Parallel general purpose multiobjective optimization framework with application to electron beam dynamics

N. Neveu[*] and L. Spentzouris

Illinois Institute of Technology, Chicago, Illinois 60616, USA

A. Adelmann,[†] Y. Ineichen, A. Kolano,[‡] and C. Metzger-Kraus

PSI, Villigen 5232, Switzerland

C. Bekas and A. Curioni

IBM Research, Zurich, Rüschlikon 8803, Switzerland

P. Arbenz

Department of Computer Science, ETH Zurich, Zurich 8092, Switzerland

Particle accelerators are invaluable tools for research in the basic and applied sciences, such as materials science, chemistry, the biosciences, particle physics, nuclear physics and medicine. The design, commissioning, and operation of accelerator facilities is a nontrivial task, due to the large number of control parameters and the complex interplay of several conflicting design goals. The Argonne Wakefield Accelerator facility has some unique challenges resulting from its purpose to carry out advanced accelerator R&D. Individual experiments often have challenging beam requirements, and the physical configuration of the beam lines is often changed to accommodate the variety of supported experiments. The need for rapid deployment of different operational settings further complicates the optimization work that must be done for multiple constraints and challenging operational regimes. One example of this is an independently staged two-beam acceleration experiment which requires the construction of an additional beam line (this is now in progress). The high charge drive beam, well into the space charge regime, must be threaded through small aperture (17.6 mm) decelerating structures. In addition, the bunch length must be sufficiently short to maximize power generation in the decelerator. We propose to tackle this problem by means of multiobjective optimization algorithms which also facilitate a parallel deployment. In order to compute solutions in a meaningful time frame, a fast and scalable software framework is required. In this paper, we present a general-purpose framework for simulation-based multiobjective optimization methods that allows the automatic investigation of optimal sets of machine parameters. Using evolutionary algorithms as the optimizer and OPAL as the forward solver, validation experiments and results of multiobjective optimization problems in the domain of beam dynamics are presented. Optimized solutions for the new high charge drive beam line found by the framework were used to finish the design of a two beam acceleration experiment. The selected solution along with the associated beam parameters is presented.

I. INTRODUCTION

Particle accelerators play a significant role in many aspects of science and technology. Fields such as material science, chemistry, the biosciences, particle physics, nuclear physics, and medicine depend on reliable and effective particle accelerators, both as research and practical tools. Achieving the required performance in the design, commissioning, and operation of accelerator facilities is a complex and versatile problem. Despite the success of on-line models in some facilities [1], and various model dependent and model independent tuning and optimization

techniques, empirical tuning by operators is a common method used at many research facilities. When the beam dynamics is nonlinear, as is the case with space charge, simple and fast models are applicable only in a very restricted manner. This further complicates any multiobjective optimization by complicating the model. In order to be able to reliably identify optimal configurations of accelerators, we solve large multiobjective design optimization problems to automate the investigation for an ideal set of tuning parameters. This approach has been used in the past with much success [2–12]. The difference here being the implementation and application to a problem at the Argonne Wakefield Accelerator Facility (AWA).

A hallmark of the AWA facility is the flexibility to swap physical components in the beam lines, which enables different, often novel, accelerator research experiments to take place. Not only do the physical machine components change, the beam characteristics also vary considerably to meet different needs. The facility operates at both low and high charge (up to 100 nC), and at high charge strong nonlinearities require a full 3D space charge approach in simulations. Finding optics solutions in this regime, especially when there are additional constraints such as the small aperture two-beam accelerating structures, is challenging even without the quick turnaround of the beam line configurations. Therefore, it has been an important research objective to develop a precise, e.g., 3D model embedded into a multiobjective optimization framework that may be used as a flexible platform for optimization of changing machine configurations operated at different charge levels. While other codes, such as GPT [13] and ELEGANT [14], also have integrated genetic optimization algorithms; the OPAL [15] framework differentiates itself by being open source (i.e., free to use), massively parallel, and fully 3D.

A modular multiobjective software framework was developed (see Fig. 1) where the core functionality of the optimizer is decoupled from the "beam dynamics" but fully integrated in the OPAL framework. To that end, we use a master/slave mechanism where a master process governs a set of slave processes given some computational tasks (beam dynamics simulation) to complete. This separation allows easy interchange of optimization algorithms, forward solvers and optimization problems. A "pilot" coordinates all efforts between the optimization algorithm and the beam dynamics task. Details of the code implementation, e.g. framework components, optimizer, and forward solver implementation, and parallelization, can be found in the Supplemental Material at [16]. In combination, this forms a robust and general framework for massively parallel multiobjective optimization. Currently the framework offers one concrete optimization algorithm, an evolutionary algorithm employing a NSGA-II selector [17,18]. Normally, simulation based approaches are plagued by the trade off between level of detail and time to solution.
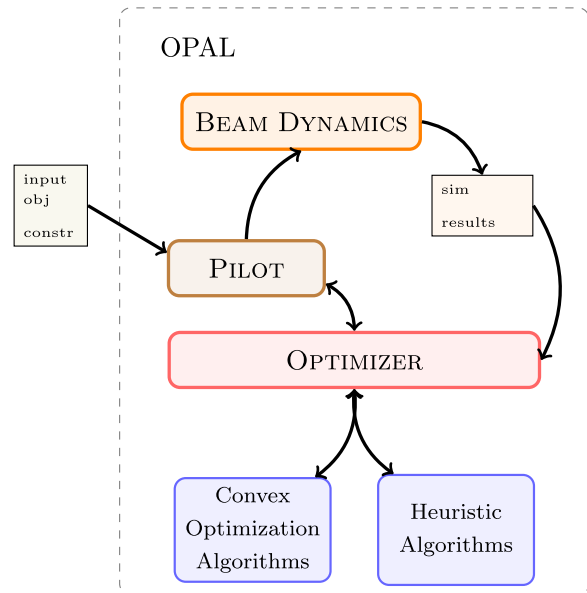

FIG. 1. Multiobjective framework: the pilot (master) solves the optimization problem specified in the input file by coordinating optimizer algorithms and workers running a forward solver.

