Robust Multi-Objective Optimization using Conditional Pareto Optimal Dominance

Seyedeh Zahra Mirjalili
School of Electrical Engineering and Computing
University of Newcastle
NSW 2308
seyedehzahra.mirjalili@uon.edu.au

Stephan Chalup
School of Electrical Engineering and Computing
University of Newcastle
NSW 2308
stephan.chalup@newcastle.edu.au

Seyedali Mirjalili
Centre for Artificial Intelligence Research and Optimisation
Torrens University Australia
Brisbane, Australia
ali.mirjalili@laureate.edu.au

Nasimul Noman
School of Electrical Engineering and Computing
University of Newcastle
NSW 2308
nasimul.noman@newcastle.edu.au

***Abstract*—Robust optimization of real-world problems is essential to reduce the significant negative impact of uncertainties and noises present in the environment. Uncertainties in the decision variables are often handled using explicit or implicit averaging methods, in which the fitness of a solution is judged based on the objective values of neighbouring solutions. Explicit averaging methods are highly reliable but require additional objective function evaluation, which can significantly increases the overall computational cost of an optimization process. On the other hand, implicit averaging techniques are computationally cheap, yet they suffer from low reliability since they use the history of search in a population-based optimization algorithm. This work proposes a conditional Pareto optimal dominance to improve the reliability of robust optimization methods that use implicit averaging methods. The proposed method is applied to Multi-Objective Particle Swarm Optimisation. Empirical study with a benchmark suite shows the benefit of the proposed conditional Pareto optimal dominance in locating robust solutions in multi-objective problems.**

Keywords—robust optimization; multi-objective particle swarm optimization, implicit averaging, uncertainties

I. Introduction

In recent years, optimization of real-world problems using heuristic algorithms have become very popular. In a wide range of fields, classical optimization algorithms are no longer able to handle the challenges involved in optimizing real-world problems. Some of these challenges are a large number of variables, a larger number of constraints, high computational cost, multiples objectives, and uncertainties.

This work tackles two of these difficulties: multiple objectives and uncertainties. In a single-objective optimization, there is one solution as the global optimum. Without the loss of generality, such problems are formulated as follows:

$$\begin{gathered} Minimize\colon f(\vec{x}) \\ Subject\ to\colon\ g_i(\vec{x}) \geq 0, i = 1,2,3,\dots,m \\ h_i(\vec{x}) = 0, i = 1,2,3,\dots,p \\ lb_i \leq x_i \leq ub_i, i = 1,2,3,\dots,n \end{gathered}$$

where $\vec{x} = \{x_1, x_2, x_3, \dots, x_n\}$ is a vector that stores *n* number of decision variables, *f* is an objective function, *m* shows the number of inequality constraints, *h* indicates the number of equality constraints, lb_i is the lower bound for the *i-th* decision variable, and ub_i is the upper bound for the *i-th* decision variable,

When solving such problems, solutions can be compared easily using relational operators. In case of using a population-based algorithms, solutions are typically compared in each generation to find the best estimation of the global optimum at any stage of optimization. The existence of one global solution allows such comparison and decision making during the optimization process.

A. Multiple objectives

In a multi-objective optimization problems, however, each solution is evaluated by multiple objectives. Therefore, we cannot use relational operators to find the best solution. Without the loss of generality, a multi-objective optimization problem is formulated as follows:

$$\begin{gathered} Minimize\colon F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_o(\vec{x})\} \\ Subject\ to\colon\ g_i(\vec{x}) \geq 0, i = 1,2,3,\dots,m \\ h_i(\vec{x}) = 0, i = 1,2,3,\dots,m \\ lb_i \leq x_i \leq ub_i, i = 1,2,3,\dots,n \end{gathered}$$

where *o* indicated the number of objectives and *m* shows the number of inequality constraints. The other notations remains the same as before.

To compare solutions in such multi-objective problems, Pareto optimality is used. Without the loss of generality, the essential definitions of Pareto optimality and Pareto optimal dominance for minimization problems are as follows:

Pareto Dominance ($\vec{x} \prec \vec{y}$):

$$\forall i \in \{1,2,3,\dots,o\}$$

$$\{f_i(\vec{x}) \leq f_i(\vec{y})\} \wedge \{\exists i \in 1,2,\dots,o:\ f_i(\vec{x}) < f_i(\vec{y})\}$$

where $\vec{x}, \vec{y}$ are two vectors with *o* objectives.

The definition shows that a solution dominates another if it shows less or equal objective values in all objectives and less in at least one of them.

Pareto Optimality: the solution $\vec{x}$ is called Pareto optimal if and only if:

$$\nexists \vec{y} \in S \mid \vec{y} \prec \vec{x}$$

where *S* represents the set of all solutions, and $\vec{x}, \vec{y}$ are two vectors with *o* objectives.

The definition shows that a solution is Pareto optimal if none of the other solutions dominates it.

Pareto Optimal Solutions Set:

$$PS := \{ \vec{x}, \vec{y} \in S \mid \nexists \vec{y} \prec \vec{x} \}$$

The definition shows that a Pareto optimal set includes all the non-dominates solutions.

