

# EVOLUTIONARY ALGORITHM FOR THE NEUTRINO FACTORY FRONT END DESIGN

ALEXEY A. POKLONSKIY

*Department of Physics & Astronomy, Michigan State University,  
Biomedical Physical Sciences Bldg., East Lansing, Michigan, 48824, USA  
poklonsk@msu.edu*

DAVID NEUFFER

*MS 221, Fermi National Accelerator Laboratory,  
Batavia, Illinois, 60510, USA  
neuffer@fnal.gov*

The Neutrino Factory is an important tool in the long-term neutrino physics program. Substantial effort is put internationally into designing this facility in order to achieve desired performance within the allotted budget. This accelerator is a secondary beam machine: neutrinos are produced by means of the decay of muons. Muons, in turn, are produced by the decay of pions, produced by hitting the target by a beam of accelerated protons suitable for acceleration. Due to the physics of this process, extra conditioning of the pion beam coming from the target is needed in order to effectively perform subsequent acceleration. The subsystem of the Neutrino Factory that performs this conditioning is called Front End, its main performance characteristic is the number of the produced muons.

Evolutionary Algorithms demonstrated themselves as a reliable and efficient tool for exploration, optimization and ultimately decision-making during the design process. In this work we describe the scenario for the Neutrino Factory Front End production optimization via the GATool Evolutionary Algorithm implemented in COSY Infinity and discuss the results of this optimization.

*Keywords:* Genetic algorithm; neutrino factory; accelerator design.

PACS numbers: 41.85.Ct, 41.75.Lx

## 1. Neutrino Factory

### 1.1. *Purpose and History*

The Neutrino Factory is a very important facility for the long-term neutrino physics program. Modern technologies of particle accelerators open the possibility to build an accelerator complex to produce and capture more than  $10^{20}$  muons per year. The idea of the accelerator where the pions are injected into a storage ring, decay to produce muons captured within the same ring and then further decay into a neutrino beam was proposed several times by different researchers starting from

Koshkarev in 1974. All proposed designs suffered from the same basic problem: the resulting neutrino beam intensity was low [1].

Current design of the Neutrino Factory was proposed by Geer in 1997 [2]. He suggested to create muons from an intense pion source at low energies, then compress their phase space to produce a bright beam which is then accelerated to the energies of several tens of GeV and injected into a storage ring with long straight sections, where they decay into highly intense neutrino beams

$$\mu^- \rightarrow e^- \nu_\mu \bar{\nu}_e, \quad \mu^+ \rightarrow e^+ \bar{\nu}_\mu \nu_e \quad (1)$$

that can be used for the extensive study of neutrino oscillations [3] and neutrino interactions with the required high precision. Studies performed so far have shown that the Neutrino Factory gives the best performance among all considered neutrino sources over virtually all of the parameter space (except, for the case when  $\theta_{13}$  turns out to be large). However, its time scale and cost are still being actively investigated [4].

## 1.2. Design Overview

The Neutrino Factory is a secondary beam machine; that is, a production beam is used to create secondary beams that eventually provide the desired flux of neutrinos. For the Neutrino Factory the production starts from a high intensity proton beam that is accelerated to a moderate energy (beams of 2–50 GeV have been considered by various groups) and impinges on a target, typically made from a high-Z material (baseline choice is a liquid Hg jet). Collisions between the proton beam and the target nuclei produce secondary beams of pions that quickly decay (26.0 ns) into longer-living (2.2  $\mu$ s) muon beams. The remaining part of the Neutrino Factory is used to condition the muon beam, rapidly accelerate it to the desired final energy of a few tens of GeV, and then store it in a decay ring with long straight sections where the intense beam of neutrinos is produced from the decaying muons (see (1)). The resulting beam can be used to hit a detector located hundreds or thousands of kilometers from the source or some other scenario.

The Feasibility Study II [5] that was carried out jointly by the Brookhaven National Laboratory (BNL) and the U.S. Neutrino Factory and Muon Collider Collaboration, established most of the current Neutrino Factory design ideas. Although a number of other ideas and their variations of existing ones was proposed since FS II, later studies mainly concentrated on the exploration of the already proposed concepts and their combinations. The main goals were set to a cost/performance analysis and development of the consensus on a baseline design for the facility [6]. It is noteworthy that the FS II design is highly influenced by a specific scenario of sending a neutrino beam from BNL to a detector in Carlsbad, New Mexico. More generally, however, this design exemplified the Neutrino Factory for which the Feasibility Studies had demonstrated technical feasibility, established a cost baseline, and the expected range of physics performance. Another important feature of this

scheme is that such a Neutrino Factory could be comfortably constructed on site of an existing U.S. laboratory, such as BNL or Fermi National Accelerator Laboratory (FNAL).

