Multi-Agent Collaborative Search Algorithm with Adaptive Scalarization Function and its Application in Aerospace Multi-objective Optimization Problems

Li Cao[1,2], Ziyu Song[3,*], Yirui Wang[2], Bo Zhang[4], Wei Li[5]
[1]*School of Computer, China University of Geosciences, Wuhan 430074, China*
[2]*Department of Mechanical & Aerospace Engineering, University of Strathclyde, Glasgow, UK*
[3]*School of Business, Sichuan University, Chengdu, China*
[4]*SLB Houston Digital Technology Center, Houston, Texas, USA*
[5]*Faculty of Engineering University of Nottingham, Nottingham, UK. NG7 2TU*
Cug_Li.Cao@outlook.com, cupszy@163.com, yirui.wang@strath.ac.uk, bzhang26@slb.com,wei.li7@nottingham.ac.uk

***Abstract*—Multi-Agent Collaborative Search Algorithm (MACS) is an effective method in handling multi-objective optimization problems. It is also regularly used in solving aerospace multi-objective optimization problems and demonstrates competitive problem-solving capabilities. The algorithm is a decomposition-based method. Tchebycheff scalarization function and dominant relationship are mixed to select individuals. The mixture with fixed p value represents constant characteristics. It may fail to meet the requirements of different stages of evolutionary algorithm. In order to prioritize convergence in early stages and diversity in later stages which may lead to final results with better quality, a new individual quality measurement parameter which adaptively emphasizes different features is introduced. To appropriately adjust p, reinforcement learning method (Upper Confidence Bound, UCB) is utilized to select p on the base of the new parameter. Building upon these improvements, a new algorithm Multi-Agent Collaborative Search Algorithm with Adaptive Scalarization Function (MACS-AS) is proposed. Two standard benchmark sets and two aerospace multi-objective problems are selected to evaluate algorithm performance. The new algorithm is compared with MACS and shows competitive results.**

***Index Terms*—Multi-objective optimization, scalarization function, evolutionary algorithm, upper confidence bound algorithm, aerospace optimization**

I. INTRODUCTION

Multi-objective optimization problems are common in engineering field. Conflicting objectives need to be optimized simultaneously in solving those problems. They have the following form [1]:

$$\min_{\boldsymbol{x}\in\boldsymbol{D}} \boldsymbol{f}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is the decision vector. $\boldsymbol{D}$ is a hyperrectangle which has the following form [1]:

$$\boldsymbol{D} = \{x_j | x_j \in [b_j^l, b_j^u] \subseteq \mathbb{R}, j = 1, \ldots, n\} \tag{2}$$

n is the dimension of decision space. b_j^l and b_j^u are upper and lower limit of jth dimension of decision space separately. The mathematical formulation for objective vector is as follows:

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T \tag{3}$$

m is the dimension of objective space. $\boldsymbol{f}$ is the objective vector. As multiple objectives are conflicting, the final results for those problems are a set of globally optimal individuals. Their related values in decision space are named as Pareto set. And their related values in objective space are named as Pareto front [2].

Evolutionary algorithms (EAs) are commonly used to solve such problems [3]. Recently, a evolutionary algorithm, Multi-Agent Collaborative Search algorithm (MACS) [2] is utilized to tackle multi-objective standard benchmarks and aerospace problems. It's an efficient algorithm for multi-objective problems solving but also has shortcoming. Tchebycheff scalarization function [4] is employed in MACS. But this method doesn't consider evolution phase and the focus on convergence and diversity is maintained uniformly in all stages. In previous research, convergence is always the primary consideration in early stage and diversity tends to be prioritized lately [5]. It can improve the quality of final solutions [5].

In this paper, a new algorithm, MACS with Adaptive Scalarization Function (MACS-AS) is proposed. The major contributions of this work can be summarized as follows: To achieve a better balance between convergence and diversity which leads to better final results, a new individual quality measurement parameter which considers both the elements is introduced. The new parameter is combined with reinforcement learning method to adjust the scalarization function. Finally, the new scalarization function is used in MACS as MACS-AS.

