Egyptian Informatics Journal 24 (2023) 100418

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

Egyptian Informatics Journal

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com/)

Full Length Article

Workflow migration in uncertain edge computing environments based on interval many-objective evolutionary algorithm

Zhenyu Shi, Tianhao Zhao, Qi Li, Zhixia Zhang, Zhihua Cui [∗](#_bookmark0)

*Shanxi Key Laboratory of Big Data Analysis and Parallel Computing, Taiyuan University of Science and Technology, Taiyuan, 030024, Shanxi, China*

A R T I C L E I N F O A B S T R A C T

*Keywords:*

Edge computing Workflow migration

Interval many-objective optimization Evolutionary algorithm

In edge computing (EC), when the edge server (ES) is processing tasks delivered by the mobile devices (MDs), the MDs move outside the coverage of the ES, where task migration is required to ensure service continuity. Most current research on task migration ignores inter-task dependencies and uncertain computing environments, and it focuses mainly on migration scenarios where MDs have a one-to-one or many-to-one relationship with ESs. Aiming at the problem of workflow migration with multi-MDs and multi-ESs in uncertain environments, this paper proposes an interval many-objective optimized workflow migration in uncertain environments (I- MaOWMUE) model that considers transforming uncertainty factors into interval parameters for processing, along with the migration delay, maximum completion time, energy consumption, and load balancing as an objective function, and at the same time, utilize real-time priority scheduling strategies to achieve the fast response of the tasks. Considering the dependency of tasks and the changing characteristics of ES load in a migration environment, this paper designs a migration-based interval many-objective evolutionary algorithm (MI-MaOEA), which adopts an interval confidence strategy to improve algorithm convergence and formulates an objective- value-dominated hierarchical sorting and dual-migration selection strategy based on the migration delay and the success rate of the migration to improve the diversity of the populations. Simulation results show that MI- MaOEA optimizes 27%, 35%, 14%, and 80% in solving the four objective values of I-MaOWMUE, and enables the solution to have faster converse speed and better distribution.

# Introduction

Wireless network communications and intelligent sensory process- ing technologies are driving the swift development of the Internet of Things (IoT) [[1](#_bookmark26)], and IoT-based fine-processing applications, such as video data analytics and augmented/virtual reality (AR/VR), need to deal with the exploding number of tasks in a real-time manner [[2](#_bookmark27)]. However, MD’s limited hardware resources and battery life make it unable to meet users’ real-time requirements and provide them with overloaded services. EC as a new network computing paradigm pushes computing resources to the edge of the network [[3](#_bookmark28)], and resource-poor MDs can oﬄoad tasks to ES to relieve the computational pressure. In most EC scenes, it is usually assumed that the MD is at rest, when in fact the positional state changes of the MD are not to be ignored. Considering the limited service scope of ES, when the MD exceeds the coverage of ES, service interruption may occur, and to ensure the con- tinuity of service, the unexecuted tasks need to be migrated to other

available ESs within the coverage [[4](#_bookmark29)]. Existing task migration scenar- ios are mainly aimed at migrating tasks from single or multiple MDs to a single ES [[5](#_bookmark30),[6](#_bookmark31)]. However, the computational resources of a single ES are still limited and cannot satisfy the computational requests of a large number of users, and collaborative computing (CC) can be adapted to unite multiple ESs to provide migration services [[7](#_bookmark32)]. In multi-MDs and multi-ESs migration scenarios, it is necessary to consider the switch- ing of network connections, the control of migration latency, and the state information of server clusters in the new region, etc., and formu- late a reasonable migration strategy for the originally unexecuted tasks to improve the user quality of experience (QoE).

In current computing migration scenarios, it is often overlooked that there are many uncertainties present, which include uncertain compu- tational and network environments [[8](#_bookmark33)], all of which can have an impact on the processing eﬃciency of a task. However, in real applications, net- work congestion, excessive load leading to task execution failure, and

\* Corresponding author.

*E-mail addresses:* [shizy19990102@163.com](mailto:shizy19990102@163.com) (Z. Shi), [zhaotianhao1015@163.com](mailto:zhaotianhao1015@163.com) (T. Zhao), [liqi2575564568@163.com](mailto:liqi2575564568@163.com) (Q. Li), [15634969919@163.com](mailto:15634969919@163.com) (Z. Zhang), [cuizhihua@gmail.com](mailto:cuizhihua@gmail.com) (Z. Cui).

<https://doi.org/10.1016/j.eij.2023.100418>

Received 20 August 2023; Received in revised form 13 October 2023; Accepted 5 November 2023

Available online 21 November 2023

1110-8665/© 2023 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-nc-nd/4.0/).](http://creativecommons.org/licenses/by-nc-nd/4.0/))

other problems are unavoidable. In particular, when migrating multiple tasks at the same time, the increasing migration link resource con- tention and the unavoidable serial interference caused between links. Therefore, the uncertainty factor in the computational migration en- vironment has an important impact on the eﬃciency and stability of task processing. In the vast majority of computing migration scenarios, scholars consider independent task migration, which somewhat weak- ens the dependency relationship and execution order between tasks. However, in practical application scenarios, tasks do not exist in isola- tion, and there are often mutual constraint relationships between tasks [[9](#_bookmark34)], such as medical diagnosis and treatment, logistics, and supply chain management. Therefore, we apply workflow to a computational migra- tion scenario to achieve task passing and parallel processing between different Ess [[10](#_bookmark35)], which implies that task constraint relationships need to be considered comprehensively when making task migration deci- sions.

A lot of research has been done on task computation and resource allocation in EC today, aiming to formulate the computational problem as an optimization model to minimize objectives such as total delay, to- tal energy consumption, and total cost [[11](#_bookmark37),[12](#_bookmark38)]. However, most of them consider one optimization objective, explore the optimal solution in a one-dimensional decision space [[13](#_bookmark39)], or consider multiple objectives but weigh them to reduce the dimensionality, but this way of weight- ing with subjective preferences may lead to errors in the results [[14](#_bookmark40)]. In the actual task computation process, in the face of different user re- quirements, it is necessary to comprehensively consider the trade-offs between multiple objectives [[15](#_bookmark41)].

Multi-MDs and multi-ESs workflow migration problems in uncertain

environments face two key challenges. On the one hand, migrating tasks from the original ES cluster to the target ES cluster requires comprehen- sive consideration of the dependencies and requirements of the tasks, the ES cluster resource status, energy consumption, and other factors, as well as uncertainties such as network congestion and ES resource availability, etc. Through real-time detection and analysis, in complex migration scenarios, a rational migration strategy is developed to op- timize multiple objectives so that service requests from multiple MDs can be executed in parallel and eﬃciently and to ensure effective re- source utilization. On the other hand, each ES needs to handle tasks uploaded by users in the region and tasks migrated from neighboring regions due to user location changes. Although ESs release a certain amount of storage space through computation, unreasonable migration strategies may lead to the overloading of some ESs and even excessive task lag time. For ES, due to untimely processing, there may be exces- sive loads resulting in data loss or even system crashes. To ensure that tasks are processed promptly and reduce additional delay expenses, it is necessary to comprehensively consider the dependence of the task, load changes, and storage limitations, according to the emergency degree of the task to formulate a real-time priority scheduling strategy, which is necessary to determine the priority of the task execution relationship, thereby improving user satisfaction, so that the EC system is eﬃciently executed.

At present, interval many-objective evolutionary algorithms (I- MaOEAs) can effectively solve uncertain many-objective optimization problems (U-MaOPs). In this paper, the above problem is modeled as an I-MaOWMUE model, and MI-MaOEA is proposed for this model, the algorithm adopts the interval confidence strategy to solve the un- certainty distress and optimize the interval many-objective, and at the same time, it weighs the consideration of ES load situation and inter- task dependency in the migration environment, and takes the migration delay and the success rate of the migration as the basis of the selec- tion to formulate the objective-value-dominated hierarchical sorting and dual-migration selection strategies, so that these selected vectors are uniformly distributed on the Pareto Frontier (PF), thus providing the decision makers with appropriate migration strategies. The main contributions of this paper are as follows:

* In this paper, we discuss the multi-MDs and multi-ESs workflow mi- gration problem in uncertain environments and propose the MaOW- MUE model, where we consider uncertainties such as network band- width and server computing power to be transformed into interval parameters, and migration delay, maximum completion time, en- ergy consumption, and load balancing as optimization objectives.
* To achieve a rational allocation of resources and rapid response to tasks, a real-time priority scheduling strategy is proposed, which uses information from real-time surveillance based on task depen- dencies and characteristics to assign different task priorities and degrees of urgency.
* To solve the I-MaOWMUE model, this paper designs MI-MaOEA, which adopts an interval confidence strategy to improve the con- vergence of the algorithm and proposes objective-value-dominated hierarchical sorting and dual-migration selection strategies to in- crease the diversity of the population.
* Simulation results show that MI-MaOEA optimizes the objective of the I-MaOWMUE model much better compared to other algorithms, and the solution set has a significant advantage over other algo- rithms in terms of mean, maximum, and minimum values.

The rest of the paper is organized as follows. In section [2](#_bookmark1), related work is presented. In section [3](#_bookmark2), the I-MaOWMUE model is constructed. In section [4](#_bookmark10), MI-MaOEA is proposed to solve the I-MaOWMUE model. In section [5](#_bookmark19), simulation experiments are performed and the results are summarised and analyzed. In section [6](#_bookmark36), the paper is summarised and conclusions are drawn.

# Related work

MD transfers the generated tasks to ES execution by means of computational oﬄoading, exploiting terminal-edge collaboration to im- prove QoE. Zakaryia et al. [[16](#_bookmark42)] consider oﬄoading tasks from mobile devices to cloudlets for execution and achieving eﬃcient oﬄoading of tasks through the strategy of the queuing networks and an evolutionary algorithm. Liu et al. [[17](#_bookmark43)] acquire energy through hybrid access points (HAPs) and choose to execute the task locally or oﬄoad it to a single fog/cloud server for execution. They propose the Generalized Bending Decomposition (GBD) method to maximize the minimum energy bal- ance among users. However, the limited resources of a single server are not enough to meet the real-time demands of MD. To this end, Do- Duy et al. [[18](#_bookmark44)] considered multiple MDs oﬄoading tasks to multiple ESs to reduce latency in the presence of limited computational and ser- vice resources. Ding et al. [[19](#_bookmark45)] proposed two computing architectures, hierarchical end-edge-cloud computing (Hi-EECC) and horizontal end- edge-cloud computing (Ho-EECC), and proposed two potential game algorithms based on this architecture. The above computational ap- proaches consider terminal-edge and edge-edge collaboration for com- putational tasks while developing suitable computational strategies to improve the overall effectiveness of the system. However, these studies consider the state of the MD to be stationary and do not take into ac- count the effect of changes in the user’s location on the computational decision.

In fact, the MD may change its position during EC computation, and

to ensure service continuity, the unexecuted tasks need to be migrated to realize the edge-edge collaborative computation. Kim et al. [[20](#_bookmark46)] of- fload the user tasks to a nearby ES, and as the user moves, the tasks are migrated, and heuristics are proposed to solve it to reduce the computa- tional cost and service latency. Similarly, in [[21](#_bookmark47)], vehicles carrying 6g network in boxes (NIBs) can communicate with other NIBs in real time to reduce the energy consumption and cost incurred during the service migration process through the NIB task migration method (NTM) and to develop a suitable migration strategy through the strength Pareto evolutionary algorithm (SPEA2). However, the above studies ignore the impact of environmental changes on the computational results during the task computation process, and at the same time, they focus on in-

dividual tasks from time to time in computational migration scenarios while ignoring the characteristics of inter-task constraint relationships. In real EC systems, uncertain computing and network environments will affect the processing eﬃciency of tasks. In [[22](#_bookmark48)], a constraint mech- anism is proposed to cope with the uncertainty in the processing cycle of the task and minimize the energy consumption while satisfying the deadlines and designing an online selection scheme to solve the prob- lem. In [[23](#_bookmark49)], the channel and ES statistical characteristics are constantly changing, and by sequentially selecting ESs and using historical time and energy consumption to make new oﬄoading decisions. Xu et al.

[[24](#_bookmark50)] address uncertainty issues such as resource competition and link outages in the Internet of Vehicles (IoV), for which a software-defined network-based service management framework for IoV is proposed. Therefore, considering the uncertainty factor in the migration environ- ment, it is more in line with practical computing scenarios.

In practical applications, tasks do not exist independently and need to be endowed with task states, attributes, and user requirements [[25](#_bookmark51),[26](#_bookmark52)]. In [[27](#_bookmark53)], the focus is on the division of tasks for different ap- plication types and the development of suitable oﬄoading strategies for joint optimization. He et al. [[28](#_bookmark54)] designed a hybrid task oﬄoad- ing problem with hard and soft deadlines and proposed the CONFECT oﬄoading method to handle it. Sun et al. [[29](#_bookmark55)] considered the lim- ited capacity of MD and the dependencies between tasks, proposed a series of task allocation strategies for different types of tasks in com- plex network environments, and found feasible solutions that satisfied the constraints. Huang et al. [[30](#_bookmark56)] formulated risk-constrained workflow scheduling as a Markov Decision Process (MDP) and designed a rein- forcement learning-based security-aware workflow scheduling (SAWS) scheme. The study of task flows with constraint relationships is of prac- tical interest by considering the states of the tasks and the strong con- nections between them. However, the above studies have not developed reasonable computation strategies and resource allocation strategies for the characteristics of dependency tasks to improve the overall system eﬃciency.

