Title
Advances in global optimization of high brightness beams

Journal
International Journal of Modern Physics A, 34(36)

ISSN
0217-751X

Author
Qiang, Ji

Publication Date
2019-12-30

DOI
10.1142/s0217751x19420168

Peer reviewed

International Journal of Modern Physics A

Advances in Global Optimization of High Brightness Beams

Ji Qiang

Lawrence Berkeley National Laboratory, 1 Cyclotron Road
Berkeley, California 94720, USA
jqiang@lbl.gov

Received Day Month Year
Revised Day Month Year

High brightness electron beams play an important role in accelerator based applications such as driving x-ray free electron laser radiation. In this paper, we report on advances in global beam dynamics optimization of an accelerator design using start-to-end simulations and a new parallel multi-objective differential evolution optimization method. The global optimization results in significant improvement of the final electron beam brightness.

Keywords: unified differential evolution; global optimization; high brightness beam.

PACS numbers:29.27.Bd,41.60.Cr

1. Introduction

High brightness, coherent x-ray free electron laser (FEL) light sources provide an invaluable tool for scientific discovery in biology, chemistry, physics, and material science. Most of these FEL light sources use an accelerator as a beam delivery system to generate high quality electron beam needed for coherent x-ray radiation in undulators. As the quality of the electron beam such as emittance, peak current, and energy spread plays a critical role in the production of the x-ray radiation, it is important to optimize the electron beam quality during the accelerator design.

In the accelerator community, multi-objective genetic algorithm (MOGA) such as NSGA-II[1] has been widely used for beam dynamics optimization. However, in the evolutionary computation community, another population based optimization method, differential evolution method, has been actively studied in recent years.[2] The differential evolution method is a simple yet efficient population-based, stochastic, evolutionary algorithm for global parameter optimization.[3,4] In a number of studies, the differential evolution algorithm performed effectively in comparison to several stochastic optimization methods such as simulated annealing, controlled random search, evolutionary programming, the particle swarm method, and genetic algorithm.[3–6] In past studies, multiple mutation strategies were proposed in the differential evolution literature. The use of multiple mutation strategies makes the differential evolution algorithm complicated to use appropriately. Recently, we devel-

oped a new adaptive unified differential evolution (AuDE) algorithm for global optimization.[7] This algorithm uses only a single mutation expression, but encompasses almost all commonly-used mutation strategies as special cases. It is mathematically simpler than the conventional algorithm with its multiple mutation strategies, and also provides users the flexibility to explore new combinations of conventional mutation strategies during optimization. This single unified mutation strategy is further tuned to improve the performance by using only three control parameters instead of four control parameters in the original unified algorithm.[8]

The accelerator system for x-ray FEL typically consists of a photo-injector as a front end to produce a high brightness electron beam, a linear accelerator (or equivalent accelerator) to accelerate the electron beam to the designed energy and to compress the beam to high peak current, and a final beam transport system to deliver the beam for different undulator radiation stations. In past studies, the accelerator design was typically divided into two sections, the injector section and the linear accelerator (linac) section. The injector was designed using the theory of space-charge emittance compensation and the multi-objective beam dynamics optimization.[9–17] After the injector optimization, an optimal solution from the injector output was selected as an input to the downstream linear accelerator. Using the electron beam information from the injector, the linear accelerator was then designed using analytical model, single pass tracking, and multi-objective optimization.[18–21] However, the final beam quality does not only depend on the linear accelerator settings, but also depend on the initial electron beam phase space distribution. An optimal solution from the injector does not necessarily mean the best solution for the final beam quality. In this paper, we report on advances in global optimization of high brightness electron beam in a future x-ray FEL light source accelerator design based on start-to-end beam dynamics simulation using a new variable population with external storage (VPES) parallel multi-objective differential evolution method.

The organization of the paper is as follows: after the Introduction, we review the conventional differential evolution algorithm for a single objective optimization in Section 2; we present the new developed adaptive unified differential evolution algorithm in Section 3 and the variable population with external storage parallel multi-objective differential evolution algorithm in Section 4; we show some benchmark examples of the new optimization algorithms in Section 5; we present an application to the global longitudinal beam dynamics optimization in a future x-ray FEL accelerator design in Section 6.

2. Conventional Differential Evolution Algorithm for Single Objective Optimization

The differential evolution algorithm starts with a population initialization like the other evolutionary algorithms. Compared with the other evolutionary algorithms such as the genetic algorithm, the differential evolution algorithm makes use of the differences of parent solutions to attain the gradient information. This helps improve

the convergence speed of the algorithm in comparison to the genetic algorithm. Meanwhile, compared with the particle swarm method, the differential evolution algorithm has a cross-over stage to enhance the diversity of solutions. This helps the differential evolution algorithm to avoid converging to a local minimum solution.

A set of NP solutions in the control parameter space is randomly generated to form the initial population. This initial population can be generated by sampling from a uniform distribution within the allowed parameter space if no prior information about the optimal solution is available, or by sampling from a known distribution (e.g., Gaussian) if some prior information is available.

