GENETIC ALGORITHMS AND THEIR APPLICATIONS IN ACCELERATOR PHYSICS*

A.S. Hofler†, Thomas Jefferson National Accelerator Facility, Newport News, VA 23608, USA

Abstract

Multi-objective optimization techniques are widely used in an extremely broad range of fields. Genetic optimization for multi-objective optimization was introduced in the accelerator community in relatively recent times and quickly spread becoming a fundamental tool in multi-dimensional optimization problems. This discussion introduces the basics of the technique and reviews applications in accelerator problems.

GENETIC ALGORITHM OVERVIEW

Genetic and evolutionary algorithms (GAs and EAs, respectively) are examples of adapting models for processes observed in nature to create novel approaches to solving seemingly unrelated problems. These methods mimic biological reproduction, the theory of evolution, and the behavior of biological populations to solve mathematical optimization problems. Sets of solutions to an optimization problem are considered successively. The solutions from set i that more closely meet the optimization goal(s) are considered fitter, more inclined to survive, and are used to create the next $i+1$ set of solutions. As the selection and reproduction cycle repeats, the solutions produced are refined, and the defining characteristics for the best solutions coalesce to the optimal value(s). While not an immediately obvious method for optimization, GAs and EAs have proven to be highly effective in solving engineering and accelerator physics problems. Two recommended references on GAs, EAs, and optimization in general are [1, 2].

Optimization Problem Statements

Commonly, optimization problems are cast as minimizing or maximizing a cost function, and optionally, the independent variables may be subject to boundary constraints. Real world optimization problems tend to be multi-objective meaning there are multiple cost functions that must be minimized or maximized simultaneously subject to bounds and inequality constraints. Written generally, a multi-dimensional, multi-objective optimization (MOO) problem is

$$\begin{aligned}
\text{Optimize } & f_j\left(x_1, x_2, ..., x_N\right) & j = 1, 2, ..., J, \\
& g_k\left(x_1, x_2, ..., x_N\right) \geq 0 & k = 1, 2, ..., K, \\
& x_{\min}^{(n)} \leq x_n \leq x_{\max}^{(n)} & n = 1, 2, ..., N, \quad (1)
\end{aligned}$$

where Optimize can be either Minimize or Maximize for each objective function f_j, J is the number of objective functions to be simultaneously optimized, g_k are K inequality constraints, and x_n are the N independent variables to vary subject to a set of bounds constraints, $x_{\min}^{(n)}$ and $x_{\max}^{(n)}$. For a single-objective optimization (SOO), $J = 1$ and $K = 0$. GAs and EAs can be applied to both single- and multi-objective problems.

Solution existence and uniqueness is not guaranteed for SOO and MOO making them difficult to perform, but MOO has the additional complication that the objectives may conflict, meaning a set of independent variables that satisfies one objective goal causes another to move away from its optimal value. An example is minimizing two objective functions that inversely depend on the same independent variable, e.g. $f_1 = x_1$ and $f_2 = x_1^{-1} + x_2$ for $0.5 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$. No single combination of x_1 and x_2 clearly minimizes both f_1 and f_2 simultaneously. In contrast, changing the problem to minimize one function and maximize the other creates two different optimization problems where the objectives do not conflict, and fortunately, both problems have unique solutions ($x_1 = 0.5$ and $x_2 = 1$ minimize f_1 to 0.5 and maximize f_2 to 3, while $x_1 = 1$ and $x_2 = 0$ maximize f_1 to 1 and minimize f_2 to 1).

