

Optimization of Supercontinuum Spectrum Using Genetic Algorithms on Service-Oriented Grids

G. Moltó¹, M. Arevalillo-Herráez², C. Milián¹, M. Zacarés⁴, V. Hernández¹, A. Ferrando⁴

¹ Instituto ITACA

Universidad Politécnica de Valencia
Camino de Vera S/N, 46022 Valencia, Spain
[gmolto,carmien,vhernand]@itaca.upv.es

² Departamento de Informática
Universidad de Valencia

Avda. Vicente Andrés Estellés, S/N, 46100 Burjassot, Valencia, Spain
miguel.arevalillo@uv.es

³ Departament d'Òptica
Universitat de València

Dr. Moliner, 50, 46100 Burjassot, Valencia, Spain
albert.ferrando@uv.es

⁴ Grupo de Modelización Multidisciplinar Intertech
Universidad Politécnica de Valencia
Camino de Vera S/N, 46022 Valencia, Spain
mzacares@upvnet.upv.es

Abstract. This paper describes the developments made towards the optimization of photonic crystal fibers using automated procedures based on Genetic Algorithms. The large amount of computations involved during the optimization process demand the usage of Grid technologies that are able to cope with these computational requirements. This paper describes the integration of a service-oriented Grid metascheduler and a Genetic Algorithm in order to create an automated tool which is able to optimize this kind of fibers. This work also provides an evaluation of these computational techniques for the optimization of the output supercontinuum (SC) spectrum generated in an optical fiber of a fixed length and fixed dispersion properties. The developed tool provides an scalable solution to this computationally intensive problem, and enable the optimization of other more complex designs.

1 Introduction

The generation of optical homogeneous spectra covering a range from the near infrared (IR) to the ultra violet (UV) is known as Supercontinuum (SC) generation. This phenomenon is usually observed in photonic crystal fibers and it has been a subject of intense study in the field of temporal nonlinear optics for the last decade. Nowadays there is a deep understanding of the underlying physics associated to this process [1, 2], opening a scope for engineering problem proposals.

Since its discovery lots of applications have been found in optical metrology, pulse compression and white light sources generation, among many others. Our interest here focuses on nonlinear microscopy of biological samples. To excite certain transition of a given protein or any other biological molecule, this technique needs very specific wavelengths which, in general, can't be efficiently provided by any standard source. Hence, the possibility of maximizing the spectral output in an arbitrarily chosen spectral window is extremely useful.

This kind of applications can be mathematically stated as an optimization problem that consists of determining the set of configuration parameters which yields the best results. Unfortunately, the evaluation of each combination of parameters is computationally expensive and the exploration of the entire space is an unfeasible option. In this context, the use of meta-heuristics such as genetic algorithms are an alternative to approximate a global optimum.

Current Grid computing technologies [3] use standard protocols for sharing both computational power and data storage capacity among geographically distant resources. Therefore, they are an ideal infrastructure for the execution of the high throughput problem that lies beneath the optimization via Genetic Algorithms.

The remainder of the paper is structured as follows: First, section 2 describes the process of designing a crystal fiber. Next, section 3 focuses on the Genetic Algorithm and the underlying strategy for the optimization procedure. Later, section 4 describes the service-oriented metascheduler and its integration with the Genetic Algorithm. Then, section 5 applies the optimization strategy to an specific case study in order to assess its effectiveness, showing the main results achieved. Finally, section 6 concludes the paper pointing out some future research work.

2 The Optimization of Supercontinuum Spectrum

When an optical pulse propagates down an optical fiber with sufficiently high power, the material becomes nonlinear and many different such effects start playing a critical role in the dynamics. In particular, if a single pulse is launched in the anomalous dispersion regime of the fiber (where higher frequencies travel faster) above certain power threshold, it breaks up into several stable pulses which propagate with a well defined shape. These pulses are called fundamental solitons (solitary waves) and the breaking up process is referred to as soliton fission. The number of ejected fundamental solitons (N) is given by the initial pulse parameters and is computed as $N = T_0 \sqrt{\frac{\gamma P_0}{|\beta_2|}}$. Here, T_0 is a typical time related to the full width at half maximum (FWHM) of the pulse intensity by $FWHM = 2 \ln(1 + \sqrt{2}) T_0$, γ is the nonlinear coefficient of the material, P_0 is the initial peak power and β_2 defines the group velocity dispersion (GVD) of the fiber ($\beta_2 > 0$ for normal GVD and $\beta_2 < 0$ for anomalous GVD). Each one of these pulses, resulting from the fission process, travels with a different carrier frequency and radiates at lower wavelengths (dispersive waves) propagated in the normal GVD regime of the fiber. These effects all together result in SC generation [1, 2]. The SC can be modeled by solving the following equation for the complex electric field $A(z, t)$:

