

Nonlinear optimization for a low-emittance storage ring

Bonghoon Oh,^a Jinjoo Ko,^a Seunghwan Shin,^{a,c,*} Jaehyun Kim,^{b,*} Jaeyu Lee^b and Gyeongsu Jang^b

^aDepartment of Accelerator Science, Korea University, 2511 Sejong-ro, Sejong 30019, South Korea, ^bPohang Accelerator Laboratory, POSTECH, Pohang, Kyungbuk 37673, South Korea, and ^cMultipurpose Synchrotron Radiation Construction Project, Korea Basic Science Institute, 162 Yeongudanji-ro, Cheongwon-gu, Cheongju, Chungcheongbukdo 28119, Republic of Korea. *Correspondence e-mail: tlssh@korea.ac.kr, picoma@postech.ac.kr

Received 4 January 2024

Accepted 15 May 2024

Edited by G. Kamel, SESAME, Jordan

Keywords: storage ring; dynamic aperture; MOGA; MGGPO; self-adaptive crossover.

A multi-objective genetic algorithm (MOGA) is a powerful global optimization tool, but its results are considerably affected by the crossover parameter η_c . Finding an appropriate η_c demands too much computing time because MOGA needs to be run several times in order to find a good η_c . In this paper, a self-adaptive crossover parameter is introduced in a strategy to adopt a new η_c for every generation while running MOGA. This new scheme has also been adopted for a multi-generation Gaussian process optimization (MGGPO) when producing trial solutions. Compared with the existing MGGPO and MOGA, the MGGPO and MOGA with the new strategy show better performance in nonlinear optimization for the design of low-emittance storage rings.

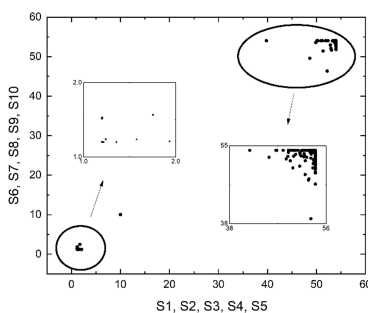
1. Introduction

In the design of an accelerator complex, nonlinear beam dynamics optimization is often repeatedly conducted to ensure the accelerator meets the required performance. A larger dynamic aperture (DA) secures a high injection efficiency as it helps to capture the injected beam of large amplitude. A longer Touschek lifetime implies that the loss of stored beam due to energy transfer between electrons in the beam is minimized. The DA area and Touschek lifetime can be increased by imposing appropriate sextupole strengths. Optimization algorithms can help to find suitable sextupole strengths.

Several stochastic global optimization schemes such as the multi-objective genetic algorithm (MOGA) (Deb *et al.*, 2002; Yang *et al.*, 2009), multi-objective particle swarm optimization (MOPSO) (Pang & Rybarczyk, 2014; Lin *et al.*, 2015) and multi-generation Gaussian process optimization (MGGPO) (Song *et al.*, 2020) have been successfully adopted in optimizing the lattice design.

MOGA exploits survival of the fittest by analogy with biological evolution. First, N candidate solutions are produced randomly. These parent solutions are combined to yield child solutions by crossover, and some variables of child solutions are changed by mutation. The N best, *i.e.* fittest, solutions are retained and others are abandoned so that a population of solutions is maintained. These solutions become parents of the next generation. This process is carried out iteratively until the N solutions converge sufficiently (Pareto front).

In contrast, MGGPO generates more solutions (trial solutions) than the population by utilizing crossover and mutation of MOGA and random movement using the MOPSO mechanism which mimics animals' organized movement such



as flocking behavior to find food. MGGPO uses Gaussian process (GP) regression, a kind of supervised machine learning, to predict objectives of trial solutions because this prediction requires negligible computing time. The N best solutions are retained and others are discarded. Objectives of selected solutions are evaluated to establish a GP model for GP regression in the subsequent generation.

In MOGAs, crossover, *i.e.* simulated binary crossover (SBX), and mutation are quantified using the crossover parameter η_c and the mutation parameter η_m ; these parameters affect the Pareto front and the convergence speed. In existing MOGAs, η_c is fixed at every generation. In this paper, we introduce the self-adaptive crossover parameter (SAXP) scheme, which uses a new η_c at each generation (Deb *et al.*, 2007). A SAXP strategy was also adopted to a MOGA crossover for manufacturing trial solutions in MGGPO. Section 2 introduces the SAXP strategy. Section 3 compares the results of MGGPO and MOGA in optimizing nonlinear dynamics of low emittance storage rings, each using either a fixed crossover parameter (FXP) or SAXP scheme. Section 4 presents conclusions.