Pareto Optimal Front:

$$\forall i \in \{1,2,3,\dots,o\}$$
$$PF: \{f_i(\vec{x}) \mid \vec{x} \in PS\}$$

The definition shows that a Pareto optimal front includes the objective values of the solutions in the Pareto optimal solution set.

When solving a multi-objective problem, we are looking for the Pareto optimal solutions set. This set represents the best trade-offs between the multiple objectives that we are trying to optimize.

The existence of multiple objectives is one of the challenges that should be addressed when solving real-world problems. Another challenge is the existence of uncertainties in the optimization system.

B. Uncertainties

When we are using computers to solve real-world problems, we often use simulators and similar tools to emulate the problem. Uncertainties may arise from other tools involved in an experiment or system including sensors, actuators, measurements, etc. Despite the accuracy of such tools, there are always deviations from what is considered in the simulation and what occurs in reality. Failure in considering them before, during, or after the optimization process might results in unexpected outcomes when the system is operating in an environment.

As discussed in the abstract, a robust optimization algorithm tries to minimize the negative impacts of such uncertainties. Uncertainties may occur in different components of an optimization system as follows:

- Inputs: in this type, the decision variables are fluctuated. The main source of such uncertainties are manufacturing or production imperfection.
- Outputs: in this type, the objectives face noises. The main source of such uncertainties is the use or simulators and digital devices.
- Constraints: in this type, the constraints that we consider to find desirable solution for an optimization system can be noisy in simulation.
- Operating conditions: in this type, the operating conditions that a system faces in real-world environment fluctuate.

There are different techniques to design robust algorithms for handing each of these uncertainties. In this work, uncertainties in the decision variables are considered. The problem formulation of a single-objective optimization problem when considering uncertainties in the parameters is as follows:

$$Minimize: f(\vec{x} + \vec{\delta})$$
$$Subject\ to:\ g_i(\vec{x} + \vec{\delta}) \geq 0, i = 1,2,3,\dots,m$$
$$h_i(\vec{x} + \vec{\delta}) = 0, i = 1,2,3,\dots,p$$
$$lb_i \leq x_i \leq ub_i, i = 1,2,3,\dots,n$$

where $\vec{\delta} = \{\delta_1, \delta_2, \delta_3, \dots, \delta_n\}$ is a vector representing the maximum level of deviations in each of the decision variables,

In case of robust multi-objective optimization, we are looking for optimal solutions that are fault tolerant. This means that the expected outputs of the optimization system do not change significantly in case of errors in the decision variables. The most ideal case is when the global optimum is the most robust solution. However, a solution that is the most robust is often one of the locally optimal solution in the search space.

A similar formulation can be written for multi-objective problems as follows:

$$Minimize: F(\vec{x} + \vec{\delta}) = \{f_1(\vec{x} + \vec{\delta}), f_2(\vec{x} + \vec{\delta}), \dots, f_o(\vec{x} + \vec{\delta})\}$$
$$Subject\ to:\ g_i(\vec{x} + \vec{\delta}) \geq 0, i = 1,2,3,\dots,m$$
$$h_i(\vec{x} + \vec{\delta}) = 0, i = 1,2,3,\dots,m$$
$$lb_i \leq x_i + \delta_i \leq ub_i, i = 1,2,3,\dots,n$$

where $f_j(\vec{x})$ is the *j-th* objective function and o indicates the number of objectives,

It was discussed above that a set of solution called Pareto optimal solutions set is the answer to a multi-objective problem. When considering uncertainties in the parameters, all or some of these solutions are no longer Pareto optimal. Depending on the nature of problem and optimization algorithm used to find robust Pareto optimal solutions, there might be four cases as follows:

- The Pareto optimal solution set (or front) is identical to the robust Pareto optimal solution set
- A portion of the Pareto optimal solutions are robust and there exists no other robust solution
- A portion of the Pareto optimal solutions are robust and there are other dominated solutions that are robust
- All Pareto optimal solutions in the Pareto optimal solutions set are not robust, and a set of dominated solutions (locally optimal front) is robust.

The ultimate goal of a robust multi-objective optimization process is to find the robust Pareto optimal solution set, which includes a set of solutions that are non-dominated considering the objectives and the applied noise intensity.

In this work, conditional operators are used to compare and confirm the robustness of Pareto optimal solutions obtained during a robust optimization process. The operators are then applied to Multi-Objective Particle Swarm Optimization (MOPSO) to showcase their merits. The rest of the paper is organizes as follows:

In Section II, related works are presented and discussed. The conditional Pareto optimal dominance and the robust variant of MOPSO are presented in Section III. Section IV provides experimental results and discussions. Finally, Section V concludes the work and suggest future directions.

II. Related work

As discussed above, the main objective of robust optimization, which is the focus of this work, is to minimize the impact of uncertainties in the decision variables of an optimization system.

In a non-robust algorithm, the only measure of evaluation and comparison is the objective values for the solutions. For finding robust optimal solutions, however, we need more evaluation metrics. All the methods in the literature are divided into two classes: those use an expectation measure and those use a variance measure [1].

