Journal of Industrial and Systems Engineering
Vol. 15, No. 1, pp. 69-87
Winter (January) 2023

JISE

A novel dynamic multi-objective optimization algorithm based on EFO and quantum mechanics

Ataollah Taghaddosi[1], Mohammad Ali Afshar Kazemi[2*], Arash Sharifi[3], Mohammad Ali Keramati[2], Amir Daneshvar[4]

[1] Department of Information Technology Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[2] Department of Industrial Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[3] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[4] Department of Information Technology Management, Electronic Branch, Islamic Azad University, Tehran, Iran.

a.taghaddosi@gmail.com, dr.mafshar@gmail.com, a.sharifi@srbiau.ac.ir, m.keramati@gmail.com, a_daneshvar@iauec.ac.ir

Abstract

Considering the extensive application of dynamic multi-objective optimization problems (DMOPs) and the significance of the quality of solutions, developing optimization methods to find the finest solutions takes a privileged position, attracting considerable interest. Most optimization methods involve multiple conflicting objectives that change over time. The present article develops an electromagnetic field optimization (EFO) using decomposition, crowding distance, and the quantum behavior of particles techniques to solve multi-objective problems. In the proposed algorithm, the position of new particles is determined between the neighbors within the MOEA/D by drawing inspiration from the quantum delta potential well model, the nonlinear trajectory of quantum-behaved particles, and the interactions of electromagnetic particles introduced from positive and negative fields, which can offer superior exploration and exploitation. To develop the proposed algorithm for solving dynamic problems, the mean difference between particles' center of mass in the two latest changes to predict the extent of change is applied along with polynomial mutation and random reproduction. A total of 9 benchmarks from the set of DF functions and two metrics, i.e., MIGD and MHV, are used to assess the performance of the proposed algorithm. The results from 20 independent runs of the proposed algorithm on each benchmark function are compared with the results from other algorithms. The Wilcoxon Rank-Sum non-parametric statistical test is applied at the significance level of 5% to compare the mean results. The experimental results indicated that the proposed algorithm gains a significant superiority in metrics MIGA and MHV in most experiments. The simultaneously great results of these two metrics indicate a superior distribution and approximation of the Pareto front.

Keywords: Dynamic, multi-objective optimization, electromagnetic field optimization (EFO), quantum mechanics

*Corresponding author
ISSN: 1735-8272,

1-Introduction

We should make different decisions in our life, which are accompanied by problem-solving and data analysis. The best solution for these problems is usually searched, known as finding the optimal point or simply "optimization." Solving optimization problems requires advanced optimization techniques most of the time. Nature-inspired algorithms can be a good and efficient option for solving these problems. It is not an exaggeration if saying that optimization exists everywhere, from engineering design to commercial planning. Since resources are always limited in the real world, solutions must be found for the optimal use of these resources under different constraints. Because of the nonlinear nature of most real-world problems, complicated optimization tools are required to deal with them (Yang, 2020).

Classical optimization algorithms are insufficient in large scale combinatorial problems and in nonlinear problems. Hence, metaheuristic optimization algorithms have been proposed. General purpose metaheuristic methods are evaluated in nine different groups: biology-based, physics-based, social-based, music-based, chemical-based, sport- based, mathematics-based, swarm-based, and hybrid methods which are combinations of these (Akyol and Alatas, 2017) and (Chobar et al., 2022). Each group can be divided into other sub-groups. For instance, Alatas and Bingol (2020) and Zandbiglari et al., (2023) introduce two ray optimization and optics-inspired optimization algorithms, as light-based intelligent optimization algorithms and analyses their performance.

On the other hand, optimization algorithms are evaluated in two different groups: single-objective and multi-objective. The majority of problems in the real world have different inconsistent objectives. In such problems, instead of a single optimal solution, there are a set of optimal solutions that are not superior to each other if all objectives are considered.

When more than one objective is considered, a metaheuristic algorithm cannot compare solutions. In this situation, researchers have compared solutions using the Pareto dominance operator (Asrari et al. 2015) and (sharifzadegan and Pourghader Chobar, 2022). The mechanism of most metaheuristic multi-objective algorithms is almost the same. An important component is an archive or repository to store dominated solutions during the optimization. Multi-objective explorations continuously update this archive to improve the non-dominated solution in the archive. Another responsibility is finding different non-dominated solutions that are uniformly distributed between all objectives (Deb, 2014), (Pourghader Chobar et al., 2021) and (Eshghali et al., 2023).

In the literature, many algorithms have been presented for solving multi-objective problems. One of the most focused algorithms is the non-dominated sorting genetic algorithm (NSGA-II) which uses non-dominated sorting methods and crowding distance (Deb et al., 2002), (Pourghader Chobar et al., 2022) and (Asgari et al., 2022). Simulation results have shown better expansion of solutions and better convergence close to the Pareto optimal front for this algorithm relative to PAES and SPEA. Another outlined algorithm is the multi-objective particle swarm optimization (MOPSO) (Coello and Lechuga, 2002 May) and (Maadanpour Safari et al., 2021). In this algorithm, a repository is designed that stores non-dominated solutions. The repository size must be kept constant at the required level for which the adaptive network mechanism is used. So, for a larger number of non-dominated solutions in a specific part of the network, it is more probable to eliminate a solution and less probable to select the global optimum from these solutions.

Some conventional methods decompose multi-objective problems into several single-objective problems to solve them conveniently. To this end, Zhang and Li (2007) proposed a multi-objective evolutionary algorithm based on decomposition (MOEA/D), which initially decomposes a multi-objective problem into some sub-problems and then simultaneously optimizes these sub-problems. Every sub-problem is optimized only using information about its adjacent sub-problem. Empirical results showed that MOEA/D with simple decomposition methods outperforms MOGLS and NSGA-II in continuous multi-objective optimization problems. This method has been the centre of attention. Zhang et al. (2009, May) designed a numerical resource allocation strategy that focused the algorithm on more promising sub-problems.

If the optimization problem is time-dependent, the optimization will be dynamic. Deb et al. (2007, March) extended NSGA-II to solve dynamic multi-objective problems and named it dynamic NSGA-II (DNSGA-II). This algorithm employs polynomial mutation for the optimization process, non-dominated sorting for selecting non-dominated solutions, and crowding distance concept to broaden solutions' dispersion. In order to recognize variations in the environment, this algorithm selects some

population members randomly and re-evaluates their objective functions. If the environment changes, a percentage of population members are randomly selected and replaced by random solutions (DNSGA-II-A), or some are randomly selected and substituted for their previous value after applying mutation to them (DNSGA-II-B).

Zeng et al. (2006, July) proposed a dynamic multi-objective evolutionary algorithm (DOMOEA) for solving dynamic multi-objective problems. This algorithm uses orthogonal crossover and linear crossover operators to produce offspring. If the number of obtained solutions exceeds the intended population, a sorting method named crowding distance is used to eliminate excessive solutions. Some members of the population are randomly selected and reevaluated to detect changes in the environment. All population members are reevaluated in the new environment if a change occurs.

Du and Li (2008) developed a new algorithm for adapting to environmental variation by modifying PSO. In this algorithm, all particles are divided into two parts, and two new strategies, Gaussian local search for enhance the convergence ability and differential mutation for extend the searching area, are introduced into these two parts. This can enhance the ability of catching up with the changing optimum. This algorithm has been compared with other algorithms, and results have confirmed the better performance of the proposed algorithm.

Physics is a natural science with infinite fascinating phenomena. Quantum mechanics is a theory in physics that describes many aspects of nature at small (atomic and subatomic) scales. Sun et al. (2004, June) proposed a new strategy based on quantum mechanics. Inspired by the convergence analysis of PSO, they studied a single particle of the PSO system, which moved in multi-dimensional quantum space and developed a quantum-delta potential well model of PSO. Then, they proposed a trial method of parameter control for this model. The experiment result shows many advantages of quantum-behaved particle swarm optimization (QPSO) to the conventional PSO. Wang et al. (2007) introduced a new version of PSO based on quantum angles and their update using PSO. Their results indicate the much better performance of this algorithm relative to comparative algorithms. Inspired by concepts in quantum mechanics, Li et al. (2012) presented a cooperative QPSO, where a particle firstly obtains several individuals using the Monte Carlo method, and these individuals cooperate between them. Experiment results indicate that this algorithm outperforms other benchmark algorithms.

While there are ample successful optimization algorithms, the philosophy of scientific advancement requires a constant search for designing better methods. We still lack an algorithm that can provide the best results for each and every problem. Therefore, it is crucial to propose modifications or new approaches. Along the same line, the present research seeks to present an algorithm to solve dynamic and multi-objective optimization problems efficiently.