Nowadays, multi-objective evolutionary algorithms can effectively

solve multi-objective optimization problems [[31](#_bookmark57)–[35](#_bookmark62)]. However, for many-objective optimization problems, the scale of the problem in- creases as the number of objectives increases, and the complexity of searching for and evaluating the solution increases, which requires an appropriate many-objective evolutionary algorithm to solve the prob- lem [[36](#_bookmark64)–[38](#_bookmark66)]. Cui et al. [[39](#_bookmark55)] propose a many-objective evolutionary algorithm based on three-way decision (MaOEA-TWD) to solve the problem of convergence and diversity conflict as the number of objec- tives increases. Bozorgchenani et al. [[40](#_bookmark58)] consider that in mobile edge computing (MEC) and fog computing (FC), the task is oﬄoaded from the client to the ES or other clients to minimize latency and energy con- sumption. For this purpose, the problem is modeled as a constrained multi-objective optimization problem (CMOP), and an evolutionary al-

gorithm is designed to solve it. However, uncertain optimization prob-

algorithm for knee joint decision-making for a multi-objective optimiza- tion problem with concurrent risk-benefit. Therefore, I-MaOEAs can be used to solve U-MaOPs.

In this paper, we consider a computational migration scenario where the user location changes, in which multiple MDs are located in the overlapping region of multiple BSs. When there is a user lo- cation change, the unexecuted tasks need to be migrated to multiple servers within the coverage area based on the system state. In addi- tion, this paper considers the dependent task computation problem in uncertain environments, such as network bandwidth and server com- puting power, and models it as the I-MaOWMUE model to optimize the four objectives of migration delay, maximum completion time, energy consumption, and load balancing. Meanwhile, a real-time pri- ority scheduling strategy is formulated for workflow characteristics to achieve reasonable resource allocation and fast task response. To solve the I-MaOWMUE model, MI-MaOEA is designed in this paper, which solves the uncertainty-troubling problem and ensures the convergence and diversity of the solution.

# The proposed I-MaOWMUE model

* 1. *System model*

As shown in Fig. [1](#_bookmark4), we consider a multi-area EC system consisting of multiple base stations (BAs) and multiple MDs (e.g., computers, mobile

phones, tablets, etc.). We denote by *𝐵𝑀* = {1*,* 2*, ..., 𝑚, ..., 𝑀* } the set of

base stations and *𝑀𝑈* = {1*,* 2*, ..., 𝑢, ..., 𝑈* } the set of users. Multiple BAs

and multiple MDs are randomly distributed within each area, and each

mation, which is denoted by *𝑆𝐼*, i.e., *𝑆𝐼* = (*𝑐𝑝, 𝑠𝑠, 𝑐𝑜*), *𝑐𝑝*, *𝑠𝑠*, and *𝑐𝑜* base station is equipped with an ES. Each ES has different state infor-

denote the computing power, storage space, and coverage of the ES, re-

to the ES cluster within the *𝑐𝑜* to provide computing services to it. Dif- spectively. Each MD generates multiple task requests and oﬄoads tasks

ferent link connections (e.g., wifi and 5G) are used for communication between users and base stations (MD-BA) and between base stations and base stations (BA-BA).

In this paper, we focus on EC scenarios for workflow migration in uncertain environments, and we will introduce the workflow model in Section [3.2](#_bookmark3), the migration model in Section [3.3](#_bookmark5), and the optimization objective in Section [3.4](#_bookmark7).

* 1. *Workflow model*

ate *𝐾*(*𝐾* ≥ 0) heterogeneous workflows *𝑊* = {*𝑤*1*, 𝑤*2*, ..., 𝑤𝑘, ..., 𝑤𝐾* }. The Multiple MDs within the coverage area of each ES locally gener-

heterogeneity of the workflows is mainly reflected in the different de-

usually *𝑤𝑘* consists of a set of tasks and dependencies between them, pendencies and number of tasks of the workflows. As shown in Fig. [2](#_bookmark6), which we represent by a directed acyclic graph (DAG). For each *𝑤𝑘* , it

consists of the quintuple *𝐷* = (*𝑇 𝑐, 𝑃𝑟𝑒𝑘, 𝑆𝑢𝑘, 𝑁𝑘, 𝑃𝑆𝑘*), where *𝑇 𝑐* denotes

*𝑘 𝑘*

lems often place higher demands on the solution capabilities of tradi- tional many-objective evolutionary algorithms, which makes it diﬃcult to obtain satisfactory solutions.

There is a growing tendency among scholars to adopt methods such as random, fuzzy, and interval variables to deal with multi-objective optimization problems that are fraught with uncertainty [[41](#_bookmark59),[42](#_bookmark60)]. In practical scenarios, determining the exact probability distribution of a random variable or the exact aﬃliation function of a fuzzy number is often challenging, whereas upper and lower bounds, or midpoints and radii of interval parameters, are relatively easier to obtain and provide more reliable information. Because of this, the use of interval-based ap- proaches in uncertain optimization problems has become highly sought- after due to their ability to better solve complex problems in practice. To solve the interval multi-objective optimization problem, Jin et al. [[43](#_bookmark61)]

the completion time of the terminal node of the workflow *𝑤𝑘* , which represents the completion of the whole workflow, *𝑃𝑟𝑒𝑘* denotes the set of predecessor nodes of the workflow *𝑤𝑘* , *𝑆𝑢𝑘* denotes the set of suc- cessor nodes of the workflow *𝑤𝑘* , *𝑁𝑘* denotes the number of tasks of the workflow *𝑤𝑘* , and *𝑃𝑆𝑘* denotes the strong constraint relationship between the two sets *𝑃𝑟𝑒𝑘* and *𝑆𝑢𝑘* . For each workflow, there are of-

ten many-to-many dependencies between tasks. There may exist one or

more predecessor nodes for each child node except the start node, and one or more successor nodes for each child node except the terminal

node. For each subtask in the workflow *𝑤𝑘* , it consists of the quintu-

ple *𝑊𝑇* = (*𝑤𝑘,𝑛, 𝑢𝑠𝑘,𝑛, 𝑎𝑐𝑡𝑘,𝑛, 𝜒𝑚 , 𝜔𝑘,𝑛*), where *𝑤𝑘,𝑛* denotes the nth task of the workflow *𝑤𝑘* , *𝑢𝑠𝑘,𝑛* denotes the upload size of the task *𝑤𝑘,𝑛* , *𝑎𝑐𝑡𝑘,𝑛* denotes the activation time of the task *𝑤𝑘,𝑛* , except for the start node,

*𝑘,𝑛*

and the task is activated when and only when the execution of all the

*𝑘,𝑛*

proposed a decomposition-based interval multi-objective evolutionary

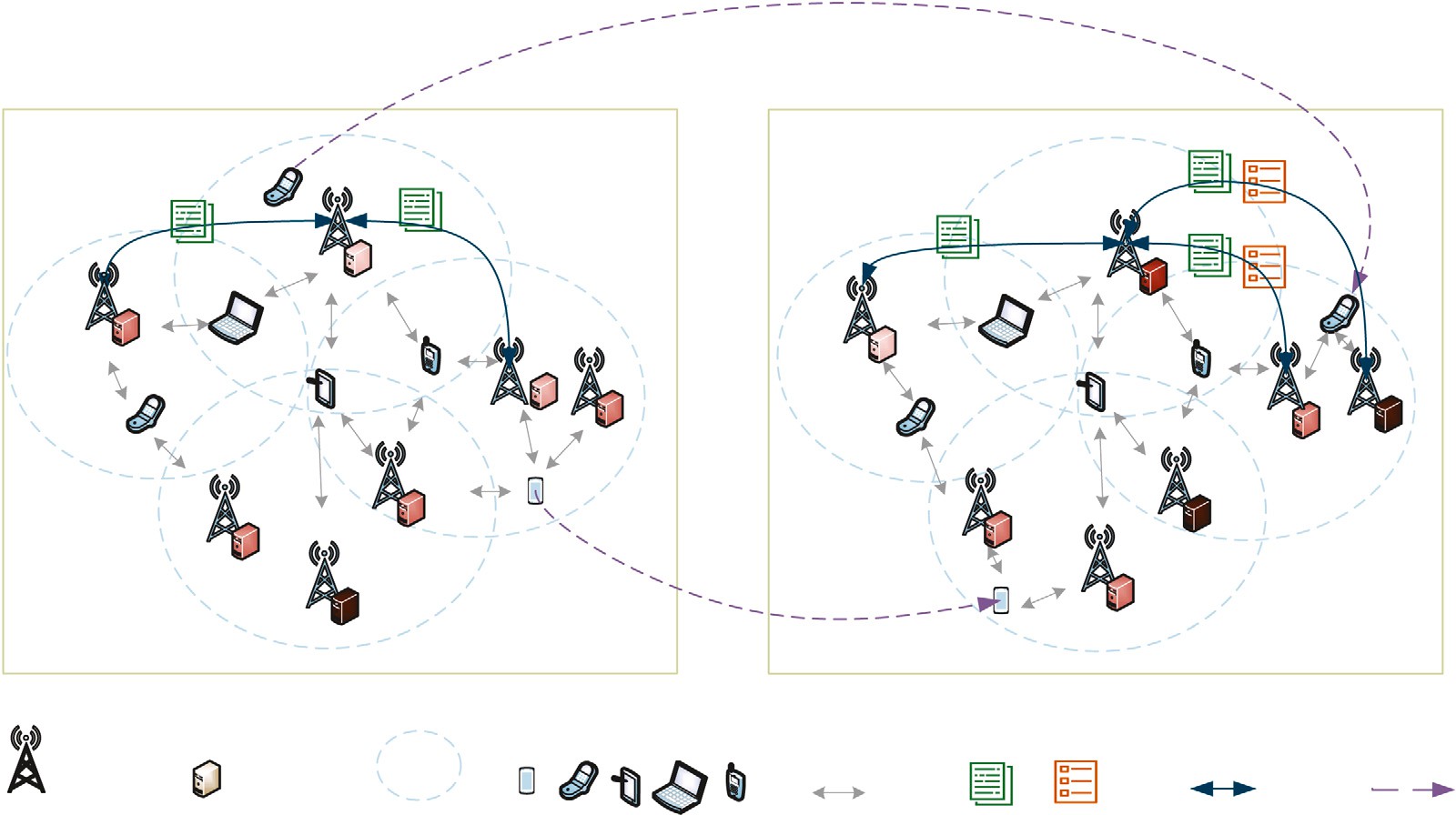
predecessor nodes is completed, *𝜒𝑚*

denotes the priority level of the

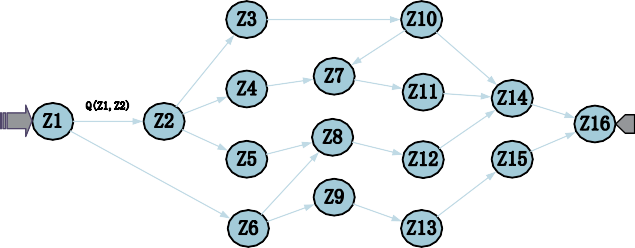
algorithm with adaptive adjustment of weight vectors and neighbor- hoods. He et al. [[44](#_bookmark63)] proposed a multi-objective interval evolutionary

task *𝑤𝑘,𝑛* in the ES *𝑚*, and *𝜔𝑘,𝑛* denotes the degree of urgency of the task

*𝑤𝑘,𝑛* .



**Fig. 1.** Migration model framework.



**Fig. 2.** An example of a workflow.

* 1. *Migration model*

As shown in Fig. [1](#_bookmark4), the location state of the MD may change during the execution of the oﬄoading task by the ES, and when its location is

not in the *𝑐𝑜* of the ES, the ES that originally served it cannot continue

server *𝑐𝑝*, a part of the tasks are not processed promptly during the to serve it, and at the same time, considering the limitations of the

oﬄoading phase. To ensure the continuity of the service and to improve the QoE, the sub-tasks of each workflow that were not executed and partially executed but not completed by the original ES were migrated to different ESs using cross-area edge-edge collaborative computation.

However, each ES is unable to support strong computational demand

is performed. *𝑎𝑐𝑡𝑘,𝑛* will be described in detail in Section [3.4](#_bookmark7) Maximum and an orderly sorting of the execution order of tasks in the server completion time. For each server *𝑚*, the specific steps of the real-time

priority scheduling strategy are as follows:

placed in the set *𝑆𝐿* . Some tasks are migrated, and the tasks that are (1) Pre-processing: tasks that have not been executed by ES are migrated away need to be deleted from *𝑆𝐿* and the migrated over tasks put into *𝑆𝐿* ;

1. Assigning *𝜒𝑚* and ordering:
   1. If there are tasks without predecessors in *𝑆𝐿* , assign the highest

*𝑘,𝑛*

priority level *𝜒𝑚* =*𝜒𝑚*, and put it into the set *𝑅𝐿*. For tasks that match

*𝑘,𝑛* 1

with limited *𝑐𝑝* due to receiving tasks oﬄoaded within the *𝑐𝑜* as well

*𝜒𝑚*, compare the migration delay *𝑇 𝑚𝑖𝑔* as *𝜔𝑘,𝑛* of these tasks, with the

1 *𝑘,𝑛*

as tasks migrated from multiple areas, which may lead to overloading. A migration strategy can be utilized to select appropriate servers for migrating tasks that have not finished executing based on the current system state information, and the ES can also utilize real-time priority scheduling policies to allocate resources to tasks based on the degree of urgency of the task to meet the four objectives of migration delay, maximum completion time, energy consumption, and load balancing.

