Evaluating Evolutionary Multiobjective Algorithms for the *in silico* Optimization of Mutant Strains

Paulo Maia, Isabel Rocha, Eugénio C. Ferreira and Miguel Rocha

Abstract— In Metabolic Engineering, the identification of genetic manipulations that lead to mutant strains able to produce a given compound of interest is a promising, while still complex process. Evolutionary Algorithms (EAs) have been a successful approach for tackling the underlying *in silico* optimization problems. The most common task is to solve a bi-level optimization problem, where the strain that maximizes the production of some compound is sought, while trying to keep the organism viable (maximizing biomass). In this work, this task is viewed as a multiobjective optimization problem and an approach based on multiobjective EAs is proposed. The algorithms are validated with a real world case study that uses *E. coli* to produce succinic acid. The results obtained are quite promising when compared to the available single objective algorithms.

Keywords: Multiobjective Evolutionary Algorithms, Metabolic Engineering, Flux-Balance Analysis, Systems Biology.

I. INTRODUCTION

Maximizing production has always been one of the top goals in any industrial environment, while other concerns such as environmental costs have been gaining importance in the last few years. In this context, biotechnological processes are increasingly used in the production of a number of valuable products, such as pharmaceuticals, fuels or food ingredients, replacing traditional chemical synthesis. The Metabolic Engineering arena has provided promising tools to select genetic modifications capable of achieving improved production of the desired products [1][2].

This effort has relied on the development of techniques for simulating genome-scale models of organisms, that recently suffered major improvements (although they are still incomplete), enabling researchers with tools to simulate *in silico* hundreds or thousands of mutant strains of certain organisms. Algorithms such as Flux Balance Analysis (FBA) [3] or Minimization of Metabolic Adjustment (MOMA) [4] were proposed and quickly became popular.

A bi-level optimization problem can then be formulated, by adding a layer that searches for the best mutant that can be obtained by simply deleting a few genes from the wild type. The idea is to force the microorganisms to produce the desired product by selected gene deletions and simultaneously keep the organism viable. The underlying optimization problem consists in reaching an optimal subset of gene deletions to maximize an objective function related with the production of a given compound (e.g. yield or productivity). This function typically depends on the values of two fluxes: the desired product and the biomass or growth.

Several different algorithms have been proposed to address this problem, namely mixed integer linear programming (MILP) [5] and more recently stochastic meta-heuristics, such as Evolutionary Algorithms (EAs) [6][7] and Simulated Annealing (SA) [8]. Although these approaches have provided good results, they all share a common drawback: they provide only one single optimal (or near optimal) solution to the problem. In many situations, a set of solutions with different trade-offs between the production of the desired compound and the biomass production would be desirable.

In this work, an approach based on Multi-Objective Evolutionary Algorithms (MOEAs) is proposed to this problem. In fact, since the mid-1980's, MOEAs are being used to solve all kinds of multiple-criterion problems in distinct scenarios and the multiobjective nature of the problem suggests that this is a good candidate for MOEAs. The MOEAs chosen for this task are two of the most popular algorithms, namely the SPEA2 and the NSGA-II, widely accepted as two of the algorithms with best overall performance.

The MOEAs are validated using a case study that considers the production of succinic acid, using *E. coli* where a genome scale metabolic model is available. The results are compared to previous work [7][8] regarding the same case study, where single objective EAs and Simulated Annealing (SA) approaches have been proposed. The results obtained are quite promising, since the MOEAs were able to find in a single run, a set of trade-offs between the two optimization aims that could only be reached by single-objective algorithms with several runs varying a threshold parameter.

