Multiobjective optimization of the dynamic aperture using surrogate models based on artificial neural networks

M. Kranjčević,* B. Riemann, A. Adelmann, and A. Streun
Paul Scherrer Institut, CH-5232 Villigen PSI, Switzerland

(Received 7 August 2020; accepted 14 December 2020; published 19 January 2021)

Modern synchrotron light source storage rings, such as the Swiss Light Source upgrade (SLS 2.0), use multibend achromats in their arc segments to achieve unprecedented brilliance. This performance comes at the cost of increased focusing requirements, which in turn require stronger sextupole and higher-order multipole fields for compensation of their effects on particles with energy deviation and lead to a considerable decrease in the dynamic aperture and/or energy acceptance. In this paper, to increase these two quantities, a multiobjective genetic algorithm (MOGA) is combined with a modified version of the well-known tracking code TRACY. As a first approach, a massively parallel implementation of a MOGA is used. Compared to a manually obtained solution this approach yields very good results. However, it requires a long computation time. As a second approach, a surrogate model based on artificial neural networks is used in the optimization. This improves the computation time, but the quality of the results deteriorates beyond that of the manually obtained solution. As a third approach, the surrogate model is retrained during the optimization. This ensures a solution quality comparable to the one obtained with the first approach while also providing an order of magnitude speedup. Finally, good candidate solutions for SLS 2.0 are shown and further analyzed.

DOI: 10.1103/PhysRevAccelBeams.24.014601

I. INTRODUCTION

The upgrade of the Swiss Light Source, named SLS 2.0, is scheduled for 2023–2024. To increase the brilliance, the current third generation storage ring will be replaced by one employing seven-bend achromats, including reverse bends and longitudinal gradient bends [1]. The stronger focusing requirements need higher sextupole and higher-order multipole fields for chromatic compensation. This makes finding a reasonably large dynamic aperture (DA) for injection and an energy acceptance for a sufficient beam lifetime more challenging and more important. It can either be done indirectly, by computing and minimizing the dominant resonance driving terms [2], or directly, by computing and maximizing the DA and energy acceptance [3,4].

In this work the latter approach is used and a constrained multiobjective optimization problem is formulated (Sec. II). The search space comprising the strengths of sextupole families, as well as horizontal and vertical linear chromaticity is considered. Similarly to the approach in [3], the objective functions are defined to maximize the transverse DAs at three different energies and to prevent the tune resonances from being crossed, thus maximizing the energy acceptance and beam lifetime. These figures of merit are computed using direct particle tracking with a modified version of the well-known tracking code TRACY [5].

Out of the many multiobjective optimization algorithms, particle swarm optimization [6], differential evolution [7,8] and multiobjective genetic algorithms [3,9–14] have already been successfully applied to the problem of optimizing the DA. In this work a multiobjective genetic algorithm (MOGA) is chosen and further extended with constraint-handling methods (Sec. III).

Previous work includes approaches that speed up the convergence of the multigeneration optimization method by, e.g., preselecting points that are likely to be good using k-means clustering [4] or a *surrogate model* [15–17], i.e., an approximation model which captures the significant properties of a given simulation model and is also very cheap to evaluate. In this work an artificial neural network (ANN) surrogate model is used for the optimization, in combination with a MOGA. The approach is similar to the one from [18]. In particular, the solution quality is improved by retraining the surrogate model during the optimization.

First, the ANN surrogate model is built to approximate the objective functions (Sec. IV). In particular, good hyperparameters are determined and the surrogate model quality is shown. The surrogate model is then used for

[*] marija.kranjcevic@psi.ch

optimization and augmented by a retraining procedure (Sec. V). The run-time and the solution quality of this approach are compared with those of a massively parallel implementation of a MOGA coupled with TRACY. The solution quality is determined in comparison with an existing manually obtained solution.

Finally, good candidate solutions are shown and further analyzed (Sec. VI). In particular, the transverse DAs at the three considered energies are shown and compared with those of the manually obtained solution.

II. OPTIMIZATION PROBLEM

A. Dynamic aperture (DA)

The DA can be loosely defined as an area in the transverse phase space in which stable particle motion can occur. To quantify the size of the DA the approach from [3] together with the modifications from [9] is adopted. As the DA area is dependent on the local linear optics given by the Twiss parameters α and β at the starting location of the particle tracking, the DA coordinates r and θ in Floquet space are mapped to the coordinates for tracking via

$$\begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} \beta_x \\ -\alpha_x \end{pmatrix} \frac{r\cos\theta_k}{\sqrt{\beta_x/I}},$$
$$\begin{pmatrix} y \\ y' \end{pmatrix} = \begin{pmatrix} \beta_y \\ -\alpha_y \end{pmatrix} \frac{r\sin\theta_k}{\sqrt{\beta_y/I}} \tag{1}$$

with the invariant $I = 1\ \mu$m. For SLS 2.0, $r = 1$ corresponds to $\sqrt{I/\epsilon_x} \approx 80$ horizontal rms beamwidths, with ϵ_x being the horizontal emittance (see Table I). The particle trajectories along $2K$ rays in the (x, y) Floquet space starting at the origin are considered. The angles between these rays and the x axes are

$$\theta_k = k\pi/K \quad \text{for } k \in \{0, \ldots, 2K-1\}. \tag{2}$$

Since particles get lost on the vacuum chamber walls, it is reasonable to assume that a realistic DA will not exceed the aperture at the reference energy that would exist if all sextupoles and higher-order magnets were turned off. This assumed upper limit is referred to as the *linear aperture*, and the length it spans on the kth ray is denoted by $\bar{L}(\theta_k)$. Similarly, the length that the DA at a relative energy offset δ spans on this ray is denoted by

$$L(\theta_k, \delta). \tag{3}$$

In order not to reward cases with $L(\theta_k, \delta) > \bar{L}(\theta_k)$, the *line objective* is defined as

$$f_{k,\delta} = \frac{\max\{0, \bar{L}(\theta_k) - L(\theta_k, \delta)\}}{\bar{L}(\theta_k)}. \tag{4}$$

Both $\bar{L}(\theta_k)$ and $L(\theta_k, \delta)$ are computed using the biased binary search as presented in [9]. For the $\bar{L}_k$ search the dimensionless initial length in Floquet space is set to a sufficiently large reference radius of 10, corresponding to $x \approx 41$ mm and $y \approx 26$ mm at the insertion position for the SLS 2.0 lattice, which is well outside of the vacuum chamber. The initial radius for $L(\theta_k, \delta)$ is then set to $\bar{L}_k$. The *DA objective* for a given relative energy offset δ is defined as

$$\mathrm{DA}_\delta = \frac{1}{2K} \sum_{k=0}^{2K-1} f_{k,\delta}^2. \tag{5}$$

In a flat lattice there is vertical symmetry of the aperture area, so (5) becomes

$$\mathrm{DA}_\delta = \frac{1}{2K} \left(f_{0,\delta}^2 + f_{K,\delta}^2 + 2 \sum_{k=1}^{K-1} f_{k,\delta}^2 \right). \tag{6}$$

Due to the normalizations in Eqs. (4) and (5), the DA objective is always in [0,1].

In this paper the on-momentum and off-momentum DA objectives are considered. In particular,

$$\mathrm{DA}_{-\delta}, \mathrm{DA}_0 \quad \text{and} \quad \mathrm{DA}_\delta, \tag{7}$$

where $\delta = 0.03$. Note that the maximization of the DA corresponds to the minimization of the DA objectives $\mathrm{DA}_{-\delta}$, DA_0 and DA_δ.

TABLE I. Basic parameters and design constraints of the considered SLS 2.0 lattice.

Parameter	Value
Beam energy	2.7 GeV
Circumference	288 m
Horizontal emittance ϵ_x (rms, no insertions)	157 pm
Energy spread (rms)	1.1×10^{-3}
Horizontal tune Q_x	39.35
Vertical tune Q_y	15.25
Horizontal chromaticity ξ_x	$0 < \xi_x < 1$
Vertical chromaticity ξ_y	$0 < \xi_y < 1$

B. Crossing tune resonances

The DA objectives in Eq. (7) are computed to ensure a sufficiently large aperture region in phase space. Unfortunately, the binary search used to compute the line objectives in Eq. (4) is sufficient only when the aperture region is *simply connected*, i.e., every closed curve within the region can be shrunk continuously to a point that is in the region. This is not always the case, especially when particles cross tune resonances. Therefore, additional requirements that take into account the crossing of tune resonances need to be defined. For this, the tune

$$\vec{Q}(x,y,\delta) = (Q_x(x,y,\delta), Q_y(x,y,\delta)) \tag{8}$$

and the fractional tune

$$\vec{\nu}(x,y,\delta) = (\nu_x(x,y,\delta), \nu_y(x,y,\delta)) \tag{9}$$

are considered as functions of the initial positions x, y in transverse Floquet space and the relative energy deviation δ. In the case of the considered SLS 2.0 lattice their relationship is given by

$$\nu_x = Q_x - 39, \qquad \nu_y = Q_y - 15. \tag{10}$$

For a sufficient energy and amplitude acceptance it is beneficial to constrain the variation of tunes so that no low-order resonances are crossed. In particular, in this paper the variation of tunes, known as the tune footprint, is constrained inside the triangle formed by three first- and second-order resonances around the on-momentum tune. In the case of the considered SLS 2.0 lattice the on-momentum tune is

$$\vec{Q}(0,0,0) = (39.35, 15.25) \tag{11}$$

(see Table I) and the vertices of this triangle (see Fig. 7 on p. 13, red lines) are

$$\{(39,15), (39.5,15.5), (39.5,15)\}. \tag{12}$$

The triangle is also shown in the fractional tunes coordinates ν_x and ν_y in Fig. 1 (dashed line). To prevent particles from getting lost on resonance stop bands, a margin of 0.025 around the resonance lines is used in this work (see the blue triangle in Fig. 1 or Fig. 7).