Electromagnetic field optimization (EFO) (Abedinpourshotorban et al., 2016), is a relatively new physics-based algorithm, outperforms widely-known single-objective optimization algorithms. However, it still lacks a dynamic and multi-objective version. In this research, we present a method to develop an electromagnetic field algorithm for solving dynamic and multi-objective problems efficiently. This method uses the techniques and concepts of decomposition, crowding distance, the quantum behavior of particles, and new location prediction. The proposed algorithm is inspired by the quantum delta potential well model and the nonlinear trajectory of quantum-behaved particles in Sun (2004, June), integrating them with electromagnetic field optimization. While this method offers an excellent rate of convergence, an accelerated rate of convergence implies a rapid decline in diversity. This decreased diversity can result in the aggregation of electromagnetic particles in a single region, which contradicts our objective to achieve a well-distributed Pareto set. Considering the significance of diversity in multi-objective problems, the introduction of electromagnetic particles from positive and negative fields based on crowding distance offers the chance of detecting a better-distributed Pareto set. A fine combination of diversity and convergence, which ultimately gives fine solutions, is provided through the interaction and competition between these particles in the quantum delta potential well model between the neighbors within the MOEA/D and based on the decomposition technique and Tchebycheff approach to determine the position of new particles. For extending the proposed method to solve dynamic problems as well, the average difference in the centre of gravity of particles in two previous changes is used to predict the new change.

In the following, the second section presents those works related to this study, including EFO, MOEA/D using the Tchebycheff approach, and QPSO. The third section outlines the proposed multi-

objective algorithm and applied approaches. In the fourth section, the simulation and experiment results are analysed. Finally, the fifth section describes the conclusions.

2-Related works

2-1-Electromagnetic field optimization

EFO is a meta-heuristic algorithm inspired by physics and the behavior of electromagnets with different polarities and takes advantage of a nature-inspired ratio known as the golden ratio. In this algorithm, a possible solution is assumed to be an electromagnetic particle made of electromagnets, and the number of electromagnets is determined by the number of variables of the problem. EFO is a population-based algorithm where the population is divided into three fields (positive, negative, and neutral). Attractive and repulsive forces among electromagnets of these three fields direct particles toward global minima. The golden ratio determines the ratio between attraction and repulsion forces to help particles converge quickly and effectively (Abedinpourshotorban et al., 2016).

In contrast to a permanent magnet, an electromagnet has a single polarity determined by the electrical current direction. There are two different forces among electromagnets: attraction and repulsion. Electromagnets with the same polarity repel each other, and those with opposite polarity attract each other. The proposed algorithm utilizes these concepts and replaces the ratio between attraction and repulsion forces with the golden ratio, which helps particles to adequately explore the problem search space and find a near-optimal solution (Abedinpourshotorban et al., 2016).

This algorithm is population-based, and each solution vector is represented by one group of electromagnets. The number of electromagnets of an electromagnetic particle is determined by the number of variables of the optimization problem. Moreover, all electromagnets of the same electromagnetic particle have the same polarity. However, each electromagnet can apply a force of attraction or repulsion on the peer-electromagnets that correspond to the same variable of the optimization problem (Eshghali et al., 2023).

In this algorithm, a population of electromagnetic particles is initially generated randomly, and a fitness function evaluates the fitness of each particle; then, particles are sorted according to their fitness. Second, sorted particles are divided into three groups. The first group, the positive field, consists of the fittest electromagnetic particles with positive polarity, and the second group, the negative field, consists of the electromagnetic particles with the lowest fitness and negative polarity. The remaining electromagnetic particles form a group called the neutral field, which has a slight negative polarity almost near zero. Figure 1 illustrates the force direction between electromagnets (Abedinpourshotorban et al., 2016).

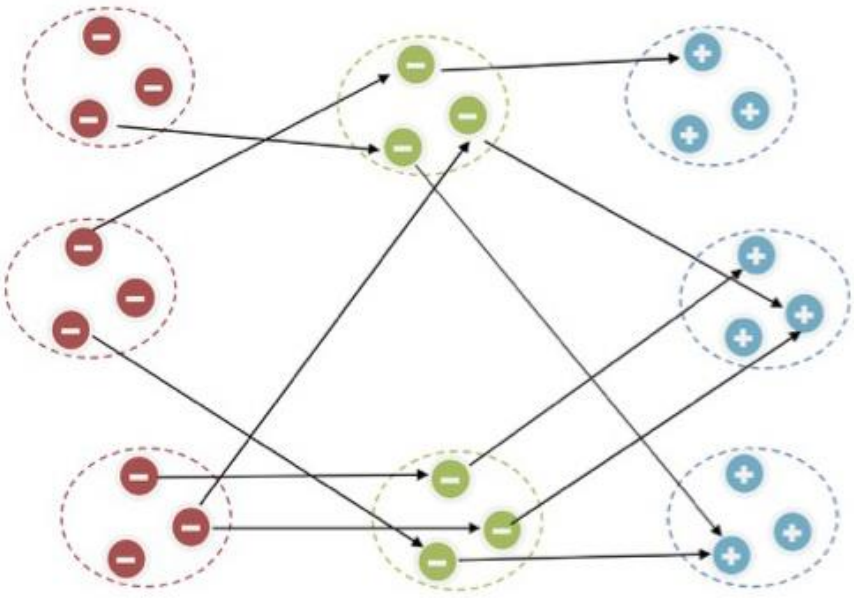

Fig 1. Force direction among electromagnets (Abedinpourshotorban et al., 2016)

In each iteration of the algorithm, a new electromagnetic particle is shaped and evaluated by a fitness function. If the generated electromagnetic particle is fitter than the worst particle in the population, then the generated particle will be inserted into the sorted population according to its fitness and obtain a polarity based on its position in the population; moreover, the worst particle will be eliminated. This process continues until the termination conditions are met. The coexistence of two opposite forces among electromagnets and the fact that the new solution is generated by moving a distance away from bad solutions and moving closer to the good solutions cause effective search and fast convergence. However, to keep diversity and avoid local minima, randomness is an indispensable part of this algorithm. Therefore, for some of the generated electromagnetic particles, only one of the electromagnets is changed with a randomly generated electromagnet (Abedinpourshotorban et al., 2016).

For developing the search strategy in this algorithm, a randomly selected electromagnet from the neutral field is affected (virtually) by selected electromagnets from the positive field (attraction) and negative field (repulsion) to determine the position of the generated electromagnet. The new position is calculated as equation (1).

$$EMP_j^{New} = EMP_j^{K_j} + \left((\varphi * r) * \left(EMP_j^{P_j} - EMP_j^{K_j}\right)\right) - \left(r * \left(EMP_j^{N_j} - EMP_j^{K_j}\right)\right) \quad (1)$$

Where EMP is the electromagnetic particle; r is the random value between 0 and 1 (generated once for each new electromagnetic particle; P is the random index from the positive field (generated for each electromagnet of the generated particle); N is the random index from the negative field.

The general flow of EFO is presented in figure 2. Ps-rate is the probability of selecting electromagnets of the generated electromagnetic particle from electromagnets of the positive field without changing them, and R_rate is the possibility of changing one electromagnet of the generated electromagnetic particle with a randomly generated electromagnet. Also, N_Var represents the number of variables of the problem.