Here we list the main components of a Neutrino Factory (see example of the RLA-acceleration based variant of the FS IIa design in Figure 1) and their primary functions:

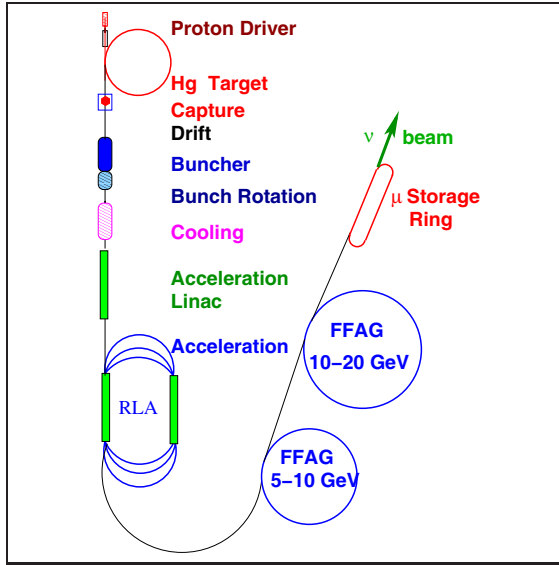


Fig. 1. Neutrino Factory schematics from the Feasibility Study IIa (RLA acceleration variant).

- **Proton Driver** provides 4 MW beam of a moderate energy (several GeV) protons on target.
- **Target** A high-Z target is put inside a 20 T solenoidal field (superconducting solenoid) to capture pions produced in the interactions of the incident proton beam with the nuclei of the target material (liquid Hg jet) (see the longitudinal distribution of the particles 12 m from the target obtained from the MARS simulation code [7] in Figure 2).
- **Front End** consists of the parts of the Neutrino Factory between the target and the acceleration section. As can be seen in Figure 2, pions that are produced by nuclear collisions on target occupy significantly large longitudinal phase space. The transverse phase space is mainly determined by the strength of the magnetic field inside the solenoidal capture channel. According to the properties of the dynamics of particles in the solenoid, the particles with the transverse momentum satisfying the following condition

$$p_{\perp} < 0.3 \frac{BR}{2},$$

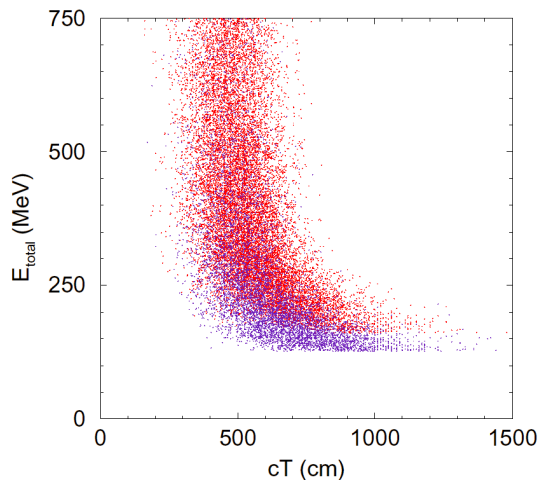


Fig. 2. Distribution of particles' energies 12m from the target calculated by MARS,  $E_{\text{total}} = E_0 + T$ , where  $E_0$  is a rest energy (105.6 MeV for muons),  $T$ —kinetic energy.

where  $B$  is a solenoidal field strength,  $R$  is a radius of the solenoid, are captured after the target. In order to efficiently accelerate the beam, it has to be pre-conditioned to be fully contained within the capture transverse acceptance ( $30 \pi$  mm-rad) and the longitudinal acceptance (150 mm) of the subsequent accelerating section. Another constraint that the resulting beam has to satisfy is that only the particles that are contained within the longitudinal bucket of the accelerating system (bucket area depends on the RF frequency, phase and a field gradient) are considered captured to the accelerating regime. Transverse emittance has to be decreased by cooling in order to achieve optimal intensity. Hence the main figure of merit for the Front End is the ratio of the produced and captured muons to the number of incoming pions. See example of the longitudinal dynamics of a beam with a relatively small initial phase space in Figure 3.

Front End consists of the following subsystems:

- **Capture.** The magnetic field at the target is smoothly tapered down to a much lower value (2 T) which is then maintained through the bunching and phase rotation sections to keep the beam confined in the channel.
- **Decay.** This region is just an empty magnetic lattice where pions are allowed to decay to muons and where the particles of the resulting beam develop the correlation between a longitudinal coordinate and an energy.
- **Bunching and Phase Rotation.** First, the large beam of muons is bunched with RF cavities of modest gradient. Their frequencies decrease as we proceed down the beam line. After bunching, another set of RF cavities with changing frequencies is used to rotate the beam in longitudinal phase space in order to reduce its energy spread and match the frequency to the one of the downstream RF cavities, for efficient acceleration.