A. Expectation measures

In the former case, an expectation measure is calculated first and used instead of the objectives. This expectation measure can be calculated using analytical integration or the Monto Carlo simulation as follows:

$$Minimize\text{: } \frac{1}{|B_\delta(\vec{x})|}\int \dots \int_{\vec{y}\in B_\delta(\vec{x})} F(\vec{y})\overrightarrow{dy}$$

$$Subject\ to\text{: } g_i(\vec{x}) \geq 0, i = 1,2,3,\dots,m$$

$$h_i(\vec{x}) = 0, i = 1,2,3,\dots,m$$
$$lb_i \leq x_i \leq ub_i, i = 1,2,3,\dots,n$$

where $\vec{x}$ represents a solution, $B_\delta(\vec{x})$ is the neighbourhood around the solution $\vec{x}$ with the radius of δ in the search space.

$$Minimize\text{:} \frac{1}{H}\sum_{i=1}^{H} f(\vec{x}+\vec{\delta})$$

$$Subject\ to\text{: } g_i(\vec{x}) \geq 0, i = 1,2,3,\dots,m$$

$$h_i(\vec{x}) = 0, i = 1,2,3,\dots,m$$
$$lb_i \leq x_i \leq ub_i, i = 1,2,3,\dots,n$$

where $\vec{x}$ is a solution and H shows the number of samples in the Monte Carlo simulation.

Regardless of the method of calculating the expectation measure, a population-based optimization algorithm evaluates each of the solutions using the expectation measure. Since the expectation measure is calculated based on the objective values of a neighborhood instead of a single solution, it favors robust solutions.

The issue with this method is that the analytical integration of the mathematical equations to represent most real-world problems are unknown, so we have to rely on the Monte Carlo approximation, which adds another degree of inaccuracy to the system. Another challenge is that for each of the sampled points in Monte Carlo approximation, the objective function(s) should be evaluated on multiple neighbouring solutions to calculate the average. This significantly increases the computational cost of the robust optimization process.

B. Variance measures

As an alternative to expectation measure, researchers have been largely using variance measures. In such methods, the original objective functions are maintained. A measure to quantify the variance of objectives in a certain neighborhood around the solution in the search space is then used. This measure is then used to decide whether a solution is valid or invalid. In other words, a variance measure is treated as a constraint.

Without the loss of generality, robust multi-objective optimization using a variance measure is as follows:

$$Minimize\text{: } f(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}) \dots, f_o(\vec{x})\}$$

$$Subject\ to\text{:} V(\vec{x}) = \frac{||F^*(\vec{x}) - f(\vec{x})||}{||f(\vec{x})||} \leq \eta$$

$$g_i(\vec{x}) \geq 0, i = 1,2,3,\dots,m$$
$$h_i(\vec{x}) = 0, i = 1,2,3,\dots,m$$
$$lb_i \leq x_i \leq ub_i, i = 1,2,3,\dots,n$$

where $F^*(\vec{x})$ calculates the average of the H sampled point around the solution $\vec{x}$ within the neighborhood of $\vec{\delta}$ and η is a threshold in [0, 1] defining the level of robustness of solutions.

C. Current gaps

In the preceding subjections, both expectation and variance measures were presented. The common feature of both methods is the use of sampled points to quantify the robustness. Based on the way the sampled points are generated, robust optimization methods are divided into two classes:

- Explicit averaging: in this approach, sampled points are explicitly generated in the neighbourhood to confirm robustness. This process is done considering the expected maximum uncertainties in real-world environment.
- Implicit averaging: in this approach, sampled are taken from a repository that is created by the algorithm itself during the optimization process.

The advantage of explicit averaging method is the highest level of reliability since the points are generated and evaluation explicitly using the objective functions. On the other hand, such methods suffer from high computational cost. For instance, let us consider a real-world problem that is optimized using a global optimizer in one week, the same problem is optimized using a robust optimizer in 10 weeks if we assume generating nine additional neighbouring sample points each time. Therefore, the computational cost of such methods are very high.

Implicit averaging methods alleviate the drawbacks of explicit averaging approaches by relying on the repository of sampled points made of the previously sampled solutions. Due to the stochastic nature of population-based algorithm that use implicit averaging, however, the number and distribution of solutions within a certain neighborhood around every solution is not guaranteed. Therefore, the main drawback of implicit averaging is low reliability.

This problem has been mentioned and tackled in several works within the context of single-objective optimization. For instance, Archive Sample Approximation was proposed in [2]. The authors leveraged Wassertein distance to find neighboring solutions and the calculated the implicit averaging.

A confidence measure was proposed in [3] to measure the confidence that one might have when evaluating each solution based on a robustness measure. In other words, confidence and robust measures were used to consider the number, distribution, and radius of sampled points in the neighborhood of a solution. We also proposed a method to tackle this problem for single-objective optimization algorithms using conditional relational operators [4].

