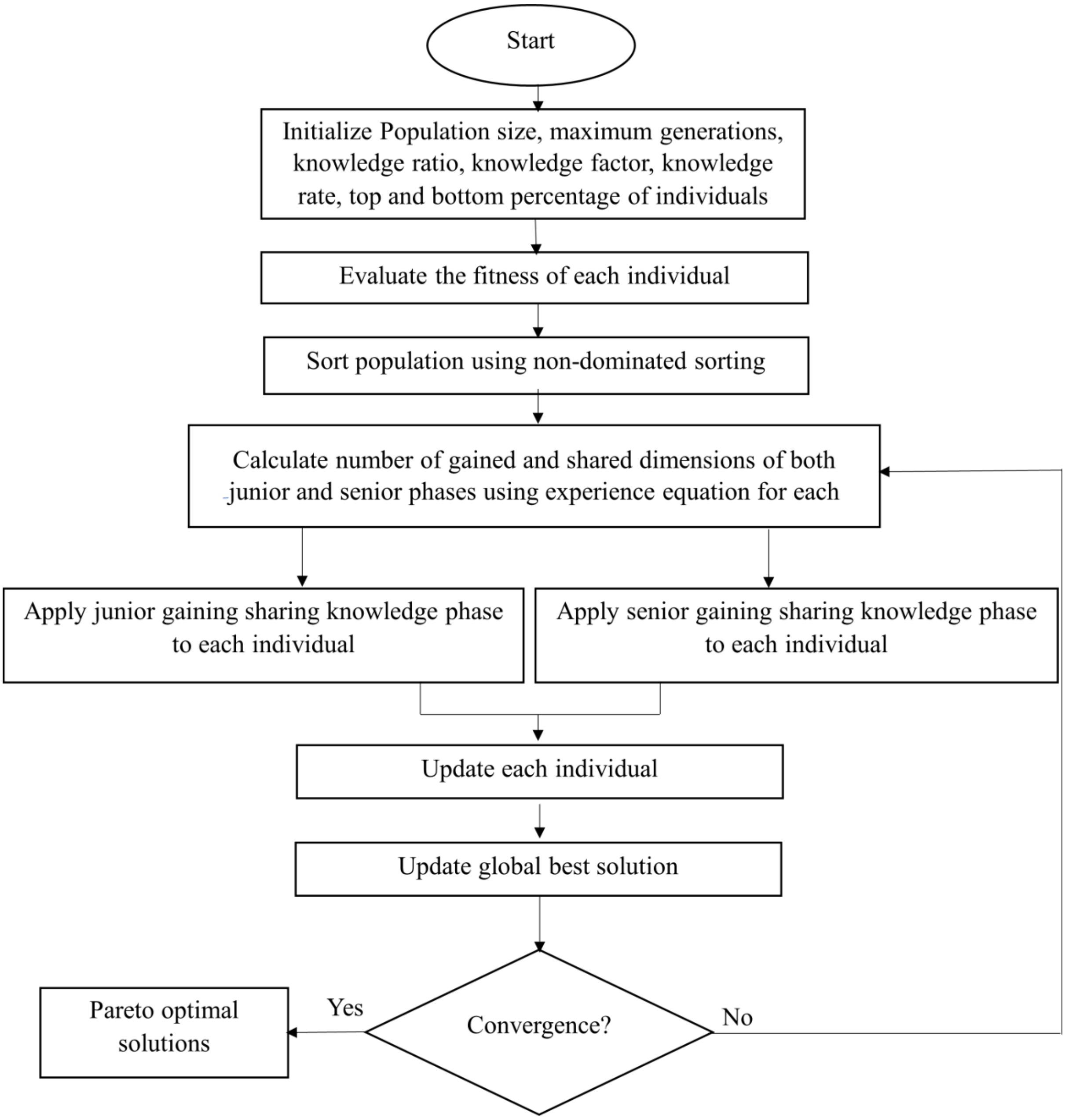


Figure 3.20: Flow chart illustration for MOGSK algorithm

3.7.3 Simulation Setup

The goal of the service composition optimization problem is to minimize the dual objectives of cost and time, with the fitness function directing the search. The parameters used to assess the algorithm's efficacy are listed in detail in Table 3.6. When the trade-off considerations persist unchanged after three iterations—typically within 1000 iterations—the search process is said to be over.

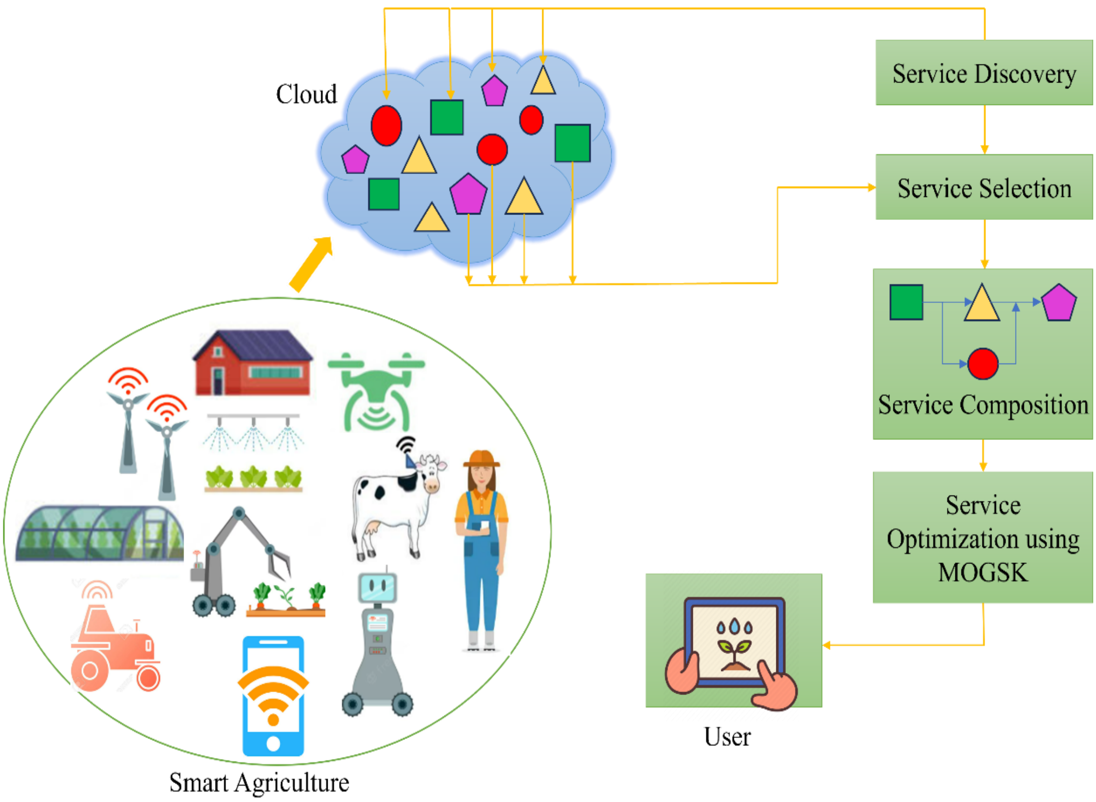


Figure 3.21: Proposed framework for optimization using MOGSK

Table 3.6: Simulation parameters

Parameters	Values
Population size	200
No. of iterations	1000
Knowledge rate	10
Knowledge factor	0.5
Knowledge ratio	0.9

3.7.4 Results and Discussions

The Pareto optimum solutions found following the simulation are depicted in the graph in Figure 3.22. The solutions are shown to be getting closer to the coordinate axes, indicating that time and cost have been reduced at the same time. The trade-off between these two goals is highlighted by this convergence toward the origin, where each point denotes a distinct time-cost balance. The MOGSK algorithm's efficacy in managing such multi-objective problems is further supported by the even distribution of the Pareto front, which indicates that it may produce a wide range of optimal solutions.

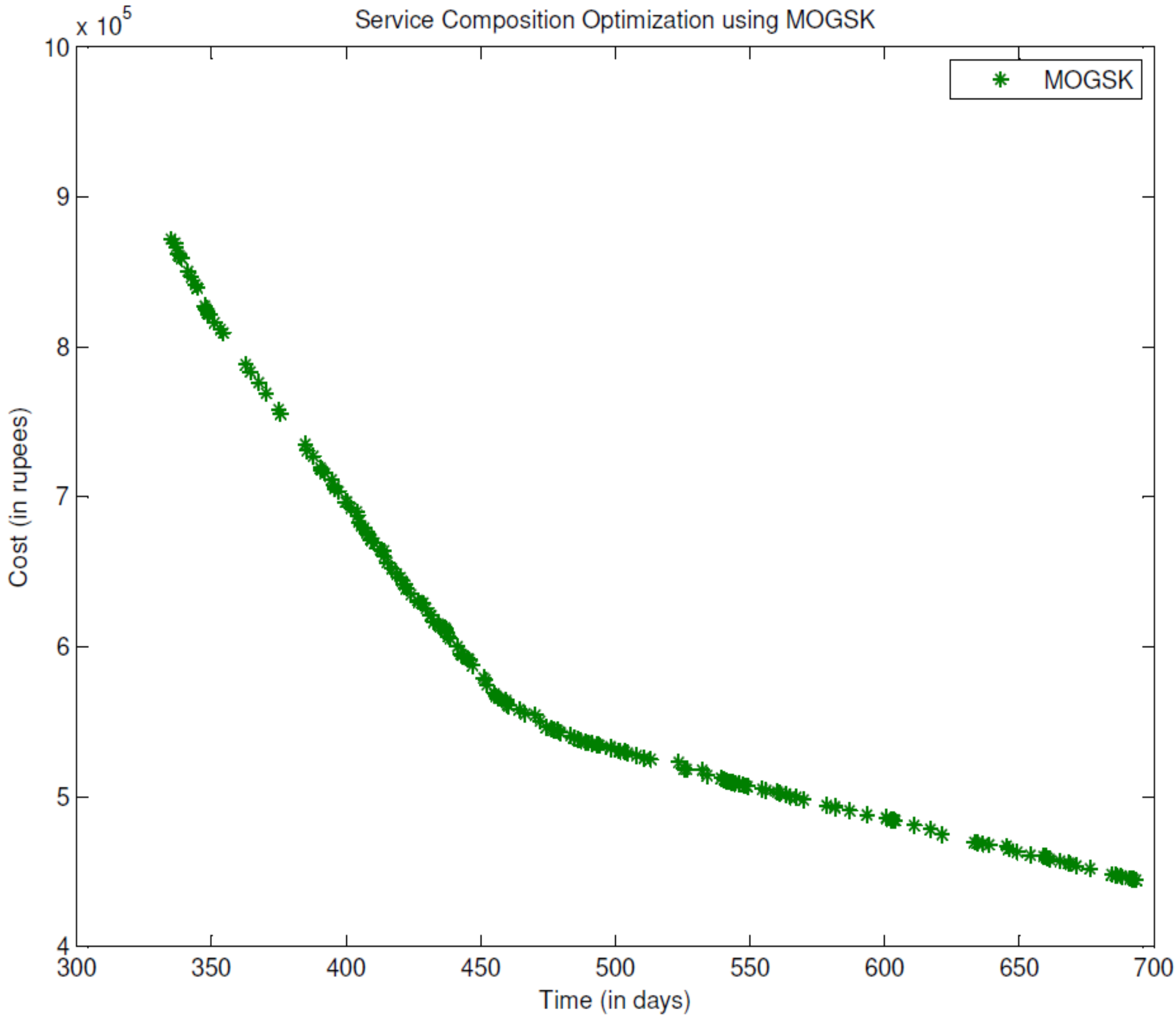


Figure 3.22: Pareto optimal solutions obtained using MOGSK

For an accurate view of the results obtained, statistical analysis has been tabulated in Table 3.7.

Table 3.7: Statistical analysis

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
MOGSK	Time	693.4	335	99.59	487.2	465.5	335	358.5
	Cost	8.72e+05	4.444e+05	1.169e+05	5.935e+05	5.568e+05	4.444e+05	4.276e+05

3.8 Comparison of EC Algorithms

This section compares the Pareto optimal solutions derived from the service composition optimization problem in smart agriculture using three different EC approaches. Time and cost minimization are the main goals of this optimization, and their interaction is governed by a linear relationship. The effectiveness of each algorithm in balancing these two competing goals is the main focus of this investigation. There are two ways to decide which algorithm is performing better for multi-objective optimization problems. One is the Pareto front analysis and another is statistical analysis.

3.8.1 Pareto Front Analysis

Pareto front analysis involves graphing the solutions generated by each algorithm, with the axes representing competing objectives, such as cost and time. The resulting visual representation allows for the examination of trade-offs between these objectives. The Pareto front, composed of non-dominated solutions, illustrates the optimal compromises attainable. By examining and contrasting the configurations and distributions of Pareto fronts produced by different algorithms, its capacity to deliver diverse and optimal solutions can be assessed. Generally, a superior algorithm generates a Pareto front that is nearer to the graph's origin, signifying reduced costs and time. Figure 3.23 illustrates the comparison graph of Pareto optimal solutions obtained utilizing MOGSK, NSGA-II, and MOGA.

As illustrated in Figure 3.23, NSGA-II generates a diversified set of solutions than MOGSK and MOGA. A more extensive and diverse Pareto front which is represented by the line connecting the non-dominated solutions indicates superior performance as well as a broader range of efficient solutions in NSGA-II.

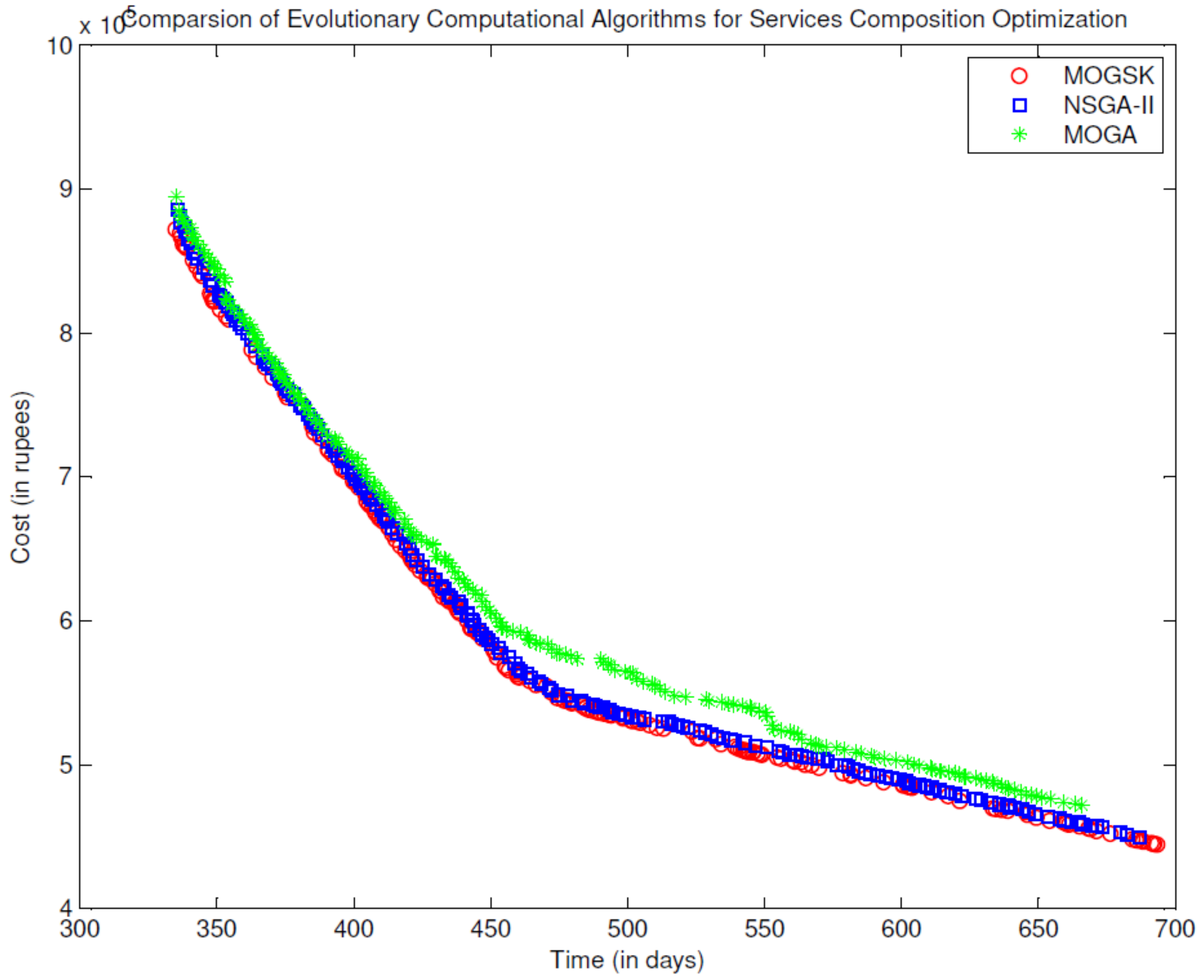


Figure 3.23: Comparison of various evolutionary algorithms for service composition optimization

3.8.2 Statistical Analysis

Pareto front analysis is enhanced by statistical analysis, which offers a numerical method to evaluate algorithm performance. This approach involves computing various metrics for the Pareto optimal solutions, including mean, standard deviation, and range. These measurements aid in quantifying the convergence of solutions produced by each algorithm. For example, a larger standard deviation in costs or times may suggest diverse solutions across multiple runs. Statistical tests can be employed to determine if the observed differences in performance metrics between algorithms are statistically significant, enabling the identification of superior algorithms in generating optimal solutions.

Those numerous performance measures that demonstrate NSGA-II's efficiency and dependability in comparison to the other algorithms are shown in the form of statistical analysis in Table 3.8. It shows a larger standard deviation across its results which depicts the capability of producing a diversified set of Pareto optimal solutions. This suggests that NSGA-II can investigate a broader solution space and offer a wider range of trade-offs that efficiently balance time and cost.

Table 3.8: Statistical analysis of various optimization algorithms

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
MOGSK	Time	693.4	335	99.59	487.2	465.5	335	358.5
	Cost	8.72e+05	4.444e+05	1.169e+05	5.935e+05	5.568e+05	4.444e+05	4.276e+05
NSGA-II	Time	686.9	335.6	101.6	474.5	449.9	335.6	351.3
	Cost	8.854e+05	4.494e+05	1.301e+05	6.206e+05	5.853e+05	4.494e+05	4.361e+05
MOGA	Time	666	335.2	97.48	467.7	445.7	335.2	330.8
	Cost	8.951e+05	4.719e+05	1.25e+05	6.424e+05	6.183e+05	4.719e+05	4.232e+05

3.9 Summary

This chapter discusses the service composition optimization problem in the context of smart agriculture zooming in around two principal objectives of minimizing cost and time. For a more straightforward approach, the relationship is established as linear between these objectives. Three EC approaches—MOGA, NSGA-II, and MOGSK—are used to address this problem. Every algorithm is used to optimize the composition of services while accounting for specific requirements and characteristics of agricultural services. Using a variety of performance metrics, the chapter offers a thorough comparison of these three approach's performances to assess how well they accomplish the optimization goals. According to the results, NSGA-II performs superior to MOGA and MOGSK, showing better outcomes in terms of time and cost minimization. This highlights the algorithm's capacity to more effectively negotiate the trade-offs between the competing goals. Thus, the chapter concludes by emphasizing the significance of applying EC approaches to improve service composition optimization in smart agriculture and proving that NSGA-II is the best approach for this linear multi-objective service composition optimization problem.

CHAPTER-4
NON-LINEAR MULTI-OBJECTIVE SERVICE COMPOSITION OPTIMIZATION IN SMART AGRICULTURE USING EVOLUTIONARY COMPUTATIONAL TECHNIQUES

CHAPTER 4

NON-LINEAR MULTI-OBJECTIVE SERVICE COMPOSITION OPTIMIZATION IN SMART AGRICULTURE USING EVOLUTIONARY COMPUTATIONAL TECHNIQUES

4.1 Chapter Overview

QoS-based service composition is crucial for providing superior and efficient services across diverse interconnected systems. This challenge frequently requires balancing multiple competing goals, such as minimizing duration and expenses while adhering to particular quality criteria for each service element. The intricacy and interrelation of these objectives classify it as an NP-hard problem, making conventional mathematical approaches inadequate for finding optimal solutions in a reasonable timeframe. Thus, comes population-based meta-heuristics in the frame to address this sort of real-world issues.

This chapter presents the idea of optimizing service composition in smart agriculture applications, with a focus on the non-linear relationship between the goals of minimizing cost and time. The emphasis on non-linear relationships reflects real-world complexities, where factors are often more intricate than simple linear correlations. Additionally, the section delves into the optimization of these combined services using three distinct EC techniques, each designed to address the unique challenges presented by non-linear dynamics in service composition.

4.2 Non-linear Service Composition Model

The process of QoS-based service composition entails the selection and combination of individual atomic services to form a composite service that satisfies predetermined QoS standards. These standards typically encompass metrics namely reliability, time, cost, and availability. For every atomic service, multiple candidate services may be available for selection. The candidate service that optimally satisfies the QoS criteria, is selected for incorporation into the composite service. As this study is based on providing an optimal solution for apple orchard establishment and management in distinct regions, suppose there exists a total of "s" services with each having various candidate services represented by "k" along with their corresponding minimum time, maximum time, minimum cost and maximum cost QoS metrics. The entire mathematical framework for this concept is properly explained in equations 3.1 to 3.8, which can be found in section 3.2. Additionally, to characterize the non-linear relationship between time and cost objectives to deal with the non-linearities present in services of real-world scenarios, Lagrange's interpolation method is employed. This approach is elaborated in section 4.2.1.

4.2.1 Basics of Lagrange's Interpolation

Lagrange interpolation is a method for determining a polynomial that precisely matches observed values at specific points. It is the method of choice since it is easy for researchers to calculate and provides accurate estimation [113, 114].

Given "s" distinct services, each of which is associated with a time t_i $(i = 1, 2, 3, \dots, s)$ and a corresponding cost c_i, there exists a total cost "C" for all services. The non-linear relationship between t_i and c_i is defined using Lagrange's polynomial which is expressed from equation 4.1 to 4.2 where equation 4.1 gives the Lagrange's function to calculate the cost of each i^{th}service $(i = 1, 2, 3, \dots, s)$ and equation 4.2 provides the total cost "C" of all service for minimizing objectives.

$$C_i(t) = \sum_{j=1}^{k} c_j \prod_{\substack{m=1 \\ m \neq j}}^{k} \frac{t - t_m}{t_j - t_m} \tag{4.1}$$

$$C = \sum_{i=1}^{s} C_i(t) \tag{4.2}$$

C could be represented in the form of Lagrange's using equation 4.3 given below.

$$C = \sum_{i=1}^{k} c_i . l_i(t) \tag{4.3}$$

This concept of the non-linear relationship between time and cost can be illustrated with the help of Figure 4.1 [119].

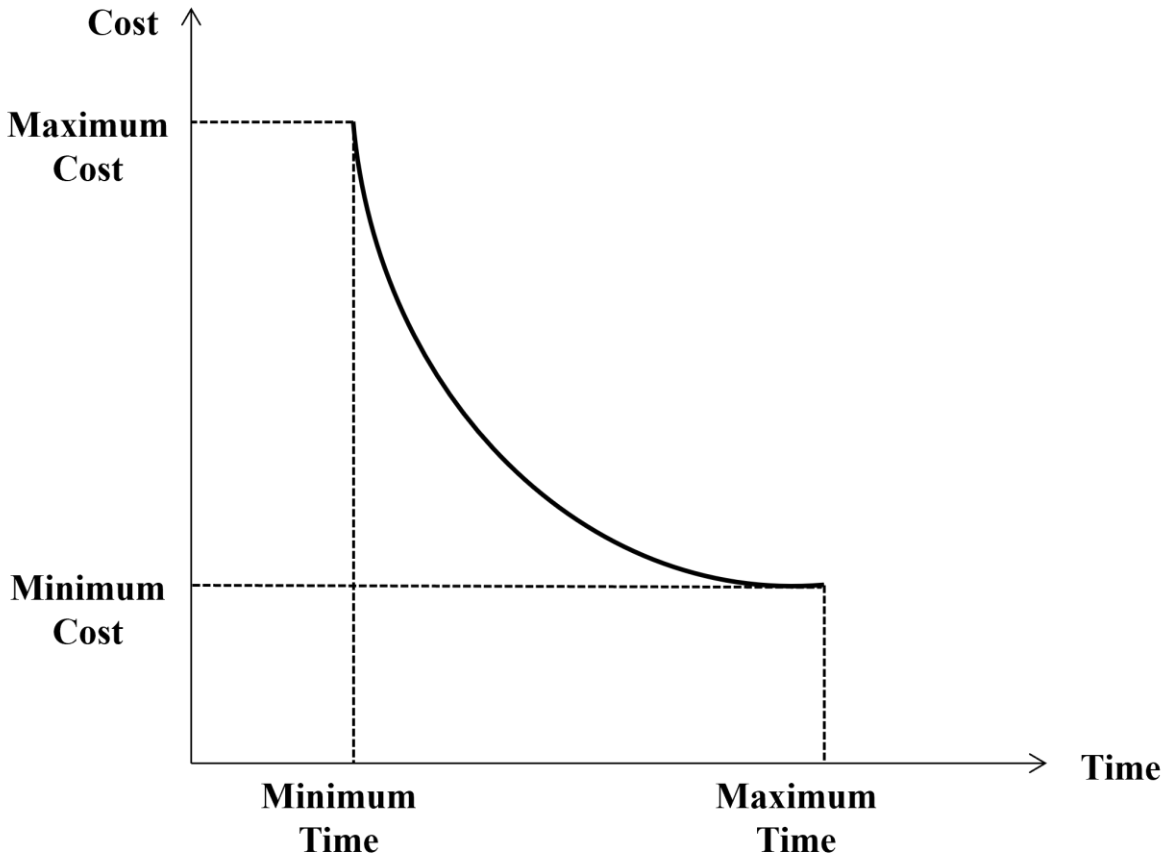


Figure 4.1: Non-linear time-cost trade-off of services [119]

4.3 Non-Linear Dataset Description

Agricultural data sometimes shows complex interactions and non-linear patterns instead of simple, linear tendencies. Thus, this work intends to provide more efficient non-linear optimization, which is more suited to capturing the complex dynamics found in agricultural service data, by modeling these non-linearities. Table 4.1 shows this non-linear dataset and catalogs fourteen basic services that are essential to the establishment and management of apple orchards. Per-acre data is used in this investigation.

Table 4.1: Non-linear dataset showcasing atomic services in smart agriculture

Service Number	Atomic Services	Cost (in rupees)	Time (in days)
1	Soil Testing and Analysis	10000	7

		9500	8
		7000	10
		5700	13
		5000	14
2	Apple Variety Selection	4000	1
		3700	1.5
		3000	2
		2400	2.5
		2000	3
3	Orchard Establishment	200000	30
		174000	45
		125000	54
		65000	77
		50000	90
4	Tree Planting	10000	2
		9600	3
		8200	4
		7400	5
		7000	6
5	Irrigation System Installation	150000	7

		127000	9
		97000	10
		75000	13
		50000	14
6	Fertilizer Application	100000	14
		96000	17
		81000	21
		73000	25
		50000	28
7	Pruning and Training	30000	7
		27000	12
		21000	15
		19000	19
		15000	21
8	Pest and Disease Control	100000	14
		97000	17
		87000	21
		76000	27
		70000	28
9		50000	60

	Crop Monitoring and Management	46000	77
		34000	91
		25000	111
		20000	120
10	Harvesting	70000	14
		68000	19
		49000	23
		41000	25
		35000	28
11	Sorting and Grading	30000	7
		28000	8
		26000	11
		19000	13
		15000	14
12	Packaging and Labelling	90000	14
		88000	17
		76000	22
		69000	26
		60000	28
13		50000	60

	Storage and Cold Chain Management	48000	72
		42000	89
		29000	107
		25000	120
14	Marketing and Distribution	80000	90
		78000	97
		61000	122
		44000	167
		40000	180

Distinct options are available for the user to select any one of them in terms of cost and time to get his/her customized optimal service composition plan. For instance, take service number 5 which is the irrigation system installation. This service offers several choices according to different costs and time duration. The fastest option, which costs 150,000 rupees and takes 7 days to complete, is perfect for people who value time. Under this, an automatic drip system can be installed. There are other less expensive options, including paying 50,000 rupees to have the installation finished in 14 days for a pipe-based system. Time and cost can be balanced with intermediate alternatives like 9 days for 127,000 rupees where a basic sprinkler system can be installed or 13 days for 75,000 rupees where a semi-automated sprinkler system can be installed. So, the users can select the best choice based on their urgency and financial limitations. The same rule follows for other services too. Thus, a customized plan can be made to satisfy each user's particular demands by choosing the best alternative for each service depending on their priorities, including financial limits, and time constraints.