This problem is addressed later in Sec. IV B 1 by using forward solvers with different time and resolution complexity.

The framework used here, incorporates the following three contributions: (1) Implementation of a scalable optimization algorithm capable of approximating Pareto fronts in high dimensional spaces, (2) design and implementation of a modular framework that is simple to use and deploy on large scale computational resources, and (3) demonstration of the usefulness of the proposed framework on a real world application in the domain of particle accelerators. This is done with the optimization problem set as the high charge photoinjector at the AWA.

The next section introduces the notation of multiobjective optimization theory and describes the first implemented optimizer. A validation and a proof of concept application in the beam dynamics problems mentioned above is discussed in Sec. IV. The implementation of the framework and forward-solver is discussed in Supplemental Material at [16].

II. MULTIOBJECTIVE OPTIMIZATION

Optimization problems deal with finding one or more feasible solutions corresponding to extreme values of objectives. If more than one objective is present in the optimization problem this is called a multiobjective optimization problem (MOOP). A MOOP is defined as

$$\min \quad f_m(\mathbf{x}), \qquad m = 1, \dots, M, \qquad (1)$$

$$\text{s.t.} \quad g_j(\mathbf{x}) \geq 0, \qquad j = 0, \dots, J, \qquad (2)$$

$$x_i^L \le \mathbf{x} = x_i \le x_i^U, \qquad i = 0, \ldots, n, \qquad (3)$$

where $\mathbf{f}$ denotes the objectives (1), $\mathbf{g}$ the constraints (2), and $\mathbf{x}$ the design variables (3). Often, conflicting objectives are encountered, and this complicates the concept of optimality. *Pareto optimality* is often used in such situations. The set of Pareto optimal points forms the Pareto front or surface. All points on this surface are considered to be Pareto optimal. Sampling Pareto fronts is far from trivial. A number of approaches have been proposed, e.g., evolutionary algorithms [19], simulated annealing [20], swarm methods [21], and many more [22–25]. In the next section, we briefly introduce the theory of evolutionary algorithms used in the present work.

A. Evolutionary algorithms

Evolutionary algorithms are loosely based on nature's evolutionary principles to guide a population of individuals towards an improved solution by honoring the "survival of the fittest" principle. This "simulated" evolutionary process preserves entropy (or diversity in biological terms) by applying genetic operators, such as mutation and crossover, to remix the fittest individuals in a population. Maintaining diversity is a crucial feature for the success of all evolutionary algorithms.

In general, a generic evolutionary algorithm consists of the following components: (i) *Genes*: traits defining an individual, (ii) *Fitness*: a mapping from genes to a fitness value for each individual, (iii) *Selector*: selecting the k fittest individuals of a population based on some sort of ordering, (iv) *Variator*: recombination (mutations and crossover) operators for offspring generation.

Applied to multiobjective optimization problems, genes correspond to design variables. The fitness of an individual is loosely related to the value of the objective functions for the corresponding genes. Figure 2 schematically depicts the connection of the components introduced above. The process starts with an initially random population of individuals, each individual with a unique set of genes and corresponding fitness, representing one location in the search space. In the next step, the population is processed by the selector determining the k fittest individuals according to their fitness values. While the k fittest individuals are passed to the variator, the remaining $n - k$ individuals are eliminated from the population. The VARIATOR mates and recombines the k fittest individuals to generate new offspring. After evaluating the fitness of all the freshly born individuals a *generation* cycle has completed and the process can start anew.

Since there already exist plenty of implementations of evolutionary algorithms, it was decided to incorporate the PISA library [18] into our framework. One of the advantages of PISA is that it separates variator from selector, rendering the library expandable and configurable. Implementing a variator was enough to use PISA in our
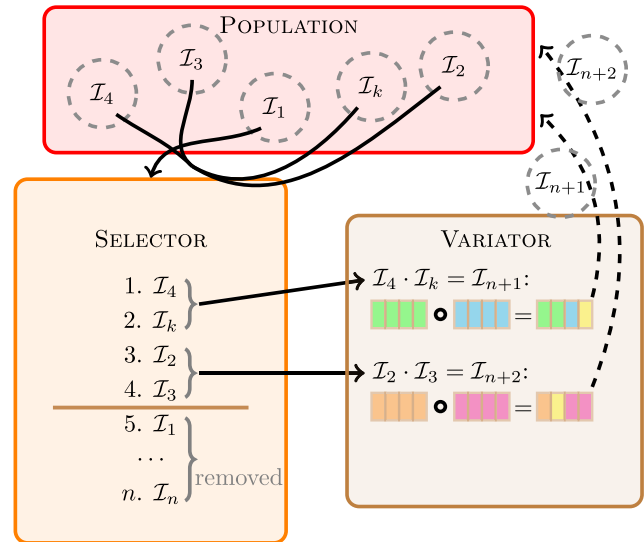

FIG. 2. Schematic view of interplay between selector and variator. The selector ranks all individuals in the population according to fitness and subsequently the variator uses the fittest individuals to produce new offspring. Finally, the new children are reintroduced in the population.

framework and retain access to all available PISA selectors. As shown in Fig. 2, the selector is in charge of ordering a set of d-dimensional vectors and selecting the k fittest individuals currently in the population. The performance of a selector depends on the number of objectives and the surface of the search space. So far, the NSGA-II selector [17] has been used and exhibits satisfactory convergence performance.

The task of the variator is to generate offspring and ensure diversity in the population. The variator can start generating offspring once the fitness of every individual of the population has been evaluated. This explicit synchronization point defines an obvious bottleneck for parallel implementations of evolutionary algorithms. In the worst case, some message passing interface processes are taking a long time to compute the fitness of the last individual in the pool of individuals to evaluate. During this time all other resources are idle and wait for the result of this one individual in order to continue to generate and evaluate offspring. To counteract this effect, the selector is already called when two individuals have finished evaluating their fitness, lifting the boundaries between generations and evaluating the performance of individuals. New offspring will be generated and message passing interface processes can immediately return to work on the next fitness evaluation. Calling the selector more frequently (already after two offspring individuals have been evaluated) results in better populations since bad solutions are rejected earlier. On the other hand, calling the selector more often is computationally more expensive. Note this capability is also present in Cornell's [2] and GPT's [13] optimization system.

