Computer Science Journal of Moldova, vol.**28**, no.3(84), 2020

Loser-out multi metaheuristic framework for multi-objective optimization

Jamshid Tamouk, Nasser Lotfi

Abstract

This paper proposes a multi metaheuristic framework consisting of four multi-objective optimization (MOO) algorithms in which they compete with each other along four phases to be surviving in the next phases. Likewise, it is assumed that number of phases is equal to the number of metaheuristics. The proposed method, named as Loser-Out-Framework (LOF) from this point on, runs in consecutive sessions so that a session starts with dividing global population into several subpopulations. Thereafter in the first phase, entire set of metaheuristics is assigned to each subpopulation and then metaheuristics are performed over subpopulations to modify and improve them. In continuation of each phase, non-dominated solutions extracted by all metaheuristic sets are stored in global archive, and then the most ineffective metaheuristic of each subpopulation is eliminated. The proposed method is evaluated and tested over the well-known DTLZ and WFG benchmarks. Comparative evaluations against several state-of-the-art algorithms exhibits that the proposed framework outperforms others in terms of extracted Pareto front quality.

Keywords: Multi-Objective Optimization, Metaheuristic, Metaheuristic Based Framework.

1 Introduction

Even though metaheuristics have been widely applied for solving NP-hard optimization problems, no individual metaheuristic performing well enough over all optimization problems can be found. Therefore, having an appropriate framework consisting of various metaheuristics became significantly important when high quality solutions for a vast variety of optimization problems are needed. In this paper, four recently published metaheuristic algorithms, namely NSGA-III, R-NSGA-III, U-NSGA-III and the well-known MOEA/D, are implemented and used in the proposed framework for solving multi-objective optimization (MOO) problems. Four MOO algorithms in the proposed framework compete with each other along four phases to be surviving in the next phases. The proposed method, named as Loser-Out-Framework (LOF) from this point on, runs in consecutive sessions so that a session starts with dividing global population into several subpopulations. Afterwards in the first phase, all metaheuristics are assigned to each subpopulation, and then metaheuristics are performed over subpopulations to improve them. Later on, the non-dominated solutions found by all metaheuristic sets are kept in global archive, and then the weakest metaheuristic is eliminated from each subpopulation. The proposed method is tested over three-objective DTLZ and WFG benchmarks by following the process described for these benchmark problems. Success of the proposed framework on most of the test cases, in comparison to state-of-the-art algorithms puts it in the first position among its well-known competitors. A detailed description of the proposed architecture is given in Section 4. The rest of this paper is organized as follows: next section (Section 2) indicates a brief discussion on state-of-the-art algorithms in metaheuristic-based approaches for MOO problems. In Section 3, MOO metaheuristics used in the proposed framework are discussed and then the proposed framework for MOO is introduced in Section 4. In Section 5 experimental results and their evaluations are given, and finally, the conclusion and feature research directions are presented in Section 6.

2 State-of-the-art in metaheuristic-based approaches for MOO

Pretty rich amount of state-of-the-art methods for solving multi-objective optimization problems can be found in the literature [8]-[12]. Some recently proposed algorithms, namely NSGA-III-WA, NSGA-III, VAEA, RVEA, MOEA/D and MOEA/D-M2M, are selected from state-of-the-art methods to be used for evaluation of our proposed method. NSGA-III-WA algorithm was proposed by Wang et al. in order to improve standard NSGA-III. Their proposed NSGA-III-WA improves NSGA-III in terms of evolution strategy (using combination of the new differential evolution strategy proposed in this article together with genetic evolution strategy) and weight vector adjustment (to decompose the objective space into several subspaces) [7]. The proposed algorithm speeds up the method without reducing the performance by adding a discriminating condition to the algorithm. Authors tested NSGA-II-WA over DTLZ benchmark set and found that it is effective in terms of convergence and distribution [7]. Generation framework of NSGA-III-WA is presented in Algorithm 2.1, where in the algorithm: N is population size, P_t is the initial population, Q_t is the population result from applying the differential operator on population and W_{unit} is the weight vector.

Input: N structured reference points W_unit, parent population P_t
Output: P_{t+1}

(1) Initialization (P_t , W_unit)
(2) Gen = 1
(3) **While** Gen $\leq$ Gen_max **do**
(4) Qt = Evolutionary strategy (P_t)
(5) Rt = $P_t \cup Q_t$
(6) P_t + 1 = Environmental_selection (R_t)
(7) W_unit = Weight_Adjustment (W_unit)
(8) Gen++
(9) **End While**
(10) **Return** P_t+1

Algorithm 2.1. Generation framework of the proposed NSGA-III-WA[7]

The population size of R_t is set to 2N (due to combining two populations P_t and Q_t together with the size of N) and differential operator is taken into account as an evolutionary strategy. Also for selecting the best individuals as a next generation, the environmental selection operation (which includes: normalization of objective values, associate of each individual resulted from normalization with the reference points, Compute the Niche Count of the Reference Point, Niche Preservation Operation) is employed. Kalyanmoy and Himanshu [15] proposed NSGA-III in 2014 which differs from NSGA-II in terms of individual selections. NSGA-III is a robust technique that eliminates the weaknesses of NSGA-II [7],[15], e.g., it solves the issue of lacking uniform diversity. The diversity is maintained by computing crowding distance in NSGA-II, but in NSGA-III it is realized by using reference direction niching. NSGA-III attempts to find Pareto-front optimal solutions that are close to reference points. The algorithm has been tested over different benchmarks and outcomes proved its robustness. Algorithm 2.2 shows the general steps of NSGA-III in search process. More details of NSGA-III can be found in [2], [7], [15].