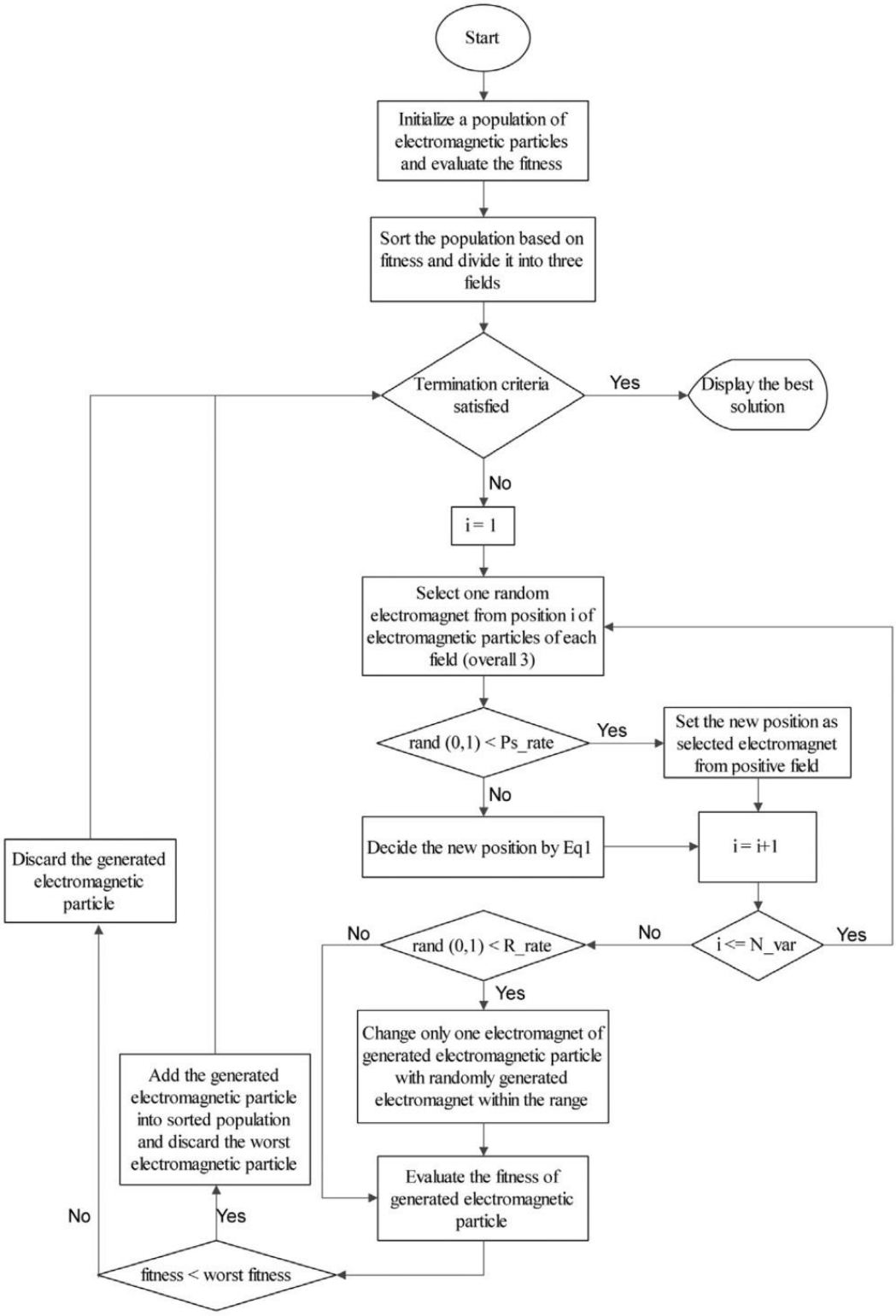

Fig 2. General flow of EFO (Abedinpourshotorban et al., 2016)

2-2-MOEA/D based on Tchebycheff approach

In the Tchebycheff approach, $\lambda^1, \ldots, \lambda^n$ is the weight vector, and z^* is the reference point. This approach can decompose the Pareto front approximation sub-problem into N scalar optimization problems, and the objective function of the j^{th} sub-problem is defined as equation (2) (Zhang and Li, 2007).

$$g^{te}(x|\lambda^j, z^*) = \max_{1\le i\le m}\{\lambda_i^j|f_i(x) - z_i^*|\} \quad (2)$$

Where $\lambda^j = (\lambda_1^j, \dots, \lambda_n^j)^T$. MOEA/D minimizes all objective functions simultaneously in one run. In this algorithm, the neighborhood of the weight vector λ^i is defined as a set of several adjacent weight vectors in $\{\lambda_1, \dots, \lambda_n\}$. The population includes the obtained best solution for sub-problems. For optimization of a sub-problem in MOEA/D, only the current solutions of its neighboring sub-problems are used. Information exchanges always exist between neighborhoods in MOEA/D. Each sub-problem is optimized only using the information of sub-problems in its neighborhood. The main steps of this algorithm are as follows:

Step 1: Initialization

Set archive equal to null. Calculate the Euclidean distance between each pair of weight vectors. Then, specify the closest weight vectors to every weight vector. Suppose $B(i) = \{i_1, \dots, i_T\}$ for $i = 1, \dots, N$, where T represents the closest weight vectors.

Afterward, generate an initial population, calculate their objective functions, and initialize the ideal vector z.

Step 2: Updating

Generate a new solution (y) collaborating with neighboring solutions. Then, update the ideal vector z. Update the neighboring solutions so that if $g^{te}(y|\lambda^j, Z) \le g^{te}(x^j|\lambda^j, Z)$ for each $j \in B(i)$, then $x^i = y$. Lastly, update the archive.

Step 3: Stop if the termination conditions are met; otherwise, return to step 2 (Zhang and Li, 2007).

2-3- Quantum-behaved particle swarm optimization

Inspired by the analysis of the convergence of PSO, this algorithm investigates a single particle from a system of particles in multi-dimensional quantum space and proposes a quantum delta-potential well model of PSO (Sun et al., 2004 June).

In quantum mechanics, the state of motion of a particle is given by wave function ψ. Schrodinger equation is expressed as equation (3) to present the state of a particle over time.

$$j\hbar\frac{\partial}{\partial t}\psi(x,t) = (-\frac{\hbar^2}{2m}\nabla^2 + V(x))\psi(x,t) \quad (3)$$

The potential energy of each particle in the one-dimensional delta potential well is defined by equation (4), and the position of each particle is expressed by equation (5). Point P is the center of the potential well, and L is the length of the potential well in which particles are placed. Parameter L is formulated as equation (6).

$$V(x) = -\gamma * \delta(x - p) \quad (4)$$

$$x = p \pm \frac{L}{2} ln(\frac{1}{u}) \quad (5)$$

$$L(t) = \frac{1}{g}.|X_k(t) - P| \quad (6)$$

So, the position of each particle is given by equation (7).

$$X_{i,j}(t+1) = P_{i,j}(t) \pm \beta.\left|P_{i,j}(t) - X_{i,j}(t)\right|.ln(\frac{1}{u}) \quad (7)$$

2-4-Dynamic multi-objective optimization algorithm based on EFO and quantum mechanics

The strategies and parameters applied in the proposed algorithm for solving multi-objective problems include Tchebycheff decomposition, crowding distance, and particle modeling in the delta potential well.

The proposed algorithm is developed using the general framework of MOEA/D (Zhang and Li, 2007). First, weight vectors λ are generated with a uniform distribution to all population members. Then, the Euclidean distance between these vectors is calculated, and based on this distance, the closest weight vectors to a specific weight vector are specified. Each particle $i = (1, \dots, N)$ has a set of neighbors $B(i) = \{i_1, \dots, i_T\}$, where T represents the closest weight vectors. Afterward, an initial population is generated, their objective functions are determined, and ideal vector Z is initialized. In the collaboration phase, different methods can be applied to generate a new solution. In the proposed algorithm, the position of particles is determined by getting inspired by the quantum delta-potential well model and nonlinear movement of quantum-behaved particles (Sun et al., 2004 June), as well as the interaction of electromagnetic particles introduced from positive and negative fields and cooperation between neighbors.

The positive field is created in every execution of the algorithm through the collection of particles, which are non-dominated or compromised relative to other particles and each other and have the largest crowding distance. The variable P_field determines the ratio of the number of these particles to the non-dominated ones. Also, the negative field is generated in every execution of the algorithm from collecting those particles, which are dominated relative to a single or some particles or each other and have the smallest crowding distance. N_field represents the ratio of the number of these particles to all dominated particles. Therefore, sorting all particles in terms of crowding distance in each iteration is required. The population is first sorted in ascending order to calculate the crowding distance based on the objective function value. Then, the distance is considered equivalent to infinity for each dimension's first and last solutions. For the middle solutions, the crowding distance is considered equal to the normalized difference of the value of the objective function between two adjacent neighbors. The overall value of the crowding distance is the sum of distances obtained in each dimension.

In the proposed strategy, P is an electromagnetic particle from the positive field, and N is an electromagnetic particle from the negative field. Equations (8) and (9) define the positions associated with the former and latter particles.

$$P_j = \frac{\left(\phi_1 * {EMP_j}^{p_j} + \phi_2 * {EMP_j}^{3j}\right)}{\phi_1 + \phi_2} \tag{8}$$

$$N_j = \frac{\left(\phi_1 * {EMP_j}^{n_j^1} + \phi_2 * {EMP_j}^{n_j^2}\right)}{\phi_1 + \phi_2} \tag{9}$$

Where EMP^p is the random electromagnetic particle from the positive field, and EMP^3 is the third particle with the largest crowding distance in the positive field, and EMP^{n^1} and EMP^{n^2} are the random electromagnetic particles from the negative field. j is the variable index (index of generated electromagnet). Also, ϕ_1 and ϕ_2 represent random numbers between 0 and 1.