$$-i\partial_z A(z, t) + \frac{1}{2}\partial_t^2 A(z, t) = -i\epsilon\partial_t^3 A(z, t) + A(z, t) \int_{-\infty}^{\infty} dt' R(t') |A(z, t - t')|^2 \quad (1)$$

where $\epsilon = \frac{\beta_3}{6T_0^2|\beta_2|}$ and $R(t)$ is an empirical function for the material response. In this work we have optimized the spectral output of a 1 cm long fiber in the range of [1780, 1820] THz ([1035, 1060] nm in vacuum). The nonlinear coefficient used in the simulations is given by $\gamma = 0.11(Wm)^{-1}$ and the dispersion parameters are $\beta_2(\lambda = 780nm) = -11.83ps^2/Km$ and $\beta_3(\lambda = 780nm) = 0.08ps^3/Km$. The initial peak power is expressed in terms of these quantities, $P_0 = (1 + 63\alpha) \frac{|\beta_2|}{\gamma T_0^2}$ and the initial conditions (T_0), initial wavelength (λ_0) and peak power tunability (α) are the parameters to vary in order to optimize the simulation results.

The evolution program is a matlab code that solves Eq. (1) using the fourth order Runge-Kutta method for the z-derivative and the fast Fourier transform (FFT) and its inverse (IFFT) for the rest of the operations.

3 The Genetic Algorithm

Let $d : R^{m \times n} \times R^{m \times n} \rightarrow R$ be a function which determines the distance between two spectra and let s_0 (defined in $R^{m \times n}$) be the reference (target) spectrum. Under this setting, the problem can be formulated as an optimization problem in which the aim is to find the vector \mathbf{x} which minimizes the value of an objective function $f : R^3 \rightarrow R$, defined as $f(\mathbf{x}) = d(s(\mathbf{x}), s_0)$. Besides, the three parameters are limited to the intervals [0.01, 0.05], [700, 900] and [0, 1] respectively. In summary, the problem can be expressed as shown in Eq. 2.

$$\operatorname{argmin}_{\mathbf{x}: T_0 \in [0.01, 0.05], \lambda_0 \in [700, 900], \alpha \in [0, 1]} d(s(\mathbf{x}), s_0) \quad (2)$$

Although minimization problems can be solved by using deterministic and metaheuristic algorithms, the latter are required when exhaustive enumeration of the search space is not possible. For that, a shorter runtime is achieved by accepting solutions which approximate a global optima, but may not exactly match it.

Genetic Algorithms (GA) are a class of evolutionary algorithms, which place a special emphasis on the application of genetic operators, such as mutation and crossover. A GA uses an encoding method to represent potential solutions to the problem, and a measure that allows a quantitative evaluation of each candidate, called a fitness function.

Starting from an initial population, a GA uses the fitness function to evaluate each candidate. The most promising individuals are allowed to reproduce and determine the next generation of individuals, according to a series of pre-established evolution rules. The evolution rules refer to the parent selection method, and the definition of the mutation and cross-over operations to obtain the offspring for the next generation. Most parent selection methods are stochastic in order to keep the diversity of the population, preventing premature convergence to a sub-optimal solution.

3.1 Computational issues

In this particular case, no prior knowledge about the search space is available, and a rugged landscape is expected. Indeed, there exist efficient evolutionary algorithms for general purpose applications. As an example, CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy) [4, 5] is an evolutionary algorithm for difficult non-linear non-convex optimization problems in rugged search spaces.