After initialization, the differential evolution algorithm updates the population from one generation to the next generation until reaching a convergence condition or until the maximum number of function evaluations is reached. At each generation, the update step consists of three operations: mutation, crossover, and selection. The mutation and the crossover operations produce new candidates for the next generation population and the selection operation is used to select the appropriate solutions among these candidates to be included in the next generation.

2.1. *Mutation strategies*

During the mutation stage, for each population member (target vector) $\vec{x}_i$, $i = 1, 2, 3, \cdots, NP$ at generation G, a new mutant vector $\vec{v}_i$ is generated by following a mutation strategy. Some commonly used conventional mutation strategies are:[3,4,22]

$$\text{DE/rand/1}: \ \vec{v}_i = \vec{x}_{r_1} + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{1}$$

$$\text{DE/rand/2}: \ \vec{v}_i = \vec{x}_{r_1} + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5}) \tag{2}$$

$$\text{DE/best/1}: \ \vec{v}_i = \vec{x}_b + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) \tag{3}$$

$$\text{DE/best/2}: \ \vec{v}_i = \vec{x}_b + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) + F_{xc}(\vec{x}_{r_3} - \vec{x}_{r_4}) \tag{4}$$

$$\text{DE/current-to-best/1}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) \tag{5}$$

$$\text{DE/current-to-best/2}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) + F_{xc}\ (\vec{x}_{r_3} - \vec{x}_{r_4}) \tag{6}$$

$$\text{DE/current-to-rand/1}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_{r_1} - \vec{x}_i) + F_{xc}\ (\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{7}$$

$$\text{DE/current-to-rand/2}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_{r_1} - \vec{x}_i) + F_{xc}\ (\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}\ (\vec{x}_{r_4} - \vec{x}_{r_5}) \tag{8}$$

$$\text{DE/rand-to-best/1}: \ \vec{v}_i = \vec{x}_{r_1} + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}\ (\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{9}$$

$$\text{DE/rand-to-best/2}: \ \vec{v}_i = \vec{x}_{r_1} + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5}) \tag{10}$$

where the integers r_1, r_2, r_3, r_4 and r_5 are chosen randomly from the interval $[1, NP]$ and are different from the current index i, F_{xc} is a real scaling factor that controls the amplification of the differential variation, $\vec{x}_b$ is the best solution among the NP population members at the generation G, and F_{cr} is a weight for the combination between the original target vector and the best parent vector or the random parent vector. The strategy DE/rand/1 proposed in the original paper of Storn and Price is the most widely used mutation strategy. It has stronger exploration capability but may converge slower than the strategies that use the best solution from the parent generation. The strategy DE/rand/2 uses two difference vectors to increase the diversity of the mutant. The strategies DE/best/1 and DE/best/2 take advantage of the best solution found in the parent population and have a faster convergence towards the optimal solution.[23] However, they may be stuck at a local minimum point during multimodal function optimization. The DE/current-to-best/1 and DE/current-to-best/2 strategies provide a compromise between exploitation of the best solution and exploration of the parameter space. The DE/current-to-rand/1 and DE/current-to-rand/2 mutation strategies are rotation-invariant strategies.[24] The DE/rand-to-best/ strategies are similar to the DE/current-to-best/ strategies, but larger diversity of the solutions after mutation is attained by using a randomly selected parent vector instead of the current target parent vector.

2.2. *Crossover*

A crossover operation between the new generated mutant vector $\vec{v}_i$ and the target vector $\vec{x}_i$ is used to further increase the diversity of the new candidate solution. This operation combines the two vectors into a new trial vector $\vec{U}_i, i = 1, 2, 3, \cdots, NP$, where the components of the trial vector are obtained from the components of $\vec{v}_i$ or $\vec{x}_i$ according to a crossover probability Cr. In the binomial crossover scheme for a D dimensional control parameter space, the new trial vector $\vec{U}_i$, $i = 1, 2, \cdots, NP$ is generated using the following rule:

$$\vec{U}_i = (u_{i1}, u_{i2}, \cdots, u_{iD}) \tag{11}$$

$$u_{ij} = \begin{cases} v_{ij}, \text{ if } \ \text{rand}_j \leq Cr \ \ \text{or } j = \text{mbr}_i \\ x_{ij}, \text{ otherwise} \end{cases} \tag{12}$$

where rand_j is a randomly chosen real number in the interval $[0, 1]$, and the index mbr_i is a randomly chosen integer in the range $[1, D]$. This ensures that the new trial vector contains at least one component from the new mutant vector.

2.3. *Selection*

The newly generated trial solution $\vec{U}_i$ is checked against the boundary of the control parameter space. If the solution lies outside the boundary, a new trial solution is generated by randomly sampling from solutions within the boundary.

The selection operation in DE is based on a one-to-one comparison. The new trial solution $\vec{U}_i$ is checked against the original target parent solution $\vec{x}_i$. If the objective value from the new trial solution produces is not worse than the original objective function value from $\vec{x}_i$, it will be put into the next generation $(G+1)$ population. Otherwise, the original parent $\vec{x}_i$ is kept in the next generation population.

The above procedure is repeated for all NP parents to generate the next generation population. Many generations are used to attain the final globally optimal solution.