II. SIMULATION ALGORITHMS FOR THE PREDICTION OF METABOLIC BEHAVIOR

The reconstruction of genome-scale metabolic networks has been one of the most common applications of the annotated microorganism's genomes. Through the annotation and mathematical definition of the reactions obtained it is possible, to some degree, to simulate the phenotypic behaviour of these microorganisms. One approach is to write dynamic mass balances for each metabolite in the network, generating a set of ordinary differential equations that may be used to simulate the dynamic behavior of metabolite

P. Maia and M.Rocha are with the Department of Informatics /CCTC, University of Minho, 4710-057 Braga, Portugal {paulo.maia,mrocha}@di.uminho.pt

I.Rocha and E.C. Ferreira are with IBB - Institute for Biotechnology and Bioengineering, Center of Biological Engineering, University of Minho, 4710-470 Braga, Portugal {irocha,ecferreira}@deb.uminho.pt

concentrations. However, there is still insufficient data on kinetic expressions and parameters [9].

Therefore, a steady state approximation is generally applied, where for each metabolite in the network, the sum of all productions and consumptions will be zero, weighted by the stoichiometric coefficients. Thus, for metabolite i, where $i = 1, \ldots, M$ (M is the number of metabolites) the following constraint is defined:

$$\sum_{j=1}^{N} S_{ij} v_j = 0 \qquad (1)$$

where S_{ij} is the stoichiometric coefficient for metabolite i in reaction j and v_j is the reaction rate or flux over the reaction j. It is possible to define a matrix S, composed of the S_{ij} values, $j = 1, \ldots, N$ (N is the number of reactions); v is the N-dimensional vector of the fluxes of the reactions.

The mass balances are therefore reduced to a set of linear homogeneous equations. The maximum/minimum values of the fluxes can be set by constraints in the form $\alpha_j \leq v_j \leq \beta_j$, that are used to specify both thermodynamic and environmental conditions (e.g. availability of nutrients).

For most metabolic networks, since the number of fluxes is greater than the number of metabolites, the set of linear equations obtained from the application of Equation 1 to the M metabolites usually leads to an under-determined system, for which there exists an infinite number of feasible flux distributions. However, if a given linear function over the fluxes is chosen to be maximized, it is possible to obtain a single solution by applying standard algorithms (e.g. *simplex*) for linear programming problems. This methodology is known as Flux Balance Analysis (FBA) [3].

The combination of this technique with validated genome-scale stoichiometric models [10] allows to simulate the phenotype of a microorganism under defined environmental conditions without performing any experiments. The most common flux chosen for maximization is the biomass, based on the premise that microorganisms have maximized their growth along natural evolution, a premise that has been confirmed experimentally in some cases [11].

III. EVOLUTIONARY MULTIOBJECTIVE OPTIMIZATION

A. Multiobjective Optimization

In the real world, the full complexity of many problems can not be reduced to the matter of a single objective. By defining multiple objectives, it is possible to describe more adequately all the features of a given problem. In the last two decades, multiobjective optimization applications have been increasing. In a single-objective problem, the search space is often well defined, but when several conflicting objectives arise, the simultaneous optimization of all the aims becomes more complex. In this case, there is no longer a single optimal solution but rather a set with equivalent, or at least non comparable quality.

The classical definition of a multiobjective optimization problem can be written in the form $minimize[f_1(x), f_2(x), ..., f_n(x)]$ for k objective functions

$f_i : R^n \rightarrow R$ subject to several several equality and inequality constraints. For $x = [x_1, x_2, ..., x_n]^T$, which is our vector of decision variables, the task is to determine from the set F of vectors which satisfy all the constraints, the particular set of values $[x_1^*, x_2^*, ..., x_n^*]$ that also yields the optimum values for all the objective functions [12].

In multiobjective optimization problems, it is very rare to find a solution that simultaneously optimizes all the objective functions. Therefore, what we usually look for is the optimal set of trade-offs rather than a single solution. This clearly changes the concept of *optimality*. Nowadays, the adopted concept of optimality is that proposed by Edgeworth [13] and later generalized by Pareto [14]. This is known as the *Edgeworth-Pareto optimality*.