The variator implementation uses the master/slave architecture, presented in the next section, to run as many function evaluations as possible in parallel. Additionally, various crossover and mutation policies are available for tuning the algorithm to the optimization problem.

III. THE FRAMEWORK

Simulation based multiobjective optimization problems are omnipresent in research and industry. These simulation and optimization problems are in general very big and computationally demanding. This motivated us to design a massively parallel general purpose framework. The key traits of such a design can be summarized as: (i) support any multiobjective optimization method, (ii) support any function evaluator: simulation code or measurements, (iii) offer a general description/specification of objectives, constraints, and design variables, (iv) run efficiently in parallel on current large-scale high-end clusters and supercomputers.

A. Related work

Several similar frameworks, e.g., [26–29], have been proposed. Commonly these frameworks are tightly coupled to an optimization algorithm, e.g., only providing evolutionary algorithms as optimizers. Users can specify optimization problems, but cannot change the optimization algorithm. Our framework follows a more general approach, providing a user-friendly way to introduce new or choose from existing built-in multiobjective optimization algorithms. Tailoring the optimization algorithm to the optimization problem at hand is an important feature due to the many different characteristics of optimization problems that should be handled by such a general framework. As an example, it is shown how PISA [18], an existing evolutionary algorithm library, was integrated with ease. Similarly, other multiobjective algorithms could be incorporated and used to solve optimization problems.

The framework presented in [29] resembles our implementation the most, aside from their tight coupling with an evolutionary algorithm optimization strategy. The authors propose a plug-in based framework employing an island parallelization model, where multiple populations are evaluated concurrently and independently up to a point where some number of individuals of the population are exchanged. This is especially useful to prevent the search algorithm getting stuck in a local minimum. A set of default plug-ins for genetic operators, selectors and other components of the algorithms are provided by their framework. User-based plug-ins can be incorporated into the framework by implementing a simple set of functions (See Supplemental Material [16]).

Additionally, as with simulation based multiobjective optimization, we can exploit the fact that both the optimizer and simulation part of the process use a certain amount of
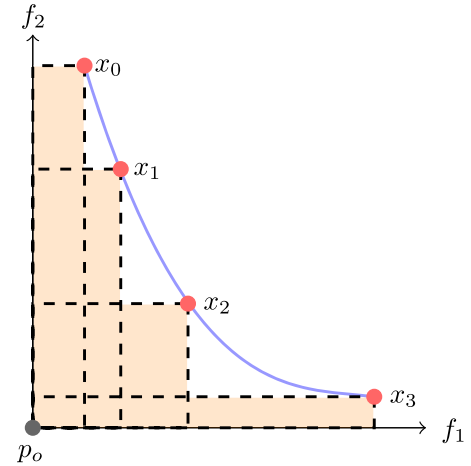

FIG. 3. The hypervolume for a two-objective optimization problem corresponds to the shaded area formed by the dashed rectangles spanned by all points on the Pareto front and an arbitrary selected origin p_o.

resources. The ratio of work between optimizer and simulation costs can be reflected in the ratio of number of processors assigned to each task. This not only provides users with great flexibility in using any simulation or optimizer, but renders influencing the role assignment easy as well.

IV. EXPERIMENTS

In this section numerical results of the validation benchmark and optimization of a photoinjector operated in the space charge dominated regime is presented.

A. Optimizer validation

To ensure that the optimizer works correctly, the benchmark problem presented in [30],
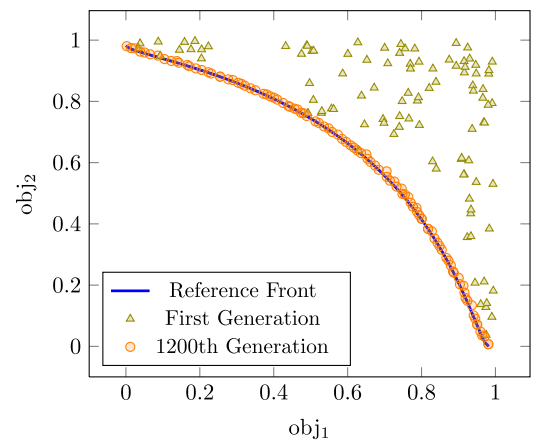

FIG. 4. Variator benchmark after 1100 function evaluations using binary crossover and independent gene mutations (each gene mutates with probability $p = \frac{1}{2}$) on a population of 100 individuals.

TABLE I. Convergence of benchmark problem with errors relative to hypervolume of sampled reference solution.

Total function evaluations	Hypervolume	Relative error
100	0.859 753	3.076×10^{-1}
200	0.784 943	1.938×10^{-1}
500	0.685 183	4.210×10^{-2}
900	0.661 898	6.689×10^{-3}
1100	0.657 615	1.749×10^{-4}

$$\min \left[1 - \exp \left\{ -1 \left[\left(x_1 - \frac{1}{\sqrt{3}} \right)^2 + \left(x_2 - \frac{1}{\sqrt{3}} \right)^2 \right. \right. \right.$$
$$\left. \left. + \left(x_3 - \frac{1}{\sqrt{3}} \right)^2 \right] \right\},$$
$$1 - \exp \left\{ -1 \left[\left(x_1 + \frac{1}{\sqrt{3}} \right)^2 + \left(x_2 + \frac{1}{\sqrt{3}} \right)^2 \right. \right.$$
$$\left. \left. \left. + \left(x_3 + \frac{1}{\sqrt{3}} \right)^2 \right] \right\} \right]^T$$
$$\text{s.t.} \quad -1 \leq x_i \leq 1, \qquad i = 1, 2, 3. \tag{4}$$

was solved with the framework. To that end, we use a metric for comparing the quality of a Pareto front. Given a point in the Pareto set, we compute the m dimensional volume (for m objectives) of the dominated space, relative to a chosen origin. This is visualized for 2 objectives in Fig. 3. For further information and details of the implementation see [31]. Figure 4, and the corresponding hypervolume values in Table I, show expected convergence. The reference Pareto front is well approximated. It

took a total of 1100 function evaluations to perform this computation. The hypervolume of the reference solution (0.6575) for our benchmark was computed by sampling the solution provided in [30]. Table I shows satisfactory convergence to the sampled reference Pareto front after 1000 (plus the additional 100 evaluations for the initial population) function evaluations.