3. Adaptive Unified Differential Evolution Algorithm for Global Optimization

Ten different mutation strategies have been proposed for the conventional standard differential evolution algorithm (Eqs. 1-10). The presence of multiple mutation strategies complicates the use of the differential evolution algorithm. Recently, we proposed a single mutation expression that can unify most conventional mutation strategies used by the differential evolution algorithm. This single unified mutation expression can be written as:

$$\begin{aligned}\vec{v}_i = \vec{x}_i + F_1(\vec{x}_b - \vec{x}_i) + F_2(\vec{x}_{r_1} - \vec{x}_i) \\ +F_3(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_4(\vec{x}_{r_4} - \vec{x}_{r_5})\end{aligned} \tag{13}$$

Here, the second term on the right-hand side of equation (13) denotes the contribution from the best solution found in the current generation, the third term denotes the rotationally invariant contribution from the random solution,[24] and the fourth and fifth terms are the same terms as those used in the original differential evolution algorithm to account for the contribution from the difference of parent solutions. Those last three terms divert the mutated solution away from the best solution and help to improve the algorithm's exploration of the decision parameter space. The four parameters F_1, F_2, F_3 and F_4 are the weights from each contribution. This unified expression represents a combination of exploitation (using the best found solution) and exploration (using randomly chosen solutions) when generating the new mutant solution.

From the above equation, one can see that for $F_1 = 0$, $F_2 = 1$, and $F_4 = 0$, this equation reduces to DE/rand/1; for $F_1 = 0$, $F_2 = 1$, and $F_3 = F_4$, it reduces to DE/rand/2; for $F_1 = 1$, $F_2 = 0$, and $F_4 = 0$, it reduces to DE/best/1; for $F_1 = 1$, $F_2 = 0$, and $F_3 = F_4$, it reduces to DE/best/2; for $F_2 = 0$ and $F_4 = 0$, it reduces to DE/current-to-best/1; for $F_2 = 0$ and $F_3 = F_4$, it reduces to DE/current-to-best/2; for $F_1 = 0$, and $F_4 = 0$, it reduces to DE/current-to-rand/1; for $F_1 = 0$, and $F_3 = F_4$, it reduces to DE/current-to-rand/2; for $F_2 = 1$, and $F_4 = 0$, it reduces

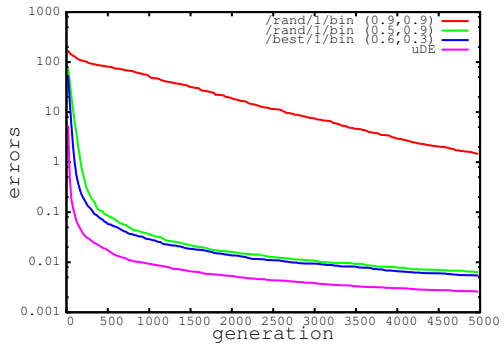

Fig. 1. Evolution of the average error in the noisy quartic function with 50 dimensions.

to DE/rand-to-best/1; for $F_2 = 1$, and $F_3 = F_4$, it reduces to DE/rand-to-best/2. Using the equation (13), the ten mutation strategies of the standard differential evolution algorithm can be included in a single expression. This new expression provides an opportunity to explore more broadly the space of mutation operators. Using a different set of parameters F_1, F_2, F_3, F_4, a new mutation strategy can be achieved. For example, Fig. 1 shows a plot of errors in the objective function value from a numerical test by using a new mutation strategy with $F_1 = F_2 = F_4 = 0.2, F_3 = 0.5$ and $Cr = 0.8$ from the unified differential evolution algorithm (pink). Also shown is the standard differential evolution algorithm DE/rand/1 with $F = 0.9, Cr = 0.9$ (red), DE/rand/1 with $F = 0.5, Cr = 0.9$ (green), and DE/best/1 with $F = 0.6, Cr = 0.3$ (blue). Here the test function is a 50 dimensional quartic function with noise given by:[29]

$$F_{\rm qrt}(\vec{x}) = \sum_{i=1}^{N} i x_i^4 + {\rm rand}[0,1); \quad -1.28 \le x_i \le 1.28 \tag{14}$$

It is seen that by expanding the space of mutation strategies (using the unified mutation strategy), it is possible to find a better solution than the conventional standard differential evolution algorithm in some applications. Moreover, by adaptively adjusting these parameters during the evolution, the multiple mutation strategies and their combinations can be used during different stages of optimization. Thus, the unified mutation expression has the virtue of mathematical simplicity and also provides the user with flexibility for broader exploration of different mutation strategies.

The unified mutation strategy provides a method to combine and to use different mutation strategies. However, choosing appropriate control parameters F_1, F_2, F_3, F_4 by trial-and-error approach can be challenging and time consuming for application users. Also, using a set of fixed control parameters does not necessarily lead to the best performance of the algorithm since different mutation strategies

might have superior performance at different generations during the process of evolutionary optimization. A self-adaptive method to select these control parameters will free the user from such a burden and also improve the performance of the algorithm.