The definition of dominance is put forward in this context: when comparing two solutions, one can dominate the other or the two can be non comparable. A solution a dominates another solution b, if a is not worse than b in any of the objectives and it is at least better in one of them. An optimal solution is one that is not dominated by any other in the search space. This kind of solutions are called *Pareto-optimal* and the entire set of such optimal solutions is called *Pareto-optimal set*. The image of the *Pareto optimal set* under the objective functions is called the *Pareto-front* [15]. More formally, a vector of decision variables $x^* \in F$ is called *Pareto optimal* if there is no other $x \in F$ such that $f_i(x) < f_i(x^*)$ for all $i = 1, ..., k$ and $f_j(x) < f_j(x^*)$ for at least one j.

B. Why use EAs in multiobjective problems?

The populational nature of EAs, whose main features are illustrated in Figure 1, provides the algorithm with a set of solution candidates and the reproduction process creates new solutions from older ones. This process enables the algorithm with the capability of finding several possible members of the *Pareto-front* in a single run. The selection process decides which individuals of the current population take part of the new generation.
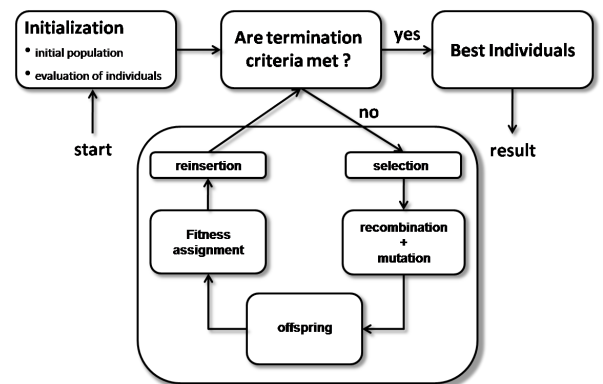

Fig. 1. Graphical representation of the generic working procedure of an Evolutionary Algorithm (EA).

MOEAs can generate and keep sets of solutions that are all Pareto optimal. The first incursions in developing MOEAs

has been performed by Schaffer in 1985 [16], soon followed by several other approaches that have been developing more specialized EAs with better performance in these tasks. One advantage of using EAs is the few constraints imposed over the objective function. In fact, they are agnostic to the shape or continuity of the Pareto-front, they are easy to implement, very robust and can even be used in a parallel computing environment. A general idea of MOEAs and the recent history of the field can be found in [12] and [17]

IV. THE PROPOSED APPROACH

A. Algorithms

In this work, two well known MOEAs were tested: the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [18] and the Non-Dominated Sorting Genetic Algortithm II (NSGA-II) [19]. Both are last generation MOEAs, incorporating improvements over previous versions. A graphical representation of their structure is given in Figure 2.

a) SPEA 2: The SPEA2 algorithm is an evolution of the previous SPEA [20] algorithm. The main improvements include: (1) a better fitness assignment strategy that takes into account the number of individuals that each individual dominates and is dominated by, (2) the use of a nearest neighbour density estimation technique to guide the search more efficiently and (3) an archive truncation technique to help prevent the loss of boundary solutions.

b) NSGA-II: The NSGA-II is also an evolution of the previous NSGA [21] algorithm. In contrast with the previous firt-generation algorithm, this one uses elitism. Also, a crowding comparison operator was introduced to help keep the diversity of the population. NSGA-II is not archive-based, its elitism mechanism combines the best parents with the best offspring obtained.

B. Representation scheme and mutation operator

The problem addressed in this work consists in selecting, from a set of genes in a microbe's genome, a subset to be deleted in order to maximize a set of objective functions related to the microorganism's metabolism. The encoding of a solution is achieved by a set-based representation, where only gene deletions are represented. Each solution consists of a set of integer values representing the genes that will be deleted. Therefore, if the value i is in the set, this means the i-th gene in the microbe's genome is knocked out. Each value in the set is an integer with a value between 1 and N.