The number of attempts to alleviate this drawback in multi-objective problems is a significantly less than what was discussed in the preceding paragraph. In this next section, a conditional Pareto optimal dominance is proposed to consider the number of sampled points within the neighborhood of a solution.

III. The Proposed Conditional Pareto Optimal Dominance and Robust MOPSO

As discussed in the preceding section, what reduces the reliability of implicit averaging methods might be due to the lack of:

- Enough number of sampled points in the δ radius around a solution
- Uniform distribution of sampled points in the δ radius around a solution

In the former case, a population-based algorithm might not provide enough sampled points due to several reasons, including but not limited to, being at the initial iteration of optimization, focusing on the exploration a new region of the search space, or not exploiting in the δ radius around a solution.

In the latter case, there might be enough solutions in the δ radius around a solution, but the distributions are not desirable. This might be again due to similar reasons discussed above. Due to the stochastic nature of population-based algorithms, however, we can assume that the distribution of solutions is close to uniform if there are enough sampled points in the δ radius around a solution.

After this assumption, the only factor left to decide whether we can calculate the robustness of a solution or not reliably is the number of sampled solutions within the neighborhood. In [4], we proposed a constraint to count the number of sampled points in the δ radius around a solution as follows:

$$cons(\vec{x},\vec{\delta},A) > \zeta$$

where $\vec{x}$ shows the solution that we are evaluating for its robustness, $\vec{\delta}$ is a vector that stores the maximum level of uncertainties for each decision variables in $\vec{x}$, A is a repository that stores all the sampled points during the optimization process regardless of their distances to the solution $\vec{x}$, and ζ indicates the desired minimum number of sampled point that we would like to have in $\vec{\delta}$-radius around the solution $\vec{x}$.

This constraint allows us to make reliable decisions when judging about the robustness of solutions using implicit averaging methods. As discussed above, in a multi-objective problem, solutions are compared using Pareto optimal dominance operator. To consider the proposed constraint and make such comparison more reliable, we propose the following Pareto dominance:

Reliable Pareto Dominance ($\overrightarrow{x_1} \prec_{cons} \overrightarrow{x_2}$):

$$\forall i \in \{1,2,3,\dots,o\}$$
$$\{f_i(\vec{x}) \le f_i(\vec{y})\} \wedge \{\exists i \in \{1,2,\dots,o\} :\ f_i(\vec{x}) < f_i(\vec{y})\ \wedge cons(\overrightarrow{x_1}) > \zeta \wedge cons(\overrightarrow{x_2}) > \zeta\}$$

where $\vec{x}$ and $\vec{y}$ are two vectors with o objectives.

The definition of proposed reliable Pareto dominance shows that for a solution to dominate another one, it must comply with regular Pareto dominance conditions. In addition, there must be a desired number of solutions around both solutions. There are four cases here to consider:

- If there are ζ sampled points around both $\overrightarrow{x_1}$ and $\overrightarrow{x_2}$: $\prec_{cons}$ is equivalent to $\prec$
- Otherwise: $\prec_{cons}$ is not equivalent to $\prec$

As the above list shows, the only case in which a solution reliably dominates the other one is the first one when there are enough sampled points around both solutions. In all other cases, we cannot reliably compare solutions.

Now the question is how the proposed method is used in conjunction with expectation and variance measures. In both cases, we can replace $f_i(\vec{x})$ with $R_i(\vec{x})$, which can be any robustness measure. This means that instead of comparing the objectives, we use expected objectives calculated using an expectation measure.

Just like any other new operator or approach, there are a number of advantages and drawbacks with the proposed methods. The main advantage of this method is improving the reliability of any implicit averaging method [4]. The proposed reliable Pareto dominance requires solutions to have a minimum number of sampled points in their neighborhood to be involved in the comparison and selection process of population-based algorithms.

The proposed reliable Pareto dominance method can be incorporated in most of the multi-objective algorithms in a straight-forward way. For example, in the Multi-Objective Particle Swarm (MOPSO) [5], the archive should be updated completely based on the reliable Pareto optimal dominance. The archive is a repository to store non-dominated solutions obtained by the population of particles in MOPSO in each of the iterations. In Non-dominated Sorting Genetic Algorithm (NSGA-II) [6], the dominance hierarchy can be entirely built based to the dominance level of each candidate using the proposed reliable Pareto optimal dominance.

As an example of how the proposed reliable Pareto optimal dominance can be used in population-based algorithms, in this work MOPSO is used as a case study. The pseudocode of the robust MOPSO algorithm with the proposed method is presented in Fig. 1.

```
Create the first population randomly (X and V)
Evaluate all particles using R(x)
Update sampled points' repository (A)

while the end condition is not satisfied
      Update w, c1, and c2

      for all particles
          Choose a GBEST from the archive as the leader
          Update X and V
          Update PBEST
          Evaluate using R(x)
      End

      Update sampled points' repository (A)
      Update archive using $\prec_{cons}$
end

Return archive
```