1. Generate N reference points regarding to the population size.
2. Generate initial population.
3. Repeat until termination criterion is satisfied:
   3.1. Apply related operators to generate new population
   3.2. Combine the populations and apply non-dominated sorting.
   3.3. Assign points to the reference points.
4. END

Algorithm 2.2. NSGA-III steps

VAEA (Vector Angle-Based Evolutionary Algorithm) was proposed by Xiang et al. in 2017 which is based on angle decomposition [14]. VAEA tunes convergence and diversity of search space without needing reference points and it uses maximum-vector-angle-first principle to guarantee the wideness and uniformity of the solution set. It has a similar environmental selection to the NSGA-II and NSGA-III, but with different niche preservation operation. The authors evaluated VAEA over many-objective benchmarks, and the obtained results illustrated that VAEA tunes convergence and diversity in a good manner. Algorithm 2.3 shows the steps of VAEA framework, where: P is the initial population with the size of N, Q is the population result from applying the Mating_selection and Variation operators on P, S is the union set of P and Q, G is the number of generations, and G_{max} is the maximum number of generations.

```
1: Initialization (P)
2: G = 1
3: While G leq G_max do
4:     P' = Mating_selection(P)
5:     Q = Variation(P')
6:     S = P ∪ Q
7:     P = Environmental_selection(S)
8:     G + +
9: End While
10: Return P
```

Algorithm 2.3. Framework of the VAEA [14]

Mating selection is the operation to select more potential solutions for mating pool (i.e.$P^{'}$) based on the fitness value of each individual. Variation is the operation to generate a set of offspring solutions (i.e. Q) by applying crossover and mutation. Environmental selection is a procedure to select N solutions from union set of P and Q.

Ran, Yaochu, Markus and Bernhard [13] suggested RVEA (Reference Vector Guided Evolutionary Algorithm) in 2016 in which the novelty of this approach is to employ two components, namely, reference vector guided selection and the reference vector adaptation (which is used to dynamically tune the weight vectors in accordance to objective functions) to improve the performance. The authors tested RVEA against 5 state-of-the-art methods and found that RVEA is effective and cost-efficient. Algorithm 2.4 indicates the RVEA framework, where: P_0 is the initial population, P_t is the population resulted after t iterations, Q_t is the population resulted from applying offspring-creation operation on P_t, N is the population size, V_0 is the initial reference vectors, V_t is the reference vectors resulted after t iterations, P_{t+1} is the new population resulted from applying reference-vector-guided-selection operator on the current population, and V_{t+1} is the new reference vectors resulted from applying reference-vector-adaptation operator on the current reference vectors.

Input: the maximal number of generations t_{max}, a set of unit reference vectors $V_0 = \{v_{0,1}, v_{0,2}, \ldots, v_{0,N}\}$;

Output: final population $P_{t_{max}}$;
/*Initialization*/

Initialization: create the initial population P_0 with N randomized individuals;
/*Main Loop*/
while t ¡ t_{max} **do**
  Q_t = offspring-creation(P_t);
  $P_t = P_t \cup Q_t$;
  P_{t+1} = reference-vector-guided-selection(t, P_t, V_t);
  V_{t+1}= reference-vector-adaptation(t, P_{t+1}, V_t, V_0);
  t = t + 1;
end while

Algorithm 2.4. RVEA framework

In the algorithm, offspring-creation is the operation used to generate offspring using crossover, mutation and elitism selection.

MOEA/D-M2M was proposed by Hai-Lin, Fangqing and Quingfu [22] in 2014 which is based on the divide and conquer technique. The proposed algorithm divides the Pareto-front and search space into segments and sub-spaces, then each sub-problem regarding to its own segment and sub-space is solved separately. This way the distribution of extracted solutions is increased. Algorithm 2.5 shows the steps of MOEA/D-M2M.

Input :
- MOP (1);
- A stopping criterion;
- K: the number of the subproblems;
- $v^1, \ldots, v^K$: K unit direction vectors;
- S: the size of subpopulation;
- Genetic operators and their associated parameters.

Output: Ψ: a set of nondominated solutions
Initialization: Uniformly randomly choose K $\times$ S points from $[a, b]^n$, compute their F-values and then use them to set $P_1, \ldots, P_K$.
while the stopping criterion is not met **do**
  Generation of New Solutions:
  Set R = ϕ;
  for k $\leftarrow$ 1 to K **do**
    foreach x $\in P_K$ **do**
      Randomly choose y from P_K;
      Apply genetic operators on x and y to generate a new solution z;
      Compute F(z);
      R := R $\cup \{z\}$;
  end
  Q := R $\cup(\cup_{k=1}^{K} P_K)$;
  use Q to set $P_1, \ldots, P_K$;   **end**
  Find all the nondominated solutions in $\cup_{k=1}^{K} P_K$ and output them.
End

Algorithm 2.5. MOEA/D-M2M

Quingfu and Hui [3] proposed MOEA/D (Multi-objective Evolutionary Algorithm) in 2007 which is based on decomposition. MOEA/D divides a problem into sub-problems and then optimizes them at the same time. MOEA/D speeds up the optimization process without affecting the quality of results. Meanwhile, it provides a better distribution over the extracted objectives. The MOEA/D became very popular after earning the first place (among the 13 competitors) in CEC2009 contest. Back then, diverse versions of MOEA/D have been proposed by applying different decomposition methods, e.g. MOEA/D-DE [16], MOEA/D-DRA [18], MOEA/D-XBS [19]. More details of MOEA/D can be found in [3].

3 MOO metaheuristics used within the proposed framework

3.1 NSGA-III (Non-dominated Sorting Genetic Algorithm)

NSGA-III was introduced by Kalyanmoy and Himanshu [15] in 2014 (designed for problems with many objectives) so that its implementation is based on NSGA- and NSGA-II. NSGA-III and NSGA-II are basically similar (in both of them, nondominated sorting is applied to rank the population into a number of fronts), but mostly different in terms of mechanism, e.g., in NSGA-III, the diversity and convergence metrics are enhanced by using a set of reference points and directions for selecting the nondominated solutions for the next generation (i.e., it is the combination of Pareto-based evolutionary multiobjective optimization/EMO algorithm and decomposition). Using a reference direction, it would be possible to start from a reference point and pass over reference direction. Meanwhile in each epoch, a population member is found for each reference direction. Likewise, diversity is maintained by computing crowding distance in NSGA-II, but in NSGA-III it is realized by using reference direction niching. NSGA-III attempts to find Pareto-front optimal solutions that are close to reference points, but in the most of optimization problems optimal Pareto-front is not known.

