Stellarator optimization using metaheuristics

A. Gómez-Iglesias[1], F. Castejón[1], M. A. Vega-Rodríguez[2]
[1] *Euratom-CIEMAT Fusion Association, Madrid, Spain*
[2] *University of Extremadura, Cáceres, Spain*

Introduction: Metaheuristics for stellarator Optimization

The search for an optimised configuration that fulfils multiple optimization criteria is a key topic in the path to the stellarator reactor. The investigation of the relevance of the optimization criteria is a fundamental step to propose such a stellarator-based configuration for a reactor. There are different optimized stellarators configurations following several criteria like diminishing the neoclassical transport or improving the stability. The next stellarator W7-X, under construction in IPP-Greifswald, Germany, as well as QPS and NCSX designed in USA, are examples of optimised magnetic configurations.

The reduction of neoclassical transport, the confinement of fast particles, based on the omnigeneity property, the Mercier stability criterion and the ballooning stability are the main topics that are considered in the stabilization process. Once the optimized configuration is found, the necessary coils to create it are evaluated in order to make an engineering assessment of their complexity. All these steps are included in the code Stellop (see [1] as an example of optimization with such a code), based on the Levenberg-Marquardt algorithm. Stellop looks for the optimal configuration in the huge phase space taking into account the optimization functions cyclically. Nevertheless, it is desirable to have available other searching procedure to be able to compare the results with Stellop, since the phase space is huge and it is mandatory to determine to what extent the found configuration depends on the searching algorithm. Therefore, it would be desirable to benchmark the results of Stellop with other calculations.

In this work, we develop an algorithm based on metaheuristics to look for such an optimised configuration. A metaheuristic is a combinatorial optimization process that tries to maximize or minimize a function defined by the user, which is called objective function. All the metaheuristics concepts have some commonalities like the exploration of the phase space or the use of a predefined number of candidate solutions. As a first step to test the algorithm, we have introduced two optimization criteria: the minimization of the neoclassical transport by reducing the average sum of the Bxgrad(B) drift and the Mercier criterion. The first one implies the minimization of the following objective function:

$$F_{objective\ function} = \sum_{i=1}^{N} \left\langle \left| \vec{B} \times \nabla B / B^3 \right| \right\rangle_i \qquad (1)$$

In this equation, i is the magnetic surface label and ***B*** is the magnetic field, tangent to the magnetic surface. The equilibrium is calculated using the VMEC code [2], whose last version includes the calculation of the Mercier stability criterion, which allows us to introduce this criterion is in our optimization process, thus having two target functions presently. Mercier criterion has been introduced in a way that if a given configuration does not satisfy the condition established by the user, the configuration is rejected regardless of the value given by Eq. (1). In the same way, the user can specify a desired value for β or for any other parameter given by VMEC. Moreover, the ballooning stability code COBRA [3] will be introduced to include this criterion in the optimization process.

Distributed Asynchronous Bees Algorithm

DAB algorithm is a metaheuristic process designed to solve large scale computational problems using distributed environments. It evolves a population consisting of a set of individuals, where each individual represents a candidate solution for the given problem. In this case, an individual is an equilibrium configuration calculated with VMEC. The algorithm explores the solution space defined by the Fourier modes describing the plasma. Due to the fact that all the individuals are independent, grid computing techniques are suitable for this process. Many metaheuristics, as our case, are based on two processes:

- Exploration: this process explores the solution space, which has to be done in such a way that all the main areas of the solution space should be well-balanced explored (high dispersion). In our case, this balance is assured by creating a set of individuals, obtaining their distances to the previously selected individuals and selecting the individual with the higher distance. Therefore a distance must be defined in the solution space as follows:

$$d(\vec{s},\vec{q}) = \sum_{j=1}^{n} \left| s_j - q_j \right| \qquad (2),$$

where ***s*** is the current candidate solution, ***q*** is each of the solutions previously found, j is the variable index and n is the total number of parameters that define a solution (the Fourier modes in our case).

- Exploitation: this process introduces convergence in the algorithm when good configurations have been found. In this case, due to the grid paradigm, we divide this process into two more: I) Mutation based exploitation: following the behaviour of evolution in nature, by mutation or crossover of chromosomes (each of the Fourier modes). The mutation uses the previously good candidate solutions found. The number of chromosomes to be modified depends on a configurable mutation rate, which in fact limits the phase space to explore. II) Local search exploitation: the best configuration found so far receives more computational resources and performs local searches using small modifications over just a few chromosomes. To prevent the best configuration to always get more computational resources, a probability value is assigned to each candidate solution previously found following eq. (3), where the fitness value of all the solutions found when exploiting a given solution is used.

$$p_k = fit_k \Big/ \sum_{l=1}^{n} fit_l \qquad (3)$$

These two processes run on the grid, following a master-slave model with a decentralized intelligent behaviour where the information is distributed among all the resources involved in the optimization process.

Results

We have used this algorithm for TJ-II optimization. In order not to be far from the actual TJ-II configuration, we have allowed the input Fourier modes that describe the plasma boundary to vary ±15%. We have started from the 42_100_68 configuration, which is characterised by its indentation that makes it suitable for a flux expansion divertor development [4] and by its shallow magnetic well, which reduces the Mercier β limit to 0.3%. This configuration is taken as input for the algorithm and our final results present a diminishing of the neoclassical target function given by Eq. 1 of 22% and the equilibrium exists up to β=20% while it is Mercier-stable up to β=0.65%. Fig. 1 shows the cut of the magnetic surfaces both for the starting (left) and the optimised (right) configurations. The main differences between the two configurations are the shape and the magnetic well, deeper in the optimised case. Allowing an arbitrary variation of the Fourier modes one can obtain much different configurations like the two-period stellarator shown in Fig 2.

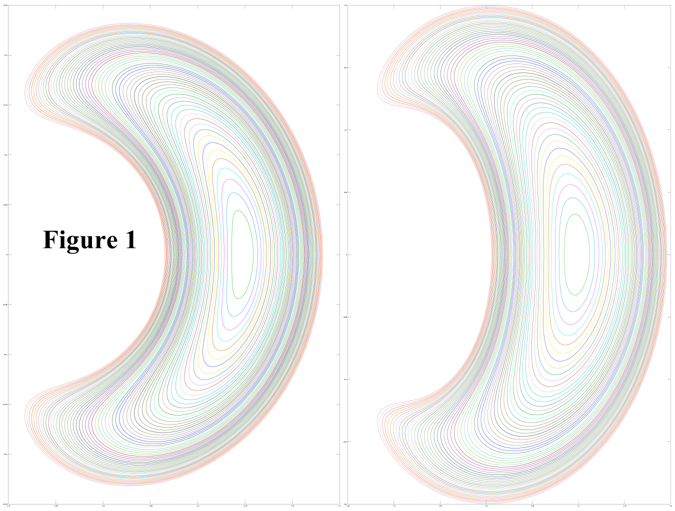

Figure 1

Figure 2

Acknowledgements

This work has been performed under EUFORIA project, which is funded through the Research Infrastructures initiative of the 7th Framework Programme of the European Commission, grant agreement Number 211804.

[1] A.A. Subbotin et al. Nucl. Fusion 46 (2006) 921

[2] S. P. Hirshman and J. C. Whitson. Physics of Fluis 26 (1983) 3553

[3] R. Sanchez, S.P. Hirshman, J.C. Whitson, and A.S. Ware. Journal of Computational Physics 2 (2000) 161

[4] F. Castejón et al. Nuclear Fusion 49 (2009) 085019