Then we compare MACS-AS and MACS on two sets of standard benchmarks. Additionally, the two algorithms are evaluated on two real aerospace multi-objective problems. The new method shows competitive performance.

The paper is organized as follows: Section II is about

* Corresponding author.

existing problems and background knowledge. Section III presents the details of the new algorithm. The experimental results and related discussions are in Section IV. Finally, the conclusions are in Section V.

Unfortunately, we cannot share our code as we have agreements with collaborators. The code contains sensitive information that requires permission from our collaborators before code release.

II. RELATED WORK

First, this section demonstrates the background knowledge of MACS. Subsequently, foundational knowledge of the scalarization function is described. Finally, the limitation on solving multi-objective optimization problems for MACS is illustrated.

A. MACS algorithm

MACS [6] is an effective algorithm for solving multi-objective optimization problems. The initial version [6] is proposed in 2005, the core principle of this algorithm is combining the advantages of the population-based search in evolutionary algorithms with local convergence mechanisms. The most classic version of MACS, which is introduced in 2019, is MACS2.1 [1]. In MACS2.1, the search is based on differential evolution and local search. The local search is executed by the combination of pattern search, differential evolution and an inertia operator. In subsequent sections, discussions focus on this version and MACS is employed to represent MACS2.1.

The details of MACS are illustrated in Algorithm 1 [1].

Algorithm 1 MACS

1: Initialise weight vectors set $\boldsymbol{w}$, population $\boldsymbol{P}$, empty discarded pool $\boldsymbol{P}_d$ and global archive $\boldsymbol{A}$
2: **while** $n\_fun\_eval < max\_fun\_eval$ **do**
3: Perform individualistic actions on $\boldsymbol{P}$
4: Update archive $\boldsymbol{A}$ with energy based archiving strategy
5: Perform social actions on $\boldsymbol{P}$
6: Update archive $\boldsymbol{A}$ with energy based archiving strategy
7: Update n_fun_eval
8: **end while**

B. Individual updating in MACS

In line 3 and 5 of Algorithm 1, the old individual and new generated one will be compared by Tchebycheff scalarization function and dominant relationship [1]. Tchebycheff scalarization function [4] is a commonly used function in this kind of algorithms for quantifying the quality of one individual. Given new generated individial $\boldsymbol{I}_{new}$, its corresponding old individial $\boldsymbol{I}_{old}$, their related ideal point $\boldsymbol{z}^*$ and weight vector $\boldsymbol{w}$, the new individual's Tchebycheff scalarization function value $Tch(\boldsymbol{I}_{new})$ can be calculated as follows [4]:

$$
\begin{aligned}
& g^{tc}(\boldsymbol{I}_{new}|\boldsymbol{w},\boldsymbol{z}^*) = \max_{1\le i\le m}\{\lambda_i|f_i(\boldsymbol{I}_{new}) - z_i^*|\} \\
& \lambda_i = (\frac{1}{w_i})^p
\end{aligned} \tag{4}
$$

where is $\boldsymbol{z}^* = [z_1^*, z_2^*, \dots, z_m^*]^T$ is the reference point. In the case of the following Equation 5 [7], $\boldsymbol{I}_{new}$ dominates $\boldsymbol{I}_{old}$ (namely, $\boldsymbol{I}_{new} \prec \boldsymbol{I}_{old}$).

$$
\begin{aligned}
\text{if} \quad & \forall i \in \{1,\dots,m\},\ f_i(\boldsymbol{I}_{new}) \le f_i(\boldsymbol{I}_{old}) \\
\text{and} \quad & \exists j \in \{1,\dots,m\},\ f_j(\boldsymbol{I}_{new}) < f_j(\boldsymbol{I}_{old})
\end{aligned} \tag{5}
$$

When the following Equation 6 holds, $\boldsymbol{I}_{new}$ will replace $\boldsymbol{I}_{old}$.