NSGA-III works robustly against the problems with many objectives (three or more); and, similar to NSGA-II, there is no need for any parameter setting other than population size, termination criteria, crossover, and mutation probabilities. There is a none-algorithmic parameter for number of reference points in which population size is dependent on it (they should be approximately equal). The extracted Pareto-front guarantees a good distribution of points. NSGA-III follows the steps shown in Algorithm 3.1 in search process. More details of NSGA-III can be found in [2], [7], [15].

1. Generate N reference points regarding the population size.
2. Generate initial population.
3. Repeat until termination criterion is satisfied:
    3.1 Apply related operators to generate new population
    3.2. Combine the populations and apply non-dominated sorting.
    3.3. Assign points to the reference points.
4. END

Algorithm 3.1. NSGA-III steps

3.2 R-NSGA-III

R-NSGA-III was proposed by Yash, Kalyanmoy and Julian [4] in 2018 (based on recently proposed NSGA-III and R-NSGA-II method) in order to find a desired part of optimal Pareto-front. To do so, R-NSGA-III defines some reference points and carries out multi-criterion decision-making method. Also, the method uses an epsilon parameter to represent minimum distance between neighbors. When the epsilon value is larger, large amount of solutions around reference points are selected. R-NSGA-III applies clearing based niching instead of crowding distance based niching. In this algorithm the authors extend the R-NSGA-II method to get a more uniform emphasis in finding solutions for all supplied aspiration points. Moreover, they extend the NSGA-III with reference point concept for the same purpose [4] by taking into account a novel reference point generation method based on user-supplied aspiration points (z^K) in which K aspiration points ($z_i^{1,\dots,K}$) are provided by the user in M-dimensional objective space:

$$r^{(K)} = \left(z_1^{(K)}, z_2^{(K)}, \dots, z_M^{(k)}\right) \quad k = 1, 2, \dots, K,$$

where, M is the number of objectives, K is the number of aspiration points, and z^K is the k_{th} aspiration point. R-NSGA-III employs the same genetic operators and survival selection process as NSGA-III. More details of RNSGA-III can be found in [4].

3.3 MOEA/D (decomposition-based multi-objective evolutionary algorithm)

MOEA/D (multi-objective evolutionary algorithm based on decomposition) was proposed by Quingfu and Hui [3] in 2007 which is one of the most popular multi-objective evolutionary algorithms. MOEA/D breaks down the original multi-objective problem into several single-objective sub-problems using decomposition method and then optimizes them concurrently (with regard to optimization, a number of weight vectors with good distribution are often required) by taking into account the neighborhood relationships among sub-problems (for selecting mating parents and population replacement).

Diverse versions of MOEA/D have been proposed by applying different decomposition methods, e.g. MOEA/D-DE [16], MOEA/D-DRA [18], MOEA/D-XBS [19], and MOEA/D-GR [20]. Figure 3.1 represents the flowchart of MOEA/D algorithm [21]. Steps of this algorithm are illustrated in Figure 3.1, where N is the population size and weight vector size $(\lambda_1, \ldots, \lambda_N)$; x is the initial population $(x_1, \ldots, x_n)$, in which x_i is considered as the i_{th} solution; f_i is the i_{th} objective; z_i in formula $Z = (z_1, \ldots, z_m)$ is the best value found so far for objective f_i; λ_i is the weight vector; T is the number of the closest weight vectors to each weight vector (λ_i); and EP is the external population for storing the found NDSs.

Due to optimizing the sub-problems using some neighborhood sub-problems, MOEA/D is not time consuming method. Even though MOEA/D is faster than NSGA-II, the obtained results are similar or even better than NSGA-II. Likewise MOEA/D achieves better distribution for three-objective problems than other methods. Detailed information can be found in [3].

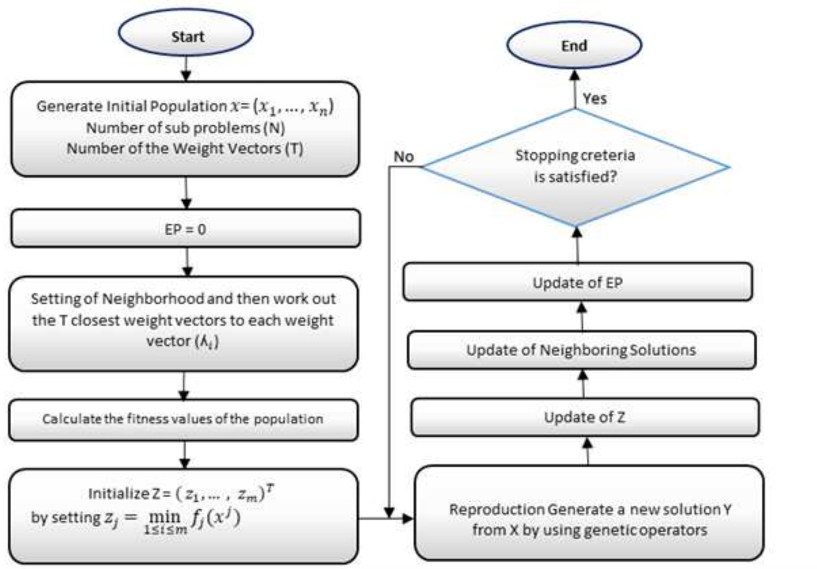


Figure 3.1. MOEA/D Flowchart [21]

3.4 U-NSGA-III

U-NSGA-III is a unified evolutionary optimization algorithm based on NSGA-III. As described in Section 3.1, NSGA-III was proven to work well on many objectives problems. But it has difficulties when applying it to single or two-objective problems; therefore, U-NSGA-III comes up with certain changes in NSGA-III algorithm to overcome this shortage by unifying all three types of optimization problems together (namely, single-objective, multi-objective and many-objective problems). Some changes are:

- Unlike NSGA-III, N (population size) and H (number of reference points) are different parameters with a condition that $N \geq H$ and N is a multiple of four.

- It increases the performance of NSGA-III by replacing tournament pressure (niching-based tournament selection) with random selection.