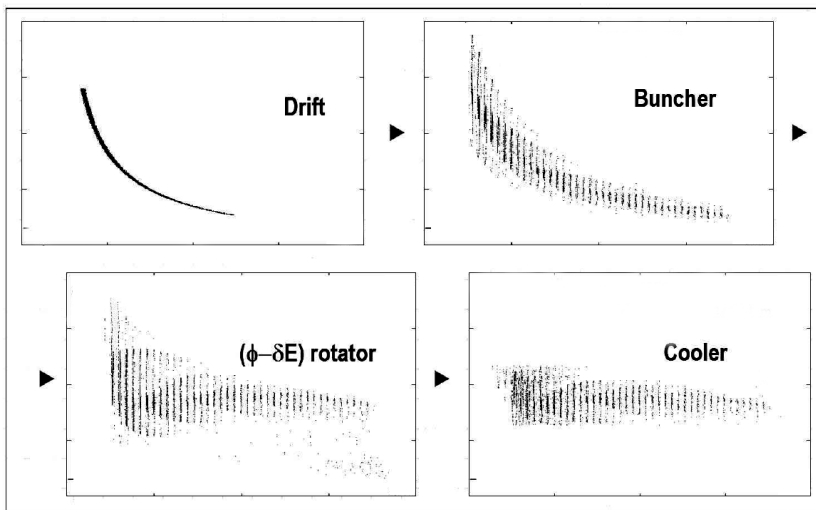


Fig. 3. Example of the longitudinal beam dynamics in the Front End (phase-energy plane).

- **Ionization Cooling.** A solenoidal focusing channel filled with high-gradient RF cavities and LiH absorbers cools the transverse normalized RMS emittance of the beam. On this stage muons in the momentum range of 150–400 MeV/c pass through the absorbers (made from LiH in the baseline design) thus losing the total momentum (both longitudinal and transverse components). They are then reaccelerated in RF cavities to regain only the longitudinal momentum component. The total effect is a decrease in the transverse momentum spread and, therefore, reduction of the transverse emittance.
- **Acceleration.** Increases the beam kinetic energy from  $\approx 138$  MeV to a final energy in the range of 20–50 GeV. A superconducting pre-acceleration linac with solenoidal focusing is used to increase the muon beam energy to 1.5 GeV. It is then followed by a Recirculating Linear Accelerator (RLA), arranged in a dogbone geometry, that increases the beam energy to 5 GeV. Finally a pair of cascaded Fixed-Field Alternating Gradient (FFAG) rings with combined-function doublet magnets is used to bring beam energy up to 20 GeV. Additional FFAG stages could be added to reach a higher beam energy.
- **Storage and Decay Ring.** A compact racetrack-shaped superconducting storage ring in which  $\approx 35\%$  of the stored muons decay to neutrinos and are then sent towards the detector located approximately 3500 km from the ring. Muons survive in a ring for  $\approx 500$  turns.

The baseline Front End schematics from the latest International Scoping Study [8] is demonstrated in Figure 4. The baseline proton driver has an energy of 10 GeV, capture system is a 12 m long channel where the solenoidal field drops from initial 20 T to 2 T at the end while the channel radius increases from 75 mm to 250 mm.

It is followed by a 100 m long decay section where the pions decay to muons and develop a correlation between the temporal position and an energy. This correlation is then employed by the 50 m long bunching section in order to split the beam into a train of bunches via a set of RF cavities of a modest gradient and decreasing frequencies. Then another set of RF cavities with higher gradients in the 50 m long rotator section is employed to rotate the beam in the longitudinal phase space to reduce its energy spread. The achieved final RMS energy spread in this scheme is  $\approx 10.5\%$ . Then 80 m long channel filled with high-gradient 201.25 MHz RF cavities and LiH absorbers in the solenoidal field is used to cool the transverse normalized RMS emittance from  $17\pi$  mm-rad to  $\approx 7$  mm-rad at a central muon momentum of  $\approx 220$  MeV/c.

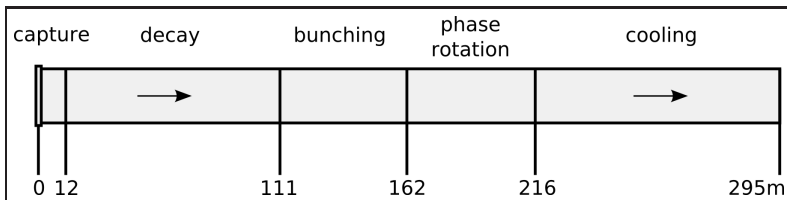


Fig. 4. The baseline Front End schematics from the latest International Scoping Study.