* + 1. *Queuing model*

As shown in Fig. [3](#_bookmark8), to ensure that tasks in ES can be processed

scheduling strategy based on the priority level *𝜒𝑚* and degree of ur- quickly, this paper formulates a corresponding real-time priority gency *𝜔𝑘,𝑛* of the tasks, with the higher *𝜒𝑚* and the higher *𝜔𝑘,𝑛* being executed first. In the migration phase, except for the start node, *𝑎𝑐𝑡𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

is redefined as the maximum value of the execution completion time of all the predecessor nodes and the time of task migration to ES,

smaller *𝜔𝑘,𝑛* is, the higher the order of execution, *𝜔𝑘,𝑛* will be introduced

in section [3.3.2](#_bookmark9);

* 1. If the server has a task in progress, assign that task a priority level

*𝑚* m

*𝜒* =*𝜒* ;

*𝑘,𝑛* 2

into the set *𝑅𝐿* , assign these tasks a priority level *𝜒𝑚* =*𝜒𝑚*, compare 3) For tasks that have been partially executed and migrated and put

*𝑘,𝑛* 3

*𝑎𝑐𝑡𝑘,𝑛* as their *𝜔𝑘,𝑛* , with the smaller *𝑎𝑐𝑡𝑘,𝑛* is, the higher the order of

execution;

level *𝜒𝑚* =*𝜒𝑚*. Real-time monitoring of the server, if the server tem- 4) For other tasks waiting to be executed, assign the lowest priority

*𝑘,𝑛* 4

whether there is a task in the set *𝑆𝐿* at this time has been activated, porarily has no task or a task and task execution is completed, check if it exists, then put into the set *𝑅𝐿* , and according to *𝑎𝑐𝑡𝑘,𝑛* to sort, the smaller *𝑎𝑐𝑡𝑘,𝑛* , the higher the degree of urgency *𝜔𝑘,𝑛* , the higher the order of execution. Therefore, *𝜔𝑘,𝑛* is calculated as follows:

*⎧**⎪* 1*𝑚𝑖𝑔 , 𝜒𝑚* = *𝜒 𝑚*

1+*𝑇*

*𝑘,𝑛*

1

*⎪⎨*

**Fig. 3.** Real-time priority scheduling model.

does not consider the effect of uncertainties in the oﬄoading phase on



*𝜔𝑘,𝑛* = 0*, 𝜒𝑚* = *𝜒 𝑚*

*𝑘,𝑛*

2

1 *, 𝜒𝑚* = *𝜒 𝑚* or *𝜒𝑚* = *𝜒 𝑚*

*⎪*

*𝑘,𝑛*

(1)

the migration phase, the transmission and computation process before

migration is considered as a deterministic problem in this paper. The

completion time *𝑇 𝑐* of task *𝑤* can be redefined as:

1+act*𝑘,𝑛*

*⎩*

*𝑘,𝑛* 3

*𝑘,𝑛* 4

*𝑘,𝑛*

∨ ∧

*𝑇* = [*𝑇 , 𝑇* ]= *𝑇 𝑚𝑖𝑔* + *𝑇*

*𝑘,𝑛*

+*𝑇*

+*𝑇*

(3) Repeat step (2) until the task execution in server *𝑚* is complete.

* + 1. *Migration cycle model*

*𝑐*

*𝑘,𝑛*

*𝑐*

*𝑘,𝑛*

∨

*𝑘,𝑛*

*𝑐*

*𝑘,𝑛*

*𝑘,𝑛*

∧

and *𝑇*

*𝑤𝑎𝑖𝑡*

*𝑘,𝑛*

*𝑒𝑥𝑒*

*𝑘,𝑛*

*𝑑𝑡*

*𝑘,𝑛*

The life cycle of each *𝑤𝑘,𝑛* in the migration process contains 3 or

where *𝑇 𝑐*

*𝑐*

*𝑘,𝑛*

denote the lower and upper bounds on the comple-

4 phases, denoted as *𝑇* = {*𝑇*1*, 𝑇*2*, 𝑇*3*, 𝑇*4} which denotes the migration

transfer phase, respectively, where *𝑇*1 is optional, and this phase is ig- phase, the service waiting phase, the execution phase, and the data

nored when the task is not migrating. If the server is in the process of processing user-delivered tasks and the user location has exceeded the

*𝑐𝑜* of the server, then the unexecuted and partially implemented but not

completed tasks need to be migrated and the time spent is the migra-

tion delay *𝑇 𝑚𝑖𝑔* , *𝑇*2 denotes the waiting delay *𝑇 𝑤𝑎𝑖𝑡* of task *𝑤𝑘,𝑛* at the

tion time of task *𝑤𝑘,𝑛* , respectively.

* 1. *Objective function*

1. Migration delay

The MD position changes during the process of calculating the up- loaded tasks of local users by the ES, and when the MD position exceeds the co of the ES and the uploaded tasks are not processed in time, it is

*𝑘,𝑛 𝑘,𝑛*

server, *𝑇*3 denotes the execution delay *𝑇 𝑒𝑥𝑒* of task *𝑤𝑘,𝑛* at the server,

*𝑘,𝑛*

necessary to migrate the task *𝑤𝑘,𝑛* and mark it with *𝜙𝑘,𝑛* = 1, or *𝜙𝑘,𝑛* =0

otherwise. *𝑇 𝑚𝑖𝑔* for task *𝑤* is calculated as follows:

and *𝑇*4 denotes the data transfer delay *𝑇 𝑑𝑡* used by task *𝑤𝑘,𝑛* to transfer

*𝑘,𝑛*

the generated data to the successor node after its execution. The com-

*𝑘,𝑛*

∨ ∧

*𝑘,𝑛*

*𝜙* ⋅ *𝑇𝑆*

⋅ *𝐷𝑚𝑚*′

pletion time *𝑇 𝑐*

for each subtask *𝑤𝑘,𝑛* for a complete workflow *𝑤𝑘* is

*𝑚𝑖𝑔*

*𝑚𝑖𝑔*

*𝑚𝑖𝑔*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

denoted as:

*𝑘,𝑛*

*𝑇𝑘,𝑛* = [*𝑇𝑘,𝑛 , 𝑇𝑘,𝑛* ]=

*𝑚𝑖𝑔*

*𝑘,𝑛*

*𝑆*

(4)

*𝑇 𝑐*

= *𝑇 𝑚𝑖𝑔* +*𝑇 𝑤𝑎𝑖𝑡* + *𝑇 𝑒𝑥𝑒*+*𝑇 𝑑𝑡*

(2)

∨ ∧

*𝑝* ⋅ *𝑔*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑆𝑚𝑖𝑔* = [*𝑆𝑚𝑖𝑔, 𝑆𝑚𝑖𝑔*]= *𝐵𝑚𝑖𝑔* ⋅ log2(1 +

*𝑘,𝑛*

*𝑘,𝑛*

) (5)

*3.3.3. Uncertainty model*

Uncertain communication link environments and unstable smart de- vice execution capabilities will certainly affect the eﬃciency of task

*𝑘,𝑛*

*𝑇𝑆𝑘,𝑛*

*𝑘,𝑛 𝑘,𝑛*

*𝑢𝑠𝑘,𝑛, 𝜆𝑘,𝑛* =0

= *{*

*𝑚𝑠𝑘,𝑛, 𝜆𝑘,𝑛* =1

*𝛿*2 + *𝑝𝑘,𝑛* ⋅ *𝐼*

(6)

execution. Communications noise, communications interference, com- munications strength, etc. are the uncertainty factors affecting the transmission of task communication links when the intelligent device requests a relatively large number of tasks, the channel link resource contention and serial interference between the links lead to poor com- munication status, on the contrary, the communication link is more stable when there are fewer tasks grabbing resources. Bursty computa- tional requests often lead to an insuﬃcient supply of ES resources, re- sulting in ES corruption directly affecting the computational capability of the ES, which may lead to slower or even inoperable task processing. Therefore, we model it in the form of an interval to represent the un-

Since task *𝑤𝑘,𝑛* cannot complete its computation in the server where it certainty factors that affect the channel state and computational power.

was originally oﬄoaded, it needs to be migrated, and the bandwidth of

*𝜙𝑘,𝑛* ∈ {0*,* 1}*,* ∀*𝑤𝑘,𝑛* ∈ *𝑤𝑘*

*𝜆𝑘,𝑛* ∈ {0*,* 1}*,* ∀*𝑤𝑘,𝑛* ∈ *𝑤𝑘*

′where *𝑇𝑆* denotes the size of task *𝑤* in the migration phase; *𝐷𝑚𝑚*

*𝑘,𝑛*

*𝑘,𝑛 𝑘,𝑛*

denotes the link migration distance between servers *𝑚* and *𝑚*′; *𝑆𝑚𝑖𝑔* de-

*𝑘,𝑛*

and after the migration; *𝑝𝑘,𝑛* denotes the upload power of task *𝑤𝑘,𝑛* ; *𝑔𝑘,𝑛* notes the migration speed between the servers serving the task before denotes the communications gain of task *𝑤𝑘,𝑛* ; *𝛿* denotes the communi- cations noise; *𝐼* denotes the communications interference; *𝜆𝑘,𝑛* denotes

not completed, yes then *𝜆𝑘,𝑛* = 1, otherwise *𝜆𝑘,𝑛* = 0; and *𝑚𝑠𝑘,𝑛* denotes whether the task was partially executed before the migration but was task size of the remaining unexecuted portion of task *𝑤𝑘,𝑛* tagged by the

server m before the migration starts, which is calculated as follows:

∨ ∧ *𝑟*

the migrated link can be denoted as *𝐵𝑚𝑖𝑔* =[ *𝐵𝑚𝑖𝑔, 𝐵𝑚𝑖𝑔* ] during the mi-

gration process. The data transfer bandwidth of the predecessor node

to transfer the data to the server where the successor node is located

*𝑚𝑠𝑘,𝑛*=*𝑢𝑠𝑘,𝑛* ⋅ (1 −

*𝜏* − *𝑇𝑘,𝑛*

*𝑒𝑥𝑒*

*𝑇*

*𝑘,𝑛*

) (7)

∨ ∧

where *𝜏* denotes the time interval of the oﬄoading phase, and *𝑇 𝑟*

de-

after calculating the task is denoted as *𝐵𝑑𝑡*=[ *𝐵𝑑𝑡, 𝐵𝑑𝑡* ]. The number of

notes the response time of task *𝑤𝑘,𝑛* .

*𝑘,𝑛*

CPU cycles per second that can be computed by server *𝑚* is denoted

Objective *𝑓*1 is to minimize the migration delay for all migration

∨ ∧

as *𝑅𝑚*=[ *𝑅𝑚, 𝑅𝑚* ]. Since this paper focuses on the migration phase and

tasks:

∨ ∧ *{∑𝑉*

*∑𝐴 }*

∨ ∧ *{*

*𝑚𝑖𝑔*

*{ 𝑒𝑐𝑡,𝑗*

*𝑗 }}*

1 1 1

min *𝑓* =*𝑚𝑖𝑛*[ *𝑓 , 𝑓* ]= min

*𝑣*=1 *𝑎*=1

*𝑣,𝑎*

*𝑗*∈*𝑃𝑟𝑒𝑘,𝑛*

(13)

1. Maximum completion time

*𝑇 𝑚𝑖𝑔*

(8)

After the task *𝑤𝑘,𝑛* is delivered to the ES, it is sorted according to

*𝑤𝑠𝑘,𝑛* ∈ {0*,* 1}*,* ∀*𝑤𝑘,𝑛* ∈ *𝑤𝑘*

the real-time priority scheduling strategy, and the task starts to execute

when the task execution condition is satisfied, and the execution delay

where *𝑤𝑠*

*𝑘,𝑛*

denotes whether task *𝑤𝑘,𝑛*

is the start node, if yes then

*𝑇 𝑒𝑥𝑒* of task *𝑤*

is denoted as:

*𝑤𝑠𝑘,𝑛* = 0, otherwise *𝑤𝑠𝑘,𝑛* = 1; *𝑃𝑟𝑒𝑘,𝑛* denotes the set of predecessor tasks

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

of task *𝑤𝑘,𝑛* ; *𝑇 𝑒𝑐𝑡,𝑗* denotes the execution completion time of the *𝑗*th

*𝑇𝑆* ⋅cc*𝑚*

*⎧⎪ 𝑘,𝑛*

*𝑘,𝑛 , 𝜙𝑘,𝑛* =0

∨

∧

*𝑅𝑚*

predecessor task of task *𝑤𝑘,𝑛*

; and *𝐷𝑇 𝑗*

denotes the data transfer time

*𝑇 𝑒𝑥𝑒* = [*𝑇 𝑒𝑥𝑒, 𝑇 𝑒𝑥𝑒*]= *⎪⎨*

*𝑘,𝑛*

*𝑚*

*𝑘,𝑛*

′

(9)

*𝑘,𝑛*

*𝑘,𝑛*

*⎪*

∧

*𝑘,𝑛*

*𝑚,*max

*𝑘,𝑛*

*𝑇𝑆𝑘,𝑛* ⋅cc*𝑘,𝑛*

*𝑅𝑚*′

*⎩ 𝑘,𝑛*

*, 𝜙𝑘,𝑛* =1

Due to the limited *𝑐𝑜* of the ES, when there is a large number of tasks

in demand, it may not be able to serve them in time, the task arrives at

of the *𝑗*th predecessor task to transfer data to task *𝑤*

.

the server even if it is activated may still have to wait in the server, the

waiting time *𝑇 𝑤𝑎𝑖𝑡* for task *𝑤𝑘,𝑛* can be calculated as:

*𝑚*

*𝑅*

*𝑘,𝑛*

∧

≤ *𝑅*

*𝑘,𝑛*

*,* ∀*𝑤𝑘,𝑛* ∈ *𝑤𝑘*

*𝑘,𝑛*

∨ ∧

*𝑚*′

*𝑚*′*,*max

*𝑇 𝑤𝑎𝑖𝑡* = [*𝑇 𝑤𝑎𝑖𝑡, 𝑇 𝑤𝑎𝑖𝑡*]= *𝑇 𝑟*

– *𝑎𝑐𝑡𝑘,𝑛* (14)

*𝑅𝑘,𝑛* ≤ *𝑅*

*𝑘,𝑛*

*,* ∀*𝑤𝑘,𝑛* ∈ *𝑤𝑘*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

where cc*𝑚*

*𝑘,𝑛*

denotes the CPU cycles required by server *𝑚* to compute

Therefore, for workflow *𝑤𝑘* the maximum completion time is:

task *𝑤*

; *𝑅𝑚*

denotes the number of CPU cycles per second that can be *𝑐* ∨ ∧ *𝑐*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑇* = [*𝑇 𝑐, 𝑇 𝑐*]= max *𝑇*

(15)

computed by the server *𝑚* on which task *𝑤* resides; and *𝑅𝑚*′ denotes

*𝑘,𝑛 𝑘,𝑛*

*𝑘 𝑘 𝑘*

*𝑜*∈*𝑤𝑘*

*𝑘,𝑜*

server *𝑚*′ on which task *𝑤𝑘,𝑛* resides. the number of CPU cycles per second that can be computed by the