4.4 Methodology for Non-linear Service Composition Optimization

In service composition, meta-heuristic algorithms are commonly employed to find optimal solutions. These methods typically involve two crucial steps: initializing a set of potential solutions and assessing objectives which are time and cost in our case to steer the optimization process. Every algorithm utilized in this work includes both steps.

4.4.1 Population Initialization

A critical step in meta-heuristics is population initialization, which generates a varied set of potential solutions and guarantees a thorough investigation of the solution space. A properly initialized population can greatly increase the algorithm's efficiency by speeding up the convergence rate and the search for optimal solutions. For a population size of "N", each solution can be represented with a string $[t_1, t_2, t_3, \ldots, t_i, \ldots, t_t]$ where min $\_time \leq t_i \leq$ max $\_time$. A comprehensive description of this is provided in section 3.4.1.

4.4.2 Evaluation of Objectives

After the population is initialized, the next step involves evaluating the objectives. This study considers time and cost as minimizing objectives. So, the total time (T) is calculated by taking a summation of all the times associated with various services where the total cost (C) is evaluated using Lagrange's interpolation method (refer section 4.2.1) as both objectives have non-linear relationship between them. For a visual representation, refer to Figure 3.5. Thus, this step creates a population of possible solutions by applying Lagrange's interpolation method to calculate the associated costs with composite services.

The pseudocode of the designated meta-heuristic algorithm is then followed after these two steps to generate Pareto optimal solutions for multi-objective problems.

4.5 Non-Linear Service Composition Optimization using MOGA

This section describes how the composited services with a non-linear relationship between cost and time objectives are optimized using MOGA and named Lagrange's multi-objective genetic algorithm (La-MOGA).

4.5.1 Optimization Algorithm: MOGA

MOGA is an optimization method that imitates genetic processes that occur in nature. It originates with an arbitrarily generated population, with each member consisting of a single chromosome. The fitness function is calculated at every iteration, ensuring optimal performance. Parent selection is crucial, as the fitness of the next generation affects optimizations. Only the fittest chromosomes survive, eliminating unwanted ones. Pareto optimum solutions are those where the population converges after multiple repetitions.

4.5.2 Proposed Framework

The proposed framework is shown in Figure 4.2. This architecture works across multiple IoT tiers. Cloud services are used to store IoT sensor data. Although the functionality of many services is similar, their QoS features are not. Consequently, during the service discovery phase, services with comparable functionality were first found.

Consequently, services with comparable functionality were initially found during the phase of service discovery. Selecting the services that best fit the user's needs from the list of options is the next step. This decision is based on features that are consistent with the time and cost metrics used to measure QoS. A single service cannot handle the user's complicated demands. As a result, the following step completes service composition.

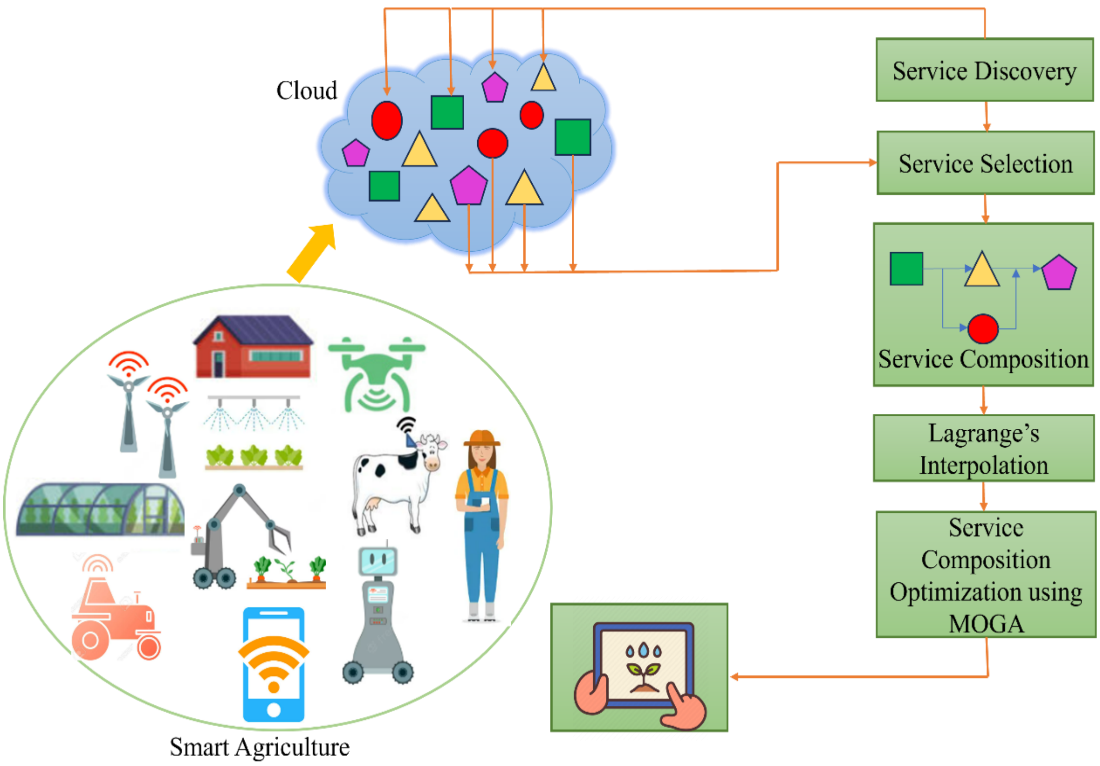


Figure 4.2: Proposed framework for La-MOGA

The next step involves initializing all genetic operators like the size of the population, maximum termination criterion in terms of generations, and probabilities of both crossover and mutation. The cost of each service is then calculated using Lagrange's interpolation during the population initialization step, which corresponds to the generation of random time between each service's maximum and minimum times. The procedure is then carried out by computing the crowding distance and producing non-dominated solutions. Lastly, the offspring is produced by crossing, mutation, and selection processes. The complete process is iterated till the convergence requirement is satisfied. The flow chart depicted in Figure 4.3 shows the stages involved.

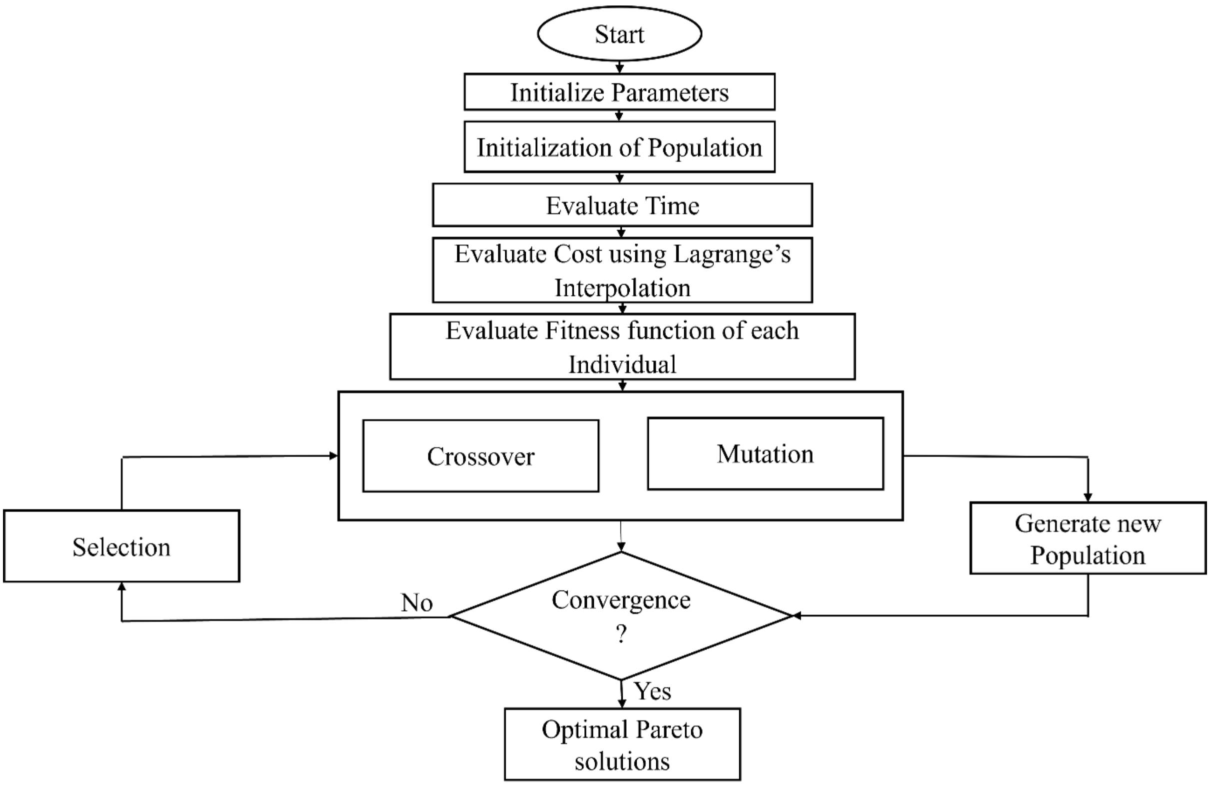


Figure 4.3: Flow chart illustration of La-MOGA

4.5.3 Simulation Setup

The proposed approach is run on a desktop computer with MATLAB R2013a version installed. The details of the simulation parameters used are tabulated in Table 4.2. When the trade-off points stay the same for three consecutive iterations—achieved in 1000 generations—the search is terminated.

Table 4.2: Simulation parameters

Parameters	Values
Population size	200
Number of generations	1000
Crossover type	SBX crossover
Crossover probability (P_c)	0.9
Mutation type	Polynomial mutation
Mutation probability (P_m)	0.07

4.5.4 Results and Discussions

Figure 4.4 displays the Pareto optimal solutions obtained for the service composition optimization using La-MOGA where the impact of non-linearities on cost has been examined which illustrates the real-life scenario of any agriculture problem. According to the analysis, the profile is heading in the direction of the coordinate axes, minimizing time and cost while obtaining trade-off points. These trade-off points depict the various options a farmer can have to choose from as per their requirements.

4.5.5 Comparative Behavioral Analysis of La-MOGA and Li-MOGA

The behavior of the proposed algorithm La-MOGA is evaluated with the Li-MOGA algorithm (a linear time-cost relationship) and shown in Figure 4.5. It can be concluded from Figure 4.5 that both La-MOGA and Li-MOGA provide diversified Pareto solutions. The complexities and non-linearities inherent in real-world smart agriculture systems deter a linear relationship between cost and time objectives. Consequently, the Pareto solutions derived from La-MOGA and Li-MOGA algorithms exhibit marginal disparities due to these non-linearities.

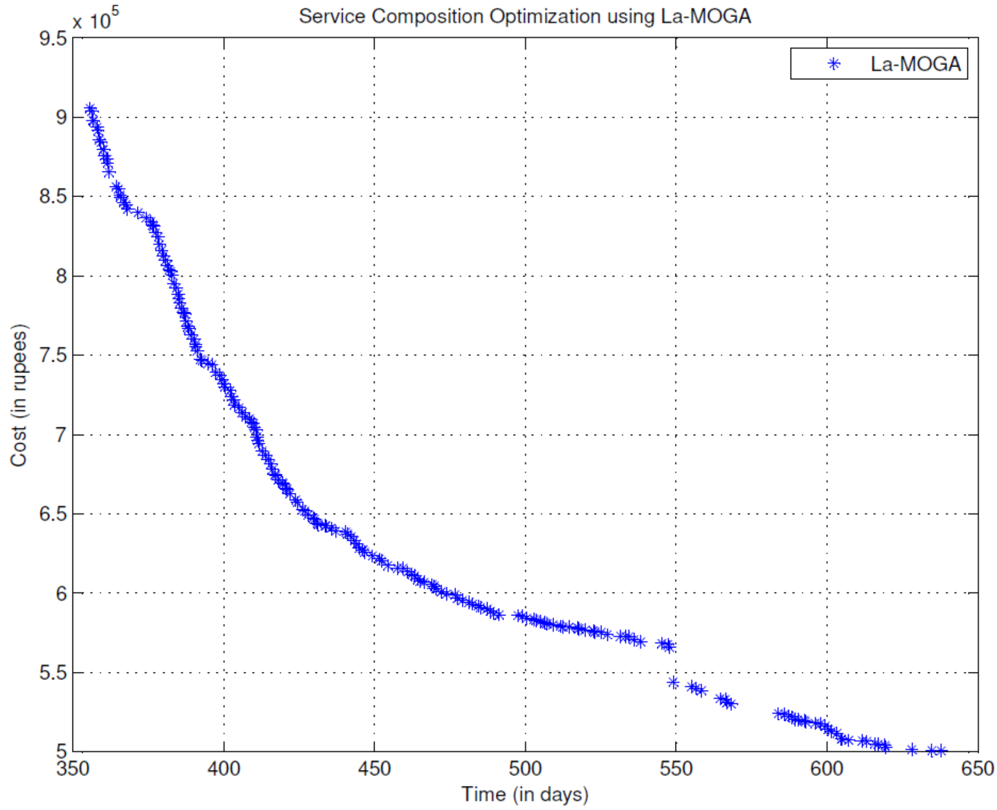


Figure 4.4: Pareto optimal solutions obtained using La-MOGA

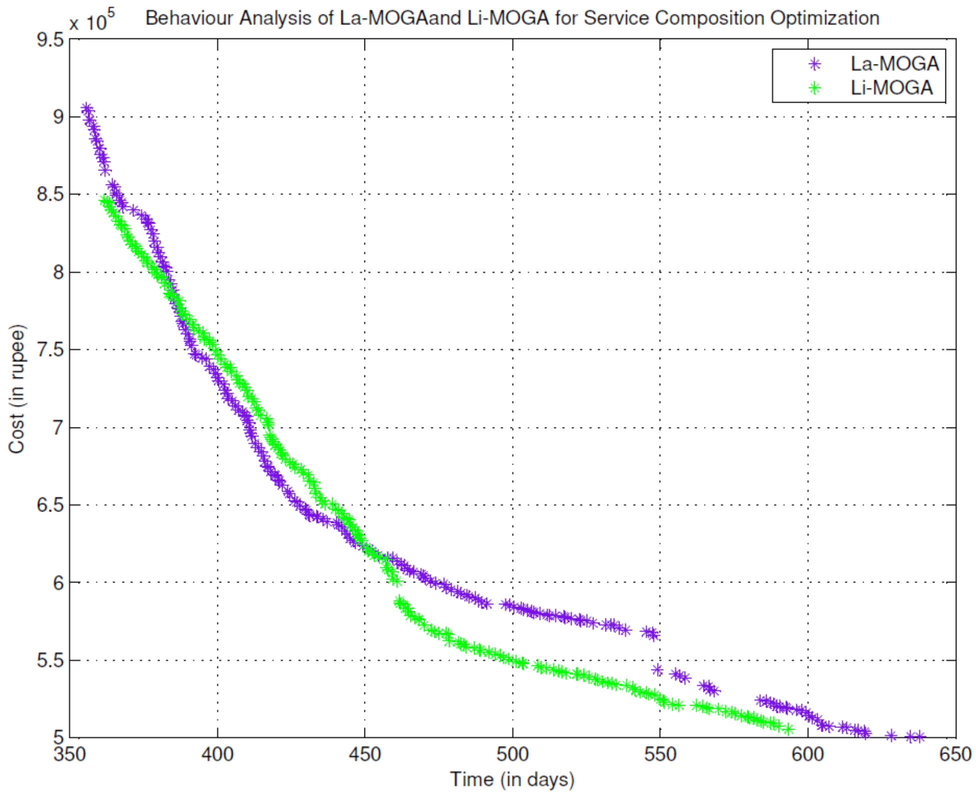


Figure 4.5: Behaviour analysis of La-MOGA and Li-MOGA

Statistical analysis is the best method for fully understanding the findings. Thus, both La-MOGA and Li-MOGA are statistically summarized in Table 4.3.

Table 4.3: Statistical analysis of both La-MOGA and Li-MOGA

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
La-MOGA	Time	637.9	355.7	79.18	457.8	433.8	355.7	282.2
	Cost	9.057e+05	5.007e+05	1.151e+05	6.669 e+05	6.425 e+05	5.007e+05	4.05e +05
Li-MOGA	Time	593.5	361.7	65.4	454.4	444.2	361.7	231.8
	Cost	8.464e+05	5.058e+05	1.07e+05	6.514 e+05	6.407 e+05	5.058e+05	3.406 e+05

4.6 Non-Linear Service Composition Optimization using NSGA-II

This section elaborates on how the service composition with a non-linear relationship between objectives is optimized using NSGA-II and named Lagrange's multi-objective non-dominated sorting genetic algorithm (La-NSGA-II).

4.6.1 Optimization Algorithm: NSGA-II

NSGA-II is a popular meta-heuristic evolutionary algorithm, developed in 2002 by K. Deb. It uses the concept of non-dominated sorting and crowding distance to find uniformly distributed solutions for multi-objective optimizations. The algorithm starts by sorting random individuals, and then forming a parent population using binary tournament selection. After the parent population undergoes crossover and mutation operators to produce offspring, the combined population is used to construct the subsequent population [105].

4.6.2 Proposed Framework

The proposed framework for optimizing service composition involves five layers: sensor, network, cloud, service composition, and application layer. Information from the IoT sensors is collected through the sensor layer, the network layer connects data to servers, the cloud layer offers various sub-services, and the service composition layer composes services based on user demands. The application layer makes these services available to end users, ensuring efficient and effective service composition in the apple crop production process. Figure 4.6 illustrates the proposed framework for La-NSGA-II. The whole concept of the proposed La-NSGA-II is illustrated through a flow chart in Figure 4.7.

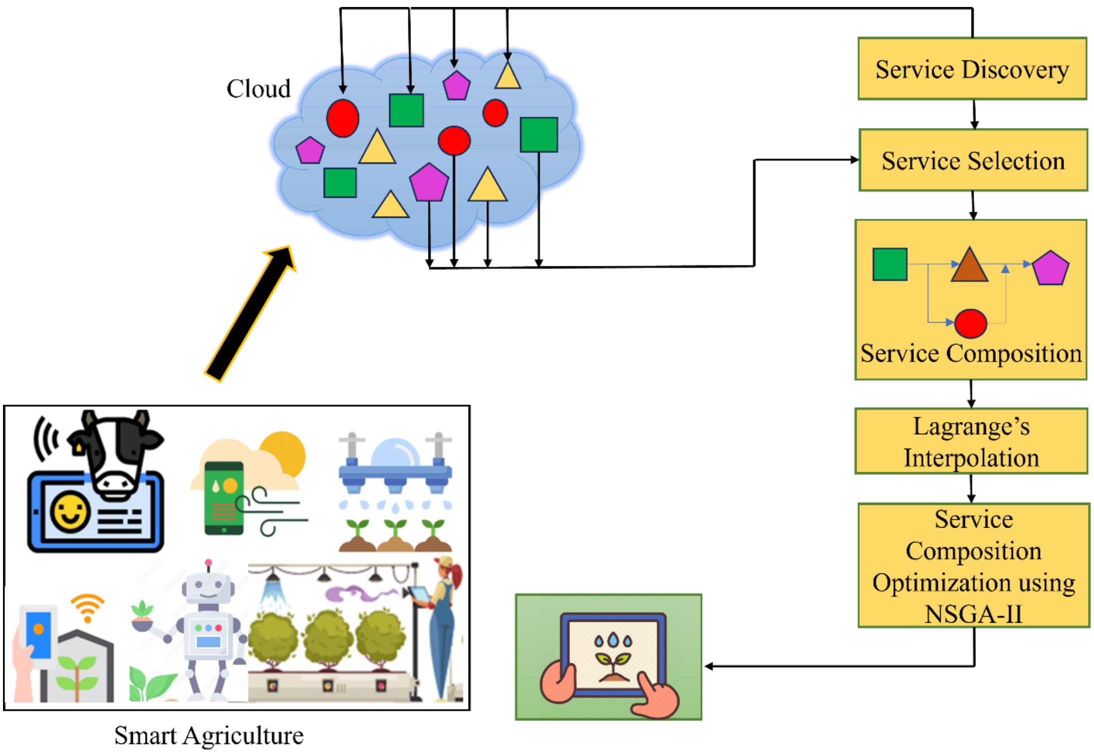


Figure 4.6: Proposed framework for La-NSGA-II

4.6.3 Simulation Setup

The algorithm's effectiveness is assessed using a set of simulation parameters outlined in Table 4.4. The primary goal is to reduce both time and cost across various smart agriculture services. The algorithm continues its search until the balance between these objectives remains stable for three successive iterations.

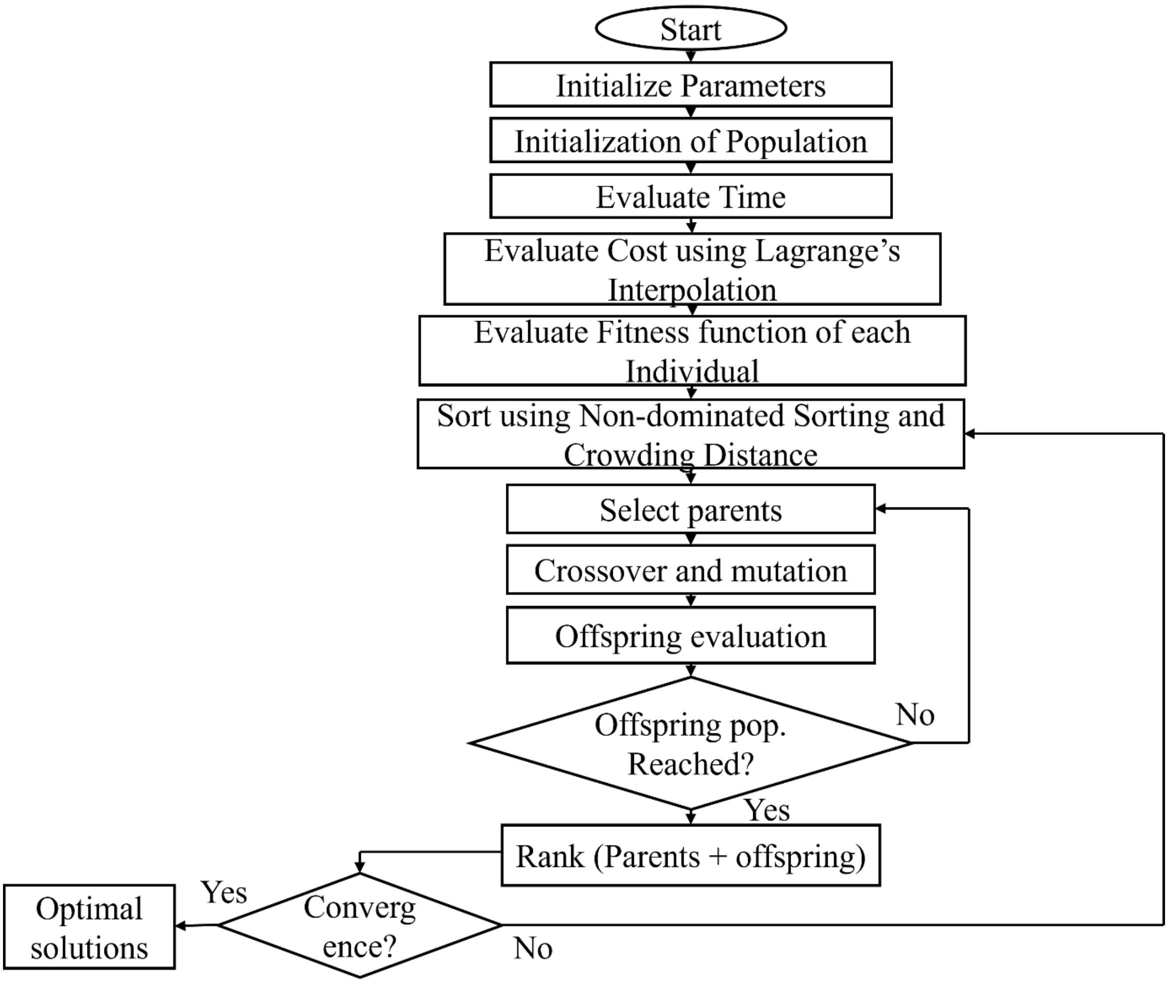


Figure 4.7: Flow chart illustration of proposed La-NSGA-II

Table 4.4: Simulation operators of NSGA-II

Parameters	**Values**
No. of iterations	1000
Population Size	200
Mutation Probability (P_m)	0.07
Crossover Probability (P_c)	0.9

4.6.4 Results and Discussions

The simulation outcomes of service composition optimization using La-NSGA-II produce a range of Pareto-optimal solutions that successfully strike a balance between time and cost factors. A distinct movement toward the coordinate axes is shown in Figure 4.8. This diverse set of options along the Pareto front enables farmers to choose the solution that aligns best with their particular requirements, finding an ideal compromise between time and cost considerations. These findings demonstrate the capability of La-NSGA-II to deliver adaptable and efficient solutions for services in the realm of smart agriculture.

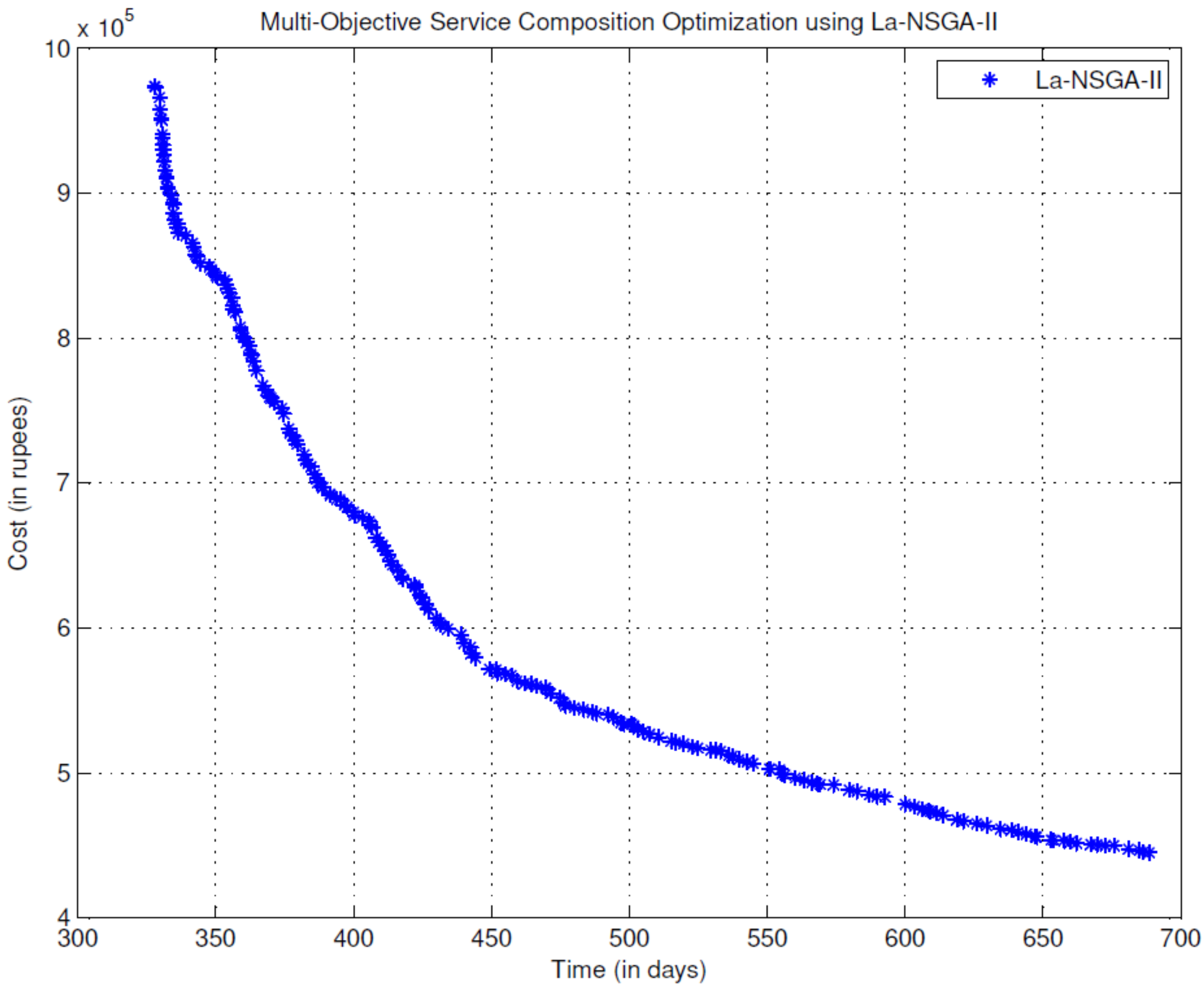


Figure 4.8: Pareto optimal solutions obtained using La-NSGA-II

4.6.5 Comparative Behavioral Analysis of La-NSGA-II and Li-NSGA-II

The behavior of the proposed algorithm La-NSGA-II is evaluated with the Li-NSGA-II algorithm (a linear time-cost relationship) and shown in Figure 4.9. Different solution patterns show how adaptable two algorithms—Li-NSGA-II and La-NSGA-II—are to multi-objective optimization in smart agriculture, with one method using a linear relationship between objectives and the other a non-linear relationship. A smoother, more uniformly distributed

Pareto front is produced by the Li-NSGA-II, emphasizing consistent trade-offs between goals. The intrinsic complexity and non-linearities of real-world smart agriculture scenarios, such as resource interdependencies and changing environmental circumstances, are better captured by the non-linear relationship. Because agricultural service optimization is complex and dynamic, the non-linear approach produces a wide range of Pareto optimal solutions.