Parameter tuning has been widely used in evolutionary optimization.[25–27] In general, these methods can be classified as deterministic parameter control, adaptive parameter control, and self-adaptive parameter control.[28] In deterministic parameter control, the parameters used in the algorithm evolve following a pre-determined rule (which can be time-dependent). In adaptive parameter control, the parameters are dynamically updated based on learning during the evolution. In self-adaptive parameter control, the parameters are encoded within each individual solution and evolve together with the solution during the process of optimization. In this study, we follow the self-adaptive method in reference[29] to allow the five control parameters (F_1, F_2, F_3, F_4 and Cr) to evolve dynamically in the unified differential evolution algorithm. This self-adaptive scheme is simple to implement and achieved good performance in a number of benchmark tests.

During the mutation stage, the self-adaptive method used in this study assumes that at generation G, each individual solution $\vec{x}_i^G$, $i = 1, 2, 3, \cdots, NP$ has a set of control parameters $F_{1,i}^G, F_{2,i}^G, F_{3,i}^G, F_{4,i}^G$ and Cr_i^G associated with it. Before generating a new mutant solution using the unified differential evolution expression (13), a new set of control parameters $F_{1,i}^{G+1}, F_{2,i}^{G+1}, F_{3,i}^{G+1}, F_{4,i}^{G+1}$ and Cr_i^{G+1} are calculated as:

$$F_{j,i}^{G+1} = \begin{cases} F_{jmin} + r_{j1}(F_{jmax} - F_{jmin}), & \text{if } r_{j2} < \tau_j \\ F_{j,i}^G, & \text{otherwise} \end{cases} \tag{15}$$

$$Cr_i^{G+1} = \begin{cases} Cr_{min} + r_3(Cr_{max} - Cr_{min}), & \text{if } r_4 < \tau_5 \\ Cr_i^G, & \text{otherwise} \end{cases} \tag{16}$$

where $r_{j1}, r_{j2}, j = 1, 2, 3, 4$, r_3, r_4 are uniform random values in the interval $[0, 1]$, F_{jmin} and F_{jmax} for $j = 1, 2, 3, 4$ are the minimum and the maximum allowed values of those control parameters, Cr_{min} and Cr_{max} are the minimum and the maximum cross-over probability, and $\tau_j, j = 1, 2, 3, 4, 5$ represents the probability to use a new value or to keep the old value for the j^{th} control parameter. The values of τ_j are normally kept small so that better control parameters associated with surviving solutions will be reused to generate the new trial solution. In this study, we set $\tau_j = 0.1$ following the reference.[29] We also did numerical tests with $\tau_j = 0.05, 0.15$, and 0.2 using the benchmark functions in the following section and did not see significant differences for most functions. The values of F_{jmin} and F_{jmax} are set to 0 and 1 respectively in this paper. We also set $Cr_{min} = 0$ and $Cr_{max} = 1$. The selection of these values is based on the consideration that the various conventional differential evolution mutation strategies of Eqs. (1-10) can be covered by the settings of those parameters, and in the literature, F_3 and F_4 are rarely greater than one. The new set of control parameters (Eqs. 15-16) are used to generate the mutant solution in Eq. 13. The initial values of these control

parameters are uniform random values between the minimum and the maximum values.

4. Variable Population with External Storage Parallel Differential Evolution for Multi-Objective Optimization

In many accelerator applications, one needs to optimize more than one objective function. The problem of multi-objective optimization can be stated in the general mathematical form as:

$$min \begin{cases} f_1(\vec{x}) \\ \cdots \\ f_n(\vec{x}) \end{cases} subject\ to\ g_i(\vec{x}) \leq 0, h_i(\vec{x}) = 0 \tag{17}$$

Here, $f_1, \cdots, f_n$ are n objective functions to be optimized, $\vec{x}$ is a vector of control parameters, and g_i and h_i are constraints to the optimization. The goal of multi-objective optimization is to find the Pareto front in the objective function solution space. The Pareto optimal front is a collection of all non-dominated solutions in the whole feasible solution space. Any other solution in the feasible solution space will be dominated by those solutions on the Pareto optimal front. In the multi-objective optimization, a solution A is said to dominate a solution B if all components of A are at least as good as those of B (with at least one component strictly better). Here, a component of A corresponds to one objective function value, i.e. $A_i = f_i(\vec{x})$. The solution A is non-dominated if it is not dominated by any solution within the group. An example of the Pareto front is shown as the green line within the feasible solution space in Fig. 2 with two objective functions.

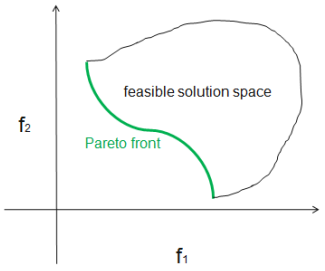

Fig. 2. Feasible solution space and the Pareto optimal front in a two-objective function optimization.