$$
Tch(\boldsymbol{I}_{new}) < Tch(\boldsymbol{I}_{old}) \quad or \quad \boldsymbol{I}_{new} \prec \boldsymbol{I}_{old} \tag{6}
$$

C. Background knowledge for scalarization function

Tchebycheff scalarization in MACS belongs to the family of L_p scalarizing methods with $p = \infty$. Given an individial $\boldsymbol{I}$, its related ideal point $\boldsymbol{z}^*$ and weight vector $\boldsymbol{w}$, the family can be summarized as Equation 7 [8]:

$$
\begin{aligned}
& g^{\text{wd}}(\boldsymbol{I} \mid \boldsymbol{w}, p, \boldsymbol{z}^*) = \left(\sum_{i=1}^{m} \lambda_i (f_i(\boldsymbol{I}) - z_i^\star)^p\right)^{\frac{1}{p}}, \quad p \ge 1 \\
& \lambda_i = (\frac{1}{w_i})^p
\end{aligned} \tag{7}
$$

Scalarization functions with different p values have distinctly different search characteristics. The specific reasons can be summarized as follows:

1) p value decides the probability of finding a better solution and obtaining all the Pareto optimal solutions [9].
2) By analyzing the contour curves of scalarization functions with different p values, the conclusion can be drawn that the probability of finding a better solution tends to decreases with the p value increasing [8], [9].
3) The opening angle influences the scalarization function's diversity maintenance ability [10]. Scalarization functions with distinct p values have different opening angles and exhibit different characteristics in diversity maintaining.
4) When considering convergence and diversity separately, scalarization functions with different p values exhibit distinct performance across these two aspects [8]. The decomposition-based methods partition the optimization problem into many sub-problems [4]. As each sub-problem corresponds uniquely to one individual, observing the evolutionary process of one individual with different p values can provide insights for all the population. Let $\boldsymbol{I}$, $\boldsymbol{z}^*$, $\boldsymbol{w}$ and $\boldsymbol{V}$ represent one individual, its related ideal point, its related weight vector and the orthogonal projection of individual $\boldsymbol{I}$ onto $\boldsymbol{w}$, respectively. d_1 is the distance between $\boldsymbol{V}$ and $\boldsymbol{z}^*$. d_2 is the distance between $\boldsymbol{V}$ and $\boldsymbol{I}$. For individual $\boldsymbol{I}$, d_1 and d_2 can be used to measure its convergence and diversity correspondingly [4]. Our goal is to simultaneously minimize both of them.

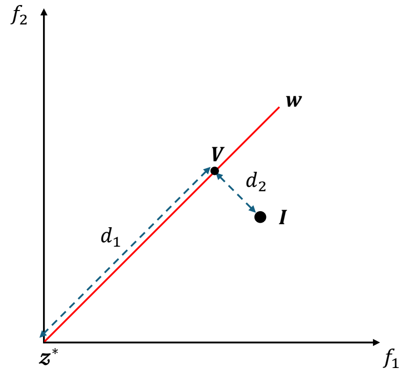

Fig. 1. The d_1 and d_2 parameters.

As previous research shows [8], the Lp scalarizing methods appears to become more effective in reducing d_2 than d_1 with p increasing.

D. Limitation in MACS

To obtain better quality population, the varying criteria should be applied at different stages to filter solutions [5]. Typically, convergence is emphasized first, followed by diversity [5]. In MACS, the dominant relationship and Tchebycheff scalarization function with fixed $p = \infty$ are mixed in individual picking. There is a conflict between these two aspects which might lead to quality decrease. The reason will be elaborated in the following section.

Scalarization function is considered first. Although the previous section illustrates the relation between d_1 or d_2 and p values, the actual quantitative relationship between diversity or convergence and p remains to be explored. Furthermore, as the scalarization function is used after being combined with dominant relationship, it becomes increasingly difficult to analyze. However, it is evident that scalarization functions with different p values demonstrate divergent characteristics. Consequently, employing adaptive p in MACS to meet the requirement to prioritize different elements at various phases might bring improved performance.