## 2. Evolutionary Algorithms

Evolutionary algorithms form a family of optimization methods that use a set of points to explore the objective function landscape. They are inspired by the process of evolution described by Darwin in his revolutionary work “Origin of Species” first published in 1859 [9]. According to it, the main driving forces of evolution are the variability in living organisms and the natural selection implicitly performed on them by the environment. Over time those forces shape different species to be very sophisticated inhabitants of the environment, i.e. make them fit to it.

If we view an objective function as an environment and points in a search space as organisms evolving to find the best places in this environment (which are for our purposes minima), we can easily sketch a general model of evolution suitable for optimization which is called *Evolutionary Algorithm* (EA) (see Figure 5). Having the evidence of the efficiency of this algorithm in the form of a variety of very well-fit organisms on Earth, there emerged a strong belief that its main principles can be applied to function optimization problems equally successfully.

Note that it actually is a meta-algorithm and that many conventional optimization algorithms described earlier can be formulated in this form. Because of this generality, the family of Evolutionary Algorithms typically includes only the ones that imitate the processes of evolution more straightforwardly and more accurately. A particularly important distinctive feature of EAs is that the members of the population actively exchange information about the search space. However the boundary

```
Generate initial population, evaluate fitness
While stop condition not satisfied do
    Produce next population by
        Selection
        Recombination
    Evaluate fitness
End while
```

Fig. 5. Evolutionary Algorithm.

is blurry and some EAs, for example the Differential Evolution [10], are closer to multi-point direct search methods than to “true” Evolutionary Algorithms.

It is worth noting that EA does not pose any restrictions on the search space and members of the population which, multiplied by a variety of different approaches to define fitness, selection, recombination and mutation, leads to a very broad field of applications. Examples include a wide variety of the optimization problems: numerical optimization, combinatorial optimization, circuit design, scheduling problems, video and sound quality optimization, control systems, image analysis, marketing and economics, traffic control, manufacturing and many others. While EAs do not explicitly guarantee to find even a local minimum, practical applications demonstrate that frequently they are able to find a global minimum or at least produce a practically acceptable solution.

Each of those applications is usually tied to a particular flavour of the Evolutionary Algorithms. Genetic Algorithms (GAs) [11] often encode parameters as strings of bits and modify them with logical operators thus are better suited for combinatorial optimization, for example for a class of problems equivalent to the famous Traveling Salesman Problem [12]. Genetic and Evolutionary Programming [13] evolve computer programs. Evolution Strategies (ES) [14] and Differential Evolution (DE) both use real numbers and arithmetic evolutionary operators for continuous function optimization. It is also worth noting a rapidly increasing interest in the development of the optimizers mimicking various optimization and search processes of nature: *Particle Swarm Optimization*, *Ant Colony Optimization*, *Tabu Search*, *Cultural algorithm*, etc. [15] and their successful application to many real-world problems.

Distinctive advantages of the EAs include:

- relative ease of implementation,
- ability to efficiently find global optima avoiding local ones even in very large search spaces,
- no requirements on the objective function other than the ability to calculate its value at every point of the search space,

- good tolerance to noise,
- ability to work even when the traditional search methods fail.

The interest to the field of EAs is steadily growing, active research on the development of the EAs and their applications produced a large number of publications: bibliography on Evolutionary Computation as of now contains more than 4000 entries on Evolutionary Computation and related areas [16].

### 3. Implementation

The algorithm we implemented uses the best features of the Evolutionary Strategies (ES), Genetic Algorithms (GA) and Differential Evolution (DE). From Evolutionary Strategies we adopted the representation of a potential solution as a vector of the real numbers, i.e. vector of a problem arguments:

$$\mathbf{x} = (x_1, x_2, \dots, x_v)^T. \quad (2)$$

Then the population members are:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iv})^T, \quad i = 1, \dots, N$$

and

$$\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N))^T = (f_1, f_2, \dots, f_N)^T \quad (3)$$

is a vector containing function evaluations for the members of the population,  $\underline{f}$  and  $\overline{f}$  denote the minimum and maximum function values of the population members, correspondingly. Noting the success of the ES and DE both using such representation, we suggested that it is more adequate for the optimization of the problems with the real-valued parameters than the binary encoding frequently used in GAs.