B. AWA photoinjector optimization

Next the optimization framework is applied to the high charge beam line at the Argonne Wakefield Accelerator (AWA) facility. The goal of this optimization is to produce beams of electrons that meet design specifications; this includes number of particles (charge), energy, and particle distribution (characterized by beam sizes and energy spread). As shown in Fig. 5, the installed portion of the beam line consists of an rf photocathode gun, two solenoids, and six linear accelerating cavities followed by four quadrupoles and a stripline kicker. The charge of interest, 40 nC, is needed for two beam acceleration (TBA) experiments performed at the AWA [32,33], which motivates this work. Prior experimental results were limited by beam size when the beam passed through small aperture wakefield structures located downstream. In an attempt to maximize charge transmission in upcoming experiments, magnet strengths of the solenoids and quadrupoles leading into the TBA section of the beam line were optimized, shown in Fig. 6. The simulation model includes all elements from the gun to the septum. The optimization location is chosen as the entrance to the first quadrupole on the dog leg (s_3), see Fig. 6. Minimizing beam sizes here will enable capture and
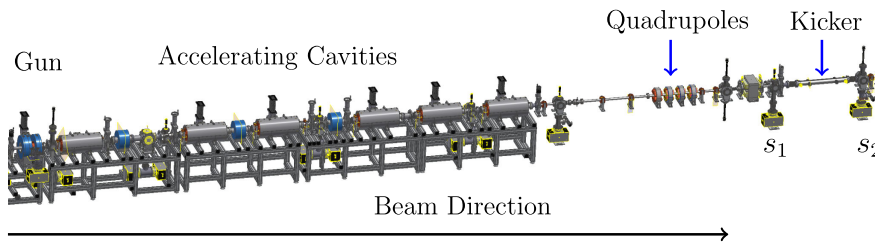

FIG. 5. Side view of the high charge linac at the AWA. All hardware in this drawing is currently installed. Note locations s_1 and s_2, before and after the kicker.
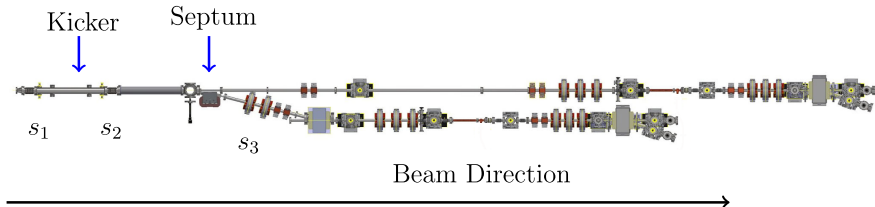

FIG. 6. Continuation of the high charge beam line layout at the AWA, top view. This is the proposed two beam acceleration section. Only the kicker in this drawing is installed. Note s_3, the entrance to the fifth quadrupole on the beam line. This is the optimization location.

TABLE II. Checkmarks (✓) indicate desired beam parameters are resolved at that time step. An (✗) indicates the time step is too large, and results are nonphysical.

Time Step, $dT(s)$	Linac	Drift	Quadrupoles
5×10^{-10}	✗	✗	✗
1×10^{-10}	✗	✗	✗
5×10^{-11}	✗	✗	✗
1×10^{-11}	✓	✓	✗
5×10^{-12}	✓	✓	✗
1×10^{-12}	✓	✓	✓

further focusing before space charge effects dominate the beam. This will also enable cleaner transport through downstream elements.

In addition to addressing the challenge of producing an optimized beam, this model was chosen to demonstrate the ability of the framework to tackle large problems. Six design variables and objectives were used, along with three constraints. The objectives include transverse and longitudinal beam sizes, transverse momentum, and longitudinal energy spread. The design variables include the two gun solenoids and the first four quadrupoles strengths. This problem encompasses high dimensionality and nonlinear effects such as space charge. Using this model it was possible to find good solutions that meet the operational goals for the new beam line. One of these solutions is presented in the next section.

1. Time step scan

Before running a full scale optimization of the problem described in Sec. IV B, a study on time step and number of particles in the simulation model was done to reduce the time of the simulation while maintaining the physics of interest. The grid size $16 \times 16 \times 32$ was chosen, and parallelized in the x and y directions. After comparing several options (1000, 10 000, 20 000, 50 000, 100 000) with a small time step, the number of particles was fixed at 10 000. Next several time steps were explored, see Table II. The largest steps were too big to resolve the beam parameters accurately. See low fidelity plot in Fig. 7 for $dT = 5 \times 10^{-11}$ results.