2. Self-adaptive crossover parameter strategy

We used a real-coded genetic algorithm (GA) for nonlinear beam dynamics optimization and, therefore, for crossover we used an SBX formula,

$$c_{1,i} = \frac{1}{2}[(1 + \beta_i)p_{1,i} + (1 - \beta_i)p_{2,i}], \quad (1)$$

$$c_{2,i} = \frac{1}{2}[(1 - \beta_i)p_{1,i} + (1 + \beta_i)p_{2,i}], \quad (2)$$

where i is the decision-variable index, $p_{1,i}$ and $p_{2,i}$ are decision variables of the parent solutions, $c_{1,i}$ and $c_{2,i}$ are decision variables of child solutions, and

$$\beta_i = \begin{cases} (2u)^{1/(\eta_c+1)} & \text{if } 0 \leq u \leq 0.5, \\ \{1/[2(1-u)]\}^{1/(\eta_c+1)} & \text{if } 0.5 < u < 1, \end{cases} \quad (3)$$

where $0 \leq u \leq 1$ is a uniformly distributed random number. In MOGA, η_c is set by the user and generally remains constant during the whole MOGA process.

Equations (1) and (2) demonstrate that two child solutions $c_{1,i}$, $c_{2,i}$ lie on a line between two parent solutions $p_{1,i}$, $p_{2,i}$ in decision-variable space, and that the children are located inside a region bounded by parents if $0 \leq \beta_i \leq 1$ (*i.e.* $0 \leq u \leq 0.5$), and outside of that region if $\beta_i > 1$ (*i.e.* $0.5 < u < 1$).

η_c in equation (3) determines the probability distribution of offspring. As η_c increases, the probability distribution of offspring around the parents narrows in the variable space (Fig. 1); *i.e.* when random number u is fixed, the child solution nears the parent as η_c increases. For example, when $p_1 = 5$, $p_2 = 10$ and $u = 0.1$, the first child c_1 created with $\eta_c = 15$ is 5.24 and the second child c'_1 created with $\eta'_c = 5$ is 5.59 (Fig. 2). This characteristic can be exploited using the principle of Nelder and Meade's simplex (Press *et al.*, 1992) to improve the child

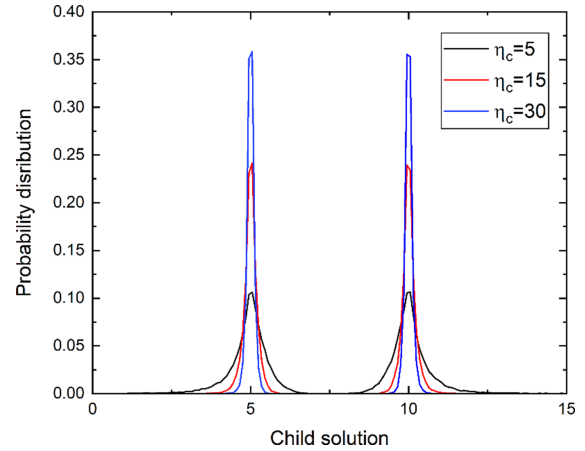


Figure 1 Distribution of two offspring according to the crossover parameter η_c .

solutions. Firstly, offspring are produced by crossover using a fixed η_c that is pre-defined by the user. Objectives of each child and two parents are compared. If a child solution is better than both parent solutions, then a better solution than this child can probably be found at a farther position from the nearest parent than this child. This principle was created by exploiting the *expansion* concept of the simplex. To obtain this ‘farther’ solution, η_c is replaced with a new parameter $\eta'_c < \eta_c$ and another crossover is conducted. If the resulting child is inferior to the parents, then another crossover is conducted using $\eta'_c > \eta_c$ by exploiting *contraction* of the simplex. Otherwise, the crossover uses $\eta'_c = \eta_c$. The same u must be used for the first and second crossovers.

The new η'_c is chosen using a protocol. The relationship between η_c and β_i differs [equation (3)] according to whether $0 \leq u \leq 0.5$ ($0 \leq \beta_i \leq 1$) or $0.5 < u < 1$ ($\beta_i > 1$). The formula for η'_c also differs according to whether the child solution is superior or inferior to the parent solutions. Therefore, four η'_c formulae will be listed.