1. Chromatic tune footprint (CTFP)

The chromatic tune footprint is approximately computed by sampling the energy-dependent fractional tunes ν_x and ν_y at the energy offsets

$$\delta_p = \frac{p \cdot \delta_{\max}}{P}, \qquad p = -P, \ldots, P. \tag{13}$$

There are cases for which the particle motion is unstable and the tunes cannot be computed. Denoting by $g(\vec{\nu})$ the squared Euclidean distance of $\vec{\nu}$ from the aforementioned triangle with margins (see Fig. 1, blue triangle) the *tune footprint distance* is defined as

$$\texttt{ctfp} = \sum_{\substack{p\in\{-P,\ldots,P\}\\ \text{computable}}} g(\vec{\nu}(0,0,\delta_p)). \tag{14}$$

Taking into account that [see Eqs. (11) and (10)]

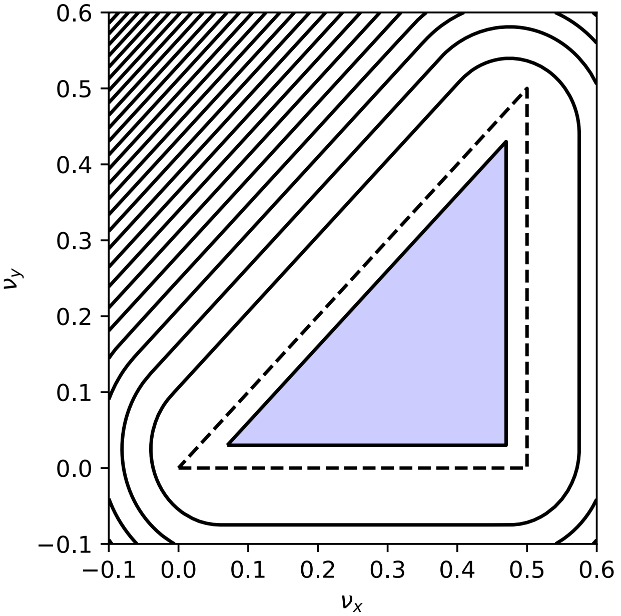


FIG. 1. The isolines of the function g from Sec. II B 1. The dashed lines show the resonance triangle. The blue area denotes the triangle with margins, which is derived from the resonance triangle using a margin of 0.025. Inside this blue area g vanishes. Outside of it $g > 0$ grows quadratically with the distance to that triangle.

$$\vec{\nu}(0,0,\delta_0) = \vec{\nu}(0,0,0) = (0.35, 0.25) \tag{15}$$

is inside the triangle with margins, it holds that

$$g(\vec{\nu}(0,0,0)) = 0, \tag{16}$$

so Eq. (14) becomes

$$\texttt{ctfp} = \sum_{\substack{p\in\{-P,\ldots,P\}\setminus\{0\}\\ \text{computable}}} g(\vec{\nu}(0,0,\delta_p)). \tag{17}$$

If the chromatic tune footprint extends outside of the triangle with margins, $\texttt{ctfp} > 0$. Furthermore, as in [3],

$$\texttt{unstable}_{\pm} = 1 - |\delta_{u,\pm}|/\delta_{\max} \tag{18}$$

is defined, with $\delta_{u,+}$ and $\delta_{u,-}$ denoting the first (i.e., smallest in magnitude) positive and negative values, respectively, for which the tunes are located outside the triangle or not computable. It follows from Eqs. (13) and (18) that the set of values which $\texttt{unstable}_{\pm}$ can attain is

$$\{1 - p/P | p \in \{0,1,\ldots,P\}\}. \tag{19}$$

The value of P needs to be high enough to achieve a sufficient resolution in tune space, without superfluous computational overhead—in this paper $P = 25$ is used.

The required energy acceptance for SLS 2.0 is defined by the requirements for sufficient Touschek lifetime. The momentum aperture has the approximate range

$$\delta \in [-0.062, 0.047] \tag{20}$$

and the positive values dominate the lifetime contributions. Therefore, for this study $\delta_{\max} = 0.05$ is used.

2. Amplitude-dependent tune shifts (ADTS)

The betatron oscillation is nonlinear and thus anharmonic. Therefore, a number of different amplitudes have to be considered. In this work, to achieve a sufficient resolution, $Q = 20$ equidistant points are taken on each of the following two line segments in the transverse Floquet plane: the horizontal line segment [see the sentence containing Eq. (3) for the definition of $L(\cdot, 0)$]

$$\{(t, \Delta) | t \in]0, L(0, 0)]\} \tag{21}$$

and the vertical line segment

$$\{(\Delta, t) | t \in]0, L(\pi/2, 0)]\}. \tag{22}$$

In these points (with $\delta = 0$) the tunes are computed as the fundamental frequencies of turn-by-turn data in each plane. To compute these frequencies, the FFT of 128 tracked turns with zero padding to 512 samples is used. To excite both oscillation modes $\Delta = 10^{-4}$ is used, offsetting these line segments from the original rays of the DA computation.

As in the case of computing the tune footprint distance `ctfp` in Eq. (17), the squared Euclidean distance of the footprint points from the triangle with margins (blue triangle in Fig. 1) is subsumed into the *amplitude-dependent tune footprint distance*

$$\texttt{adts} = \sum_{\substack{q=1\\ \text{computable}}}^{Q} g(\vec{\nu}(x_q, \Delta, 0)) + \sum_{\substack{q=1\\ \text{computable}}}^{Q} g(\vec{\nu}(\Delta, y_q, 0)). \tag{23}$$

Here “computable” refers to the tracked particle not being lost in 512 turns.

C. Search space

Sextupoles are mainly used to compensate chromaticity, but they also limit the on-momentum transverse DA. To have the possibility to extend the DA limits, more than two sextupole families are used. The strengths of these two sextupole families are subsumed into a vector of tuning sextupole strengths $\vec{t} = (t_1, t_2)$, whose linear relationship with chromaticity is quantified by the matrix $\mathbf{T}$. The sextupole strengths of the remaining families are grouped into a vector $\vec{\kappa}$, and their influence on (linear) chromaticity $\vec{\xi} = (\xi_x, \xi_y)$ is characterized by a matrix $\mathbf{M}$, so that

$$\vec{\xi} = \mathbf{M}\vec{\kappa} + \mathbf{T}\vec{t} + \vec{\xi}_{\mathrm{ua}}, \tag{24}$$

where $\vec{\xi}_{\mathrm{ua}}$ is the chromaticity of the unaltered lattice. By magnet design, the applicable sextupole strength is limited to some interval $[-\kappa_{\max}, \kappa_{\max}]$.

In addition to the sextupole strengths $\vec{\kappa}$, the chromaticity is also taken to be a part of the search space. To prevent head-tail instability, it must be non-negative. On the other hand, the upper limit $\xi_{\max}$ can be adjusted.

To sum up, a design point in the search space is

$$\vec{d} = (\xi_x, \xi_y, \kappa_1, \ldots, \kappa_5), \tag{25}$$

where

$$\xi_{x,y} \in [0, \xi_{\max}], \qquad \kappa_i \in [-\kappa_{\max}, \kappa_{\max}]. \tag{26}$$

The SLS 2.0 sextupoles have a bore of 22 mm and a maximum poletip field of 0.71 T at 2.7 GeV, which corresponds to $\kappa_{\max} = 650\ \mathrm{m}^{-3}$. Furthermore, in this work $\xi_{\max}$ is set to 1.

D. Multiobjective optimization problem

Summing up the three preceding sections, the constrained multiobjective optimization problem considered in this paper is [see Eqs. (6), (7), (18) and (25)]

$$\min_{\vec{d}}(\overbrace{\mathrm{DA}_{-\delta}}^{F_1}, \overbrace{\mathrm{DA}_0}^{F_2}, \overbrace{\mathrm{DA}_\delta}^{F_3}, \overbrace{\texttt{unstable}_{\mp}}^{F_4, F_5}), \tag{27}$$

subject to [see Eqs. (17) and (23)]

$$t_1, t_2 \in [-\kappa_{\max}, \kappa_{\max}] \quad \text{and} \quad \texttt{ctfp} + \texttt{adts} = 0. \tag{28}$$

The second constraint ensures that the tune footprint distance `ctfp` and the amplitude-dependent tune footprint distance `adts` from the triangle with margins (see blue triangle in Fig. 1) are both zero.

III. MULTIOBJECTIVE GENETIC ALGORITHM (MOGA)

In Eq. (27) multiple objectives have to be optimized simultaneously. There are many multiobjective algorithms for this purpose, such as particle swarm optimization [6,19–21], ant colony optimization [22], simulated annealing [23], artificial immune system [24], differential evolution [7,8] or genetic algorithm [25]. Multiobjective genetic algorithms (MOGA) are probably the most popular and they have already been successfully applied in the field of particle accelerator physics [12,18,26–29], in particular also for the DA optimization [3,4,10,11,13,14].

A design point $\vec{d}_1$ [see Eq. (25)] *dominates* $\vec{d}_2$ if it is not worse in any of the objectives [see Eq. (27)], and it is strictly better in at least one objective. A MOGA allows independent evaluations of solution candidates and is therefore suitable for parallelization. In this work a massively parallel implementation of a MOGA [28–30] is used to find points that are not dominated by any other point, called *Pareto optimal points*. The basic steps of a MOGA are shown in Algorithm 1.