The representation scheme used is based on variable-sized genomes, hence sets with distinct cardinalities can be encoded and compete in the search process. Two types of reproduction operators were used: crossover and mutation. The only crossover operator used was inspired on *uniform crossover* and works as follows: the genes that are present in both parent sets are kept in both offspring; the genes that are present in only one of the parents are sent to one of the offspring, selected randomly with equal probabilities. Regarding mutation, three operators were used:

- *Random* mutation, that replaces a gene in the set by a random value in the allowed range.
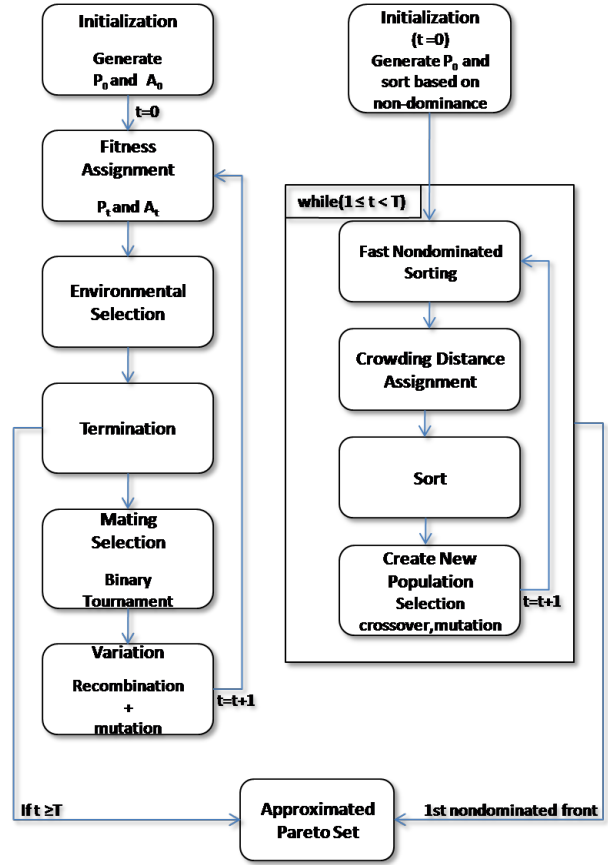

Fig. 2. Simplified graphical representation of the working procedure of the SPEA2 (left) and NSGA-II (right) algorithms.

- *Grow*: consists of the introduction of a new gene into the chromosome, whose value is randomly generated in the available range (avoiding duplicates in the set).
- *Shrink*: a randomly selected gene is removed from the genome.

C. Decoding and evaluating

The principle considered is a correspondence between the values in the set and metabolic reactions, i.e., each value represented in the set represents a particular enzyme that catalyzes a metabolic reaction. That enzyme is associated with a particular gene (or genes) that should be deleted for that reaction to be eliminated. The decoding process works by taking each value in the set and forcing the flux it indexes to the value 0, therefore disabling that reaction from the metabolic model.

The process proceeds with the simulation of the mutant using FBA. The output is the set of values for the fluxes of all reactions, that are then used to compute the fitness value, given by an appropriate objective function. One possible objective function is the Biomass-Product Coupled Yield (BPCY) [6], given by $BPCY = \frac{PG}{S}$, where P stands for the flux representing the excreted product; G for the organism's growth rate (biomass flux) and S for the substrate intake flux. Besides optimizing for the production of the desired

product, this function also allows to select for mutants that exhibit high growth rates, i.e., that are likely to exhibit a higher productivity.

An alternative, is to maximize only the value of the product's flux (P), but imposing a minimum threshold to the value of the biomass (G_{min}). Therefore, a distinct objective function, denoted as Product Flux with Minimum Biomass (PFMB), will be defined, where solutions that obtain the minimum value of G are given as fitness the value of P; otherwise, if the minimum threshold is not obtained the fitness is 0. By varying the value of G_{min} a set of different trade-offs between P and G can be obtained.

When dealing with multi-objective problems, we seek for solutions maximizing two different objective functions - ($OF1$) the product and ($OF2$) the biomass:

- **Objective 1:** *maximize* **P**
- **Objective 2:** *maximize* **G**

D. Initialization

The initial solution is a set with randomly generated elements. In this variable size variant, the size of the individual is randomly created in the range 1 to 12. The same process is used in all algorithms to initialize each individual in the population.