Objective *𝑓*2 is to minimize the average of the maximum completion

times of all workflows *𝑊* :

2 *𝑓*2 *𝑓*2 *𝐾*

In the workflow *𝑤𝑘* , for each task *𝑤𝑘,𝑛* with a successor node, the resulting data is transferred to each subtask of its successor set *𝑆𝑢𝑘,𝑛*

after the task execution, *𝑇 𝑑𝑡* denoted by:

*𝑘,𝑛*

min *𝑓* =*𝑚𝑖𝑛*[ ∨ *,* ∧ ] =*𝑚𝑖𝑛 {* 1 ⋅

*𝐾*

*𝑐*

*𝑇*

*∑ }*

*𝑘*

*𝑘*=1

(16)

∨ ∧

*𝑆𝑢𝑘,𝑛*

1. Energy consumption

In be sustainable and use resources wisely, energy consumption is

*𝑇 𝑑𝑡* =[ *𝑇 𝑑𝑡 , 𝑇 𝑑𝑡* ]= *𝜌𝑘,𝑛* ⋅ *∑ 𝐷𝑇 𝑠𝑢*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑠𝑢*=1

*𝑘,𝑛*

1

some tasks need to be migrated, the migration process task *𝑤𝑘,𝑛* gen-

(10)

one of the key objectives considered in the EC environment. *𝑇*

phase

*𝑠𝑢*

∨

∧

*𝐷𝑇 𝑠𝑢* =[ *𝐷𝑇 𝑠𝑢 , 𝐷𝑇 𝑠𝑢* ]= *ℎ*=1

(11)

*𝑚𝑚*′*,𝑠𝑢*

*∑𝐻*

*𝑠𝑢,ℎ*

erates energy consumption for *𝐸𝑚𝑖𝑔* , *𝑇*3 phase task *𝑤* generates exe-

*𝑘,𝑛*

*𝑇𝐷𝑆𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑑𝑡*

*𝑘,𝑛*

*𝑆*

*𝜐𝑘,𝑛* ⋅ *𝐷𝑘,𝑛* ⋅

*𝑘,𝑛*

*𝑤𝑘,𝑛* after the completion of the computation of the data generated by

*𝑘,𝑛*

*𝑘,𝑛*

cution energy consumption during its execution as *𝐸𝑒𝑥𝑒*, *𝑇*4 phase task

the transmission of the data to the successor task inevitably generates

*𝑑𝑡*

∨ ∧

*𝑝𝑘,𝑛* ⋅ *𝑔𝑘,𝑛*

the data transmission energy consumption for *𝐸𝑑𝑡* , for the total energy

*𝑆* = [*𝑆𝑑𝑡 , 𝑆𝑑𝑡* ]= *𝐵𝑑𝑡* ⋅ log2(1 +

) (12)

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝛿*2 + *𝑝𝑘,𝑛𝐼*

consumption for the task *𝑤𝑘,𝑛*

is:

*𝜌𝑘,𝑛* ∈ {0*,* 1}*,* ∀*𝑤𝑘,𝑛* ∈ *𝑤𝑘*

∈ {0*,* 1}*,* ∀*𝑤*

∈ *𝑤*

*𝑘,𝑛*

*𝑘,𝑛*

(17)

=[ ∨ ∧ ]= *⎧⎪𝐸𝑒𝑥𝑒* + *𝐸𝑑𝑡 , 𝜙𝑘,𝑛* =0

*𝑘,𝑛*

*𝜅*

*⎩*

*𝑘,𝑛 𝑘*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*⎨⎪𝐸𝑚𝑖𝑔* + *𝐸𝑒𝑥𝑒* + *𝐸𝑑𝑡 , 𝜙* =1

*𝜐𝑠𝑢* ∈ {0*,* 1}*, 𝑃𝑆𝑠𝑢* ∈ *𝑃𝑆𝑘*

*𝐸𝑡𝑜𝑡*

*𝐸𝑡𝑜𝑡 , 𝐸𝑡𝑜𝑡*

*𝑘,𝑛*

*𝑘,𝑛*

where *𝜌𝑘,𝑛* denotes whether the task is a terminal task, *𝜌𝑘,𝑛* =0 denotes that the task is a terminal task, otherwise *𝜌𝑘,𝑛* = 1; *𝐷𝑇 𝑠𝑢* denotes the

*𝑘,𝑛*

as:

The energy consumption generated at each stage can be expressed

∨ ∧

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

delay of data transmission to the server where the *𝑠𝑢*th successor node

*𝐸𝑚𝑖𝑔* = [*𝐸𝑚𝑖𝑔 , 𝐸𝑚𝑖𝑔* ]=*𝑇 𝑚𝑖𝑔* ⋅ *𝑃 𝑚𝑖𝑔*

(18)

is located; *𝜐𝑠𝑢* =0 denotes that the predecessor and successor tasks are

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

′ ∨ ∧

executed on the same server, otherwise *𝜐𝑠𝑢* = 1; *𝐷𝑚𝑚 ,𝑠𝑢* denotes the dis-

*𝐸𝑑𝑡* = [*𝐸𝑑𝑡 , 𝐸𝑑𝑡* ]=*𝑇 𝑑𝑡* ⋅ *𝑝𝑑𝑡*

(19)

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

tance between the server where the predecessor task *𝑤𝑘,𝑛* is located ∨ ∧

and the server where the corresponding zero or *𝑠𝑢*th successor task is *𝐸𝑒𝑥𝑒* = [*𝐸𝑒𝑥𝑒, 𝐸𝑒𝑥𝑒*]

located; *𝑇𝐷𝑆𝑠𝑢,ℎ* denotes the size of the *ℎ*th data generated after the ex-

*𝑘,𝑛*

ecution of the predecessor task *𝑤* ; and *𝑆𝑑𝑡* denotes the transmission

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛 𝑘,𝑛*

*⎧⎪𝑇 𝑒𝑥𝑒* ⋅ *𝑝𝑚,𝑒𝑥𝑒, 𝜙𝑘,𝑛* = 0*𝑎𝑛𝑑𝜆𝑘,𝑛* =0

*𝑒𝑥𝑒*

*𝑚*′*,𝑒𝑥𝑒*

(20)

*𝑘,𝑛 𝑘,𝑛*

*⎪⎨*

rate of the link between the servers where the predecessor and succes- sor tasks reside.

Since the start node has no predecessor task, it is always active i.e.

= *𝑇𝑘,𝑛* ⋅ *𝑝𝑘,𝑛 , 𝜙𝑘,𝑛* = 1*𝑎𝑛𝑑𝜆𝑘,𝑛* =0

*⎪*[*𝜏* − *𝑇 𝑟* ] ⋅ *𝑝𝑚,𝑒𝑥𝑒*+ *𝑘,𝑛* ⋅ *𝑝𝑚 ,𝑒𝑥𝑒, 𝜙𝑘,𝑛* = 1*𝑎𝑛𝑑𝜆𝑘,𝑛* =1

*𝑚𝑠𝑘,𝑛* ⋅cc*𝑚*′

′

*⎪⎩*

*𝑘,𝑛*

*𝑘,𝑛*

*𝑅𝑚*′

*𝑘,𝑛*

*𝑘,𝑛*

*𝑎𝑐𝑡𝑘,𝑛* = 0, for other subtasks *𝑤𝑘,𝑛* activation requires two conditions: (1)

where *𝑃 𝑚𝑖𝑔* denotes the power to migrate the task *𝑤* ; *𝑝𝑑𝑡* denotes the

Ensure that the tasks have been migrated to the server at this point.

*𝑘,𝑛*

*𝑘,𝑛*

*𝑘,𝑛*

(2) Since there are one or more predecessor tasks for *𝑤*

and each

power in which task *𝑤𝑘,𝑛* transfers the data; *𝑝𝑚,𝑒𝑥𝑒* denotes the execu-

*𝑘,𝑛*

*𝑘,𝑛*

′

predecessor task generates one or more pieces of data when it is com-

tion power of the server *𝑚* to perform the task *𝑤𝑘,𝑛* ; *𝑝𝑚 ,𝑒𝑥𝑒* denotes the

pleted, all predecessor tasks need to transfer data to the server where

execution power of the server *𝑚*′ to perform the task

*𝑘,𝑛*

*𝑤𝑘,𝑛* .

conditions, task *𝑤𝑘,𝑛* can be activated, so except for the start node, the the successor task is located after completion of the task, with the above activation of the task time *𝑎𝑐𝑡* is the migration delay and the maxi-

*𝑘,𝑛*

mum value of the last feedback time for all precursor tasks:

Objective *𝑓*3 is to minimize the total energy consumption:

∨ ∧ *𝐾 𝑁*

*∑ ∑*

*𝐸*

min *𝑓*3 = min[*𝑓*3*, 𝑓*3]= min

*𝑡𝑜𝑡*

*𝑘,𝑛*

(21)

*𝑎𝑐𝑡𝑘,𝑛* = [*𝑎𝑐𝑡𝑘,𝑛, 𝑎𝑐𝑡𝑘,𝑛*]= *𝑤𝑠𝑘,𝑛* ⋅ max

*𝑇𝑘,𝑛 ,* max

*𝑇𝑘,𝑛* + *𝐷𝑇𝑘,𝑛*

*𝑘*=1 *𝑛*=1

1. Load balancing

The migration controller obtains the status information of each area server cluster, including the computing capacity of each server, the impact of the degree of completion of server tasks on load before migra- tion, and joint consideration of the impact on server load of inter-area migration of tasks generated by users in surrounding areas, and chooses to migrate the various subtasks of the workflow to a suitable new server farm within MD’s coverage area. To ensure eﬃcient execution while balancing the workload and avoiding overloading to reduce operational

eﬃciency, the specific calculation of load balancing *𝐿* is as follows:

*𝑀*  2

*∑*

 ∨ ∧ [*𝐿𝑚* − *𝐿𝑎*]

# Proposed MI-MaOEA

* 1. *Algorithmic framework*

In this paper, MI-MaOEA is proposed for solving the I-MaOWMUE model, and Algorithm [1](#_bookmark11) demonstrates the algorithmic framework of

MI-MaOEA. First, the population *𝑃𝑡* is randomly initialized. Then, the

mating pool is constituted utilizing *𝑃𝑡* . Similar to other evolutionary al- gorithms, the offspring *𝑄𝑡𝑚* is generated by simulated binary crossover (SBX) and polynomial mutation (PM). Finally, *𝑃𝑡* is merged with *𝑄𝑡𝑚*

to select the next-generation individuals through an objective-value- dominated hierarchical sorting and dual-migration selection strategy.

*𝐿* = [*𝐿, 𝐿*]= *𝑚*=1

*𝐿𝑎*

*𝑀*

*∑*

∨

∧

*𝐿𝑚*

(22)

The above steps are repeated until the maximum number of iterations is reached and MI-MaOEA ends.

*𝑎 𝑎 𝑎 𝑀*

**Algorithm 1** MI-MaOEA

*𝐿*

= [*𝐿 , 𝐿* ]= *𝑚*=1

∧

**Output:** Last generation of populations: *𝑃𝑡*max

*𝑚𝑎𝑥*

0 ≤ *𝐿𝑚* ≤ *𝑠𝑠𝑚,* ∀*𝑚* ∈ *𝑀*

(23)

**Input:** Population size *𝑁* , Maximum number of iterations *𝑡*

where *𝐿𝑚* denotes the storage space occupied by server *𝑚* during the migration phase and *𝐿𝑎* denotes the average load of all ESs, calculated

as follows:

1: Initialize: *𝑃𝑡* = {*𝑃*1 *, 𝑃*2 *, 𝑃*3 *, ..., 𝑃𝑁* }

2: **for** *𝑡* ← 1 to *𝑡𝑚𝑎𝑥* **do**

3: *𝑃𝑡𝑚* = Mating selection(*𝑁* , *𝑃𝑡* ) // Refer to Algorithm [2](#_bookmark13) 4: *𝑄𝑡𝑐* = SBX(*𝑁* , *𝑃𝑡* , *𝑃𝑡𝑚* )

5: *𝑄𝑡𝑚* = PM(*𝑁* , *𝑄𝑡𝑐* , *𝑃𝑡𝑚* )

6: *𝑍* = *𝑃𝑡* ∪ *𝑄𝑡𝑚*

∨ ∧ *𝐿𝑜𝑓 𝑓* − *𝐿𝑐* + *𝐿𝑚𝑡* − *𝐿𝑚𝑎*

*𝐿* = [*𝐿 , 𝐿* ]=

*𝑚 𝑚 𝑚 𝑚*

(24)

7: *𝑃𝑡*+1 = Environmental selection(*𝑁* ,*𝑍*) // Refer to Algorithm [3](#_bookmark17)

8: **end for**

*𝑚 𝑚 𝑚*

*𝑅*

*𝑚*

*𝑘,𝑛*

*𝑚*

where *𝐿𝑜𝑓𝑓*

*𝑚*

denotes the amount of tasks oﬄoaded to server *𝑚*; *𝐿𝑐*

* 1. *Mating selection*

denotes the amount of tasks that server *𝑚* finished performing before

migration; *𝐿𝑚𝑡* denotes the amount of tasks that have been migrated to server *𝑚* from other areas; and *𝐿𝑚𝑎* denotes the amount of tasks that have been migrated away from server *𝑚*. The calculation is as follows:

*𝑚*

*𝑚*

*𝐺𝑂𝑚*

*𝐿* = *∑ 𝑢𝑠*

Comparison of each individual through Pareto non-dominance sort- ing to determine the relationship between them. Diversity of solutions and flexibility in decision-making can be ensured by dividing candidate solutions into dominated and non-dominated solutions and obtaining

*𝑜𝑓 𝑓*

*𝑚*

g=1

*𝐺𝐸𝑚*

*𝑚,𝑔*

(25)

relatively optimal non-dominated solutions. However, the above meth- ods do not allow direct comparison of interval objective values, and in order to measure the quality of optimal solutions to interval many-

*𝐿𝑐* = *∑* [𝓁*𝑚,𝑔* ⋅ (*𝑢𝑠𝑚,𝑔* − *𝑚𝑠𝑚,𝑔* )+ (1 − 𝓁*𝑚,𝑔* ) ⋅ *𝑢𝑠𝑚,𝑔* ] (26)

*𝑚*

objective optimization problems, this paper performs a comparison of dominance relationships by means of the interval confidence strategy.