Algorithm 1. Multiobjective genetic algorithm.

```
1: random population of individuals, $\vec{d}_i$ for $i = 1, \ldots, M$
2: compute $\vec{F}(\vec{d}_i)$ for $i = 1, \ldots, M$
3: while a stopping criterion not reached do
4:     for pairs of individuals $\vec{d}_i$, $\vec{d}_{i+1}$ do
5:         crossover($\vec{d}_i$, $\vec{d}_{i+1}$), mutate($\vec{d}_i$), mutate($\vec{d}_{i+1}$)
6:     for each new individual $\vec{d}_{\text{new}}$, compute $\vec{F}(\vec{d}_{\text{new}})$
7:     choose $M$ fittest individuals for the next generation
```

In the context of a MOGA a design point is referred to as an *individual*. First, in line 1, M individuals are chosen uniformly at random from intervals in Eq. (26). In line 2 their objective function values [Eq. (27)] are computed. Then, in lines 3–7, a number of cycles are performed, each resulting in a new generation. In every cycle new individuals are created from the existing ones using two recombination operators: crossover and mutation (lines 4–5). The crossover operator used in this paper is the simulated binary crossover. It creates two new individuals (children) from two existing individuals (parents). The mutation operator creates a new individual from a single existing individual to ensure diversity in the population. In this paper independent bit mutation is used, which means that each of the design variable values of the existing individual is changed with a given probability. In particular, it is ensured that the new individuals belong to the search space, i.e., that their design variable values are in the bounds from Eq. (26). The objective function values of the newly created individuals are then computed in line 6. Finally, in line 7, approximately M fittest individuals are chosen to comprise the new generation. The implementation from [26,30] (called OPT-PILOT) is used, where the algorithm is implemented in C++ and parallelized using MPI such that a new generation is created (line 7) once the objective function values have been computed for n new individuals (line 6).

A. Particle tracking and lattice configuration

The particle tracking code TRACY [5] is used to compute the objective function values [Eq. (27)] and constraint violations [Eq. (28)] for a given design point $\vec{d}$ [Eq. (25)]. TRACY is a flexible and well-tested beam dynamics library that was also used for SLS [31]. It uses a fourth-order symplectic integrator [32] for all multipole orders and allows fast tracking of single particles, enabling a trade-off between computation time and accuracy.

For the purpose of this paper the tracking code is modified, including the computation of the amplitude-dependent tune shifts and chromatic tune footprints (see Sec. II B) and the DA (see Sec. II A). Furthermore, an interface was created so that the values needed in Eqs. (27) and (28) could be obtained by OPT-PILOT.

In this paper the current lattice for SLS 2.0 is used [33]. This lattice is based on the multibend achromat scheme, including reverse bends and bends with a three-step longitudinal profile and additional quadrupole focusing. The magnet lattice has a threefold symmetry. For on-momentum particles a “virtual” 12-fold symmetry exists due to the proper adjustment of betatron phase advances between sextupoles in the insertion spaces. In this work the particles are tracked for 500 turns.

B. Constraint handling

Only some randomly chosen individuals (around 48%) satisfy the first constraint in Eq. (28), i.e., their tuning sextupoles are inside of the bounds. In the following, such individuals are called *feasible*. If an individual is infeasible, its objective function values are not computed. Instead, infeasible individuals are compared based on the severity of their constraint violations. Since the objective function values in Eq. (27) are at most one, this can be done by simply setting [see Eq. (28)]

$$F_i \leftarrow 2 + \max\{0, |t_i| - \kappa_{\max}\} \quad \text{for } i = 1, 2, \tag{29}$$

$$F_i \leftarrow 2 \quad \text{for } i = 3, 4, 5. \tag{30}$$

On the other hand, for individuals that violate the second constraint in Eq. (28) (i.e., when tune resonances are crossed) the objective function value computed in line 2 or 6 of Algorithm 1 is penalized as

$$F_i \leftarrow F_i + \mathtt{penalty}, \tag{31}$$

where [see Eqs. (17) and (23)]

$$\mathtt{penalty} = \alpha_1 \cdot \mathtt{ctfp} + \alpha_2 \cdot \mathtt{adts}. \tag{32}$$

If $\mathtt{penalty} \geq 1$ at least one of the tune footprints extends so far outside the triangle that this individual cannot be considered better than all infeasible individuals. Therefore, the possibility that the penalized objectives of this feasible individual are compared with constraint violations of an infeasible individual is allowed. This either results in the standard behavior, i.e., the feasible individual being chosen, or the infeasible individual being chosen—in which case its tuning sextupoles are likely close to the admissible bounds.

In this work $\alpha_1 = 0.01$ and $\alpha_2 = 1$ are used.

C. Results

All computations in this paper are run on Intel Xeon Gold 6152 nodes of the PSI Merlin cluster. An optimization using OPT-PILOT that was run for 48 h on three nodes (i.e., 132 processes), with $M = 300$ and $n = 130$ (see Sec. III above III A), computed 829 generations. The quality of a generation is quantified by counting the number of distinct design points in that generation which satisfy the constraints in Eq. (28) and have all of the objective function values better than those of the manually obtained solution. The values for a few representative generations, including the last one, are shown in Table II. In particular, since the values of the nonnegative objective functions F_4 and F_5 for the manually obtained solution are zero, they are also zero for all of the newfound points counted in Table II. This follows from the fact that the newfound points counted in Table II have all of the objective function values better than those of the manually obtained solution, in particular F_4 and F_5.

When comparing the results it should also be taken into account that the DAs of the manually obtained solution were optimized beyond the linear aperture limits. The last step of finding the manually obtained solution includes supervising the amplitude-dependent footprints, chromatic footprints, and dynamic aperture estimate while changing the multipole knobs accordingly [34]. This can be considered a "fuzzy" manual local optimization routine. The starting point of this local procedure is given by the lattice designer and involves a less quantifiable procedure.

From these 31 good points from the last generation (see Table II) three points are chosen based on different criteria and their objective function values are compared to those of the manually obtained solution in Table III. The design point called `point-1` is chosen because it has the lowest value of the objective F_1, $\mathrm{DA}_{-0.03} = 0.021$, and `point-2` because it has the lowest value of F_2 and, coincidentally, also of F_3, with values $\mathrm{DA}_{-0.03} = 0.001$ and $\mathrm{DA}_{0.03} = 0.002$, respectively. The design point `point-3` was chosen because it improves all three objectives by a comparable amount (for all of these points $F_4 = F_5 = 0$).

To sum up, using the massively parallel implementation OPT-PILOT of a MOGA, many design points with very good objective function values are found, but the run-time (48 h) is quite long.

TABLE II. The number of (nof) design points (pts) in a specific generation that satisfy the constraints in Eq. (28) and have all of the objective function values better than those of the manually obtained solution, referred to as the "design solution." This is abbreviated as "nof pts better." Generation 829 is the last one that is considered because the optimization was stopped after 48 h. All of the objective function values are computed with 500 turns in TRACY.

Generation	100	200	300	400	500	829
nof pts better	1	10	17	18	26	31

TABLE III. A comparison of the manually obtained design solution with three good points found in the optimization with OPT-PILOT. The objective function values are computed with 500 turns in TRACY, and all of these design points satisfy the constraints in Eq. (28). Out of the 31 design points in generation 829 computed in 48 h (see also Table II), `point-1` is chosen as the design point that has the lowest value of F_1 and `point-2` is chosen to have the lowest value of F_2 (coincidentally, it also has the lowest value of F_3). `point-3` is chosen to improve all of these three objectives by a comparable amount. The column labeled "gen" shows the generation in which the specific point was found.

Objective	F_1	F_2	F_3	F_4	F_5	gen
Design solution	0.032	0.004	0.011	0	0	
`point-1`	**0.021**	0.003	0.010	0	0	763
`point-2`	0.031	**0.001**	**0.002**	0	0	769
`point-3`	0.025	0.001	0.005	0	0	807

IV. BUILDING THE SURROGATE MODEL

The convergence of the optimization method can be improved by using, e.g., k-means clustering [4], ANN [15] or Gaussian process models [16,17] to preselect the points that need to be evaluated. Alternatively, an ANN surrogate model can be trained to approximate the objective function values and then used in the optimization [18]. Due to the encouraging results shown in [18], available tools and promising preliminary computations, the approach in this paper is based on training and using an ANN surrogate model.

First, a random feasible sample is created and evaluated using TRACY. In particular, around 7.5×10^4 design points [see Eq. (25)] are chosen uniformly at random from the intervals in Eq. (26). Their feasibility (see Sec. III B) is then checked according to Eq. (24) and 3×10^4 feasible points are chosen. The run-time for this is negligible. These feasible points are evaluated using TRACY and divided into training (70%), validation (20%) and test set (10%). This took 9 h 6 min on five nodes (220 cores). Since all of these 3×10^4 feasible points can be evaluated at the same time, more nodes can efficiently be utilized. For comparison, in Sec. III C the number of new individuals in a generation is $n = 130$, so only 132 processes, i.e., three nodes, are used. In the case of the optimization from Sec. III C it can be seen from $n = 130$ that in 829 generations around 108×10^3 points are evaluated. However, only some of those evaluations include the expensive TRACY run. The sample size 3×10^4 is chosen as a sizable fraction of this number, while also taking into account that the goal of this approach is to speed up the optimization from Sec. III C.