For the simultaneous minimization problem, each possible x_1 value gives an equally valid solution to the minimization problem for $x_2 = 0$. Absent additional information about a preferred solution or system behavior, there is nothing to differentiate any particular solution from the rest in the x_1 interval. For instance, at $x_1 = 0.5$ and $x_2 = 0$, while $f_2 = 2$ is not its minimum value on the x_1 interval, f_1 is minimized to 0.5. This is the best solution for $x_1 = 0.5$ since $f_2 > 2$ for $x_2 > 0$. Similarly, $f_1 = 1$ and $f_2 = 1$ comprise the best solution for $x_1 = 1$. For the remaining values of the x_1 interval with $x_2 = 0$, neither f_1 nor f_2 attains its individual minimum value, but their values are the optimal ones for each fixed value of x_1. These solutions for $0.5 \leq x_1 \leq 1$ and $x_2 = 0$ are equivalent in terms of the optimization goals, and they form the set of solutions to this optimization problem. When objective functions conflict, a series of individual optimizations must be considered and solved, and the set of solutions is called the Pareto optimal front. The Pareto optimal front is important because it elucidates trade-offs between objectives.

Comparison to Standard Techniques

Historically, standard techniques used for multidimensional optimization problems are either iterative derivative-based methods or systematic parameter scans. GAs and

* Authored by JSA, LLC under U.S. DOE Contract DE-AC05-06OR23177. The U.S. Govt. retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this for U.S. Govt. purposes.

† hofler@jlab.org

ISBN 978-3-95450-138-0

EAs are iterative but not derivative-based. They analyze a series of relatively large sets of solutions or populations to identify candidate solutions to produce new sets of solutions. Each new population is a generation and represents an iteration. Populations are composed of individuals, realizations of Eq. (1). A systematic parameter scan produces one very large exhaustive population in a single generation, and the data are analyzed at the end. GAs and EAs are smarter parameter scans because interleaving parameter space sampling and objective value analysis allows them to identify and concentrate on promising parameter space regions as they progress.

Genetic Algorithm Processing

The distinguishing characteristic of GAs and EAs is the simple but effective process by which the N-dimensional parameter space is sampled and the resulting objective values are analyzed. By way of caveat, there is great variability in algorithm designs, and this discussion may describe features and concepts that are not explicitly used or are omitted in a particular algorithm or application. The general process starts with an initial population created from a set of independent variable values that are randomly produced within the bounds constraints. This generation is a random sample of the N-dimensional parameter space. The objective values are ranked against the optimization goals, and a new population is generated from favorably ranked individuals. The cycle of ranking objectives and generating new populations repeats until the optimization goals are met within some specified tolerance or the maximum number of generations limit is reached.

Fitness is used to rank individuals in a population and indicates the relative strength of an individual. It is a function of the objectives and can include the constraints. Its exact form varies with algorithm and application. In the single objective case, the fitness function can be the objective value itself. Alternatively, fitness may be a measure of how well the objective values meet the optimization goals and adhere to the problem constraints [1, 3, 4]. Typically, stronger individuals have larger fitness values, but not always [3]. In some cases, the fitness concept is not explicitly invoked [5] but can be assumed to be the objective function value(s).

Some general fitness definitions for MOOs use the dominance concept [1] because it is suited to systems with conflicting objectives. An individual is said to dominate another if it is better in at least one objective and no worse in the others where the sense of "better" and "no worse" depends on the optimization goal for each objective. For an objective to be minimized, better means "$<$", and no worse means "$\leq$". In the minimization example above, $A = (x_1, x_2, f_1, f_2) = (0.5, 0, 0.5, 2)$ dominates $B = (0.5, 1, 0.5, 3)$ and $C = (1, 1, 1, 2)$ but not $D = (1, 0, 1, 1)$. Also, D dominates C but not A or B. B and C do not dominate each other, A, or D. If a dominance-based fitness function, for example, tallies for each individual the number of other individuals in a generation that dominate it, then the tally can be used to identify individuals that best approximate the Pareto optimal front because their tallies will be zero [3]. Applied to A, B, C, and D, the tallies are 0, 1, 2, and 0, respectively. A and D are the best estimates of the Pareto optimal front and are in fact on the Pareto optimal front. The individuals on the Pareto optimal front are said to be non-dominated with respect to each other and dominate at least one other individual in the generation.