It supports several methods for fitness scaling: Linear, Proportional and Rank; several methods for selection: Roulette Wheel, Stochastic Uniform and Tournament; two types of mutation: Uniform and Gaussian; Arithmetic (or Continuous) crossover with additional randomization. The next generation is produced from the previous by means of mutation and crossover essential for exploration and exploitation of the search space as well as elite members transfer which is essential for the preservation of the already found solutions. The algorithm itself is presented in Fig. 6, its default parameters are listed in Fig. 7, for more detailed description of the algorithm and its parameters see [17].

### 4. Optimization

As can be seen from the Front End design description examples of the parameters that can be changed for different sections include:

- (1) *Capture and Decay*: the length of the section  $L_D$  and the focusing fields.



```

Randomly generate initial population, set predefined
    members, if any
Calculate objective function values, scale to fitnesses
Update statistics
While any of the stop conditions is not satisfied do
    Perform Roulette Wheel/Stochastic Uniform/Tournament
        Selection
    Generate next population
        Produce mutants by Uniform/Gaussian Mutation
        Produce children by Continuous Crossover
        Copy elite members
    Replace old population with newly generated
    Calculate objective function values, scale to
        fitnesses
    Update statistics
End while

```

Fig. 6. GATool search algorithm.

```

Reproduction:    number of elite = 10, mutation rate = 0.2
Mutation:        UNIFORM, gene mutation probability = 0.1
Crossover:       HEURISTIC, ratio = 0.8, randomization is on
Fitness scaling: RANK
Selection:       STOCHASTIC UNIFORM
Creation:        UNIFORM
Areal:           initial box = [-10,10] x ... x [-10,10]
                  global box  = [-10,10] x ... x [-10,10]
                  killing is off
Stopping:        max generations = 1000,
                  stall generations = 25,
                  tolerance = 1E-5

```

Fig. 7. GATool's default parameters (see Appendix B, [17] for detailed description of parameters).

- (2) *Bunching*: the length of the section  $L_B$ , RF voltages  $V_B^i$ ,  $i = \overline{1, n_{\text{rfs}}}$  or initial and final voltage and the law of voltage increase (linear, quadratic, etc.). Final frequency is usually strictly specified by the cooling/accelerating subsections of the whole accelerator, but can also be varied if it can benefit the total Neutrino Factory performance.
- (3) *Phase Rotation*: the length  $L_{\varphi_R}$ , RF voltage  $V_{\varphi_R}$  of the phase-energy rotation section, number  $N$  of full RF field oscillation periods separating the main central particle (with  $n = 0$ ) and the second central particle, and the vernier parameter  $\delta$  (see [18] for detailed explanation). Also the kinetic energy  $T_c$  of the main central particle can be changed (usually  $T_c$  is taken as the peak of energy distribution of the particles of the beam).
- (4) *Ionization Cooling*: parameters of the RF cavities ( $\nu_{\text{rf,cool}}$ ,  $V_{\text{rf,cool}}$ ,  $\varphi_{\text{rf,cool}}$ ); placement, width, material and the location of absorbers, focusing field strength and profile.

For our study we consider the exploration of the cooling section: we varied the RF cavities parameters and the desired momentum of the central particles in the beam within the ranges obtained from the physical considerations in order to find a configuration which would provide optimal production parameters.

Most of the numerical studies of the beam dynamics in the Front End is done with ICOOL—the de-facto standard Muon Collaboration particle tracking code originally developed for the ionization cooling simulations of the muon beams in 1999 [19] and actively developed over years to include new elements and models (available at <http://pubweb.bnl.gov/users/fernow/www/icool/readme.html>). It simulates beam dynamics in accelerator coordinates, performing particle-by-particle propagation through materials and electromagnetic fields. Field can given by built-in models that include most common accelerator elements and their approximations, be calculated from field maps or Fourier coefficients and read from files. ICOOL accurately models the decays of particles and their interactions with matter including energy loss, energy straggling and multiple Coulomb scattering. The beam can be generated from the uniform or Gaussian distributions or read from an input file.

ICOOL belongs to the family of so-called ray tracing codes, i.e. it calculated particle dynamics via the Runge-Kutta or the Boris numerical integration methods. For the description of dynamics it uses Frenet-Serret coordinate system. This is a right-handed system where  $s$  is tangent to the reference orbit,  $y$  is vertical and  $x$  is the third orthogonal coordinate. In a circular orbit  $x$  is in the radial direction. The reference orbit is defined to be that path where the transverse coordinates  $x$  and  $y$  and the transverse momenta  $p_x$  and  $p_y$  always remain zero. The shape of the reference orbit in a global Cartesian coordinate system is determined by the curvature parameter.