E. Pre-processing

In genome-scale models the search space is usually large enough to cause problems to the optimization algorithms. Thus, in order to reduce the search space, some operations were conducted. These are described in detail in [8].

F. Implementation issues

The implementation of the proposed algorithms was performed by the authors in the *Java* programming language. This was based on a previously developed platform for general purpose use of EAs in several optimization tasks. This strategy allowed a general purpose implementation of MOEAs and the comparison with single objective EAs using the same representation and reproduction operators. The implementation of the MOEAs also used components from the jMetal [22] implementation. In the implementation of FBA, the *GNU linear programming package (GLPK)*[1] was used to run the *simplex* algorithm.

V. Experiments

A. Case study

One case study was used to test the aforementioned algorithms that considers the microorganism *Escherichia coli* and the aim is to produce succinic acid (succinate). Succinate is one of the key intermediates in cellular metabolism and therefore an important case study for metabolic engineering [23]. The knockout solutions that lead to an improved phenotype regarding its production are not straightforward to identify since they involve a large number of interacting reactions. Succinic acid and its derivatives have been used as

[1]http://www.gnu.org/software/glpk/

common chemicals to synthesize polymers, as additives and flavoring agents in foods, supplements for pharmaceuticals, or surfactants. Currently, it is produced through petrochemical processes that can be expensive and have significant environmental impacts.

The genome-scale model for this microorganism used in the simulations was developed by Reed et al [24]. This model considers the *E. coli* metabolic network, including a total of $N = 1075$ fluxes and $M = 761$ metabolites. After the pre-processing stages, the simplified model remains with $N = 550$ and $M = 332$ metabolites. Furthermore, 227 essential genes are identified, which leaves 323 variables to be considered by the optimization algorithms. The proposed algorithms were compared with the SA proposed in [8] and with the EAs proposed in [7]. In both cases, the termination criteria was defined based on a maximum of 50000 fitness evaluations. For each experimental setup, the process was repeated for 30 runs and the mean and 95% confidence intervals were calculated. The SPEA2 archive was set to 100 as well as the population for the NSGA-II.

B. Results

The proposed MOEAs were benchmarked against the previous single objective algorithms. In the single objective approaches, the PFMB objective function was used, because it is more easily comparable with the results obtained from the multiobjective algorithms. When several runs with different values of G_{min} are executed, these results become prone to be represented as Pareto-like curves as shown in Figure 3. In order to assemble the Pareto fronts for the single objective alternatives, several values of G_{min} have to be considered. For each algorithm (EA and SA), 30 Pareto sets were created, each with 11 points that are the result of choosing different values for G_{min} (0% to 100% of the wild type biomass, in 10% steps).

On the other hand, the MOEAs are a more natural approach to this kind of problem. In order to build the 30 Pareto sets for each MOEA, only 30 runs for each (NSGA-II/SPEA2) were necessary and 100 points are obtained in each run.

One dificulty that arises in the comparison is the fact that the optimal Pareto set is not known *a priori*. So, it was necessary to generate a reference set that could approximate the optimal set, given the solutions found by all the algorithms. This approximated reference set was achieved by aggregation of all the solutions found by the SPEA2/NSGA-II and all the best solutions found by the EA/SA. A filtering step was conducted *a posteriori* keeping only the non-dominated and non-repeated solutions. The final reference set contains 774 solutions, resulting in our approximation of the Pareto Optimal Solution Set (P). Figure 3 shows the dominance relation between P and the non-dominated solutions found for the EA and SA algorithms.