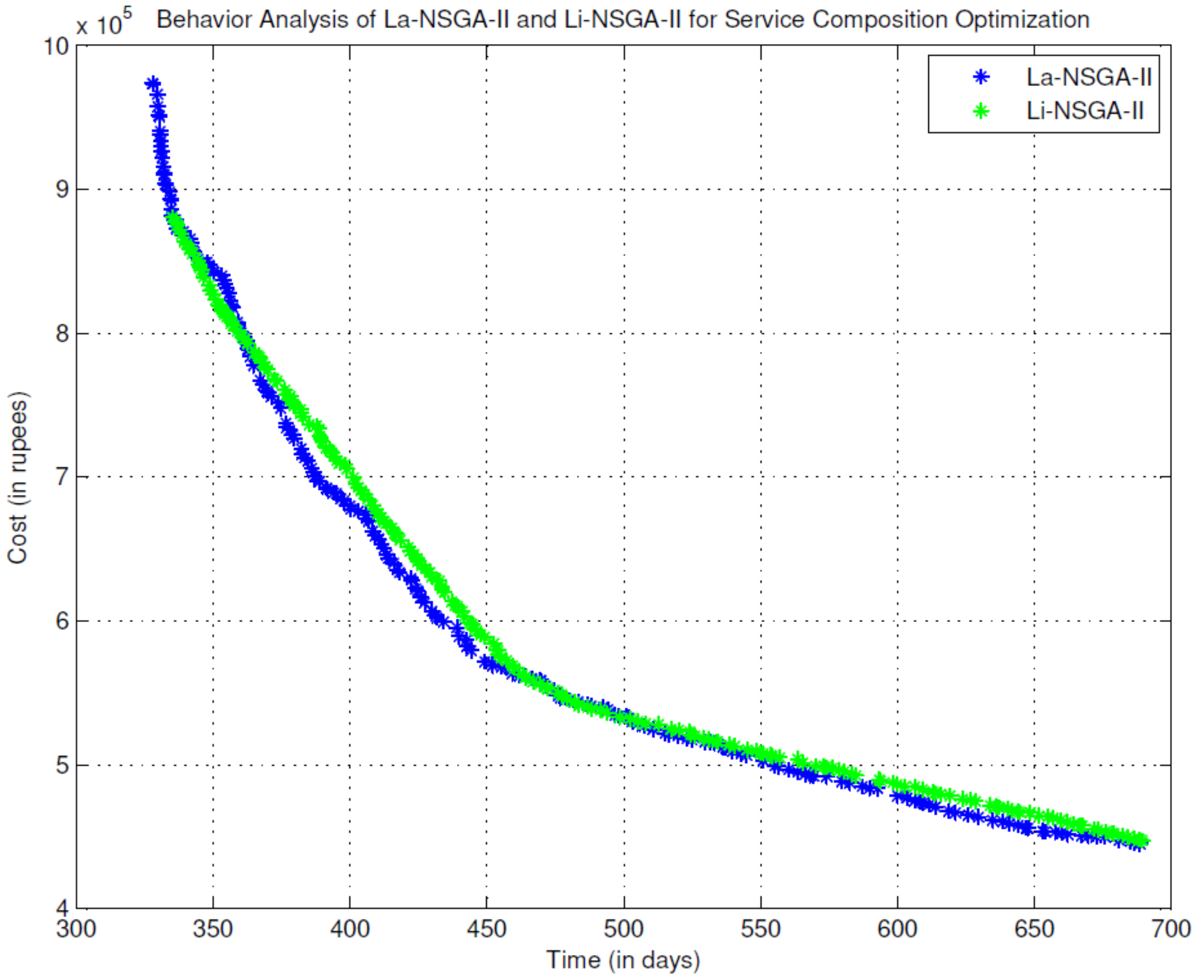


Figure 4.9: Behaviour analysis of La-NSGA-II and Li-NSGA-II

A thorough statistical analysis is provided in Table 4.5 to support the interpretation of the simulation results. This analysis provides a deeper understanding of the algorithm's applicability for real smart agricultural scenarios by highlighting key performance indicators that evaluate each algorithm's behavior and efficacy under linear and non-linear objective relationships.

Table 4.5: Statistical analysis of La-NSGA-II and Li-NSGA-II

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
La-NSGA-II	Time	688.6	328	106.8	456.1	425.3	328	360.6
	Cost	9.737e+05	4.451e+05	1.576e+05	6.504 e+05	6.188 e+05	4.451e+05	5.286 e+05
Li-NSGA-II	Time	689.4	335.1	106.6	476.3	446	335.1	354.3
	Cost	8.805e+05	4.473e+05	1.32e+05	6.21e +05	5.937 e+05	4.473e+05	4.331 e+05

4.7 Non-Linear Service Composition Optimization using MOGSK

The non-linear link between objectives of service composition optimization is addressed in this section using the MOGSK algorithm, an optimization technique inspired by human behavior. It is named Lagrange's multi-objective gaining sharing knowledge-based algorithm (La-MOGSK).

4.7.1 Optimization Algorithm: MOGSK

MOGSK is an optimization algorithm that focuses on acquiring and disseminating global information just as humans do, thereby making it an algorithm based on human behavior. It relies on two phases: JGSK and SGSK. Early on in life, humans learn from small social networks, sharing their knowledge with others. In their middle years, they interact with larger networks, sharing their knowledge and opinions. This process helps them categorize and rate individuals as good or wicked. Knowledge rate, knowledge ratio, and knowledge factor are three crucial factors that are used in both the JGSK and SGSK phases. The quantity of knowledge that will be passed down across the generations using the JGS and SGS strategy will be controlled by the value of the knowledge rate. Another criterion is knowledge factor (any real number > 0) that controls the entire knowledge that has been acquired and disseminated to the present generation of individuals over the course of generations and knowledge ratio (any number between 0 and 1 including them) that controls the entire gained shared knowledge to be passed down over generations [110].

4.7.2 Proposed Framework

The proposed algorithm of La-MOGSK has used the principle of non-dominated sorting and crowding distance to generate non-dominated solutions that promote diversity, enhance exploitation and exploration, help to increase coverage and hasten convergence to the Pareto optimal solutions. The proposed framework is portrayed in Figure 4.10.

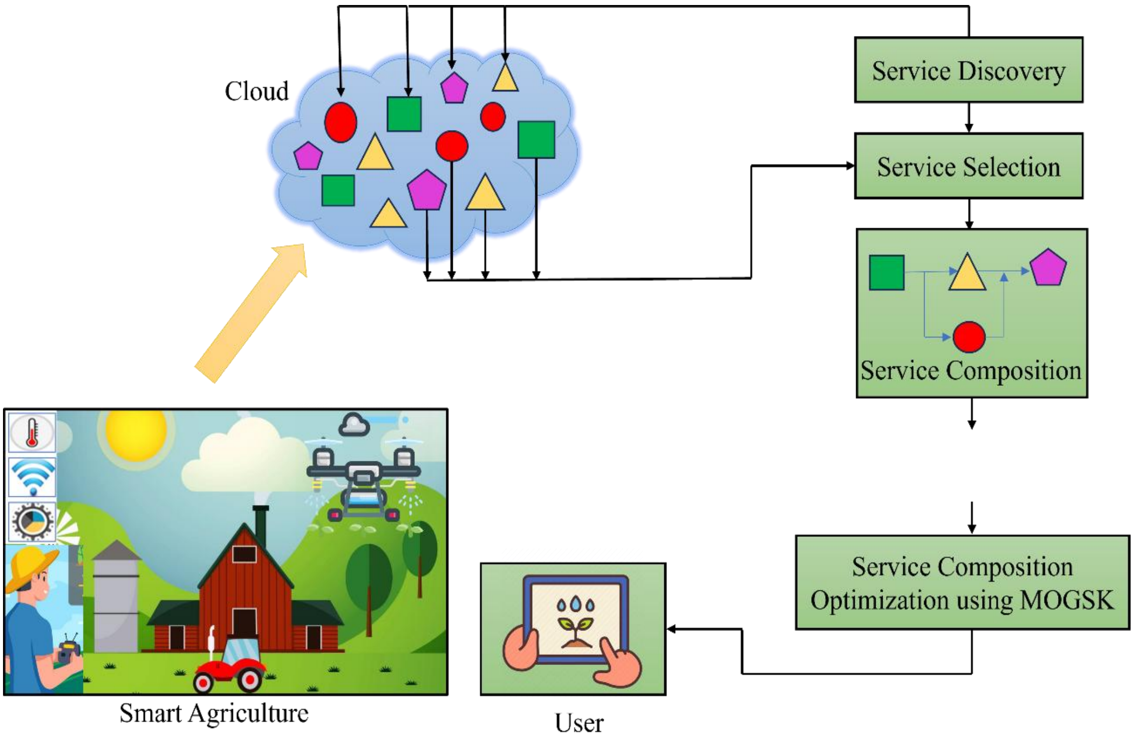


Figure 4.10: The proposed framework for La-MOGSK

For LA-MOGSK, parameters such as number of generations, population size, knowledge rate, knowledge ratio, and knowledge factor are initialized. Following a random initialization of the population, the fitness value of each individual is assessed. The cost objective function is calculated using Lagrange's interpolation method. Non-dominated sorting is utilized on the original population to provide non-dominated plus sorted solutions based on crowding distance and different fronts. Then, La-MOGSK adjusts the junior/senior population status using the junior/senior gaining sharing phase. Until the eventual requirement of the maximum generations is met, these procedures are carried out repeatedly. The flow chart illustration for La-MOGSK is shown in Figure 4.11.

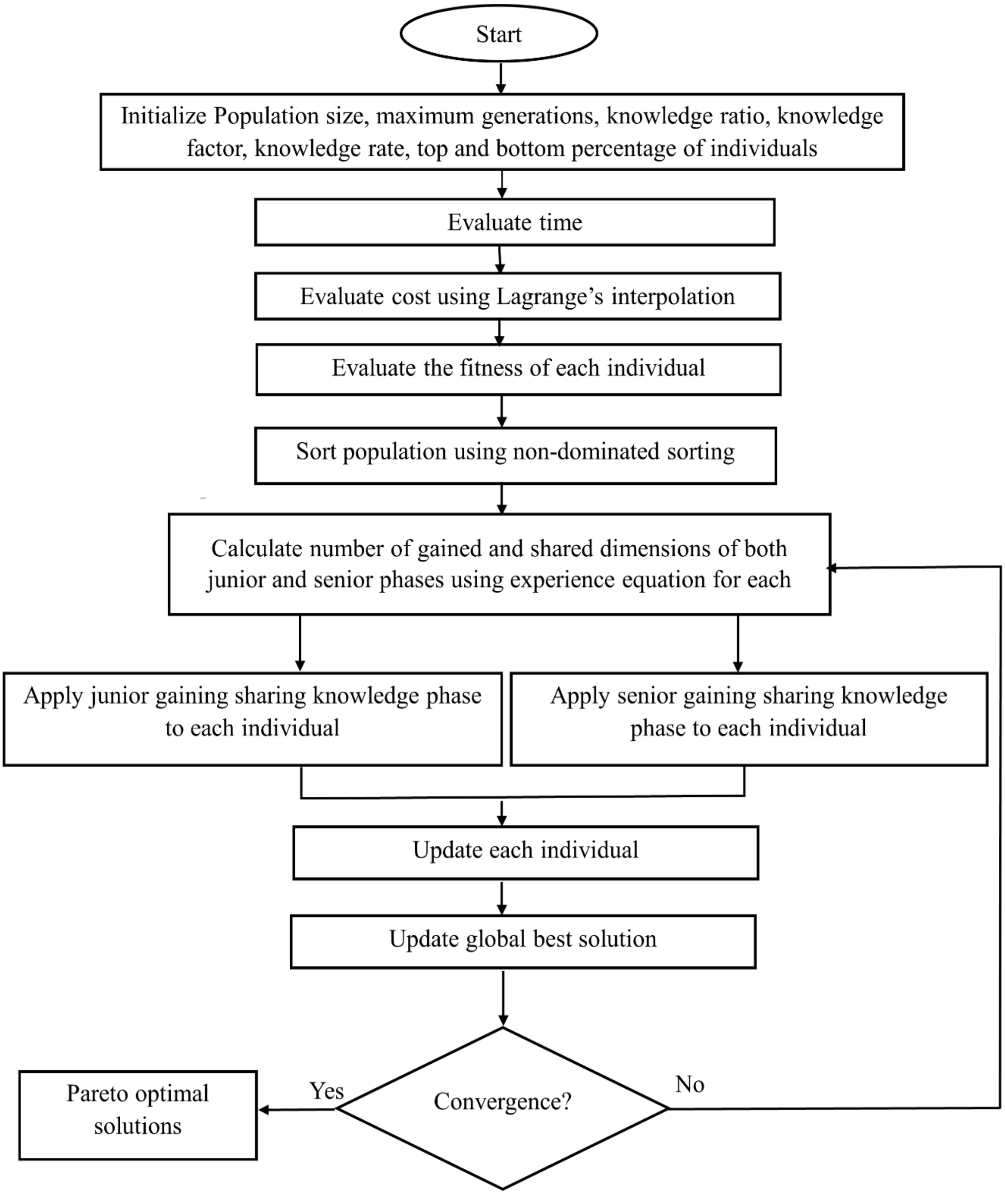


Figure 4.11: The flow chart illustration of proposed La-MOGSK

4.7.3 Simulation Setup

The simulation parameters used for service composition optimization using the proposed framework La-MOGSK are tabulated in Table 4.6. It continues to search in the solution space until solutions remain unchanged for three successive iterations.

Table 4.6: Simulation parameters

Parameters	Values
Population size	200
No. of iterations	1000
Knowledge rate	10
Knowledge factor	0.5
Knowledge ratio	0.9

4.7.4 Results and Discussions

The Pareto optimum solutions obtained using La-MOGSK are illustrated through the graph pictured in Figure 4.12. It is evident that it produces solutions closer to the origin, indicating the successful minimization of both time and cost objectives. A well-optimized set of trade-offs is indicated by the concentration of points nearer the origin, where lower values for both objectives are attained. These results demonstrate the algorithm's capacity to generate superior solutions crucial for decision-makers in intricate optimization scenarios, like smart agriculture, where effective resource allocation is vital.

4.7.5 Comparative Behavioral Analysis of La-MOGSK and Li-MOGSK

The behavior of the proposed La-MOGSK is evaluated with Li-MOGSK (a linear time-cost relationship) and is illustrated in Figure 4.13. It can be observed that both La-MOGSK and Li-MOGSK provide diversified Pareto solutions. The complexities and non-linearities inherent in real-world smart agriculture systems deter a linear relationship between time and cost objectives. Consequently, the Pareto solutions derived from La-MOGSK and Li-MOGA algorithms exhibit marginal disparities due to these non-linearities.

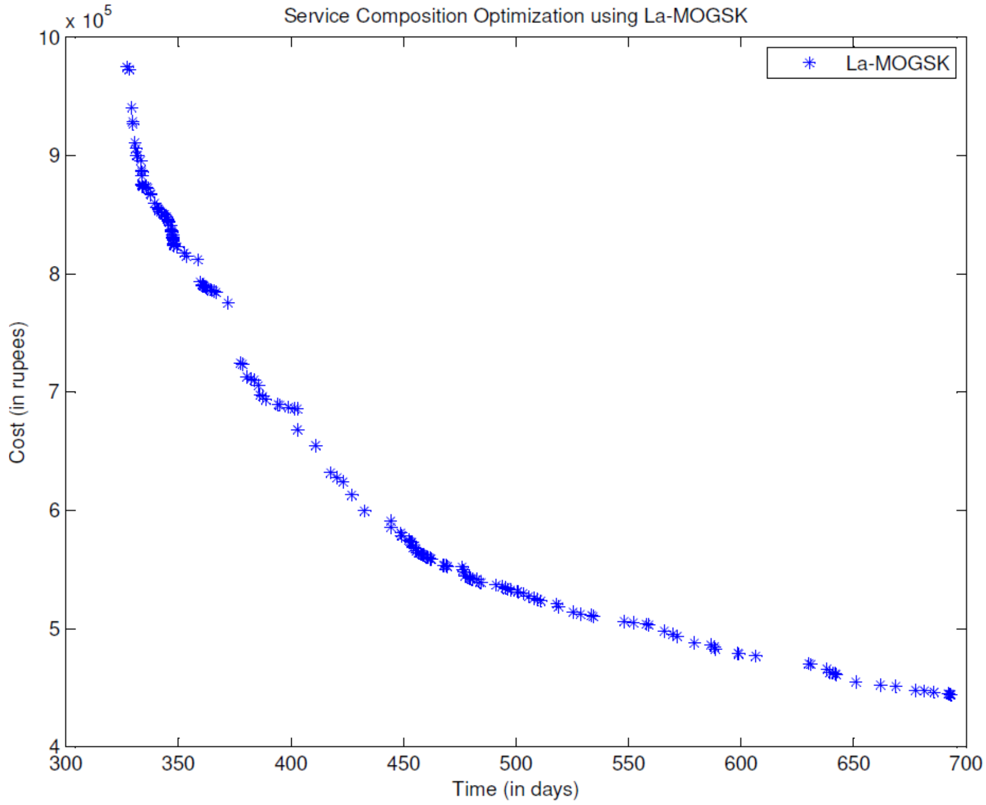


Figure 4.12: Pareto optimal solutions obtained using La-MOGSK

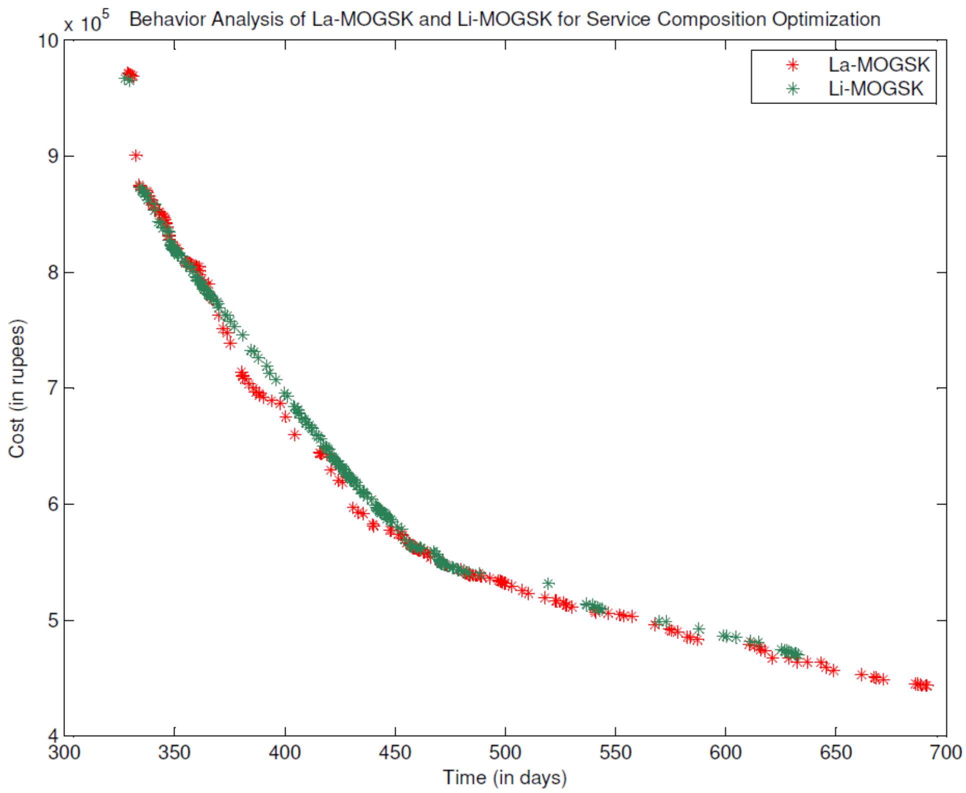


Figure 4.13: Behaviour analysis of La-MOGSK and Li-MOGSK

For an accurate view of the results obtained, statistical analysis has been tabulated in Table 4.7.

Table 4.7: Statistical analysis of both La-MOGSK and Li-MOGSK

Algorithm	**Objectives**	**Maximum**	**Minimum**	**Standard Deviation**	**Mean**	**Median**	**Mode**	**Range**
La-MOGSK	Time	691.2	328.9	98.33	451.8	456.8	328.9	362.3
	Cost	9.715e+05	4.437e+05	1.513e+05	6.443 e+05	5.658 e+05	4.437e+05	5.278 e+05
Li-MOGSK	Time	632.8	327.5	75.06	432.1	426.9	327.5	305.3
	Cost	9.669e+05	4.701e+05	1.203e+05	6.578 e+05	6.298 e+05	4.701e+05	4.969 e+05

4.8 Comparison of EC Algorithms

This section of the chapter analyses the Pareto optimal solutions derived from three distinct EC techniques by considering cost and time as objective functions with a non-linear relationship between them. To ascertain which algorithm performs best, the evaluation uses two techniques: Pareto front analysis and statistical analysis.

4.8.1 Pareto Front Analysis

In Pareto front analysis, the solutions produced by every algorithm are graphed, with the axes signifying conflicting goals like cost and time. It reveals optimal compromises and compares different algorithm's metrics. Plotting the Pareto fronts of various algorithms allows one to determine which strategy offers a better trade-off between goals; convergence to the true Pareto front and diversity of solutions are significant indicators. Normally, superior algorithms produce a Pareto front closer to the graph's origin, indicating reduced costs and time. Figure 4.14 illustrates the comparison graph of Pareto solutions obtained using La-MOGSK, La-NSGA-II, and La-MOGA.

It is evident from Figure 4.14 that the La-NSGA-II produces a more diversified set of solutions than La-MOGSK and La-MOGA. In contrast to La-MOGSK and La-MOGA, the La-NSGA-II algorithms exhibit improved performance by offering diversified solutions along the Pareto front, enabling a wider exploration of the solution space.

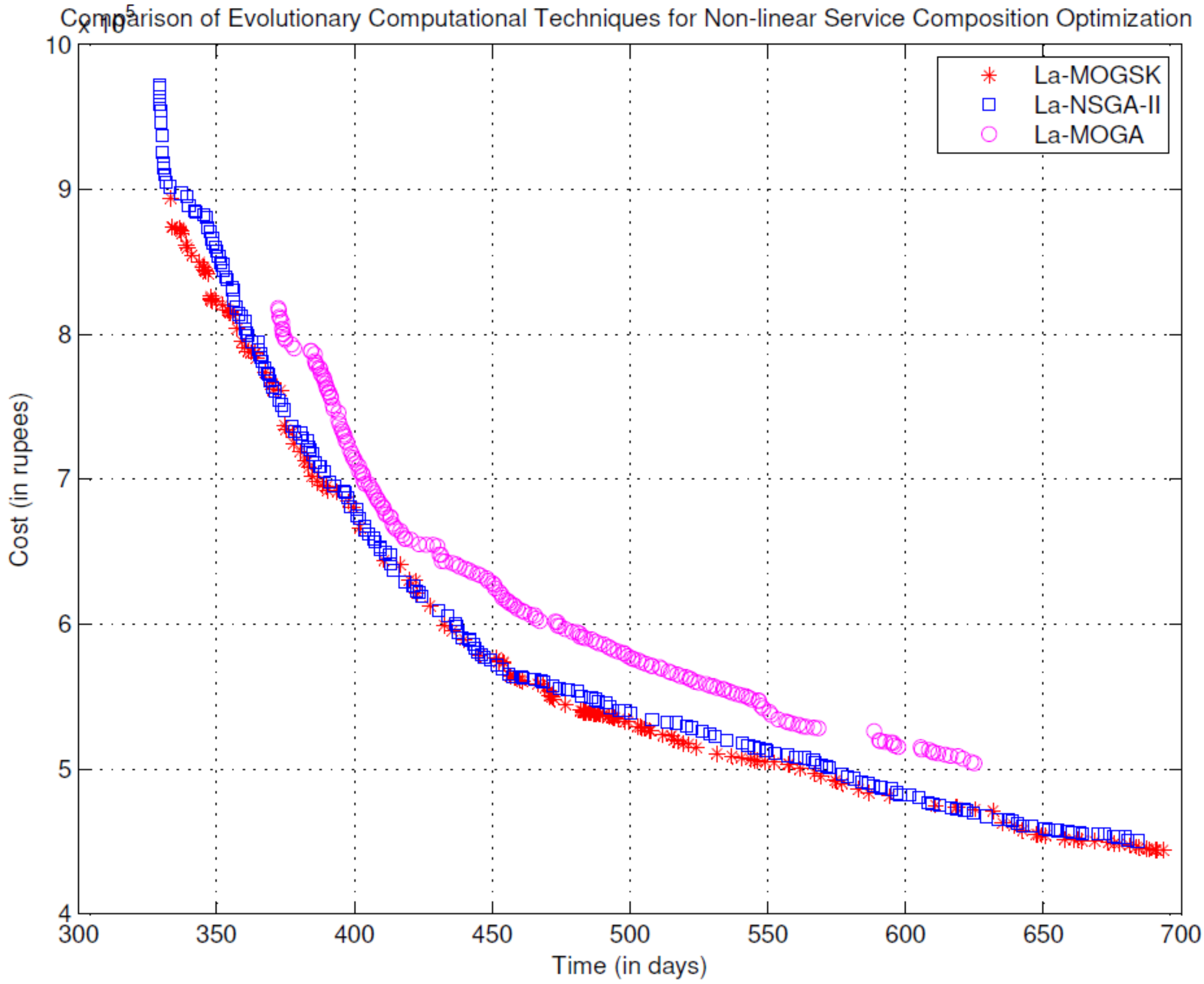


Figure 4.14: Comparison of various evolutionary algorithms for service composition optimization

4.8.2 Statistical Analysis

Statistical analysis is carried out to provide more depth to the algorithm comparison. The mean, standard deviation, and range are among the metrics that are obtained for the Pareto optimum solutions using this method. For instance, in a multi-objective optimization context, a lower mean value typically denotes better performance for time or cost objectives. Table 4.8 presents a statistical analysis of the various performance metrics that illustrate the effectiveness of La-NSGA-II compared to alternative algorithms. Its results exhibit a higher standard deviation, indicating that it is capable of producing a diverse range of Pareto optimal solutions. This implies that La-NSGA-II can efficiently balance time and cost by exploring a larger solution space and providing a greater range of trade-offs.