- In multi-objective optimization, U-NSGA-III has multiple population members for each reference direction, and its non-dominated sorting method is used to divide population into multiple non-dominated fronts. It focuses on non-dominated solutions and solutions closer to reference points.

```
Input: Two parents: p1 and p2
Output: Selected individual, ps
if π(p1) = π(p2) then
    if p1.rank ¡ p2.rank then
        ps = p1
    else
        if p2.rank ¡ p1.rank then
            ps = p2
        else
            if d⊥(p1) < d⊥(p2) then
                ps = p1
            else
                ps = p2
            end if
        end if
    end if
else
    ps = random Pick (p1, p2)
end if
```

Algorithm 3.2. Niching based selection

But, similar to NSGA-II and NSGA-III, this algorithm also doesn't need additional parameters than usual optimization algorithms [5], and the rest of process is similar to standard NSGA-III. Algorithm 3.2 indicates the pseudo-code for niching-based selection in U-NSGA-III, where $\pi(s)$ denotes the closest reference point, p_1 and p_2 are the parents passed to the algorithm as an input, and p_s is the best individual selected so far (i.e., winner). As it is shown in the pseudo-code in Algorithm 3.2, if two solutions under consideration belong to different associated niches (i.e., reference directions), in that case one of them should be selected randomly. Otherwise, the solution that belongs to a better non-dominated rank should be selected. However, if both solutions belong to the same niche and the same non-dominated front, the closest one to the reference direction should be selected.

4 The proposed multi metaheuristic framework for MOO

Loser-Out-Framework (LOF) consists of four different robust metaheuristics cooperating together in collecting Non-Dominated Solutions (NDS) in a global archive and also updating subpopulations with the improved ones during execution. All metaheuristics compete with each other to survive and no to be eliminated. The proposed process starts with dividing the global population to 'p' sub-populations and then, as the first phase, applying entire set of metaheuristics (which contains 'n' number of metaheuristics) to all 'p' sub-populations for fitness evaluations of '$\left(\frac{\alpha}{s-n+i}\right) t^{s-n+i} + \beta$' (which totally should not exceed the maximum number of fitness evaluations allowed for each test instance) times. For example, let's assume that $\alpha = 100, t = 10, s = 5, \beta = 100$, and $i = 0, \ldots, n$. So, the first phase will run for $\left(\frac{100}{5-4-0}\right) t^{5-4-0} + 100 = 1100$ times to find the first loser of each sub-population and remove it from their metaheuristic set. Non-dominated sets, Pareto front, found by all metaheuristics including the loser will be stored in a global archive while the evolved-populations by the loser are discarded (to avoid trapping into local optimum) and this process will continue until the number of survived metaheuristics in each set becomes one. However, the rest of the fitness evaluations (which is a considerably big number) will be assigned to the winner of the competition in each subpopulation. At the end, all Non-Dominated Solutions found by all metaheuristics are combined and then a constant number of non-dominated solutions is selected as a final result. Thereafter, for comparison, the IGD values are computed. Process and steps of the proposed framework are shown in Figure 4.1.

There is a simple but efficient logic behind having smaller number of fitness evaluations in earlier phases and increasing it in the next phases, that is, one metaheuristic (Loser) is eliminated from metaheuristic set of each sub-population in each step, therefore, in the next phases more effective metaheuristics remain in sets. This is how to give more chances to effective ones and increase the quality of extracted solutions.

Figure 4.2 illustrates an example of LOF approach containing four metaheuristics (m1, m2, m3, and m4) applied to population which has been divided into four sub-populations (sub-pop1, sub-pop2, sub-pop3, and sub-pop4). The following indicates the calculation of fitness evaluations for all four phases under the assumption of total fitness evaluations of 300,000.

$$\left(\frac{\alpha}{s-n+i}\right)t^{s-n+i}+\beta$$

$$\left(\frac{\alpha}{s-n+0}\right)t^{s-n+0}+\left(\frac{\alpha}{s-n+1}\right)t^{s-n+1}+\left(\frac{\alpha}{s-n+2}\right)t^{s-n+2}+ \left(\frac{\alpha}{s-n+3}\right)t^{s-n+3}+\beta=300,000$$

$$\left(\frac{100}{5-4+0}\right)10^{5-4+0}+\left(\frac{100}{5-4+1}\right)10^{5-4+1}+\left(\frac{100}{5-4+2}\right)10^{5-4+2}+ \left(\frac{100}{5-4+3}\right)10^{5-4+3}+\beta=300,000$$

$$100\times 10+50\times 100+33\times 1000+25\times 10000+\beta=1000+5000+ 33000+250000+\beta\approx 300,000$$

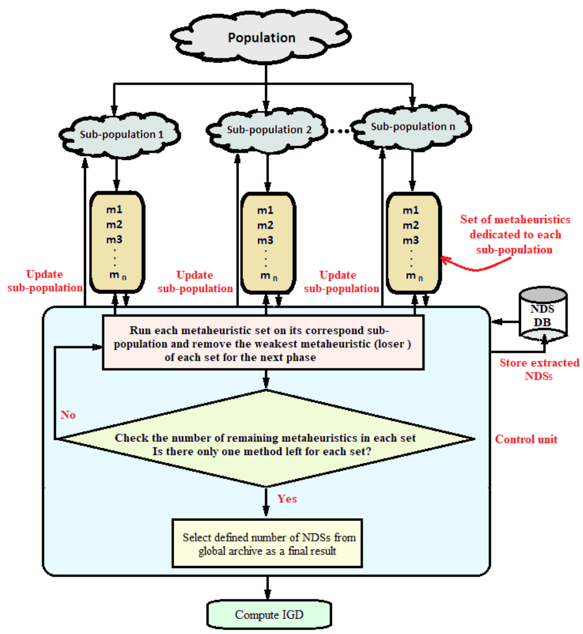


Figure 4.1. Loser-Out-Framework steps and components

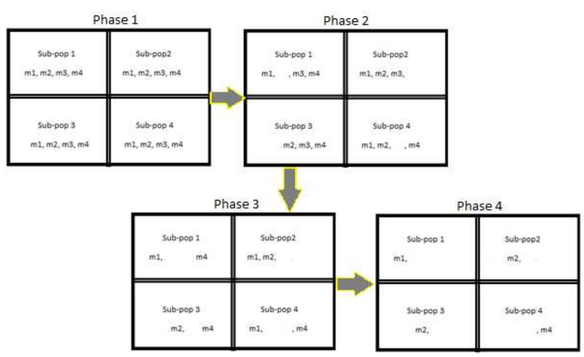


Figure 4.2. Process of dropping the loser metaheuristics of each sub-population

5 Evaluations and Experimental results

The proposed method is evaluated over the well-known three-objectives DTLZ and WFG benchmarks [6], and inverse generational distance (IGD) values are computed over 30 independent runs of each problem. Thereafter, the proposed framework is compared to six state-of-the-art algorithms [7]. For each of the test functions, the number of fitness evaluations is set according to values reported in references [7].