In the drifts and linac tanks, $dT = 1 \times 10^{-11}$ was sufficient. However, it was not acceptable near the quadrupoles. For all models, the longitudinal parameters (rms$_s$ and energy) are calculated correctly, but discrepancies are seen in the transverse (rms$_x$ and ϵ_x) for low fidelity results. This discrepancy is what led to the decision to adjust the time steps with respect to beam line elements. In the linac and drift sections $dT = 1 \times 10^{-11}$ was used. Near sensitive elements such as the quadrupoles, kicker, and septum, a time step of $dT = 1 \times 10^{-12}$ was used. The resulting simulations are low fidelity in most places, but closely approximate the mid fidelity simulations for metrics of interest, as shown in Fig. 7. Midfidelity simulations used steps of $dT = 1 \times 10^{-12}$ everywhere. The average run time of each simulation with the adjusted time steps was 1.6 minutes. In comparison, the midfidelity simulation
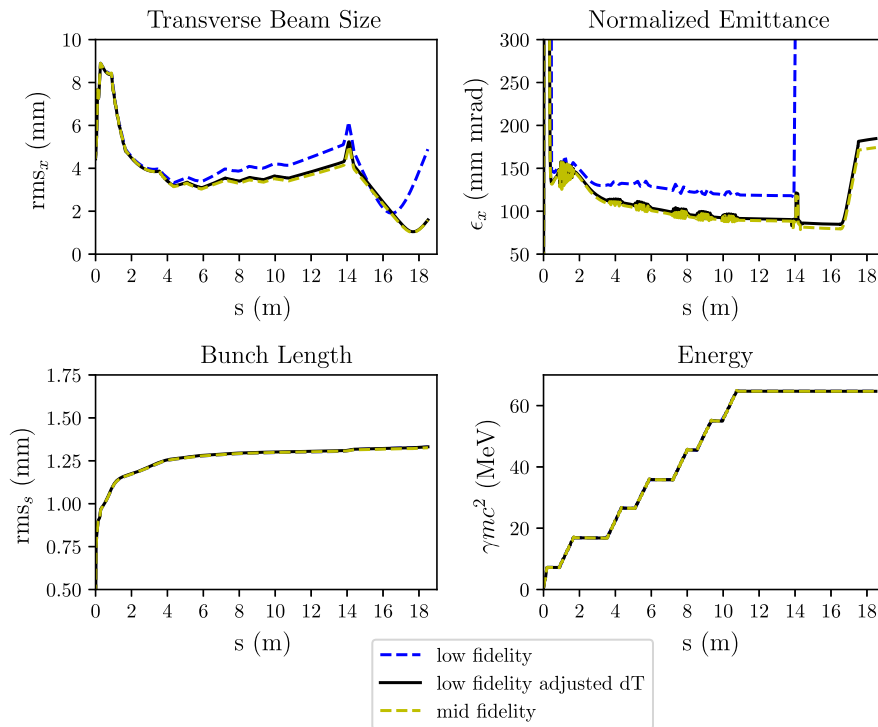

FIG. 7. Comparison of different fidelity models (dT stands for time step).

TABLE III. Input parameters for initial twenty-four hour AWA optimization experiments. The gene mutation probability was equal to the mutation probability (not shown) in all four experiments. The maximum number of individuals per generation was 80.

Experiment	Gene mutation probability	Recombination probability	Number of completed generations
ex-1	0.1	0.9	96
ex-2	0.3	0.7	81
ex-3	0.8	0.2	53
ex-4	0.01	0.09	95

ran for 18 minutes. Note, a smaller time step, 1×10^{-13}, is always used in the gun where the beam has low energy and is changing rapidly.

2. Hyper parameter scan

While the optimization problem and goals were well defined (Sec. IV B), it was not clear what the best hyper-parameters for the genetic algorithm would be. These parameters include gene mutation probability, mutation probability, recombination probability, number of individuals, and number of generations to complete. Given the beam line in Fig. 5, four small optimization experiments were done with various hyperparameters. Similar to the time step scan, the goal of this exercise was to determine which set of optimization parameters strongly influence the results, and whether there was a time to solution difference. From here on, we will reference each experiment as ex-1, ex-2, ex-3, and ex-4 as shown in Table III.

The maximum number of individuals per generation was fixed at 80. This number was chosen based on the node architecture, and to prevent a prohibitive computational

cost. Each experiment was allowed to run for twenty-four hours, with a maximum generation limit of 100. We reduced the six objectives to four, and shortened the simulation time by moving the objectives further upstream to s_1 and s_2, the locations before and after the kicker, see Fig. 6. The objectives include: $\varepsilon_x(s = s_1)$, $\varepsilon_x(s = s_2)$, $\mathrm{rms}_s(s = s_1)$, and $\mathrm{rms}_s(s = s_2)$. An example OPAL input file, used to perform the optimization in ex-1, can be found in Supplemental Material at [16]. All other optimization runs in the paper follow this template.

After collection of the data for all four experiments, several metrics were compared, including number of generations completed in twenty-four hours and Pareto fronts at s_1 and s_2, see Fig. 5. From Table III, we clearly see ex-3 is significantly slower, as it evaluated only 53 generations compared to the experiment with the maximum number, ex-1 at 96 generations. Perhaps this trade off would be acceptable if the Pareto front was significantly improved, but from Fig. 8, this is not the case. Similar arguments can be made for ex-2, which evaluated about 15 fewer generations. The Pareto fronts at s_2, are nearly identical. It is expected this trend would continue given more time. When looking at the Pareto front at s_1, only ex-4 has a slightly larger range compared to the others. With ex-2 and ex-3 eliminated due to evaluation time, and a slightly better Pareto front at s_1 for ex-4, the hyperparameters in ex-4 were chosen as the default values for subsequent runs.

3. TBA optimization problem

With computational and hyperparameters set, the optimization problem of interest is explored. The objectives (beam sizes and energy spread) are calculated at $s_3 = 19.4\,\mathrm{m}$, located downstream of the septum, see Fig. 6. Given the longitudinal location of s_3 (unless otherwise noted), we define the objectives and input parameters as:
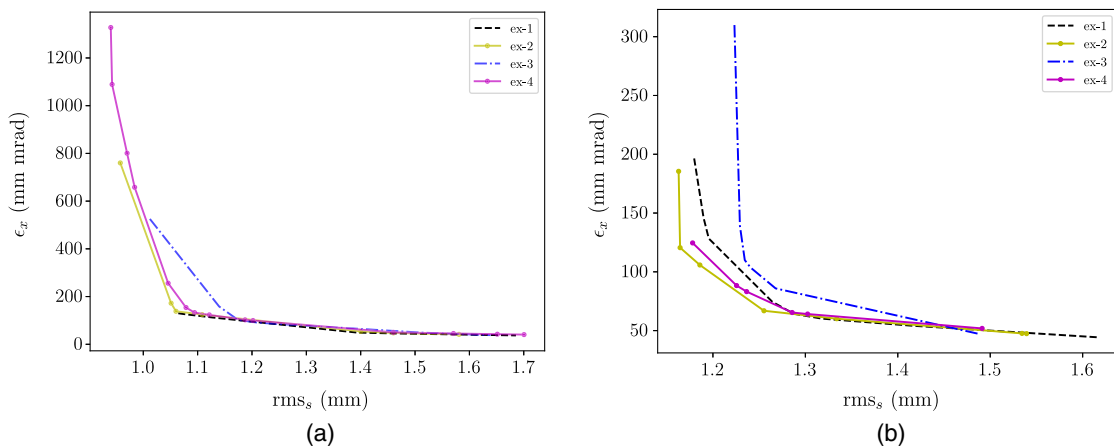

