

GENETIC ALGORITHMS WITH NICHING

S. Ramberger

CERN, 1211 Geneva 23, Switzerland

Abstract

This chapter describes the use of genetic algorithms with the concept of niching for the conceptual design of superconducting magnets for the Large Hadron Collider, LHC at CERN. The method provides the designer with a number of possible solutions which can then be further optimized for field quality and manufacturability.

1 The conceptual design phase

A particularity of the conceptual design phase is the “fuzziness” of the objectives and constraints. The main objectives are a small content of unwanted field errors in the main dipole field, low sensitivity of the field quality to manufacturing errors, easy manufacturing, and possibility to tune the geometry after the pre-series manufacture of the magnets. The “fuzziness” in these objectives is due to the fact that it is difficult to mathematically formulate all objectives.

- The amount of unwanted multipoles is expressed by the coefficients of the Fourier series expansion of the radial field component in the aperture where in the assumed symmetric case only the odd b_n (coefficients of the sine terms) are to be minimized. However, using goal programming methods the weight for the components showing different sensitivity has to be found in an iterative procedure, as the effects of the components have to be examined using beam tracking.
- As the electro-magnetic forces are enormous (about 4000 kN/m radially) the local force distribution in the coil collar structure has to be optimized. However, this requires computations of coupled electro-magnetic mechanical problems.
- Manufacturing considerations include ease of the coil winding and collaring, i.e. geometrical constraints on the pole angle. Setting too many geometrical constraints results in ill-conditioned optimization problems.
- Requirements concerning the protection of the superconducting coils include low inductance, and a small current density in the copper stabilizer at a quench. These requirements are partly contradictory as with fewer turns the operating current increases. The calculation of peak voltage and temperature in the coils during a quench therefore requires network analysis and heat propagation calculations.
- During manufacturing, systematic errors occur due to the applied tooling. After the pre-series construction of the magnets the coils have to be repositioned to compensate for these systematic errors. It will be impossible to change the topology of the coil which therefore has to have sufficient flexibility for adjustments.

An optimization method is therefore required that not only converges towards a “global” optimum but provides the user with a number of proposals (local minima) which can then be “post-processed” using different methods and tools.

2 The optimization problem

The design variables for the optimization problem are the number of turns per coil block \vec{n} , the positioning and inclination angles of the blocks \vec{X} , and the current I in each turn. The current has to be included as a design variable in the optimization in order to guarantee an operation point not higher than 95 % on the load-line of the superconducting wires, which depends on the local magnetic field. The magnetization of the iron yoke with an inner radius of 98.5 mm is calculated by means of a reduced vector potential finite-element formulation [4], [5] which does not require the meshing of the coil, as the source terms can easily be calculated using Biot-Savart's law. With a relatively sparse mesh with 1725 nodes and quadrilateral iso-parametric elements the computing time for the nonlinear problem is about 20 seconds for each function evaluation on a 333 MHz DEC alpha station. The fact that the coils don't have to be meshed is important because the topology of the coil is changed during the optimization process by omitting and adding conductors to coil blocks. The objective function reads: $\min \left\{ t_0 p(QM - 0.05) + t_1 \frac{1}{B_1^2} + \sum_{n=3,5,\dots}^{11} t_n \frac{B_n^2}{B_1^2} \right\}$ where the coefficients t_i fit the sensitivities of the components, and the penalty function $p(x) = x^2$ for $x < 0$ guarantees a quench-margin QM of at least 5%.

3 Genetic algorithms

For the minimization of the resulting objective function genetic algorithms are used. The first steps to the development of genetic algorithms were set in the fifties when models for biological processes were sought. Though at that time not apt for function optimization the potential of adaption was recognized and was evaluated by putting the models on a firm basis by Holland [1]. Since then the research on those procedures meanwhile called genetic algorithms boosted. [2, 3] Increasing computing capabilities nowadays allow for complex problem solving and global optimization in multidimensional function spaces.

