Multi-guiders and Cross-searching Approaches in Multi-objective Particle Swarm Optimization for Electromagnetic Problems

Minh-Trien Pham, Wei Li, Chang Seop Koh, *Senior Member, IEEE*
School of Electrical & Computer Engineering, Chungbuk National University
San 48, Cheongju, Chungbuk 361-763, Korea
pmtrien@chungbuk.ac.kr

Abstract — The main difference in single objective and multi-objective optimization using particle swarm optimization is how to define the guider to locate the global optimal and non-dominated solutions in corresponding optimization problems. In general multi-objective particle swarm optimization, only one guider is selected, and, in order to reduce the non-dominated solution for more diversity in the external achieve, only crowding distance in objective spaces is considered. This paper presents a new approach of selecting multi-guiders to lead a swarm toward a Pareto-front. Aside from considering the crowding distance of solutions in objective spaces to maintain the diversity of solutions, the distance between them in variable spaces also taken into account. The proposed algorithm is compared with recent approaches of multi-objective optimizer in solving a multi-objective version of TEAM 22 benchmark optimization problem.

I. INTRODUCTION

In real world application, dealing with optimization problems involves optimizing multiple solutions together with considering many criteria and constraints are popular. Therefore, many multi-objective optimization (MO) algorithms have been proposed. In recent years, particle swarm optimization (PSO) algorithm is widely applied in MOs, the PSO was known as the simple method in programming and high speed of convergence. However, the high speed of convergence of the global best version of PSO is often associated with a rapid loss of diversity during the optimization process, leading to undesirable premature convergence. Thus, a new MO based on PSO should deal with the premature convergence and the convergence speed [1].

In order to overcome above problem, a new MOPSO algorithm is proposed, which is based on Multi-objective Gaussian PSO (G-MOPSO) [2] with further improved by using multi-guiders and cross-searching (MGC-MOPSO). To show the behavior of the proposed MGC-MOPSO comparing with G-MOPSO, both of them are tested on the multi-objective version of TEAM 22 using FEM analysis for calculating objective function and constraint function values.

II. FUNDAMENTAL OF MGC-MOPSO

In this algorithm, two guiders are chosen among the non-dominated solutions by using the crowding distance information. The effectiveness of the second guider is controlled by the cross-searching factor to guide the swarm during searching process. Furthermore, to keep diversity of solutions, the distance of non-dominated solutions in variable spaces is also considered together with the crowding distance in objective spaces to remove non-dominated solutions. The implementation of proposed MOPSO is based on the following steps:

Step 1: Initialize a swarm of particles.

- Generate *NP* particles with random velocity and position, and set the iteration counter $t = 0$.
- Evaluate all objective function and constraint values.
- Update non-dominated solutions and store in the external achieve *A*.
- Calculate the crowding distance for all solutions in *A*.
- Sort *A* in the descending order of crowding distance.

Step 2: Update a new personal best position.

- If current position of particle *i*-th dominates the previous its personal best position, update current position as a new personal best position, otherwise, do not update.
- Randomly select a new personal best position between current position and previous personal best position, if they are not dominated by each other.

Step 3: Update the guiders.

- Randomly select the first guider $\mathbf{g}_1$ from the top 10% crowding distance of sorted *A*.
- If $\mathbf{g}_1$ is an extreme solution, the second guider is not necessary to be considered. Otherwise, by considering the two non-dominated solutions beside the first guider in Pareto-front, the bigger crowding distance among them is selected as the second guider $\mathbf{g}_2$ as shown in Fig. 1.

Step 4: Update velocities and positions.

$$\begin{aligned}\mathbf{v}_i(t) &= \omega\cdot\mathbf{v}_i(t-1) + c_1\cdot Ud\cdot\left(\mathbf{p}_i(t-1)-\mathbf{x}_i(t-1)\right)\\ &+c_2\cdot Gd\cdot\left(\mathbf{g}_1(t-1)-\mathbf{x}_i(t-1)\right)\\ &+c_2\cdot\alpha\cdot\chi\cdot Gd\cdot\left(\mathbf{g}_2(t-1)-\mathbf{x}_i(t-1)\right)\end{aligned}\tag{1}$$

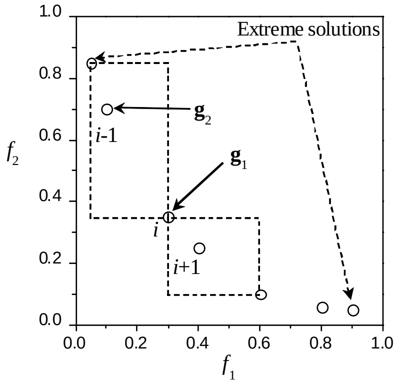

Fig. 1. Selection of two guiders

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{v}_i(t) \quad (2)$$

where, ω is initial weight, constants c_1 and c_2 are the acceleration factors, which are further improved by cognitive and social time-variant factors [2]. Variables *Ud*, *Gd* are random numbers in [0,1]; however, *Ud* is the uniform distribution and *Gd* is Gaussian distribution. $\mathbf{v}_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]^T$ and $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]^T$ stand for velocity and position of i-th particle in D-dimension, respectively. $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iD}]^T$ represents the personal best position of i-th particle. $\mathbf{g}_k = [g_{k1}, g_{k2}, \ldots, g_{kD}]^T$, k=1,2, are two guiders. Extreme factor χ is set to zero if $\mathbf{g}_1$ is extreme solution and 1 otherwise. The effectiveness of the second guider $\mathbf{g}_2$ is controlled by cross-searching factor α. In general, α is time-variant factor, which varies linearly from 0 to 1 depending on the iteration counter increasing from 0 to maximum number of iteration.