With fitness values calculated, the selection and reproduction process starts. First, individuals are chosen to form the mating pool through a competition mechanism. For example, in tournament selection, individuals are picked at random, and a comparison of fitness values decides the winner. Only competition winners are placed in the mating pool, and only members of the mating pool are selected to reproduce. Thus it is possible to lose good individuals either through losing a competition or failing to be selected to participate. Elitist strategies counter this by giving the best individuals an advantage specific to the algorithm. Strength Pareto Evolutionary Algorithm 2 (SPEA2) [3] creates an archive of past and present best individuals, and only individuals from the archive are considered for the mating pool.

Next, individuals are chosen pair-wise from the mating pool to produce offspring through crossover, also known as recombination. GAs typically operate on binary encoded string representations of the independent variables, while EAs operate on vectors of decimal values. In the GA form, crossover mirrors the biological process. The two parent strings are split at the same randomly selected locations, and the substrings are exchanged to make the offspring. In the EA equivalent, the values of the two parents are directly exchanged to create the offspring, or linear combinations of the parent values are passed to the offspring. Offspring mutation follows. For the binary string, mutation is a randomly flipped bit ($0 \rightarrow 1$ or $1 \rightarrow 0$), and a randomly applied and generated offset for the EA version. Probability density functions (pdfs) determine the behavior of the EA crossover and mutation. The pdf configuration parameters can be tuned to affect the convergence behavior.

Differential Evolution (DE) [5] is often used for SOOs in accelerator applications while SPEA2 and Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [4] are preferred for MOOs. SPEA2 and NSGA-II are elitist strategies, and now, an elitist DE is available for MOOs [6].

BEAMLINE COMPONENTS

Most beamline component GA- and EA-based design and optimization applications center on magnets and radio frequency (RF) and acceleration related devices. Early examples concentrate on magnet design and construction, and later ones are geared more toward acceleration systems.

Two magnet examples are wiggler and undulator magnet ordering and superconducting magnet design. Wiggler and undulator magnets are constructed from several smaller dipole magnets, and it is important to arrange the constituent magnets to minimize field errors. GAs and EAs can search through the many magnet order permutations and find the optimal arrangement [7], and GA-based magnet sorting and shimming continue to be integral to wiggler and undulator construction [8, 9, 10]. One of the earliest uses of Pareto optimality is a GA-based superconducting magnet design tool [11]. The tool identifies and creates multiple similar conceptual magnet designs meeting specified criteria that a magnet designer can then further evaluate with respect to manufacturing considerations. With this tool, a new magnet coil design was identified for use in the Large Hadron Collider.

The first application of GAs to RF devices was to optimize cavity designs based on higher order mode and resonance requirements [12]. With the renewed interest in GA and EA methods, the number of RF cavity optimizations has grown recently to include a variety of RF and superconducting RF (SRF) cavity designs: crab cavities [13], spoke cavities [14], choke-mode damped structures [15], microwave tubes [16], and klystron interaction structures [17]. Lastly, geometric dimension and relative dielectric permittivity optimizations for cylindrical and slab wakefield accelerators provide insights into the tradeoffs between transformer ratio and maximum accelerating wakefield electric field amplitude [18].

ACCELERATOR DESIGN

There are two modes of accelerator design using GAs and EAs. The first assumes a machine element layout where the field behavior of the magnets and accelerating devices is known, and the optimization searches for optimal set points, field amplitudes and phases, and element spacings to achieve specific beam quality and operating conditions. In other words, the optimization acts to tune the machine. The other mode builds on the first additionally varying physical features of beamline elements and systems to discern optimal designs or requirements for them; beamline element and operational set point optimizations are combined. Most applications fall in the first category, and with the growing appreciation for the power and flexibility of GAs and EAs in the accelerator field, the number of more holistic machine design applications is growing. Pareto optimality is especially important here because the front provides information about the many possible configurations and capabilities letting the designer choose the most suitable combination. In some cases, the front has even revealed unorthodox operating schemes that are comparable to or better than the standard solutions [19, 20, 21].