*𝑔*=1

∨ ∧ ∨ ∧ ∨ ∧

*𝑀 𝐺𝑚*′ *𝑚*

For the interval values *𝜅*1 = [*𝜅*1 *, 𝜅*1] and *𝜅*2 = [*𝜅*2 *, 𝜅*2], we use *𝜅* = [*𝜅, 𝜅*]

*𝐿𝑚𝑡* = *∑*

*𝑚*

*𝑚*′=1*𝑎𝑛𝑑𝑚*′ ≠*𝑚 𝑔*=1

*𝑚 ,𝑔*

*𝑚 ,𝑔*

*𝑚 ,𝑔*

*∑* [*𝜓* ′

⋅ *𝑚𝑠* ′

+ (1− *𝜓* ′

) ⋅ *𝑢𝑠* ′

] (27) ∨ ∧

*𝑀 𝐺𝑚𝑚*′

*𝑚 ,𝑔*

*∑*

*𝜅*1

*𝜅*1

*𝜅*2

*𝜅*2

*𝐿𝑚𝑎* =

*𝑚*

′ ′

*∑* [*𝜉𝑚*′ *,𝑔*

⋅ *𝑚𝑠𝑚*′ *,𝑔*

+ (1− *𝜉𝑚*′*,𝑔*

) ⋅ *𝑢𝑠𝑚*′*,𝑔*

] (28)

the confidence level of the intervals in which *𝜅*1 is less than *𝜅*2 as:

*𝑑*(*𝜅 , 𝜅*)

as the minimum interval, where *𝜅* and *𝜅* are the minimum and second

smallest values of the ∨ , ∧ , ∨ and ∧ , respectively, and we represent

*𝑚* =1*𝑎𝑛𝑑𝑚* ≠*𝑚 𝑔*=1

*𝑃* (*𝜅*1 *< 𝜅*2)= 2

(31)

where *𝐺𝑂𝑚* denotes the number of tasks oﬄoaded to server *𝑚*; *𝐺𝐸𝑚*

*𝑑*(*𝜅*1*, 𝜅*)+ *𝑑*(*𝜅*2*, 𝜅*)

denotes the number of tasks executed by server *𝑚*; *𝐺𝑚*′*𝑚* denotes the number of tasks migrated from *𝑚*′ to *𝑚*; *𝐺𝑚𝑚*′ denotes the number of

′

where *𝑑*(*𝜀*1*, 𝜀*2) denotes the distance from the objective value of the interval and *𝑑*(*𝜀*1*, 𝜀*2) is calculated as follows:

tasks migrated from *𝑚* to *𝑚* ; 𝓁*𝑚,𝑔* denotes whether the task has been

then 𝓁*𝑚,𝑔* = 1, otherwise it means that the task execution is completed, at partially executed before the migration or not, respectively, and if yes,

*𝑑*(*𝜀 , 𝜀* )= *√*(∨ − ∨ 2 + (∧ − ∧ )2

(32)

this time 𝓁*𝑚,𝑔* = 0; and *𝜓𝑚*′*,𝑔* denotes whether the task has been partially executed before the migration or not, and if yes, then *𝜓𝑚*′ *,𝑔* = 1. Other-

wise, it means that the task has not been executed yet, when *𝜓𝑚*′*,𝑔* = 0.

It is known from the interval confidence that the solution *𝑥*1 in-

terval dominates the solution *𝑥*2 (*𝑥*1 *≺ 𝑥*2 ) if and only if the following

1 2 *𝜀*1 *𝜀*2) *𝜀*1 *𝜀*2

conditions are met:

*𝜉𝑚*′*,𝑔* is consistent with the definition of *𝜓𝑚*′ *,𝑔* . Also needs to be satisfied:

*{* ∀*𝑙* ∈ {1*,* 2*,* 3*, ..., 𝜛*} *, 𝑃*(*𝑓𝑙* (*𝑥*1*, 𝑔*) *< 𝑓𝑙* (*𝑥*2*, 𝑔*)) ≥ 0*.*5

(33)

𝓁*𝑚,𝑔*

∈ {0*,* 1}*,* ∀*𝑔* ∈ *𝐺𝐸𝑚*

∃*𝑗* ∈ {1*,* 2*,* 3*, ..., 𝜛*} *, 𝑃*(*𝑓𝑗* (*𝑥*1*, 𝑔*) *< 𝑓𝑗* (*𝑥*2*, 𝑔*)) *>* 0*.*5

where *𝜛* denotes the objective number.

*𝜓𝑚*′*,𝑔* ∈ {0*,* 1}*,* ∀*𝑔* ∈ *𝐺𝑚*′ *𝑚*

*𝜉𝑚*′*,𝑔* ∈ {0*,* 1}*,* ∀*𝑔* ∈ *𝐺𝑚𝑚*′

Objective *𝑓*4 is to minimize load balancing:

selects two solutions *𝑥*1, *𝑥*2 of the parent and compares their relations In Algorithm [2](#_bookmark13) we give a matching selection method that randomly of domination: 1) if *𝑥*1 *≺ 𝑥*2 add solution *𝑥*1 to the mating pool, 2) if

*𝑥*2 *≺ 𝑥*1 add solution *𝑥*2 to the mating pool, and 3) otherwise randomly

∨ ∧  select a solution to be added to the mating pool.

*𝑚𝑖𝑛𝑓*4 = *𝑚𝑖𝑛*[*𝑓*4 *, 𝑓*4]= *𝑚𝑖𝑛𝐿* (29)

In this paper, the constructed I-MaOWMUE model is defined as a minimization optimization problem, defined as follows:

*𝑚𝑖𝑛𝐹* (*𝑋*)= *𝑚𝑖𝑛 {𝑓*1*, 𝑓*2*, 𝑓*3*, 𝑓*4*}* (30)

* 1. *Environmental selection*

Considering that the interval objective values cannot be compared between individuals in the same non-dominated layer, they can be com-

**Algorithm 2** Mating Selection

**Input:** Population size *𝑁* , Parent *𝑃𝑡*

**Output:** mating pool *𝑃𝑡𝑚*

1: *𝑃𝑡𝑚* = []

2: **for** *𝑖* ← 1 to *𝑁* **do**

3: *𝑥*1 , *𝑥*2 = random select(*𝑃𝑡* ) 4: **if** *𝑥*1 *≺ 𝑥*2 **then**

5: *𝑃𝑡𝑚 .𝑎𝑝𝑝𝑒𝑛𝑑*(*𝑥*1) 6: **else if** *𝑥*2 *≺ 𝑥*1 **then** 7: *𝑃𝑡𝑚 .𝑎𝑝𝑝𝑒𝑛𝑑*(*𝑥*2)

8: **else**

9: *𝑠* = random select(*𝑥*1 ,*𝑥*2 )

10: *𝑝𝑡𝑚 .𝑎𝑝𝑝𝑒𝑛𝑑*(*𝑠*)

11: **end if**

12: **end for**

pared using the interval crowding distance, which will be presented in Section [4.3.1](#_bookmark15). For the interval many-objective optimization problem of

Specifically, in the last level of selection, to increase the search range

the selection according to the following steps: (1) Select *𝑞𝑞* = *𝛾* ⋅ *𝐽* in- of feasible solutions and ensure the diversity of solutions, we complete dividuals based on interval crowding distance, where *𝛾*(*𝛾* ≥ 1) denotes

*⌊ ⌋*

layer *𝐹ℎ* . If *𝑞𝑞* is greater than the number of individuals in layer *𝐹ℎ* , the environmental factor for interval crowding distance selection in step 2 is performed, otherwise, step 3 is performed; (2) Use *𝑂𝑑𝑖𝑠* to sort the individuals in stratum *𝐹ℎ* in descending order and select *𝐺* individ- uals; (3) Sort the individuals of stratum *𝐹ℎ* in descending order using

*𝑂𝑑𝑖𝑠* and select *𝑞𝑞* individuals from them, and then select *𝐽* individuals

are then selected from them using *𝑆* ascending order. We take the mid- point of *𝑓*1 to provide a diversity of choice support. *𝑆* is calculated as

follows:

*𝑆* = *𝑓*1 ⋅ *𝑆𝑅* (35)

*𝑀 𝐺𝑚𝑖𝑔*

*∑ ∑*

Γ*𝑚,𝑔* ⋅ *𝑇𝑆𝑚,𝑔*

computational migration, this paper proposes a dual-migration selec-

tion strategy, which will be introduced in Section [4.3.2](#_bookmark18).

*𝑆𝑅* =

*𝑚*=1 *𝑔*=1

*𝑀 𝐺*

⋅ 100% (36)

*∑ ∑𝑚𝑖𝑔*

*𝑇𝑆𝑚,𝑔*

* + 1. *Interval crowding distance*

Individuals in stratum *𝐹ℎ* were sorted according to interval con-

fidence to determine the relationship of individuals. In general, we

the distribution of the solution *𝑥* to the two nearest individuals on ei- measure the interval crowding distance in terms of the denseness of ther side of it, and we make use of the interval crowding distance *𝑂𝑑𝑖𝑠*

for the selection, computed as follows:

*⎧ ∑𝜛 𝑑*(*𝑓 𝑜*+1 *,𝑓 𝑜*−1)

=0

*𝑚*=1 *𝑔*=1

where *𝑆𝑅* denotes the migration success rate and Γ*𝑚,𝑔* denotes whether the migration of the *𝑔*th task to the *𝑚*th server is successful or not, if the migration is successful then Γ*𝑚,𝑔* = 1, otherwise Γ*𝑚,𝑔* = 0.

First, the *𝑍* individuals from the merger of the parent *𝑃𝑡* and the Algorithm [3](#_bookmark17) gives the overall process of environmental selection. offspring *𝑄𝑡𝑚* are sorted according to the interval dominance, and di- vided into *𝑙* layers, where layer 1 is the lowest and preferred to be

selected, followed by layer 2, and so on. Then, individuals of size *𝑁*

are then selected layer by layer for the next generation. When the se-

*𝑂*dis = *⎪⎨*

*⎪⎩*

*𝑖*

*𝑖*

*,* Υ

*𝑖*

*𝑖*

*𝑑*(*𝑓* max *,𝑓* min ) *𝑂*

(34)

Υ*𝑂*

*𝑖*=1

+∞*,* Υ*𝑂* =1

∈ {0*,* 1}*,* ∀*𝑂* ∈ *𝐹ℎ*

lected individuals belong to the same non-dominated stratum, we select individuals by interval crowding distance and dual-migration selection strategy as the selection basis.

where *𝑓 𝑜*+1 and *𝑓 𝑜*−1 denote the ith objective value of the *𝑂* + 1th and

*𝑖 𝑖*

*𝑂* − 1th solution of the *𝑂*th individual of the sorted set, respectively.

*𝑓* max and *𝑓* min denote the maximum and minimum values of the ith

**Algorithm 3** Environment Selection

*𝑖 𝑖*

objective function. Υ*𝑂* =0 means that the individual is not a boundary solution, otherwise Υ*𝑂* = 1.

* + 1. *Dual-migration selection strategy*

In the migration phase, the excessive migration delay of the task cor- respondingly increases the migration cost, and even though the server computing time is shortened, it is not enough to compensate for the additional consumption caused by the migration process, which is not permissible for the whole system. For ES, it needs to receive migration tasks from multiple neighboring regions at the same time. Even if ES re- leases part of the storage resources during the computation process, the

limited server *𝑐𝑝* and *𝑠𝑠* cannot support a huge amount of task migra-

tion. The insuﬃcient supply of computational capacity is likely to lead

to server damage. To improve the overall effectiveness of the server, it is necessary to seek a suitable method to formulate a reasonable mi- gration strategy. For this reason, we propose a dual-migration selection strategy. We chose to use individuals with small migration delays and low migration success as an important basis for selection at the final level. The small migration delays mean that a large number of tasks are selected to be migrated to the nearby servers, which significantly reduces task migration delay costs and energy consumption. Although the above approach makes the migration delay shorter, the servers near the users may be overloaded, which reduces the migration success rate, which is not the best solution for task migration. However, from an- other perspective, although the server is overburdened, there may be a situation where tasks of the same workflow are migrated to the same server, which in turn significantly reduces the data transfer delay and energy consumption, etc., and the servers, although in a state of im- balance for a short period, reduce the time for the completion of the workflow.