III. The New Algorithm: MACS with Adaptive Scalarization Function

In the new algorithm, an adaptive scalarization function is embedded in MACS. The new algorithm is named as MACS with Adaptive Scalarization Function (MACS-AS) and the adaptive function should select appropriate p value to emphasis convergence or diversity at different stages.

As d_1 and d_2 are simple and effective quality measure parameters for one individual. In MACS-AS, a composite parameter integrating d_1 and d_2 is proposed to measure quality. It considers convergence and diversity simultaneously. Furthermore, the effect of p on the hybrid approach combining dominant relationship and scalarization function is difficult to analyze. Reinforcement learning enables agents to pick optimal actions without explicit modeling of action-reward relationship [11]. It balances exploration and exploitation to maximize long-term cumulative rewards [11]. Reinforcement learning is introduced to pick p with the composite parameter as the relation between individual quality and p is not clear. p value is the action and the composite parameter is the reward.

A. A composite individual quality measure parameter

The composite quality parameter has the following form:

$$Q(\boldsymbol{I} \mid \boldsymbol{w}, \boldsymbol{z}^*, n\_fun\_eval, max\_fun\_eval, C_1, C_2) = \\ (C_1 - (C_1 - C_2) \times \frac{n\_fun\_eval}{max\_fun\_eval})d_1 + d_2 \tag{8}$$

where C_1 and C_2 are weighting parameters controlling the bias towards convergence and diversity. With the process progresses, the weighting parameter imposed on convergence parameter (d_1) changes from C_1 to C_2. To emphasize convergence at early stage and diversity at late stage, $C_1{>}1$ and $C_2{<}1$ are appropriate setting. From the Equation 8, it can be observed that weighting parameters change linearly. Dynamically tune p value to minimize Q parameter throughout the evolution process yields the high quality final solutions.

B. Utilize Upper Confidence Bound to determine p value

More specifically, the evolutionary process for one sub-problem can be viewed as a multi-armed bandits problem [12]. It sequentially selects actions (p values) to maximize cumulative rewards (quality measure parameter, namely Q) over time. Upper Confidence Bound (UCB) algorithm [13]–[15] is an effective method in solving multi-armed bandits problems and has been applied in evolutionary algorithms to pick operators [12]. In MACS-AS, a p pool is set up and the UCB method is employed to pick p value from the pool. For clarity, we introduce the process with a flowchart as Figure 2. The process is based on previous research [12].

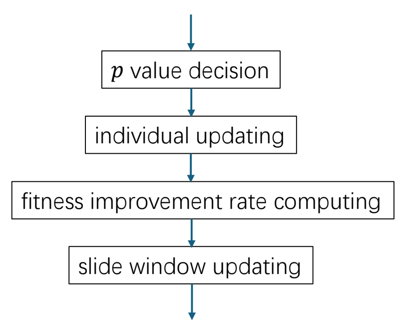

Fig. 2. UCB process.

1) p value decision: In the start of MACS-AS, an empty slide window for historical improvement is initialized for each individual. The slide window length is finite, since distant historical records have negligible influence on current choices [12]. For ith individual, its slide window is structured as $\boldsymbol{SW}_i$ in Figure 3. The first line of the window is the p value values chosen in the past and the second line is the related fitness improvement rate (FIR) [12]. Lower-right (t, $t-1$,...) indexes are iteration indexes and tth is the newest iteration.