The position of two new electromagnetic particles, with inspiration from the quantum delta-potential well model (Sun et al., 2004 June), as well as interaction of electromagnetic particles from positive and negative fields in collaboration with neighbors' k and l, is determined according to equations (10) and (11). In these equations, φ is a random number in the range of (0, 1), u is another random number in the range of (0,1], and g is the golden ratio (approximately 1.61). The new electromagnets are generated with the probability of Pg_rate.

$$X_j^i(t+1) = P_j(t) + \varphi * \left(g * (P_j(t) - EMP_j^k(t)) - (N_j(t) - EMP_j^k(t))\right).ln(\frac{1}{u}) \tag{10}$$

$$Y_j^i(t+1) = P_j(t) + \varphi * \left(g * (P_j(t) - EMP_j^l(t)) - (N_j(t) - EMP_j^l(t))\right).ln(\frac{1}{u}) \tag{11}$$

Where EMP^k is a k-neighboring electromagnetic particle, and EMP^l is an l-neighboring electromagnetic particle. With the contribution of two neighbors, k and l, the position of a new particle is obtained from equation (12):

$$EMP_j^i(t+1) = \frac{\left(\phi_1 * X_j^i(t+1) + \phi_2 * Y_j^i(t+1)\right)}{\phi_1 + \phi_2} \quad (12)$$

The new particle is evaluated, and ideal point Z is updated. In the competition phase, if $g^{te}(EMP^i(t+1)|\lambda^j, Z) \leq g^{te}(EMP^j(t)|\lambda^j, Z)$ for each $j \in B(i)$, then $EMP^j(t) = EMP^i(t+1)$.

Then, each particle is compared with others, and if any other particle does not dominate, it is assigned to the positive pole. Suppose the memory of the positive pole is filled. In that case, the second measure (crowding distance) is employed to identify those particles with more contribution in the approximate Pareto distribution, and particles with lower preference are eliminated. Also, particles, which are dominated by even one other particle, are inserted into the negative pole.

At the end of each iteration, all particles in the positive pole are compared, and dominated particles are removed. Then, the remaining particles are sorted with their crowding distance.

For developing an algorithm to solve time-dependent problems, in order to recognize changes in the environment, ten percent of the population with uniform probability distribution is selected randomly, and their objective functions are evaluated. This portion of the population is named the detection population. In each iteration, if some members of the detection population confirm the change by reevaluating the objective function and comparison with the value of the objective function in the previous iteration, the algorithm starts the change adaptation phase. In this phase, the average position of particles is first calculated. Mutation and random regeneration are applied to half of the particles if the change has occurred for the first time. In this process, the polynomial mutation of Deb and Deb, (2014) is used, as expressed by equation (13):

$$EMP' = EMP + \overline{\delta}(Ub - Lb) \quad (13)$$

Where EMP' is mutated electromagnetic particle, Ub is the upper bound, and Lb is the lower bound of the space in each dimension, and δ is given by equation (14):

$$\begin{cases} \overline{\delta} = (2 * u)^{\frac{1}{1+\eta_m}} - 1, & for\ u \leq 0.5. \\ \overline{\delta} = 1 - (2 * (1-u))^{\frac{1}{1+\eta_m}}, & for\ u > 0.5 \end{cases} \quad (14)$$

If the change has occurred for the second time, half of the particles shift their positions as much as the difference between two recent average points; another half is subjected to polynomial mutation and random regeneration and updated. Finally, in all the next changes, half of the particles shift their positions as much as the difference between two recent average points; another half is subjected to polynomial mutation and random regeneration and updated. Algorithm 1 indicates this process.

Begin
Initialize: the electromagnetic population (nEMP); the number of the subproblems (N); a uniform spread of N weight vectors (λ); the number of the Neighbors of each weight vector (T); portion of population, which belongs to positive field (P_field); portion of population, which belongs to negative field (N_field); probability of generating electromagnets (Pg_rate); golden ratio (g = 1.618); Estimated pareto front (EP).
EP ⟵ 0; Z ⟵ infinity;
D ⟵ Calculate the euclidean distances (λ);
Determine Neighbors (λ, D, T);
For i=1 **to** Population size **do**
    Pop_i position⟵ Random Position (Problem size);
    Pop_i Cost ⟵ Evaluate Pop_i;
    Z ⟵ Minimum (Z, Pop_i Cost);
End
For i=1 **to** Population size **do**
    Pop_i g⟵ Decomposed Cost (Pop_i Cost', Z, λ);
End
Pop ⟵ Determine domination (Pop);
EP ⟵ Population is not dominated and sorted base on crowding distance;

```
NPol ⟵ Population is dominated and sorted base on crowding distance;
DP ⟵ Create diagnostic random population (10% of population);
DP Cost Reference ⟵ Evaluate DP (DP);
Change Count ⟵ 0;
While Stop Condition () do
DP Cost ⟵ Evaluate DP (DP);
Change ⟵ Detect the change (DP Cost Reference , DP Cost);
If Change = True then
   Change Count⟵ Change Count + 1;
   MEP ChangeCount ⟵ Calculate the mean of EP population positions (EP);
   DP Cost Reference ⟵ DP Cost;
   If Change Count =1 then
      Pop ⟵ Gaussian Mutation on 50% of population (Pop);
   End
   If Change Count = 2 then
      D ChangeCount ⟵ Calculate the diferance of tow before MEP
      Pop ⟵ Add the D to 50% of population position (P, D);
      Pop ⟵ Gaussian Mutation on 50% of population (Pop);
   End
   If Change Count > 2 then
      D ChangeCount ⟵ Calculate the diferance of tow before MEP;
      MD ⟵ Calculate the mean of tow before D;
      Pop ⟵ Add the MD to 50% of population position (Pop, MD);
      Pop ⟵ Gaussian Mutation on 50% of population (Pop);
   End
Pop Cost, Z ⟵ Update (Pop, Z);
Pop ⟵ Determine domination (Pop);
EP ⟵ Population is not dominated and sorted base on crowding distance;
NPol ⟵ Population is dominated and sorted base on crowding distance;
End
For i=1 to Population size do
    (K, l) ⟵ Randomly select tow indexes from Neighbors (Neighbors);
    For j=1 to Number of variables do
        I_pos ⟵ Index of a random particle from positive field (1, floor (nEP * P_field));
        I_neg1,2 ⟵ Index of tow random particles from negative field ((floor (1 - N_field) * nNPol), nNPol);
        Pi positionj ⟵ Calculate by using equation 8 (EP(I_pos), EP (3));
        Ni positionj ⟵ Calculate by using equation 9 (NPol(I_neg1), NPol(I_neg2));
        if rand (0,1) < Ps_rate then
            Popi positionj ⟵ Generate a new solution Pop from Popk and Popl by using equation 10, 11 and 12
        End
    End
    Popi Cost ⟵ Evaluate Popi;
    Z⟵ Minimum (Z, Popi Cost);
    For j=1 to T do
        Pop j g⟵ Decomposed Cost (Popj Cost', Z, λ);
        Pop i g⟵ Decomposed Cost (Popi Cost', Z, λ);
        if Popi g <= Pop j g then
          Pop j g = Pop i g
        End
    End
End
Pop ⟵ DetermineDomination (Pop);
EP⟵ Population is not dominated and sorted base on crowding distance;
NPol⟵ Population is dominated and sorted base on crowding distance;
End While
Return EP;
End
```

Fig 3. The proposed algorithm 1

2-5-Setting DMOQEFO parameters

One of the important parameters is nEMP, which determines the number of electromagnetic particles in the population. For lower values of nEMP, adequate awareness of the search space is not realized,

and convergence will slow down for higher values. This parameter depends on the problem type, the number of dimensions, and the number of objectives. In conducted experiments in this study, it is empirically found that a population of almost 100 particles can give a suitable performance. The proposed algorithm has two groups with positive and negative polarities. P_field is the percentage of the population allocated to the positive field. Based on experience, the best value for P_field is between 0.1 and 0.15. N_field is the percentage of the population allocated to the negative field. Empirically, the best value for this parameter is between 0.4 and 0.5. Another important parameter is Pg_rate which determines the probability of generating a new electromagnet in the electromagnetic particle. Based on experiments, the best value for Pg_rate is between 0.2 and 0.4.

3-Results and discussion

This section describes the conducted experiments to examine the performance of comparative algorithms and their results. All the experiments are conducted under similar conditions. The algorithms are simulated on MATLAB software installed on a PC with Intel Core i5 CPU with 3 GHz, and RAM of 8 GB. Initializing searching solutions for all algorithms is done completely randomly and by 100 particles.

Initially, the parameters of examined algorithms are presented. Then, the applied benchmark functions are introduced. Afterward, the indices for measuring the performance of algorithms are explained. Finally, the results of conducted experiments are compared and analyzed.