Table 4.8: Statistical analysis of various optimization algorithms

Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
La-MOGSK	Time	693.5	333.4	103.4	473	482.6	484.8	360.1
	Cost	8.94e+05	4.441e+05	1.397e+05	6.13e+05	5.401e+05	5.385e+05	4.5e+05
La-NSGA-II	Time	684.2	329.3	107.7	465	438.4	329.3	354.9
	Cost	9.716e+05	4.504e+05	1.53e+05	6.415e+05	5.927e+05	4.504e+05	5.213e+05
La-MOGA	Time	624.9	372.4	72.27	467.6	454.5	372.4	252.5
	Cost	8.18e+05	5.041e+05	9.004e+04	6.361e+05	6.174e+05	5.041e+05	3.14e+05

4.9 Summary

This chapter examines the service composition optimization problem in the context of smart agriculture focusing on two major objectives minimizing cost and time. Using the Lagrange interpolation method, a non-linear relationship between the objective functions is constructed to capture the inherent non-linearities involved with such applications. To solve this optimization problem, three evolutionary computation methods are used: La-MOGA, La-NSGA-II, and La-MOGSK. A thorough analysis of the algorithms using a variety of performance criteria shows that La-NSGA-II performs better than La-MOGA and La-MOGSK. The capacity of La-NSGA-II to successfully negotiate trade-off points between competing objectives is demonstrated by this finding. The importance of using evolutionary computation approaches for service composition optimization in smart agriculture is highlighted in the chapter’s conclusion, which also shows that NSGA-II performs superior for non-linear multi-objective service composition optimization.

CHAPTER-5
IMPACT OF UNCERTAINTIES ON BOTH LINEAR AND NON-LINEAR MULTI-OBJECTIVE SERVICE COMPOSITION OPTIMIZATION USING EVOLUTIONARY COMPUTATIONAL TECHNIQUES

CHAPTER 5

IMPACT OF UNCERTAINTIES ON BOTH LINEAR AND NON-LINEAR MULTI-OBJECTIVE SERVICE COMPOSITION OPTIMIZATION USING EVOLUTIONARY COMPUTATIONAL TECHNIQUES

5.1 Chapter Overview

Technological advancements have optimized conventional farming processes, enabling the agriculture sector to meet population growth demands. Selecting the best services out of all the services available is crucial to meeting the user's complex requirements. The composition of those selected services is called service composition and evolutionary optimization is emerging to achieve it. Real-world smart agriculture applications involve many uncertain factors that create obstacles to retrieving critical findings from the data and are a prime concern for modern farmers. Thus, fuzzy set theory has been developed to better manage the intricacies of uncertain data.

This chapter assesses the impact of various uncertain factors that occur in real-world agriculture scenarios on the optimization of composited services. It illustrates how these uncertainties, which range from human to environmental factors, impact the process of proficient service compositions through the use of the NSGA-II algorithm.

5.2 Fuzzy Logic System

In the year 1965, Lofti Zadeh formally established fuzzy logic (FL), a branch of Boolean logic. Contrary to the principles of modal logic, it is a modification of classical set theory. This has the advantage of introducing the idea of confidence to verify an event, enabling it to continue to occur in a state that is not either true or false [115]. It is a more successful method for making decisions to problems because it can mimic human reasoning flexibility and the ability to handle uncertain and non-linear systems. Figure 5.1 illustrates the fuzzy logic architecture [116].

The detailed parts of a typical fuzzy logic system are listed below [116].

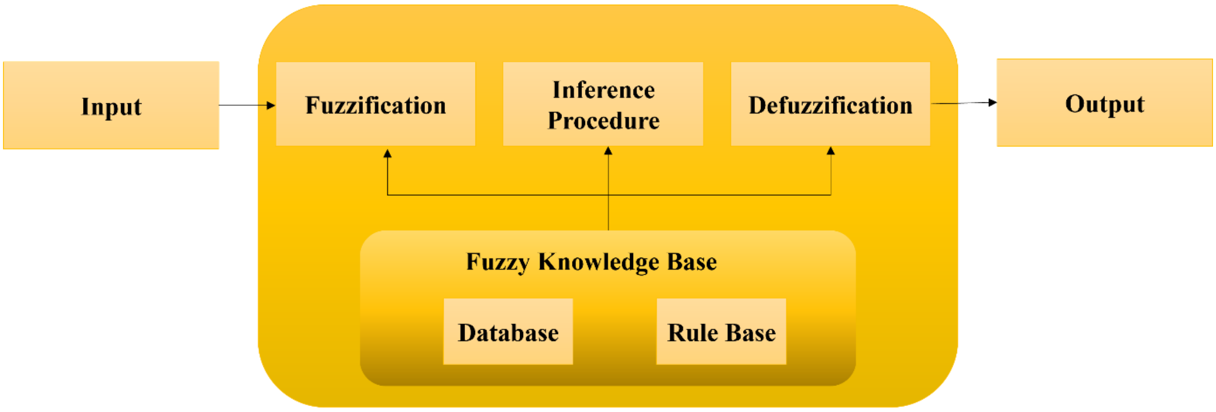


Figure 5.1: Architecture of fuzzy logic [116]

a) Fuzzifier: This segment transforms quantitative numerical input into qualitative linguistic variables by applying a membership function. Although there are many different functions in the literature, the Gaussian, triangular, and trapezoidal functions are the most commonly used ones.

b) Knowledge base: A database and a rule base form the basis of this unit. Databases assign Fuzzy Sets (FSs) to inputs, which FSs subsequently translate into fuzzy membership values. After getting FSs from the database, the rule base builds a set of few rules for rule inference. Stated differently, inference rules are collections of numerous rules that link the system's fuzzy inputs and outputs. These rules appear as "IF-THEN" rules:

IF< Condition-I > OR/ AND < Condition-II > (OR/AND…) Then action on the outputs.

This indicates that rules have an antecedent and consequent structure.

c) Inference Engine: The inference block, which is the central component of FLC, uses fuzzy contribution and inference rules in FL to mimic human reasoning. The numerical processing of these rules can yield the linguistically fuzzy output of the controller. There are two types of FL systems: Mamdani type and Sugeno type. Sugeno is more accurate at approximation, while Mamdani-type is more-effective in interpretation [117].

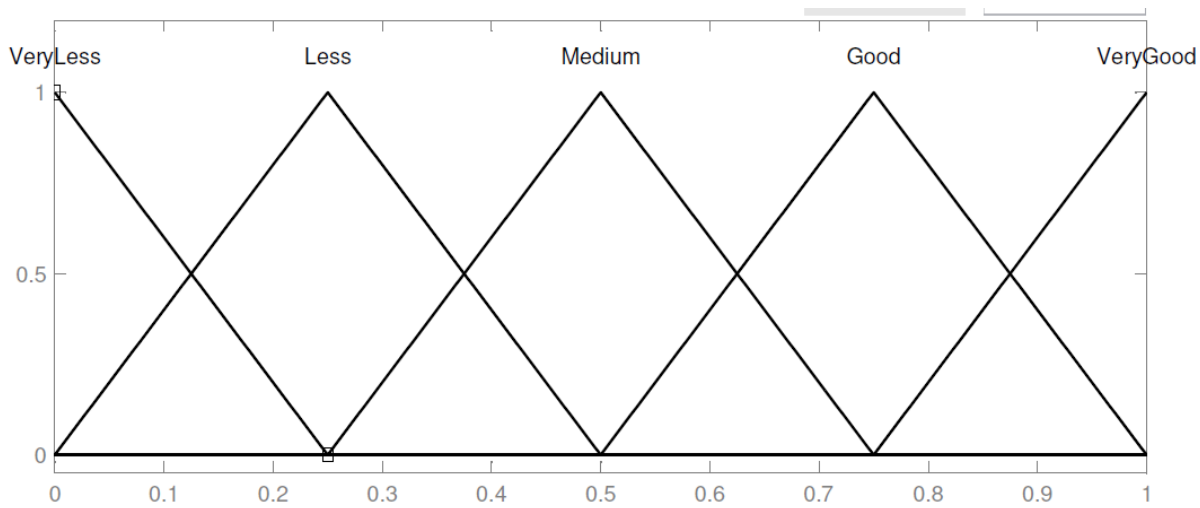
d) Defuzzifier: This part is employed in the defuzzification process.

At this point, the inference engine's multiple commands could be integrated into one cohesive output, transforming the qualitative linguistic variable into numerically-based quantitative data. Center of gravity (COG) and mean of maximum (MOM) defuzzification methods are the two most widely used ones [118].

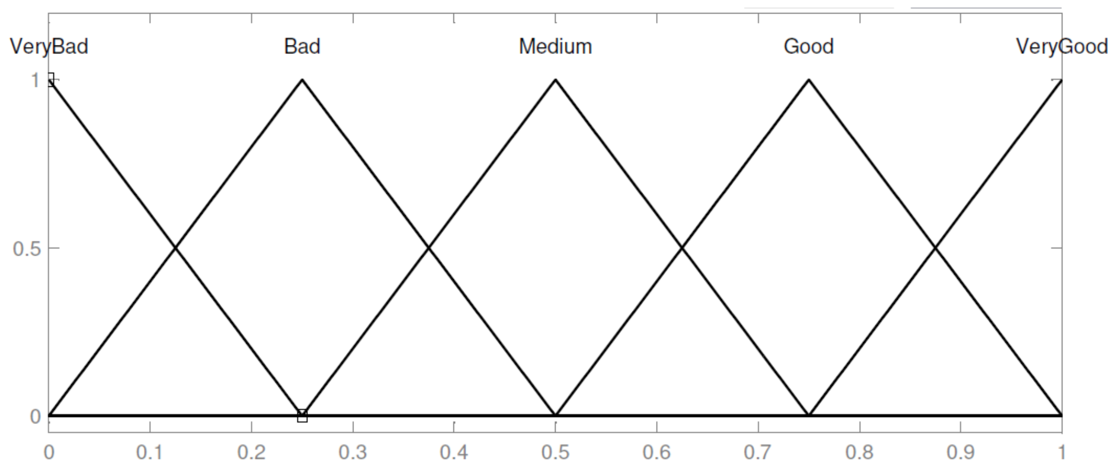
5.2.1 Fuzzy Inference System for Proposed Architecture

This work employs an inference system of the Mamdani type. The input attributes selected for modeling the proposed framework are *Management Skills (MS), Weather Conditions (WC), and Farmer Skills (FS).* Each attribute is partitioned into five variables that are linguistic by utilizing the Mamdani inference system. For the various combinations of input attributes, time and cost are taken as output. Five fuzzy sets, *VeryLess, Less, Average, Good,* and *Excellent*, characterize the first input; *VeryPoor, Poor, Fair, Good, and VeryGood*, describe the second input & *VeryLow, Low, Medium, High, and VeryHigh*, describe the third input. Seven linguistic values have been obtained for the outputs: *VerySmall, Small, SmallMedium, Medium, LargeMedium, Large* and *VeryLarge* for the cost; *VerySmall, Small, SmallMedium, Medium, LongMedium, Long* and *VeryLong* for the time. These membership functions are defined over the "Universe of Discourse (UOD)". It is presumed that the range of UOD for cost and time is $C \pm 0.2C$ and $T \pm 0.2T$, respectively.

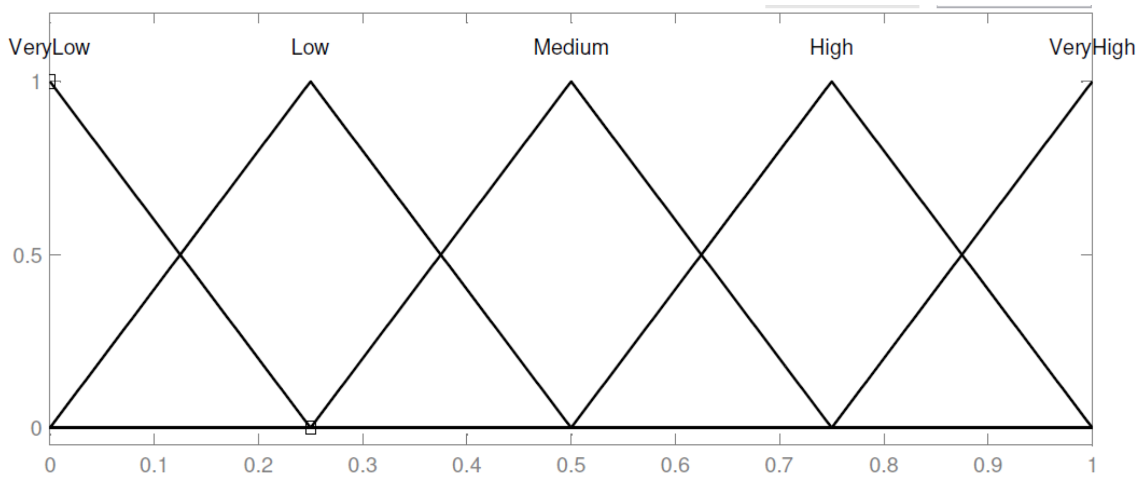
The specifics of the input membership functions are displayed in Figure 5.2.



(a)



(b)



(c)

Figure 5.2: Input membership functions (a) Management skills (b) Weather conditions (c) Farmer skills

The mathematical formulation of each membership function for input attribute *Management Skills (MS)* is provided in equations 5.1 to 5.5 given below.

$$\mu_{VLess}(u) = \begin{cases} 0 & ; \quad u < 0 \\ \dfrac{0.25 - u}{0.25} & ; \quad 0 \leq u \leq 0.25 \\ 0 & ; \quad u > 0.25 \end{cases} \tag{5.1}$$

$$\mu_{Less}(u) = \begin{cases} 0 & ; \quad u < 0 \; or \; u > 0.5 \\ \dfrac{u}{0.25} & ; \quad 0 \leq u < 0.25 \\ \dfrac{0.5-u}{0.25} & ; \quad 0.25 \leq u \leq 0.5 \end{cases} \tag{5.2}$$

$$\mu_{Medium}(u) = \begin{cases} 0 & ; \quad u < 0.25 \; or \; u > 0.75 \\ \dfrac{u-0.25}{0.25} & ; \quad 0.25 \leq u < 0.5 \\ \dfrac{0.75-u}{0.25} & ; \quad 0.5 \leq u \leq 0.75 \end{cases} \tag{5.3}$$

$$\mu_{Good}(u) = \begin{cases} 0 & ; \quad u < 0.5 \; or \; u > 1 \\ \dfrac{u-0.5}{0.25} & ; \quad 0.5 \leq u < 0.75 \\ \dfrac{1-u}{0.25} & ; \quad 0.75 \leq u \leq 1 \end{cases} \tag{5.4}$$

$$\mu_{VGood}(u) = \begin{cases} 0 & ; \quad u < 0.75 \\ \dfrac{u-0.75}{0.25} & ; \quad 0.75 \leq u \leq 1 \\ 0 & ; \quad u > 1 \end{cases} \tag{5.5}$$

For each membership function of input attributes *Weather Conditions (WC)* and *Farmer Skills (FS)*, the corresponding equations are defined from equations 5.6 to 5.10 and 5.11 to 5.15, respectively.

$$\mu_{VBad}(u) = \begin{cases} 0 & ; \quad u < 0 \\ \dfrac{0.25-u}{0.25} & ; \quad 0 \leq u \leq 0.25 \\ 0 & ; \quad u > 0.25 \end{cases} \tag{5.6}$$

$$\mu_{Bad}(u) = \begin{cases} 0 & ; \quad u < 0 \; or \; u > 0.5 \\ \dfrac{u}{0.25} & ; \quad 0 \le u < 0.25 \\ \dfrac{0.5 - u}{0.25} & ; \quad 0.25 \le u \le 0.5 \end{cases}$$

(5.7)

$$\mu_{Medium}(u) = \begin{cases} 0 & ; \quad u < 0.25 \; or \; u > 0.75 \\ \dfrac{u - 0.25}{0.25} & ; \quad 0.25 \le u < 0.5 \\ \dfrac{0.75 - u}{0.25} & ; \quad 0.5 \le u \le 0.75 \end{cases}$$

(5.8)

$$\mu_{Good}(u) = \begin{cases} 0 & ; \quad u < 0.5 \; or \; u > 1 \\ \dfrac{u - 0.5}{0.25} & ; \quad 0.5 \le u < 0.75 \\ \dfrac{1 - u}{0.25} & ; \quad 0.75 \le u \le 1 \end{cases}$$

(5.9)

$$\mu_{VGood}(u) = \begin{cases} 0 & ; \quad u < 0.75 \\ \dfrac{u - 0.75}{0.25} & ; \quad 0.75 \le u \le 1 \\ 0 & ; \quad u > 1 \end{cases}$$

(5.10)

$$\mu_{VLow}(u) = \begin{cases} 0 & ; \quad u < 0 \\ \dfrac{0.25 - u}{0.25} & ; \quad 0 \le u \le 0.25 \\ 0 & ; \quad u > 0.25 \end{cases}$$

(5.11)

$$\mu_{Low}(u) = \begin{cases} 0 & ; \quad u < 0 \; or \; u > 0.5 \\ \dfrac{u}{0.25} & ; \quad 0 \le u < 0.25 \\ \dfrac{0.5 - u}{0.25} & ; \quad 0.25 \le u \le 0.5 \end{cases}$$

$$(5.12)$$

$$\mu_{Medium}(u) = \begin{cases} 0 & ; \quad u < 0.25 \; or \; u > 0.75 \\ \dfrac{u - 0.25}{0.25} & ; \quad 0.25 \leq u < 0.5 \\ \dfrac{0.75 - u}{0.25} & ; \quad 0.5 \leq u \leq 0.75 \end{cases}$$

$$(5.13)$$

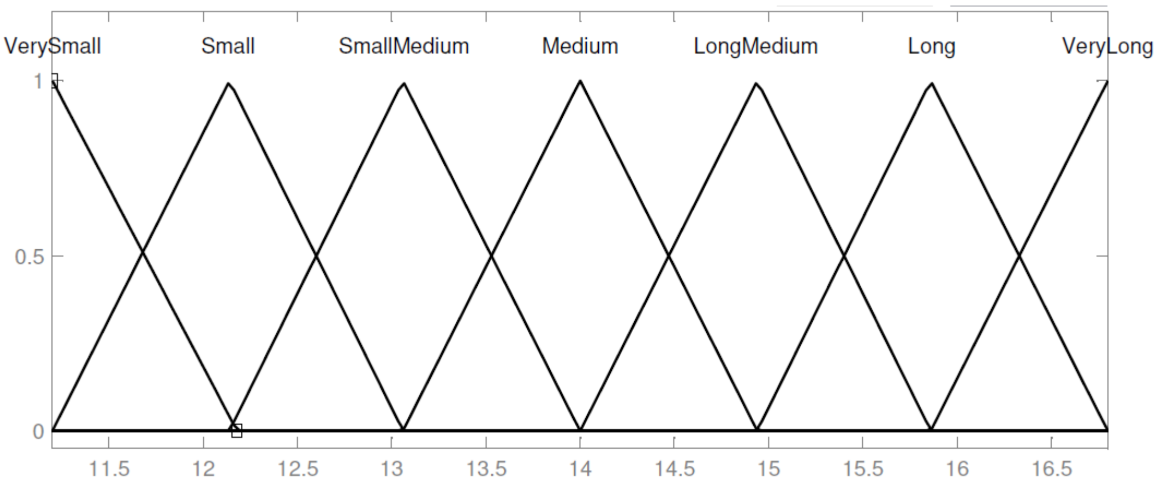
$$\mu_{High}(u) = \begin{cases} 0 & ; \quad u < 0.5 \; or \; u > 1 \\ \dfrac{u - 0.5}{0.25} & ; \quad 0.5 \leq u < 0.75 \\ \dfrac{1 - u}{0.25} & ; \quad 0.75 \leq u \leq 1 \end{cases}$$

$$(5.14)$$

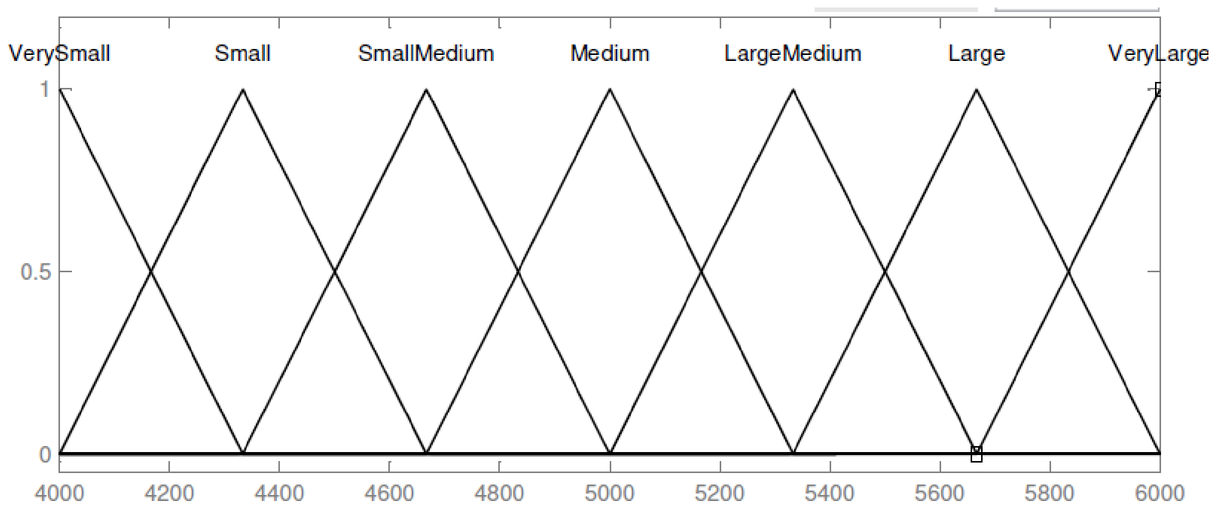
$$\mu_{VHigh}(u) = \begin{cases} 0 & ; \quad u < 0.75 \\ \dfrac{u - 0.75}{0.25} & ; \quad 0.75 \leq u \leq 1 \\ 0 & ; \quad u > 1 \end{cases}$$

$$(5.15)$$

The specifics of the output membership functions are displayed in Figure 5.3.



(a)



(b)

Figure 5.3: Output membership functions (a) Time (b) Cost

The mathematical formulation of each membership function of output attributes *Time* and *Cost* are provided in equations 5.16 to 5.22 and 5.23 to 5.29, respectively.

$$\mu_{VSmall}(u) = \begin{cases} 0 & ; \quad u < 11.2 \\ \dfrac{12.14 - u}{0.94} & ; \quad 11.2 \leq u \leq 12.14 \\ 0 & ; \quad u > 12.14 \end{cases} \tag{5.16}$$

$$\mu_{Small}(u) = \begin{cases} 0 & ; \quad u < 11.2 \; or \; u > 13.06 \\ \dfrac{u - 11.2}{0.94} & ; \quad 11.2 \leq u < 12.14 \\ \dfrac{13.06 - u}{0.92} & ; \quad 12.14 \leq u \leq 13.06 \end{cases} \tag{5.17}$$

$$\mu_{SMedium}(u) = \begin{cases} 0 & ; \quad u < 12.14 \; or \; u > 14 \\ \frac{u-12.14}{0.92} & ; \quad 12.14 \leq u < 13.06 \\ \frac{14-u}{0.94} & ; \quad 13.06 \leq u \leq 14 \end{cases} \tag{5.18}$$

$$\mu_{Medium}(u) = \begin{cases} 0 & ; \quad u < 13.06 \; or \; u > 14.94 \\ \frac{u-13.06}{0.94} & ; \quad 13.06 \le u < 14 \\ \frac{14.94-u}{0.94} & ; \quad 14 \le u \le 14.94 \end{cases}$$

(5.19)

$$\mu_{LMedium}(u) = \begin{cases} 0 & ; \quad u < 14 \; or \; u > 15.86 \\ \frac{u-14}{0.94} & ; \quad 14 \le u < 14.94 \\ \frac{15.86-u}{0.92} & ; \quad 14.94 \le u \le 15.86 \end{cases}$$

(5.20)

$$\mu_{Long}(u) = \begin{cases} 0 & ; \quad u < 14.94 \; or \; u > 16.8 \\ \frac{u-14.94}{0.92} & ; \quad 14.94 \le u < 15.86 \\ \frac{16.8-u}{0.94} & ; \quad 15.86 \le u \le 16.8 \end{cases}$$

(5.21)

$$\mu_{VLong}(u) = \begin{cases} 0 & ; \quad u < 15.86 \\ \frac{u-15.86}{0.94} & ; \quad 15.86 \le u \le 16.8 \\ 0 & ; \quad u > 16.8 \end{cases}$$

(5.22)

$$\mu_{VSmall}(u) = \begin{cases} 0 & ; \quad u < 4000 \\ \frac{4333-u}{333} & ; \quad 4000 \le u < 4333 \\ 0 & ; \quad u > 4333 \end{cases}$$

(5.23)

$$\mu_{Small}(u) = \begin{cases} 0 & ; \quad u < 4000 \; or \; u > 4667 \\ \frac{u-4000}{333} & ; \quad 4000 \le u < 4333 \\ \frac{4667-u}{334} & ; \quad 4333 \le u \le 4667 \end{cases}$$

(5.24)

$$\mu_{SMedium}(u) = \begin{cases} 0 & ; \quad u < 4333 \; or \; u > 5000 \\ \frac{u-4333}{334} & ; \quad 4333 \leq u < 4667 \\ \frac{5000-u}{333} & ; \quad 4667 \leq u \leq 5000 \end{cases}$$

(5.25)

$$\mu_{Medium}(u) = \begin{cases} 0 & ; \quad u < 4667 \; or \; u > 5333 \\ \frac{u-4667}{333} & ; \quad 4667 \leq u < 5000 \\ \frac{5333-u}{333} & ; \quad 5000 \leq u \leq 5333 \end{cases}$$

(5.26)

$$\mu_{LMedium}(u) = \begin{cases} 0 & ; \quad u < 5000 \; or \; u > 5667 \\ \dfrac{u-5000}{333} & ; \quad 5000 \leq u < 5333 \\ \dfrac{5667-u}{334} & ; \quad 5333 \leq u \leq 5667 \end{cases}$$

(5.27)

$$\mu_{Long}(u) = \begin{cases} 0 & ; \quad u < 5333 \; or \; u > 6000 \\ \dfrac{u-5333}{334} & ; \quad 5333 \leq u < 5667 \\ \dfrac{6000-u}{333} & ; \quad 5667 \leq u \leq 6000 \end{cases}$$

(5.28)

$$\mu_{VLong}(u) = \begin{cases} 0 & ; \quad u < 5667 \\ \dfrac{u-5667}{333} & ; \quad 5667 \leq u \leq 6000 \\ 0 & ; \quad u > 6000 \end{cases}$$

(5.29)

Fuzzy rules govern the controller's operation. A total of 6125 (5 × 5 × 5 × 7 × 7) rules are necessary, with 250 IF-THEN rules defined using Mamdani inference. These fuzzy rules integrate prior experience and expert knowledge to establish connections between input and output variables [119]. Table 5.1 presents a selection of fuzzy inference rules. These guidelines provide logical direction for selecting the optimal set of solutions for the composite services.

Table 5.1: Fuzzy rules

Rule	Management Skills	Weather Conditions	Farmer Skills	Time	Cost
1	VeryGood	Good	VeryHigh	Small	Small
2	VeryGood	VeryBad	VeryHigh	Long	Large
3	Good	Good	VeryLow	SmallMedium	SmallMedium
4	Medium	VeryBad	Medium	Long	Large
5	Less	Good	Low	LongMedium	LargeMedium
………	VeryLess	Medium	VeryLow	VeryLong	VeryLarge

These guidelines will determine the smart choices for identifying the best set of solutions for the combined services. The optimal choices have been determined through the application of expert knowledge and empirical data.

5.3 Impact of Uncertainties on Linear Service Composition Optimization

This section covers how various uncertain factors like environmental, human-based, or economic influence the optimization process of service composition problem by using a FIS considering a linear type of relationship between cost and time objectives.