The length of the window is predefined as $L_{\boldsymbol{SW}}$. Furthermore, a reward matrix $\boldsymbol{RM}_i$ has the structure as Figure 3 for reward recording. It's a $1 \times S_{pool}$ matrix if there are S_{pool} values in p pool and jth column's value in $\boldsymbol{RM}_i$ corresponds to the jth p. The matrix can be calculated as Equation 9.

$$RM_i(1,j) = \frac{Sum_i(1,j)}{\sum_{k=1}^{S_{pool}} Sum_i(1,k)} \tag{9}$$

The $Sum_i(1,j)$ represents the sum of the FIR in $\boldsymbol{SW}_i$ corresponding to the jth p. Meanwhile, if one p has no record in $\boldsymbol{SW}_i$, all the values in $\boldsymbol{RM}_i$ are set as zero for fairness. The pseudocode for reward assignment is shown in Algorithm 2.

Finally, for ith individual, the p which has the biggest value in Equation 10 is picked [12]. The pseudocode is in Algorithm 3. n_j is the the number of times that jth p appears in $\boldsymbol{SW}_i$ and C_3 is a predefined constant.

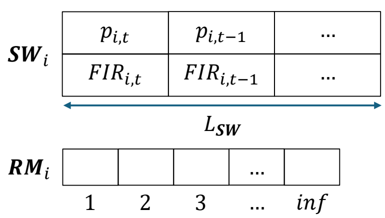

Fig. 3. Slide window and reward matrix.

Algorithm 2 Reward Assignment

1: Summarize the FIR values for each p in the sliding window $\boldsymbol{SW}_i$ as $Sum_i(1,j)$
2: **if** all the $Sum_i(1,j)$ are not zero **then**
3:     **for** $j \leftarrow 1$ **to** S_{pool} **do**
4:         $RM_i(1,j) = Sum_i(1,j) / \sum_{k=1}^{S_{pool}} Sum_i(1,k)$
5:     **end for**
6: **else**
7:     Set all the $RM_i(1,j)$ as zero
8: **end if**

$$opt = \underset{j=1,\ldots,S_{pool}}{\arg\max} \left(RM_i(1,j) + C_3 \times \sqrt{\frac{2 \times ln \sum_{k=1}^{S_{pool}} n_k}{n_j}} \right) \tag{10}$$

Algorithm 3 Choose p value

1: **if** exist $RM_i(1,j) == 0$ **then**
2:     randomly choose one p
3: **else**
4:     Choose the p which has the biggest opt in Equation 10
5: **end if**

---

2) individual updating: In social and individualistic actions parts, if new individual is better in adaptive scalarization function or dominant relationship, the old one will be replaced.

3) fitness improvement rate computing: Individual with lower Q exhibits better quality. For ith individual, the fitness improvement rate in tth iteration can be computed as Equation 11. The related fitness improvement rate instead of raw fitness improvement is utilized here as different problems have various range of raw improvement at different stages of optimization. The algorithm's robustness may be deteriorated with using of raw fitness improvement value [16].

$$FIR_{i,t} = \frac{Q_{i,t-1} - Q_{i,t}}{Q_{i,t-1}} \tag{11}$$

4) slide window updating: Once individual updating occurs, the fitness improvement rate and related p value insert into the slide window from the left. If the length of the window exceeds the maximum L_{SW}, the oldest record will be removed from the right.

C. Framework of the new algorithm

The process for new algorithm is in Algorithm 4. The difference is the using of adaptive scalarization function.

Algorithm 4 MACS-AS

1: Initialise weight vectors set $\boldsymbol{w}$, population $\boldsymbol{P}$, empty discarded pool $\boldsymbol{P}_d$ and global archive $\boldsymbol{A}$
2: **while** $n\_fun\_eval < max\_fun\_eval$ **do**
3:     Perform individualistic actions with adaptive scalarization function on $\boldsymbol{P}$
4:     Update archive $\boldsymbol{A}$ with energy based archiving strategy
5:     Perform social actions with adaptive scalarization function on $\boldsymbol{P}$
6:     Update archive $\boldsymbol{A}$ with energy based archiving strategy
7:     Update n_fun_eval
8: **end while**

IV. Experiment and Results

In this part, the benchmark instances evaluated in the experiments are illustrated, followed by a discussion of the results. Two standard benchmark sets and two aerospace multi-objective testing cases are used to test the performance of our new algorithm.