The ability to optimize a known machine layout is extremely useful. This form of optimization is well established, and only a fraction of the many examples will be highlighted here. The earliest application determines the quadrupole settings in injection and transfer lines for BESSY-II [22].

The tuning approach is used in mainly four ways. The most obvious use establishes the expected performance of a proposed or upgraded machine. A sample shows: positron momentum selection in a continuous wave positron source [23]; dynamic aperture optimization for a storage ring upgrade [24]; asymmetric distribution of longer insertion device straight sections in a storage ring light source [25]; a superconducting tape undulator-based light source [26]; a two-loop compact energy recovery linac (ERL) [27]; an electron linac for a gamma-ray source [28]; compression schemes for a free electron laser (FEL) [29]; and a bunch compressor for electron beam slicing [30].

Second, optimizing operational settings and relative element spacing can establish that a particular technology is suitable for a new or more demanding accelerator application. Most examples are direct current (DC) and RF gun injectors for FEL [31, 32, 33] and ERL light sources [34].

Third, accelerator physicists look for performance improvements for existing machines through alternative machine settings. Storage ring light sources benefit from this most to optimize brightness [20] and increase dynamic and momentum apertures [21, 35].

Lastly, this mode can identify tuning knobs or their settings to correct or avoid operational problems. Identification is needed for storage ring orbit bumps [36] and collider final focus optics tuners [37]. Finding knob settings is useful for controlling transport optics for muon experiments [38], restoring machine performance until failed components can be repaired [39], protecting equipment [40], and making machines insensitive to destructive beam instabilities such as beam break-up [41].

The second more encompassing accelerator design approach allows beam and beamline element performance to be considered together. This flips the model described above determining the suitability of a given technology or machine layout for an application. Instead the desired beam performance directly drives element design. Two advance examples of this approach are a preliminary spallation neutron source accumulator ring design study [42] and a muon solenoidal decay channel optimization [43].

The watershed accelerator physics applications showcasing the power of EAs, GAs, and Pareto optimality in general and this combined beam and element performance optimization are a laser profile and DC gun injector optimization for an ERL light source [19] and a damping ring optics evaluation tool that creates all viable optics layouts fitting in a given machine circumference [44]. They demonstrate that automating accelerator design is possible because GAs and EAs can manage the complexity of these problems and organize the results in a way to help the accelerator designer choose the best design among all possible designs meeting a set of criteria. The success and flexibility of these two optimizations led to the explosive growth in accelerator EA- and GA-based optimizations of all kinds. These two examples led to beam performance-based systems devoted to finding optimal: geometries for injector guns [45, 46] and accelerating elements [47, 48]; field profiles for storage ring dipoles [49] and fixed-field alternating gradient accelerator magnets [50]; undulator tapering profiles [51]; and an RF pulse time structure to compensate for beam loading

ISBN 978-3-95450-138-0

in a linac [52].

REAL TIME AND DIAGNOSTIC APPLICATIONS

Two very successful real time applications of GAs are photocathode laser profile generation [53, 54] and phase space reconstruction [55]. Both examples configure several similar or identical elements. The shear number of permutations on the settings precludes enumerating and evaluating each setting in turn, so a GA is used to sample the available permutations and identify an optimal configuration. A third example extracts cavity parameters from pass band data for an SRF cavity installed in a multi-cavity assembly [56].

ACCELERATOR DESIGN TOOLS AND COMPUTATIONAL CONSIDERATIONS

Execution time is an important factor to consider when choosing GA and EA tools and the underlying computational structure. These population-based methods assume computations for individuals are independent and self-contained, so they are amenable to parallelization. The maximum time to calculate all results for a single individual determines the level of necessary parallelization. If an individual is modeled with a few finite polynomial expressions for example, serially processing individuals in a population is reasonable, and optimization extensions for Excel [57], MATLAB, and Mathematica are appropriate to use. Most problems in accelerator physics require simulation codes to model the underlying physical system and involve non-negligible execution times. For these situations, the results for individuals must be produced in parallel.