5.3.1 Optimization Algorithm: NSGA-II

The algorithm starts with a randomized population, which is subsequently organized using the non-dominated sorting procedure, where all solutions that are not dominated are assigned rank 1 and have been temporarily eliminated from the initial population, followed by the subsequent set of solutions being ranked as 2, and so on until all possible solution sets are ranked. Then, the current population is subjected to a binary tournament selection technique, which selects one solution based on rank from the current population, and when two solutions are on the same front, the crowding distance theory is utilized for the selection mechanism.

Once parents are selected, offspring are produced by applying crossover and mutation operators to the parent population. The subsequent population is formed by selecting the best solutions from the blended pool of offspring and parents. This process repeats until the criteria for termination are met, which could be either predetermined generations or when the solutions reach a saturation level [105].

5.3.2 Proposed Fuzzy-based Architecture

Application-based model's uncertain, imprecise, and subjective behavior can be solved by using either fuzzy logic or fuzzy set theory. Fuzzy logic models have been demonstrated to be capable of handling the unpredicted behavior of the environment variables about agricultural datasets in many recent tests and studies [120].

The primary goal of modeling systems for smart agriculture is to determine the best way to optimize the system for the particular kind of dataset being studied. Finding an algorithm that can resolve numerous uncertain attributes in agricultural data sets is challenging because these properties are extremely variable and dependent on other factors. It performs a comparable role to that of human perception. Fuzzy logic can be utilized to create agricultural decisions since it can handle uncertainty [121]. The proposed framework for Fuzzy Linear NSGA-II (Fuzzy-Li-NSGA-II) is portrayed in Figure 5.4.

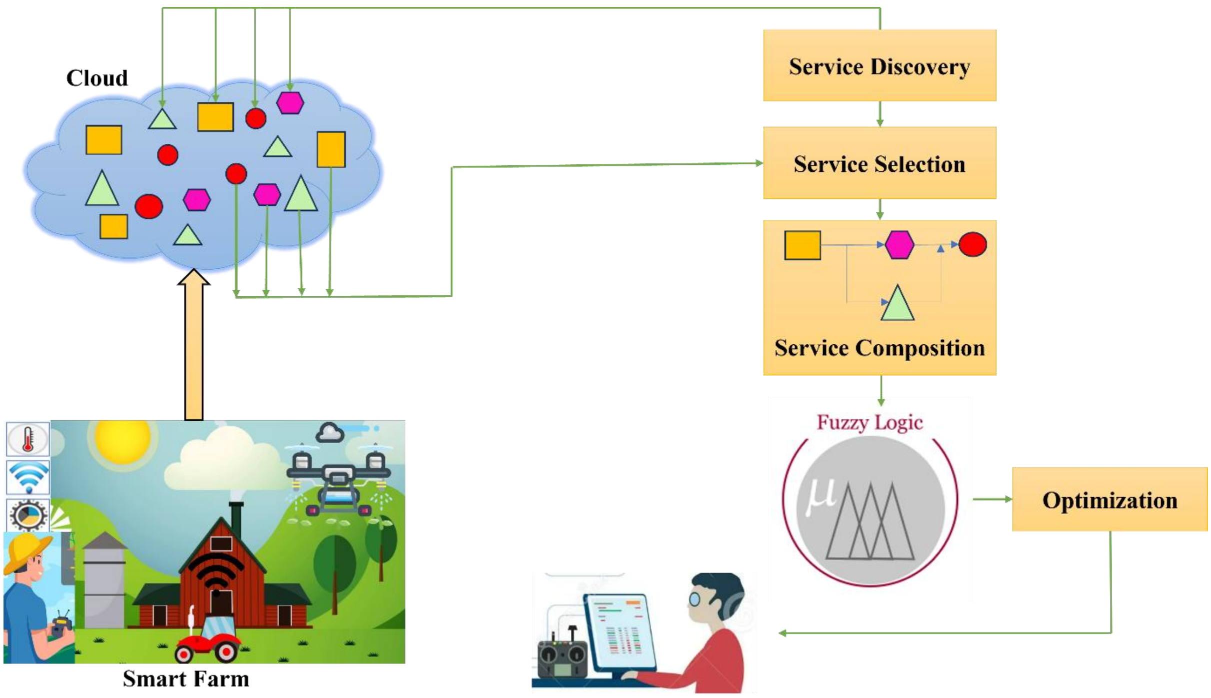


Figure 5.4: Proposed architecture for Fuzzy-Li-NSGA-II

This system functions across several tiers within an IoT ecosystem, where IoT sensor data is preserved in cloud-based services. While several providers offer comparable characteristics, their QoS attributes differ. The initial step involves a service discovery process to identify functionally similar services. Following this, the required services are chosen from the discovered options to meet user needs, with selection guided by QoS-based criteria. Since individual services cannot fully address complex user requests, a service composition phase is implemented. Various uncertain elements can indirectly affect smart agriculture services. To evaluate the impact of these uncertainties, a fuzzy logic controller is employed. The process concludes with the application of optimization operators to an initialized population, aiming to find Pareto optimal solutions that ultimately satisfy user requirements.

Figure 5.5 illustrates the flowchart for the proposed Fuzzy-Li-NSGA-II algorithm.

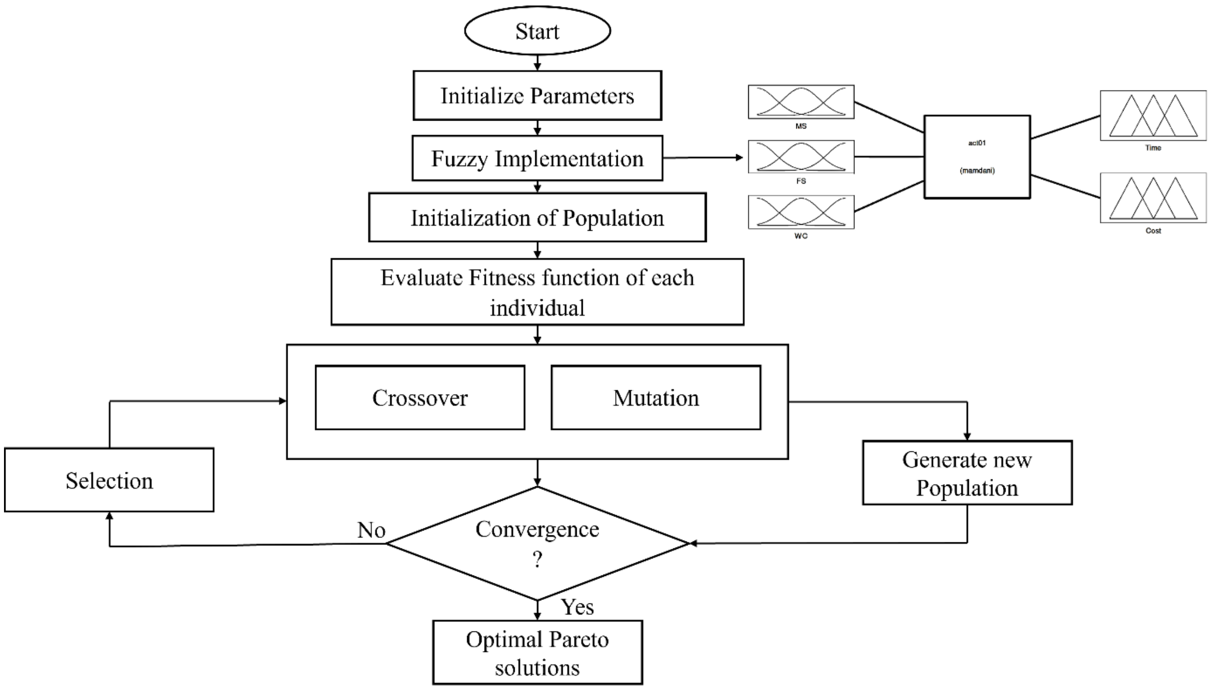


Figure 5.5: Flow chart of proposed Fuzzy-Li-NSGA-II approach

5.3.3 Simulation Setup

MATLAB R2013a version is used to run the proposed algorithm. An FIS has been used to model it to ascertain how uncertainties would affect the specified multiple objectives and NSGA-II is used as an optimization algorithm (Optimal algorithm out of MOGA, NSGA-II, and MOGSK).

Table 5.2 contains a tabulation of the parameters that were utilized to validate the algorithm's performance.

Table 5.2: Simulation parameters

Parameters	**Values**
Population Size (N_P)	200
Number of Iterations	1,000
Mutation Probability	0.07
Crossover Probability	0.9

5.3.4 Results and Discussions

Examining how the uncertainties present in real-life smart agriculture applications influence the overall composited services is the aim of this study. For this, a Mamdani FIS has been designed and after that, the composited services are optimized using the NSGA-II algorithm, thereby, producing a set of Pareto optimal solutions. Different input variable values are used to assess their influence on the output variables. Figure 5.6 illustrates four possible distinct cases of membership functions.

In the first case, each of the three input membership functions—*MS, WC,* and *FS*—is equal to 0.2 which can be regarded as a worst-case scenario. This indicates that *Management Skills (MS)* are *VeryLess*, *Weather Conditions (WC)* are *VeryBad*, and *Farmer Skills (FS)* are *VeryLow*. For the second case, all *MS*, *WC*, and *FS* are equal to 0.5 meaning all *MS*, *WC*, and *FS* are at *Medium* level, indicating the normal case scenario. For the third instance, *MS* = 0.5, *WC* = 0.8, and *FS* = 0.2 meaning that *Management Skills (MS)* are *Medium*, *Weather Conditions (WC)* are VeryGood, and *Farmer Skills (FS)* are VeryLow. This shows the mixed-case scenario. The last case depicts *MS, WC*, and *FS* = 0.9 where *Management Skills (MS)* are *VeryGood*, *Weather Conditions (WC)* are *VeryGood*, and *Farmer Skills (FS) are VeryHigh.* showing the best-case scenarios. It can be observed from Pareto front analysis that best-case scenario provides a more diversified solution and is also closer to the origin, indicating better solutions than other scenarios.

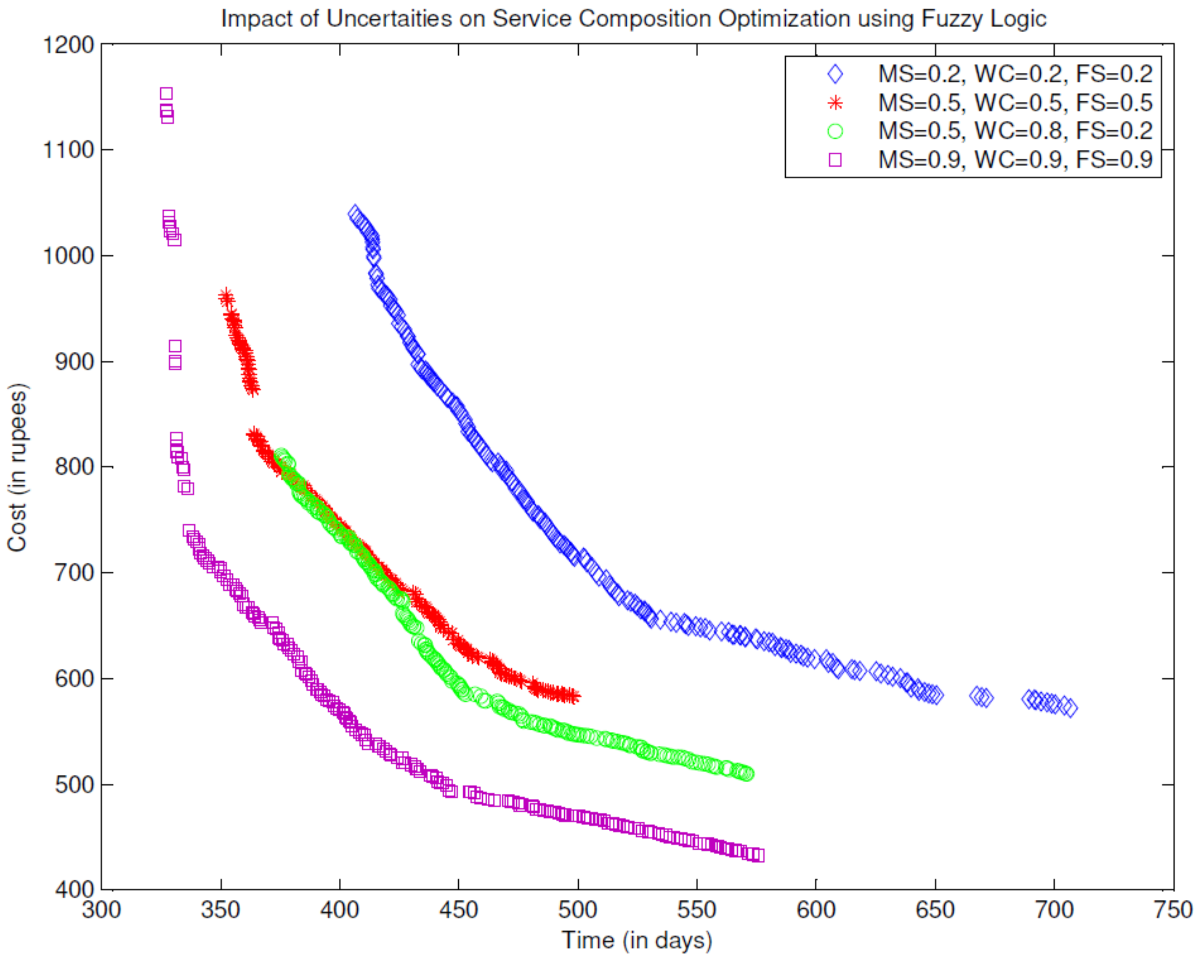


Figure 5.6. Distinct possible case scenarios of smart agriculture

To improve readability and clarity, a statistical analysis of the findings is presented in Table 5.3.

Table 5.3: Statistical analysis

Case Scenarios	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
Worst Case	Time	706.7	406.6	81.94	511.3	490	406.6	300.1
	Cost	1039	572	138.3	759.9	734.9	572	467.3
Normal Case	Time	498.3	352.2	42.89	413.1	409.3	352.2	146.1
	Cost	962.6	583.1	108.2	734.3	722.9	583.1	379.5

Mixed Case	Time	570.9	375.6	56.04	457	443.8	375.6	195.3
	Cost	810.8	509.8	91.39	629	607.9	509.8	301
Best Case	Time	575.9	327.4	76.04	429.1	411.4	327.4	248.5
	Cost	1153	432.6	157.3	590.8	540.2	432.6	720.5

5.4 Impact of Uncertainties on Non-Linear Service Composition Optimization

This part of the chapter embraces the impact of various environmental and non-environmental factors on the optimization process of service composition by using a FIS considering a non-linear relationship between cost and time objectives to represent the real-world scenarios of smart agriculture. The non-linear relationship between the cost and time objectives is defined by Lagrange's interpolation, which will be covered first. An overview of the NSGA-II optimization algorithm will come next, and then a description of the proposed architecture Fuzzy Lagrange's NSGA-II (Fuzzy-La-NSGA-II).

5.4.1 Phase 1: Lagrange's Interpolation

A strong mathematical method for estimating unknown values within a certain range of known data points is Lagrange interpolation. This method is very helpful for interpolation jobs in a variety of applications because it allows values to be calculated at defined intervals by building a polynomial that goes through a given collection of points. The non-linear relationship between the cost and time objectives in service composition optimization for smart agriculture is established in this work using Lagrange interpolation. This method enables better decision-making and optimization results by providing a more realistic depiction of the difficulties in striking a balance between these two crucial goals [122].

5.4.2 Phase 2: NSGA-II

The algorithm begins with a randomized population, organized using non-dominated sorting. It then uses a selection method of binary tournament to create a parent population, using crowding distance. Offspring are produced using crossover and mutation operators, and the subsequent population is formed by selecting the best solutions from the blended pool. This process repeats until a termination criterion is satisfied [105].

5.4.3 Proposed Fuzzy-based Architecture

Fuzzy set theory and fuzzy logic models can resolve ambiguous behavior in application-based models, particularly in agricultural data sets. These models can handle uncertain attributes, similar to how the brain functions, making smart agricultural decisions easier and more adaptable to the specific data set being considered [123]. Thus, the proposed fuzzy-based system explores the influence of fuzzy systems on optimization algorithms for smart agriculture, illustrating the architecture in Figure 5.7.

This architecture operates on several IoT structure tiers. IoT sensor data is kept in the cloud as a service. While many services have comparable functionality, their QoS features differ. As a result, during the service discovery phase, services with comparable functionality were initially found. The next step is to choose the services from the pool of available options that best suit the user's needs. This choice is based on characteristics that are in accordance with cost and time taken as QoS metrics. Because the requests of user's are multifaceted, one service couldn't be used to satisfy them. Therefore, service composition is completed in the following stage. The relationship between the cost and time metrics is non-linear since the real scenario is used. Thus, it is defined using Lagrange's interpolation method. The services that smart agriculture offers are indirectly impacted by numerous unknown factors. Fuzzy logic controllers have thus been used to assess the effects of those factors on services. After initializing the population, additional optimization operators were applied to obtain Pareto optimal solutions, which ultimately satisfied user demands. The flowchart to illustrate the proposed Fuzzy-La-NSGA-II approach is shown in Figure 5.8.

5.4.4 Simulation Setup

The proposed Fuzzy-La-NSGA-II algorithm is tested using MATLAB R2013a version. Mamdani FIS is used to model the various environmental and human-based uncertainties to check their impact on the real-world scenario of smart agriculture applications with NSGA-II as optimization algorithm (Optimal algorithm out of MOGA, NSGA-II, and MOGSK). Simulation parameters for the optimization algorithm are tabulated in Table 5.4.

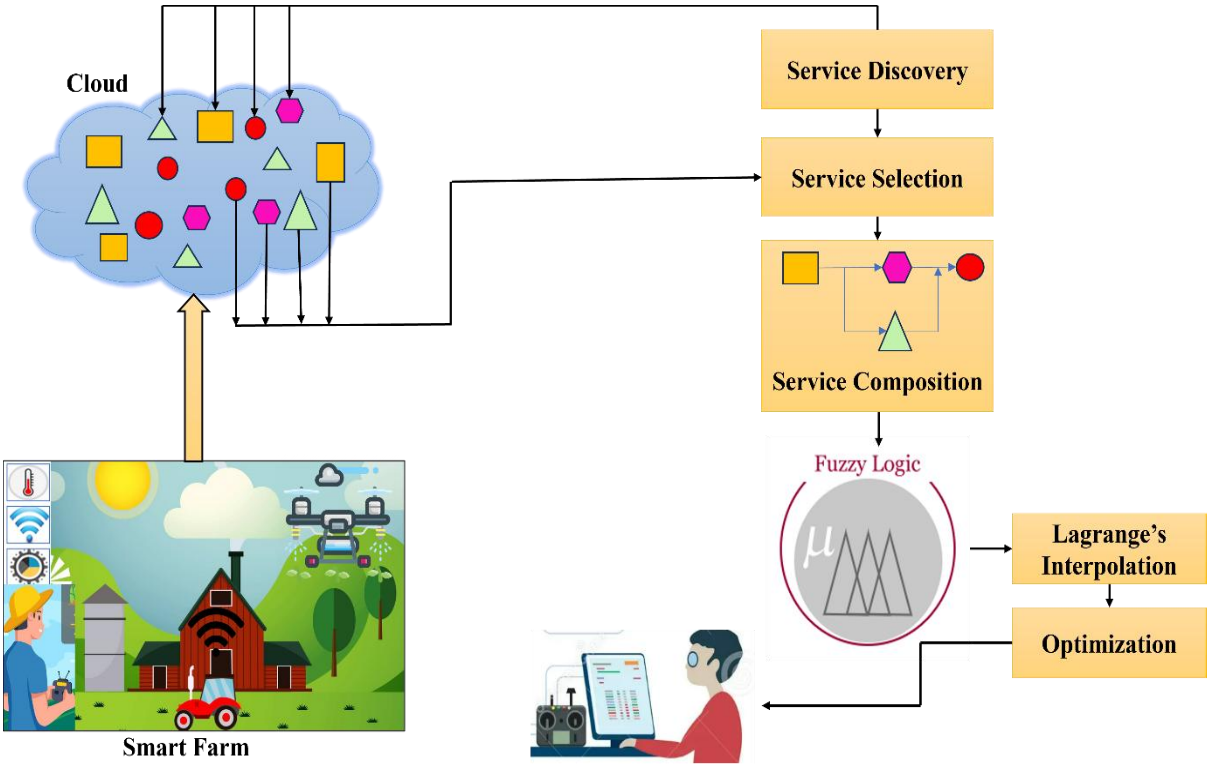


Figure 5.7: Proposed architecture for Fuzzy-La-NSGA-II

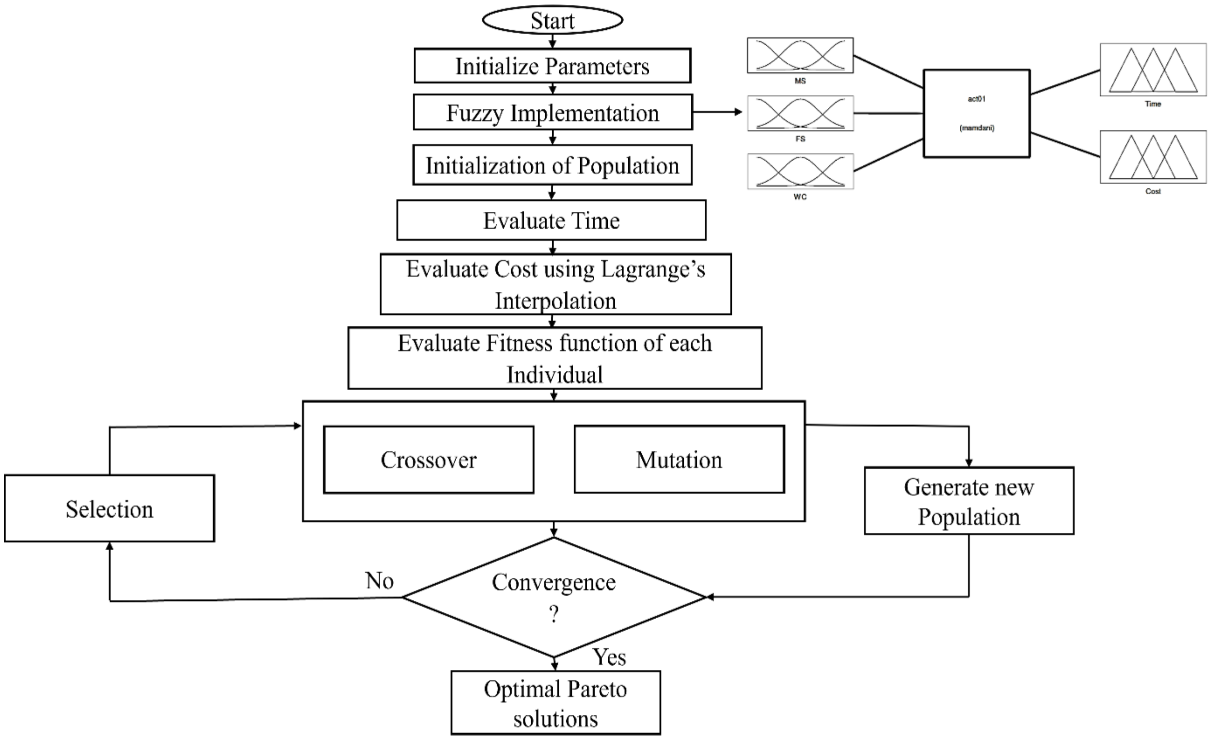


Figure 5.8: Flow-chart illustration of Fuzzy-La-NSGA-II

Table 5.4: Simulation parameters

Parameters	**Values**
Population Size (N_P)	200
Number of Iterations	1,000
Mutation Probability	0.07
Crossover Probability	0.9

5.4.5 Results and Discussions

Smart agriculture faces uncertainties like environmental and economic factors, requiring robust optimization strategies to ensure adaptability and resilience. Technology and data-driven approaches help address these uncertainties, but robust optimization strategies are needed for optimal results. Thus, this part of the objective has examined the impact of uncertainties on the proposed architecture of optimizing time and cost by considering a non-linear relationship between them and optimizing them.

To determine how input variables affect output variables, distinct values are obtained. Four scenarios involving fuzzy membership functions are depicted in Figure 5.9. All three of the input membership functions *(MS, WC,* and *FS)* are equal to 0.3 in the first case, which can be considered a worst-case scenario; in the second case, which can be considered a normal-case scenario, they are all equal to 0.5. A mixed-case scenario is considered by taking $MS = 0.5$, $WC = 0.9$, and $FS = 0.3$ whereas a best-case scenario is taken by considering all $MS=WC=FS=0.8$. As can be seen from Figure 5.9, the best-case scenario offers the most favorable set of Pareto optimal solutions while the worst-case scenario displays, when compared, the poorer optimal Pareto solutions.

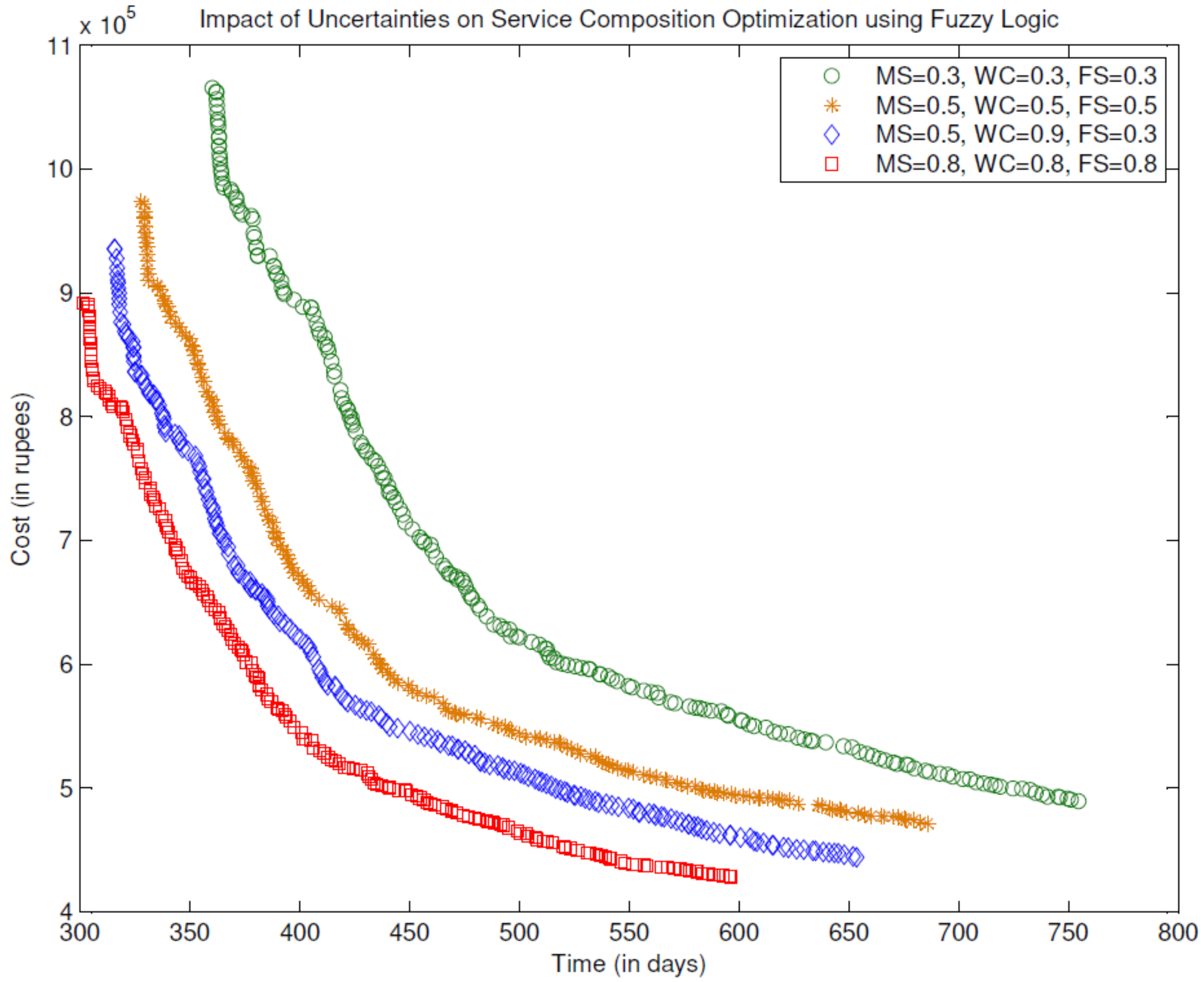


Figure 5.9: Distinct possible case scenarios of Fuzzy La-NSGA-II

Table 5.5 presents a statistical analysis to improve the interpretation of the results.