A. Standard benchmark

ZDT [17] and UF [18] standard benchmark sets are picked. They are multi-objective optimization cases.

B. Three impulsive manoeuvres

The goal of three impulsive manoeuvres (3imp) [1], [2] problem is to transfer a spacecraft from a circular Low Earth Orbit to a circular Geostationary orbit with three impulsive manoeuvres like Figure 4. The circular Low Earth Orbit has a radius of 7000km and the Geostationary orbit has a radius of 42000km. The objective space has two objectives. One is the total transfer time T and the other is the sum of the three impulses Δv_{tot}.

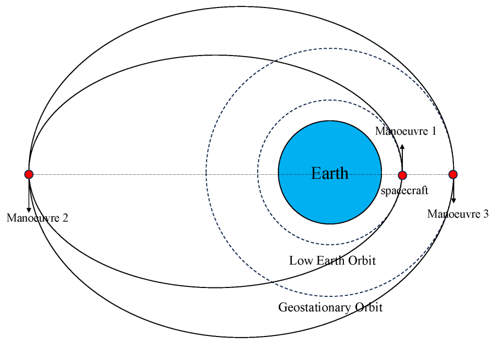

Fig. 4. 3imp problem [2].

C. Cassini problem

The aim of Cassini is launching a spacecraft from earth to land Saturn through a sequence of swing-by's with the planets: Venus, Venus, Earth, and Jupiter following [1], [2] Figure 5.

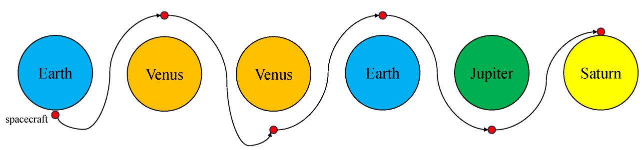

Fig. 5. Cassini problem [2].

The objectives are two comprehensive parameters and can be found in previous literature [19].

D. Experimental setting

The settings for those testing cases are in Table I. And the settings for the two algorithms are in Table II. Some parameter settings are inherited from previous research [8], [12].

TABLE I
SETTINGS FOR TESTING CASES.

Problem	Population size (N)	Number of objectives (M)	Size of global archive ($\|\boldsymbol{A}\|$)	Maximum fitness evaluation number
UF1-7	100	2	100	300000
UF8-10	91	3	91	273000
ZDT1-6	100	2	100	300000
Cassini	30	2	30	15000
3imp	30	2	30	15000

TABLE II
SETTINGS FOR ALGORITHMS.

Algorithms	Parameters
MACS	$Cr = 0.5$ $F = 0.5$
MACS-AS	$Cr = 0.5$ $F = 0.5$ $C_1 = 10$ $C_2 = 0.1$ $C_3 = 5$ [12] p value pool: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, inf]$ [8] $L_{SW} = 0.5N$ [12]

E. Results

To measure the performance of each algorithm, HV [20] and IGD [21] are used as the measurement parameters.

F. Results for standard testing cases

The result for ZDT and UF sets are in Table III and IV. The HV/IGD values of each solution set are calculated with a set of reference points which are generated in a famous open source platform [22]. Every instance was run for 30 times independently, and in the following bracket we listed the standard deviation of HV and IGD. The Wilcoxon rank sum test with a significance level of 0.05 is adopted to do the analysis. The marks "+", "−" and "=" mean that the result of another algorithm is significantly better, worse and similar to the result of MASC-AS. The best result for each problem are emphasize with "*".

The Table III and IV show new algorithm's competitive performance. The added adaptive scalarization function seems to achieve a balanced trade-off between convergence and diversity and enhance final solutions' quality.