When child c_1 is superior to both parents p_1 and p_2 , it must be that $|c'_1 - p_1| > |c_1 - p_1|$. If $\beta_i > 1$, the child is located outside

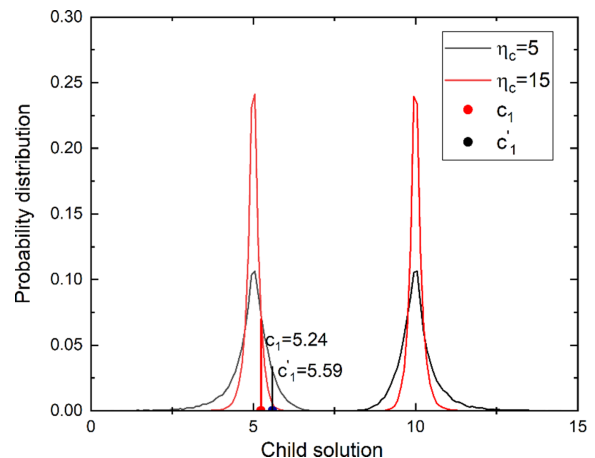


Figure 2 Locations of c_1 and c'_1 with $\eta_c = 5$ and $\eta'_c = 10$ and the same random number $u = 0.1$.

of the two parents by external division from equation (2). From equation (2), β_i can be written as

$$\beta_i = 1 + \frac{2(c_{1,i} - p_{1,i})}{p_{1,i} - p_{2,i}}. \quad (4)$$

When $\beta_i > 1$, equation (3) can be rearranged to

$$\eta_c = - \left\{ 1 + \frac{\log[2(1-u)]}{\log \beta_i} \right\}. \quad (5)$$

We introduce parameter

$$\alpha = \frac{c'_{1,i} - p_{1,i}}{c_{1,i} - p_{1,i}} \quad (6)$$

to quantify how much changing the crossover parameter affects the difference between c' and c .

Here, $|c'_1 - p_1| > |c_1 - p_1|$, so $\alpha > 1$. As α increases, $c'_{1,i}$ moves away from p_1 . From equation (4), β'_i can be calculated by replacing $c_{1,i}$ with $c'_{1,i}$,

$$\beta'_i = 1 + \frac{2(c'_{1,i} - p_{1,i})}{p_{1,i} - p_{2,i}} \frac{c_{1,i} - p_{1,i}}{c'_{1,i} - p_{1,i}} = 1 + \alpha(\beta_i - 1). \quad (7)$$

Using equation (5), the new η'_c can be written as

$$\begin{aligned} \eta'_c &= - \left(1 + \frac{\log[2(1-u)]}{\log \beta'_i} \frac{\log \beta_i}{\log \beta_i} \right) \\ &= - \left[1 - (\eta_c + 1) \frac{\log \beta_i}{\log \beta'_i} \right] \\ &= -1 + \frac{(\eta_c + 1) \log \beta_i}{\log[1 + \alpha(\beta_i - 1)]}. \end{aligned} \quad (8)$$

In equation (8), $\eta'_c < \eta_c$ if $\alpha > 1$ and $\beta_i > 1$. The right-hand side of equation (8) includes β_i , which has decision variable index i ; this condition means that η'_c is determined differently according to all variables of all solutions. Strictly speaking, η'_c should be denoted as $\eta'_{c,i}$. If child c_1 is inferior to both parents p_1 and p_2 , then α can be replaced with $1/\alpha$ in equation (8) to make c'_1 be created closer than c_1 to p_1 . Therefore, η'_c can be presented as

$$\eta'_c = -1 + \frac{(\eta_c + 1) \log \beta_i}{\log\{1 + [(\beta_i - 1)/\alpha]\}}. \quad (9)$$

If $0 \leq \beta_i < 1$, offspring are between the two parents by internal division from equation (2). When $\beta_i < 1$, from equation (3),

$$\eta_c = -1 + \frac{\log(2u)}{\log \beta_i}. \quad (10)$$

From equation (3), β'_i must not be negative. If we define β'_i as equation (7), then β'_i may be negative, when $\beta_i < 1$, because α can be any value > 1 . We need a new α formula to assure that $0 \leq \beta'_i \leq 1$ when $0 \leq \beta_i \leq 1$. When c_1 is superior to p_1 and p_2 , increased α must move c'_1 farther away from p_1 than c_1 . Here, when $0 \leq \beta_i \leq 1$ and c_1 is superior to p_1 and p_2 , then we can introduce α with the relationship