Various tools are developed to analyze the data produced by ICOOL simulation. The code that is used as a standard for the emittance calculation is called ECALC9

[20]. It allows user to compute number of particles in the fixed phase space volumes. The input is given as a file that contains the particle type, maximum and minimum value for  $p_z$  in GeV/c, two different cuts for transverse acceptance in m-rad (to obtain number of particles that correspond to two different acceptance cuts with all other cuts the same at once), a longitudinal acceptance cut in m-rad and a RF frequency to determine the RF bucket area for another longitudinal cut.

The toolchain that was developed for the optimization consists of the 4 programs:

- *COSY Infinity* [21]: provided the implementation of GATool optimization method.
- *ICOOL*: performed actual simulations of the beam dynamics in the Front End with the parameters values passed from COSY.
- *ECALC9*: performed analysis of the results of the ICOOL simulations, calculated the number of particles within desired acceptance that served as the objective function value.
- *glue.pl* is a Perl script that was used to control other programs in the toolchain. It was setting up the Front End lattice to ICOOL based on the control parameters provided by COSY, running ICOOL and then ECALC9 to obtain the objective function value which it was then passing back to COSY for the whole optimization run.

Initial distribution consists of 8000 particles after the target was generated by MARS simulation code for the 24 GeV proton beam on the Hg jet target [7]. The Front End lattice that was used for this study starts from the target and included capture, decay, bunching and phase rotation regions as well as cooling section and a matching between phase rotation and cooling subsystems:

- Capture: 15.25 m of the vacuum channel immersed in a solenoidal field that falls off starting from 20 T on the target to 2 T at the end of the channel. At the same time radius of the channel increases from 0.075 m to 0.3 m.
- Decay: vacuum channel of a constant aperture of 0.3 m immersed in a constant solenoidal field of 2 T.
- Bunching: vacuum channel of a constant aperture of 0.3 m and a total length of  $L = 21$  m immersed in a constant solenoidal field of 2 T. An array of RF cavities separated by drifts so as to perform the adiabatic bunching (28 cells, each consists of the drift of 0.125 m, followed by RF cavity of 0.5 m and another drift of 0.125 m). Particles are bunched around the central momentum of 0.280 GeV/c. An integer number of wavelength that separate two reference particles is 7, initial RF gradient is set to 15 MV/m, the law that determines the RF gradient at the longitudinal coordinate  $z$  counted from the start of the buncher is

$$V_{\text{rf}} = V_{0,\text{rf}} \frac{z}{L}.$$

- Phase rotation: vacuum channel of a constant aperture of 0.3 m and a total length of  $L = 24$  m immersed in a constant solenoidal field of 2 T. An array of RF cavities separated by drifts so as to perform the rotation of the beam in the longitudinal space by decelerating higher-energy bunches and accelerating lower-energy ones (32 cells, each consists of the drift of 0.125 m, followed by RF cavity of 0.5 m and another drift of 0.125 m). Vernier offset  $\delta$  is 0.1, RF gradient is 15 MV/m for all cavities.
- Cooling: vacuum channel of a constant aperture of 0.3 m and a total length of  $L = 93$  m immersed in an alternating solenoidal field of the maximum strength  $\approx 2.5$  T. An array of the 124 cells (0.75 m each), containing LiH absorbers to provide total momentum loss and RF field to provide longitudinal momentum regain to cool the transverse emittance of the beam. The first four cells are immersed in the solenoidal field designed so as to match the transverse particle dynamics in phase rotation section to the one of the cooling section. All RF cavities have the frequency of 201.25 MHz, field gradient is 18 MV/m and the RF phase is 30 degrees.

This particular design is shorter than the one of the baseline and is aimed to study the cost gains versus the performance losses resulted from shortening the Front End by removing some of its elements that were originally present. Another goal is to study the potential of this design for the Muon Collider project [22].

We used the described lattice as a reference design and explored its performance related to changes in the following control parameters:

- RF frequency in cooling section (also influences the following accelerator section):  $\nu_{\text{rf,cool}} \in [200, 204]$  MHz.
- RF field gradient in cooling section:  $V_{\text{rf,cool}} \in [12, 20]$  MV/m.
- RF field phase in cooling section:  $\varphi_{\text{rf,cool}} \in [0, 360]$  degrees.
- Central momentum in the first 4 matching sections of the cooling channel:  $p_{\text{c,match}} \in [0.22, 0.24]$  GeV/c.

The values of the cuts to set up ECALC9 analysis were obtained as an estimate of the acceptance of the subsequent acceleration subsystem:

- minimum and maximum  $p_z$ : 0.100 GeV/c and 0.300 GeV/c, correspondingly;
- transverse acceptance cut: 30E-3 m-rad;
- longitudinal acceptance cut: 0.25 m-rad;
- RF frequency for the bucket calculation set to a value used by RF cavities of the cooling section (on of the control parameters).