Second, this random feasible sample is used to train the ANN surrogate model. In particular, a feed-forward ANN with N_{layers} hidden layers is used. The first hidden layer has N_{neurons} neurons while the others have $2N_{\text{neurons}}$ neurons. The activation function is ReLU and the loss function is the mean squared error. The model is generated in PYTHON using the KERAS[35] API on top of TENSORFLOW [36], with some functionality taken from MLLIB [37]. The TALOS framework [38] is used to find good hyperparameters,

$$N_{\text{layers}} \in \{4, 5, 6\}, \qquad N_{\text{neurons}} \in \{32, 64, 128\}, \tag{33}$$

and the batch size for the stochastic gradient descent algorithm Adam [39],

$$N_{\text{batch}} \in \{128, 256\}. \tag{34}$$

Other parameters for Adam are the default ones [40] from KERAS, including the learning rate 0.001. A comparison of the six best combinations is shown in Table IV. On one core this took 52 min.

The dependence of the training and validation loss on the number of training epochs for the case with the smallest validation loss (Table IV), i.e., the hyperparameters,

$$N_{\text{layers}} = 5, \qquad N_{\text{neurons}} = 64 \quad \text{and} \quad N_{\text{batch}} = 128, \tag{35}$$

is shown in Fig. 2. A comparison of this ANN surrogate model with the particle tracking results in TRACY is shown in Fig. 3. The comparison is performed on the test set, i.e., random feasible design points that were not used for training. In each subplot the x and y coordinates are the values computed with the ANN surrogate model and TRACY, respectively. The line $y = x$ indicates perfect agreement. In the case of F_4 and F_5 the surrogate model prediction is replaced by the closest value from the set in Eq. (19) and, to facilitate the presentation of the results, the average of these values is shown in the second row, first column. Moreover, for `ctfp` and `adts` negative predictions are set to zero.

TABLE IV. A comparison of hyperparameters (only the six best combinations). The training is stopped if there is no improvement in the validation loss for 100 epochs.

	Epochs	Validation loss	N_{layers}	N_{neurons}	N_{batch}
1	454	0.002487	5	64	128
2	554	0.002494	4	64	128
3	330	0.002504	6	64	128
4	415	0.002521	6	64	256
5	513	0.002532	4	64	256
6	922	0.002550	5	32	128

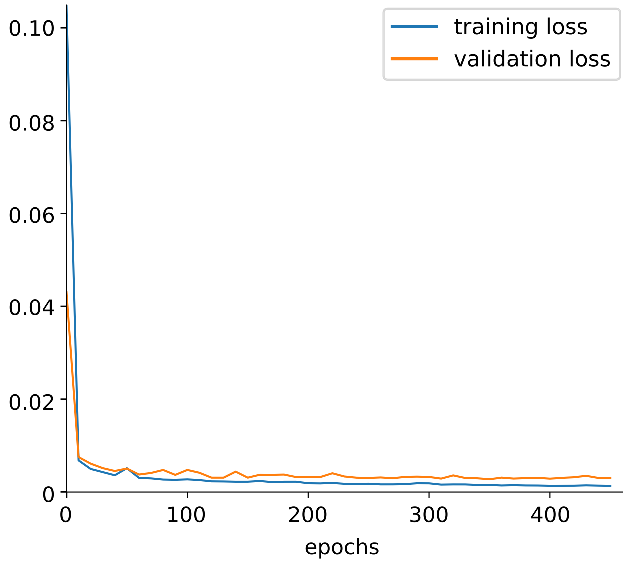


FIG. 2. The training loss (blue) and the validation loss (orange) as a function of the number of training epochs. The design point from Eq. (25) is considered and the functions that are approximated are the five objective functions from Eq. (27), as well as `ctfp` and `adts` from Eq. (28) (seven in total). The size of the random sample is 3×10^4 and the hyperparameters are: $N_{\text{layers}} = 5$, $N_{\text{neurons}} = 64$ and $N_{\text{batch}} = 128$, with the ReLU activation function. The training was stopped once there was no improvement in the validation loss for 100 epochs.

V. OPTIMIZING WITH THE SURROGATE MODEL

In this section, the MOGA implemented in the PYTHON PYMOO [41] module is used for the optimization, with the ANN surrogate model from Sec. IV used to predict the necessary figures of merit [see Eqs. (27) and (28)].

This is compared to the optimization using OPT-PILOT as described in Sec. III C, to preserve the solution quality while speeding up the optimization. Additionally, as in Sec. III C, the candidate solutions are again compared with the manually obtained solution.

The crossover, mutation and constraint handling used with PYMOO are therefore chosen to be as close as possible to the ones used with OPT-PILOT.

A. Direct approach

As a first approach, an optimization with $M = 10^4$ individuals in a generation is run for 1000 generations. This took 60 min on one core. The points in generation 1000 are then reevaluated using TRACY (3 h 50 min on five nodes) and the comparison is shown in Fig. 4, blue color. It can be seen from the scale that the objective function values of these design points are very good compared to the values computed for random points (see Fig. 3). For example, in the case of $F_2 = \mathrm{DA}_0$ the test sample (Fig. 3) achieved values up to around 0.8 and the optimized set of design

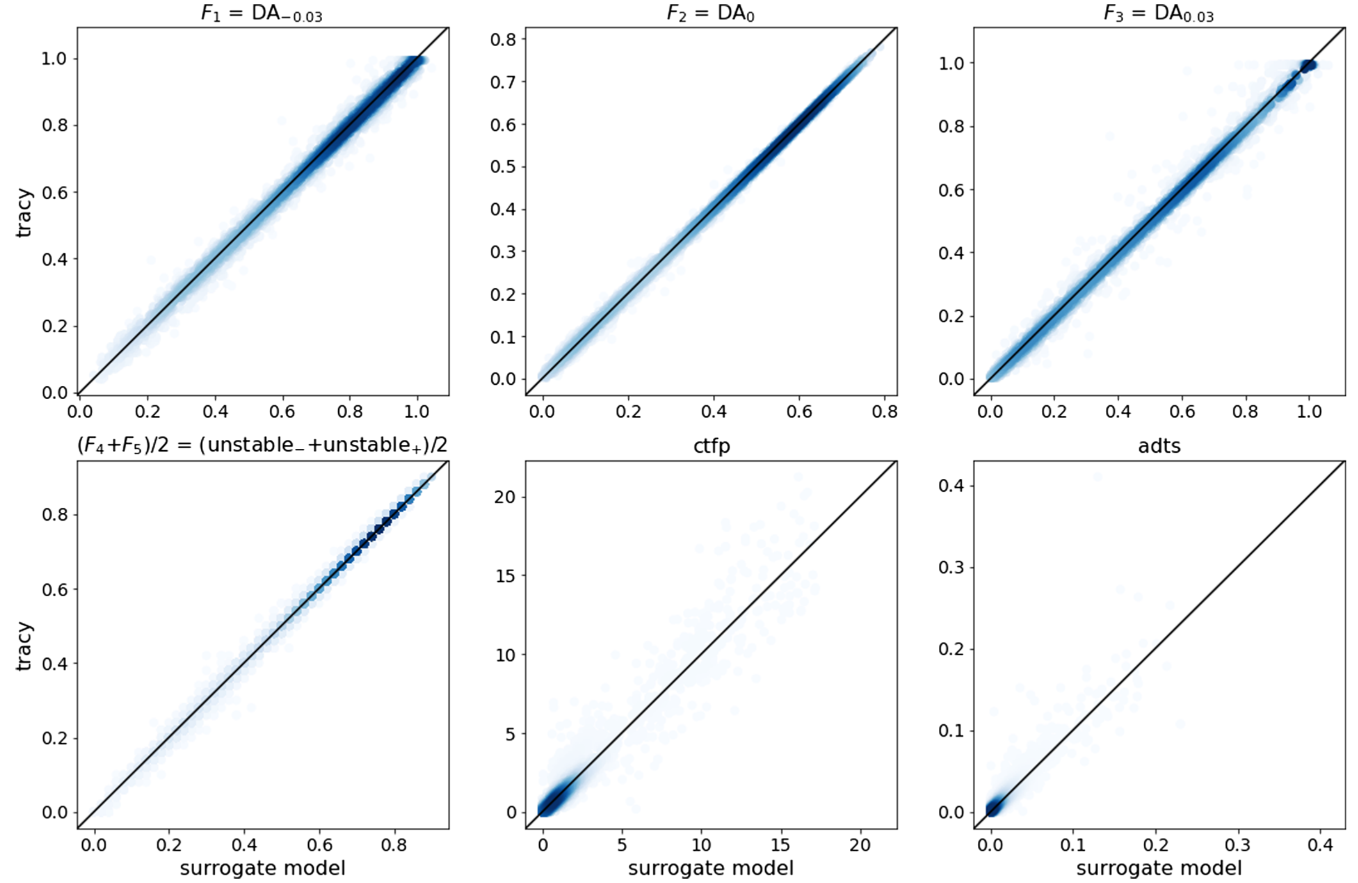


FIG. 3. The surrogate model quality on the test set, i.e., random design points that were not used for training. The surrogate model was trained using a training set of size 2.1×10^4 and a validation set of size 6000. Its quality is tested on a test set of size 3000. In each subplot the x and y coordinates are the values computed with the surrogate model and TRACY, respectively (a point on the line $y = x$ would be perfect agreement). Darker blue colors represent higher design point densities.

points (Fig. 4, blue) always has this value below 0.031. However, the agreement between the surrogate model predictions (x axis) and the values obtained with TRACY (y axis) for the optimized set of design points (Fig. 4, blue) is quite poor. In particular, for F_1, F_2 and F_3 the surrogate model prediction (Fig. 4, blue, x axis) is generally much smaller than the value computed in TRACY (Fig. 4, blue, y axis)—the points that seem good during the optimization with the surrogate model turn out to be mediocre. Therefore, despite the initial surrogate model quality seen in Fig. 3 as evaluated on random feasible points, the surrogate model quality evaluated on good design points, such as those computed in generation 1000, is not adequate for optimization. For example, none of the design points in generation 1000 have F_3 below 0.012 (Fig. 4, first row, third column, blue color). Since the value of F_3 for the manually obtained solution is 0.011 (see Table III), none of these points outperform it.