Since standard genetic algorithms work on the bit-level an encoding for the parameters is necessary. Real variables have to be quantized into integers. Then all integers are Grey-encoded to provide for better behavior of the genetic algorithm. Put on this basis, genetic algorithms do not have to care about the properties of the parameters anymore and can directly operate on their encoding as bit-strings.



3.1 Quantization

Quantization is achieved by linear sampling of the parameter range. The parameter mapping is defined by giving lower and upper bounds, as well as the range for the integers. In the current approach the full integer range of n bits with 2^n values is used. The quantization has to be chosen fine enough as to limit the quantization errors. The chromosomes are then found by Grey-encoding these integers and packing them together into a bit-string where the least number of necessary bits is used.

3.2 Grey-coding

Though the integers as found by the quantization could be immediately used as a parameter representation, a further coding step was introduced improving the quality of the optimization process. According to the following table (shown in Fig. 3.2 for only 4 bits) each bit combination is uniquely transformed into another. This scheme is called Grey-coding.

	binary	→	Grey		binary	→	Grey
0	0000		0000	8	1000		1100
1	0001		0001	9	1001		1101
2	0010		0011	10	1010		1111
3	0011		0010	11	1011		1110
4	0100		0110	12	1100		1010
5	0101		0111	13	1101		1011
6	0110		0101	14	1110		1001
7	0111		0100	15	1111		1000

The coding is achieved by a bit-shift right and exclusive-or of the unshifted and the shifted version of the binary code. Decoding proceeds by multiply shifting right and exclusive-or operations on the shifted versions.

Example: 13:	coding		decoding
	1101		1011
	<u>0110</u>	eor	0101 eor
	1011		0010 eor
			<u>0001</u> eor
			1101

The advantage of Grey-codes is because chromosomes close to each other in parameter space show similar patterns with higher probability. For example the codes of decimal 3 and 4, 011 and 100 in a binary representation differ in 3 bits, but 010 and 110 as Grey-encoded differ only in one bit. The other way round this means that mutation produces nearby strings with higher probability.

Nevertheless the percentage of close strings after a one bit change is dependent on the overall number of bits n since normally only two strings are close in Grey-codes and only one in binary codes, but all the rest of bit changes is further distant as for example 0000 and 1000 is 0 and 8 in binary or even 0 and 15 in Grey-codes.

3.3 Operators

Genetic algorithms are driven by 3 main operators, selection, crossover and mutation. The selection operator guarantees convergence to an optimum by keeping the better chromosomes and discarding the less fit ones. A standard method is fair-wheel selection where chromosomes are chosen proportional to their fitness value. Since this mechanism is sensible to the quality of the fitness function the simpler alternative of retaining the better half of the population may give better results. Diversity is reduced generation by generation thus leading to a single solution.

Crossover is a recombination of bit strings of two chromosomes by swapping the strings at random points. This is the major influence directing the search process to good solutions. A recombination of

good parameters of any chromosome with good parameters of another should lead to a better set [1]. The type of crossover which is normally considered is single-point crossover. Two chromosomes are taken out of the gene-pool. A certain splitting point is determined. Each chromosomes is cut at this point and recombined with the part of the other chromosome.

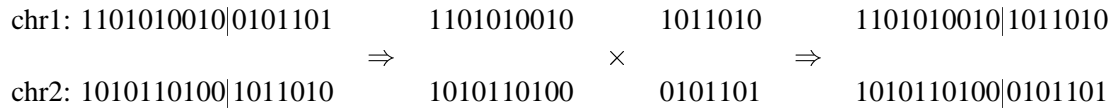


Fig. 1: Single-point crossover of two chromosomes

As shown in Fig. 3.3 the bits on the lefthandside of chromosome 1 are connected to the bits on the righthandside of chromosome 2 and vice versa. This exchange results in two probably new chromosomes which resemble two parameter sets that are a combination of two different earlier parameter sets. Taking a closer look on the effects of the operator on the parameter sets itself we see as well that cutting the chromosomes introduces some randomness regarding the very parameter that may be split by the operation. This effect is owing to the bit representation of the parameter sets. Implementation on a parameter level can avoid this random behavior but since randomness is introduced by other operators on purpose, there is not much reason to do so.