**Input:** Population size *𝑁* , Parental generation *𝑃𝑡* merges with paternal generation *𝑄𝑡𝑚* :

*𝑍*, *𝑌* = 0, *ℎ* = 1, *𝐴* = [], *𝐵* = [], *𝑃𝑡*+1 = []

**Output:** The next generation of populations: *𝑃𝑡*+1

1: *𝐹* = *𝐹*1 *, 𝐹*2 *, ..., 𝐹ℎ, ..., 𝐹* = Non-dominated ordering descending(*𝑍*) 2: **while** *𝑌 < 𝑁* **do**

3: *𝑌* = *𝑌* ∪ *𝐹ℎ* , *ℎ* = *ℎ* +1

4: **end while**

5: *𝑃𝑡*+1 = *𝑃𝑡*+1 ∪ *𝐹*1 *, 𝐹*2 *, ..., 𝐹ℎ*−1 , *𝑌* = *𝑌* − *𝐹ℎ*

6: The number of individuals selected in the final layer: *𝐽* = *𝑁* − *𝑌*

7: *𝑞𝑞* = *𝛾* ⋅ *𝐽*

*⌊ ⌋*

8: **if** *𝑞𝑞* > length(*𝐹ℎ* ) **then**

9: *𝐺* = *𝐽*

10: **else**

11: *𝐺* = *𝑞𝑞*

12: **end if**

13: **for** each individual in *𝐹ℎ* **do**

14: Sort *𝑂𝑑𝑖𝑠* by eq. ([34](#_bookmark16)) in descending order and put the result into *𝐴*

15: **end for**

16: **for** *𝑖* ← 1 to *𝐺* **do**

17: *𝐵.𝑎𝑝𝑝𝑒𝑛𝑑*(*𝐴*[*𝑖*])

18: **end for**

19: **for** each individual in *𝐵* **do**

20: Sort *𝑆* by eq. ([35](#_bookmark14)) in ascending order and put the result into *𝐵*

21: **end for**

22: **for** *𝑖* ← 1 to *𝐽* **do**

23: *𝑃𝑡*+1 *.𝑎𝑝𝑝𝑒𝑛𝑑*(*𝐵*[*𝑖*])

24: **end for**

# Experiments

In order to evaluate the performance of MI-MaOEA on the I- MaOWMUE model, we need to conduct validation experiments and compare it with the MI-MaOEA algorithm using state-of-the-art inter- val multi-objective evolutionary algorithms, DI-μMOGA [[45](#_bookmark62)], InMaOEA

**Table 1**

Parameters of the MI-MaOEA algorithm.

|  |  |  |
| --- | --- | --- |
| Variable | Description | Value |
| *𝑁* | Population size | 100 |
| *𝑀* | Objective number | 4 |
| *𝑡𝑚𝑎𝑥* | Maximum number of iterations | 100 |
| *𝜂* | Distribution factor | 20 |
| *𝑝𝑚* | Crossover probability | 1.0 |
| *𝑞𝑚* | Mutation probability | 0.01 |

[[46](#_bookmark64)], and II-MOEA [[47](#_bookmark65)]. DI-μMOGA proposes to use Monte Carlo stochastic simulation methods to seek optimization objective intervals and uses the degree of interval constraint violation to deal with con- straints. InMaOEA uses this interval credibility strategy to improve the convergence of the algorithm and the interval congestion distance strat- egy to improve population diversity. II-MOEA is a classical method for solving U-MaOPs problems, which defines the dominance rela- tion through the interval confidence level and the crowding distance through the location and volume of the hyper-cuboids.

* 1. *Simulation environment and parameter settings*

Simulation environment: Windows 11 Home Edition; AMD Ryzen 7 4800H with Radeon Graphics with 2.9 GHz; NVIDA GeForce GTX 1650 with 8 GB memory; MATLAB R2023a development platform.

Table [1](#_bookmark20) provides information about the parameter settings of the MI-MaOEA algorithms in this simulation experiment. To make the com- parison of the algorithms fairer, for all the algorithms we use the same parameter settings, setting the population size to 100, the number of ob-

the parameter setting sizes for *𝜂*, *𝑝𝑚* , and *𝑞𝑚* are also listed in Table [1](#_bookmark20), jectives to 4, and the maximal number of iterations to 100. In addition,

and we will verify in Section [4.2](#_bookmark12) that the reasonableness of the param- eter size settings, and the rest of the parameter size settings are based on the original literature. Table [2](#_bookmark21) gives the constraints on the impor- tant parameter settings for constructing the I-MaOWMUE model, and we randomly assign values to each parameter within the given range of values to apply to different migration scenarios and computational conditions.

We simulate task migration in real IoT environments using five dif- ferently structured benchmark workflows for discussion and evaluation of the performance of EC systems from well-known scientific appli- cations [[48](#_bookmark67)], namely the Montage astronomy workflow, Epigenomics workflow, CyberShake earthquake hazard characterization workflow, SIPHT workflow that searches for small untranslated RNAs, and Inspiral physics workflow, which are widely used to evaluate the performance of workflow scheduling problems.

* 1. *Results and analysis*

The parameters of the proposed MI-MaOEA algorithm should be ad- justed and evaluated according to the proposed I-MaOWMUE model to find the combinations that perform well on the I-MaOWMUE model.

distribution index *𝜂*, the crossover probability *𝑝𝑚* , and the mutation The setting of the values of the three important parameters of the probability *𝑞𝑚* in the MI-MaOEA algorithm directly affects the search eﬃciency and the quality of the solution. In general, a higher *𝜂* in-

the range of values is generally controlled between 5 and 20. Higher *𝑝𝑚* creases the probability of the offspring approximating the parent, and

increases population diversity, but occasionally over-convergence oc- curs, and the range of values was kept between 0.7 and 1.0. Higher

*𝑞𝑚* increases stochasticity, but unstable values may occur and keep the

range of values between 0.001 and 0.01. For this reason, this paper

conducts several sets of experiments to determine the most appropriate parameter settings so that the solution can satisfy faster convergence and better diversity.

Table [3](#_bookmark22) shows the impact of the MI-MaOEA algorithm on the four optimization objectives of migration delay, maximum completion time, energy consumption, and load balancing in the I-MaOWMUE model for different parameter value settings, and we compare the mean, maxi- mum, and minimum values on the four optimization objectives respec- tively, where the average value is a clearer expression of the degree to which the algorithm is good or bad, and our goal is to minimize each objective function value, and the best results are shown in bold. By an-

alyzing the table, it can be concluded that when the parameter *𝜂* is set

to 20, parameter *𝑝𝑚* is set to 1, and parameter *𝑞𝑚* is set to 0.01, the av-

erage and minimum values of migration delay, maximum completion

time, and energy consumption are optimal as compared to the other pa- rameter combinations of the value settings, and the upper and lower bounds of each objective are smaller than those of the other param- eter combinations, which is a desirable optimization result. Although it fails to outperform the other algorithms at the maximum value of energy consumption and load balancing, it shows outstanding perfor-

summary, adopting the parameter combination settings of *𝜂*= 20, *𝑝𝑚*= 1, mance in terms of migration time and maximum completion time. In and *𝑞𝑚*= 0*.*01 makes the algorithm MI-MaOEA more superior and robust

in solving the I-MaOWMUE model.

In Table [4](#_bookmark23), in order to compare the performance effects of differ- ent algorithms on the four optimization objectives of the I-MaOWMUE model, we compare the mean, maximum, and minimum values on the four optimization objectives, where the mean value more clearly ex- presses the degree of goodness of the algorithms, and for each objective, the smaller the objective value represents the better the performance effect and the best results are marked in bold. From Table [4](#_bookmark23), it can be seen that the MI-MaOEA algorithm shows consistent and superior

**Table 2**

I-MaOWMUE model parameter constraints.

Variable Description Value

Number of mobile devices 5

Number of base stations 20

Workflow type 5

Area 1 communications coverage 6 km\*6 km

Area 2 communications coverage 4 km\*6 km

*𝑝𝑘,𝑛* Upload power for task *𝑤𝑘,𝑛* [100,500] W

*𝑢𝑠𝑘,𝑛* Task *𝑤𝑘,𝑛* size [50,500] MB

*𝛿* Communications noise [-100-10] W

*𝐼* Communications interference [0.01,0.02]

*𝑔𝑘,𝑛* Communication gain of task *𝑤𝑘,𝑛* [0.2,0.4] Mbps

*𝐵𝑚𝑖𝑔* Migration bandwidth for task *𝑤𝑘,𝑛* [5,20] Mbps

*𝐵𝑑𝑡* Task *𝑤𝑘,𝑛* data transmission bandwidth [1.5,3] Mbps

*𝑅𝑚* The number of CPU cycles per second that can be computed by server m [400,450] cycle/s

*𝑚𝑖𝑔*

*𝑃*

*𝑘,𝑛*

*𝑑𝑡*

*𝑝*

*𝑘,𝑛*

Migration energy consumption [5,6] MJ

Data transfer energy consumption [0,1] MJ

*𝑚,𝑒𝑥𝑒*

*𝑝*

*𝑘,𝑛*

Energy consumed by task *𝑤*

*𝑘,𝑛*

to compute 1 CPU cycle at server m [2,3] MJ

**Table 3**

Parameter setting experiment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *𝜂 𝑝𝑚 𝑞𝑚* | Migration Delay | Maximum Completion Time | Energy Consumption | Load Balancing |
| 20 1.0 0.01 | **[29.711161,30.1336498]** | **[22.804685,23.3616373]** | **[136.7282042,137.3782608]** | **[0.0848622,0.7509751]** |
| Mean 15 0.9 0.007 | [30.284265,30.669576] | [23.5883838,24.0716393] | [141.3749482,141.851065] | [0.3398962,0.837589] |
| 10 0.8 0.004 | [32.1544108,32.3979383] | [27.6639578,28.0678613] | [140.8662545,141.3589961] | [0.4072463,0.9352577] |
| 5 0.7 0.001 | [32.6213206,33.155733] | [27.6949102,28.2571371] | [146.9854565,147.5287044] | [0.4445313,1.0148102] |
| 20 1.0 0.01 | **[31.3272123,32.1932377]** | **[24.2524596,25.2522005]** | [146.3478425,147.2138679] | [0.6957946,1.6380555] |
| Max 15 0.9 0.007 | [32.5683788,33.2341189] | [29.0679496,29.3812936] | [147.6913113,147.8724276] | [0.8311289,**1.4845266]** |
| 10 0.8 0.004 | [33.4876592,33.9678162] | [31.8904347,32.7328054] | **[142.5541255,143.5517943]** | [2.5855946,3.4025645] |
| 5 0.7 0.001 | [34.1015124,34.6210964] | [27.923119,28.9228598] | [154.170455,154.9874249] | **[0.5135437**, 1.5008733] |
| 20 1.0 0.01 | **[29.3096505,29.4005854]** | **[21.6134043,21.7945205]** | **[135.9613069,136.0468097]** | **[0.0053804,0.1112645]** |
| Min 15 0.9 0.007 | [29.9449029,30.0307611] | [22.1152358,22.2061706] | [140.9512022,141.2042048] | [0.1001481,0.1456627] |
| 10 0.8 0.004 | [31.8792113,31.9692206] | [24.627043,24.7179778] | [140.3557766,140.59765] | [0.1603905,0.2965571] |
| 5 0.7 0.001 | [32.3079131,32.4440797] | [27.3135959,27.6434887] | [145.7673041,146.0092017] | [0.3796501,0.4705849] |

**Table 4**

Comparison of the objective values of four algorithms on the I-MaOWMUE model.

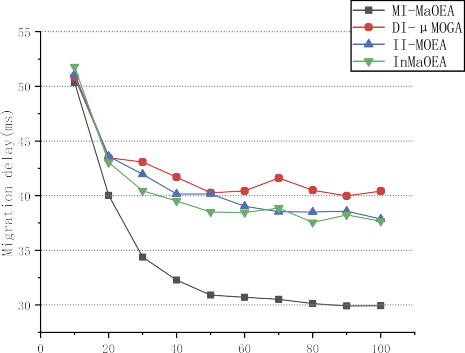
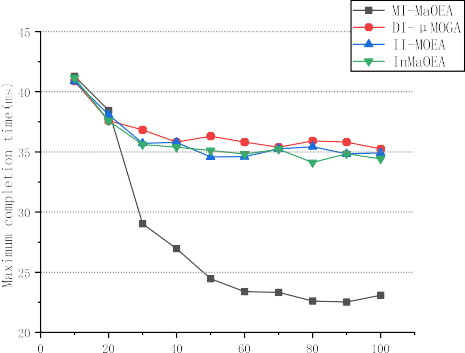
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Migration Delay | Maximum Completion Time | Energy Consumption | Load Balancing |
| MI-MaOEA | **[29.711161,30.1336498]** | **[22.804685,23.3616373]** | **[136.7282042,137.3782608]** | **[0.0848622,0.7509751]** |
| Mean DI-μMOGA | [40.090148,40.7592748] | [34.9260306,35.5662294] | [147.8948972,148.5279753] | [1.4806598,2.0857283] |
| II-MOEA | [37.5503117,38.1812611] | [34.6287875,35.2204535] | [159.0845489,159.7607118] | [1.6637146,2.3347694] |
| InMaOEA | [37.3343011,37.9819818] | [34.0898393,34.7578041] | [155.9052669,156.539373] | [1.8459827,2.4281237] |
| MI-MaOEA | **[31.3272123,32.1932377]** | **[24.2524596,25.2522005]** | **[146.3478425,147.2138679]** | **[0.6957946,1.6380555]** |
| Max DI-μMOGA | [50.5688727,51.0086075] | [39.6541748,39.6541748] | [175.1968137,175.1968137] | [2.7586038,3.2907886] |
| II-MOEA | [46.3692592,47.159135] | [39.9270916,40.8959008] | [210.1066616,210.5868185] | [2.6972205,3.6495187] |
| InMaOEA | [46.7995969,47.465337] | [42.1384334,43.1175175] | [191.1862898,191.367406] | [3.849449,4.79171] |
| MI-MaOEA | **[29.3096505,29.4005854]** | **[21.6134043,21.7945205]** | **[135.9613069,136.0468097]** | **[0.0053804,0.1112645]** |
| Min DI-μMOGA | [32.1034223,32.6558227] | [28.385888,29.3732177] | [139.2552522,139.6373911] | [0.1126588,0.3824556] |
| II-MOEA | [33.4377613,33.8401625] | [31.5321978,32.0698001] | [139.062606,139.1081206] | [1.2233632,1.3595299] |
| InMaOEA | [33.1676769,33.566078] | [28.9984236,29.8863088] | [141.1859596,142.0938643] | [1.1911343,1.3270775] |

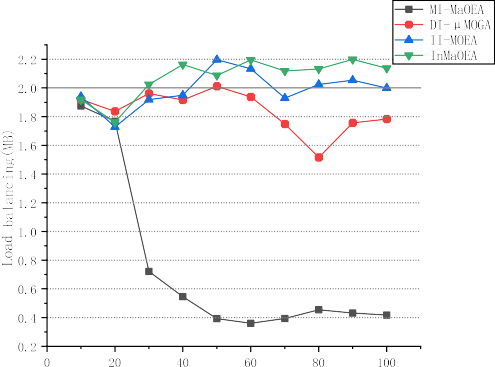
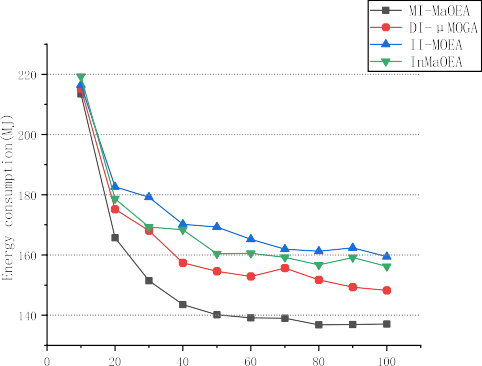
performance in the four optimization objectives of migration delay, maximum completion time, energy consumption, and load balancing for the I-MaOWMUE model, and it achieves the optimal results in terms of the mean, maximum and minimum values compared to the DI-μMOGA, II-MOEA and InMaOEA algorithms. MI-MaOEA algorithm can greatly shorten the migration delay and maximum completion time of the task, and ensure that the task can be more evenly distributed to the various servers to achieve load balancing, for the more distant servers, even if it increases the migration energy consumption, but also to ensure that the rapid processing of the dependent tasks, reducing the corre- sponding energy consumption of the execution, which indicates that the MI-MaOEA algorithm to maintain the balance of the four conflicting ob- jectives at the same time to ensure that the quality of the solution, to avoid falling into the local optimum, which is due to the fact that we use the dual-migration selection strategy, so that the iterative process can be jumped out of the local optimum, to ensure that the diversity of the solution. This indicates that the MI-MaOEA algorithm can effectively solve the I-MaOWMUE problem and provide better migration strate- gies for decision-makers. In addition, when the MI-MaOEA algorithm solves the problem, the upper bound of the interval for each objective is smaller than the lower bound of the interval for the corresponding objective of the other algorithms, which indicates that the MI-MaOEA algorithm not only achieves better results overall but also provides bet- ter performance on each objective. In summary, based on the analysis of the results in Table [4](#_bookmark23), the MI-MaOEA algorithm shows the best per- formance on all four optimization objectives of the I-MaOWMUE model, with significant advantages.