Table 5.1 illustrates the rank of each metaheuristic used in Loser-Out-Framework for DTLZ test cases. As an example, for DTLZ1 test case, MOEA/D was the worst one, and it has been removed at the end of first phase. Similarly, NSGA-III was the best one which survives until the last phase and earns a big portion of fitness evaluations. Based on Table 5.1, R-NSGA-III is the best performing method winning 3 out of 6 problems and NSGA-III takes the second position with 2 out of 6 problems.

Table 5.1. Rank of the metaheuristics used in Loser-Out-Framework applied to three-objectives DTLZ

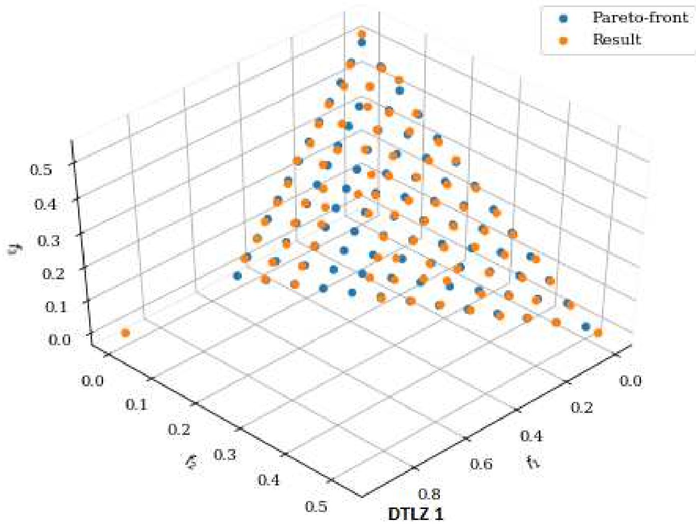
Function	MOEA/D	NSGA-III	R-NSGA-III	U-NSGA-III
DTLZ1	4	1	3	2
DTLZ2	4	2	3	1
DTLZ3	2	4	1	3
DTLZ4	4	1	3	2
DTLZ5	4	2	1	3
DTLZ6	3	4	1	2

Table 5.2 illustrates the average IGD scores obtained by LOF and its six recently published state-of-the-art competitors [8]-[12] over three objective DTLZ instances.

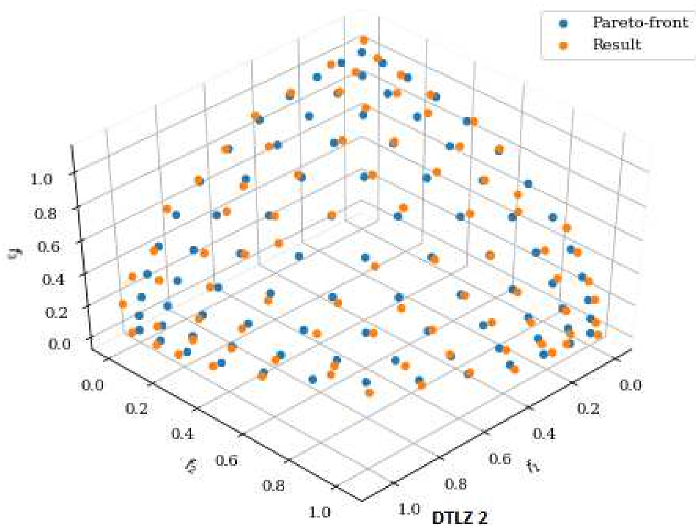
Table 5.2. IGD values obtained by LOF and its six competitors for three-objectives DTLZ

Function	NSGA-III-WA	NSGA-III	VAEA	RVEA	MODA/D	MODA/D-M2M	LOF	LOF Rank out of 7
DTLZ1	0.0314 ±0.0006	0.0209 ±0.0006	0.0777 ±0.0008	0.0620 ±0.0027	0.0408 ±0.0071	0.0431 ±0.0055	0.01787 ±0.0021	1
DTLZ2	0.0547 ±0.0002	0.05457 ±0.0004	0.0563 ±0.0008	0.0549 ±0.0001	0.0639 ±0.0007	0.0941 ±0.002	0.04353 ±0.0044	1
DTLZ3	0.0589 ±0.0007	0.0993 ±0.0008	0.0559 ±0.0019	0.0660 ±0.0044	0.0638 ±0.0014	0.0949 ±0.001	0.05673 ±0.0039	2
DTLZ4	0.0029 ±0.0001	0.0036 ±0.0007	0.0553 ±0.1937	0.0033 ±0.0002	0.0643 ±0.1009	0.0793 ±0.0316	0.00344 ±0.0050	3
DTLZ5	0.1281 ±0.0158	0.1143 ±0.0056	0.1674 ±0.0570	0.2057 ±0.0032	0.4196 ±0.0023	0.0432 ±0.0088	0.09230 ±0.0078	2
DTLZ6	0.9766 ±0.0252	1.516 ±0.0912	1.656 ±0.0509	1.303 ±0.0202	1.515 ±0.0075	1.826 ±0.0036	0.94340 ±0.0044	1
DTLZ7	NA	NA	NA	NA	NA	NA	0.08722 ±0069	NA