Fig. 1. The pseudocode of the proposed RMOPSO_IC

This algorithm is called robust MOPSO with implicit constraint (RMOPSO_IC), and it can be seen that the proposed reliable Pareto dominance is used when updating the archive. This allows the proposed algorithm to always have reliable, robust Pareto optimal solutions in the archive. The justification for doing this is that the GBESTs are selected from the archive as leaders to update the the positions and velocities of particles independently. Adding any non-robust solution to the archive can bias the whole search process towards non-robust regions of the search space. The position and velocity formula of RMOPSO_IC similar to those in MOPSO:

$$\overrightarrow{X_i(t+1)} = \overrightarrow{X_i(t)} + \overrightarrow{V_i(t+1)}$$

$$\overrightarrow{V_i(t+1)} = w\overrightarrow{V_i(t)} + c_1 r_1 \left(\overrightarrow{P_i(t)} - \overrightarrow{X_i(t)}\right) + c_2 r_2 \left(\overrightarrow{G(t)} - \overrightarrow{X_i(t)}\right)$$

where $\overrightarrow{X_i(t)}$ is a vector repressing the current position of a particle in the search space, $\overrightarrow{V_i(t)}$ is the velocity vector that is used to update the position vector, w is an inertial weight, c_1 indicates the cognitive constant, c_2 shows the social constant, $\overrightarrow{P_i(t)}$ is the best position of *i-th* particle, $\overrightarrow{G(t)}$ is the best position vector found by all particles from the first to the *t-th* iteration, and r_1/r_2 are randomly generated numbers in [0, 1].

In the next section, this algorithm will be tested and compared with three other algorithms.

IV. Experimental Results

In this section the proposed RMOPSO_IC is benchmarked. Just like other studies in this area, test functions are used as test beds to perform comparative study. In multi-objective optimization, people use well-known test suites such as DTLZ, ZDT, CEC, etc. However, none of these functions have been designed to test robust multi-objective optimization algorithms. Therefore, specific test functions are need for comparisons.

Several test functions, taken from different publications, are used in this work. The summary of those functions are presented in Table I. We used a wide range of challenging test functions to effectively test and compare the performance of the proposed method. To perform the comparison, we used different variants of the robust MOPSO algorithm. The three variants of the MOPSO that are used in this work are called RMOPSO_I, RMOPSO_E, and CRMOPSO as follows:

TABLE I. Benchmark problems used in this work

Name	Details
MTP1	First test functions in [7] with $\delta = 0.007$
MTP2	Second test problem in [7] with $\delta = 0.007$
MTP3	Third test problem in [7] with $\delta = 0.3$
MTP4	Test problem 1 in [8] with $\delta = 0.4$
MTP5	Test problem 2 in [8] with $\delta = 0.4$
MTP6	Test problem 3 in [8] with $\delta = 0.6$
MTP7	Test problem 4 in [8] with $\delta = 0.6$
MTP8	Test problem 5 in [8] with $\delta = 0.6$

- RMOPSO_I: An implicit averaging robust MOPSO algorithm that uses the expectation measure calculated using the Monte Carlo simulation.
- RMOPSO_E: An explicit averaging robust MOPSO algorithm that uses the expectation measure calculated using the Monte Carlo simulation.
- CRMOPSO: An implicit averaging robust MOPSO algorithm, equipped with the confidence measure proposed in [9], that uses the expectation measure calculated using the Monte Carlo simulation.

The RMOPSO_I has the lowest level of reliability and assists us to see how effective the reliable Pareto optimal dominance is in improving the reliability of this algorithm. The RMOPSO_E explicitly creates sampled points for particles, so this algorithm shows the highest level of reliability. RMOPSO_E works as a ground truth for the comparison and outperforming this algorithm is not expected by any of the algorithms. These two algorithms provide different points in the spectrum of reliability. With such comparisons, we would like to if the proposed method improve the reliability of RMOPSO_I when generating no extra sampled point. The CRMOPSO algorithm is chosen as a recent algorithm that tries to improve the reliability of RMOPSO_I as well.

In order to have a fair comparison, we use 100 particles, 1000 iterations, and a repository size of 100 for RMOPSO_I, RMOPSO_IC, and CRMOPSO. Therefore, the number of function evaluations for these three algorithms are 100,000. In case of RMOPSO_E, however, we used a different setting to have a fair comparison. Since RMOPSO_E explicitly uses three sampled points for calculating the expectation measure for each solution, so we use 100 particles, 250 iterations, and the archive size of 100.

All algorithms are run 30 times on each test function and the best robust Pareto optimal front are provided in Figs. 2 and 3. The test functions in Fig. 2 have multiple local fronts of which one is global and one is robust. Fig. 3 shows the test functions with the same global and robust front. However, some sections of the front is more robust than others, an algorithm should find. Note that the red lines in all figures show the robustness of the robust front, which can be calculated by generating and averaging random solutions in the vicinity of the true Pareto solutions. Hence, low value of the robustness curve indicates low sensitivity to perturbations in the related region of the Pareto optimal front. Since the second objective function is linear for all the test functions, it is a good indication of the robust regions of the robust Pareto optimal front. It helps us to see which algorithm finds more solutions in robust regions of the robust front.