Several parallel frameworks are available affording various levels of flexibility and abstraction. The basic design has two main parts. The first is the GA or EA wrapped around a simulation job management system. The job management system handles communication between the optimization processing and the simulation code(s), and it dispatches simulations for individuals to available computer resources. Program and script based systems include: Alternate PISA (A Platform and Programming Language Independent Interface for Search Algorithms [58]) (APISA) [19, 45] and variations from TRIUMF [59] and Jefferson Lab [46], geneticOptimizer [60], and OPT-PILOT [61]. APISA and variants mainly use SPEA2, and the others use NSGA-II. Note OPT-PILOT has a Python-based graphical interface for viewing Pareto optimal fronts and associated decision variable values.

The parallel frameworks commonly assume access to a large number of compute nodes found in batch farm and high performance computer cluster configurations. Alternatively, idle desktop computers in an organization can be co-opted with APISA or Condor [17], a system that coordinates, monitors, and dispatches simulations to underutilized desktop computers. The advent of commercial cloud computing resources may provide on-demand high performance computer clusters to extend existing systems or obviate the need for local installations [62].

Some accelerator simulation systems include GAs and EAs in their optimization suite: TAO [63], COSY-GO [64], G-optimizer in TRACK [65], GeneticTRACY [62], elegant COGA [66], and a symbolic beam propagator [67]. Lastly, libraries are available such as PGAPack [68] and pikaia [69] to build custom systems.

CONCLUSION

The use of GAs and EAs is well established in accelerator physics as demonstrated by the variety of applications and tools presented here. The expanded capability of GAs and EAs to directly solve MOOs and identify inherent trade-offs has substantially increased their utility in the field and enables automated optimization of systems that are complex in terms of dynamics and the number of variables involved. The basics of GAs and EAs have been introduced to facilitate understanding of how these algorithms work and interpreting the results they produce.

REFERENCES

[1] K. Deb. John Wiley and Sons, Chichester, 2001.

[2] A. Konak et al. *Reliability Engineering and System Safety*, 91(9):992–1007, 2006.

[3] Eckart Zitzler et al. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100, 2001.

[4] K. Deb et al. *IEEE Trans. Evolut. Comput.*, 6:182–197, 2002.

[5] R. Storn and K. Price. *J. Global Optimization*, 11:341–359, 1997.

[6] J. Qiang et al. IPAC 2013, Shanghai, China, May 2013, MOPWO062, pp. 1031-1033.

[7] R. Hajima et al. *Nucl. Instrum. Methods Phys. Res., Sect. A*, 318(1-3):822–824, 1992.

[8] F. Briquez et al. EPAC 2006, Edinburgh, Scotland, June 2006, THPLS118, pp. 3556-3558.

[9] J.-M. Ortega et al. IPAC 2012, New Orleans, LA, USA, May 2012, TUPPP048, pp. 1709-1711.

[10] S.C. Gottschalk et al. FEL 2012, Nara, Japan, August 2012, THPD11, pp. 567-570.

[11] S. Ramberger and S. Russenschuck. EPAC 1998, Stockholm, Sweden, June 1998, TUP10H, pp. 2014-2016 and TUP11H, pp. 2017-2019.

[12] A. Chincarini et al. *IEEE Transactions on Magnetics*, 31(3):1566–1569, 1995.

[13] C. Lingwood et al. IPAC 2010, Kyoto, Japan, May 2010, TUPEC056, pp. 1850-1852.

[14] M. Sawamura et al. SRF 2011, Chicago, IL, USA, July 2011, MOPO036, pp. 165-168.