G. Results for aerospace testing cases

The results for 3imp and Cassini are in Table V. The real Pareto front for those problems is difficult to locate. The no-dominated solutions in each run are collected as reference points for HV/IGD computing. Further, pure diversity (PD) [23] is calculated as a supplement. PD is pure diversity measuring parameter which doesn't need reference points for computation. Every instance ws run for 30 times independently, and in the following bracket we listed the standard deviation of HV, IGD and PD. The Wilcoxon rank sum test with a significance level of 0.05 is adopted to do the analysis. The marks "+", "−" and "=" mean that the result of another algorithm is significantly better, worse and similar to the result of MASC-AS. The best result for each problem are emphasize with "*".

Apart from the similar results on PD value of 3imp, MACS-AS demonstrates better performance than MACS in all cases.

TABLE III
THE RESULTS ON HV FOR ZDT AND UF SETS.

problem	MACS	MACS-AS
UF1	2.9922e-1 (4.66e-3) -	7.1816e-1 (1.14e-3)*
UF2	6.7306e-1 (2.59e-3) -	7.1040e-1 (2.42e-3)*
UF3	5.8486e-1 (2.24e-2) -	6.4000e-1 (2.26e-2)*
UF4	3.8751e-1 (2.58e-3) -	4.0302e-1 (1.30e-3)*
UF5	1.7315e-1 (1.39e-4) -	5.0540e-1 (8.68e-3)*
UF6	1.6414e-1 (7.63e-3) -	4.5122e-1 (9.84e-3)*
UF7	4.6542e-1 (4.46e-3) -	5.8009e-1 (7.24e-4)*
UF8	4.0436e-1 (4.66e-4) -	4.0569e-1 (2.97e-4)*
UF9	7.3898e-1 (3.72e-4) -	7.3959e-1 (5.53e-4)*
UF10	1.6664e-1 (1.24e-2) -	1.8051e-1 (1.24e-2)*
ZDT1	3.8903e-1 (2.16e-2) -	7.2053e-1 (4.31e-5)*
ZDT2	1.7355e-1 (0.00e+0) -	4.4491e-1 (5.43e-5)*
ZDT3	4.6237e-1 (1.28e-1) -	5.9992e-1 (4.29e-6)*
ZDT4	9.4328e-2 (4.29e-2) -	7.1602e-1 (2.50e-3)*
ZDT6	3.8874e-1 (7.16e-3) -	3.9865e-1 (8.65e-6)*
$+/-/=$	0/15/0	

V. CONCLUSION

In this paper, we try to improve the Multi-Agent Collaborative Search algorithm (MACS) with an adaptive scalariza-

TABLE IV
THE RESULTS ON IGD FOR ZDT AND UF SETS.

problem	MACS	MACS-AS
UF1	3.3288e-1 (2.74e-3) -	5.1650e-3 (6.68e-4)*
UF2	5.5015e-2 (2.95e-3) -	1.4033e-2 (2.66e-3)*
UF3	1.0735e-1 (1.98e-2) -	4.9922e-2 (1.51e-2)*
UF4	2.6332e-1 (1.10e-3) -	2.5669e-1 (1.09e-3)*
UF5	3.3588e-1 (6.88e-4) -	5.1307e-2 (6.28e-3)*
UF6	3.2852e-1 (2.10e-2) -	5.4969e-2 (5.79e-3)*
UF7	2.0124e-1 (3.12e-3) -	1.1876e-1 (2.04e-4)*
UF8	3.5047e-1 (3.75e-3) +*	3.5668e-1 (1.68e-4)
UF9	9.9295e-2 (1.40e-3) +*	1.0098e-1 (1.42e-4)
UF10	3.6870e-1 (3.52e-2) +*	3.8904e-1 (3.83e-2)
ZDT1	2.8474e-1 (1.32e-2) -	3.6120e-3 (1.51e-5)*
ZDT2	3.9412e-1 (1.69e-16) -	2.3011e-1 (8.76e-6)*
ZDT3	2.9033e-1 (4.58e-2) -	4.3457e-3 (1.27e-5)*
ZDT4	4.6392e-1 (6.51e-2) -	5.5871e-3 (1.61e-3)*
ZDT6	2.4674e-1 (1.05e-3) -	2.4582e-1 (2.69e-6)*
$+/-/=$	3/12/0	