Although the comparison between pareto fronts is neither an easy nor a straightforward task, some metrics have been developed enabling to better evaluate the quality of the solutions and compare the performance of the algorithms. The
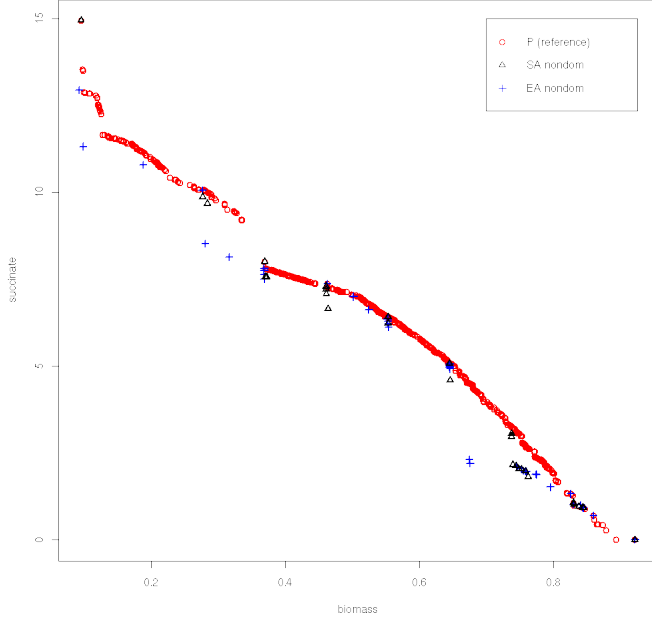
Fig. 3. Non-dominated solutions found for the EA (blue crosses), SA (black triangles) and the aggregated Pareto Optimal Solution Set (P)

definition of quality in multi-objective optimization is quite different, since there are a number of optimization goals: the distance of the resulting set to the Pareto-optimal front should be minimized; a good distribution of the solutions along the front should be obtained; and, the spread of the front should be maximized, i.e., for each objective, a wide range of values should be covered. In this work, three well known Performance Indexes (PIs) were used:

- **Generational Distance (GD)** [25]: This metric calculates how far, on average, the experimental pareto front (PF_{known}) is from P. It is mathematically defined by:

$$GD \triangleq \frac{(\sum_{i=1}^{n} d_i^p)^{\frac{1}{p}}}{n} \qquad (2)$$

where n is the number of vectors in PF_{known}, $p = 2$ and d_i is the Euclidean Distance (in the objective space) between each vector and the nearest member in P.

- **Hypervolume (HV)** [20]: This indicator is defined as the area of coverage of PF_{known} with respect to the objective space (in a bi-objective problem). Its mathematical definition can be interpreted as the sum of all the rectangular areas (volumes, hypervolumes,...) bounded by some reference point and $(f_1(x), f_2(x),...,f_n(x))$ for a n-objectives problem:

$$HV \triangleq \left\{ \bigcup_i vol_i | vec_i \in PF_{known} \right\} \qquad (3)$$

where vol_i is the calculated volume (area, hypervolume, ...) correspondent to each vec_i contained in PF_{known}.

- **Δ-index (Δ)** [26]: This is a PI that includes information about distribution and spread. It is a distance based PI

that has the major drawback of being applicable to bi-objective problems only. Mathematically it is defined by:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|PF_{known}|-1} |d_i - \bar{d}|}{d_f + d_l + (|PF_{known}| - 1)\bar{d}} \qquad (4)$$

where d_f and d_l are the Euclidean distances between the extreme and boundary solutions of PF_{known}, $\bar{d}$ is the average of all distances $d_i, i \in [1, |PF_{known}| - 1]$.

The application of these metrics to the results obtained from the algorithms, allowed the compilation of Table I where the mean values (of 30 runs) and 95% confidence intervals for each metric are shown.

TABLE I
MEAN AND 95% CONFIDENCE INTERVALS OF THE PIS FOR THE WHOLE
SET OF ALGORITHMS.