Recently, we developed a new parallel multi-objective differential evolution algorithm with varying population size in each generation and external storage to save all non-dominated solutions. The use of variable population is based on the observation that during the early stage of evolution, the number of nondominated solutions is small. There is no need to keep many dominated solutions in the parent population. As the search evolves, more and more nondominated solutions are

obtained. Those nondominated solutions are stored in an external storage so that they can be used to select for the new parent population. The advantage of using a variable population size with external storage is to make more use of the non-dominated solutions but less use of the dominated solutions to improve the speed of convergence. The new algorithm is summarized in the following steps:

- Step 0: Define the minimum parent size, $NPmin$ and the maximum size, $NPmax$ of the parent population. Define the maximum size of the external storage, $NPext$.
- Step 1: An initial $NPini$ population of parameter vectors are chosen randomly to cover the entire solution space.
- Step 2: Generate the offspring population using the above adaptive unified differential evolutionary algorithm.
- Step 3: Check the new population against the constraints.
- Step 4: Combine the new population with the existing parent population from the external storage. Non-dominated solutions ($Ndom$) are found from this group of solutions and $min(Ndom, Next)$ of solutions are put back to the external storage. Pruning is used if $Ndom > Next$. NP parent solutions are selected from this group of solutions for next generation production. If $NPmin \leq Ndom \leq NPmax$, $NP = Ndom$. Otherwise, $NP = NPmin$ if $Ndom < NPmin$ and $NP = NPmax$ if $Ndom > NPmax$. The elitism is emphasized through keeping the non-dominated solutions while the diversity is maintained by penalizing the over-crowded solutions through pruning.
- Step 5: If the stopping condition is met, stop. Otherwise, return to Step 2.

The above population based differential evolution optimization algorithm naturally leads to a multi-processor parallel implementation. Our method contains two levels of parallelization. First, the entire population is distributed among a number of groups of computer processors. Each group of processors contains a subset of the population. Different subsets of the population will evolve simultaneously. Second, for each population solution, the objective function values such as transverse emittances and bunch length are extracted from the outputs of parallel beam dynamics simulations using multiple processors within the group.

5. Benchmark Optimization Examples

The above adaptive unified differential evolution optimization algorithm is tested using the following analytical functions:[29,30]

$$F_1(\vec{x}) = \sum_{i=1}^{N} x_i^2; \quad -100 \leq x_i \leq 100 \tag{18}$$

$$F_2(\vec{x}) = \sum_{j=1}^{N} \left(\sum_{i=1}^{j} x_i \right)^2; \quad -100 \leq x_i \leq 100 \tag{19}$$

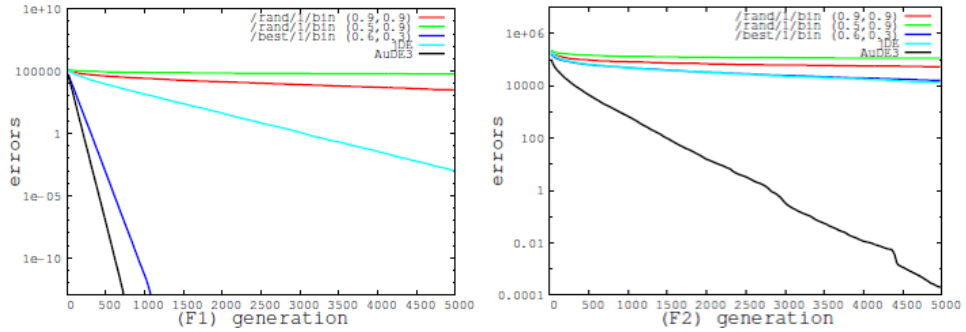

Fig. 3. Evolution of the average error in the sphere function (left) and the Schewefel's function (right) from the adaptive unified differential evolution method and some conventional differential evolution methods.

Here, the first sphere function F_1 is a continuous, unimodal and separable function. The second function F_2 is the Schewefel's problem 1.2 that is a non-separable unimodal function. Figure 3 shows the evolution of the error relative to the true global minimum objective function value of these test functions using the above adaptive unified differential evolution algorithms with dimension $N = 50$. At each generation, the objective function value has been averaged over 25 random seeds. It is seen that the adaptive unified differential algorithm performs quite well in these test examples in comparison to the other conventional standard differential evolution algorithms and quickly converges to the true minimum.

As a test of above parallel multi-objective differential evolution algorithm, we used the following two objective functions:[1]

$$\begin{aligned} f_1(\vec{x}) &= x_1 \\ f_2(\vec{x}) &= g(\vec{x})[1 - (x_1/g(\vec{x}))^2] \\ g(\vec{x}) &= 1 + 9(\sum_{i=2}^{n} x_i)/(n-1) \end{aligned} \tag{20}$$

The optimal Pareto front for these two objectives is:

$$\begin{aligned} f_2 &= 1 - f_1^2 \\ x_1 &\in [0, 1] \\ x_i &= 0, \ i = 2, \ldots, n \end{aligned} \tag{21}$$

The final optimal Pareto fronts from both the numerical solution using the variable population differential evolution algorithm and the analytical solution are shown in Fig 4. The multi-objective differential evolution algorithm converges to the true Pareto front solution with less than 5000 thousand function evaluations.