TABLE V
THE RESULTS ON 3IMP AND CASSINI.

problem/ parameter	MACS	MACS-AS
3imp/HV	6.4201e-1 (6.49e-3) -	6.4796e-1 (5.04e-3)*
3imp/IGD	4.2711e-1 (1.37e-1) =	3.9434e-1 (1.61e-1)*
3imp/PD	4.6216e+4 (1.00e+4) =*	4.6165e+4 (1.08e+4)
Cassini/HV	6.9632e-1 (1.87e-2) =	7.0589e-1 (1.64e-2)*
Cassini/IGD	7.5367e+1 (2.11e+1) =	6.6707e+1 (2.71e+1)*
Cassini/PD	4.3870e+5 (1.37e+5) =	5.0456e+5 (1.90e+5)*
$+/-/=$	0/1/5	

tion function. The new algorithm is named as Multi-Agent Collaborative Search Algorithm with Adaptive Scalarization Function (MACS-AS). First, a new individual quality measure parameter is proposed. Next, reinforcement learning method (Upper Confidence Bound, UCB) is utilized to adjust the p value in scalarization function with the new quality measure parameter to improve the quality of final solutions.

Two standard benchmark sets and two aerospace multi-objective testing cases are picked for evaluating. In those cases, MACS-AS shows competitive performance.

In future work, we will try to search for appropriate parameter combinations of C_1, C_2, C_3 and L_{SW} on different problems. The execution details of the hybrid approach of scalarization function with different p and dominant relationship deserve further exploration.

REFERENCES

[1] L. A. Ricciardi and M. Vasile, "Improved archiving and search strategies for multi agent collaborative search," *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, pp. 435–455, 2019.

[2] L. Cao, M. Wang, M. Vasile, and G. Dai, "Multi agent collaborative search algorithm with adaptive weights," *Expert Systems*, vol. 41, no. 12, 2024.

[3] M. Méndez, D. A. Rossit, B. González, and M. Frutos, "Proposal and comparative study of evolutionary algorithms for optimum design of a gear system," *IEEE Access*, vol. 8, pp. 3482–3497, 2020.

[4] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[5] Y. Tian, C. He, R. Cheng, and X. Zhang, "A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 9, pp. 5880–5894, 2021.

[6] M. Vasile, "Robust mission design through evidence theory and multiagent collaborative search," *Annals of the New York Academy of Sciences*, vol. 1065, no. 1, pp. 152–173, 2005.

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

[8] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using pareto adaptive scalarizing methods," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 821–837, 2016.

[9] I. Giagkiozis and P. Fleming, "Methods for multi-objective optimization: An analysis," *Information Sciences*, vol. 293, pp. 338–350, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025514009074

[10] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 296–313, 2018.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[12] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2013.

[13] C. Li and H. Wang, "Asynchronous upper confidence bound algorithms for federated linear bandits," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 6529–6553.

[14] N. Manome, S. Shinohara, and U.-i. Chung, "Simple modification of the upper confidence bound algorithm by generalized weighted averages," *PLoS One*, vol. 20, no. 5, p. e0322757, 2025.

[15] L. Fan and P. W. Glynn, "The fragility of optimized bandit algorithms," *Operations Research*, vol. 73, no. 6, pp. 3173–3198, 2025.

[16] Á. Fialho, "Adaptive operator selection for optimization," Ph.D. dissertation, Université Paris Sud-Paris XI, 2010.

[17] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[18] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, S. Tiwari *et al.*, "Multiobjective optimization test instances for the cec 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, vol. 264, pp. 1–30, 2008.

[19] M. Vasile and F. Zuiani, "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, pp. 1211 – 1227, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:17060952

[20] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[21] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *International conference on evolutionary multi-criterion optimization*. Springer, 2015, pp. 110–125.

[22] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[23] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 6, pp. 1510–1522, 2017.