However, we face a problem in which the fitness function has a high and varying computational cost (ranging from minutes to several hours). Under these circumstances, the parallelization of generational algorithms in a Grid may not be an efficient approach. Instead, a steady state genetic algorithm has been used, changing one member of the population at a time. This allows that several fitness values are computed in parallel (after an initial population has been built), and processed once they are available. To this end, a “replace the worst” strategy has been adopted. In this context, the use of a steady state algorithm fully exploits the processing power of the Grid, keeping it constantly computing solutions until a maximum number of evaluations is reached. In its current state, the algorithm uses a number of general parameters, namely:

- The approximate number of computing elements in the Grid (n). This is used to determine how many individual evaluations are processed in parallel.
- The population size (p).
- The maximum number of evaluations to perform (MAX). Once MAX evaluations have been performed, the result with the highest fitness value is returned.
- The number of individuals which are generated before any genetic operator is applied (c). The algorithm is initiated with c random seeds before any genetic operator (selection, crossover and mutation) is applied.

Figure 1 provides a flowchart of the algorithm. Every time a potential solution is completed, this is added to population (if the current population size contains less than p elements), replaces another individual (if its fitness is better than that of the worst individual) or is simply discarded (otherwise). Then a new offspring is generated and sent to the Grid for evaluation. Although steady state algorithms do not explore the search space as well as other generational GAs, its effectiveness has shown sufficient for our specific purposes.

The steady state GA has been implemented combining the service-oriented GMarteGS metacheduler [6] and the JMetal library [7]. GMarteGS provides meta-scheduling functionality for the concurrent execution of parallel applications on resources based on the Globus Toolkit, the standard software in computational Grids. JMetal is an object-oriented framework (also written in Java) aimed at the development, experimentation, and study of metaheuristics for solving optimization problems.

A Graphical User Interface (GUI) has also been built to allow the user to manipulate some of the parameters provided by JMetal. This interface constitutes a first step towards the standardization of the approach, providing a general mechanism to run the algorithm to solve other problems which may benefit from the

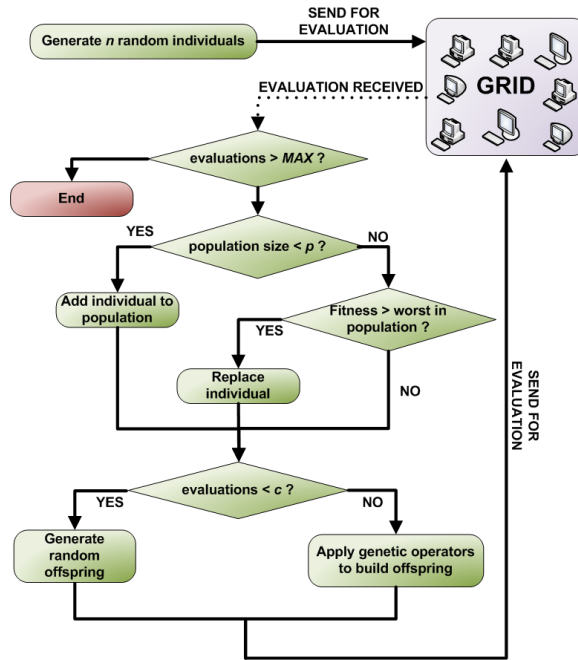


Fig. 1. Steady state event driven approach used.

use of heuristic approaches in a Grid. As both the interface and the GA have been developed in Java, we can use the Java Web Start technology to automatically deploy the application to the client with minimal requirements (a web browser with Java support). This allows the application to be easily accessible from practically any platform.

4 Computational Support for the Genetic Algorithm: A Grid Computing Approach

The GA requires that candidates solutions (individuals) are evaluated. Each evaluation typically requires an execution time that may be in the order of minutes or even hours, depending on the parameter values. The amount of combinations of parameters evaluated has an important impact on the ability of the GA to find an optimum solution. Typically, it is required to evaluate thousands of individuals. Clearly, this cost exceeds the computational capabilities of a single machine.

If a steady state genetic algorithm is used, each individual can be evaluated independently of the others. Therefore, this turns out into a high throughput problem which can be easily addressed in parallel, for example using clusters of PCs. In particular, we have used an integrated computational approach, using clusters of PCs within a Grid infrastructure. The usage of Grid protocols to support these executions makes it possible to provide the GA with an scalable solution to its

demanding computational requirements. If additional computational resources are required, the Grid infrastructure can be enlarged and the applications employed to submit the executions remain unchanged, thus reducing development efforts.

The following subsections briefly describe the GMarteGS service-oriented meta-scheduler employed to support the evaluation of individuals.