$$\beta'_i = (\beta_i)^\alpha. \quad (11)$$

With equation (11), if $0 \leq \beta_i \leq 1$, then $0 \leq \beta'_i \leq 1$. If $\alpha \geq 1$, then $\beta'_i \leq \beta_i$ and, as α increases, β'_i decreases. From equation (4), when $0 \leq \beta'_i \leq 1$, a decreased β'_i increases $|2(c'_{1,i} - p_{1,i})/p_{1,i} - p_{2,i}|$ because $\beta'_i - 1 = 2(c'_{1,i} - p_{1,i})/p_{1,i} - p_{2,i} \leq 0$; *i.e.* when $\alpha \geq 1$, $c'_{1,i}$ is produced at a farther position from $p_{1,i}$ than $c_{1,i}$, and when α increases, $c'_{1,i}$ moves away from $p_{1,i}$. Using equations (10) and (11), η'_c is calculated as

$$\begin{aligned} \eta'_c &= -1 + \frac{\log(2u)}{\log \beta'_i} \frac{\log \beta_i}{\log \beta_i} \\ &= -1 + (\eta_c + 1) \frac{\log \beta_i}{\log \beta'_i} \\ &= -1 + \frac{(\eta_c + 1)}{\alpha}. \end{aligned} \quad (12)$$

In equation (12), $\alpha > 1$, so $\eta'_c < \eta_c$. If c_1 is inferior to p_1 and p_2 , then we replace α with $1/\alpha$, so η'_c becomes

$$\eta'_c = -1 + \alpha(\eta_c + 1). \quad (13)$$

Otherwise, if a child solution is not inferior to and superior to both two parents, then we set $\eta'_c = \eta_c$. If η'_c calculated using equations (8), (9), (12) and (13) is negative then we set $\eta'_c = 0$. If $\alpha = 1$, the scheme of SAXP is the same as that of FXP because η'_c will always be the same as η_c . Different η'_c determined by equations (8), (9), (12) and (13) are adopted to each variable of each solution. This means that all variables of new child solutions are determined by the same crossover formulas equations (1)–(3) and different crossover parameter η'_c .

3. Comparison of results of fixed and self-adaptive strategy

The SAXP scheme was first applied to MGGPO. MGGPO produces as many as $10N$ trial solutions by using random movement and $10N$ solutions by using crossover and mutation, where N is the population. Crossover can use the SAXP or the FXP scheme. To fulfill the SAXP strategy, objectives of child solutions must be calculated twice. This means that MGGPO using SAXP consumes double the computing time compared with MGGPO using FXP. However, first objectives investigation of the child solution to determine η'_c was not conducted using particle tracking but GP regression, meaning that not the calculation values but the expectation values of objectives were used to determine η'_c . GP regression demands insignificant computing time, so MGGPO takes similar times when using SAXP and FXP.

MGGPO exploiting both SAXP and FXP was used to simultaneously increase the on and off momentum dynamic apertures (DAs) of the Korea-4GSR lattice. Korea-4GSR is a 28-cell low-emittance electron storage ring with hybrid seven-bend achromat optics (Oh *et al.*, 2021). The ring has six sextupoles in each cell, and the ring is assumed to have a periodicity of 14 in terms of sextupole configuration (*i.e.* 12 power supplies for sextupoles in the ring). In the simulation, we used two of the 12 sextupoles to correct chromaticity and the remaining 10 sextupoles as decision variables of MGGPO.