Illustrating single point crossover in a pseudo 4-dimensional lattice (Fig. 2) shows that such an operator produces offspring on one of two trajectories whereas two point crossover covers additional points (here: 0000 and 1011). Nevertheless the larger space of possible offspring due to 2 crossover points can be equally produced by two successive iterations using single point crossover. This certifies on the bit-level that the offsprings are not fully random.

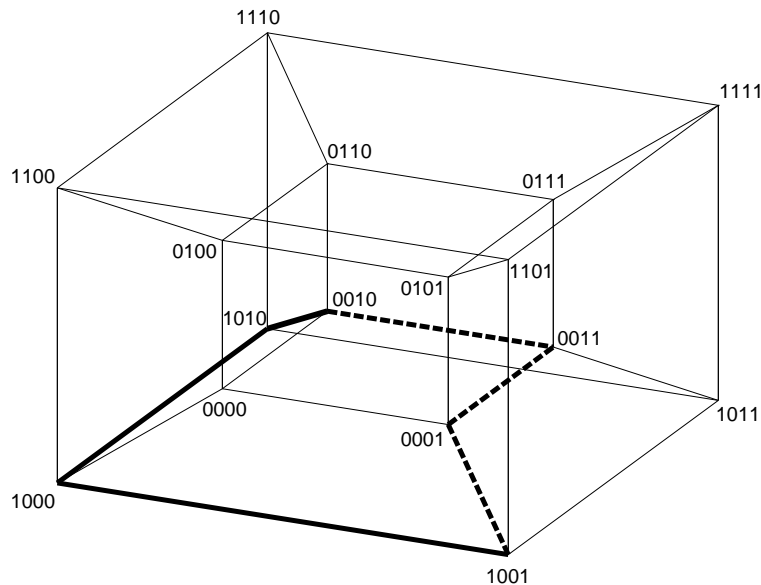


Fig. 2: Hypercube with transition patterns for single point crossover of 0010 and 1001

After selection and crossover, genes are chosen for mutation. The effect of the mutation operator is twofold. First it avoids preliminary convergence of the entire population towards a local minimum and

secondly it improves local convergence by a hill-climbing like mechanism. Though both mechanisms seem to be contradictory, they result from the different significance of bits in the bit-string.

3.4 Niching

Though most shortcomings of using deterministic algorithms are already overcome by the standard procedure, a few problems persist:

- Because of the different sensitivity of the multiple objectives in the different sub-domains of the feasible domain the solution might not represent the designer’s preference, in particular when utility functions are used.
- Not all of the geometrical constraints and manufacturing considerations for coil winding and col-laring can be taken into account.

We therefore apply the concept of niching that provides the designer with a set of solutions rather than one solution which can then be “post-processed”. Fig.3 shows a comparison of the classical (Royal Road) genetic algorithm with the applied method of niching. Niching genetic algorithms differ in the selection process where for each offspring the chromosome with the smallest hamming-distance $HD = \sum_i u_i \oplus v_i$ (least number of different bits) is located and selected if its fitness is worse than that of the offspring. Whereas in the Royal Road genetic algorithm the whole population is subject to a fitness ranking, the selection in the niching genetic algorithm is performed on the level of each individuum. Selected chromosomes are then immediately joined to the population. The effect of the niching method is that **a number** of local optima are found which can then be further investigated and compared.