	NSGA-II	SPEA2	EA	SA
GD	0.00395 ±0.00067	**0.00363** ±0.00046	0.02848 ±0.00158	0.02673 ±0.00127
HV	**0.58160** ±0.00911	0.58344 ±0.00626	0.61486 ±0.00971	0.60103 ±0.00832
Δ-index	0.66429 ±0.04375	**0.71910** ±0.03896	0.43977 ±0.02781	0.37232 ±0.01722

An initial empirical analysis of this table allows to conclude that the best results are yield either by the SPEA2 or the NSGA-II algorithms that also show smaller variance, showing that the results are more consistent. In this analysis, keep in mind that for GD and HV a smaller value is better, while for Δ a higher value is preferable. An important point is that the number of non-dominated solutions in a typical pareto front for the MOEAs comprises a lot more solutions than the ones obtained by the single-objective algorithms. Nevertheless, these PIs were selected because they do not depend on the cardinality of the sets.

The values from the table show that not only the NSGA-II and SPEA2 are able to find all the best solutions, they are able to do it in a more efficient way and with a far better resolution. A comparison between the two does not show relevant changes. To have a similar resolution in the EA/SA, it would be necessary to make the G_{min} range from 0% to 100% in 1% steps in the G_{min} (100 runs of the algorithms).

A good distribution of the solutions along the fitness space and a good resolution are both important characteristics in the performance of these algorithms when applied to this particular problem. In a single run of one of these MOEAs, a researcher obtains a panoply of solutions representing possible trade-offs between objectives and is able to easily focus on those of interest.

The figure below shows a graphical comparison between the Pareto sets found for each algorithm in all the runs. In this case, only non dominated solutions found by each algorithm in the 30 runs were kept, so this shows the best solutions of each algorithm. Looking at the figure confirms the conclusions drawn before, showing the ability of both SPEA2 and NSGA-II to find a good set of non-dominated solutions that are near the reference set, and well distributed

along the pareto front. Keep in mind that the reference set can be seen as the line that marks the best solutions over all algorithms.
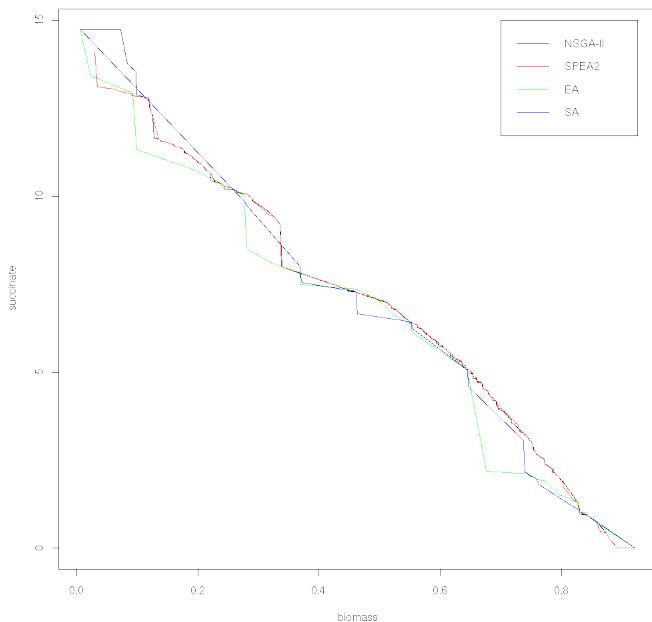

Fig. 4. Graphical representation of the Pareto sets obtained by each algorithm (only non dominated solutions).

VI. CONCLUSIONS AND FURTHER WORK

In this work, a novel approach was proposed to the task of *in silico* optimization of mutant strains in Metabolic Engineering. It was shown that the previous algorithms (EA, SA) can be replaced by multi-objective EAs with several advantages, since the researcher gets a more complete set of results, showing the different trade-offs between the desired compound production and the viability of the strain measured by a biomass flux. The experiments conducted on the proposed case study show that both SPEA2 and NSGA-II, two state of the art MOEAs, are able to obtain good results in terms of solution quality and distribution of the solutions over the Pareto front. Therefore, an interesting tool for Metabolic Engineering has been provided.