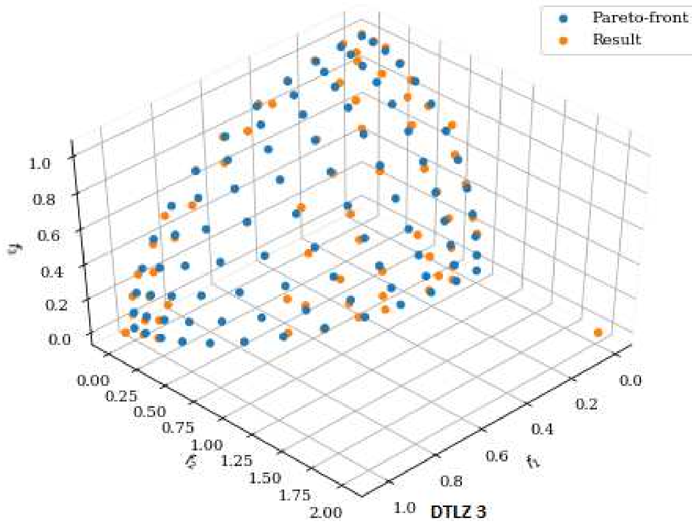
Based on Table 5.2, LOF takes the first position for DTLZ1, DTLZ2 and DTLZ6, the second position – for DTLZ3 and DTLZ5, and the third position – for DTLZ4 test instances. It means that in 50% of the test cases, LOF takes the first position and for the rest of the test cases it performs very close to the best ones. Likewise, Pareto-Fronts extracted by LOF for problems DTLZ1 to DTLZ7 are visually represented in Figure 5.1.



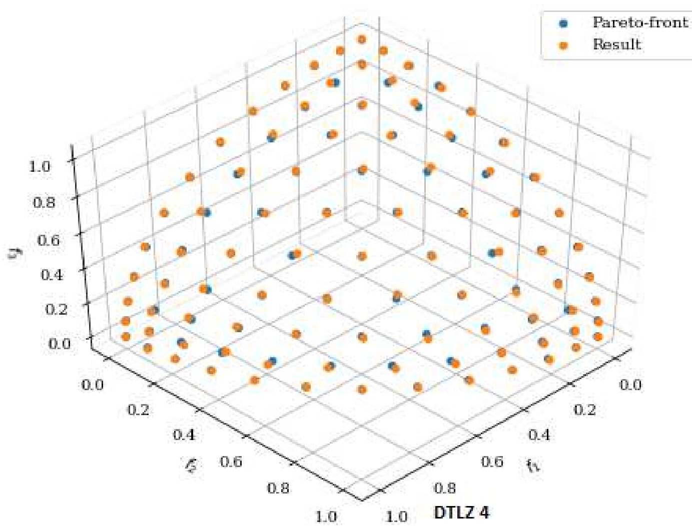
(a) **Figure 5.1.a.** Pareto front and optimal solution set for DTLZ1



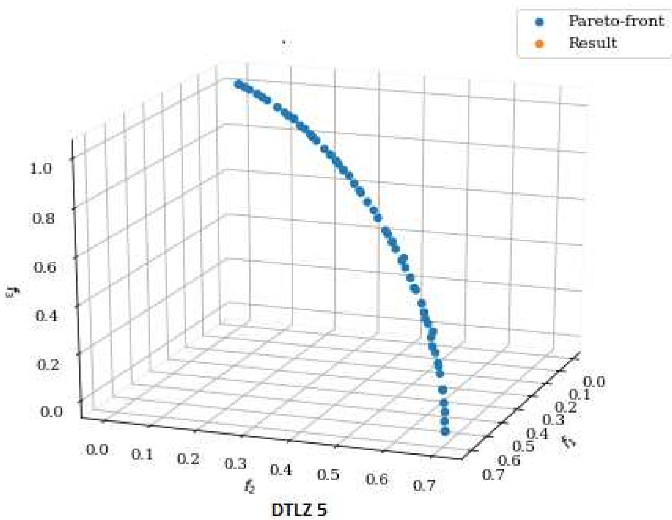
(b) **Figure 5.1.b.** Pareto front and optimal solution set for DTLZ2



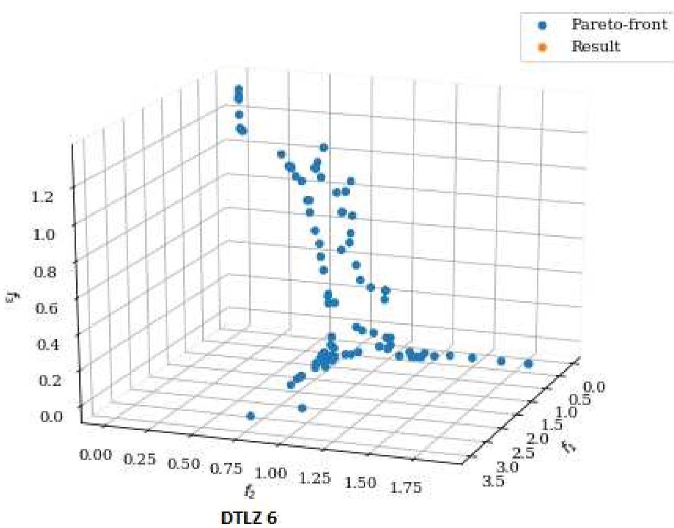
(a) **Figure 5.1.c.** Pareto front and optimal solution set for DTLZ3



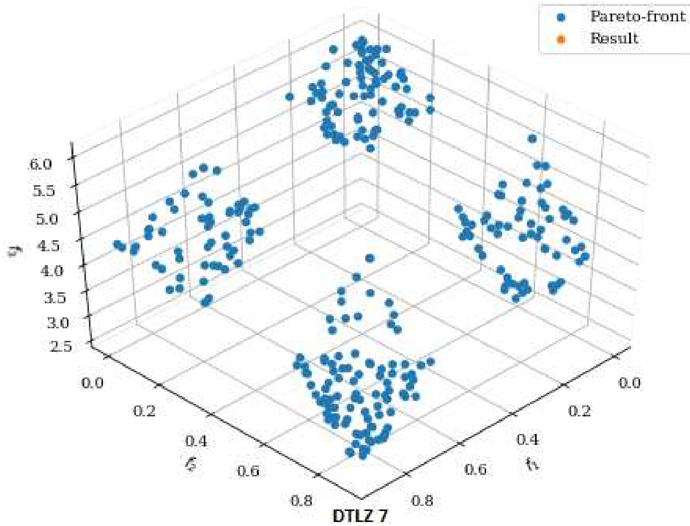
(b) **Figure 5.1.d.** Pareto front and optimal solution set for DTLZ4