Fig. [4](#_bookmark24) shows the effect of all the algorithms on the optimization of each objective value of the I-MaOWMUE model in the first 100 itera- tions to demonstrate the convergence of the MI-MaOEA algorithm. To give credibility to the comparison results, we set the same parameters for all the algorithms as a way to assess the convergence speed and ob- jective optimization effect of the different algorithms, while we took

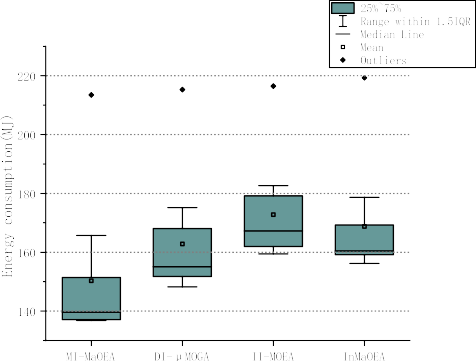
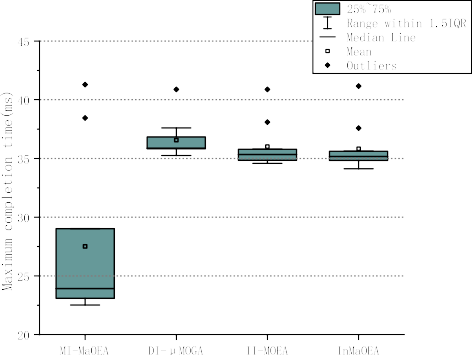
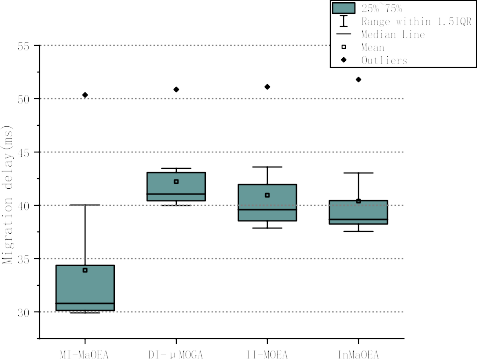
the objective values every 10 generations and averaged these objec- tive values to plot them in Fig. [4](#_bookmark24). From Fig. [4](#_bookmark24), we can see that when MI-MaOEA, DI-μMOGA, II-MOEA, and InMaOEA solve the problem, the value of the objective function of the problem gradually tends to sta- bilize as the number of iterations increases. In particular, in the first 30 iterations, the MI-MaOEA algorithm obtained the most significant changes in the objective values, with faster convergence, which means that the MI-MaOEA algorithm obtains the optimal solution faster dur- ing the optimization process and obtains the best four objective values compared to the other three algorithms. In the environment selection phase, we adopt an individual-based interval objective value hierar- chical sorting strategy, which is ordered according to the individual objective value and can more effectively screen out the excellent indi- viduals and introduce them to the next generation, and the MI-MaOEA algorithm utilizes this strategy to exert greater selection pressure, thus pushing the algorithm to converge to the optimal solution more quickly in the convergence phase. In the middle of the iterations (30-60), the convergence of the MI-MaOEA algorithm slows down as the number of iterations increases, good individuals are selected frequently, and poorer individuals are progressively eliminated, leading to a decrease in the diversity in the population. As the objectives become more and more conflicting with each other, making it unable to sort individuals for non-dominance, the convergence of the MI-MaOEA algorithm slows down. Since the algorithm MI-MaOEA employs a dual-migration selec- tion strategy to provide new possible solutions to the decision space and to avoid falling into local optima, the optimal values of the four objec- tive values obtained are better than those of the other algorithms and reach the optimal values after 60 iterations. In summary, based on the analysis of the results in Fig. [4](#_bookmark24), the MI-MaOEA algorithm has a faster convergence rate compared to other algorithms.

Fig. [5](#_bookmark25) obtains the box diagrams of the four objective function values in the model as a way of observing the distribution of the overall solu- tions of all algorithms solving the I-MaOWMUE model, which includes



**Fig. 4.** Trends in the search process for each objective value.



**Fig. 5.** Box diagrams of the four algorithmic solutions for each objective function.

the outliers, maximum, 1/4 and 3/4 distributions, mean, median, and minimum values of the overall solutions. From Fig. [5](#_bookmark25), it can be seen that among the four objective values obtained by each algorithm, the per- formance of the overall solution of the MI-MaOEA algorithm is better than the other algorithms, which indicates that the optimization effect of the MI-MaOEA algorithm is better. The results in Fig. [5](#_bookmark25)(a), (b), and

(d) show that the MI-MaOEA algorithm has more solution distributions in terms of migration delay, maximum completion time, and load bal- ancing, which is because we use a dual-migration selection strategy to provide a new direction of choice for decision-making schemes in the process of non-dominated sorting. Fig. [5](#_bookmark25)(c) shows that the MI-MaOEA algorithm has a similar overall solution distribution to DI-μMOGA and II-MOEA in terms of the energy consumption objective, but the overall solution is optimal for each value. The effect of the variational operator makes the solution have outliers, causing it to deviate from the over- all solution. In conclusion, the proposed MI-MaOEA algorithm provides the best solution set optimization compared to other algorithms.

# Conclusion

In this paper, we consider the problem of computing migration in an uncertain environment and transform the uncertain environmental fac- tors into interval parameters, while taking into account the relevance of the tasks and constraints such as server computing power, capacity limitations, and service scope, the task migration strategy and real-time priority scheduling strategy are proposed to achieve a rational alloca- tion of resources and fast response to dependent tasks, and the four optimization objectives of migration delay, maximum completion time, energy consumption, and load balancing are jointly optimized, and the I-MaOWMUE model is established. For this reason, this paper designs the MI-MaOEA algorithm, which uses interval confidence to represent the interval dominance relationship and serves as an important basis for matching selection, which in turn ensures the convergence of the so- lution, and utilizes objective-value-dominated hierarchical sorting and dual-migration selection strategies to ensure diversity of solutions. To evaluate the effectiveness of MI-MaOEA on the I-MaOWMUE model, MI- MaOEA is compared with three algorithms, DI-μMOGA, II-MOEA, and InMaOEA, respectively. The solution set shows advantages in the mean, maximum, and minimum values and has a better convergence rate for the four optimization objectives. The solutions have a better distribu- tion and maintain the best performance, which can provide a better migration solution for decision-makers.

In future work, we will focus on privacy and security issues in migra-

tion scenarios by using encrypted communication and authentication to achieve reliable transmission of tasks and prevent malicious users from accessing the private information of ordinary users. In addition, we will extend the constructed interval optimization model by introducing new uncertainty factors that are consistent with migration scenarios to pro- vide feasible resource allocation schemes for IoT systems and introduce dynamic factors to make our model more flexible to adapt to dynami- cally changing computing environments in terms of user requirements and resources.

# Declaration of competing interest

No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all authors for publication.

# Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under [Grant No.61806138]; in part by the China University Industry-University-Research Collaborative Innovation Fund (Future Network Innovation Research and Application Project) [Grant 2021FNA04014]; in part by the Key R&D program of Shanxi Province, under Grant No. 202202020101012; in part by Taiyuan University of

Science and Technology Scientific Research Initial Funding (TYUST SRIF), No. 20232087; and in part by the Shanxi University Science and Technology Innovation Funding, No. 2023L177.