In total, this approach took around 14 h 48 min. For comparison, the optimization with OPT-PILOT that was run for 48 h on three nodes computed 829 generations (with $M = 300$ and $n = 130$), where 31 points satisfy the constraints and have all objective functions better than the design solution (see Sec. III C). The optimization with the surrogate model is 3.2× faster, but the solution quality is not as good—there are no points in the last generation that satisfy the constraints in Eq. (28) and have all of the objective function values better than those of the manually obtained solution. The comparison is clearly presented in Table V.

To improve the solution quality, the quality of the surrogate model predictions has to be much better for points with good objective function values. To achieve this, in the next section the surrogate model will be retrained during the optimization. This way, instead of using additional random feasible points to improve the surrogate model predictions on points with good objective function values, points found during the optimization will be used, similarly to [18].

B. Retraining the surrogate model

The second approach, intended to leverage the fast computation time of the ANN surrogate model optimization

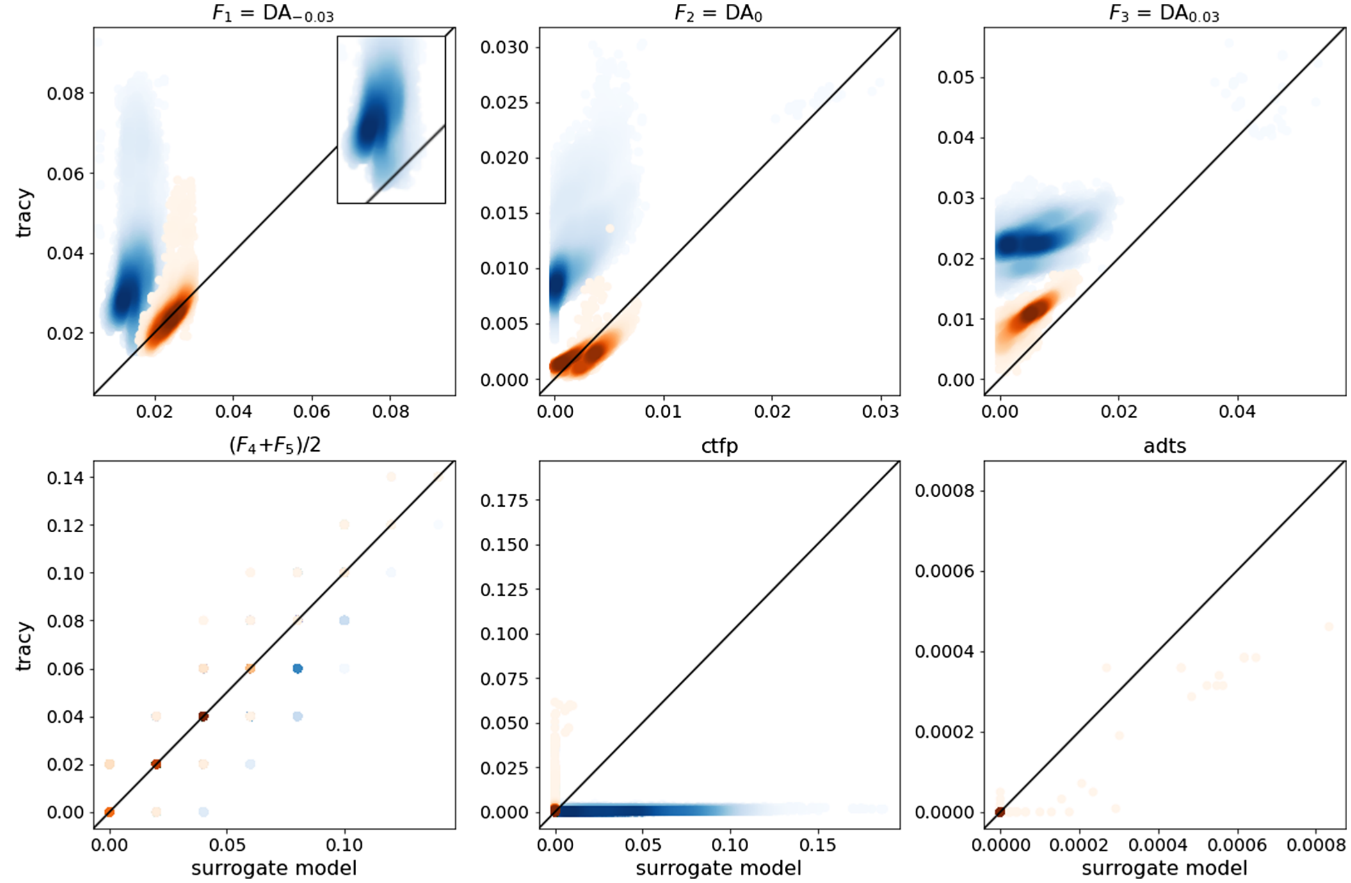


FIG. 4. Blue color: the quality of the predictions of the surrogate model from Fig. 3 on the design points in generation 1000 (computed with the approach from Sec. V A). For all points the value of `adts` is computed as zero using TRACY and predicted to be zero using the surrogate model. Orange color: the quality of the predictions of the third surrogate model from the approach in Sec. V B on the design points in generation 1000. In each subplot the x and y coordinates are the values computed with the surrogate model and TRACY, respectively (a point on the line $y = x$ would be perfect agreement). Darker colors represent higher design point densities. Since the plot in the first row, first column contains some overlap, the blue part underneath is shown in the top right corner. There is virtually no overlap in the second and third column of the first row. In the second row it can be concluded that the blue regions are located below the orange ones whenever the blue color is virtually absent from the plot—this is the case in the first and third column of the second row. The benefit of the approach from Sec. V B can clearly be seen. First, the orange regions overlap better with the line $y = x$ which means that the surrogate model predictions at the end of the optimization are more accurate. Second, the orange points mostly have smaller y coordinate values which means that they have better objective function values as computed with TRACY.

and the solution quality of the OPT-PILOT optimization, is the following. Instead of training the surrogate model once and then using it for a MOGA optimization with "nof generations," the surrogate model is retrained "nof retrainings" times during the optimization, where

$$\text{“nof retrainings”} \in \{1, \text{“nof generations”} - 1\}. \qquad (36)$$

In particular, the surrogate model is retrained after generations $m_1, \ldots, m_{\text{“nof retrainings”}}$. As in Sec. V A, "nof generations" $= 1000$ is used. To facilitate parallel computation and keep the total run-time for training the surrogate models low, the number of retrainings, denoted "nof retrainings" in Eq. (36), is kept low. Therefore, the values of "nof retrainings" that are considered are, in order, $1, 2, \ldots$. To keep the total number of TRACY evaluations below 3×10^4, the surrogate model is first trained on 10^4 random feasible points and retrained in generation m_i ($i \in \{1, \ldots, \text{“nof retrainings”}\}$) with the random feasible points used previously and 5000 points from generation m_i. This way the combined size of the samples is below 3×10^4 for "nof retrainings" $\in \{1, 2, 3\}$ and exactly 3×10^4 for "nof retrainings" $= 4$. First, $m_1 = 50$ was chosen due to a more rapid change of the objective function values in the beginning of the optimization. For example, in generation 50 all of the points had $F_2 = \mathrm{DA}_0 < 0.07$, and the values achieved in 1000 generations are generally comparable to the value of the manually obtained solution, $F_2 = 0.004$ (Table III). Second, $m_2 = 500$ was chosen based on the quality of the retrained surrogate model on a few of the subsequent generations which are multiples of 100. As will be shown in the rest of this section, the results

TABLE V. Solution quality and run-time for different optimization methods. "SM" is an abbreviation for "surrogate model." OPT-PILOT denotes the massively parallel MOGA implementation combined with TRACY, in particular the optimization from Sec. III C. "SM (3×10^4)" and "SM + retrain (2×10^4)" refer to the approaches described in Secs. V A and V B, respectively. "SM + retrain (10^4)" and "SM + retrain (5000)" are described in Sec. V C for $N = 5000$ and $N = 2500$, respectively. The number in the parentheses is the combined size of the used samples. "SM + retrain (2×10^4)" therefore corresponds to $N = 10^4$. "nof pts better" refers to the number of design points in the last generation that satisfy the constraints in Eq. (28) and have all objectives better than the manually obtained solution. To determine the quality of a generation, "nof pts better" should be used together with the information in Table VI. For example, "nof pts better" is higher for $N = 5000$ than for $N = 10^4$, but better individual values for F_1 and F_2 are achieved when $N = 10^4$ (see Table VI, point-4 and point-5). The quality of both approaches is comparable to the quality of the approach from Sec. III C, denoted OPT-PILOT. "reeval all" refers to the case where all 10^4 individuals in the last generation are reevaluated using TRACY, which accounts for around 3 h 50 min of the total run-time. "reeval 10%" refers to the case where only 1000 of the 10^4 individuals in the last generation are reevaluated. The speedup is computed with respect to the OPT-PILOT approach in the first column.

	OPT-PILOT	SM (3×10^4)	SM + retrain (2×10^4)	SM + retrain (10^4)	SM + retrain (5000)
nof pts better	31	0	148	368	87
Run-time (reeval all)	48 h	14 h 48 min	12 h 15 min	8 h 31 min	6 h 33 min
Core hours (reeval all)	6336	2847	2325	1593	1210
Speedup (reeval all)	1.0	3.2	3.9	5.6	7.3
Run-time (reeval 10%)		11 h 21 min	8 h 52 min	5 h 5 min	3 h 10 min
Core hours (reeval 10%)		2089	1578	838	465
Speedup (reeval 10%)		4.2	5.4	9.4	15.1

quality in generation 1000 is in this case comparable to the one in Sec. III C, so "nof retrainings" is set to two.