FIG. 8. Comparison of Pareto fronts for initial optimization experiments, ex-1 through ex-4. (a) Pareto fronts for ex-1 through ex-4 at s_1, (b) Pareto fronts for ex-1 through ex-4 at s_2.

$$\text{min} \quad \text{rms}_x, \qquad \text{rms}_y \qquad\qquad (5)$$

$$\text{rms}_{px}, \qquad \text{rms}_{py}, \qquad\qquad (6)$$

$$\text{rms}_s, \qquad dE \qquad\qquad (7)$$

$$\text{constraints} \quad \text{rms}_x < 0.1(\text{m})|_{s=s_1} \qquad (8)$$

$$\text{rms}_y < 0.1(m)|_{s=s_1} \qquad\qquad (9)$$

$$|\text{rms}_y - \text{rms}_x| < 0.005(\text{m})|_{s=s_1} \qquad (10)$$

$$\text{subject to} \quad q = 40(\text{nC}) \qquad\qquad (11)$$

$$\text{Volt}_{\text{Gun}} = 64(\text{MV/m}) \qquad\qquad (12)$$

$$\text{Volt}_{\text{Linac}} = 24 \quad \text{or} \quad 25(\text{MV/m}) \qquad (13)$$

$$R_x = R_y = 9(\text{mm}) \qquad\qquad (14)$$

$$\phi_{\text{gun}} = -20° \qquad\qquad (15)$$

$$\phi_{\text{linac}} = -20°. \qquad\qquad (16)$$

The first four objectives, parameters (5) to (6), minimize the transverse ($\text{rms}_{x,y}$) beam size and transverse momentum ($\text{rms}_{px,py}$) at the location of interest in the beam line (s_3, see Fig. 6). Minimizing the beam size at this location is essential to preventing loss of particles by scraping; which ensures better transmission through the wakefield structures downstream. These structures are called power extractor and transfer structure (PETS), and will be located after the septum and quadrupoles in Fig. 6. This device is putting the tightest constraints on the beam parameters. The aperture diameter is 17.6 mm and the bunch length should be below 2 mm to facilitate large power extraction [34]. Less divergence in the beam (lower transverse momentum spread) reduces growth of transverse beam size after the focal point (location of min beam size). This reduces halo by ensuring the beam is not over focused through a hard waist. The momentum spread is also critical to preventing large growth during transport. All of these factors help with transmission downstream.

The next two objectives in parameter (7) minimize the longitudinal beam size (rms_s), and energy spread (dE) at location s_3. This helps reduce the transverse beam size growth in bending elements. A small bunch length (rms_s) is also critical to the goals of TBA experiments. The power generated in the wakefield structures designed for TBA is related to the bunch length [33,34]. Equations (8) to (10) define three constraints used to guide the algorithm. However, it is important to not overconstrain the problem, which would prevent the algorithm from converging. The difference constraint, Eq. (10), is used to favor nearly

round beams. This prevents one dimension from becoming disproportionately large compared to the other. At the AWA, there is some room in the beam pipe to allow the y dimension to grow, but round beams are preferred. Equations (11) to (16) define the charge, gun voltage, linac voltages, laser radius, gun phase, and linac cavity phases (in that order). These are parameters in the simulation that must be defined, but do not vary during the optimization. For setup of the AWA design variables, objectives, and constraints in the OPAL input file, refer to the repository above.

Design variables include the currents in two gun solenoids (IBF and IM), and four quadrupole strengths (KQ1-KQ4). The objectives include beam size (transverse and longitudinal), transverse momentum, and energy spread as defined in Eqs. (5) to (7). The location at the entrance of the kicker is $s_1 = 16.45$ (m), and the objectives are optimized at location $s_3 = 19.4$ (m). This is the entrance to the fifth quad in the beam line. This location is where the beam should be captured and focused through subsequent elements.

4. AWA optimization results

All simulations for this experiment were carried out on Bebop, a high performance computing (HPC) cluster provided by the Laboratory Computing Resource Center (LCRC) at Argonne National Laboratory (ANL). Intel Knights Landing (KNL) processors at 1.3 GHz with 128 GB of memory and 64 cores per node were used for all runs. There are 352 compute nodes available on Bebop, with a total of 22,528 cores. All jobs were run and compared on 8 cores each, which allowed 8 jobs per node on the KNLs. This in combination with the number of nodes available allows for very large optimization jobs, like the AWA case. Typical runs for this paper used 41 nodes, which corresponds to 2624 KNL cores and a generation size of 328 individuals.

With the time steps and hyperparameters set by the work in Sec. IV B 1, the optimization problem described in Sec. IV B was run for 200 generations. The initial number of individuals was fixed at 656, and the minimum number individuals in later generations was fixed at 328. These numbers were in part based on the architecture of the KNL's. Since each simulation takes 8 cores, and there are 64 cores per KNL node, a large population size that would fit evenly on these resources was chosen. Again, the location of optimization is $s_3 = 19.4$ (m).