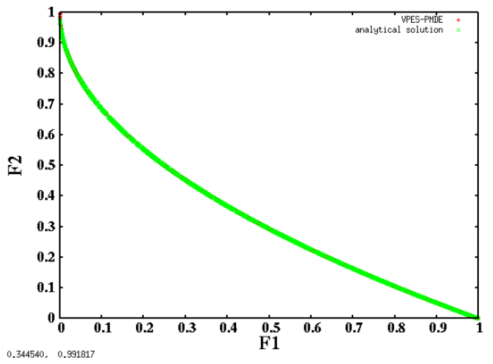

Fig. 4. The Pareto optimal front from the VPES solution and the analytical solution.

6. Application to Global Beam Dynamics Optimization

In this section, we apply the above parallel multi-objective differential evolution algorithm to global longitudinal beam dynamics optimization in a next generation x-ray FEL light source (LCLS-II) design with 20 pC charge. The LCLS-II is a high repetition rate (1 MHz) x-ray FEL that will deliver photons of energy between 200 eV and 5 keV.[31,32] It consists of a high repetition rate photo-injector to generate and accelerate the electron beam to about 100 MeV, a laser heater (LH) to suppress microbunching instability, a section of superconducting linac L1 to accelerate the beam to 250 MeV, a bunch compressor BC1, a second section of superconducting linac L2 to accelerate the beam to 1.6 GeV, a bunch compressor BC2, and a third section of superconducting linac L3 to accelerate the beam to 4 GeV, a long bypass transport line, and a magnetic kicker to spread the electron beam to a soft x-ray transport beam line and to a hard x-ray transport beam line. The superconducting linacs in all three sections are made of 1.3 GHz 9 cell superconducting cavities except the two cryomodules of 3.9 GHz third harmonic cavities right before the BC1 to linearize longitudinal phase space.

The objective functions used in the multi-objective optimization are outputs from the start-to-end beam dynamics simulations using the 3D parallel beam dynamics simulation framework IMPACT.[33–35] This framework includes a time-dependent 3D space-charge code module IMPACT-T for injector simulation and a position-dependent 3D space-charge code module for linac and beam transport system simulation. The simulation starts from the generation of photo-electrons at the photo-cathode following the initial laser pulse distribution and the given initial thermal emittance. The electron macroparticles out of the cathode will be subject to both the external fields from a DC/RF gun and a solenoid, and the space-charge/image charge fields from the Coulomb interaction of the particles among themselves. After exiting from the injector, the electron macroparticle will pass through a linear accelerator and beam transport system that includes laser heater, bunch compressors, accelerating RF cavities, harmonic linearizer, and magnetic fo-

cusing elements. Besides the 3D space-charge effects, the simulation also includes coherent synchrotron radiation (CSR) effects through a bending magnet, incoherent synchrotron radiation inside the bending magnet, RF cavity structure wakefield, and resistive wall wakefield.

The start-to-end beam dynamics simulation is integrated with the parallel multi-objective optimization program described above. Figure 5 shows a schematic diagram of the global optimization including both the injector control parameters and the linac control parameters in the start-to-end beam dynamics optimization. Here, the outputs from the start-to-end simulation are treated as objective functions in the parallel multi-objective optimizer. The parallel optimizer will call the IMPACT simulation by passing the injector control parameters and the linac control parameters into the objective function. There are total 22 control parameters. The 12 control parameters in the injector are laser transverse size, laser pulse flat-top length, VHF gun RF phase, buncher cavity amplitude and phase, two solenoid strengths, the 1st boosting cavity amplitude and phase, and the 2nd boosting cavity amplitude and phase and the last cavity phase. The 10 control parameters in the linac are the linac section one amplitude and phase, 3rd harmonic cavity amplitude and phase, bending angle in bunch compressor one, linac section two amplitude and phase, bending angle in bunch compressor two, and linac amplitude and phase. Instead of starting with direct global optimization in the entire control parameter space, we start the optimization with reduced control parameter space that contains only the injector control parameters. The two objective functions, final project transverse emittance and rms bunch length (directly related to peak current) at the exit of the injector are optimized subject to a number of constraints. These constraints are final electron beam energy and beam energy chirp. After a Pareto optimal front is attained for these two objective functions at the exit of the injector, these optimal injector control parameters are combined with randomly sampled control parameters in the linac to form a vector of total 22 control parameters. Using the optimal injector control parameters as a partial initial component in the global control parameter solution space significantly saves the computational time and speeds up the convergence of the final global solution. During the global beam dynamics optimization, one of the objective (transverse emittance) from the original injector optimization becomes a constraint to the new objective functions. Those solutions at the exit of the injector that can not satisfy this constraint for final start-to-end optimization will be automatically excluded during the global optimization. For the global longitudinal beam dynamics optimization, we would like to attain a higher peak current with flatter longitudinal phase space. The two objective functions are negative fraction of charge and rms energy spread inside a given longitudinal window. The larger fraction of charge inside the window, the higher peak current will be. The smaller rms energy spread inside the window, the flatter longitudinal phase space will be. A high peak current and flat longitudinal phase space will improve the x-ray FEL radiation power and reduce the radiation bandwidth. The outputs from the injector such as energy, emittance, and energy spread are used as constraints for the

global optimization. Besides the constraints at the exit of the injector, we also put constraints at the final linac output such as energy and peak current. Figure 6