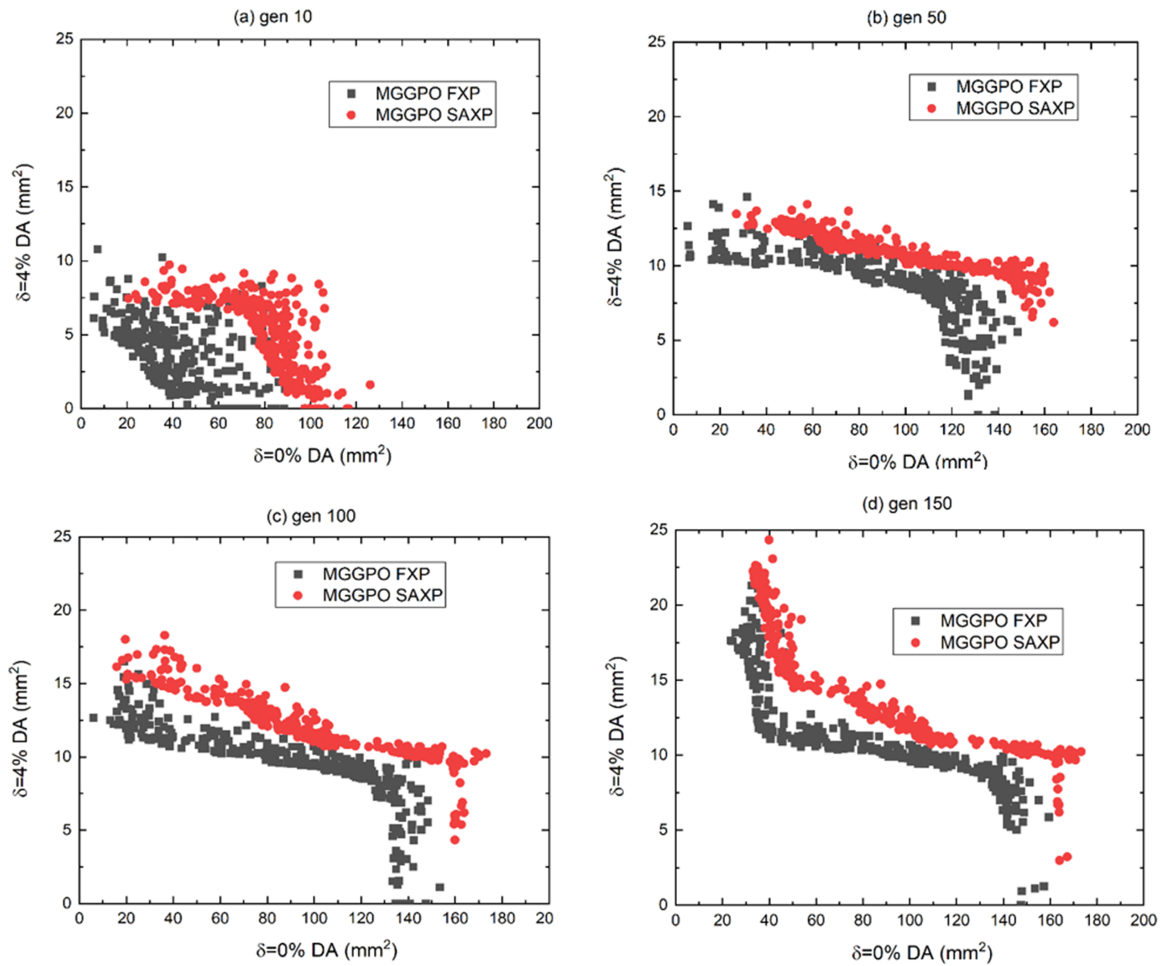


Figure 3 Comparison of objectives by MGGPO using FXP and SAXP with $\alpha = 3$ at generation (a) 10, (b) 50, (c) 100 and (d) 150.

On ($\delta = 0\%$) momentum DA area and off ($\delta = 4\%$) momentum DA area were set as the two objectives. Population N was set at 300 and DA area was calculated using 100-turn tracking to reduce computing time. $\eta_m = 60$ and $\eta_c = 60$. For SAXP, $\alpha = 3$ and η'_c was calculated using equations (8), (9), (12) and (13). MGGPO with 100 generations was performed first using the MOGA crossover in the FXP way [Fig. 3(a)] then in the SAXP way [Fig. 3(b)], and the solutions were compared at the 100th generation in each case [Fig. 3(c)]. MGGPO with SAXP reveals better solutions than MGGPO with FXP [Fig. 3(c)].

MOGA using the SAXP and the FXP method was also performed, with the same decision variables and objectives that were used for MGGPO. Other values were $N = 300$, $\eta_m = 25$ and $\eta_c = 10$. The algorithm was run for 100 generations. When MOGA using SAXP is conducted, the objectives of both child solutions c and c' were estimated using particle tracking, unlike in MGGPO, in which they were estimated by using expectation values of GP regression. For this reason, MOGA that used SAXP required twice as much computing time as MOGA that used FXP.

First, to find an appropriate α value, MOGA with SAXP was conducted using a range of α [Figs. 4(a)–(f)]. Results indicated that $\alpha = 5$ is a good choice for the SAXP scheme, and

provides better solutions than MOGA using FXP. When $\alpha = 5$, the results of MOGA using SAXP at N generations and FXP at $2N$ generations were compared with, N from 20 to 100, considering doubled computing time in the SAXP strategy compared with the FXP strategy (Figs. 5 and 6). Compared with the FXP scheme, the SAXP scheme gave inferior solutions at $N = 20, 40$ and similar solutions at $N = 60$, but superior solutions at $N = 80, 100$. The distribution of η'_c for all variables of all solutions obtained by the SAXP scheme with $\alpha = 5$ at generation 100 is shown in Fig. 7.