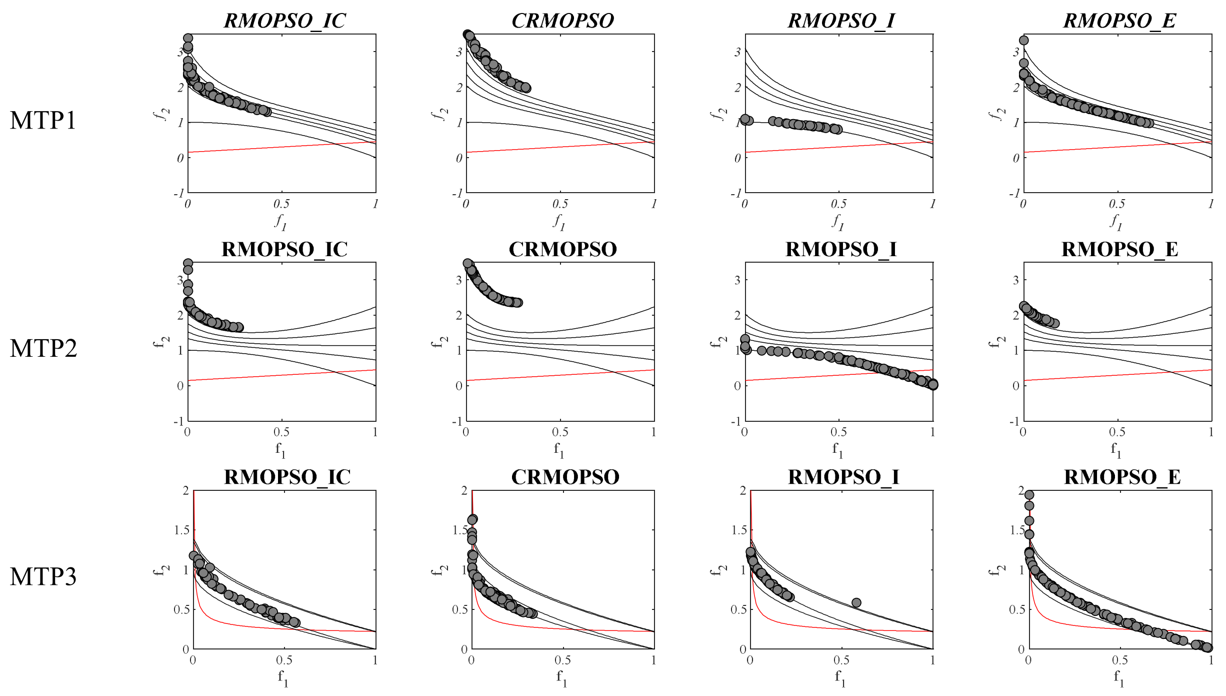

Fig. 2. Results of algorithms on test functions MTP1 to MTP3

Fig. 2 shows the performance of all algorithms on MTP1 to MTP3. In the first three test functions, the local fronts are robust. The level of perturbations (δ) considered indicates which one is the most robust local front. In all cases, the closest local front to the global front is the best robust front with the configuration considered in this work.

Inspecting the results of algorithms on MTP1, it can be seen that RMOPSO_I converged to the global front. This shows how unreliable this algorithm is by solely relying on previously sampled points. The Pareto optimal front of RMOPSO_E is nearly uniform and the solutions are gravitated towards the robust regions. A similar behavior can be observed in the results of the proposed RMOPSO_IC. However, the distribution is not as uniform as that of RMOPSO_E. These results demonstrate improvements in using implicit averaging along with the proposed method. The CRMOPSO has not converged to the robust front and is outperformed by RMOPSO_IC.

In MTP2, the robust front is the most dominated front. RMOPSO_I again finds the non-robust front. Although it is the best front, we are not interested in it since there are other robust fronts. CRMOPSO's convergence is not desirable, which is due to the impact of the confidence measure on the search pattern of MOPSO. The frequency of updating GBEST in CRMOPSO is higher than that in MOPSO. So, particles 'churn' around the same GBEST more often. This can reduce the social intelligence aspect of MOPSO. This is not observed in the results of RMOPSO_IC. The proposed algorithm converges towards robust regions of the robust front. The Pareto front obtained by RMOPSO_E is similar to that of RMOPSO_IC. However, some of the solutions obtained by the RMOPSO_IC has not converged to to robust front completely. This is due to the proposed reliable Pareto optimal dominance and its use in updating the archive. It reduces the convergence speed of the algorithm but decreases the chance of favoring non-robust solutions as well, which is desirable.

In MTP3, the first local front is the most robust one. It is evident that the best front obtained is that of RMOPSO_E. However, some solutions on the left hand sides do not have desirable accuracy. This is due to the use of less iterations for a fair comparison with other algorithms. The results provided by RMOPSO_IC, CRMOPSO, and RMOPSO_I are very competitive. Qualitatively speaking, however, the distributions and the accuracy of solutions obtained by the proposed RMOPSO_IC are slightly better than others.