As expected, the x dimension is impacted by the bending elements, and unable to reach the small beam sizes seen in the y dimension. This suggests objectives in the x dimension will drive design variable choices used during operations. However, it is still necessary to include the y dimension in the optimization. Early optimization tests showed the y dimension can easily grow out of control if it is not included in the objectives. Those results are not
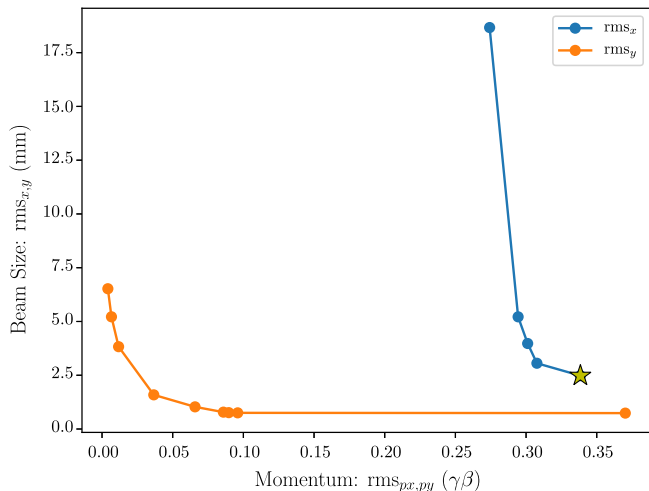
FIG. 9. Pareto front comparing transverse beam sizes ($\text{rms}_{x,y}$) and transverse momentum ($\text{rms}_{px,py}$). The yellow star indicates the point plotted in Fig. 10.

TABLE IV. Input parameters for the optimized solution shown in Figs. 9 and 10.

Design Variable	Unit	Value
Buck focusing solenoid	Amps	478
Matching solenoid	Amps	197
Quadrupole 1	T-m	-0.8
Quadrupole 2	T-m	0.9
Quadrupole 3	T-m	0.8
Quadrupole 4	T-m	-1.0
Bunch length	mm	1.5

shown here due to the unfeasible nature of the solutions (i.e., rms_y larger than the beam pipe). In the case of bunch length, there are not many options to choose from, as the phase was not varied. With these observations in mind, several beam parameters corresponding to options on the Pareto Front in Fig. 9 were plotted and compared. A select result is shown in Fig. 10, with corresponding parameters shown in Table IV. The maximum beam sizes are well below the beam pipe aperture limits, also shown in Fig. 10. The solution is nearly round, which will increase chances of keeping the beam nearly round as it travels to the last triplet in Fig. 6. Overall this solution is satisfactory, and meets all requirements for the new TBA beam line at the AWA.
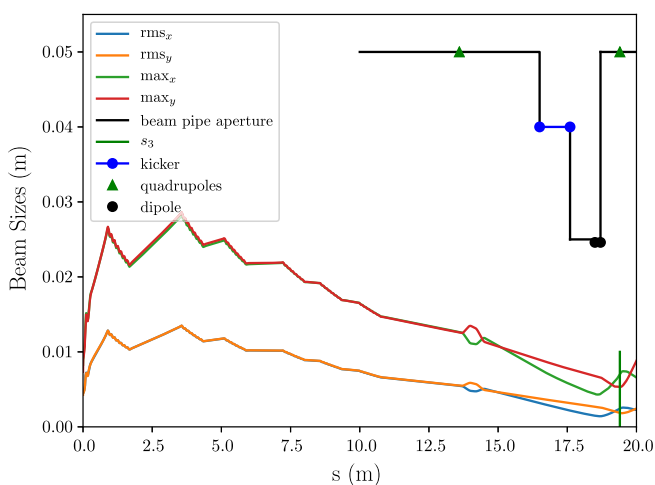
The TBA line under construction includes a kicker and septum combination to route one bunch train to its own PETS, while the other bunch train is undeflected and goes to a different PETS. The optimization results shown here start at the gun and go through to the first quadrupole after the septum [$s_3 = 19.4$ (m)]. These results were used as the basis for the full solution up to the PETS [35]. After adjustments to include 3D rf field maps and CSR in the dipoles [35], the maximum beam sizes at the PETS location were below 10 mm with a bunch length of $\text{rms}z = 1.6$ mm. Although the simulation results at the PETS location cannot yet be verified, since the beam line is not yet completely installed, some checks have been possible for the partially completed beam line. For example, beam images were taken on the YAG screen located downstream of the kicker, and it was found that the extracted beam sizes at different kicker angles were in good agreement with simulation results [35]. When it becomes possible to test the complete simulation solution, it is expected that there will not be perfect agreement with simulation; for example there may be errors in the fields of beam line elements. However, the simulation can be a tool to track down the source of discrepancies, such as understanding the effect of the gun field on the beam symmetry. More importantly, the optimization results should provide significant improvement in efficiency for achieving a good operational regime.

V. CONCLUSIONS

A general-purpose framework for solving multiobjective optimization problems was presented. Its modular design simplifies the application to simulation-based optimization problems for a wide range of problems and allows to exchange the optimization algorithm. The flexibility of being able to adapt both ends of the optimization process, the forward solver and the optimization algorithm simultaneously not only leads to broad applicability but it facilitates tailoring the optimization strategy to the optimization problem as well.

The framework was integrated into OPAL, and used to study a beam dynamics problem at the AWA. A scan of time step and hyperparameters was done to determine computational settings. Then a full scale physics optimization was performed. Optimization of the 3D beam size

FIG. 10. Optimized beam sizes along high charge beam line. The gun is located at $s = 0$, both x and y beam sizes are shown. The black line represents the relevant beam line aperture, while the vertical green line indicates the location of the optimization.

and energy spread was accomplished. The TBA beam line presented is currently being installed at the AWA. Once installation is complete, the results shown here will guide future experiments at the AWA.

In contrast to approaches that are tightly coupled to the optimization algorithm, the range of possible applications is much wider. Even in cases where the mathematical model of the forward solver is not known exactly, fixed or real time measurements can be used to guide the search for the Pareto optimal solutions. Finally, combining a multi-objective optimization framework, such as the one presented, with practical experience in the field should expedite the decision making process in the design and operation of particle accelerators.