The quality of the predictions of the third surrogate model on the 10^4 design points from generation 1000 is shown in Fig. 4, orange color. The total run-time, including reevaluating the entire last generation using TRACY, is now around 12 h 15 min, which is a speedup of 3.9× compared to the approach from Sec. III C. However, not all points in the last generation need to be reevaluated—if only 1000 of these points (i.e., 10%) are reevaluated, the total run-time is 8 h 52 min, which is a speedup of 5.4× (see Table V). The 1000 points to be reevaluated can be chosen based on the values of the predictions.

There are 148 design points in the last generation that satisfy the constraints in Eq. (28) and have all of the objective function values better than those of the manually obtained solution, which is significantly more than the 31 points found in Sec. III C (see Tables II and V).

As in Table III, out of these 148 design points, the ones with the lowest value of F_1, F_2 and F_3 are shown in Table VI and referred to as point-4, point-5 and point-6, respectively. The quality of these points is clearly comparable with point-1 and point-2 from Table III. For example, the lowest value of $F_1 = \mathrm{DA}_{-0.03}$ achieved with the new approach is better than the lowest values of F_1 achieved in Sec. III C. Namely, $F_1 = 0.020$ for point-4 (new approach) which is lower, i.e., better, than $F_1 = 0.021$ for point-1 (approach from Sec. III C). On the other hand, the lowest value of $F_3 = \mathrm{DA}_{0.03}$ achieved with the new approach is not as good as the lowest values of F_3 achieved in Sec. III C. Namely, $F_3 = 0.004$ for point-6 (new approach) which is higher than $F_3 = 0.002$ for point-2 (approach from Sec. III C). The lowest achieved value of $F_2 = \mathrm{DA}_0$ is the same for both approaches, namely $F_2 = 0.001$ (point-5 and point-2). Furthermore, there are design points whose quality is comparable to that of point-3 in Table III, e.g., a design point with

$$(F_1, F_2, F_3, F_4, F_5) = (0.026, 0.001, 0.004, 0, 0), \qquad (37)$$

$\mathtt{ctfp} = 0$ and $\mathtt{ctfp} = 0$.

TABLE VI. The points with the smallest value of F_1 (point-4, 7, 10), F_2 (point-5, 8, 11) and F_3 (point-6, 9, 11), out of all the points (first row in Table V) that satisfy the constraints in Eq. (28) and have all objectives better than the manually obtained solution (first row in Table III). All the digits of the objective function values computed with 500 turns in TRACY are used for the comparison, and the bold numbers denote the minimal value of the respective objective before rounding. The last column denotes the size of the initial sample. The case with $N = 10^4$ is described in Sec. V B. The cases with $N = 5000$ and $N = 2500$ are described in Sec. V C.

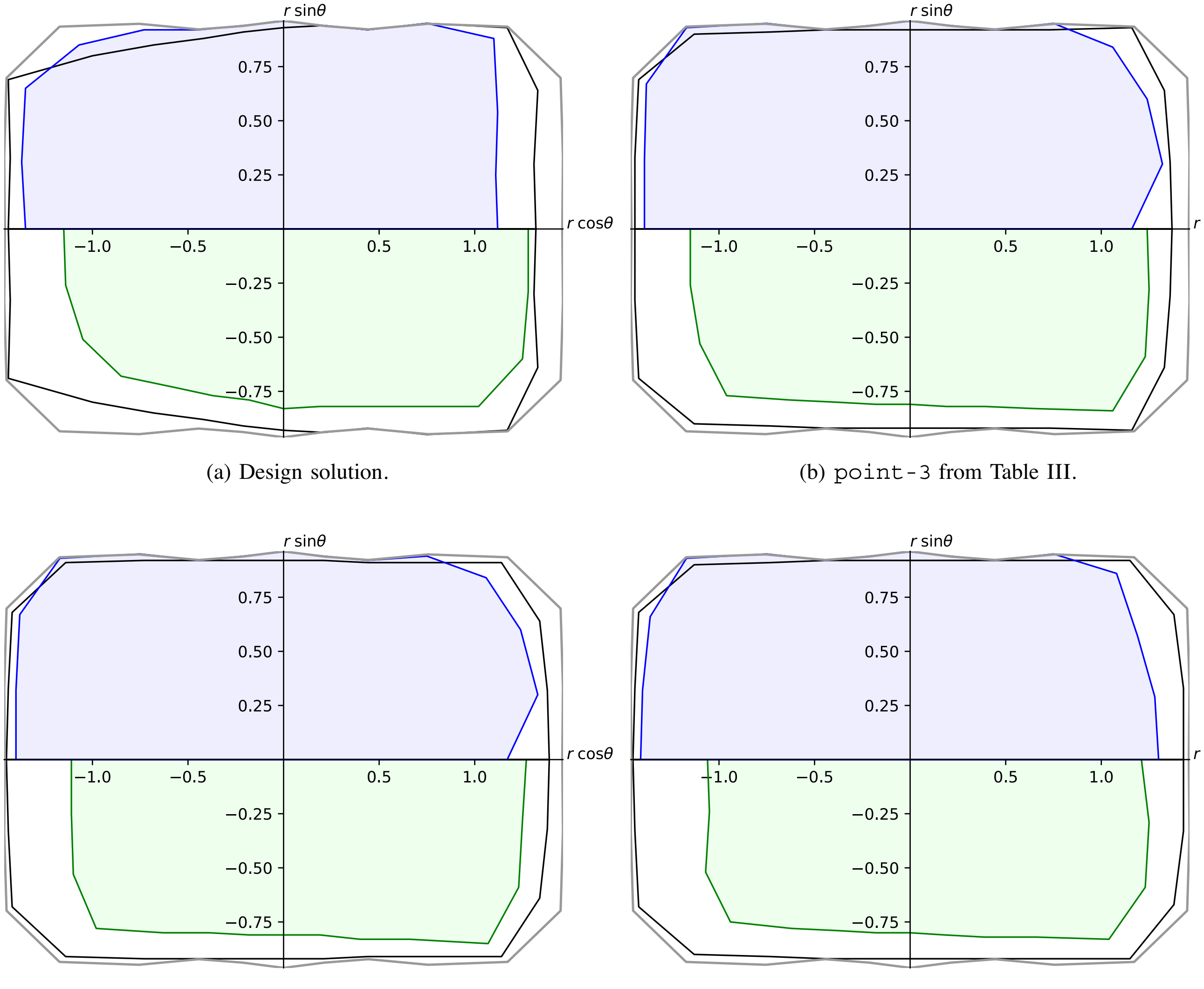
Objective	F_1	F_2	F_3	F_4	F_5	N
point-4	**0.020**	0.004	0.011	0	0	10^4
point-5	0.029	**0.001**	0.008	0	0	10^4
point-6	0.028	0.002	**0.004**	0	0	10^4
point-7	**0.024**	0.003	0.008	0	0	5000
point-8	0.030	**0.002**	0.006	0	0	5000
point-9	0.029	0.002	**0.004**	0	0	5000
point-10	**0.025**	0.003	0.011	0	0	2500
point-11	0.032	**0.001**	**0.005**	0	0	2500

C. Using a smaller sample for training

In Sec. V A a large sample of 3×10^4 random feasible design points was used to illustrate that, regardless of the surrogate model quality on random points, the quality on points with good objective functions is very poor. In Sec. V B the combined size of the samples was 2×10^4 instead of 3×10^4. In this section the size of the samples is further reduced.

The approach is the same as the one from Sec. V B. An initial sample of size N is used to train the first surrogate model. The second surrogate model is trained in generation $m_1 = 50$ using these N points together with $N/2$ of the points found in generation m_1. The third surrogate model is trained in generation $m_2 = 500$ using also $N/2$ of the points found in generation m_2. While in Sec. V B $N = 10^4$ was used, this is now lowered to $N = 5000$ and $N = 2500$.

As shown in Table V, the number of points that satisfy the constraints in Eq. (28) and have all of the objective function values better than those of the manually obtained solution is now 368 for $N = 5000$ and 87 for $N = 2500$. The run-time including the reevaluation of the entire last generation is 8 h 31 min (speedup 5.6×) and 6 h 33 min (speedup 7.3×), respectively. If only 1000 of the points in the last generation (i.e., 10%) are reevaluated using TRACY,



(a) Design solution.

(b) `point-3` from Table III.

(c) The point from Eq. (37).

(d) `point-11` from Table VI.

FIG. 5. Transverse DAs in Floquet space ($r\cos\theta, r\sin\theta$) from Eq. (1) of the solution candidates for $\delta = -0.03$ (green), $\delta = 0.03$ (blue) and $\delta = 0$ (bold black line), computed using TRACY as described in Sec. II A. The particles are tracked for 500 turns. The linear aperture, computed from the lattice model by ignoring sextupolar and higher-order fields, is shown in gray. For a clearer presentation of the results, only half of the off-momentum apertures is shown. This is sufficient due to machine-plane symmetry. Each subplot corresponds to one candidate solution: the manually obtained solution (referred to as the "design solution," top left), `point-3` from Table III (top right), the design point from Eq. (37) (bottom left) and `point-11` from Table VI (bottom right). The relative relationships of the DA area sizes are consistent with the computed objective function values.

the speedups in the cases $N = 5000$ and $N = 2500$ are 9.4× and 15.1×, respectively. A detailed comparison is shown in Table V.