4.1 GMarteGS: A Service-Oriented Metascheduler

GMarteGS [6] is a service-oriented façade to the GMarte [8] metascheduler. GMarte enables the reliable execution of high performance computing applications on Grid infrastructures based on the Globus Toolkit (GT) [3]. It manages the execution of the tasks by providing fault-tolerant data transfer services between the client machines and the remote computational resources. It also monitors the application during its lifetime to detect failures, which are silently handled by re-scheduling the tasks on other available resources.

GMarteGS represents the evolution of GMarte into a service-oriented architecture. It is a Grid service entirely developed in Java using the Globus Toolkit 4 and common standards like the Web Services Definition Language (WSDL) and the Web Service Resource Framework (WSRF). It introduces secure multi-user resource brokering on computational Grids for the execution of computationally intensive applications. Its architecture allows different users to simultaneously use its functionality to submit the execution of tasks. In addition, the Grid service uses the Grid Security Infrastructure (GSI) provided by GT to achieve privacy between the client and the service, as the principal communications are ciphered, as well as data integrity.

Thus, GMarteGS acts as a mediator between the users and the Grid infrastructure. As such, it provides a high level Application Programming Interface (API), both in WSDL and in Java, which can be easily employed to incorporate its functionality to existing programs requiring the execution of tasks in a Grid infrastructure.

Figure 2 depicts the interaction between the genetic algorithm and the Grid service. First of all, the GA creates a session and then it starts the optimization strategy according to Fig. 1. When the GA requires an individual to be evaluated, it uses the GMarteGS Client API to define a new execution task with the appropriate parameters that describe the individual. This task is submitted to the Grid infrastructure and its state can be easily monitored via high level methods that describe the state of the task (i.e. scheduled, staged in, active, staged out, completed or failed).

4.2 Interfacing the Genetic Algorithm with GMarteGS

GMarteGS exposes its functionality via WSDL. Therefore, it can be accessed from virtually every possible application, regardless of the programming language used. However, as the Genetic Algorithm has also been implemented in Java, we have developed a lightweight client-side Java API to ease the usage of GMarteGS from a client Java code. This includes methods and objects for defining tasks, creating

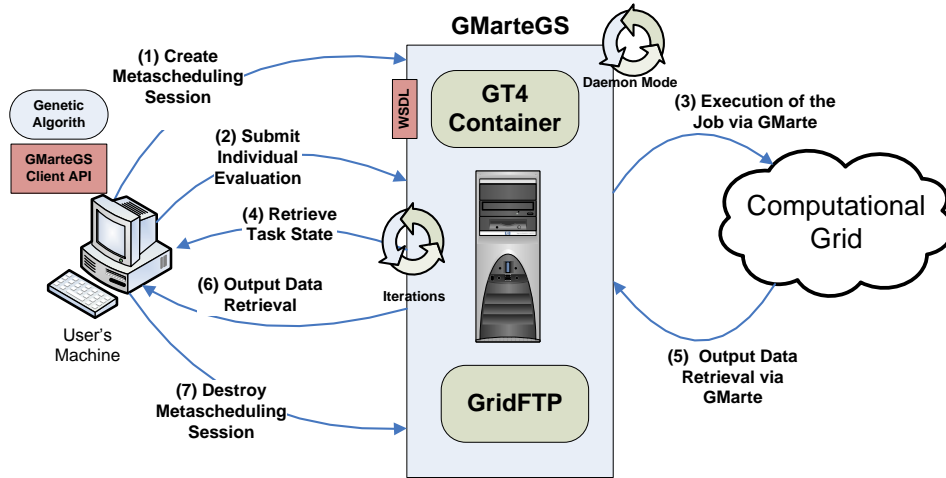


Fig. 2. Interaction diagram between the genetic algorithm and GMarteGS.

sessions, gathering the state of the metascheduling sessions and their tasks, destroying sessions and transferring data between the client machine (where the GA runs) and the Grid service machine via the GridFTP protocol.

Gathering the state of the tasks is usually done actively (the client requests this information from the Grid service). However, GMarteGS also supports the standard notification mechanisms available in GT version 4 so that the client is notified whenever the state of a session (i.e., the state of any of its tasks) has changed. The main problem is that supporting notifications by the client requires that it deploys a GT4 container (at least the Java WS Core) to receive the incoming connections. This can be a drawback, as the client requires a special firewall configuration and can no longer be considered a thin client.