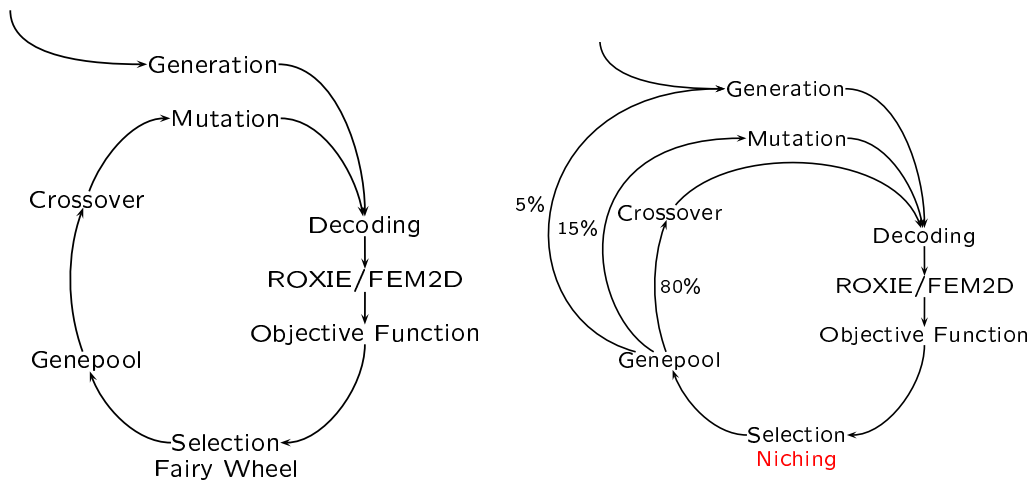


Fig. 3: Left: Royal road genetic algorithm, right: Genetic algorithm with niching

To increase the diversity in the beginning of the optimization, a fourth operator is added, i.e. generating new chromosomes in a random process. Normally this operator is only used for initialization of the genetic algorithm with random chromosomes.

After each application of an operator the offspring is evaluated and selected strings are introduced into the chromosome pool for immediate participation.

4 Parameters

For the optimization of the 2D-sections the following parameters were found to be effective: The current in the conductors was encoded by a 5 bit string and the angles of the coil-blocks by 5 bits each.

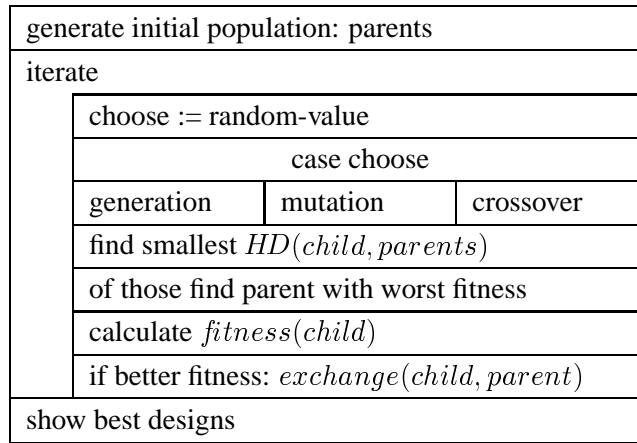


Fig. 4: Genetic algorithm with niching

The number of turns of the outer and inner blocks were encoded by 4 bit strings each, thus resulting in chromosomes of typically 50 to 60 bits. A new offspring is generated by crossover with a rate of 0.8 or mutation with a rate of 0.15. Note that this rate is related to the chromosome, i.e. the bit mutation probability is about 0.003. Generation of new chromosomes is done with a probability of 0.05. A population size of 60 chromosomes is found to be sufficient, which determines also the number of final solutions.

Those parameters, crossover, mutation, and generation rate, were chosen after a number of tests changing mutation rate, type of crossover and population size. A representative convergence graph is shown in Fig. 5a. After increasing mutation to 60% as shown in Fig. 5b the convergence is lower than for our standard genetic algorithm in the beginning. After a few thousand iterations when the population becomes stable, higher mutation rate allows for further improvement since changes of single bits often correspond to small changes in parameters. Results for single and 2-point crossover show no significant differences.

In a third test, the population size was changed to 150 individuals thereby increasing the diversity in the chromosome pool from which offspring is created (Fig. 5c). The number of necessary function evaluations is increased, resulting in better refined final solutions. Though higher quality is achieved, the longer runtime is hardly acceptable (about 4 days on a DEC alpha station). As in both cases each solution has to be further optimized by local methods, the advantage of better starting solutions vanishes.