(a) **Figure 5.1.e.** Pareto front and optimal solution set for DTLZ5



(b) **Figure 5.1.f.** Pareto front and optimal solution set for DTLZ6



(a) **Figure 5.1.g.** Pareto front and optimal solution set for DTLZ7

Figure 5.1. Pareto-Fronts extracted by LOF for problems DTLZ1 to DTLZ7

Below, in Figure 5.2, the average IGD values obtained by LOF and its competitors for DTLZ1 to DTLZ6 test cases are presented. It can be seen that the proposed method, shown in black, is either the best one or very close to the best one.

Table 5.3 illustrates the Friedman Aligned Ranks Test values calculated based on the IGD scores of 6 state-of-the-art algorithms and the new framework to find the order of the LOF among its competitors. Likewise, calculating the statistical similarity of results obtained by LOF to competitors [16], [17] is presented. Consequently, Table 5.4 shows the average of rank values, FAR and p-values for all algorithms.

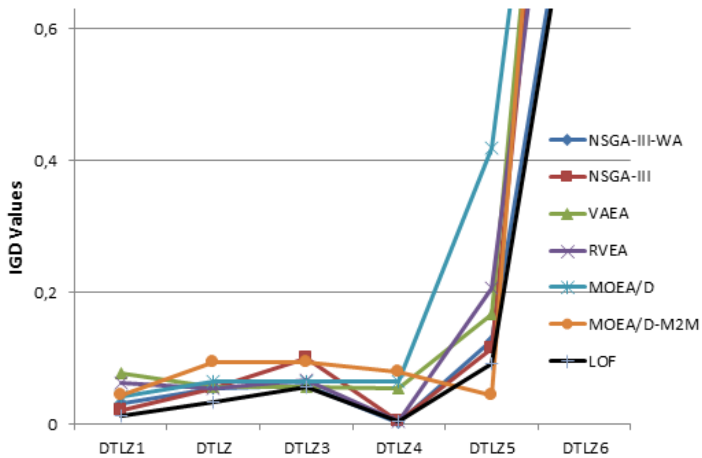


Figure 5.2. Average IGD values provided by LOF and its competitors for DTLZ test cases

Table 5.3. Calculated ranks of Friedman aligned for all pairs of DTLZ problems and Methods

Function	NSGA-III-WA	NSGA-III	VAEA	RVEA	MODA/D	MODA/D-M2M	LOF
DTLZ1	18	14	35	29	25	27	8
DTLZ2	22	21	24	23	28	34	13
DTLZ3	17	32	15	20	19	30	16
DTLZ4	9	12	31	10	33	37	11
DTLZ5	7	6	26	36	40	3	5
DTLZ6	2	39	41	4	38	42	1
Sum	75	124	172	122	183	173	54
AVG	12.50	20.67	28.67	20.33	30.50	28.83	9.00

According to Table 5.3, the average rank value of LOF is 9.00 which is the smallest one (i.e., the best performing algorithm). On the other hand, the p-value of LOF is also very close to zero which shows the remarkable statistical difference between the results of LOF and its competitors.

Table 5.4. Friedman Aligned Rank and p-value computed for all the methods for DTLZ problems

Algorithms	**Average values of Friedman Aligned Ranks overall problem instances**
NSGA-III-WA	12.50
NSGA-III	20.67
VAEA	28.67
RVEA	20.33
MODA/ D	30.50
MODA/ D-M2M	28.83
LOF	9.00
$\mathbf{F_{AR}}$	**15.054483**
P-value	**0.012837**

Finally, the proposed framework is evaluated over some three-objective WFG instances to prove the superiority of LOF when compared to its competitors.

Table 5.5 illustrates the rank of each metaheuristic used in Loser-Out-Framework for WFG test cases. According to Table 5.5, R-NSGA-III is the best performed method winning 2 out of four problems, and both NSGA-III and U-NSGA-III take the second position winning 1 out of 4 problems.

Table 5.5. Rank of the metaheuristics used in Loser-Out-Framework applied on three-objectives WFG

Function	MOEA/D	NSGA-III	R-NSGA-III	U-NSGA-III
WFG1	2	4	3	1
WFG2	4	1	2	3
WFG3	2	4	1	3
WFG4	4	3	1	2

Table 5.6 illustrates the average IGD scores for four three-objective WFG instances of LOF and its six recently published state-of-the-art competitors [8]-[12]. Based on Table 5.6, LOF takes the first position for WFG3 and WFG4, the second position – for WFG1 and the third position – for WFG3 test instances. It means that again in 50% of the test cases LOF takes the first position and for the rest of the cases it performs very close to the best ones.

Table 5.6. IGD values obtained by LOF and its six competitors for three-objectives WFG

Function	NSGA-III-WA	NSGA-III	VAEA	RVEA	MODA/D	MODA/D-M2M	LOF	LOF Rank out of 7
WFG1	1.171 ±0.2727	1.370 ±0.3356	1.324 ±0.2315	1.047 ±0.2417	1.216 ±0.2173	1.211 ±0.3725	1.157 ±0.2820	2
WFG2	0.2149 ±0.0613	0.2839 ±0.1040	0.3218 ±0.0893	0.3157 ±0.0436	1.317 ±0.0701	0.3714 ±0.0482	0.293 ±0.0451	3
WFG3	0.2163 ±0.0264	0.3791 ±0.0816	0.1489 ±0.0069	0.1977 ±0.0328	0.1793 ±0.0303	0.2361 ±0.0462	0.144 ±0.0272	1
WFG4	0.2043 ±0.0022	0.2147 ±0.0003	0.2317 ±0.0073	0.2272 ±0.0037	0.2475 ±0.0037	0.3581 ±0.0031	0.198 ±0.0008	1

Table 5.7 and Table 5.8 show Friedman Aligned Ranks Test, average of rank values, FAR and p-values for the proposed framework and its six competitors. According to Table 5.7, the average rank value of LOF is 5.00 which is the smallest one (i.e., the best performing algorithm). Small p-value of LOF also shows the remarkable statistical difference between the results of LOF and its competitors.