4. Conclusion

Previous studies that have applied MOGA have usually used a constant crossover parameter for SBX, *i.e.* the FXP strategy. We tried another way for SBX by changing the crossover parameter at each generation, *i.e.* the SAXP strategy. We compared results of the MOGA optimization with the two schemes to simultaneously optimize the on- and off-momentum DA areas for the Korea-4GSR lattice. We verified that MOGA using SAXP certainly presents a better Pareto front than MOGA using FXP when an appropriate α value is adopted

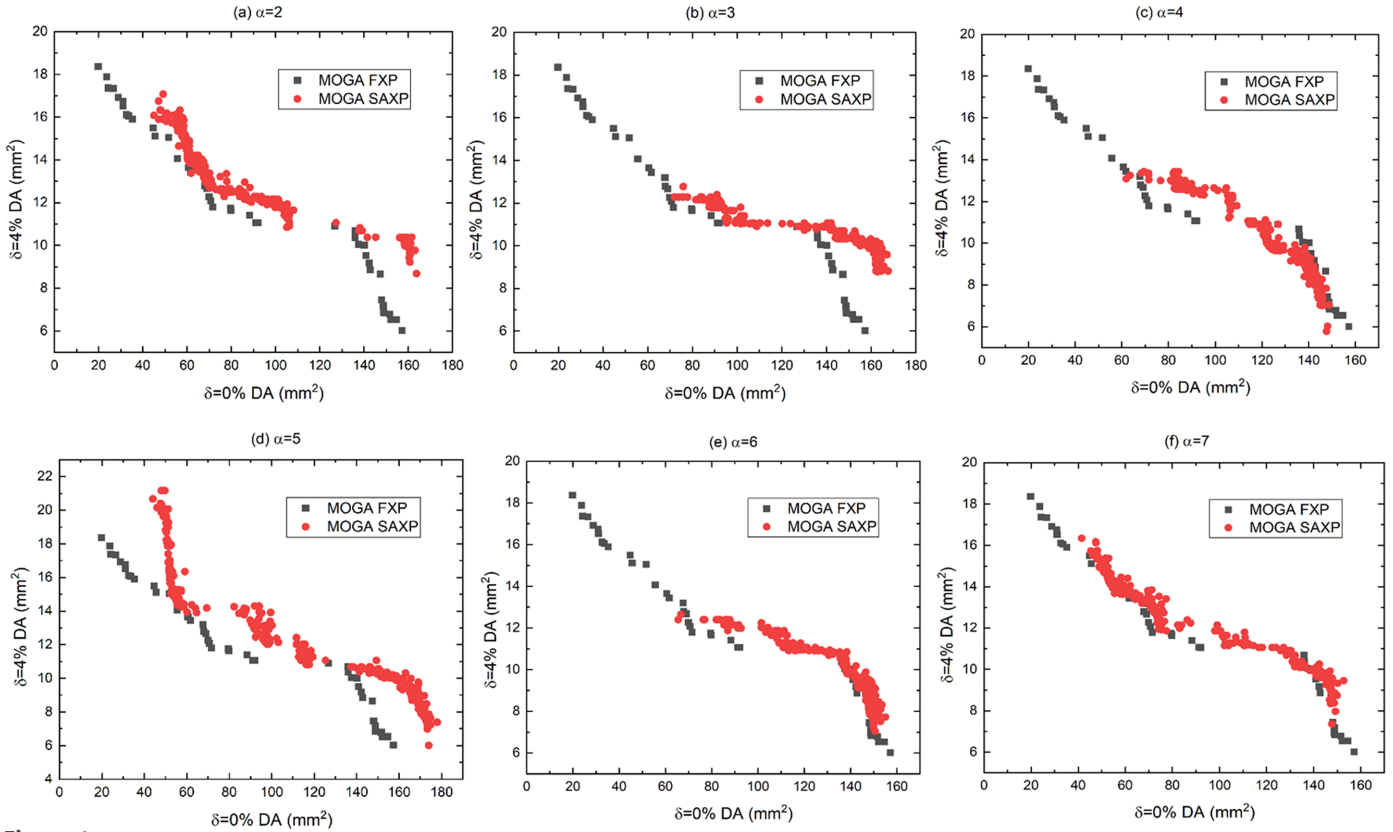


Figure 4 Comparison of objectives by MOGA using FXP and SAXP with (a) $\alpha = 2$, (b) $\alpha = 3$, (c) $\alpha = 4$, (d) $\alpha = 5$, (e) $\alpha = 6$ and (f) $\alpha = 7$.