3-1-Experimental parameters

All tested algorithms are individually executed 20 times on the benchmark functions, and their parameters are set as recommended in the original reference. The population size is assumed to be 100, and the number of decision variables is 10 in all experiments. Also, suppose the frequency of change (τ_t) to be 10 and 30 and the severity of change (n_t) to be 10. The termination condition of each individual run of the algorithms are 300 iterations. Other parameters of algorithms are presented in table 1.

Table 1. Parameters of algorithms used in experiments

Algorithms	Parameters
DMOQEFO	P_field=0.1, N_field=0.45, Pg_rate=0.25, g=1.618, T=40, EP=100, nEMP=100
DMOEA/D	Gamma=0.5, T=40, EP=100, nPop=100
DMOPSO	C_1=1, C_2=2, nGrid=20, Gamma=2, Mu=0.1, w=0.5, nRep=100, nPop=100
DNSGA II-A	pCrossover=0.7, MutationPercentage=0.4, MutationRate=0.02, nPop=100
DNSGA II-B	pCrossover=0.7, MutationPercentage=0.4, MutationRate=0.02, nPop=100

3-2-Benchmark functions

In all experiments, nine benchmark functions (DF) (Jiang et al., 2018), firstly introduced in 2018 for the CEC2018 competition on dynamic multi-objective optimization are used. The mathematical expressions of these functions and their search space are presented in table 2. These functions cover diverse properties, such as time-dependent PF/PS geometries, irregular PF shapes, disconnectivity, and knee, covering diverse properties, which nicely represent various real-world scenarios. In these problems, τ is the current iteration of the algorithm, τ_t is the frequency of change, and n_t is the severity of change.

Table 2. Benchmark test functions (Jiang et al., 2018)

Name	Formula	Search space
DF1 dynamic PF and PS	$$min\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)(1-(\frac{x_1}{g(x)})^{H(t)}) \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - G(t))^2,\ H(t) = 0.75 * sin(0.\pi t5) + 1.G\ ,25(t) = \lvert sin(0.\pi t5)\rvert.$$	$[0,1]^n$
DF2 static convex PF, dynamic PS, severe diversity loss	$$min\begin{cases} f_1(x) = x_r \\ f_2(x) = g(x)(1-\sqrt{\frac{f_1}{g}}) \end{cases}$$ $$g(x) = 1+\sum_{i=\frac{\{1,..,n\}}{\{r\}}}(x_i - G(t))^2,\ G(t) = \lvert sin(0.\pi t5)\rvert r\ , = 1+\lfloor(n-1)G(t)\rfloor.$$	$[0,1]^n$
DF3 dynamic PF and PS	$$min\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)(1-(\frac{x_1}{g(x)})^{H(t)}) \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - G(t) - x_1^{H(t)})^2,\ G(t) = sin(0.\pi t5), H(t) = 1.5 + G(t).$$	$[0,1]*[-1,2]^{n-1}$
DF4 dynamic PF and PS	$$min\begin{cases} f_1(x) = g(x)\lvert x_1 - a\rvert^{H(t)} \\ f_2(x) = g(x)\lvert x_1 - a - b\rvert^{H(t)} \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - \frac{ax_1^{\ 2}}{i})^2,\ a = sin(0.\pi t5), b = 1+\lvert cos(0.\pi t5)\rvert, H(t) = 1.5 + a.$$	$[-2,2]^n$
DF5 dynamic PF and PS	$$min\begin{cases} f_1(x) = g(x)(x_1 + 0.02\,sin(\omega_t \pi x_1)) \\ f_2(x) = g(x)(1-x_1 + 0.02\,sin(\omega_t \pi x_1)) \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - G(t))^2,\ G(t) = sin(0.\pi t5), \omega_t = \lfloor G10(t)\rfloor.$$	$[0,1]*[-1,1]^{n-1}$
DF6 dynamic PF and PS	$$min\begin{cases} f_1(x) = g(x)\left(x_1 + 0.1\,sin(\pi 3x_1)\right)^{\alpha_t} \\ f_2(x) = g(x)\left(1-x_1 + 0.1\,sin(\pi 3x_1)\right)^{\alpha_t} \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(\lvert G(t)\rvert y_i^{\ 2} - 10 * cos(\pi 2y_i) + 10),\ y_i = x_i - G(t),$$ $$G(t) = sin(0.\pi t5), \alpha_t = 0.2 + 2.8 * \lvert G(t)\rvert.$$	$[0,1]*[-1,1]^{n-1}$
DF7 convex PF, static PS centroid, dynamic PF and PS	$$min\begin{cases} f_1(x) = g(x)\frac{1+t}{x_1} \\ f_2(x) = g(x)\frac{x_1}{1+t} \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - \frac{1}{1+e^{\alpha_t(x_1-2.5)}})^2,\ \alpha_t = 5\,cos(0.\pi t5).$$	$[1,4]*[0,1]^{n-1}$
DF8 static PS centroid, dynamic PF and PS, variable-linkage	$$min\begin{cases} f_1(x) = g(x)(x_1 + 0.1\,sin(\pi 3x_1)) \\ f_2(x) = g(x)\left(1-x_1 + 0.1\,sin(\pi 3x_1)\right)^{\alpha_t} \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - \frac{G(t)\,sin(\pi 4x_1^{\ \beta_t})}{1+\lvert G(t)\rvert})^2,\ \alpha_t = 2.25 + 2.cos(\pi t2),$$ $$\beta_t G,1=(t) = sin(0.\pi t5).$$	$[0,1]*[-1,1]^{n-1}$
DF9 dynamic PS and PF, variable-linkage	$$min\begin{cases} f_1(x) = g(x)(x_1 + max\{0,(\frac{1}{2N_t}+0.1)\,sin(2N_t\pi x_1)\}) \\ f_2(x) = g(x)(1-x_1 + max\{0,(\frac{1}{2N_t}+0.1)\,sin(2N_t\pi x_1)\}) \end{cases}$$ $$g(x) = 1+\sum_{i=2}^{n}(x_i - cos(t4 + x_1 + x_{i-1}))^2,\ N_t = 1+\lfloor 10\lvert sin(0.\pi t5)\rvert\rfloor.$$	$[0,1]*[-1,1]^{n-1}$

3-3-Performance indices

Zitzler (1999) introduces the goals that must be achieved in a dynamic multi-objective optimization problem as the minimization of the distance between the non-dominated front and the Pareto optimal

front and the suitable distribution of solutions. To this end, two metrics presented by Zitzler (1999) are used to evaluate the performance of the algorithm proposed in this study.

3-3-1-Mimicked inverted generational distance (MIGD)

Inverted generational distance (IGD) gives the distance between each solution from the Pareto optimal front and the closest solution obtained from the test algorithms (Sierra and Coello, 2005 March). Mimicked inverted generational distance (MIGD) is adapted from the static IGD, and its lower values indicate the better performance of an algorithm. Suppose P_t is a set of uniformly-distributed points in real PF, and P_t^* is an approximation of PF at time t. MIGD is calculated as follows:

$$MIGD = \frac{1}{T}\sum_{i=1}^{T} IGD(P_t^*, P_t) = \frac{1}{T}\sum_{i=1}^{T}\sum_{i=1}^{np_t}\frac{d_t^i}{np_t} \quad (15)$$

Where $np_t = |P_t|$ is the Euclidean distance between the i[th] member of P_t and the closest member in P_t^*. It is expected a set of about 1000 points with uniform distribution from the real PF is used to calculate MIGD.

3-3-2-Mimicked hyper-volume (MHV)

MHV is adapted from the static hyper-volume metric and used to measure the region covered by obtained optimal solutions relative to a specific reference point. The higher the MHV, the better the algorithm performance (Guerreiro et al., 2020). MHV is expressed as follows:

$$MHV = \frac{1}{T}\sum_{i=1}^{T} HV_t(P_t^*) \quad (16)$$

Where HV (S) is the covered volume of set S relative to the reference point. The reference point for determining this metric is define as $(z_1 + 0.5, z_2 + 0.5, \dots, z_M + 0.5)$, where z_j is the value of the j^{th} objective function from real PF at time t, and M is the number of objectives.

3-4-Comparison with some competitor algorithms

This section considers four algorithms, including MOPSO (Coello and Lechuga, 2002 May), DNSGAII-B (Deb and Karthik, 2007 March), DNSGAII-A (Deb and Karthik, 2007 March), and MOEA/D (Zhang and Li, 2009), for comparison. An idea similar to DNSGAII is used to adapt MOPSO and MOEA/D with dynamic variations of problems so that after each change, half the population is subjected to polynomial mutation and random regeneration. In order to evaluate the performance of the proposed algorithm compared to others, a two-step experiment is designed.