# References

1. [Islam MM, Nooruddin S, Karray F, Muhammad G. Internet of things: device capa-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC4CA4238A0B923820DCC509A6F75849Bs1) [bilities, architectures, protocols, and smart applications in healthcare domain. IEEE](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC4CA4238A0B923820DCC509A6F75849Bs1) [Int Things J 2022;10:3611–41.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC4CA4238A0B923820DCC509A6F75849Bs1)
2. [Bai J, Huang G, Zhang S, Zeng Z, Liu A. Ga-dctsp: an intelligent active data process-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC81E728D9D4C2F636F067F89CC14862Cs1) [ing scheme for uav-enabled edge computing. IEEE Int Things J 2022;10:4891–906.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC81E728D9D4C2F636F067F89CC14862Cs1)
3. [Kong X, Wu Y, Wang H, Xia F. Edge computing for internet of everything: a survey.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibECCBC87E4B5CE2FE28308FD9F2A7BAF3s1) [IEEE Int Things J 2022;9:23472–85.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibECCBC87E4B5CE2FE28308FD9F2A7BAF3s1)
4. [Wang H, Lv T, Lin Z, Zeng J. Energy-delay minimization of task migration based](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA87FF679A2F3E71D9181A67B7542122Cs1) [on game theory in mec-assisted vehicular networks. IEEE Trans Veh Technol](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA87FF679A2F3E71D9181A67B7542122Cs1) [2022;71:8175–88.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA87FF679A2F3E71D9181A67B7542122Cs1)
5. [Liang Z, Liu Y, Lok T-M, Huang K. Multi-cell mobile edge computing: joint service](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibE4DA3B7FBBCE2345D7772B0674A318D5s1) [migration and resource allocation. IEEE Trans Wirel Commun 2021;20:5898–912.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibE4DA3B7FBBCE2345D7772B0674A318D5s1)
6. [Liu C, Tang F, Hu Y, Li K, Tang Z, Li K. Distributed task migration optimization](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1679091C5A880FAF6FB5E6087EB1B2DCs1) [in mec by extending multi-agent deep reinforcement learning approach. IEEE Trans](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1679091C5A880FAF6FB5E6087EB1B2DCs1) [Parallel Distrib Syst 2020;32:1603–14.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1679091C5A880FAF6FB5E6087EB1B2DCs1)
7. [Lin R, Xie T, Luo S, Zhang X, Xiao Y, Moran B, et al. Energy-eﬃcient computation](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib8F14E45FCEEA167A5A36DEDD4BEA2543s1) [oﬄoading in collaborative edge computing. IEEE Int Things J 2022;9:21305–22.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib8F14E45FCEEA167A5A36DEDD4BEA2543s1)
8. [Ji T, Luo C, Yu L, Wang Q, Chen S, Thapa A, et al. Energy-eﬃcient computation](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC9F0F895FB98AB9159F51FD0297E236Ds1) [oﬄoading in mobile edge computing systems with uncertainties. IEEE Trans Wirel](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC9F0F895FB98AB9159F51FD0297E236Ds1) [Commun 2022;21:5717–29.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC9F0F895FB98AB9159F51FD0297E236Ds1)
9. [Chen J, Yang Y, Wang C, Zhang H, Qiu C, Wang X. Multitask oﬄoading strategy](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib45C48CCE2E2D7FBDEA1AFC51C7C6AD26s1) [optimization based on directed acyclic graphs for edge computing. IEEE Int Things](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib45C48CCE2E2D7FBDEA1AFC51C7C6AD26s1) [J 2021;9:9367–78.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib45C48CCE2E2D7FBDEA1AFC51C7C6AD26s1)
10. [Yao S, Wang M, Qu Q, Zhang Z, Zhang Y-F, Xu K, et al. Blockchain-empowered](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD3D9446802A44259755D38E6D163E820s1) [collaborative task oﬄoading for cloud-edge-device computing. IEEE J Sel Areas](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD3D9446802A44259755D38E6D163E820s1) [Commun 2022;40:3485–500.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD3D9446802A44259755D38E6D163E820s1)
11. [Maleki EF, Mashayekhy L, Nabavinejad SM. Mobility-aware computation of-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6512BD43D9CAA6E02C990B0A82652DCAs1) [floading in edge computing using machine learning. IEEE Trans Mob Comput](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6512BD43D9CAA6E02C990B0A82652DCAs1) [2021;22:328–40.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6512BD43D9CAA6E02C990B0A82652DCAs1)
12. [Chen R, Wang X. Maximization of value of service for mobile collaborative comput-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC20AD4D76FE97759AA27A0C99BFF6710s1) [ing through situation aware task oﬄoading. IEEE Trans Mob Comput 2021.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC20AD4D76FE97759AA27A0C99BFF6710s1)
13. [Xia X, Chen F, Grundy J, Abdelrazek M, Jin H, He Q. Constrained app data caching](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC51CE410C124A10E0DB5E4B97FC2AF39s1) [over edge server graphs in edge computing environment. IEEE Trans Serv Comput](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC51CE410C124A10E0DB5E4B97FC2AF39s1) [2021;15:2635–47.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC51CE410C124A10E0DB5E4B97FC2AF39s1)
14. [Tao O, Chen X, Zhou Z, Li L, Tan X. Adaptive user-managed service placement for](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibAAB3238922BCC25A6F606EB525FFDC56s1) [mobile edge computing via contextual multi-armed bandit learning. IEEE Trans Mob](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibAAB3238922BCC25A6F606EB525FFDC56s1) [Comput 2021.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibAAB3238922BCC25A6F606EB525FFDC56s1)
15. [Luo Q, Li C, Luan TH, Shi W. Minimizing the delay and cost of computation oﬄoad-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib9BF31C7FF062936A96D3C8BD1F8F2FF3s1) [ing for vehicular edge computing. IEEE Trans Serv Comput 2021;15:2897–909.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib9BF31C7FF062936A96D3C8BD1F8F2FF3s1)
16. [Zakaryia SA, Ahmed SA, Hussein MK. Evolutionary oﬄoading in an edge environ-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC74D97B01EAE257E44AA9D5BADE97BAFs1) [ment. Egypt Inform J 2021;22:257–67.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC74D97B01EAE257E44AA9D5BADE97BAFs1)
17. [Liu J, Xiong K, Ng DWK, Fan P, Zhong Z, Letaief KB. Max-min energy balance](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib70EFDF2EC9B086079795C442636B55FBs1) [in wireless-powered hierarchical fog-cloud computing networks. IEEE Trans Wirel](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib70EFDF2EC9B086079795C442636B55FBs1) [Commun 2020;19:7064–80.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib70EFDF2EC9B086079795C442636B55FBs1)
18. [Do-Duy T, Van Huynh D, Dobre OA, Canberk B, Duong TQ. Digital twin-aided intelli-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6F4922F45568161A8CDF4AD2299F6D23s1) [gent oﬄoading with edge selection in mobile edge computing. IEEE Wirel Commun](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6F4922F45568161A8CDF4AD2299F6D23s1) [Lett 2022;11:806–10.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6F4922F45568161A8CDF4AD2299F6D23s1)
19. [Ding Y, Li K, Liu C, Li K. A potential game theoretic approach to computation of-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1F0E3DAD99908345F7439F8FFABDFFC4s1) [floading strategy optimization in end-edge-cloud computing. IEEE Trans Parallel](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1F0E3DAD99908345F7439F8FFABDFFC4s1) [Distrib Syst 2021;33:1503–19.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1F0E3DAD99908345F7439F8FFABDFFC4s1)
20. [Kim T, Sathyanarayana SD, Chen S, Im Y, Zhang X, Ha S, et al. Modems: opti-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib98F13708210194C475687BE6106A3B84s1) [mizing edge computing migrations for user mobility. IEEE J Sel Areas Commun](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib98F13708210194C475687BE6106A3B84s1) [2022;41:675–89.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib98F13708210194C475687BE6106A3B84s1)
21. [Xu X, Yao L, Bilal M, Wan S, Dai F, Choo K-KR. Service migration across edge devices](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib3C59DC048E8850243BE8079A5C74D079s1) [in 6g-enabled internet of vehicles networks. IEEE Int Things J 2021;9:1930–7.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib3C59DC048E8850243BE8079A5C74D079s1)
22. [Li S, Li C, Huang Y, Jalaian BA, Hou YT, Lou W. Enhancing resilience in mo-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibB6D767D2F8ED5D21A44B0E5886680CB9s1) [bile edge computing under processing uncertainty. IEEE J Sel Areas Commun](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibB6D767D2F8ED5D21A44B0E5886680CB9s1) [2023;41:659–74.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibB6D767D2F8ED5D21A44B0E5886680CB9s1)
23. [Ghoorchian S, Maghsudi S. Multi-armed bandit for energy-eﬃcient and delay-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib37693CFC748049E45D87B8C7D8B9AACDs1) [sensitive edge computing in dynamic networks with uncertainty. IEEE Trans Cogn](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib37693CFC748049E45D87B8C7D8B9AACDs1) [Commun Netw 2020;7:279–93.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib37693CFC748049E45D87B8C7D8B9AACDs1)
24. [Xu X, Huang Q, Zhu H, Sharma S, Zhang X, Qi L, et al. Secure service oﬄoading for](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1FF1DE774005F8DA13F42943881C655Fs1) [internet of vehicles in sdn-enabled mobile edge computing. IEEE Trans Intell Transp](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1FF1DE774005F8DA13F42943881C655Fs1) [Syst 2020;22:3720–9.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1FF1DE774005F8DA13F42943881C655Fs1)
25. [Xue F, Hai Q, Gong Y, You S, Cao Y, Tang H. Rvea-based multi-objective workflow](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib8E296A067A37563370DED05F5A3BF3ECs1) [scheduling in cloud environments. Int J Bio-Inspir Comput 2022;20:49–57.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib8E296A067A37563370DED05F5A3BF3ECs1)
26. [Dong T, Zhou L, Chen L, Song Y, Tang H, Qin H. A hybrid algorithm for workflow](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib4E732CED3463D06DE0CA9A15B6153677s1) [scheduling in cloud environment. Int J Bio-Inspir Comput 2023;21:48–56.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib4E732CED3463D06DE0CA9A15B6153677s1)
27. [Chen H, Qin W, Wang L. Task partitioning and oﬄoading in iot cloud-edge collabo-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib02E74F10E0327AD868D138F2B4FDD6F0s1) [rative computing framework: a survey. J Cloud Comput 2022;11:86.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib02E74F10E0327AD868D138F2B4FDD6F0s1)
28. [He X, Zheng J, He Q, Dai H, Liu B, Dou W, et al. Online computation oﬄoading for](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib33E75FF09DD601BBE69F351039152189s1) [deadline-aware tasks in edge computing. Wirel Netw 2022:1–20.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib33E75FF09DD601BBE69F351039152189s1)
29. [Sun J, Yin L, Zou M, Zhang Y, Zhang T, Zhou J. Makespan-minimization workflow](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6EA9AB1BAA0EFB9E19094440C317E21Bs1) [scheduling for complex networks with social groups in edge computing. J Syst Archit](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6EA9AB1BAA0EFB9E19094440C317E21Bs1) [2020;108:101799.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6EA9AB1BAA0EFB9E19094440C317E21Bs1)
30. [Huang B, Xiang Y, Yu D, Wang J, Li Z, Wang S. Reinforcement learning for security-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib34173CB38F07F89DDBEBC2AC9128303Fs1) [aware workflow application scheduling in mobile edge computing. Secur Commun](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib34173CB38F07F89DDBEBC2AC9128303Fs1) [Netw 2021;2021:1–13.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib34173CB38F07F89DDBEBC2AC9128303Fs1)
31. [Wang S, Ma D, Ren Z, Qu Y, Wu M. An adaptive multi-objective particle swarm](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC16A5320FA475530D9583C34FD356EF5s1) [optimisation algorithm based on fitness distance to streamline repository. Int J Bio-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC16A5320FA475530D9583C34FD356EF5s1) [Inspir Comput 2022;20:209–19.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibC16A5320FA475530D9583C34FD356EF5s1)
32. [Das G. Techno-economic analysis of novel multi-objective soft computing technique.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6364D3F0F495B6AB9DCF8D3B5C6E0B01s1) [Int J Bio-Inspir Comput 2022;20:172–82.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6364D3F0F495B6AB9DCF8D3B5C6E0B01s1)
33. [Xiao S, Wang W, Wang H, Huang Z. A new multi-objective artificial bee colony](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib182BE0C5CDCD5072BB1864CDEE4D3D6Es1) [algorithm based on reference point and opposition. Int J Bio-Inspir Comput](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib182BE0C5CDCD5072BB1864CDEE4D3D6Es1) [2022;19:18–28.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib182BE0C5CDCD5072BB1864CDEE4D3D6Es1)
34. [Zhao T, Wu L, Wu D, Li J, Cui Z. Multi-factor evolution for large-scale multi-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibE369853DF766FA44E1ED0FF613F563BDs1) [objective cloud task scheduling. KSII Trans Int Inf Syst 2023;17.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibE369853DF766FA44E1ED0FF613F563BDs1)
35. [Wu L, Wu D, Zhao T, Cai X, Xie L. Dynamic multi-objective evolutionary algorithm](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1C383CD30B7C298AB50293ADFECB7B18s1) [based on knowledge transfer. Inf Sci 2023;636:118886.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib1C383CD30B7C298AB50293ADFECB7B18s1)
36. [Cui Z, Xue Z, Fan T, Cai X, Zhang W. A many-objective evolutionary algorithm](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib19CA14E7EA6328A42E0EB13D585E4C22s1) [based on constraints for collaborative computation oﬄoading. Swarm Evol Comput](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib19CA14E7EA6328A42E0EB13D585E4C22s1) [2023;77:101244.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib19CA14E7EA6328A42E0EB13D585E4C22s1)
37. [Cui Z, Wen J, Lan Y, Zhang Z, Cai J. Communication-eﬃcient federated recommen-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA5BFC9E07964F8DDDEB95FC584CD965Ds1) [dation model based on many-objective evolutionary algorithm. Expert Syst Appl](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA5BFC9E07964F8DDDEB95FC584CD965Ds1) [2022;201:116963.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA5BFC9E07964F8DDDEB95FC584CD965Ds1)
38. [Cui Z, Zhang Z, Hu Z, Geng S, Chen J. A many-objective optimization based in-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA5771BCE93E200C36F7CD9DFD0E5DEAAs1) [telligent high performance data processing model for cyber-physical-social systems.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA5771BCE93E200C36F7CD9DFD0E5DEAAs1) [IEEE Trans Netw Sci Eng 2021;9:3825–34.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA5771BCE93E200C36F7CD9DFD0E5DEAAs1)
39. [Cui Z, Li B, Lan Z, Xu Y. Many-objective evolutionary algorithm based on three-way](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD67D8AB4F4C10BF22AA353E27879133Cs1) [decision. Egypt Inform J 2023;24:100388.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD67D8AB4F4C10BF22AA353E27879133Cs1)
40. [Bozorgchenani A, Mashhadi F, Tarchi D, Monroy SAS. Multi-objective computation](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD645920E395FEDAD7BBBED0ECA3FE2E0s1) [sharing in energy and delay constrained mobile edge computing environments. IEEE](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD645920E395FEDAD7BBBED0ECA3FE2E0s1) [Trans Mob Comput 2020;20:2992–3005.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD645920E395FEDAD7BBBED0ECA3FE2E0s1)
41. [Han D, Du W, Jin Y, Du W, Yu G. A fuzzy constraint handling technique for](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib3416A75F4CEA9109507CACD8E2F2AEFCs1) [decomposition-based constrained multi-and many-objective optimization. Inf Sci](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib3416A75F4CEA9109507CACD8E2F2AEFCs1) [2022;597:318–40.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib3416A75F4CEA9109507CACD8E2F2AEFCs1)
42. [Dai J, Wang Z, Huang W. Interval-valued fuzzy discernibility pair approach for](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA1D0C6E83F027327D8461063F4AC58A6s1) [attribute reduction in incomplete interval-valued information systems. Inf Sci](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA1D0C6E83F027327D8461063F4AC58A6s1) [2023;642:119215.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibA1D0C6E83F027327D8461063F4AC58A6s1)
43. [Jin Y, Zhang Z, Xie L, Cui Z. Decomposition-based interval multi-objective evolu-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib17E62166FC8586DFA4D1BC0E1742C08Bs1) [tionary algorithm with adaptive adjustment of weight vectors and neighborhoods.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib17E62166FC8586DFA4D1BC0E1742C08Bs1) [Egypt Inform J 2023;24:100405.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib17E62166FC8586DFA4D1BC0E1742C08Bs1)
44. [He Q, He Z, Duan S, Zhong Y. Multi-objective interval portfolio optimization mod-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibF7177163C833DFF4B38FC8D2872F1EC6s1) [eling and solving for margin trading. Swarm Evol Comput 2022;75:101141.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibF7177163C833DFF4B38FC8D2872F1EC6s1)
45. [Liu G, Liu S. Direct method for uncertain multi-objective optimization based on](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6C8349CC7260AE62E3B1396831A8398Fs1) [interval non-dominated sorting. Struct Multidiscip Optim 2020;62:729–45.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib6C8349CC7260AE62E3B1396831A8398Fs1)
46. [Zhang Z, Zhao M, Wang H, Cui Z, Zhang W. An eﬃcient interval many-objective](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD9D4F495E875A2E075A1A4A6E1B9770Fs1) [evolutionary algorithm for cloud task scheduling problem under uncertainty. Inf Sci](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD9D4F495E875A2E075A1A4A6E1B9770Fs1) [2022;583:56–72.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bibD9D4F495E875A2E075A1A4A6E1B9770Fs1)
47. [Gong D-w, Qin N-n, Sun X-y. Evolutionary algorithms for multi-objective optimiza-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib67C6A1E7CE56D3D6FA748AB6D9AF3FD7s1) [tion problems with interval parameters. In: 2010 IEEE fifth international confer-](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib67C6A1E7CE56D3D6FA748AB6D9AF3FD7s1) [ence on bio-inspired computing: theories and applications (BIC-TA). IEEE; 2010.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib67C6A1E7CE56D3D6FA748AB6D9AF3FD7s1)

[p. 411–20.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib67C6A1E7CE56D3D6FA748AB6D9AF3FD7s1)

1. [Juve G, Chervenak A, Deelman E, Bharathi S, Mehta G, Vahi K. Characterizing and](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib642E92EFB79421734881B53E1E1B18B6s1) [profiling scientific workflows. Future Gener Comput Syst 2013;29:682–92.](http://refhub.elsevier.com/S1110-8665(23)00074-9/bib642E92EFB79421734881B53E1E1B18B6s1)