Table 5.5: Statistical analysis

Case Scenarios	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
Worst Case	Time	754.7	360.3	117.8	502.7	471.5	360.3	394.4
	Cost	1.066e+06	4.896e+05	1.778e+05	7.164e+05	6.698e+05	4.896e+05	5.76e+05
	Time	686	327.8	108.5	461	432.4	327.8	358.2

Normal Case	Cost	9.736e+05	4.717e+05	1.558e+05	6.569e+05	6.123e+05	4.717e+05	5.019e+05
Mixed Case	Time	653.5	315.6	102.6	442.8	414.4	315.6	337.9
	Cost	9.357e+05	4.443e+05	1.484e+05	6.267e+05	5.828e+05	4.443e+05	4.914e+05
Best Case	Time	596.1	301.6	87.78	411.2	386.2	301.6	294.5
	Cost	8.921e+05	4.286e+05	1.447e+05	6.057e+05	5.712e+05	4.286e+05	4.635e+05

5.5 Behavioral Analysis Comparison of Fuzzy Li-NSGA-II and Fuzzy La-NSGA-II

This section provides a behavioral analysis comparison of Fuzzy-Li-NSGA-II and Fuzzy-La-NSGA-II by using both Pareto front and statistical analysis. Here, Fuzzy-Li-NSGA-II depicts a linear relationship between the competing goals of minimizing cost and time whereas Fuzzy-La-NSGA-II portrays a non-linear relationship for a more realistic experience of real-world smart agriculture applications. Figure 5.10 presents a behavioral analysis comparison of both approaches when MS=WC=FS=0.5 means a normal-case scenario.

It has been analyzed that both provide diversified solutions for their particular relationship between cost and time objectives. However, the Fuzzy-La-NSGA-II is more reliable in representing real-world scenarios in the context of non-linear service composition optimization problems. For a better understanding, statistical analysis is provided in Table 5.6.

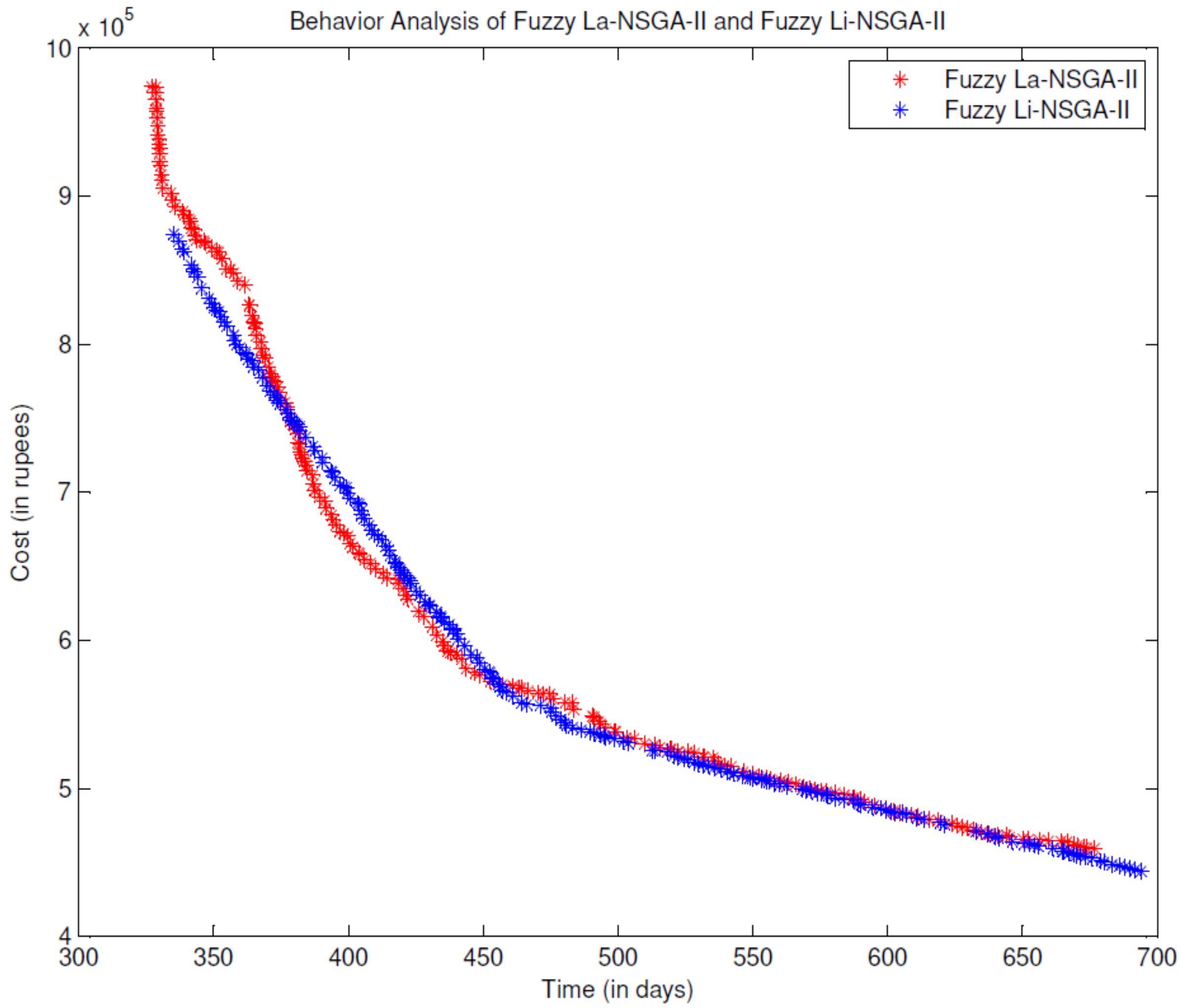


Figure 5.10: Behavioral analysis of Fuzzy La-NSGA-II and Fuzzy-Li-NSGA-II

Table 5.6: Statistical analysis

Algorithm	**Objectives**	**Maximum**	**Minimum**	**Standard Deviation**	**Mean**	**Median**	**Mode**	**Range**
Fuzzy La-NSGA-II	Time	676.9	327.1	105.3	455.8	421.8	327.1	349.8
	Cost	9.739e+05	4.595e+05	1.589e+05	6.593 e+05	6.294 e+05	4.595e+05	5.144 e+05
	Time	694	335.3	104.3	482.1	453.8	335.3	358.7

Fuzzy Li-NSGA-II	Cost	8.744e+05	4.44e+05	1.247e+05	6.076 e+05	5.743 e+05	4.44e+05	4.304 e+05

5.6 Summary

In the context of smart agriculture, this chapter explores how uncertainties affect composited service optimization. A Mamdani fuzzy inference system is used to assess these uncertainties, providing a methodical way to quantify and examine their effects. The most successful optimization method for resolving our service composition problem is NSGA-II, which builds on the results of chapters 3 and 4. As a result, NSGA-II is used as the main optimization algorithm in chapter 5. The chapter investigates how uncertainties affect both linear and non-linear objective functions by taking Fuzzy-Li-NSGA-II and Fuzzy-La-NSGA-II, respectively. To understand the difference between both, a behavioral analysis is provided. Because the relationship between services in a composition might be either linear or non-linear, the study shows that both kinds of objectives can be used depending on user requirements. This adaptability enables customized solutions that fit the unique requirements and dynamics of scenarios involving smart agriculture.

CHAPTER-6
A NOVEL NATURE-INSPIRED MULTI-OBJECTIVE ELECTRIC EEL FORAGING OPTIMIZATION ALGORITHM

CHAPTER 6

A NOVEL NATURE-INSPIRED MULTI-OBJECTIVE ELECTRIC EEL FORAGING OPTIMIZATION ALGORITHM

6.1 Chapter Overview

Solving multi-objective optimization challenges in real-world applications is challenging when using mathematical models. As a result, various nature-inspired meta-heuristic approaches are employed to address these complex problems.

This chapter reflects on the ingenious collective foraging strategies of electric eels found in nature and considers them as an inspiration for a multi-objective electric eel foraging optimization algorithm. To enable both exploitation and exploration throughout the process, the algorithm mathematically replicates the four essential foraging behaviors of interaction, hunting, migrating, and resting.

6.2 Description of Electric Eel Foraging Behavior

Electric eels, native to South America, are known for their high voltage wires, capable of releasing 300-800 V to stun prey. With thousands of electrocytes in each of their three separate sets of electric organs, their organs store energy like small batteries [124]. Figure 6.1 shows the structure of electric eel [125].

Eels generate 10 V of electrical signals to locate prey, use this feedback for defense, and communicate with each other. They emit more charge when finding prey, making it an effective foraging strategy. Eels are swarm-based creatures, using social predation for hunting. They form a “prey ball” by grouping together, swimming in circles, and herding fish into it before making a high-voltage raid. It is more likely to catch more prey when hunting in groups, especially when fish are plentiful [126].

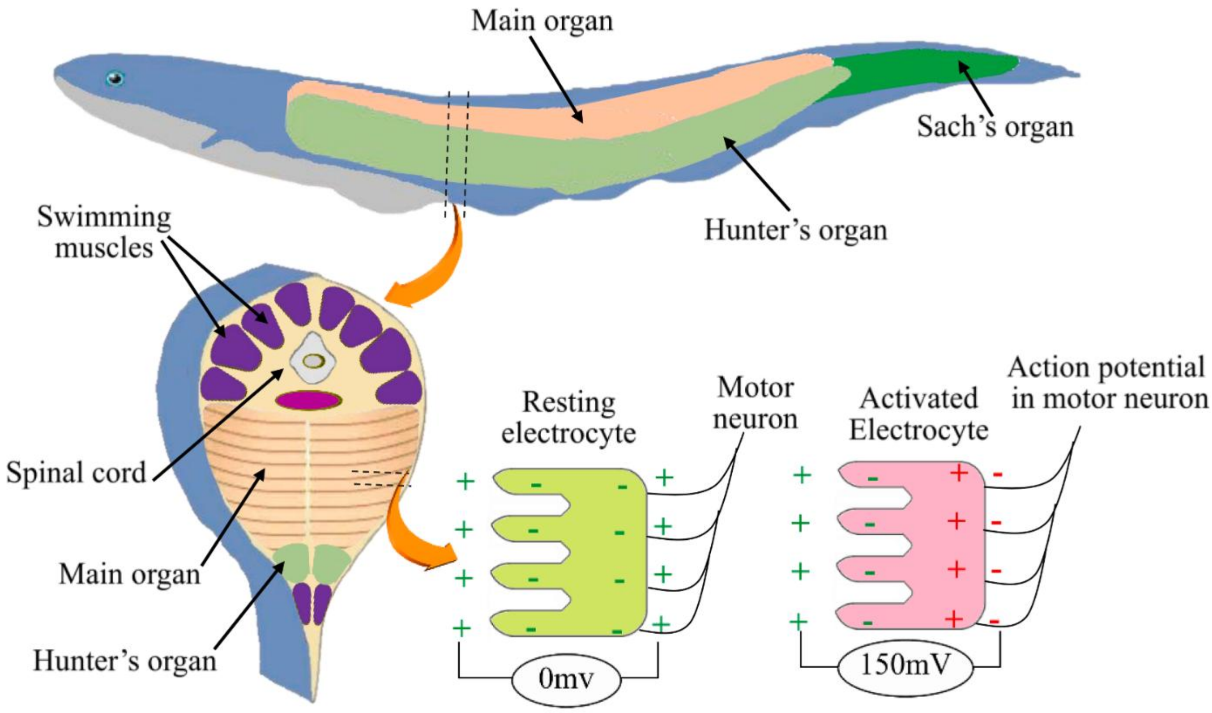


Figure 6.1: Physical structure of electric eel [125]

6.3 Mathematical Representation of Electric Eel Foraging Optimization (EEFO)

Effectively navigating intricate problem environments is made possible by EEFO's dynamic management of the exploration and exploitation phases. It incorporates effective local and global search strategies like hunting and migration, displaying higher performance in comparable tests. Because of its scalability and ease of implementation, EEFO is a reliable option for solving complex problems and producing high-quality results. The shifting of exploration to exploitation is managed by a factor called the energy factor E_f which is defined below in equation 6.1 [125].

$$E_f = 4 \times sin\left(1 - \frac{t}{maxiterations}\right) \times ln\left(\frac{1}{r_1}\right) \tag{6.1}$$

Here, r_1 is a random number within (0,1).

When $E_f \leq 1$, it performs globally whereas for $E_f > 1$, it performs local search by using resting, hunting, and migrating regions. The flowing subsection in EEFO models foraging activities.

6.3.1 Interaction

Every electric eel in EEFO is a candidate solution, and the intended prey is the one that performs best after every step. They cooperatively interact with other individuals using position information, a behaviour known as the global exploration phase. They can update their position by measuring the difference between a randomly chosen eel and the population centre. Eels churn, or move randomly in various directions, as a means of communication with one another. The equations 6.2 to 6.6 represent this churn and are given below [125].

$$C = n_1 \times [B_1, B_2, \dots, B_s, \dots, B_D] \tag{6.2}$$

$$n_1 \sim \mathcal{N}(0,1) \tag{6.3}$$

$$B(s) = \begin{cases} 1 & if\ s == g_l \\ 0 & else \end{cases} \tag{6.4}$$

$$g = randperm\ (D) \tag{6.5}$$

$$l = \left\lceil \frac{maxiterations - t}{maxiterations} \times r_2 \times (D-2) + 2 \right\rceil \tag{6.6}$$

In the above equations, $maxiterations$ defines the maximum iterations defined for convergence, t is the current iteration, C is the churning factor.

The interaction behavior of the eels can be defined using equation 6.7 given below [125].

$$v_k(t+1) = \begin{cases} x_i(t) + C \times \left(\bar{x}(t) - x_k(t)\right) \ where\ p_1 > 0.5 \\ x_i(t) + C \times \left(x_r(t) - x_k(t)\right) where\ p_1 \leq 0.5 \end{cases} \quad fit(x_i(t)) < fit(x_k(t))$$

$$v_i(t+1) = \begin{cases} x_k(t) + C \times \left(\bar{x}(t) - x_i(t)\right) \ where\ p_2 > 0.5 \\ x_k(t) + C \times \left(x_r(t) - x_i(t)\right) where\ p_2 \leq 0.5 \end{cases} \quad fit\left(x_i(t)\right) \geq fit(x_k(t))$$

(6.7)

$$\bar{x}(t) = \frac{1}{n}\sum_{k=1}^{n} x_k(t) \tag{6.8}$$

$$x_r = low + r \times (up - low) \tag{6.9}$$

In equation (6.7), p_1 and p_2 are the random numbers generated between (0,1), $fit\ (x_k)$ defines fitness of that particular candidate position of k^{th} eel, and x_i is the eel position which is picked stochastically from the population that exists at that time. Equations 6.8 and 6.9 show the mean position of eels and any random eel position, respectively. low and up are lower and upper bound, respectively which are shown in equation 6.9.

6.3.2 Resting

Electric eels in the EEFO should construct a resting area before starting resting activities. The eel's position and search space should be standardized to a range of 0-1 to increase efficiency. The anticipated position is believed to be the center of the eel's resting region. The solutions found so far in the interaction phase are refined during this phase. Equation 6.10 defines the resting area whereas equations 6.11, 6.12, and 6.13 describes the scaling factor, centre of the resting region, and normalized number, respectively [125].

$$\{X|X - Z(t) \leq \alpha_0 \times |Z(t) - X_{prey}(t)|\} \tag{6.10}$$

$$\alpha_0 = 2 \cdot (e - e^{\frac{t}{maxiterations}}) \tag{6.11}$$

$$Z(t) = low + z(t) \times (up - low) \tag{6.12}$$

$$z(t) = \frac{x_{rand\{n\}}^{rand\{d\}} \cdot \{t - low^{rand\{d\}}\}}{up^{rand\{d\}} - low^{rand\{d\}}} \tag{6.13}$$

Here, X_{prey} is the position vector of the best solution obtained till that time, α_0 is the initial scale of the resting region, the expression $\alpha_0 \times |Z(t) - X_{prey}(t)|$ defines the resting area's range.

Thus, the resting position within the resting area of a particular eel can be defined as in equation 6.14. It is performed prior to resting behavior.

$$R_k(t+1) = Z(t) + \alpha \times |Z(t) - X_{prey}(t)| \tag{6.14}$$

$$\alpha = \alpha_0 \times \sin(2\pi r_3) \tag{6.15}$$

In equation 6.15, α denotes resting region's scale. The eel's resting behavior is determined by the equation 6.16 given below.

$$v_k(t+1) = R_k(t+1) + n_2 \times (R_k(t+1) - round(rand) \times x_k(t)) \tag{6.16}$$

6.3.3 Hunting

Eels cooperatively swim in a large circle to hunt prey, communicating and cooperating with their peers by using low electric discharges. As interaction increases, the electrified circle shrinks, and eels bring fish from deeper to shallow regions, creating a hunting area where prey moves. The hunting area is defined in below given equation 6.17 [125].

$$\{X \left| X - X_{prey}(t) \right| \leq \beta_0 \times |\bar{x}(t) - X_{prey}(t)|\} \tag{6.17}$$

$$\beta_0 = 2 \cdot (e - e^{\frac{t}{maxiterations}}) \tag{6.18}$$

In equation 6.18, β_0 is the initial scale of the hunting area whereas in equation 6.17, the term $\beta_0 \times |\bar{x}(t) - X_{prey}(t)|$ defines the hunting range of the eel. Thus, newly found prey's position enclosed by the hunting area can be described using equation 6.19.

$$h_{prey}(t+1) = X_{prey}(t) + \beta \times |\bar{x}(t) - X_{prey}(t)| \tag{6.19}$$

$$\beta = \beta_0 \times \sin(2\pi r_4) \tag{6.20}$$

In equation 6.20, β is the scale of the hunting area.

An eel starts behaving like prey in that particular hunting area once it has been discovered. The eel swiftly locates its prey, coils its head and tail, and entangles it with the prey, emitting a high-voltage current. The curling behavior is described by the equation 6.21.

$$v_k(t+1) = h_{prey}(t+1) + \eta \times (h_{prey}(t+1) - round(rand) \times x_k(t) \tag{6.21}$$

$$\eta = e^{\frac{r_5 \cdot (1-t)}{maxiterations}} \cdot \cos(2\pi r_5) \tag{6.22}$$

The factor η in equation 6.22 is the curling factor.

6.3.4 Migration

Migration in EEFO entails recurring exploration of various places in the search space. Similar to electric eels migrating to new hunting regions, this mechanism maintains an equilibrium between exploitation and exploration. The following equations 6.23 and 6.24 are used to quantitatively model the eel's migration behavior [125].

$$v_k\ (t+1) = -r_6 \times R_k(t+1) +\ r_7 \times h_r(t+1) - L \times (h_r(t+1) - x_k(t)) \tag{6.23}$$

$$h_r(t+1) = X_{prey}(t) + \beta \times |\bar{x}(t) - X_{prey}(t)| \tag{6.24}$$

Here, h_r is any position within hunting area. r_6 and r_7 are the random numbers within the range (0,1). L is the Levy Flight function and the factor $(h_r(t+1) - x_k(t))$shows the movement of eels towards the hunting area.

6.4 Multi-objective Electric Eel Foraging Optimization

The proposed MO-EEFO has made a few transitions in the single-objective algorithm to make it multi-objective. One is the creation of non-dominant solutions that have been found so far. Non-dominated sorting and the crowding distance are used to obtain those non-dominated solutions, which enhance diversity and facilitate better exploitation and exploration.

MO-EEFO starts by setting up several parameters, such as the maximum iterations and the electric eel's population size. In the meantime, a uniform distribution of a set of eels is created at random to make a population of eels known as eel chromosomes. To create a solution set that is more refined than others, the concept of non-dominated sorting is applied to organize them based on their rank and crowding distance.

It creates the global best solutions. An energy factor E_f is then defined for calculating the energy of each eel chromosome at each iteration depending on which one of the four foraging behaviors of the eel chromosome will be chosen to explore and exploit the search space properly. For each iteration, if $(E_f > 1)$, then each eel chromosome uses interactive behavior to execute exploration of the search space. Each eel chromosome engages in exploitation when the energy factor $(E_f \leq 1)$, employing the resting, migrating, or hunting behaviors with an equal chance. All eel chromosomes are subjected to each situation in order to generate new offspring eels, which are then compared with their parent eel chromosomes. After that, an intermediate population is created which is the combination of parent eel chromosomes and newly formed offspring eels. Again, non-dominated sorting is applied to find the best solutions from the intermediate population. The concept of non-dominated sorting arranges the population based on rank and crowding distance. With the increase in the number of iterations, the value of E_f falls which forces eels to shift from exploration to exploitation. This process is carried out interactively up until the convergence criterion is met. The pseudocode for the MO-EEFO algorithm is given in Figure 6.2 whereas the flow chart for the same is illustrated in Figure 6.3.

Algorithm: MO-EEFO Algorithm

Set parameters population size (n) and maximum number of iterations. Initialize the eel population at random X_k where $k = 1,2,3,\ldots,n$ and
Assess their fitness $Fitness_k$
Sort population (n) using non-dominated sorting and compute crowding distance.
while the stopping requirement is not met **do**
  for each eel X_k **do**
    Calculate E_f utilizing Eq. 6.1
    if $E_f > 1$ then
      Carry out the interacting behavior utilizing Eq. 6.17
      Assess the fitness $Fitness_k$
      else
      if $rand < \frac{1}{3}$ then
        Discover the resting region utilizing Eq. 6.14
        Carry out the resting behavior utilizing Eq. 6.16
        Assess the fitness $Fitness_k$
        else if $rand > \frac{2}{3}$ then
        Carry out the migrating behavior utilizing Eq. 6.23
        else
        Discover the hunting region utilizing Eq. 6.19
        Carry out the hunting behavior utilizing Eq. 6.21
        end if
    end if
    end for
  Update the best solutions as offspring.
  Form an intermediate population by combining the parent eel population and offspring eel population.
  Perform non-dominated sorting on the intermediate population.
  $X_k :=$ Non-dominated sorting of intermediate population on the basis of rank and crowding distance.
end while
Return best Pareto optimal solutions.

Figure 6.2: Pseudocode of proposed MO-EEFO algorithm

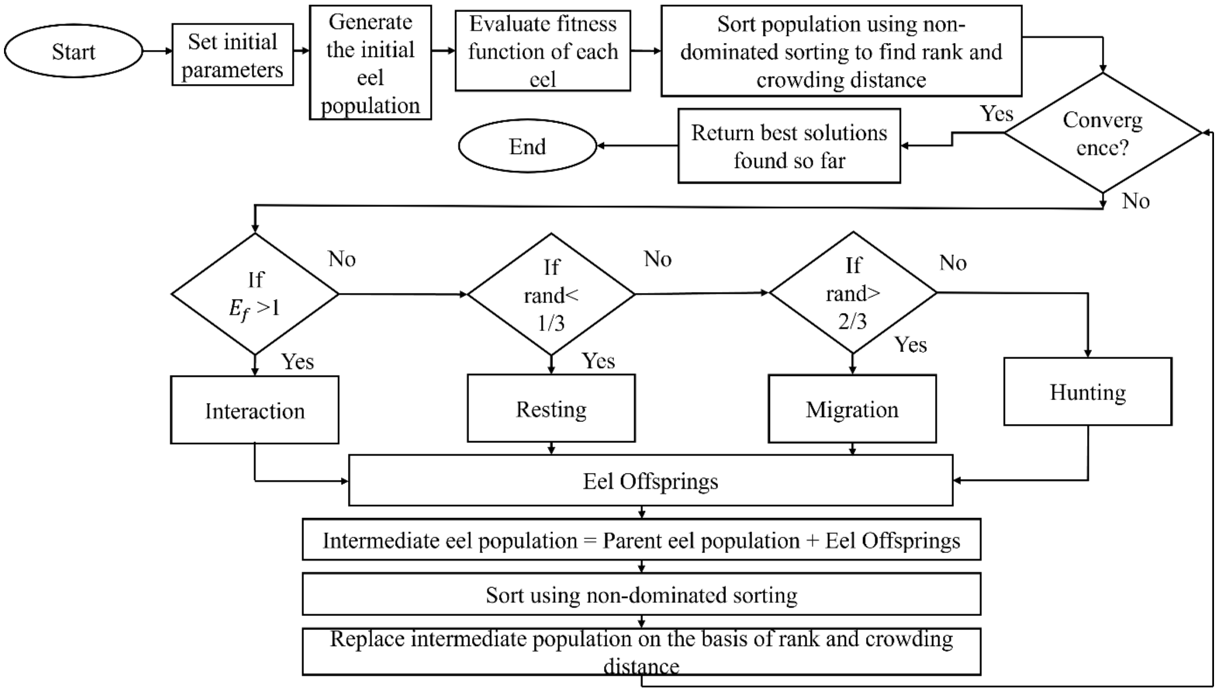


Figure 6.3: Flow chart illustration of the proposed MO-EEFO algorithm

6.5 Simulation Setup and Result Analysis

This section covers the experimental analysis of the proposed MO-EEFO algorithm's efficiency on Zitzler-Deb-Thiele (ZDT) benchmark problems, and comparison of proposed algorithm with other algorithms present in literature to verify its efficiency.

6.5.1 Benchmark Problems and Comparison with Algorithms

To evaluate the performance of proposed MO-EEFO algorithm, it is tested on ZDT benchmark problems and a comparative analysis is provided. These ZDT benchmark problem's distinctive characteristics and the broad range of challenges they pose make them popular for multi-objective optimization method evaluation. ZDT1's convex Pareto front makes it an excellent foundation for assessing the convergence and diversity capacities of optimization algorithms. ZDT2, on the other hand, displays a concave Pareto front, which makes it appropriate for evaluating how well algorithms manage non-convexity while preserving variety within the solution set. The comparative results are shown in Figures 6.4 and 6.5 for ZDT1 and ZDT2, respectively.