[15] H. Zha et al. IPAC 2013, Shanghai, China, May 2013, TUPME026, pp. 1628-1630.

[16] S.G. Tantawi. FEL 2011, Shanghai, China, August 2011, TUOCI2_TALK.PDF.

ISBN 978-3-95450-138-0

[17] J.D.A. Smith et al. PAC 2009, Vancouver, BC, Canada, May 2009, TH5PFP046, pp. 3303-3305.

[18] F. Lemery et al. IPAC 2011, Kursaal, San Sebastian, Spain, September 2011, WEPZ008, pp. 2781-2783.

[19] I.V. Bazarov and C.K. Sinclair. *Phys. Rev. ST AB*, 8(3):034202, 2005.

[20] L. Yang et al. EPAC 2008, Genoa, Italy, June 2008, THPC033, pp. 3050-3052.

[21] M. Borland et al. PAC 2009, Vancouver, BC, Canada, May 2009, TH6PFP062, pp. 3850-3852.

[22] D. Schirmer et al. PAC 1995, Dallas, TX, USA, May 1995, WAQ17, pp. 1879-1881.

[23] S. Golge et al. PAC 2007, Albuquerque, NM, USA, June 2007, THPMS067, pp. 3133-3135.

[24] C. Sun et al. PAC 2011, New York, NY, USA, March 2011, TUODN4, pp. 793-795.

[25] V. Sajaev et al. PAC 2011, New York, NY, USA, March 2011, THP125, pp. 2354-2356.

[26] D. Filippetto et al. FEL 2011, Shanghai, China, August 2011, MOPC14, pp. 129-132.

[27] M. Shimada et al. IPAC 2011, Kursaal, San Sebastian, Spain, September 2011, WEOAA03, pp. 1909-1911.

[28] D. Alesini et al. IPAC 2011, Kursaal, San Sebastian, Spain, September 2011, TUPO008, pp. 1461-1463.

[29] C.H. Yi et al. IPAC 2012, New Orleans, LA, USA, May 2012, TUPPC027, pp. 1224-1226.

[30] A. He et al. IPAC 2013, Shanghai, China, May 2013, WEPWA091, pp. 2307-2309.

[31] F. E. Hannon and C. Hernandez-Garcia. EPAC 2006, Edinburgh, Scotland, June 2006, THPLS115, pp. 3550-3552.

[32] A. Bacci et al. FEL 2006, Berlin, Germany, August 2006, MOPPH030, pp. 99-102.

[33] P.N. Ostroumov et al. LINAC 2008, Victoria, BC, Canada, September 2008, TUP117, pp. 676-678.

[34] C.X. Wang. PAC 2009, Vancouver, BC, Canada, May 2009, MO6RFP048, pp. 467-469.

[35] Z. Duan and Q. Qin. IPAC 2011, Kursaal, San Sebastian, Spain, September 2011, THPZ011, pp. 3705-3707.

[36] S.L. Smith et al. EPAC 2004, Lucerne, Switzerland, July 2004, THPKF067, pp. 2418-2420.

[37] J.K. Jones. EPAC 2006, Edinburgh, Scotland, June 2006, MOPLS117, pp. 837-839.

[38] M. Apollonio. PAC 2009, Vancouver, BC, Canada, May 2009, TH6PFP057, pp. 3835-3837.

[39] B. Sun et al. IPAC 2013, Shanghai, China, May 2013, TUPWA018, pp. 1760-1762.

[40] H. Hao et al. IPAC 2013, Shanghai, China, May 2013, TUPWO061, pp. 2006-2008.

[41] R. Nagai et al. PAC 2003, Portland, OR, USA, May 2003, FPAB078, pp. 3443-3445.

[42] J.D. Galambos et al. PAC 1997, Vancouver, BC, Canada, May 1997, 9W023, pp. 1018-1020.

[43] S.J. Brooks. EPAC 2004, Lucerne, Switzerland, July 2004, MOPLT104, pp. 776-778.