In the first step, the comparative algorithms are applied to nine benchmark functions with a frequency of 10. In order to ensure the validity of results, each experiment is conducted 20 times, and the average and standard deviation of results are presented in table 3. In the second step, experiments are repeated with a frequency of 30, and the average and standard deviation of results from 20 individual executions of algorithms are presented in table 4. The best solution for each problem is highlighted in tables 3 and 4.

Table 3. Average and standard deviation of IGD after running each algorithm 20 times

Function	τ_t	DMOQEFO (mean (std.))	DMOEA/D (mean (std.))	DMOPSO (mean (std.))	DNSGAII-A (mean (std.))	DNSGAII-B (mean (std.))
DF1	10	**0.0193 (0.0156)**	0.1001 (0.0716)	0.0687 (0.0480)	0.1170 (0.1260)	0.1197 (0.1137)
	30	**0.0115 (0.0113)**	0.0908 (0.0805)	0.0393 (0.0410)	0.0780 (0.0704)	0.0831 (0.0679)
DF2	10	**0.0176 (0.0092)**	0.0884 (0.0386)	0.0543 (0.0324)	0.0627 (0.0691)	0.0708 (0.0735)
	30	**0.0089 (0.0037)**	0.0657 (0.0436)	0.0290 (0.0260)	0.0509 (0.0544)	0.0566 (0.0570)
DF3	10	**0.2506 (0.0956)**	0.4338 (0.1600)	0.3407 (0.1852)	0.2753 (0.1653)	0.2690 (0.1674)
	30	**0.2977 (0.0769)**	0.3674 (0.1406)	0.4152 (0.1153)	0.3285 (0.0748)	0.3023 (0.0654)
DF4	10	**0.1532 (0.0954)**	0.3073 (0.3534)	0.3836 (0.4394)	0.3449 (0.2382)	0.3104 (0.2046)
	30	**0.1493 (0.1092)**	0.1793 (0.1408)	0.2138 (0.1621)	0.6503 (0.3822)	0.7429 (0.4659)
DF5	10	**0.0211 (0.0190)**	0.1351 (0.0261)	0.1266 (0.0794)	0.4710 (0.9031)	0.4565 (0.8388)
	30	**0.0116 (0.0035)**	0.0700 (0.0475)	0.0473 (0.0267)	0.2746 (0.4097)	0.3505 (0.5489)
DF6	10	**0.4227 (0.8073)**	2.5311 (3.8942)	6.7757 (6.2990)	2.9801 (3.6601)	2.6937 (3.1918)
	30	**0.5407 (0.8903)**	5.2430 (4.6225)	2.2969 (2.2698)	0.8859 (0.3483)	0.8925 (0.4762)
DF7	10	0.4067 (0.2446)	**0.3640 (0.2765)**	0.4560 (0.2744)	0.6765 (0.1930)	0.6776 (0.1970)
	30	0.4976 (0.2166)	0.3118 (0.2688)	**0.1567 (0.0454)**	0.6409 (0.1098)	0.6452 (0.1160)
DF8	10	**0.0766 (0.0244)**	0.2011 (0.0720)	0.1261 (0.0472)	0.1307 (0.0803)	0.1291 (0.0818)
	30	**0.0690 (0.0300)**	0.1847 (0.0854)	0.0930 (0.0366)	0.1589 (0.1152)	0.1532 (0.1100)
DF9	10	**0.1543 (0.1582)**	0.2204 (0.0568)	0.3988 (0.3256)	0.2086 (0.0954)	0.2189 (0.1031)
	30	**0.0441 (0.0140)**	0.1598 (0.0530)	0.1349 (0.0365)	0.0622 (0.0231)	0.0693 (0.0273)

Table 4. Average and standard deviation of HV after running each algorithm 20 times

Function	τ_t	DMOQEFO (mean (std.))	DMOEA/D (mean (std.))	DMOPSO (mean (std.))	DNSGAII-A (mean (std.))	DNSGAII-B (mean (std.))
DF1	10	**0.4997 (0.0898)**	0.4097 (0.0745)	0.4283 (0.1138)	0.3976 (0.1201)	0.3944 (0.1138)
	30	**0.4669 (0.0315)**	0.4207 (0.0690)	0.4233 (0.0439)	0.4193 (0.0741)	0.4129 (0.0676)
DF2	10	**0.6876 (0.0137)**	0.5890 (0.0427)	0.6327 (0.0447)	0.6336 (0.0771)	0.6246 (0.0809)
	30	**0.7013 (0.0061)**	0.6302 (0.0508)	0.6710 (0.0379)	0.6549 (0.0570)	0.6481 (0.0556)
DF3	10	**0.2754 (0.1312)**	0.1451 (0.0819)	0.2020 (0.1636)	0.2554 (0.1577)	0.2620 (0.1618)
	30	0.1824 (0.0245)	0.1549 (0.0370)	0.1055 (0.0496)	0.2124 (0.0486)	**0.2257 (0.0473)**
DF4	10	**0.8370 (0.0574)**	0.8227 (0.1575)	0.7026 (0.2148)	0.7647 (0.1057)	0.7732 (0.1020)
	30	**0.8340 (0.0486)**	0.8118 (0.0905)	0.7842 (0.1289)	0.6723 (0.0475)	0.6596 (0.0606)
DF5	10	**0.5561 (0.0298)**	0.3965 (0.0303)	0.4047 (0.0957)	0.3552 (0.1682)	0.3483 (0.1774)
	30	**0.5726 (0.0074)**	0.4963 (0.0538)	0.5129 (0.0412)	0.4543 (0.1060)	0.4433 (0.1208)
DF6	10	**0.5376 (0.3042)**	0.1514 (0.1051)	0.0013 (0.0049)	0.0566 (0.0834)	0.0562 (0.0855)
	30	**0.5062 (0.3114)**	0.0580 (0.1037)	0.0275 (0.0576)	0.0667 (0.1054)	0.0906 (0.1267)
DF7	10	0.3465 (0.0458)	**0.3643 (0.0604)**	0.3022 (0.0576)	0.3065 (0.0292)	0.3081 (0.0308)
	30	0.3343 (0.0427)	0.3656 (0.0682)	**0.3738 (0.0221)**	0.3048 (0.0142)	0.3023 (0.0152)
DF8	10	0.5775 (0.1564)	0.6394 (0.0609)	0.5074 (0.1634)	0.7011 (0.0195)	**0.7039 (0.0208)**
	30	0.5853 (0.1659)	0.7005 (0.0297)	0.5566 (0.1596)	0.6985 (0.0467)	**0.7012 (0.0406)**
DF9	10	**0.4061 (0.1016)**	0.3056 (0.0552)	0.1997 (0.1203)	0.3131 (0.1033)	0.3036 (0.1113)
	30	**0.4960 (0.0352)**	0.3688 (0.0537)	0.3738 (0.0646)	0.4685 (0.0500)	0.4606 (0.0523)

Table 3 presents the comparison of experimental results of MIGD on functions DF1 to DF9 with a severity of 10 and frequency of 10 and 30. At frequency 10, regarding MIGD, the most proximity to the Pareto front is related to the proposed algorithm in all functions except DF7. Also, for DF7, algorithm DMOEA/D exhibits the most closeness to the Pareto front. At frequency 30, the proposed algorithm shows the most closeness to the Pareto front in all functions except DF7. For DF7, algorithm DMOPSO records the most proximity to the Pareto front.

Table 4 compares the experimental results of MHV on functions DF1 to DF9 with a severity of 10 and frequency of 10 and 30. At frequency 10, based on MHV, the most covered volume is attributed to the proposed algorithm in all functions except DF7 and DF8. Also, algorithms DMOEA/D and DNSGA II-B covered the largest volume in DF7 and DF8, respectively. At frequency 30, the proposed algorithm indicates the largest covered volume in all functions other than DF7, DF3, and DF8. Also, algorithms DNSGA II-B, DMOPSO, and DNSGA II-B cover the largest volumes in DF3, DF7, and DF8, respectively.

Figure 3 shows the evolution of average IGD during 20 individual runs at different iterations for test algorithms in DF problems at frequency 30. A good agreement is observed between the diagrams and results presented in table 3.