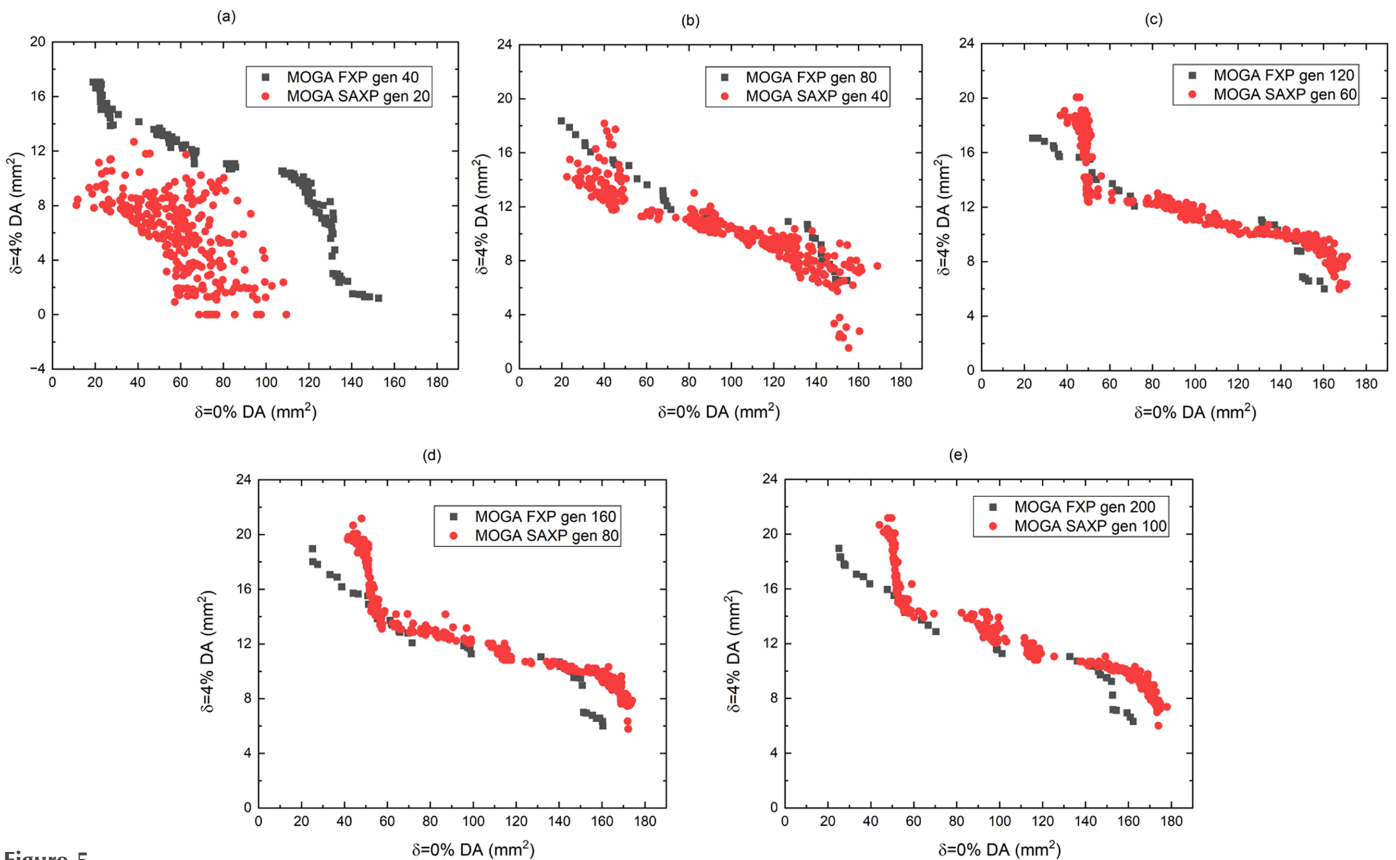


Figure 5 Results of MOGA with FXP strategy at $2N$ generations and SAXP strategy at N generations where N is (a) 20, (b) 40, (c) 60, (d) 80 and (e) 100.