<u>Step 5:</u> Evaluate objective function.

- Mutation operator is applied to prevent premature convergence due to existing local Pareto-fronts [3].
- Evaluate all objective function and constraint values.

<u>Step 6:</u> Update non-dominated solutions.

- The external achieve absorbs superior current non-dominated solutions and eliminates inferior solutions.
- While the number of solutions in *A* is bigger than expected solutions, then repeat the following process:
  + Calculate the crowding distance for all members in *A*.
  + Sort *A* in the descending order of crowding distance.
  + Calculate the distance of the lower 10% of crowding distance in variable spaces; for *i*-th solution, the distance is presented as:

$$dx_i = \min_{j\neq i;\, j=1,\cdots,RS}\left(\left|x_i - x_j\right|\right);\ i = 1,\cdots,RS \quad (3)$$

where, *RS* is number of non-dominated solutions at the lower 10% of crowding distance. |.| denotes the Euclidean distance in variable spaces.

+ Calculate the threshold distance.

$$d_{th} = \min_{i=1,\cdots,RS}(dx_i) + \left(\max_{i=1,\cdots,RS}(dx_i) - \min_{i=1,\cdots,RS}(dx_i)\right)\cdot 10\% \quad (4)$$

+ Remove the non-dominated solution which has dx_i smaller than d_{th}. If number of solutions is still bigger than limitation, d_{th} will be increased as the following:

$$d_{th} = d_{th} + \left(\max_{i=1,\cdots,RS}(dx_i) - \min_{i=1,\cdots,RS}(dx_i)\right)\cdot 10\% \quad (5)$$

<u>Step 7:</u> Termination check.

- Increase the iteration counter $t = t+1$.
- If t reaches the maximum number of iteration, t_{max}, the algorithm will stop, otherwise go to Step 2.

III. RESULTS AND CONCLUSION

The multi-objective version of TEAM workshop problem 22 proposed in [4], is continuous, constrained, eight variables with two conflicting objectives: the stray field along two given lines and the achievement of a required stored energy, and two constraint functions are defined as:

$$\begin{aligned}
&\text{Minimize}\quad \mathbf{f}(f_1, f_2) = \left(\frac{B^2_{stray}}{B^2_{normal}}, \frac{\left|Energy - E_{ref}\right|}{E_{ref}}\right)\\
&\text{subject to}\quad \left|J_1\right| \le (-6.4\left|B_1\right| + 54.0) \quad (A/mm^2) \quad (6)\\
&\qquad\qquad\quad\ \left|J_2\right| \le (-6.4\left|B_2\right| + 54.0) \quad (A/mm^2)
\end{aligned}$$

where, the reference stored energy and stray field are E_{ref}=180 MJ and B_{normal} = 200μT, respectively. J_1, B_1 and J_2, B_2 are current density and magnetic flux density in inner and outer coil, respectively. The definition of B^2_{stray} and more information about this problem can be found in [4].

In this problem, two constraint functions may cause discontinuities in the Pareto-front. Therefore, constraint handling mechanism based on the concept of constrained-domination in NSGA-II is also applied in this algorithm [5].

The parameters in G-MOPSO and MGC-MOPSO are set up exactly same as in [2]. By using the FEM analysis to calculate two objective function and two constraint function values, the first trial solutions of G-MOPSO and MGC-MOPSO are shown in Fig. 2. As the results, G-MOPSO has a better distribution at the extreme solutions but MGC-MOPSO has a better distribution where the f_1 and f_2 are both minimized without missing the extreme solutions.

In the full version of this paper, the spacing matrix, general distribution and other parameters of solutions will be compared. Also the effectiveness and behavior of the proposed MGC-MOPSO will be discussed in detail.

IV. REFERENCES

[1] D. Liu et al., “A Multiobjective memetic algorithm based on Partice swarm optimization,” *IEEE Trans. on Syst. Man Cybern*. Part B, vol.37, no.1, pp. 42-50, Feb. 2007.

[2] L. S. Coelho et al., “A Multiobjective Gaussian particle swarm approach applied to electromagnetic optimization,” *IEEE Trans. Magn.*, vol. 46, no. 8, Aug. 2010.

[3] C. A. C. Coello et al., “Handling multiple objectives with particle swarm optimization,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[4] P. Alotto et al., “SMES optimization benchmark extended: Introducing uncertainties and pareto optimal solutions into TEAM22,” *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 106–109, Jun. 2008.

[5] K. Deb et al., “A fast and elitist multiobjective genetic algorithms: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

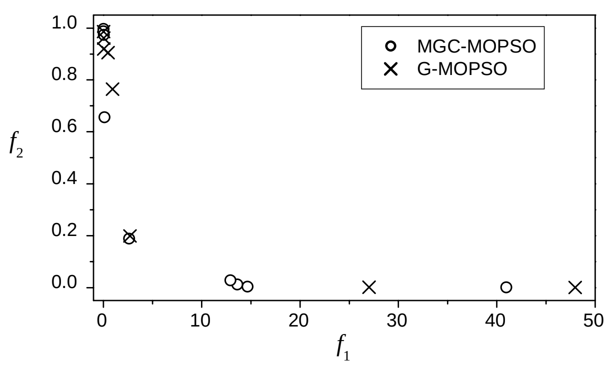

Fig. 2. Pareto-front of G-MOPSO and MGC-MOPSO.