The number of particles withing the specified acceptance ( $n_2$ ) was chosen as an objective value to be maximized. GATool parameters were set to default values (see Figure 7, p.967), population size for this 4-dimensional problem was set to 250 (dimension $\times$ 62.5) which provides a good compromise between the total time of the

search given the expensiveness of the objective function calculation (see below) and a setting that is known to provide good GATool performance.

Even though 2000 particles were used instead of 8000 in order to reduce the computation time (ICOOL integrates each particle dynamics separately) one simulation run of the described Front End lattice still takes approximately 0.4 hour on a Pentium IV 2Mhz computer with 1Gb of memory. Therefore the calculations needed to perform one step of the GATool optimization took approximately 100 hours. Since a typical number of generations needed for GATool to explore the parameters space and converge can get above 100, subsequent studies (possibly on more realistic and thus more computationally expensive lattices), one can clearly benefit from even the embarrassingly simple parallelization of the objective function evaluation.

Several of the best obtained results (elite in GATool terminology) from three runs (each of them took several months to complete on a single machine) were evaluated using the described scheme and the full initial number of particles of 8000. The control parameters and objective function values for the found designs are listed along with the reference design provided in Table 1. Range of the values of the objective function with control parameters from the regions listed earlier on combinations occurred during the optimization is from 15 to 497. From the table it is seen that the optimization of the current scheme with control parameters in the specified ranges was unable to achieve the designs that have statistically (simulation includes stochastic processes) significantly better production efficiency. Although this can not serve as a rigorous proof of the nonexistence of such designs, noting generally good performance of the GATool on other problems, we can suggest that this gives reason to believe that the reference design is, indeed, optimal. Relatively small deviations of the optimal RF frequency (201.20–201.55) solution and RF gradient (17.67–18.88) among all the solutions except for the 1st opt. run, best suggest that the reference parameters are also robust and located near the global optimum. This observation is particularly important here since the parameters of the devices that are calculated by numerical simulations eventually have to be implemented by physical devices operating with finite precision and subject to construction errors.

The best solution obtained from the 1st optimization run provided one of the best performance and with one of the smallest small RF gradient (among other solutions) which is generally easier/cheaper to manufacture. However, its final frequency is different from the frequency of the current baseline accelerating section thus additional studies on the combined performance might reveal additional benefits or drawbacks of this solution. The best solution from the 3rd optimization run provided the same performance on the frequency that is much closer to the reference 201.25 and thus might be preferable. Some of the other sets of parameters that provide similar production performances can also be useful since they might be easier or cheaper to obtain, or provide additional opportunities for the designers

of the downstream sections of the Neutrino Factory. Another important result that is obtained is that we established a framework for the Neutrino Factory Front End lattice numerical optimization. It can be used for many optimization scenarios, including, for example, a simultaneous optimization of all control parameters of the most realistic Front End simulation on the large ranges of the parameter values.

Table 1. Results of the Front End design optimization (in ascending order sorted on the production rate for the 8000 particles initial beam). Column 5 and 6 contain the results of the simulations for 2000 and 8000 particles in the initial beam, correspondingly. The number of the particles in the exit beam per each particle in the initial beam is listed in parentheses

Parameters	$\nu_{\text{rf,cool}}$	$V_{\text{rf,cool}}$	$\varphi_{\text{rf,cool}}$	$p_{\text{c,match}}$	$n_2$	$n_2$
	[MHz]	[MV/m]	[degrees]	[GeV/c]	(from 2000) [particles]	(from 8000) [particles]
reference parameters	201.25	18.00	30.000	0.220	498 (0.249)	1740 (0.218)
3rd opt. run, 6th best	201.46	17.77	11.320	0.229	480 (0.240)	1791 (0.224)
3rd opt. run, best	201.40	17.06	12.648	0.226	492 (0.246)	1782 (0.223)
1st opt. run, best	200.55	17.10	26.970	0.220	467 (0.234)	1780 (0.222)
3rd opt. run, 5th best	201.28	17.76	12.457	0.226	484 (0.242)	1773 (0.222)
3rd opt. run, 3rd best	201.47	17.67	13.470	0.228	485 (0.243)	1762 (0.220)
3rd opt. run, 2nd best	201.42	17.68	12.555	0.226	486 (0.243)	1750 (0.219)
3rd opt. run, 7th best	201.34	17.68	12.020	0.226	479 (0.240)	1746 (0.218)
2nd opt. run, 2nd best	201.24	18.91	20.520	0.228	471 (0.236)	1714 (0.214)
3rd opt. run, 4th best	201.48	17.75	11.860	0.227	485 (0.243)	1669 (0.209)
2nd opt. run, best	201.20	18.88	22.477	0.230	497 (0.249)	1643 (0.205)

## 5. Conclusions

In this work we described our Genetic Algorithm and demonstrated its applicability to a real-life problem of the accelerator physics. We developed a method for the Neutrino Factory Front End optimization and obtained some practically useful numerical results. We are planning to use this method for the subsequent studies in design and optimization of this facility. We also hope to see more problems of accelerator physics to be approached using this approach.