Furthermore, in addition to counting "nof pts better" (Table V), out of these design points the points with the lowest values of F_1, F_2 and F_3 are shown in Table VI. Both approaches from this section found numerous points with very good objective function values in a significantly shorter time—most notably the 3 h 10 min for the fastest case (instead of 48 h needed for OPT-PILOT).

VI. CANDIDATE SOLUTIONS

In this section some of the candidate solutions obtained in Secs. III and V are compared and further analyzed. In particular, out of the points found in Sec. III C with the

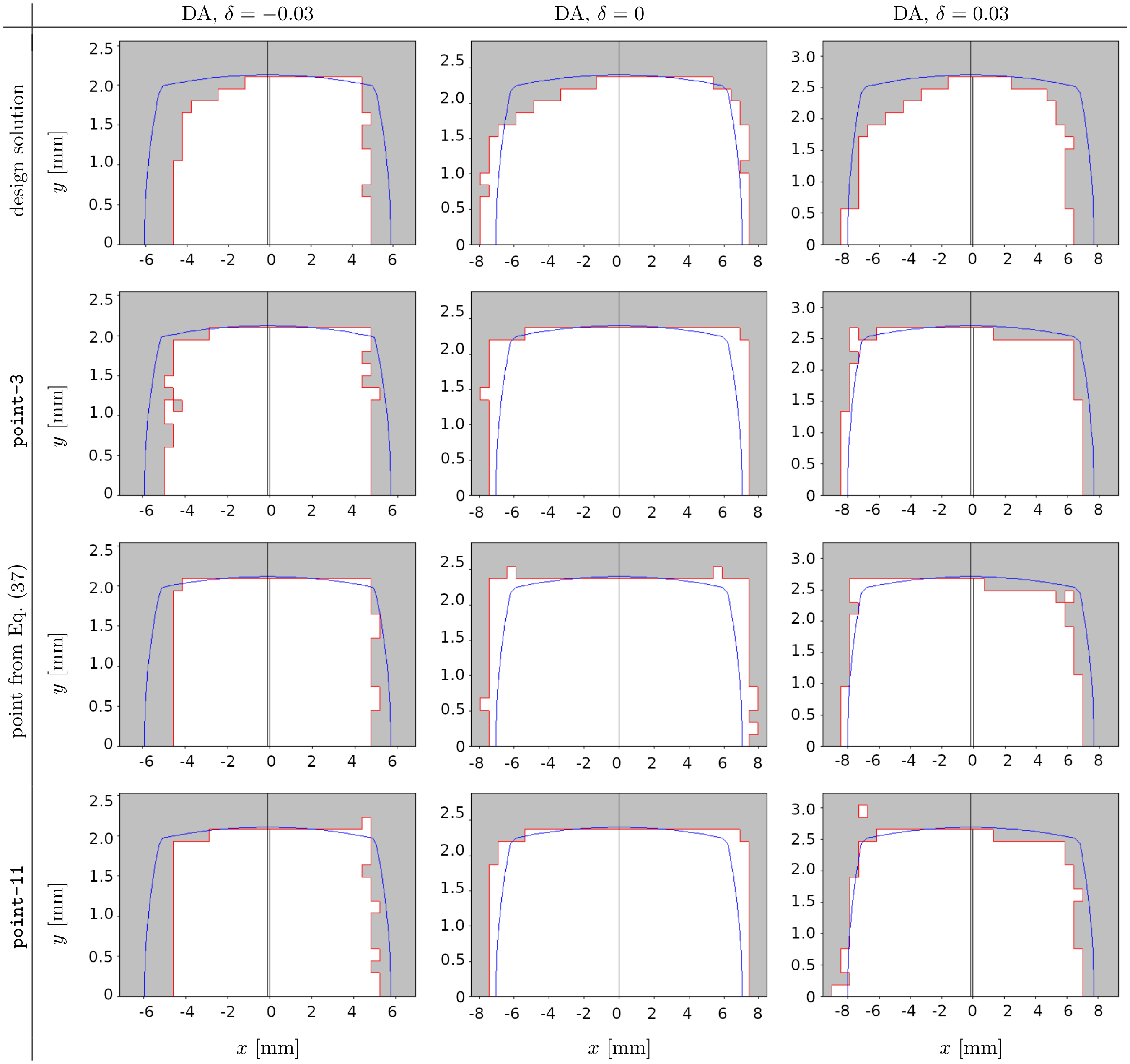


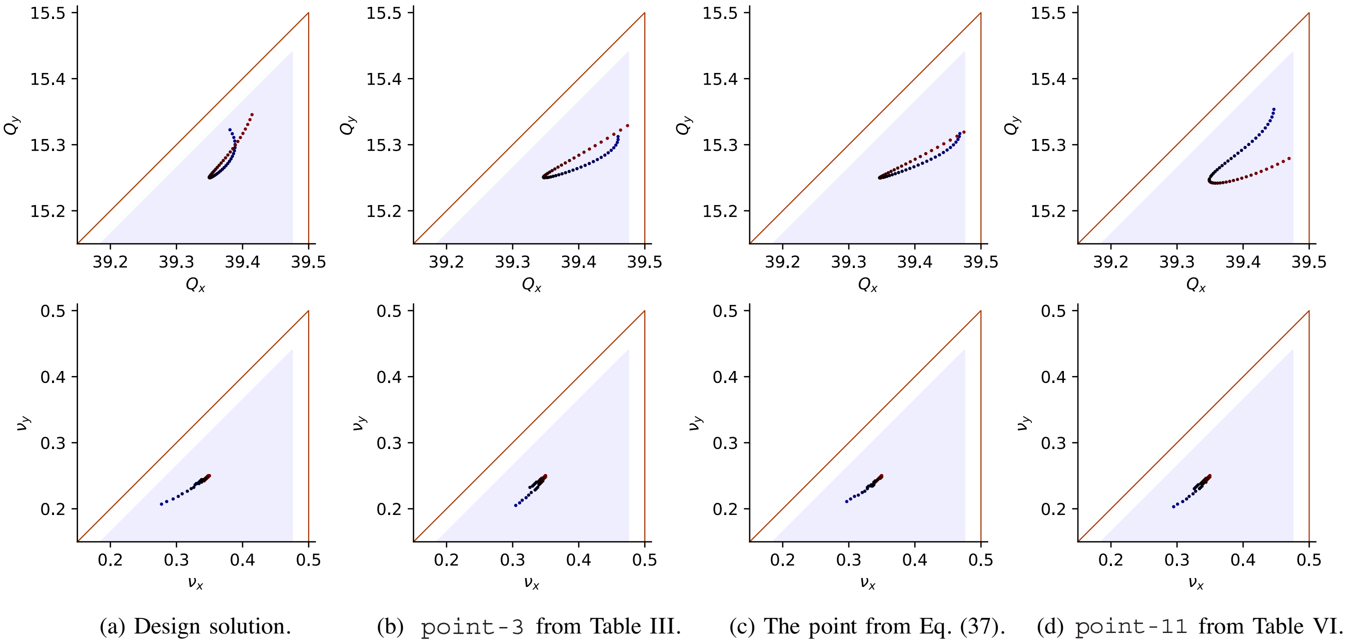
FIG. 6. The columns show the transverse DAs at $\delta = -0.03$, $\delta = 0$ and $\delta = 0.03$, recomputed in OPA for the manually obtained solution (referred to as the "design solution," first row), point-3 from Table III (second row), the design point from Eq. (37) (third row) and point-11 from Table VI (fourth row). The particles are tracked for 500 turns on a grid in the transverse plane. The blue silhouette shows the linear aperture limits computed by OPA, assuming elliptical vacuum chamber parts [42]. The results are consistent with the TRACY computation in Fig. 5.

massively parallel implementation of a MOGA, `point-3` (see Table III) is chosen. Out of the points found with the new method in Sec. V B ($N = 10^4$), the point shown in Eq. (37) is chosen. Out of the points found with the new method in Sec. V C, using the smallest considered combined sample size ($N = 2500$), `point-11` (see Table VI) is chosen. All of these points are compared with the manually obtained solution, referred to as the "design solution."

For each of these solution candidates, the transverse DAs at three different energies ($\delta \in \{-0.03, 0, 0.03\}$) are shown in Fig. 5. In each subplot the bold black line shows the boundary of the on-momentum DA, computed with 500 turns in TRACY as described in Sec. II A. As indicated by the smaller values of the objective function $F_2 = \mathrm{DA}_0$, all three candidate solutions have a larger on-momentum DA than the design solution. The green and blue areas show the off-momentum DA for $\delta = -0.03$ and $\delta = 0.03$, respectively. These transverse DAs correspond to the objective functions $F_1 = \mathrm{DA}_{-0.03}$ and $F_3 = \mathrm{DA}_{0.03}$, respectively. In the case of $\delta = 0.03$ it can clearly be seen that the computed transverse DA for all three new candidate solutions is larger than that of the design solution, as indicated by the smaller values of F_3. In the case of $\delta = -0.03$ the area of the transverse DA for `point-11` is of a similar size to that of the design solution. This agrees with the fact that for both of these design points $F_1 = 0.032$ (see Tables III and VI). The other two new candidate solutions have lower values of F_1 ($F_1 = 0.025$ in Table III and $F_1 = 0.026$ in Eq. (37) and the computed areas of the transverse DA at $\delta = -0.03$ for these two design points are larger than that of the design solution, which is again the desired behavior.

Additionally, the transverse DAs at the three considered energies are computed with OPA [42] (Fig. 6). In OPA, for each energy, the transverse DA is sampled on a two-dimensional grid of points. This results in a better approximation of the DA (cf. Sec. II A), at the expense of computation time. For the considered candidate solutions, the OPA-computed transverse DAs are larger than that of the manually obtained solution. This is in agreement with the relative relationships of the TRACY-computed transverse DAs (Fig. 5).