The application that evaluates the individuals has been implemented in Matlab. Although it is possible to create a redistributable application which only depends on the Matlab runtime, this last piece of software is heavyweight and should be transferred to the remote computational resource along with the application. Therefore, to avoid this overhead we have relied on the installation of the required Matlab libraries in the remote computational resources. Once we have assessed the effectiveness of the optimization codes, we plan to develop a C-based version which can certainly be statically-linked to minimize the runtime dependencies and be able to seamlessly run on a wide variety of machines in a Grid infrastructure.

5 Case Study: Evaluating the Genetic Algorithm

In the experiments, we have used a population size of 50, and evaluated a total of 200 candidate solutions to the problem. To achieve sufficient diversity amongst the initial population, the first potential solutions were generated randomly (in the intervals specified in Section 3). In particular, we have used SBX (Simulated Binary Crossover) [9] and a mutation rate of one mutation per individual, a common setting that has been reported to yield good results in a wide diversity of problems [10].

We have used a small Grid infrastructure which uses the Globus Toolkit 4.0.8 and is composed of two clusters of PCs (20 and 55 nodes) running on Intel Xeon 2.8 Ghz with 2.0 GBytes of RAM. A total of 12 such nodes have been used in this particular work.

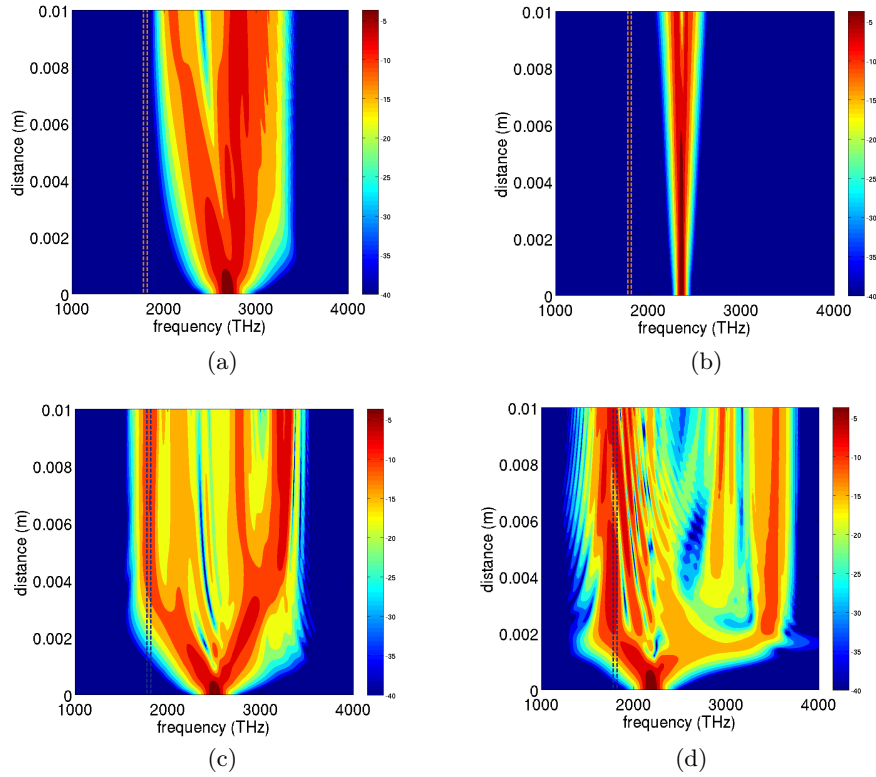


Fig. 3. Spectra evolution along a 1 cm fiber. In the upper row examples of evolution show two different possibilities in which the spectrum does not fall near the frequency range established (between the dashed lines). Figures in lower row show two solutions selected by the genetic algorithm, which maximizes the spectrum in the range specified by $\nu \in [1780, 1820] THz$.