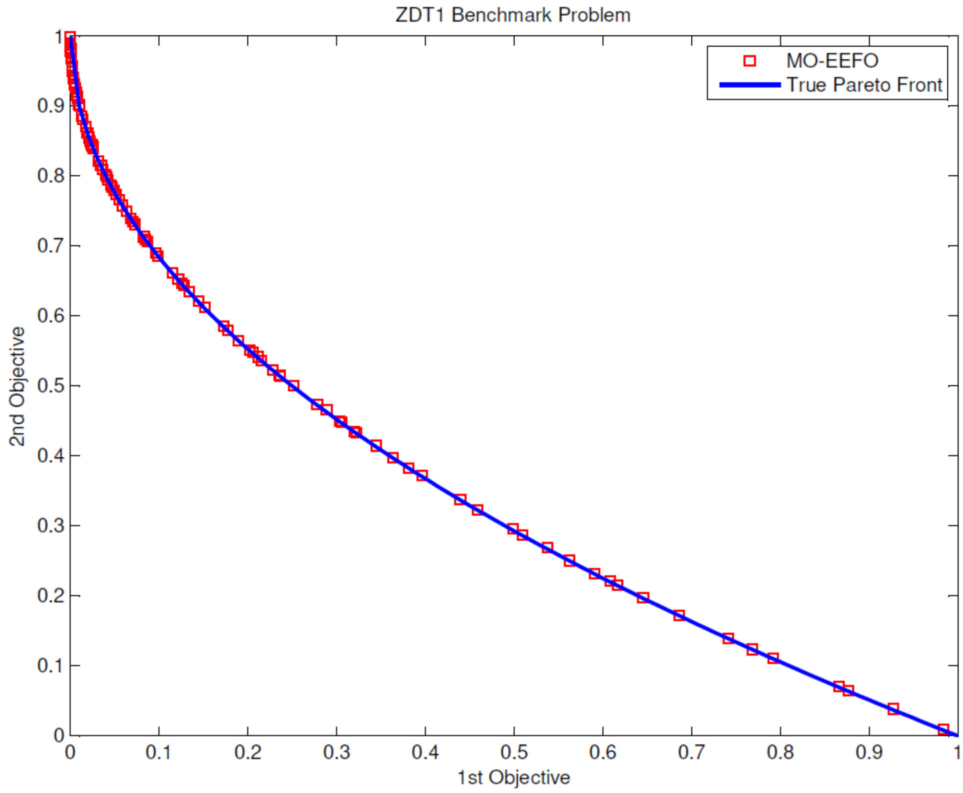


Figure 6.4: Pareto front obtained by MO-EEFO of ZDT1 function

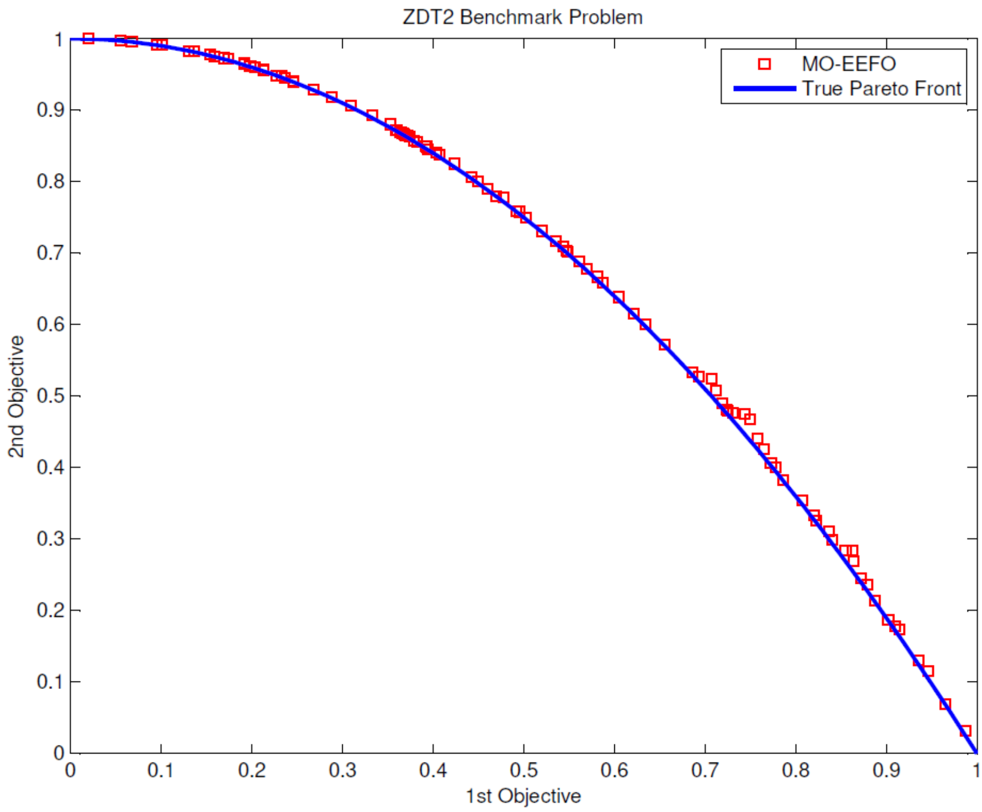


Figure 6.5: Pareto front obtained by MO-EEFO of ZDT2 function

It demonstrates that the proposed algorithm is capable of effectively navigating the solution space and consistently obtaining a distribution of solutions resembling these well-known benchmark problems. The overlap suggests that MO-EEFO, like ZDT1 and ZDT2, continues to discover the efficient trade-offs between conflicting objectives with a high degree of accuracy.

Furthermore, a comprehensive comparison of proposed MO-EEFO with a few of the well-known optimizers present in the literature including Multi-objective particle swarm optimization (MOPSO) [127], MOGSK [111], MOGA [128], Non-dominated sorting whale optimization algorithm (NSWOA) [129], and NSGA-II [105] is conducted. Parameters for all these compared algorithms are adjusted according to the data available in the literature whereas for MO-EEFO, only the population size and maximum number of iterations are set.

The analysis is regulated using a number of key performance metrics, including convergence performance, and diversity of the Pareto solutions obtained. Basic statistical measures such as range, minimum, maximum, standard deviation, mean and median are also analyzed to show the effectiveness of our proposed algorithm. Figures 6.6 and 6.7 show the comparison results of distinct optimization algorithms of ZDT1 and ZDT2 with MO-EEFO, respectively.

As it can be seen in Figures 6.6 and 6.7, the proposed MO-EEFO algorithm continuously performs equivalent to other algorithms in terms of convergence to the true Pareto front. Additionally, it maintains diversity throughout the optimization process by yielding a set of solutions that are evenly dispersed across the Pareto front.

To get a clear picture of the Pareto solutions obtained through various meta-heuristic algorithms, statistical analysis is tabulated in Table 6.1.

The proposed algorithm shows a consistent superiority over the other algorithms concerning the average objective values. Its reduced standard deviation suggested increased resilience and stability in a variety of problem scenarios. This shows that when compared to the other well-established algorithms, the proposed algorithm produces better average solutions as well as more reliable results.

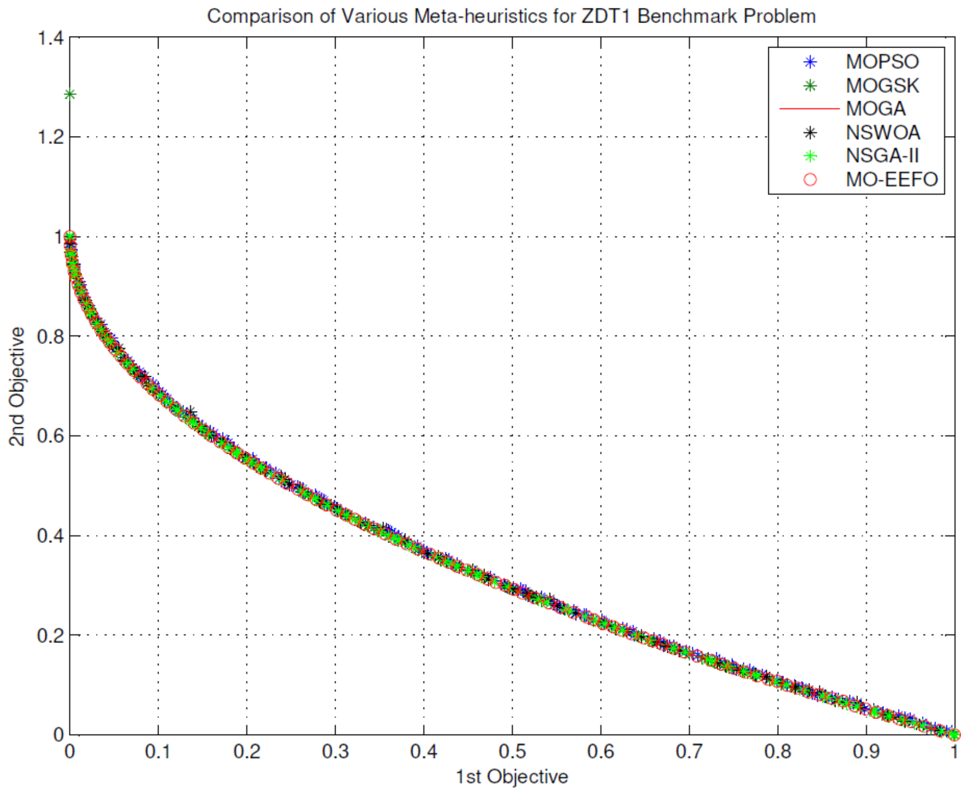


Figure 6.6: Comparison of various optimization algorithms of ZDT1 with MO-EEFO

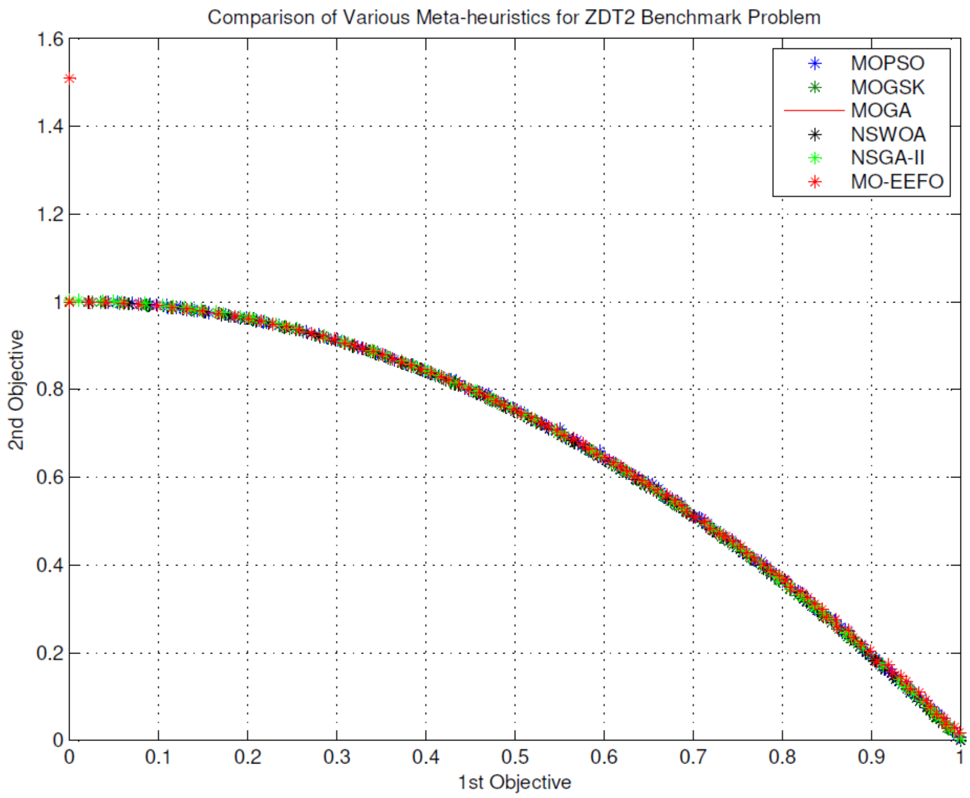


Figure 6.7: Comparison of various optimization algorithms of ZDT2 with MO-EEFO

Table 6.1: Statistical analysis of various algorithms for ZDT problems

Benchmark Problem	Algorithm	Objectives	Maximum	Minimum	Standard Deviation	Mean	Median	Mode	Range
ZDT1	MOPSO	1st objective	1	0	0.3086	0.4239	0.3934	0	1
		2nd objective	1	0	0.2793	0.4154	0.3763	0	1
	MOGSK	1st objective	0.9998	6.03e-05	0.3014	0.3885	0.3541	6.03e-05	0.9998
		2nd objective	1.286	0.0001079	0.295	0.4495	0.405	0.0001079	1.286
	MOGA	1st objective	1	0	0.293	0.5	0.5	0	1
		2nd objective	1	0	0.2421	0.3352	0.2929	0	1

	NSWOA	1st objective	1	0	0.3161	0.4056	0.3464	0	1
		2nd objective	1	0	0.2895	0.4344	0.4179	0	1
	NSGA-II	1st objective	1	0	0.3132	0.4073	0.3556	0	1
		2nd objective	1.001	0.0004385	0.2859	0.4293	0.4044	0.0004385	1
	MO-EEFO	1st objective	1	1.183e-09	0.3097	0.4156	0.3869	1.183e-09	1
		2nd objective	1	0.0003485	0.2845	0.4214	0.3785	0.0003485	0.9996
ZDT2	MOPSO	1st objective	1	0	0.2735	0.6001	0.6372	0	1
		2nd objective	1	0	0.3074	0.5703	0.5997	0	1

	MOGSK	1st objective	0.9915	2.373e-30	0.2903	0.5454	0.5726	2.373e-30	0.9915
		2nd objective	1	0.02222	0.3072	0.6197	0.6721	0.02222	0.9778
	MOGA	1st objective	1	0	0.293	0.5	0.5	0	1
		2nd objective	1	0	0.3028	0.665	0.75	0	1
	NSWOA	1st objective	1	0	0.2822	0.588	0.6233	0	0.2822
		2nd objective	1	0	0.3097	0.5765	0.6147	0	0.3097
	NSGA-II	1st objective	1	0	0.2952	0.5803	0.6184	0	1
		2nd objective	1.003	0.006427	0.3244	0.5821	0.6229	0.006427	0.9968

	MO-EEFO	1st objective	0.9991	2.727e-08	0.2818	0.5748	0.6063	2.727e-08	0.9991
		2nd objective	1.509	0.01824	0.3149	0.6024	0.6391	0.01824	1.491

6.6 Comparison of proposed MO-EEFO with other meta-heuristics

To ensure the effectiveness of the proposed algorithm in real-world scenarios, service composition optimization in smart agriculture is considered. The dataset defined in Table 3.1 which contains a set of services required for apple plant production in Shimla and Kullu regions is used to validate the proposed MO-EEFO algorithm.

These composited services are optimized using four distinct meta-heuristic optimizers named MO-EEFO, NSGA-II, MOGSK, and MOGA. Comparison results are illustrated in Figure 6.8.

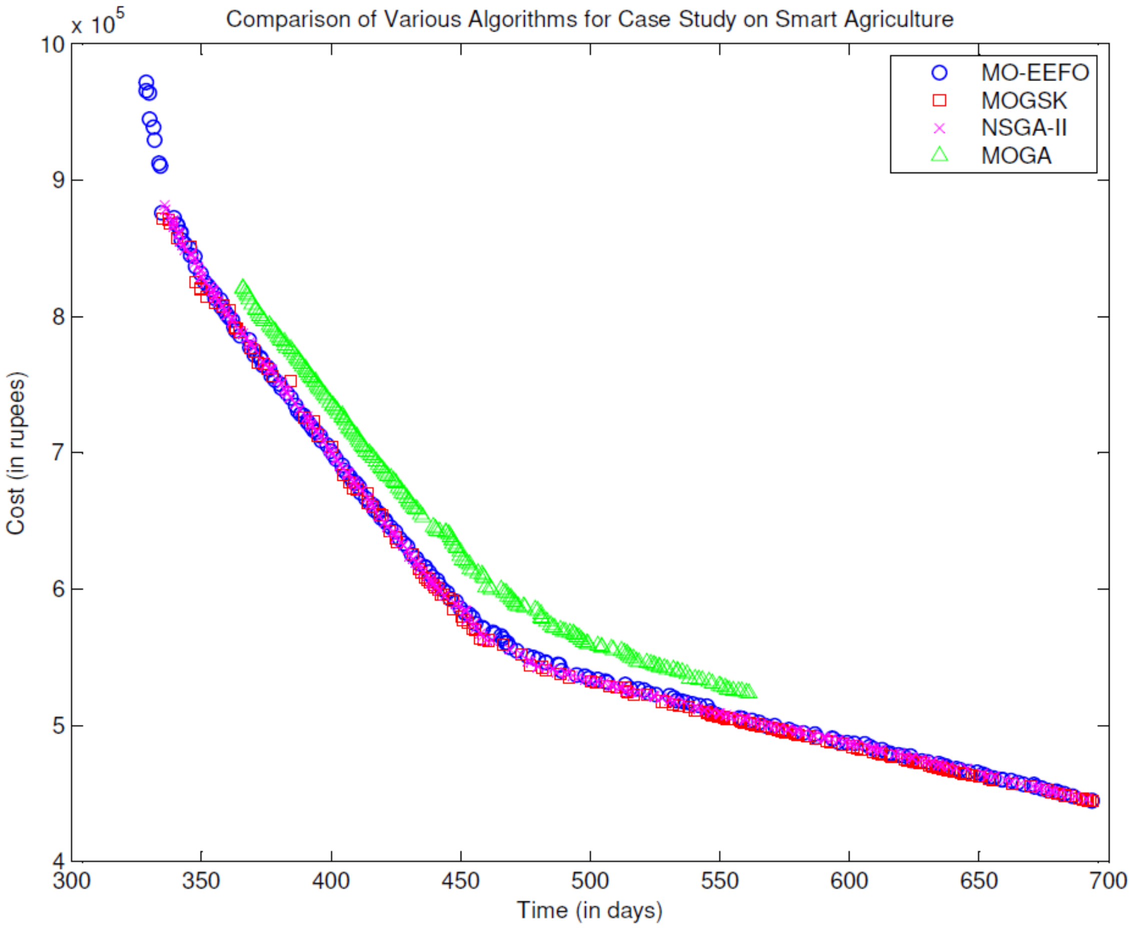


Figure 6.8: Comparative analysis of various algorithms for service composition in smart agriculture

It is evident from examining Figure 6.8 that the proposed MO-EEFO provides the best solutions for this real-world application of smart agriculture. The closeness to the origin suggests that the multi-objective optimization of minimizing both time and cost for service composition optimization in smart agriculture is completed with exceptional performance. Additionally, a greater number of Pareto points obtained from MO-EEFO facilitates a broader set of potential solutions, thereby, offering more options for users. In summary, the results imply that MO-EEFO, excels in providing a more reliable and efficient method for managing the trade-offs present in this complex optimization problem in comparison to other algorithms by

providing more solution diversity. Statistical analysis for the same is provided in Table 6.2 to get a clearer understanding of the algorithms.

Table 6.2: Statistical analysis of various compared algorithms

Algorithm	**Objectives**	**Maximum**	**Minimum**	**Standard Deviation**	**Mean**	**Median**	**Mode**	**Range**
MO-EEFO	Time	693.6	328.8	109.7	478.2	450.9	693.6	364.8
	Cost	9.716e+05	4.447e+05	1.424e+05	6.244 e+05	5.841 e+05	4.447e+05	5.269 e+05
MOGSK	Time	693.8	335.2	103	535.8	552.3	335.2	358.6
	Cost	8.716e+05	4.445e+05	1.117e+05	5.509 e+05	5.055 e+05	4.445e+05	4.271 e+05
NSGA-II	Time	691.6	335.8	105.1	480.8	455.5	335.8	335.8
	Cost	8.817e+05	4.459e+05	1.326e+05	6.147 e+05	5.731 e+05	4.459e+05	4.358 e+05
MOGA	Time	561.2	366	55.43	448.4	441.4	366	195.2
	Cost	8.208e+05	5.236e+05	9.088e+04	6.535 e+05	6.427 e+05	5.236e+05	2.972 e+05

Two evaluation methodologies have been used to show the superiority of the proposed MO-EEFO method: statistical analysis and Pareto front analysis.

The Pareto front produced by the proposed MO-EEFO method, as shown in Figure 6.8, is more diverse than that of compared algorithms, indicating that it can investigate a wider range of solutions. This increased diversity guarantees a more thorough depiction of trade-offs between competing objectives, which is a critical feature of multi-objective optimization. Furthermore, Table 6.2 displays the outcomes of the statistical analysis for several compared algorithms. It is evident that MO-EEFO produces solutions with a higher standard deviation than other algorithms, demonstrating its superior ability to produce diversified solutions. In this case, a higher standard deviation emphasizes the algorithm's capacity to investigate and preserve a wider range of solutions, proving its efficiency in striking a balance between time and cost minimization goals. When taken as a whole, these analyses present compelling proof that MO-EEFO performs better in terms of solution diversity and quality, which makes it a reliable option for resolving multi-objective optimization issues in the composition of smart agriculture services.

6.7 Summary

This chapter introduces a new nature-inspired algorithm called the multi-objective electric eel foraging optimization. The algorithm's effectiveness has been assessed through tests on standard benchmark problems, specifically ZDT1 and ZDT2. Its performance has been then compared to several well-established algorithms in the field, including MOPSO, MOGSK, MOGA, NSWOA, and NSGA-II. To further evaluate the MO-EEFO's capabilities, it has been applied to optimize service composition in smart agriculture, with its results compared against MOGA, NSGA-II, and MOGSK. The findings reveal that the MO-EEFO algorithm surpasses these alternative methods, as evidenced by its higher standard deviation. This indicates that the MO-EEFO offers superior solution diversity and robustness when tackling multi-objective optimization challenges.

CHAPTER-7
CONCLUSION AND FUTURE WORK

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The work in this thesis presents various EC techniques to solve the problem of multi-objective service composition optimization in the field of smart agriculture. In the context of smart agriculture, farmers may choose crops that will yield the most under the current and predicted climatic conditions because they have much more freedom and knowledge. Because of these advances in artificial intelligence, people's expectations have increased, resulting in complicated user demands in day-to-day life. Therefore, meeting user expectations can often be difficult. The process of combining services to meet user's complicated needs is called service composition. Put otherwise, a collection of fundamental services is what is referred to as service composition. It is an NP-hard problem so cannot be solved in the polynomial time domain thereby making traditional methods inadequate. Numerous EC approaches have been investigated in the literature to handle this complexity, providing potential solutions for these kinds of challenging optimization issues. The high-dimensional and non-linear character of service composition can be effectively addressed by EC techniques like GA, PSO, and ACO, which offer adaptive search capabilities. These techniques efficiently traverse the large solution space by mimicking evolutionary principles, providing near-optimal answers in a reasonable amount of time. As a result, EC-based methods are becoming more and more popular for optimizing service composition in intricate computational settings.

In our work, multi-objective service composition optimization in smart agriculture is done by using various EC techniques. The thesis is organized around four key objectives. Using three EC techniques - MOGA, NSGA-II, and MOGSK - the first objective focuses on linear multi-objective service composition optimization for a more straightforward approach. Cost and time are identified as the optimization problem's minimizing objective functions with a linear relationship between them. To choose the best EC technique for the defined problem, Pareto front analysis and statistical analysis are taken. According to simulation results, NSGA-II performs better than the other approaches and generates a wider variety of Pareto optimal solutions, as demonstrated by Pareto front analysis. Furthermore, NSGA-II exhibits a bigger standard deviation, which also supports its enhanced ability to produce diversified optimal solutions, making it possible for farmers to choose from the wider range of solutions available as per their requirements.

The second objective deals with non-linear multi-objective service composition optimization, in which cost and time objectives have a non-linear relationship. Lagrange's interpolation method is used to capture this non-linearity. This non-linear method is crucial since linear models are unable to adequately represent the intricacies and intrinsic non-linearities found in practical smart agriculture systems. To assess optimization performance under these non-linear conditions, three EC techniques - MOGA, NSGA-II, and MOGSK are adapted and named La-MOGA, La-NSGA-II, and La-MOGSK, respectively. La-NSGA-II performs better than the other approaches, according to Pareto front and statistical analysis. It generates a more varied range of Pareto optimal solutions and has a higher standard deviation, which suggests that it is better at handling the multi-objective service composition problem's non-linearities present in smart agriculture.

Environmental, human-based, and economic uncertainty are all unavoidable in real-world agricultural scenarios. Thus, to provide reliable and efficient solutions suited to the ever-changing requirements of smart agriculture environments, the influence of uncertainties on the optimization process is examined in the third objective. It applies fuzzy logic to both linear and non-linear multi-objective service composition optimization problems to evaluate the influence of those uncertainties on the optimization of composited services. NSGA-II is employed as the optimization algorithm for this objective since it outperformed MOGA and MOGSK in both the first and second objectives. Fuzzy-Li-NSGA-II for linear optimization problems and Fuzzy-La-NSGA-II for non-linear optimization problems are the modified versions of NSGA-II used in this objective for checking the influence of uncertainties using the Mamdani fuzzy inference system. For Fuzzy-Li-NSGA-II, four case scenarios are assessed: the worst (MS=WC=FS=0.2), the normal (MS=WC=FS=0.5), the mixed (MS=0.5, WC=0.8, FS=0.2), and the best-case (MS=WC=FS=0.9). Comparable situations for Fuzzy-La-NSGA-II are also evaluated using modified values: best-case (MS=WC=FS=0.8), mixed (MS=0.5, WC=0.9, FS=0.3), normal (MS=WC=FS=0.5), and worst (MS=WC=FS=0.3). According to the behavioral analysis, Fuzzy-La-NSGA-II more accurately depicts real-world conditions than Fuzzy-Li-NSGA-II. Furthermore, there are minor differences between the Pareto solutions produced by the two methods, with Fuzzy-La-NSGA-II better capturing the influence of the uncertainties and non-linearities present in real-world applications.

The fourth objective deals with developing a novel nature-inspired multi-objective electric eel foraging optimization algorithm for solving challenges in real-world applications. The proposed MO-EEFO reflects on the ingenious collective foraging strategies of electric eels found in nature and considers them as an inspiration for the optimization process. Its performance is validated on ZDT benchmark problems. Furthermore, a comprehension comparison of this proposed MO-EEFO is done with a few well-established algorithms present in the literature which are MOPSO, MOGSK, MOGA, NSWOA, and NSGA-II. It has been found that the proposed MO-EEFO algorithm continuously performs equivalent to other algorithms in terms of convergence to the true Pareto front. Additionally, it maintains diversity throughout the optimization process by yielding a set of solutions that are evenly dispersed across the Pareto front. To check its effectiveness in real-world scenarios, it is tested against MOGA, NSGA-II, and MOGSK for service composition optimization in smart agriculture applications. The simulation observations show that it provides more diversified Pareto optimal solutions, with a higher standard deviation as well.

In conclusion, this thesis work explores multi-objective service composition optimization in smart agriculture applications using various EC techniques along with the evolution of a novel nature-inspired MO-EEFO algorithm to meet real-world optimization challenges.

7.2 Future Work

This thesis focuses on multi-objective service composition in smart agriculture applications using distinct EC techniques. Future expansions of this work could include:

a) Integration of emerging EC techniques: Future research could explore the application of emerging nature-inspired algorithms, such as orcha predation algorithm (OPA), remora optimization algorithm (ROA), Ivy algorithm (IVYA) etc., which may offer enhanced performance, unique search dynamics, and improved convergence rates for the optimization of smart agriculture, along with the complexity analysis.
b) Meta-optimization for algorithm enhancement: Future research could apply meta-optimization techniques such as Bayesian optimization or reinforcement learning to adjust the parameters of nature-inspired algorithms dynamically.
c) Exploring hybrid algorithms: Future research may involve developing hybrid optimization algorithms to enhance diversity and improve solution quality.

d) Combining with machine learning: The work could be integrated with different machine learning models for predictive analysis, allowing for more informed decision-making by forecasting crop yields, pest infestations, or optimal planting times.
e) Industry collaboration for real-world validation: Future research could involve collaboration with smart agriculture companies to validate the optimization approaches in real-world scenarios. Through this collaboration, real-world challenges such as operational restrictions, data limitations, and environmental unpredictability can be identified. Feedback from stakeholders will help improve the models and influence future studies, with an emphasis on the agricultural system's scalability, real-time adaptation, and economic viability.
f) Incorporating Financial factors: Future research could look into incorporating financial modelling elements as long-term orchard investment planning, delayed profitability, and borrowing costs. This would enable smarter financial decision-making for stakeholders in real-world smart agriculture projects.
g) Incorporating IoT technology: The optimization framework could be enhanced by integrating various Internet of Things (IoT) sensors, enabling real-time, data-driven decisions based on soil conditions, crop growth, and weather patterns.
h) Integration of socio-economic and policy factors: Future studies might incorporate socio-economic elements such as labor availability, government regulations, and market demands into the optimization framework. This would allow for service composition decisions that not only optimize time and cost but also align with local socio-economic contexts.

REFERENCES

REFERENCES

[1] S. Herath, H. Mohri, P. Wong, B. Mishra, D. S. Sampath, S. B. Weerakoon, S. Atapattu and D. Woldie, "Traditional and modern agriculture mosaic system for improving resilience to global change," in *Proceedings of the Conference on Sri Lanka—Japan Collaborative Research, SJLCR, Peradeniya*, Sri Lanka, 2013.

[2] R. Abbasi, P. Martinez and R. Ahmad, "The digitization of agricultural industry-A systematic literature review on Agriculture 4.0," *Smart Agricultural Technology,* vol. 2, p. 100042, 2022.

[3] M. M. Bhutta and M. Ahmad, "Secure identification, traceability and real-time tracking of agricultural food supply during transportation using internet of things," *IEEE Access,* vol. 9, pp. 65660-65675, 2021.