The chromatic tune footprint (Sec. II B 1) and amplitude-dependent tune footprint (Sec. II B 2) for the three new solution candidates and the design solution are shown in Fig. 7. In each subplot, the outer triangle (red) is the one formed by three first- and second-order resonances around the on-momentum tune [see Eq. (12)]. The inner triangle (blue) includes the margin from Sec. II B 1. For all of the candidate solutions, $F_{4,5} = \mathtt{unstable}_{\mp} = 0$ [see Eq. (18)] and the second constraint in Eq. (28) is satisfied,



(a) Design solution. (b) `point-3` from Table III. (c) The point from Eq. (37). (d) `point-11` from Table VI.

FIG. 7. Chromatic tune footprint in the range $\delta \in [-0.05, 0.05]$ (top row) and amplitude-dependent tune footprint (bottom row) for the solution candidates: the manually obtained solution (referred to as the "design solution," first column), `point-3` from Table III (second column), the design point from Eq. (37) (third column) and `point-11` from Table VI (fourth column). The first- and second-order resonances are shown as red and the blue triangle area includes the additional margin around the resonance lines. One of the goals of the optimization was to prevent the tune resonances from being crossed by constraining the tune footprints inside the triangle with margins (blue triangle). It can be seen that the definition of $\mathtt{unstable}_{\mp}$, `ctfp` and `adts` in Sec. II B, as well as their treatment in the optimization problem formulation in Sec. II D successfully constrained the tune footprints.

i.e., $\mathtt{ctfp} + \mathtt{adts} = 0$ [see Eqs. (17) and (23)], so all footprints are located inside the inner triangle.

VII. CONCLUSIONS

In this paper a multiobjective genetic algorithm is used to find a good dynamic aperture and energy acceptance for the Swiss Light Source upgrade. To speed up this expensive computation, artificial neural network surrogate models are used in the optimization. It is shown that a straightforward use of the surrogate model is not good enough for this problem, so the surrogate model is retrained during the optimization. Compared to a massively parallel implementation of a multiobjective genetic algorithm, this new optimization method results in an order of magnitude speedup. At the same time, the solution quality is preserved. In particular, tens of the design points in the last generation are better than the design solution in all of the considered objective functions.

The new, faster method makes it possible to include more design parameters in the optimization problem, such as the octupole strengths, which could further improve the solution quality. Furthermore, it allows for the inclusion of a more accurate and more expensive model, e.g., a model which includes nonlinear synchrotron oscillation. In this work the focus is on the lattice for the Swiss Light Source upgrade, but an analogous procedure could easily be used for a different lattice or a different machine.

ACKNOWLEDGMENTS

M. Aiba provided the manually obtained solution. J. Kallestrup found useful reference suggestions. R. Bellotti helped with TensorFlow and with proofreading a part of the manuscript. M. Zacharias helped with TensorFlow. T. Schietinger proofread the manuscript.

[1] B. Riemann and A. Streun, Low emittance lattice design from first principles: Reverse bending and longitudinal gradient bends, Phys. Rev. Accel. Beams **22**, 021601 (2019).

[2] J. Bengtsson and A. Streun, Robust design strategy for SLS 2, Report No. SLS2-BJ84-001, PSI, 2017.

[3] M. P. Ehrlichman, Genetic algorithm for chromaticity correction in diffraction limited storage rings, Phys. Rev. Accel. Beams **19**, 044001 (2016).

[4] Y. Li, W. Cheng, L. H. Yu, and R. Rainer, Genetic algorithm enhanced by machine learning in dynamic aperture optimization, Phys. Rev. Accel. Beams **21**, 054601 (2018).

[5] J. Bengtsson, TRACY-3, Self-consistent charged particle beam tracking code based on a symplectic integrator, available for download at https://github.com/jbengtsson/tracy-3.5.

[6] X. Huang and J. Safranek, Nonlinear dynamics optimization with particle swarm and genetic algorithms for SPEAR3 emittance upgrade, Nucl. Instrum. Methods Phys. Res., Sect. A **757**, 48 (2014).

[7] R. Husain and A. Ghodke, Constrained multiobjective optimization of storage ring lattices, Nucl. Instrum. Methods Phys. Res., Sect. A **883**, 151 (2018).

[8] J. Wu, Y. Zhang, Q. Qin, Y. Wang, C. Yu, and D. Zhou, Dynamic aperture optimization with diffusion map analysis at CEPC using differential evolution algorithm, Nucl. Instrum. Methods Phys. Res., Sect. A **959**, 163517 (2020).

[9] M. Kranjčević, B. Riemann, A. Adelmann, and A. Streun, Multiobjective optimization of the dynamic aperture for the Swiss Light Source upgrade, arXiv:2002.08685.

[10] W. Gao, L. Wang, and W. Li, Simultaneous optimization of beam emittance and dynamic aperture for electron storage ring using genetic algorithm, Phys. Rev. ST Accel. Beams **14**, 094001 (2011).

[11] Y. Li and L. Yang, Multiobjective dynamic aperture optimization for storage rings, Int. J. Mod. Phys. A **31**, 1644019 (2016).

[12] A. Hofler, B. Terzić, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis, Innovative applications of genetic algorithms to problems in accelerator physics, Phys. Rev. Accel. Beams **16**, 010101 (2013).

[13] L. Yang, Y. Li, W. Guo, and S. Krinsky, Multiobjective optimization of dynamic aperture, Phys. Rev. ST Accel. Beams **14**, 054001 (2011).

[14] L. Yang, D. Robin, F. Sannibale, C. Steier, and W. Wan, Global optimization of an accelerator lattice using multiobjective genetic algorithms, Nucl. Instrum. Methods Phys. Res., Sect. A **609**, 50 (2009).

[15] F. Wang, M. Song, A. Edelen, and X. Huang, Machine learning for design optimization of storage ring nonlinear dynamics, arXiv:1910.14220.

[16] X. Huang, M. Song, and Z. Zhang, Multiobjective multigeneration Gaussian process optimizer for design optimization, arXiv:1907.00250.

[17] M. Song, X. Huang, L. Spentzouris, and Z. Zhang, Storage ring nonlinear dynamics optimization with multiobjective multigeneration Gaussian process optimizer, Nucl. Instrum. Methods Phys. Res., Sect. A **976**, 164273 (2020).

[18] A. Edelen, N. Neveu, M. Frey, Y. Huber, C. Mayes, and A. Adelmann, Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems, Phys. Rev. Accel. Beams **23**, 044601 (2020).

[19] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks* (IEEE, New York, 1995), Vol. 4, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[20] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report No. TR06, Erciyes University Press, Erciyes, 2005.

[21] H. Shah-Hosseini, The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm, Int. J. Bio-Inspired Comput. **1**, 71 (2009).

[22] M. Dorigo, V. Maniezzo, and A. Colorni, The ant system: Optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B Cybern. **26**, 29 (1996).

[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[24] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach* (Springer, New York, 2002).

[25] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms* (Wiley, New York, 2009).

[26] N. Neveu, L. Spentzouris, A. Adelmann, Y. Ineichen, A. Kolano, C. Metzger-Kraus, C. Bekas, A. Curioni, and P. Arbenz, Parallel general purpose multiobjective optimization framework with application to electron beam dynamics, Phys. Rev. Accel. Beams **22**, 054602 (2019).

[27] I. V. Bazarov and C. K. Sinclair, Multivariate optimization of a high brightness dc gun photoinjector, Phys. Rev. Accel. Beams **8**, 034202 (2005).

[28] M. Kranjčević, S. G. Zadeh, A. Adelmann, P. Arbenz, and U. van Rienen, Constrained multiobjective shape optimization of superconducting rf cavities considering robustness against geometric perturbations, Phys. Rev. Accel. Beams **22**, 122001 (2019).

[29] Y. Ineichen, A. Adelmann, C. Bekas, A. Curioni, and P. Arbenz, A fast and scalable low dimensional solver for charged particle dynamics in large particle accelerators, Comput. Sci. Res. Dev. **28**, 185 (2013).

[30] Y. Ineichen, Toward massively parallel multiobjective optimization with application to particle accelerators, ETH Research Collection, Ph.D. thesis, ETH Zurich, 2013.

[31] M. Boege and A. Streun, Beam lifetime studies for the SLS storage ring, in *Proceedings of the 18th Particle Accelerator Conference, New York, 1999* (IEEE, New York, 1999).

[32] E. Forest and R. Ruth, Fourth-order symplectic integration, Physica (Amsterdam) **43D**, 105 (1990).

[33] A. Streun, SLS 2.0 baseline lattice, Internal Report No. SLS2-SA81-004-13, PSI, 2020.

[34] M. Aiba (personal communication).

[35] F. Chollet *et al.*, KERAS: The PYTHON deep learning library (2015), https://keras.io/.

[36] M. Abadi *et al.*, TENSORFLOW: Large-scale machine learning on heterogeneous systems, arXiv:1603.04467.

[37] A. Adelmann *et al.*, MLLIB, https://gitlab.psi.ch/adelmann/mllib.

[38] Autonomio TALOS, retrieved from http://github.com/autonomio/talos.

[39] D. P. Kingma and J. Ba, ADAM: A method for stochastic optimization, arXiv:1412.6980.

[40] KERAS API reference/Optimizers/Adam, https://keras.io/api/optimizers/adam.

[41] J. Blank and K. Deb, PYMOO: Multiobjective optimization in PYTHON, arXiv:2002.04504.

[42] OPA accelerator optics software with examples, http://ados.web.psi.ch/opa.