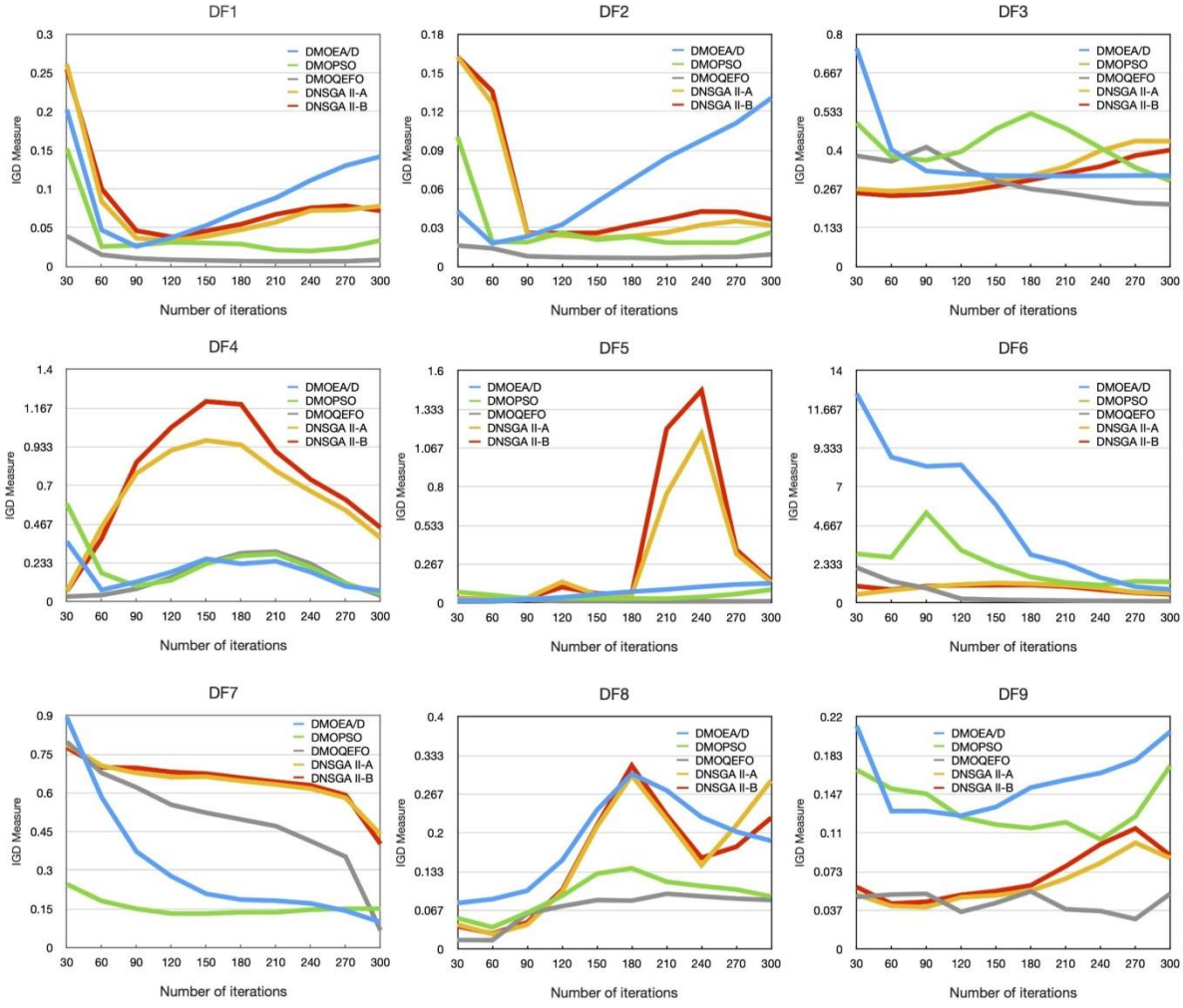

Fig 3. Evolution diagram of average IGD during 20 individual runs at different iterations for test algorithms.

Moreover, figure 4 illustrates the evolution of HV during 20 individual runs at different iterations for test algorithms in DF problems at frequency 30. There is good agreement between the diagrams and results of table 4.

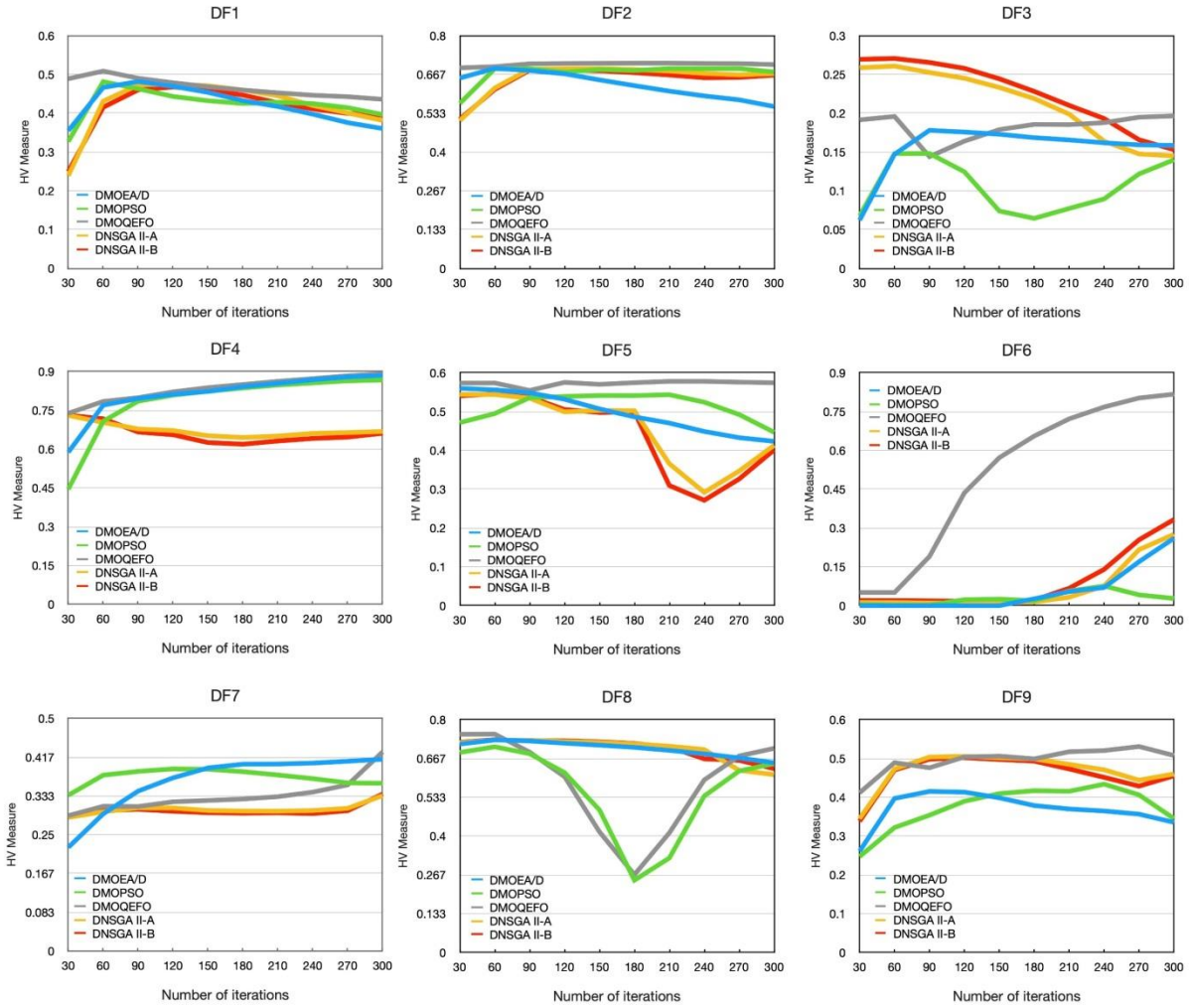

Fig 4. Evolution diagram of average HV during 20 individual runs at different iterations for test algorithms

For statistical analysis, the results obtained from 20 individual runs of the proposed algorithm on each benchmark function with different frequencies are compared to the results of test algorithms in the same conditions. For comparison of results, the Wilcoxon Rank-Sum non-parametric test at a significance level of 5% is used because there is no sufficient knowledge about datasets, and algorithm runs are independent of each other. The results of the Wilcoxon Rank-Sum test on different algorithms are reported in table 5. A given algorithm can be significantly better (+) or significantly worse (-) than the proposed algorithms; the sign = also shows no significant difference between the two algorithms.

Table 5. Summary of Wilcoxon Rank-Sum test results at a significance level of 5%.

Compared to the proposed algorithm		MIGD		MHV	
		τ_t = 10	τ_t =30	τ_t = 10	τ_t =30
DMOEA/D	better (+)	1	1	2	2
	worse (-)	8	8	7	7
	no significant difference (=)	0	0	0	0
DMOPSO	better (+)	0	1	0	1
	worse (-)	8	8	9	8
	no significant difference (=)	1	0	0	0
DNSGA II_A	better (+)	0	0	1	2
	worse (-)	8	8	8	7
	no significant difference (=)	1	1	0	0
DNSGA II_B	better (+)	0	0	1	2
	worse (-)	8	8	7	7
	no significant difference (=)	1	1	1	0

As table 5 demonstrates, the average MIGD of the proposed algorithm from 20 runs on nine benchmark functions with a variation frequency of 10 indicates better performance in about 88.9%, worse performance in about 2.8%, and no significant difference in about 8.3% of comparisons with test algorithms. Also, the MIGD of the proposed algorithm from 20 runs on nine benchmark functions with a variation frequency of 30 shows better performance in about 88.9%, worse performance in about 5.5%, and no significant difference in about 5.5% of comparisons with test algorithms.