Table 5.7. Calculated ranks of Friedman aligned for all pairs of WFG problems and Methods

Function	NSGA-III-WA	NSGA-III	VAEA	RVEA	MODA/D	MODA/D-M2M	LOF
WFG1	11	7	27	21	18	19	1
WFG2	15	14	17	16	20	26	6
WFG3	10	24	8	13	12	22	9
WFG4	2	5	23	3	25	28	4
Sum	38	50	75	53	75	95	20
AVG	9.50	12.50	18.75	13.25	18.75	23.75	5.00

6 Conclusions and future works

This paper presents a new framework, called the Loser-Out-Framework (LOF), to solve MOO problems. LOF consists of four different and robust metaheuristics cooperating with each other to collect Non-Dominated Solutions (NDS) in a global archive and also to improve subpopulations. These four metaheuristics are also in competition to survive. At the end, all Non-Dominated Solutions found by all metaheuristics are combined and then a constant number of non-dominated solutions are selected as a final result. Thereafter for comparison, the IGD values are computed. The effectiveness of the proposed method is tested over three-objective DTLZ and WFG benchmark instances, and its performance is comparatively evaluated against the well-known modern MOO algorithms. The success of the proposed approach on the most of test problems in comparison to state-of-the-art algorithms indicates the efficiency of LOF in solving MOPs and puts it in the first position against its competitors. For the future, it is planned to extend this framework by adding more well-performed metaheuristics (consequently more phases) and evaluate it by applying to some real-world problems.

Table 5.8. Friedman Aligned Rank and p-value computed for all the methods for WFG problems

Algorithms	**Average values of Friedman Aligned Ranks overall problem instances**
NSGA-III-WA	9.5
NSGA-III	12.5
VAEA	18.75
RVEA	13.25
MODA/ D	18.75
MODA/ D-M2M	2375
LOF	5
F_{AR}	**9.5901**
P-value	**0.014300**

References

[1] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[2] B. Julian, D. Kalyanmoy, and C. R. Proteek, "Investigating the normalization procedure of NSGA-III," in *Evolutionary Multi-Criterion Optimization, 10th International Conference, EMO 2019, East Lansing*, (MI, USA), 2019, pp. 229–240.

[3] Z. Qingfu and L. Hui Li, "A multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[4] Y. Vesikar, K. Deb, and J. Blank, "Reference point based NSGA-III for preferred solutions," *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1587–1594, 2018.

[5] H. Seada and K. Deb, "A unified evolutionary optimization procedure for single, multiple, and many objectives," *IEEE Transactions on Evolutionary Computation*, pp. 358–369, 2016.

[6] S. Huband, P. Hingston, L. Barone, and L. While, "A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit," *IEEE Trans. On Evolutionary Computation*, vol. 10, pp.477–506, 2006.

[7] Y. Wang and X. Sun, "A Many-Objective Optimization Algorithm Based on Weight Vector Adjustment," *Computational Intelligence and Neuroscience*, vol. 2018, 2018. Article ID 4527968.

[8] M. Ojha, K. P. Singh, P. Chakraborty P, and S. Verm, "An Aggregation Based Approach with Pareto Ranking in Multiobjective Genetic Algorithm," in *Proceedings of Fifth International Conference on Soft Computing for Problem Solving. SocProS 2015, Volume 2* (Advances in Intelligent Systems and Computing, vol. 437), Springer Singapore, 2016, pp. 261–271. eBook ISBN: 978-981-10-0451-3.

[9] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, CEC 2009," *Proc. of the IEEE Eleventh Conf. on Evolutionary Computation*, pp. 203–208, 2009.

[10] W. Li, L. Wang, Q. Jiang, X. Hei, and B. Wang, "Multiobjective cloud particle optimization algorithm based on decomposition," *Algorithms*, vol. 8, no. 2, pp.157-176, 2015.

[11] W. Mashwani and A. Salhi, "Multiobjective memetic algorithm based on decomposition," *Applied Soft Computing*, vol. 21, pp. 221–243, 2014.

[12] K. Li, R. Wang, T. Zhang and H. Ishibuchi, "Evolutionary Many-Objective Optimization: A Comparative Study of the State-of-the-Art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018. DOI: 10.1109/ACCESS.2018.2832181.

[13] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[14] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, pp. 131–152, 2017.

[15] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 577–601, 2014.

[16] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC2009 unconstrained MOP test instances," in *IEEE Congress on Evolutionary Computation – CEC2009*, pp. 203–208, 2009.

[17] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3–18, 2011.

[18] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput*, vol. 13, pp. 284–302, 2009.

[19] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive Replacement Strategies for MOEA/D," *IEEE Trans. Cybern*, pp. 474–486, 2016.

[20] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *IEEE Congr. Evol. Comput*, pp. 203–208, 2009.

[21] I. Ahmadianfar, A. Adib, and M. Taghian, "A Multi-Objective Evolutionary Algorithm Using Decomposition (MOEA/D) And Its Application In Multipurpose Multi-Reservoir Operations," *International Journal Of Optimization In Civil Engineering*, pp. 167–187, 2015.

[22] H. L. Liu, F. Gu, and Z. Quingfu, "Decomposition of a Multiobjective Optimization Problem into a Number of Simple Multiobjective Subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

Jamshid Tamouk, Nasser Lotfi Received July 26, 2020

Jamshid Tamouk
Faculty of Engineering, Eastern Mediterranean University,
Famagusta, N. Cyprus via Mersin 10, Turkey
E–mail: jamshid.tamouk@cc.emu.edu.tr

Nasser Lotfi
Faculty of Engineering, Cyprus Science University, Girne, N.
Cyprus via Mersin 10, Turkey
E–mail: nasser.lotfi2020@gmail.com