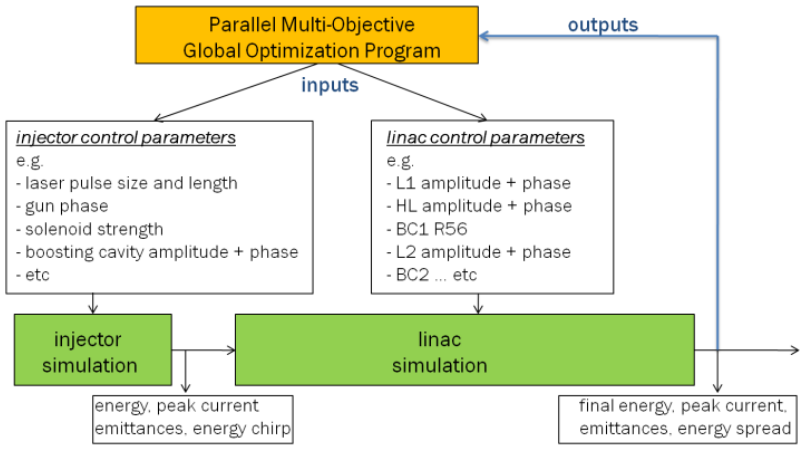

Fig. 5. A schematic diagram of the global beam dynamics optimization.

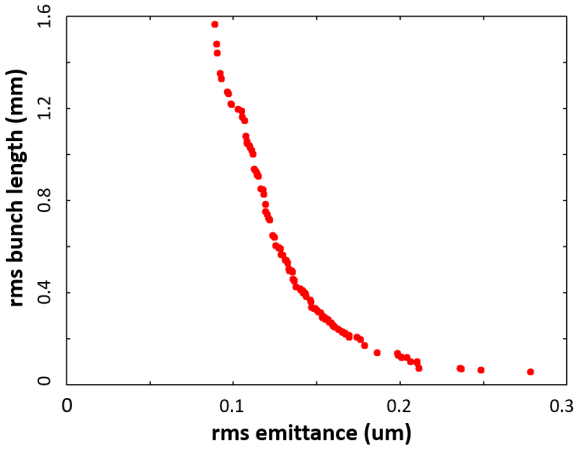

Fig. 6. The Pareto front from the injector beam dynamics optimization.

shows the Pareto front of two injector objectives (transverse rms emittance and longitudinal bunch length) from the injector optimization. Here, we have set a final peak current to be lower than 20 A, final rms emittance to be less than 1 um, final electron beam energy to be greater than 85 MeV, final rms energy spread to be less than 100 keV. It is seen that the rms emittance approaches to 0.1 um with the rms bunch length close to 1 mm in the Pareto front.

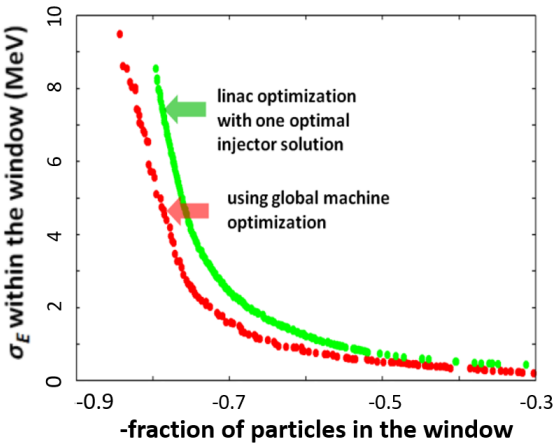

Fig. 7. The Pareto front from the global beam dynamics optimization and from the linac only optimization using one optimal injector solution.

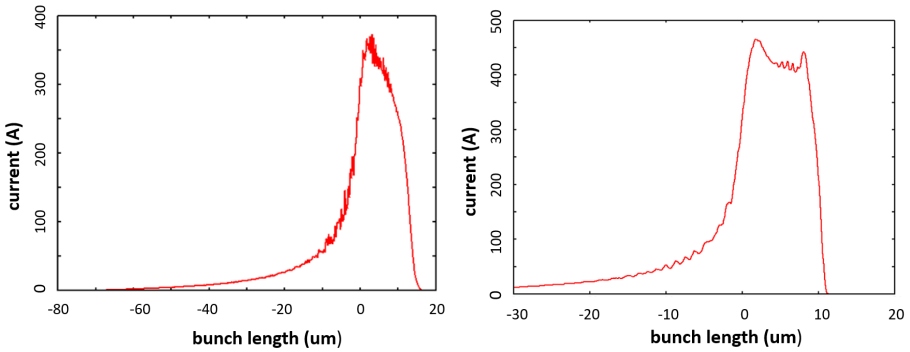

Fig. 8. The final electron beam current profile before (top) and after (bottom) global optimization.