[44] L. Emery. PAC 2005, Knoxville, TN, USA, May 2005, RPPP047, pp. 2962-2964.

[45] I.V. Bazarov et al. *Phys. Rev. ST AB*, 14:072001, 2011.

[46] A. Hofler et al. *Phys. Rev. ST AB*, 16:010101, 2013.

[47] V.A. Goryashko. FEL 2010, Malmö, Sweden, August 2010, MOPB13, pp. 71-74.

[48] A.V. Samoshin et al. RUPAC 2012, St. Petersburg, Russia, September 2012, WEPPC009, pp. 461-463.

[49] C.-X. Wang et al. IPAC 2010, Kyoto, Japan, May 2010, THPE039, pp. 4605-4607.

[50] S.J. Brooks. PAC 2009, Vancouver, BC, Canada, May 2009, FR5PFP025, pp. 4357-4359.

[51] J. Wu et al. IPAC 2012, New Orleans, LA, USA, May 2012, TUPPP082, pp. 1780-1782.

[52] O. Kononenko et al. LINAC 2010, Tsukuba, Japan, September 2010, MOP021, pp. 94-96.

[53] C. Vicario et al. EPAC 2004, Lucerne, Switzerland, July 2004, TUPLT063, pp. 1300-1302 and PAC 2005, Knoxville, TN, USA, May 2005, MOPB008, pp. 642-644.

[54] H. Tomizawa et al. LINAC 2004, Lübeck, Germany, August 2004, MOP80, pp. 207-209 and FEL 2007, Novosibirsk, Russia, August 2007, WEBAU01, pp. 298-305.

[55] A. Bacci et al. SPARC/EBD-07/004 (2007); FEL 2007, Novosibirsk, Russia, August 2007, WEAAU02, pp. 284-289; and EPAC 2008, Genoa, Italy, June 2008, TUPC089, pp. 1263-1265.

[56] N. Yi et al. *Nucl. Instrum. Methods Phys. Res., Sect. A*, 462:356–363, 2001.

[57] B. Mukherjee. Cyclotrons 2001, East Lansing, MI, USA, May 2001, P1-08, pp. 21-23.

[58] S. Bleuler et al. In *Evolutionary Multi-Criterion Optimization (EMO 2003)*, pages 494–508. Springer, 2003.

[59] C. Gong and Y.-C. Chao. ICAP 2012, Rostock-Warnemünde, Germany, August 2012, TUABI1, pp. 67-71.

[60] M. Borland and H. Shang. Unpublished program (2007).

[61] A. Adelmann et al. ICAP 2012, Rostock-Warnemünde, Germany, August 2012, TUAAI2, pp. 62-66.

[62] C. Sun et al. PAC 2011, New York, NY, USA, March 2011, WEP151, pp. 1767-1769 and IPAC 2012, New Orleans, LA, USA, May 2012, MOPPC086, pp. 340-342.

[63] D. Sagan and J.C. Smith. PAC 2005, Knoxville, TN, USA, May 2005, FPAT085, pp. 4159-4161.

[64] K. Makino and M. Berz. ICAP 2006, Chamonix Mont-Blanc, France, October 2006, TUAPMP01_TALK.PDF.

[65] B. Mustapha and P.N. Ostroumov. ICAP 2009, San Francisco, CA, USA, August 2009, FR1IOPK01, pp. 245-250.

[66] M. Borland. APS LS-287, 2000.

[67] S.N. Andrianov et al. IPAC 2011, Kursaal, San Sebastian, Spain, September 2011, WEPC115, pp. 2280-2282.

[68] D. Levine. ANL-95/18 (1996).

[69] P. Charbonneau and B. Knapp. NCAR/TN-418+IA, National Center for Atmospheric Research, Boulder, Colorado, (1995) `http://www.hao.ucar.edu/modeling/pikaia/pikaia.php`

ISBN 978-3-95450-138-0