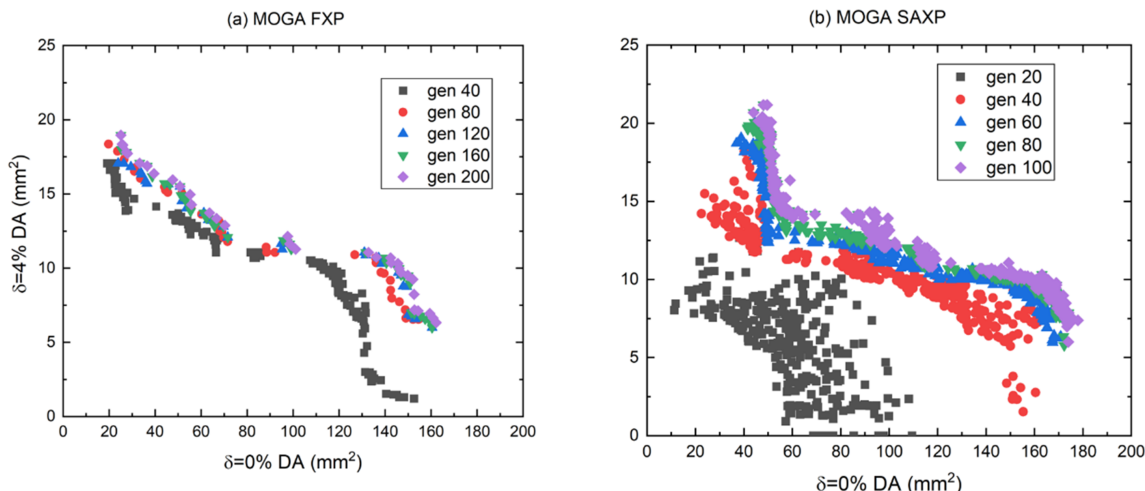


Figure 6 Objectives of all solutions obtained from MOGA using (a) FXP at $2N$ generations and (b) SAXP with $\alpha = 5$ at N generations where N is 20 to 100.

Acknowledgements

We thank M. Yoon (POSTECH) for providing helpful information and many useful discussions. This work was supported by a Korea University Grant and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (RS-2022-00155836).

References

Deb, K. A., Pratap, S., Agarwal, S. & Meyarivan, T. (2002). *IEEE Trans. Evol. Comput.* **6**, 182–197.
 Deb, M., Karthik, S. & Okabe, T. (2007). *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07)*, 7–11 July 2007, London, UK, pp. 1187–1194.
 Lin, Q. J., Li, Z., Du, J., Chen, J. & Ming, Z. (2015). *Eur. J. Oper. Res.* **247**, 732–744.
 Oh, B.-H. J., Ko, J., Lee, G., Jang, G. & Shin, S. (2021). *Appl. Sci.* **11**, 11896.
 Pang, X. & Rybarcyk, L. J. (2014). *Nucl. Instrum. Methods Phys. Res. A*, **741**, 124–129.
 Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992). *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., ch. 10.4. Cambridge University Press.

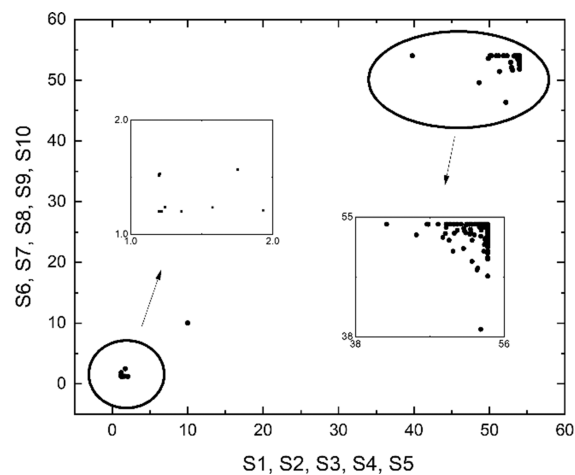


Figure 7 η_c' of all variables of all solutions obtained by the SAXP scheme with $\alpha = 5$ at generation 100. S1–S10 are decision variables that represent 10 sextupole strengths.

Song, M. X., Huang, L., Spentzouris, L. & Zhang, Z. (2020). *Nucl. Instrum. Methods Phys. Res. A*, **976**, 164273.
 Yang, D., Robin, F., Sannibale, C., Steier, C. & Wan, W. (2009). *Nucl. Instrum. Methods Phys. Res. A*, **609**, 50–57.