ACKNOWLEDGMENTS

[1] X. Huang, Robust simplex algorithm for online optimization, Phys. Rev. Accel. Beams **21,** 104601 (2018).

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

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

[4] Y. Chao, F. Ames, R. Baartman, I. Bylinskii, S. Dechoudhury, G. Goh, S. Koscielniak, R. E Laxdal, M. Marchetto, L. Merminga, V. Naik, V. A Verzilov, F. Yan, and V. Zvyagintsev, in *Proceedings of ICAP09, San Francisco, CA, 2009* (2009), p. THPSC012.

[5] C. Gong and Y. Chao, in *Proceedings of ICAP2012, Rostock-Warnemünde, Germany, 2012* (JACoW, Geneva, Switzerland, 2012), p. TUABI1.

[6] A. Hofler, B. Terzic, 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. ST Accel. Beams **16,** 010101 (2013).

[7] B. Terzic, A. S. Hofler, C. J. Reeves, S. A. Khan, G. A. Krafft, J. Benesch, A. Freyberger, and D. Ranjan, Simultaneous optimization of the cavity heat load and trip rates in linacs using a genetic algorithm, Phys. Rev. ST Accel. Beams **17,** 101003 (2014).

[8] C. Gong and Y. Chao, in *Proceedings of ICAP2015, Shanghai, China, 2015* (JACoW, Geneva, Switzerland, 2015), p. THCWC4.

[9] C. Gong, A novel optimization platform and its applications to the TRIUMF energy recovery linac, Ph.D. thesis, The University of Columbia, 2015.

[10] C. Gulliford, A. Bartnik, and I. Bazarov, Multiobjective optimizations of a novel cryocooled dc gun based ultrafast electron diffraction beam line, Phys. Rev. Accel. Beams **19,** 093402 (2016).

[11] C. Gulliford, A. Bartnik, I. Bazarov, and J. Maxson, Multiobjective optimization design of an rf gun based electron diffraction beam line, Phys. Rev. Accel. Beams **20,** 033401 (2017).

[12] M. Kranjevi, A. Adelmann, P. Arbenz, A. Citterio, and L. Stingelin, Multi-objective shape optimization of radio frequency cavities using an evolutionary algorithm, Nucl. Instrum. Methods Phys. Res., Sect. A **920,** 106 (2019).

[13] P. Physics, General Particle Tracer (GPT), (1996-2018), http://www.pulsar.nl/gpt.

[14] M. Borland, Elegant: A flexible SDDS-compliant code for accelerator simulation, Argonne National Laboratory Technical Report No. LS-287, 2000–2018.

[15] A. Adelmann, C. Baumgarten, M. Frey, A. Gsell, V. Rizzoglio, J. Snuverink, C. Metzger-Kraus, Y. Ineichen, S. Russell, C. Wang, S. Sheehy, C. Rogers, and D. Winklehner, The OPAL (object oriented parallel accelerator library) framework, Paul Scherrer Institut Technical Report No. PSI-PR-08-02, 2008–2018.

[16] Please see Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevAccelBeams.22.054602 for details on the code implementation, e.g. framework components, optimizer, forward solver implementation, and parallelization related to this work.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. **6,** 182 (2002).

[18] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, in *Evolutionary Multi-Criterion Optimization*, edited by C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele, and K. Deb (Springer, Berlin, Heidelberg, 2003), pp. 494–508.

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

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by Simulated Annealing, Science **220,** 671 (1983).

[21] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95—International Conference on Neural Networks*, (IEEE, Perth, WA, Australia, 1995), Vol. 4, pp. 1942–1948.

[22] M. Dorigo, V. Maniezzo, and A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man. Cy. B **26,** 29 (1996).

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

[24] D. Karaboga, Erciyes University Press Technical Report No. TR06, 2005.

[25] H. Shah-Hosseini, The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm, Int. J. Bio-Inspir. Com. **1,** 71 (2009).

[26] J. Durillo, A. Nebro, F. Luna, B. Dorronsoro, and E. Alba, jMetal: a java framework for developing multi-objective optimization metaheuristics, University of Málaga, ETSI

Informática, Campus de Teatinos Technical Report No. ITI-2006-10, 2006.

[27] A. Liefooghe, M. Basseur, L. Jourdan, and E. Talbi, in *Evolutionary Multi-Criterion Optimization* (Springer, New York, 2007), pp. 386–400.

[28] B. Filipič and M. Depolli, in *Parallel Computing*, edited by R. Trobec, M. Vajteršic, and P. Zinterhof (Springer, New York, 2009), pp. 217–240.

[29] C. León, G. Miranda, and C. Segura, Metco: A parallel plugin-based framework for multi-objective optimization, International Journal on Artificial Intelligence Tools (Architectures, Languages, Algorithms) **18**, 569 (2009).

[30] S. Huband, L. Barone, L. While, and P. Hingston, in *Proceedings of International Conference on Evolutionary Multi-Criterion Optimization* (Springer-Verlag, Berlin, Heidelberg, 2005), pp. 280–295.

[31] L. While, L. Bradstreet, and L. Barone, A fast way of calculating exact hypervolumes, IEEE Trans. Evol. Comput. **16**, 86 (2012).

[32] W. Gai, J. G. Power, and C. Jing, Short-pulse dielectric two-beam acceleration, J. Plasma Phys. **78**, 339 (2012).

[33] C. Jing *et al.*, Electron acceleration through two successive electron beam driven wakefield acceleration stages, Nucl. Instrum. Methods Phys. Res., Sect. A **898**, 72 (2018).

[34] F. Gao, M. E. Conde, W. Gai, C. Jing, R. Konecny, W. Liu, J. G. Power, T. Wong, and Z. Yusof, Design and testing of a 7.8 GHz power extractor using a cylindrical dielectric-loaded waveguide, Phys. Rev. ST Accel. Beams **11**, 041301 (2008).

[35] N. Neveu, Beam line design for fully staged two beam acceleration at the Argonne Wakefield Accelerator Facility, Ph.D. thesis, Illinois Institute of Technology, 2018.