5 Results

Two 6-block coil designs were found using genetic algorithms and were studied in detail. They are shown in Fig. 7 and 8. The VY version cross-section is shown in Fig. 6. The V6-1 design has a B_{ss} which is about 0.1 T higher than in the VY version. This is remarkable as it can be achieved with 1 turn less. The explanation is the reduced peak-field to main-field ratio in the inner layer. At the same time the margin in the outer layer blocks is reduced with respect to the VY version but is still higher than for the inner layer. The b_{11} is considerably reduced.

6 Conclusion

Genetic algorithms with niching can be efficiently used for the conceptual design of superconducting magnets. Although the convergence of the algorithms are in no way comparable to deterministic methods, two different designs were found which turned out to have advantages. The best coil, as far as

margin and sensitivity to tolerances are concerned, is the V6-1 version. The advantages compared to the VY version are that one turn less results in a more economical design. Further, 0.12 T more central field at quench results in an increased margin to quench due to the reduced peak field to main field ratio in block No. 6. The design also features lower inductance, smaller persistent current effects, smaller b_{11} component, smaller electromagnetic forces parallel to the broad sides on turn 40 and 41, less sensitivity to random errors on wedges and conductor dimensions, and a better tunability. The 6-block coil also has a more homogeneous force distribution resulting in less shear-stress on the wedge between block 5 and 6 compared to the big wedge between block 4 and 5 in the 5 block coil version. The disadvantages are that one additional copper wedge in the straight section and one additional end spacer has to be fitted into the coil. The margin to quench of the outer layer is reduced, although the limiting factor for the short sample current is still the inner layer.

REFERENCES

- [1] J. H. Holland, *Adaption in natural and artificial systems*. MIT Press, 1992.
- [2] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.
- [3] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag Berlin Heidelberg, 3rd, revised and extended ed., 1996.
- [4] C. Paul, K. Preis, and S. Russenschuck, "Saturation induced field errors in the LHC main dipoles," LHC-project note 84, CERN, 1997.
- [5] K. Preis, I. Bardi, O. Biro, C. Magele, W. Renhart, K. R. Richter, and G. Vrisk, "Numerical analysis of 3D magnetostatic fields," *IEEE Trans. Magn.*, vol. 27, pp. 3798–3803, Sept. 1991.