Figure 7 shows the Pareto front of the final two objective functions at the end of the accelerator from the global beam dynamics optimization. These two objective functions are the negative fraction of charge and the rms energy spread inside a window between -7 and 9 um. In this plot, we also show the Pareto front from only the linac optimization using a solution from the injector optimization as an initial distribution. It is seen that the Pareto front from the global start-to-end accelerator optimization is significantly better than that from the linac only optimization. For the same amount of charge inside the window, the global solution has 40% less energy spread in some region. For the same level of the final rms energy spread, the global solution has 15% more electron charge. In this simulation, besides those constraints for the beam at the exit of the injector, we also put constraints on the final beam energy to be greater than 3.9 GeV, final rms energy spread to be less

than 2.5 MeV, fraction of charge inside the window between 0.3 and 0.9. Figure 8 shows the final electron beam current profile from a solution without and with global design optimization. It is seen that the final current profile is significantly improved through the global optimization. This results in more than 50% improvement in the final FEL radiation pulse energy.[36]

Acknowledgments

We would like to thank the LCLS-II physics design team for the LCLS-II application study. This work was supported by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and used computer resources at the National Energy Research Scientific Computing Center.

References

1. K. Deb et al., IEEE Trans. Evol. Comp, Vol. 6, p. 182, (2002).
2. https://ewh.ieee.org/conf/cec/.
3. R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, USA, Tech. Rep. TR-95-012, 1995.
4. R. Storn and K. Price, Journal of Global Optimization 1a1:341-359, (1997).
5. M. M. Ali and A. Torn, Computers and Operations Research, Elsevier, no. 31, p. 1703, 2004.
6. K. Price et al., Differential Evolution - A Practical Approach to Global Optimization, Springer, Berlin, 2005.
7. J. Qiang and C. Mitchell, Lawrence Berkeley National Laboratory Report lbnl-6853e, 2014.
8. J. Qiang, C. Mitchell, and A. Qiang, in Proc. CEC2016, Vancouver, p. 4061, 2016.
9. B. E. Carlsten, Nucl. Instrum. Methods Phys. Res., Sect. A285, 313 (1989).
10. L. Serafini and J. Rosenzweig, Phys. Rev. E 55, 7565 (1997).
11. I. V. Bazarov and C. K. Sinclair, Phys.Rev.ST Accel. Beams 8 (2005) 034202.
12. C. F. Papadopoulos et al., in Proc. of FEL2010, Malmo, Sweden, p. 479.
13. A. Hofler et al., Phys. Rev. ST Accel. Beams 16, 010101 (2013).
14. J. Qiang et al., in Proceedings of IPAC2013, p. 1031, Shanghai, 2013.
15. J. Qiang and C. Mitchell, in the Proc. FEL2014, THP020. 2014.
16. C. Mitchell et al., in Proc. IPAC2016, Busan, Korea, 2016, p. 1699.
17. H. Qian et al., in Proc. IPAC2016, Busan, Korea, 2016, p. 3979.
18. J. Arthur et al., LCLS conceptual design report, SLAC-R-593, 2002.
19. I. Zagorodnov and M. Dohlus, Phys. Rev. ST - Accel. Beams 14, 014403, 2011.
20. S. DiMitri et al., Physics Reports 539, p.1, 2014.
21. L. Wang et al., in Proc. FEL2014, Basel, Switzerland, 2014, p. 763.
22. F. Neri and V. Tirronen, Artif. Intell. Rev. 33, p. 61, 2010.
23. E. Mezura-Montes, J. Vel'azquez-Reyes, and C. A. Coello Coello, in Proc. Genet. Evol. Comput. Conf., 2006, pp. 485- 492.
24. K. V. Price, "An introduction to differential evolution," in New Ideas in Optimization, D. Corne, M. Dorigo, and V. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 79-108.
25. L. Davis, "Adapting operator probabilities in genetic algorithms," presented at the 3rd Int. Conf. Genetic Algorithms, Fairfax, VA, USA, 1989.

26. A. E. Eiben, R. Hinterding, and Z. Michalewicz, IEEE Trans. Evol. Comput., vol. 3, no. 2, pp. 124-141, 1999.
27. J. A. Vrugt, B. A. Robinson, and J. M. Hyman, IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 243-259, Apr. 2009.
28. A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing. Berlin, Germany: Springer, 2003.
29. J. Brest, S. Greiner, B. Bovskovic, M. Mernik, and V. Zumer, IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, pp. 646-657, 2006.
30. J. Zhang and A. C. Sanderson, IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945-958, 2009.
31. T. O. Raubenheimer, "Technical challenges of the LCLS-II CW X-ray FEL," in Proceedings of IPAC2015, Richmod, VA, USA.
32. P. Emma et al., in Proceedings of FEL'14, THP025.
33. J. Qiang, R. Ryne, S. Habib, V. Decyk, J. Comp. Phys. vol. 163, 434, (2000).
34. J. Qiang, S. Lidia, R. D. Ryne, and C. Limborg-Deprey, Phys. Rev. ST Accel. Beams **9**, 044204, 2006.
35. J. Qiang, R. D. Ryne, M. Venturini, A. A. Zholents, and I. V. Pogorelov, Phys. Rev. ST Accel. Beams, vol. 12, 100702, 2009.
36. G. Marcus, J. Qiang, "LCLS-II SCRF start-to-end simulations and global optimization as of September 2016," LCLS-II TN-1704, 2017.