The rest of the results on other benchmark functions can be found in Fig. 3. As discussed above, there is not local front in these test functions but some regions of the global front is more robust than others. In RMTP4, for instance, the robustness decreases proportional to *f1*. The results of all algorithms on this test function show that the best performance is provided by the proposed RMOPSO_IC. It is evident that the solutions of RMOPSO_IC scatter in the middle on MTP4 and on the left hand side of the Pareto optimal fronts on both MTP5 and MTP6. This indicates that the RMOPSO_IC algorithm is more successful in identifying the part of the front most robust compared to other algorithms.

The most interesting behaviour can be observed in the results of algorithms on RMTP7 and RMTP8. These two test functions have three separated robust regions. RMOPSO_I, RMOPSO_E, and CRMOPSO found solutions in both robust and non-robust regions. RMOPSO_IC is the only algorithm that finds the most robust solutions. This once again highlights the merits of the proposed reliable Pareto dominance and using it in archiveupdating. Both mechanisms help MOPSO in comparing solutions reliably, finding robust non-dominated solutions, and preserving them in the archive.

To further investigate the impact of the proposed reliable Pareto optimal dominance, we have counted the number of times that the archive gets updated with a 'risky' decision in RMOPSO_I. This shows us how many times RMOPSO_I is prone to bias the search towards the solutions that we are not

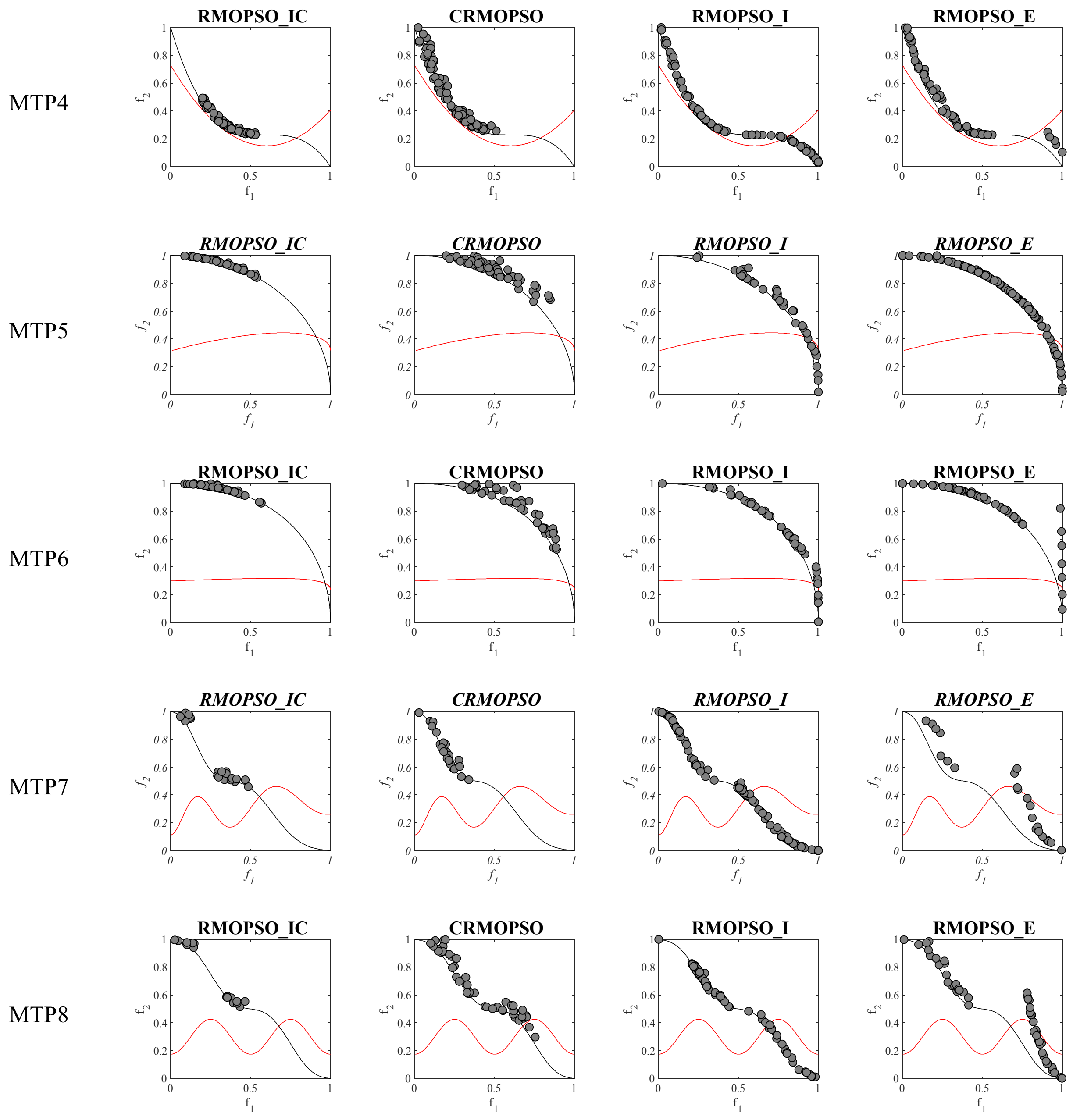

Fig. 3. Results of algorithms on test functions MTP4 to MTP8

sure about their robustness. Additionally, it also shows the significance of RMOPSO_IC in making reliable decisions. The results are presented in Table II.