[4] N. Chamara, M. D. Islam, G. F. Bai, Y. Shi and Y. Ge, "Ag-IoT for crop and environment monitoring: Past, Present, and Future," *Agricultural Systems,* vol. 203, p. 103497, 2022.

[5] A. F. Subahi and K. E. Bouazza, "An intelligent IoT-based system design for controlling and monitoring greenhouse temperature," *IEEE Access,* vol. 8, pp. 125488-125500, 2020.

[6] A. P. Nugroho, L. Sutiarso and T. Okayasu, "Appropriate adaption of precision agriculture technology in open field cultivation in tropics," in *3rd International Symposium on Agricultural and Biosystem Engineering, IOP Conference Series: Earth and Environmental Science, IOP Publishing*, South Sulawesi, Indonesia, 2019.

[7] P. Fincheira, N. Hoffmann, G. Tortella, A. Ruiz, P. Cornejo, M. C. Diez, A. B. Seabra, A. Benavides-Mendoza and O. Rubilar, "Eco-efficient systems based on nanocarriers for the controlled release of fertilizers and pesticides: Toward smart agriculture," *Nanomaterials,* vol. 13, no. 13, p. 1978, 2023.

[8] S. Terence and G. Purushothaman, "Systematic review of internet of things in smart farming," *Transactions on Emerging Telecommunications Technologies ,* vol. 31, p. 3958, 2020.

[9] R. Boucetti, S. M. Hemam and O. Hioual, "QoS aware IoT services composition: A survey," in *Distributed Sensing and Intelligent Systems*, Cham, Springer, 2022.

[10] R. Boucetti, O. Hioual and S. M. Hemam, "An approach basaed on genetic algorithms and neural networks for QoS-aware IoT services composition," *Journal of King Saud University-Computer and Information Sciences,* vol. 34, no. 8, pp. 5619-5632, 2022.

[11] H. Kurdi, F. Ezzat, L. Altoaimy, S. H. Ahmed and K. Youcef-Toumi, "Multicuckoo: Multi-cloud service composition using a cuckoo-inspired algorithm for the internet of things applications," *IEEE Access,* vol. 6, pp. 56737--56749, 2018.

[12] L. Bianchi, M. Dorigo, L. M. Gambardella and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing,* vol. 8, pp. 239-287, 2009.

[13] T. W. Bank, "World Bank Data," [Online]. Available: https://data.worldbank.org/indicator/SL.AGR.EMPL.ZS?locations=IN. [Accessed 07 February 2024].

[14] FAO, "How to Feed the World in 2050," Food and Agriculture Organization of the United Nations, Rome, 2009.

[15] J. Xu, B. Gu and G. Tian, "Review of agricultural IoT technology," *Artificial Intelligence in Agriculture,* vol. 6, pp. 10-22, 2022.

[16] V. Sharma, A. K. Tripathi and H. Mittal, "Technological revolutions in smart farming: Current trends, challenges & future directions," *Computers and Electronics in Agriculture,* vol. 201, p. 107217, 2022.

[17] K. K. Lau and S. D. Cola, "An introduction to component-based software developement," *World Scientific*, 2018.

[18] D. Arellanes and K. K. Lau, "Evaluating IoT service composition mechanisms for the scalability of IoT systems," *Future Generation Computer Systems,* vol. 108, pp. 827-848, 2020.

[19] R. Zeng, X. Hou, L. Zhang, C. Li, W. Zheng and M. Guo, "Performance optimization for cloud computing systems in the microservice era: state-of-the-art and research opportunities," *Frontiers of Computer Science,* vol. 6, p. 166106, 2022.

[20] H. Y. Paik, A. L. Lemos, M. C. Barukh, B. Benatallah and A. Natarajan, Web service implementation and composition techniques, *Springer*, 2017.

[21] I. J. Taylor, E. Deelman, D. B. Gannon and M. Shields, "Workflows for e-Science: scientific workflows for grids," *Springer*, 2007.

[22] A. L. Lemos, F. Daniel and B. Benatallah, "Web service composition: a survey of techniques and tools," *ACM Computing Surveys (CSUR),* vol. 48, no. 3, pp. 1-41, 2015.

[23] P. Asghari, A. M. Rahmani and H. H. S. Javadi, "Service composition approaches in IoT: A systematic review," *Journal of Network and Computer Applications,* vol. 120, pp. 61-77, 2018.

[24] M. E. Khansari, S. Sharifian and S. A. Motamedi, "Virtual sensor as a service: a new multicriteria QoS-aware cloud service composition for IoT applications," *The Journal of Supercomputing,* vol. 74, no. 10, pp. 5485-5512, 2018.

[25] M. Cremene, M. Suciu, D. Pallez and D. Dumitrescu, "Comparative analysis of multi-objective evolutionary algorithms for QoS-aware web service composition," *Applied Soft Computing,* vol. 39, pp. 124-139, 2016.

[26] J. H. Cho, Y. Wang, R. Chen, K. S. Chan and A. Swami, "A survey on modeling and optimizing multi-objective systems," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 3, pp. 1867-1901, 2017.

[27] C. Chitra and P. Subbaraj, "A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks," *Expert systems with applications,* vol. 39, no. 1, pp. 1518-1525, 2012.

[28] A. S. da Silva, H. Ma, Y. Mei and M. Zhang, "A survey of evolutionary computation for web service composition: A technical perspective," *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 4, no. 4, pp. 538-554, 2020.

[29] V. Sharma and A. K. Tripathi, "A systematic review of meta-heuristic algorithms in IoT based application," *Array,* vol. 14, p. 100164, 2022.

[30] S. Asghari and N. J. Navimipour, "Nature inspired meta-heuristic algorithms for solving the service composition problem in the cloud environments," *International Journal of Communication Systems,* vol. 31, no. 12, p. e3708, 2018.

[31] Z. N. Ansari and S. D. Daxini, "A state-of-the-art review on meta-heuristics application in remanufacturing," *Archives of Computational Methods in Engineering,* vol. 29, no. 1, pp. 427-470, 2022.

[32] J. H. Holland, "Genetic algorithms," *Scientific American,* vol. 267, no. 1, pp. 66-73, 1992.

[33] X. Pan and L. Jiao, "A granular agent evolutionary algorithm for classification," *Applied Soft Computing,* vol. 11, no. 3, pp. 3093-3105, 2011.

[34] H. Ma, D. Simon, P. Siarry, Z. Yang and M. Fei, "Biogeography-based optimization: a 10-year review," *IEEE Transactions on Emerging Topics in Computational Intelligence,* vol. 1, no. 5, pp. 391-407, 2017.

[35] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE control systems magazine,* vol. 22, no. 3, pp. 52-67, 2002.

[36] D. Dasgupta, Artificial immune systems and their applications, *Springer Science & Business Media*, 2012.

[37] W. Zhao, L. Wang and S. Mirjalili, "Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications," *Computer Methods in Applied Mechanics and Engineering,* vol. 388, p. 114194, 2022.

[38] X. Li, "An optimizing method based on autonomous animats: fish-swarm algorithm," *Systems Engineering-theory & Practice,* vol. 22, no. 11, pp. 32-38, 2002.

[39] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation,* vol. 214, no. 1, pp. 108-132, 2009.

[40] S. Bhardwaj and D. S. Kim, "Dragonfly-based swarm system model for node identification in ultra-reliable low-latency communication," *Neural Computing and Applications,* vol. 33, pp. 1837-1880, 2021.

[41] S. Kirkpatrick, C. J. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," S*cience,* vol. 220, no. 4598, pp. 671-680, 1983.

[42] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based Systems,* vol. 96, pp. 120-133, 2016.

[43] H. Eskandar, A. Sadollah, A. Bahreininejad and M. Hamdi, "Water cycle algorithm-A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures,* vol. 110, pp. 151-166, 2012.

[44] R. V. Rao, V. J. Savsani and D. P. Vakharia, "Teaching--learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences,* vol. 183, no. 1, pp. 1-15, 2012.

[45] Y. Shi, "Brain storm optimization algorithm," in *Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Proceedings, Part I 2*, Chongqing, China, Springer, June 12-15, 2011, pp. 303-309.

[46] A. H. Kashan, "League championship algorithm: a new algorithm for numerical function optimization," in *2009 IEEE International Conference of Soft Computing and Pattern Recognition*, Malacca, Malaysia, 2009.

[47] J. Zhang, M. Xiao, L. Gao and Q. Pan, "Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems," *Applied Mathematical Modelling,* vol. 63, pp. 464-490, 2018.

[48] A. Y. Lam and V. O. Li, "Chemical reaction optimization for cognitive radio spectrum allocation," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, FL, USA, 2010.

[49] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments,* vol. 9, no. 4, pp. 395-420, 2017.

[50] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture," *Computers and Electronics in Agriculture,* vol. 157, pp. 218-231, 2019.

[51] W. Tao, L. Zhao, G. Wang and R. Liang, "Review of the internet of things communication technologies in smart agriculture and challenges," *Computers and Electronics in Agriculture,* vol. 189, p. 106352, 2021.

[52] A. Srivastava and D. K. Das, "A comprehensive review on the application of Internet of Thing (IoT) in smart agriculture," *Wireless Personal Communications,* vol. 122, no. 2, pp. 1807-1837, 2022.

[53] V. P. Kour and S. Arora, "Recent developments of the internet of things in agriculture: A survey," *IEEE Access,* vol. 8, pp. 129924-129957, 2020.

[54] E. G. Symeonaki, K. G. Arvanitis and D. D. Piromalis, "Cloud computing for IoT applications in climate-smart agriculture: A review on the trends and challenges toward sustainability," in *Innovative Approaches and Applications for Sustainable Rural Development: 8th International Conference, HAICTA 2017, Selected Papers 8*, Chania, Crete, Greece, Springer, September 21-24, 2017, pp. 147-167.

[55] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of Internet of Things in smart agriculture: A survey," *Future Generation Computer Systems,* vol. 126, pp. 169-184, 2022.

[56] V. Saiz-Rubio and F. Rovira-Más, "From smart farming towards agriculture 5.0: A review on crop data management," *Agronomy,* vol. 10, no. 2, p. 207, 2020.

[57] C. Maraveas, P. G. Asteris, K. G. Arvanitis, T. Bartzanas and D. Loukatos, "Application of bio and nature-inspired algorithms in agricultural engineering," *Archives of Computational Methods in Engineering,* vol. 30, no. 3, pp. 1979-2012, 2023.

[58] M. Masdari, M. N. Bonab and S. Ozdemir, "QoS-driven metaheuristic service composition schemes: a comprehensive overview," *Artificial Intelligence Review,* vol. 54, pp. 3749-3816, 2021.

[59] M. Pathan, N. Patel, H. Yagnik and M. Shah, "Artificial cognition for applications in smart agriculture: A comprehensive review," *Artificial Intelligence in Agriculture,* vol. 4, pp. 81-95, 2020.

[60] S. Qazi, B. A. Khawaja and Q. U. Farooq, "IoT-equipped and AI-enabled next generation smart agriculture: A critical review, current challenges and future trends," *IEEE Access,* vol. 10, pp. 21219-21235, 2022.

[61] A. De and S. P. Singh, "Analysis of fuzzy applications in the agri-supply chain: A literature review," *Journal of Cleaner Production,* vol. 283, p. 124577, 2021.

[62] F. Valdez, P. Melin and O. Castillo, "A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation," *Expert systems with applications,* vol. 41, no. 14, pp. 6459-6466, 2014.

[63] A. Sharma, A. Jain, P. Gupta and V. Chowdary, "Machine learning applications for precision agriculture: A comprehensive review," *IEEE Access,* vol. 9, pp. 4843-4873, 2020.

[64] R. Akhter and S. A. Sofi, "Precision agriculture using IoT data analytics and machine learning," *Journal of King Saud University-Computer and Information Sciences,* vol. 34, no. 8, pp. 5602-5618, 2022.

[65] S. Wolfert, L. Ge, C. Verdouw and M. J. Bogaardt, "Big data in smart farming--a review," *Agricultural Systems,* vol. 153, pp. 69-80, 2017.

[66] A. d. Ocampo and E. P. Dadios, "Energy cost optimization in irrigation system of smart farm by using genetic algorithm," in *2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Manila, Philippines, 2017.

[67] H. Hakli and H. Uğuz, "A novel approach for automated land partitioning using genetic algorithm," *Expert Systems with Applications,* vol. 82, pp. 10-18, 2017.

[68] S. K. Roy and D. De, "Genetic algorithm based internet of precision agricultural things (IopaT) for agriculture 4.0," *Internet of Things,* vol. 18, p. 100201, 2022.

[69] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*, Newark, NJ, USA, 2020.

[70] X. Gaofeng, "Optimise financial cost for smart agriculture based on genetic algorithm," *International Journal of Wireless and Mobile Computing,* vol. 19, no. 1, pp. 80-86, 2020.

[71] D. P. Acharjya and R. Rathi, "An integrated fuzzy rough set and real coded genetic algorithm approach for crop identification in smart agriculture," *Multimedia Tools and Applications,* vol. 81, no. 24, pp. 35117-35142, 2022.

[72] R. I. Mukhamediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov and Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous UAVs group for precision agriculture," *IEEE Access,* vol. 11, pp. 5789-5803, 2023.

[73] F. Kiani, A. Seyyedabbasi, S. Nematzadeh, F. Candan, T. Çevik, F. A. Anka, G. Randazzo, S. Lanza and A. Muzirafuti, "Adaptive metaheuristic-based methods for autonomous robot path planning: sustainable agricultural applications," *Applied Sciences,* vol. 12, no. 3, p. 943, 2022.

[74] S. P. Singh, G. Dhiman, S. Juneja, W. Viriyasitavat, G. Singal, N. Kumar and P. Johri, "A New QoS Optimization in IoT-Smart Agriculture Using Rapid Adaption Based Nature-Inspired Approach," *IEEE Internet of Things Journal,* vol. 11, no. 3, pp. 5417-5426, 2024.

[75] H. Babazadeh, M. S. Tabrizi and G. Hoogenboom, "Crop Production and Water Productivity Simultaneously Optimization of Soybean Plant Using Two Meta-Heuristic Algorithms," *Agricultural Research,* vol. 39, pp. 311-325, 2022.

[76] S. Shamshirband, B. Khoshnevisan, M. Yousefi, E. Bolandnazar, N. B. Anuar, A. A. Wahab and S. U. R. Khan, "A multi-objective evolutionary algorithm for energy management of agricultural systems—a case study in Iran," *Renewable and Sustainable Energy Reviews,* vol. 44, pp. 457-465, 2015.

[77] N. Sivakumar, T. Amudha and N. Thilagavathi, "Development of a Novel Bio Inspired Framework for Fertilizer Optimization," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates, 2019.

[78] N. Thilagavathi, S. Ramakrishnan and T. Amudha, "A novel bio-inspired optimization framework for effective crop land allocation and utilization," in *2021 2nd International*

Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021.

[79] B. Saeidian, M. S. Mesgari, B. Pradhan and A. M. Alamri, "Irrigation water allocation at farm level based on temporal cultivation-related data using meta-heuristic optimisation algorithms," *Water,* vol. 11, no. 12, p. 2611, 2019.

[80] G. Sushanth and S. Sujatha, "IOT based smart agriculture system," in *2018 international conference on wireless communications, signal processing and networking (WiSPNET)*, Chennai, India, 2018.

[81] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat and P. Nillaor, "IoT and agriculture data analysis for smart farm," *Computers and Electronics in Agriculture,* vol. 156, pp. 467-474, 2019.

[82] P. P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed and A. Zaslavsky, "Internet of things platform for smart farming: Experiences and lessons learnt," *Sensors,* vol. 16, no. 11, p. 1884, 2016.

[83] A. Gupta and P. Nahar, "Classification and yield prediction in smart agriculture system using IoT," *Journal of Ambient Intelligence and Humanized Computing,* vol. 14, no. 8, pp. 10235-10244, 2023.

[84] O. Castillo and L. Amador-Angulo, "A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design," *Information Sciences,* vol. 460, pp. 476-496, 2018.

[85] C. Caraveo, F. Valdez and O. Castillo, "A new meta-heuristics of optimization with dynamic adaptation of parameters using type-2 fuzzy logic for trajectory control of a mobile robot," *Algorithms,* vol. 10, no. 3, p. 85, 2017.

[86] M. Guerrero, O. Castillo and M. García, "Fuzzy dynamic parameters adaptation in the Cuckoo Search Algorithm using fuzzy logic," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, Sendai, Japan, 2015.

[87] F. Olivas, F. Valdez, P. Melin, A. Sombra and O. Castillo, "nterval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm," *Information Sciences,* vol. 476, pp. 159-175, 2019.

[88] G. Lavanya, C. Rani and P. GaneshKumar, "An automated low cost IoT based Fertilizer Intimation System for smart agriculture," *Sustainable Computing: Informatics and Systems,* vol. 28, p. 100300, 2020.

[89] J. R. Dela Cruz, R. G. Baldovino, F. B. Culibrina, A. A. Bandala and E. P. Dadios, "Fuzzy-based decision support system for smart farm water tank monitoring and control," in *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, Melaka, Malaysia, 2017.

[90] R. P. Sharma, D. Ramesh, P. Pal, S. Tripathi and C. Kumar, "IoT-enabled IEEE 802.15. 4 WSN monitoring infrastructure-driven fuzzy-logic-based crop pest prediction," *IEEE Internet of Things Journal,* vol. 9, no. 4, pp. 3037-3045, 2021.

[91] H. Benyezza, M. Bouhedda and S. Rebouh, "Zoning irrigation smart system based on fuzzy control technology and IoT for water and energy saving," *Journal of Cleaner Production,* vol. 302, p. 127001, 2021.

[92] N. Kashyap and A. C. Kumari, "Hyper-heuristic approach for service composition in internet of things," *Electronic Government, an International Journal,* vol. 14, no. 4, pp. 321-339, 2018.

[93] P. Asghari, A. M. Rahmani and H. Haj Seyyed Javadi, "A medical monitoring scheme and health-medical service composition model in cloud-based IoT platform," *Transactions on Emerging Telecommunications Technologies,* vol. 30, no. 6, p. E3637, 2019.

[94] N. Kashyap, A. C. Kumari and R. Chhikara, "Service Composition in IoT using Genetic algorithm and Particle swarm optimization," *Open Computer Science,* vol. 10, no. 1, pp. 56-64, 2020.

[95] N. Kashyap, A. C. Kumari and R. Chhikara, "Multi-objective Optimization using NSGA II for service composition in IoT," *Procedia Computer Science,* vol. 167, pp. 1928-1933, 2020.

[96] M. Razian, M. Fathian and R. Buyya, "ARC: Anomaly-aware Robust Cloud-integrated IoT service composition based on uncertainty in advertised quality of service values," *Journal of Systems and Software,* vol. 164, p. 110557, 2020.

[97] S. Sefati and N. J. Navimipour, "A QoS-aware service composition mechanism in the internet of things using a hidden-markov-model-based optimization algorithm," *IEEE Internet of Things Journal,* vol. 8, no. 20, pp. 15620-15627, 2021.

[98] V. Pavan Kumar, S. Shetty, D. R. Janardhana and A. P. Manu, "Qos aware service composition in iot using heuristic structure and genetic algorithm," *Mathematical Statistician and Engineering Applications,* vol. 71, no. 3, pp. 750-766, 2022.

[99] R. Boucetti, O. Hioual and S. M. Hemam, "An approach based on genetic algorithms and neural networks for QoS-aware IoT services composition," *Journal of King Saud University-Computer and Information Sciences,* vol. 34, no. 8, pp. 5619-5632, 2022.

[100] Z. Zhai, J. F. Martínez Ortega, N. Lucas Martínez and J. Rodríguez-Molina, "A mission planning approach for precision farming systems based on multi-objective optimization," *Sensors,* vol. 18, no. 6, p. 1795, 2018.

[101] K. Rani and V. Kumar, "Solving travelling salesman problem using genetic algorithm based on heuristic crossover and mutation operator," *International Journal of Research in Engineering & Technology,* vol. 2, no. 2, pp. 27-34, 2014.

[102] Z. Y. Chai, S. -S. Fang and Y. -L. Li, "An improved decomposition-based multiobjective evolutionary algorithm for IoT service," *IEEE Internet of Things Journal,* vol. 8, no. 2, pp. 1109-1122, 2020.

[103] K. Deb, "Analysing mutation schemes for real-parameter genetic algorithms, " *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 1, pp.1-28, 2014.

[104] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation,* vol. 2, no. 3, pp. 221-248, 1994.

[105] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[106] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction", *Multi-objective evolutionary optimisation for product design and manufacturing,* Springer, pp. 3-34, 2011.

[107] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems,* vol. 9, no. 2, pp. 115-148, 1995.

[108] G. -Q. Zeng, J. Chen, L. -M. Li, M. -R. Chen, L. Wu and Y. -X. Dai, "An improved multi-objective population-based extremal optimization algorithm with polynomial mutation," *Information Sciences,* vol. 330, pp. 49-73, 2016.

[109] A. W. Mohamed, A. A. Hadi and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics,* vol. 11, no. 7, pp. 1501-1529, 2020.

[110] A. W. Mohamed, A. A. Hadi, P. Agrawal, K. M. Sallam and A. K. Mohamed, "Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with IMODE algorithm for solving CEC 2021 benchmark problems," in *021 IEEE Congress on Evolutionary Computation (CEC)*, Kraków, Poland, 2021.

[111] N. E. Chalabi, A. Attia, K. A. Alnowibet, H. M. Zawbaa, H. Masri and A. W. Mohamed, "A multi--objective gaining--sharing knowledge-based optimization algorithm for solving engineering problems," *Mathematics,* vol. 11, no. 14, p. 3092, 2023.

[112] P. Agrawal, T. Ganesh and A. W. Mohamed, "A novel binary gaining--sharing knowledge-based optimization algorithm for feature selection," *Neural Computing and Applications,* vol. 33, no. 11, pp. 5989-6008, 2021.

[113] Z. F. Wang, Y. -W. Ren, Z. -Y. Cao and L. -Y. Zhang, "LRBFT: Improvement of practical Byzantine fault tolerance consensus protocol for blockchains based on Lagrange interpolation," *Peer-to-Peer Networking and Applications,* vol. 16, no. 2, pp. 690-708, 2023.

[114] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang and J. Zhang, "VFL: A verifiable federated learning with privacy-preserving for big data in industrial IoT," *IEEE Transactions on Industrial Informatics,* vol. 18, no. 5, pp. 3316-3326, 2020.

[115] E. Danish and M. Onder, "Application of fuzzy logic for predicting of mine fire in underground coal mine," *Safety and Health at Work,* vol. 11, no. 3, pp. 322-334, 2020.

[116] V. Ojha, A. Abraham and V. Snášel , "Heuristic design of fuzzy inference systems: A review of three decades of research," *Engineering Applications of Artificial Intelligence,* vol. 85, pp. 845-864, 2019.

[117] T. Samavat, M. Nazari, M. Ghalehnoie, M. A. Nasab, M. Zand, P. Sanjeevikumar and B. Khan, "A comparative Analysis of the Mamdani and Sugeno Fuzzy Inference systems for MPPT of an ISlanded PV System," *International Journal of Energy Research,* vol. 2023, p. 7676113, 2023.

[118] T. J. Ross, Fuzzy logic with engineering applications, *John Wiley & Sons*, 2005.

[119] B. K. Pathak and S. Srivastava, "Integrated Fuzzy--HMH for project uncertainties in time--cost tradeoff problem," *Applied Soft Computing,* vol. 21, pp. 320-329, 2014.

[120] A. -T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares and M. Sugeno, "Fuzzy control systems: Past, present and future," *IEEE Computational Intelligence Magazine,* vol. 14, no. 1, pp. 56-68, 2019.

[121] D. K. Kwaghtyo and C. I. Eke, "Smart farming prediction models for precision agriculture: A comprehensive survey," *Artificial Intelligence Review,* vol. 56, no. 6, pp. 5729-5772, 2023.

[122] Z. Kai, S. Jinchun, N. Ke and L. Song, "Lagrange interpolation learning particle swarm optimization," *PloS one,* vol. 11, no. 4, p. E0154191, 2016.

[123] R. P. Sharma, R. Dharavath and D. R. Edla, "IoFT-FIS: Internet of farm things based prediction for crop pest infestation using optimized fuzzy inference system," *Internet of Things,* vol. 21, p. 100658, 2023.

[124] N. Geographic, "Electric Eel," [Online]. Available: https://www.nationalgeographic.com/animals/fish/facts/electric-eel. [Accessed March 2024].

[125] W. Zhao, L. Wang, Z. Zhang, H. Fan, J. Zhang, S. Mirjalili, N. Khodadadi and Q. Cao, "Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications," *Expert Systems with Applications,* vol. 238, p. 122200, 2024.

[126] N. Zoo, "Electric Eel," [Online]. Available: https://nationalzoo.si.edu/animals/electric-eel. [Accessed March 2024].

[127] J. Yang, J. Zhou, L. Liu and Y. Li, "A novel strategy of pareto-optimal solution searching in multi-objective particle swarm optimization (MOPSO)," *Computers & Mathematics with Applications,* vol. 57, no. 11-12, pp. 1995-2000, 2009.

[128] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evolutinary Computation,* vol. 7, no. 3, pp. 205-230, 1999.

[129] P. Jangir and N. Jangir, "Non-dominated sorting whale optimization algorithm (NSWOA): A multi-objective optimization algorithm for solving engineering design problems," *Global Journal of Research in Engineering,* vol. 17, pp. 15-42, 2017.

LIST OF PUBLICATIONS

LIST OF PUBLICATIONS

Journal Publications

1) S. Sharma, B. K. Pathak, and R. Kumar, "Multi-objective Service Composition Optimization Problem in IoT for Agriculture 4.0," *Computing*, vol. 106, no. 12, pp. 4039-4056, September 2024. doi: 10.1007/s00607-024-01346-2.

   (Index: SCIE, SCOPUS, Impact Factor: 3.3)

2) S. Sharma, B. K. Pathak, and R. Kumar, "A Non-Linear Multi-Objective Service Composition Optimization for Smart Agriculture with Lagrange's Interpolation-based Evolutionary Algorithm," *Electrica,* vol. 24, no. 3, pp. 670-681, October 2024. doi: 10.5152/electrica.2024.24076. ***(Index: ESCI, SCOPUS)***

3) S. Sharma, B. K. Pathak, and R. Kumar, "Adopting an Improved Genetic Algorithm for Multi-Objective Service Composition Problem in Smart Agriculture," *Austrian Journal of Statistics*, vol. 53, no. 5*,* pp. 11–25, December 2024. doi: 10.17713/ajs.v53i5.1874.

   (Index: ESCI, SCOPUS)

4) S. Sharma, B. K. Pathak, and R. Kumar, "Multi-objective Service Composition Optimization in Smart Agriculture Using Fuzzy-Evolutionary Algorithm," *Operation Research Forum*, vol. 5, no. 2, pp. 1-24, May 2024. doi: 10.1007/s43069-024-00319-7.

   (Index: SCOPUS)

5) S. Sharma, B. K. Pathak, and R. Kumar, "Understanding of Network Resiliency in Communication Networks with its Integration in Internet of Things - A Survey," *Electrica,* vol. 23, no. 2, pp. 318-328, March 2023. doi: 10.5152/electrica.2023.22126.

   (Index: ESCI, SCOPUS)

Conference Publications

1) S. Sharma, R. Kumar and B. K. Pathak, "Analyzing the Impact of Uncertainties with Fuzzy Logic on Service Composition in Smart Agriculture," *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, 2024, pp. 1-5, doi:10.1109/ESCI59607.2024.10497423.

   (Scopus Indexed)

2) S. Sharma, R. Kumar and, B. K. Pathak, "Bio-Inspired Multi-Objective Evolutionary Algorithm for Service Composition Optimization in Internet of Things".

   (Accepted in Scopus Indexed Conference)

Communicated Papers

1) S. Sharma, R. Kumar and, B. K. Pathak, "Optimizing Service Composition Problem using Human Inspired Multi-Objective Gaining Sharing Knowledge Technique in Smart Agriculture," *SN Computer Science.*

(Under Review)