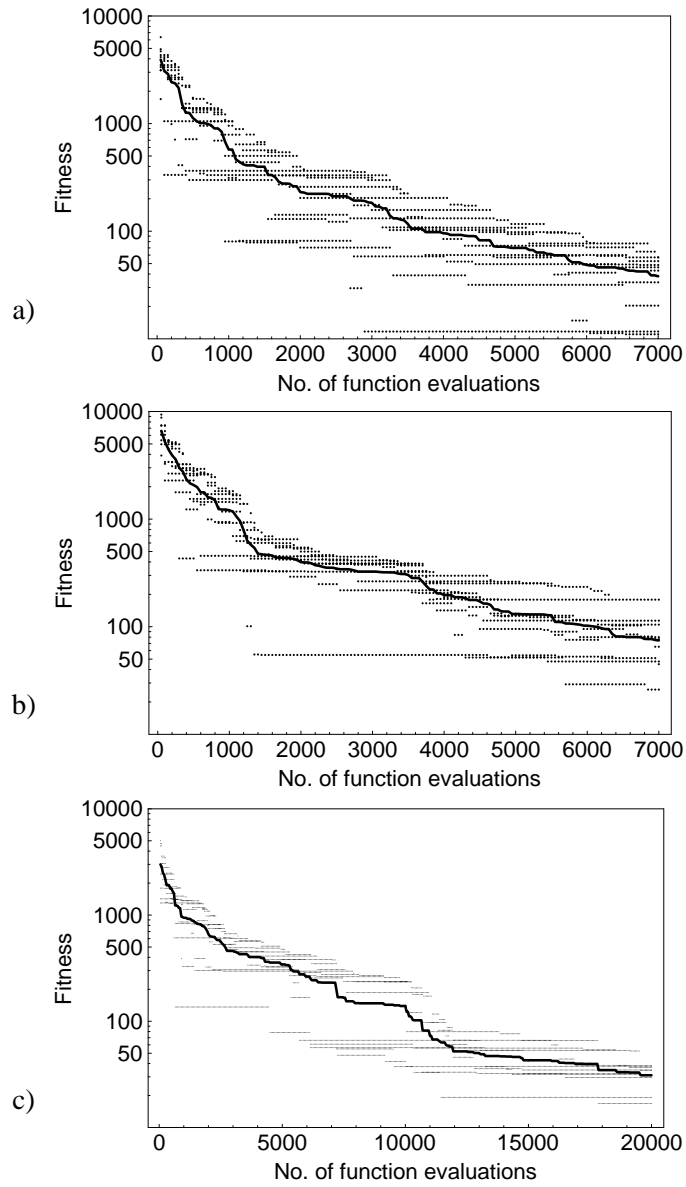


Fig. 5: Fitness of the 10 best chromosomes and its average fitness (continuous line), a) Convergence for standard parameters; crossover rate 0.8, mutation rate 0.15, generation rate 0.05, b) Convergence for mutation rate 0.60, crossover rate 0.35 c) Increased population size of 150 individuals

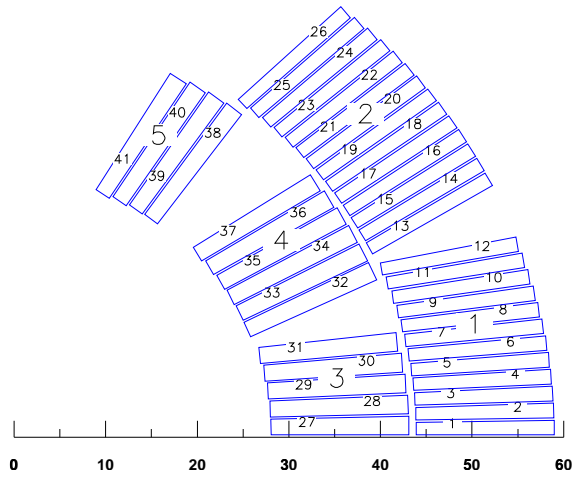


Fig. 6: Coil cross-section for the 5-block (41 turns) design (VY)

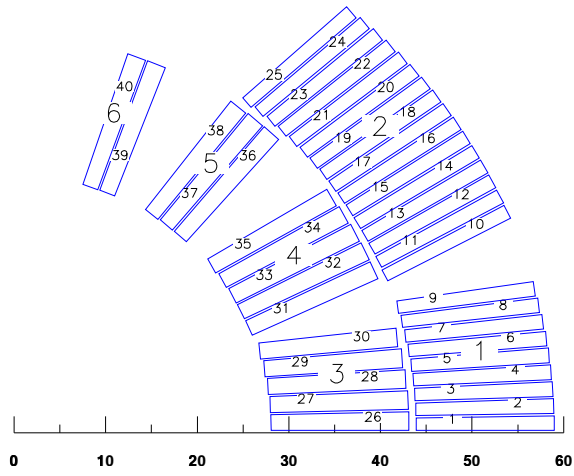


Fig. 7: Coil cross-section for the 6-block (40 turns) design (V6-1)

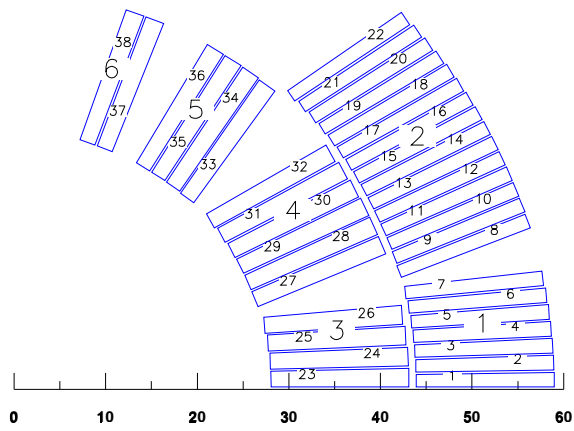


Fig. 8: Coil cross-section for the 6-block (38 turns) design (V6-3)