Further work includes the validation in other case studies, the testing of other MOEAs, as well as the implementation of these algorithms into a software application that can be used and made available to biological researchers.

REFERENCES

[1] A.P. Burgard, P. Pharya, and C.D. Maranas. Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol Bioeng*, 84:647–657, 2003.

[2] C. Chassagnole, N. Noisommit-Rizzi, J.W. Schmid, K. Mauch, and M. Reuss. Dynamic modeling of the central carbon metabolism of escherichia coli. *Biotechnology and Bioengineering*, 79(1):53–73, 2002.

[3] C.A.C. Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3):129–156, 1999.

[4] M.W. Covert, C.H. Schilling, I. Famili, J.S. Edwards, I.I. Goryanin, E. Selkov, and B.O. Palsson. Metabolic modeling of microbial strains in silico. *Trends in Biochemical Sciences*, 26(3):179–186, 2001.

[5] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, 2000.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[8] J.J. Durillo, A.J. Nebro, F. Luna, B. Dorronsoro, and E. Alba. jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical report, Technical Report ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computacion, University of Malaga, ETSI Informatica, Campus de Teatinos, December 2006.

[9] F.Y. Edgeworth. Mathematical Physics. *P. Keagan, London, England*, 1881.

[10] R.U. Ibarra, J.S. Edwards, and B.G. Palsson. Escherichia coli k-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. *Nature*, 420:186–189, 2002.

[11] K.J. Kauffman, P. Prakash, and J.S. Edwards. Advances in flux balance analysis. *Curr Opin Biotechnol*, 14:491–496, 2003.

[12] S.Y. Lee, S.H. Hong, and S.Y. Moon. In silico metabolic pathway analysis and design: succinic acid production by metabolically engineered escherichia coli as an example. *Genome Informatics*, 13:214–223, 2002.

[13] J. Nielsen. Metabolic engineering. *Appl Microbiol Biotechnol*, 55:263–283, 2001.

[14] V. Pareto. Cours D'Economie Politique, volume I and II. *F. Rouge, Lausanne*, 250, 1896.

[15] K. Patil, I. Rocha, J. Forster, and J. Nielsen. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, 6(308), 2005.

[16] J.L. Reed, T.D. Vo, C.H. Schilling, and B.O. Palsson. An expanded genome-scale model of escherichia coli k-12 (ijr904 gsm/gpr). *Genome Biology*, 4(9):R54.1–R54.12, 2003.

[17] M. Rocha, R. Mendes, P. Maia, J.P. Pinto, I. Rocha, and E.C. Ferreira. Evaluating simulated annealing algorithms in the optimization of bacterial strains. *LECTURE NOTES IN COMPUTER SCIENCE*, 4874:473, 2007.

[18] M. Rocha, J.P. Pinto, I. Rocha, and E.C. Ferreira. Optimization of Bacterial Strains with Variable-Sized Evolutionary Algorithms. In *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 331–337, Honolulu, USA, 2007. IEEE Press.

[19] I.F. Sbalzarini, S. Muller, and P. Koumoutsakos. Multiobjective optimization using evolutionary algorithms. *Proceedings of the Summer Program*, 2000, 2000.

[20] J.D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *Proceedings of the 1st International Conference on Genetic Algorithms table of contents*, pages 93–100, 1985.

[21] D. Segre, D. Vitkup, and G.M. Church. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences*, 99(23):15112, 2002.

[22] N. Srinivas and K. Deb. Muiltiobjective Optimization Using Non-dominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[23] G. Stephanopoulos, A.A. Aristidou, and J. Nielsen. *Metabolic engineering principles and methodologies*. Academic Press, San Diego, 1998.

[24] David Allen Van Veldhuizen. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. PhD thesis, Wright Patterson AFB, OH, USA, 1999. Adviser-Gary B. Lamont.

[25] E. Zitzler, M. Laumanns, L. Thiele, et al. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *EUROGEN*, pages 95–100, 2001.

[26] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case studyand the strength Pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4):257–271, 1999.