## References

1. D. G. Koshkarev. Proposal for a decay ring to produce intense secondary particle beams at the sps.
2. S. Geer. Neutrino beams from muon storage rings: Characteristics and physics potential. *Phys. Rev.*, D57:6989–6997, 1998.
3. M. Apollonio et al. Oscillation physics with a neutrino factory. ((g)) ((u)). *CERN Yellow Report on the Neutrino Factory*, 2002.
4. Michael S. Zisman. International scoping study of a future accelerator neutrino complex. In *Proceedings of EPAC 2006*, pages 2427–2429, Edinbrugh, Scotland, 2007.
5. S. Ozaki, R. Palmer, M. Zisman, J. Gallardo, and (eds.). Feasibility study-II of a muon-based neutrino source. Technical Report BNL-52623, Muon Collider Collaboration, June 2001. see <http://www.cap.bnl.gov/mumu/studyii/FS2-report.html>.

6. J. S. Berg, S. A. Bogacz, S. Caspi, J. Cobb, R. C. Fernow, J. C. Gallardo, S. Kahn, H. Kirk, D. Neuffer, R. Palmer, K. Paul, H. Witte, and M. Zisman. Cost-effective design for a neutrino factory. *Phys. Rev. ST Accel. Beams*, 9(1):011001, Jan 2006.
7. N. V. Mokhov and A. van Ginneken. Pion production and targetry at  $\mu^+\mu^-$  colliders. Technical Report Fermilab-Conf-98/041, Fermi National Accelerator Laboratory, January 1998. Published Proceedings of the 4th International Conference on Physics Potential and Development of Muon Colliders, San Francisco, California, December 10-12, 1997.
8. J.S. Berg et al. Iss scoping study: Accelerator design concept for future neutrino factories. Technical Report RAL-TR-2007-23, RAL, 2007.
9. Charles Darwin. *The Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London: John Murray, 6 edition, 1872.
10. Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA, 1995.
11. David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1st ed. edition, 1989.
12. P. Prinetto, M. Rebaudengo, and M. Sonza Reorda. Hybrid genetic algorithms for the traveling salesman problem. In *Proc. of the International Conference on Neural Networks and Genetic Algorithms*, pages 559–566, 1993.
13. John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Complex Adaptive Systems. The MIT Press, December 1992.
14. Schwefel H.-P. *Evolution and optimum seeking*. Wiley, New York, 1991.
15. Johannes J. Schneider and Scott Kirkpatrick. *Stochastic Optimization*. Scientific Computing. Springer, 1st edition, November 2006.
16. Bibliography on Evolutionary Computation. on-line. <http://www.fmi.uni-stuttgart.de/fk/evolalg/eabib.html>.
17. Alexey Poklonskiy. *Evolutionary Optimization Methods for Accelerator Design*. PhD thesis, Michigan State University, USA, 2008.
18. A.A. Poklonskiy, D.A. Ovsyannikov, A.D. Ovsyannikov, D. Neuffer, and M. Berz. Exploring the bunching section of the neutrino factory. In *Physics and Control, 2005. Proceedings. 2005 International Conference*, pages 272–277, August 2005.
19. R.C. Fernow. Icool, a simulation code for ionization cooling of muon beams. In *Proc. 1999 Particle Accelerator Conference*, page 3020, New York, 1999. see <http://pubweb.bnl.gov/people/fernow/>.
20. R.C. Fernow. Physics analysis performed by ecalc9. Neutrino Factory and Muon Collider Notes MUC-NOTE-COOL\_THEORY-280, Brookhaven National Laboratory, 2003. see <http://www-mucool.fnal.gov/notes/notes.html>.
21. M. Berz and K. Makino. COSY INFINITY Version 9.0 programmer’s manual. Technical Report MSUHEP-060803, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2006. see also <http://cosyinfinity.org>.
22. Muon Collider Collaboration.  $\mu^+\mu^-$  collider: A feasibility study. Technical Report BNL-52503, Fermilab-Conf-96/092, LBNL-38946, Muon Collider Collaboration, jul 1996. pres. at the Snowmass’96 workshop.