TABLE II. BENCHMARK PROBLEMS USED IN THIS WORK

Test case	Risky decision: $\vec{x_i} \prec \overrightarrow{GBEST}$ but $\vec{x_i} \nprec_{cons} \overrightarrow{GBEST}$
MTP1	35%
MTP2	38%
MTP3	40%
MTP4	30%
MTP5	34%
MTP6	34%
MTP7	39%
MTP8	33%

Table II shows that the –RMOPSO_I algorithm is potentially prone to making 'risky' decisions in 35% times on an average. This is a significant number that shows how unreliable an algorithm can be when relying on previously sampled points only. This was the main motivation of this work to find a way to prevent such cases. Even having one non-robust solution in the archive can bias the whole search process towards non-robust regions of the search space.

Taken together, the results of the proposed RMOPSO_IC on the test functions show that the proposed reliable Pareto optimal dominance and archive updating mechanism are beneficial in locating robust solutions during the optimization process. It was observed that this comes at the cost of slight decline in the convergence speed of the algorithm. The results show that this is not a major side-effect since the Pareto fronts obtained were better than other algorithms on most of the test functions. Another issue with the proposed method, also present in CRMOPSO, is the need to store and search the previously sampled points. This might be computationally expensive for large population sizes but for extremely expensive objective function, this will be insignificant. For in-expensive problems, in addition, we can limit the number of latest iterations used to store and search previously sampled points.

V. Conclusion

In this work, we address one of the most crucial drawbacks of robust optimization methods relying on implicit averaging. Population based robust optimization algorithms that utilizes implicit averaging based on previously sampled points only are not reliable since there is no guarantee to get enough number of sampled points with proper distributions.

This paper proposes a reliable Pareto optimal dominance technique to compare solutions in multi-objective problems. In the proposed method, a solution reliably dominates another if it complies with the conditions of the regular Pareto optimal dominance and there are a certain minimum number of sampled points around the two solutions involved in the comparison. The paper integrated the proposed mechanism in MOPSO to demonstrate its applicability. Since the archive in MOPSO with reliable Pareto optimal dominance stores the non-dominated solutions during the optimization process and the leaders get selected from it, no non-robust solutions should enter the archive. The reliable Pareto dominance is used here to update the archive each time.

The proposed method was tested on eight benchmark functions and compared with three other variants of MOPSO: RMOPSO_E, RMOPSO_E, and CRMOPSO. The best estimation of the robust Pareto optimal solutions and fronts obtained by the algorithms were compared and reported. The results show that the proposed reliable Pareto optimal dominance is able to prevent 'risky' decisions when relying on previously sampled points in calculating expectation measure using implicit averaging methods. The identified Pareto optimal fronts show that better (robust) solutions can be obtained when using the proposed reliable Pareto optimal dominance in the MOPSO algorithm.

Based on the empirical results and the analyses done in this work, we can conclude that the proposed reliable Pareto optimal dominance has merits in increasing the reliability of implicit averaging methods. For future work, it is recommended to investigate the use of reliable Pareto optimal dominance in other well-regarded multi-objective algorithms (e.g. MOEA/D, NSGA-II, etc.).

References

[1] K. Deb and H. Gupta, "Searching for robust Pareto-optimal solutions in multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2005: Springer, pp. 150-164.

[2] J. Branke and X. Fei, "Efficient sampling when searching for robust solutions," in *International Conference on Parallel Problem Solving from Nature*, 2016: Springer, pp. 237-246.

[3] S. Mirjalili, A. Lewis, and S. Mostaghim, "Confidence measure: a novel metric for robust meta-heuristic optimisation algorithms," *Information Sciences,* vol. 317, pp. 114-142, 2015.

[4] S. Z. Mirjalili, S. Mirjalili, H. Zhang, S. Chalup, and N. Noman, "Improving the reliability of implicit averaging methods using new conditional operators for robust optimization," *Swarm and Evolutionary Computation,* vol. 51, p. 100579, 2019.

[5] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2002, vol. 2: IEEE, pp. 1051-1056.

[6] 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.

[7] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization," *Evolutionary computation,* vol. 14, no. 4, pp. 463-494, 2006.

[8] A. Gaspar-Cunha, J. Ferreira, and G. Recio, "Evolutionary robustness analysis for multi-objective optimization: benchmark problems," *Structural and Multidisciplinary Optimization,* vol. 49, no. 5, pp. 771-793, 2014.

[9] S. Mirjalili, A. Lewis, and J. S. Dong, "Confidence-based robust optimisation using multi-objective meta-heuristics," *Swarm and Evolutionary Computation,* vol. 43, pp. 109-126, 2018.