Figure 3 shows the spectral evolution (distance vs optical frequency) inside the 1 cm long fiber whose parameters were specified in section 2. The spectral intensity is expressed in the colorbars in dB units (using a logarithmic scale) and the black dashed vertical lines delimit the spectral zone where we want the maximum possible light power at the output (distance = 0.01 m). To show the diversity of solutions in the range of the parameters used in this paper, we have included figures 3.a and 3.b. These two are not the result of the optimization algorithm. Instead, they show the evolution with the selected parameters ((a) $[T_0 = 0.01, \lambda_0 = 700, \alpha = 0.5]$, (b) $[T_0 = 0.03, \lambda_0 = 800, \alpha = 0.5]$) for which the spectral output has no contribution in the specified frequency range. However, the GA has found the set of parameters that correspond to Fig. 3.c $[T_0 = 0.0106, \lambda_0 = 751.84, \alpha = 0.95]$ and Fig. 3.d $[T_0 = 0.0105, \lambda_0 = 860.34, \alpha = 0.61]$ where an important fraction of the final spectrum falls into the desired range.

It is worth noting that comparing both sets, two out of the three parameters substantially differ from each other. Therefore the GA is always capable to detect different regions of parameters that maximizes the spectrum after less than 200 runs. This feature permits to reduce the complexity of the problem by focusing only on the interest regions. It is worth mentioning that some of these regions contain suboptimal solutions and from the physical point of view they are also of great interest. Once the local minima have been detected a more accurate search around them combining GA strategies and other optimization techniques must be performed in order to find the global minimum.

Therefore, the usage of a Grid-based approach enables to process more candidate individuals per time unit, and thus has a direct impact in the ability of the GA to obtain an optimum solution. In particular, the execution of this case study required 90 minutes in the Grid infrastructure considered. Running the same GA in a single PC requires a total execution time of over 10 hours.

6 Conclusion

This paper has presented the development of an automatic tool to optimize photonic crystal fibers based on Genetic Algorithms and supported by the computational power of a Grid infrastructure.

In this case, we have used a simple genetic algorithm to find the best combination of parameters. We believe that combining global optimization techniques with other local optimization strategies may considerably improve the results for certain classes of problems. Although we have used the software to solve a specific problem, the intention is to build a library that allows the user to easily solve other optimization problems using Grid computing. The extension of the software to support a wide range of other optimization techniques is currently under development.

The results presented in this paper are the simplest possible approach towards a full optimization process not predictable, in principle, just on the basis of physical grounds. The full problem involves new degrees of freedom that would include other pulse parameters (e.g, chirp) and several fibers with different lengths and fiber dispersion coefficients (β_2 and β_3). The results obtained in this case are

promising and we plan to use (and possibly extend) this architecture to face other more complex and general optimization problems.

Acknowledgements

The authors wish to thank the financial support received from the Spanish Ministry of Science and Technology to develop the project ngGrid: New Generation Components for the Efficient Exploitation of eScience Infrastructures (TIN2006-12890). This work has been partially supported by the Structural Funds of the European Regional Development Fund (ERDF). CM, MAH and MZ would also like to thank the Generalitat Valenciana for the project GV2008/032. CM gratefully thanks the *Formación de Profesorado Universitario* predoctoral grant.

References

1. A. V. Gorbach and D. V. Skryabin. Light trapping in gravity-like potentials and expansion of supercontinuum spectra in photonic-crystal fibres. *Nature Photonics*, 1:653–657, 2007.
2. J. M. Dudley, G. Genty, and S. Coen. Supercontinuum generation in photonic crystal fiber. *Reviews of Modern Physics*, 78:1135–1184, 2006.
3. I. Foster. Globus toolkit version 4: Software for service-oriented systems. *IFIP International Conference on Network and Parallel Computing, Lecture Notes in Computer Science*, 3779:2–13, 2005.
4. N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.
5. N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.
6. G. Moltó, V. Hernández, and J. M. Alonso. A service-oriented WSRF-based architecture for metascheduling on computational grids. *Future Generation Computer Systems (International Journal of Grid Computing)*, 24(4):317–328, 2008.
7. Juan J. Durillo, Antonio J. Nebro, Francisco Luna, Bernabé Dorronsoro, and Enrique Alba. jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical Report ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, December 2006.
8. J. M. Alonso, V. Hernández, and G. Moltó. GMarte: Grid middleware to abstract remote task execution. *Concurrency and Computation: Practice and Experience*, 18(15):2021–2036, 2006.
9. K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *complex systems*, 9:115–148, 1995.
10. Thomas Bäck. Optimal mutation rates in genetic search. In Morgan Kaufmann, editor, *Proceeding of the Fifth International Conference on Genetic Algorithms*, San Mateo, CA, 1993.