According to MHV of the proposed algorithm from 20 runs on nine benchmark functions with a variation frequency of 10, the proposed algorithm shows better performance in about 86.1%, worse performance in about 11.1%, and no significant difference in about 2.7% of comparisons with test algorithms. Also, based on the MHV of the proposed algorithm from 20 runs on nine benchmark functions with a variation frequency of 30, the proposed algorithm has better performance at about 80.5%, worse performance at about 19.4%, and no significant difference at about 0% of comparisons with test algorithms.

In general, the proposed algorithm gains a significant superiority in metrics MIGA and MHV in more than 85% of experiments. The simultaneously great results of these two metrics indicate a fine balance between exploration and exploitation in the proposed algorithm, which results in a superior distribution and approximation of the Pareto front. This balanced is provided through the interaction and competition between the introduced electromagnetic particles in the quantum delta potential well model between the neighbors within the MOEA/D and based on the decomposition technique to determine the position of new particles, which results in a relatively good combination of convergence and diversity. Furthermore, the strategy of new location prediction based on the mean of the two latest changes is markedly compatible with the ambient dynamics.

4-Conclusions

The performance of optimization algorithms is a challenging and important issue; in the presence of factors like the complexity of optimization problems, even a slight improvement in the algorithm performance can be highly valuable. Most optimization problems have inconsistent objectives in the real world. Also, environmental changes and different dynamic features can cause different issues for optimization algorithms.

This study aimed to present a dynamic multi-objective optimization algorithm with improved performance. In this study, EFO was extended to solve dynamic multi-objective optimization problems. The proposed algorithm utilized decomposition and crowding distance strategies. Inspired by the quantum delta-potential well model, the nonlinear motion of quantum-behaved particles, the interactions between electromagnetic particles introduced from positive and negative fields, and the collaboration of neighbours', the proposed algorithm developed a better search approach in the problem space to determine the position of particles. For further extension of the proposed algorithm to deal with dynamic problems, the average displacement of particles' centre of gravity in two last changes was used to predict the level of new change.

In order to visualize the performance of the proposed algorithm and its comparison with mentioned algorithms, including DNSGA-II-A, DNSGA-II-B, dynamic MOEA/D, and dynamic MOPSO, some experiments were conducted on nine various benchmark problems selected from DF functions. The obtained results for IGD and HV metrics after 20 individual runs were gathered, and the standard deviations of obtained values were calculated and listed in related tables. The Wilcoxon Rank-Sum test was performed at a significance level of 5% for statistical analysis of the gathered results. The test's results confirmed the proposed algorithm's superior performance compared to others at a 5% significance level.

The following topics can be investigated in the future:

1) Presenting an effective method to generate a set of weight vectors in the proposed algorithm.
2) Developing an effective method to determine and control different parameters of the proposed algorithm.
3) Using other quantum models to improve the exploration and exploitation capabilities of the proposed algorithm.

References

Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z., & Jawawi, D. N. (2016). Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, *26*, 8-22.

Alatas, B., & Bingol, H. (2020). Comparative assessment of light-based intelligent search and optimization algorithms. *Light & Engineering*, *28*(6).

Akyol, S., & Alatas, B. (2017). Plant intelligence based metaheuristic optimization algorithms. *Artificial Intelligence Review*, *47*(4), 417-462.

Asgari, T., Daneshvar, A., Chobar, A. P., Ebrahimi, M., & Abrahamyan, S. (2022). Identifying key success factors for startups with sentiment analysis using text data mining. International Journal of Engineering Business Management, 14, 18479790221131612.

Asrari, A., Lotfifard, S., & Payam, M. S. (2015). Pareto dominance-based multiobjective optimization method for distribution network reconfiguration. *IEEE Transactions on Smart Grid*, *7*(3), 1401-1410.

Chobar, A. P., Adibi, M. A., & Kazemi, A. (2022). Multi-objective hub-spoke network design of perishable tourism products using combination machine learning and meta-heuristic algorithms. Environment, Development and Sustainability, 1-28. https://doi.org/10.1007/s10668-022-02350-2

Coello, C. C., & Lechuga, M. S. (2002, May). MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)* (Vol. 2, pp. 1051-1056). IEEE.

Deb, K. (2014). Multi-objective optimization. In *Search methodologies* (pp. 403-449). Springer, Boston, MA.

Deb, K., & Deb, D. (2014). Analysing mutation schemes for real-parameter genetic algorithms. Int. J. Artif. Intell. Soft Comput., 4(1), 1-28.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

Deb, K., & Karthik, S. (2007, March). Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In *International conference on evolutionary multi-criterion optimization* (pp. 803-817). Springer, Berlin, Heidelberg.

Du, W., & Li, B. (2008). Multi-strategy ensemble particle swarm optimization for dynamic optimization. Information sciences, 178(15), 3096-3109.

Eshghali, M., Kannan, D., Salmanzadeh-Meydani, N., & Esmaieeli Sikaroudi, A. M. (2023). Machine learning based integrated scheduling and rescheduling for elective and emergency patients in the operating theatre. Annals of Operations Research, 1-24.

Guerreiro, A. P., Fonseca, C. M., & Paquete, L. (2020). The hypervolume indicator: Problems and algorithms. *arXiv preprint arXiv:2005.00515*.

Jiang, S., Yang, S., Yao, X., Tan, K. C., Kaiser, M., & Krasnogor, N. (2018). *Benchmark Functions for the CEC'2018 Competition on Dynamic Multiobjective Optimization*. Newcastle University.

Li, Y., Xiang, R., Jiao, L., & Liu, R. (2012). An improved cooperative quantum-behaved particle swarm optimization. Soft Computing, 16(6), 1061-1069.

Maadanpour Safari, F., Etebari, F., & Pourghader Chobar, A. (2021). Modelling and optimization of a tri-objective Transportation-Location-Routing Problem considering route reliability: using MOGWO, MOPSO, MOWCA and NSGA-II. Journal of optimization in industrial engineering, 14(2), 83-98.

Pourghader Chobar, A., Sabk Ara, M., Moradi Pirbalouti, S., Khadem, M., & Bahrami, S. (2022). A multi-objective location-routing problem model for multi-device relief logistics under uncertainty using meta-heuristic algorithm. Journal of Applied Research on Industrial Engineering, 9(3), 354-373.

Pourghader Chobar, A., Adibi, M. A., & Kazemi, A. (2021). A novel multi-objective model for hub location problem considering dynamic demand and environmental issues. Journal of industrial engineering and management studies, 8(1), 1-31.

Sharifzadegan, M., & Pourghader Chobar, A. (2022). Mathematical modeling and problem solving integrated production planning and preventive maintenance with limited human resources. Journal of New Researches in Mathematics, 8(39), 5-24.

Sierra, M. R., & Coello, C. A. C. (2005, March). Improving PSO-based multi-objective optimization using crowding, mutation and∈-dominance. In *International conference on evolutionary multi-criterion optimization* (pp. 505-519). Springer, Berlin, Heidelberg.

Sun, J., Feng, B., & Xu, W. (2004, June). Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)* (Vol. 1, pp. 325-331). IEEE.

Wang, Y., Feng, X. Y., Huang, Y. X., Pu, D. B., Zhou, W. G., Liang, Y. C., & Zhou, C. G. (2007). A novel quantum swarm evolutionary algorithm and its applications. Neurocomputing, 70(4-6), 633-640.

Yang, X. S. (2020). *Nature-inspired optimization algorithms*. Academic Press, 1-2.

Zandbiglari, K., Ameri, F., & Javadi, M. (2023). A Text Analytics Framework for Supplier Capability Scoring Supported by Normalized Google Distance and Semantic Similarity Measurement Methods. Journal of Computing and Information Science in Engineering, 23(5), 051011.

Zeng, S. Y., Chen, G., Zheng, L., Shi, H., de Garis, H., Ding, L., & Kang, L. (2006, July). A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In *2006 IEEE International Conference on Evolutionary Computation* (pp. 573-580). IEEE.

Zhang, Q., Liu, W., & Li, H. (2009, May). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *2009 IEEE congress on evolutionary computation* (pp. 203-208). IEEE.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, *11*(6), 712-731.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications* (Vol. 63). Ithaca: Shaker.