Start-up optimisation of a combined cycle power plant with multiobjective evolutionary algorithms

Ilaria Bertini[1], Matteo De Felice[1,2], Fabio Moretti[2], and Stefano Pizzuti[1]

[1] ENEA (Italian Energy New Technology and Environment Agency)
{ilaria.bertini, matteo.defelice, stefano.pizzuti}@enea.it
[2] Università degli Studi "Roma Tre", Dipartimento di Informatica e Automazione
moretti@dia.uniroma3.it

Abstract. In this paper we present a study of the application of Evolutionary Computation methods to the optimisation of the start-up of a combined cycle power plant. We propose a multiobjective approach considering different objectives for the optimisation in order to reduce the pollution emissions and to maximise the efficiency of the plant. We compare a multiobjective evolutionary algorithm (NSGA-II) with 2 and 5 objectives on a software simulator and then we use different metrics to measure the performances. We show that NSGA-II algorithm is able to provide a set of solutions, defined as Pareto Front, that represent the best trade-off on the different objectives among those the decision maker can choose.

1 Introduction

Air pollution emission reduction is nowadays a common requirement for the operation of industrial plants, from the Kyoto Protocol the attention about this issue is growing and therefore a more efficient utilisation of the plants is needed.

Moreover, with the liberalization of the energy market and the introduction of distributed energy power plants in the territory, it is required also for the gas-steam combined plants a greater flexibility in order to vary the provided power according to the needs or to implement variable running strategies. Such management makes more critical the problem of the identification of the best parameters during the start-up in order to reduce the emissions and the thermal stress by maintaining the production efficient.

The problem of finding the best trade-off between production and emissions (of course we can consider more factors) can be arranged like an optimisation problem. Usually such problems are solved by minimizing (or maximizing) a function through the concurrent fulfilment of some constraints, often conflicting each others (multiobjective optimisation, MOOP).

This kind of problems can be solved with a single-objective function approach, which is a combination of various objective functions, but also with a multiobjective approach based on the Pareto Theory.

Evolutionary Computation (EC) methods can be considered a good choice to cope with multiobjective optimisation problems and several applications with effective results in different fields can be found in literature [1]. The main advantage of such methods consists in performing a parallel search of the optimal solution without a priori information about the problem.

In section 2 we introduce the multiobjective optimisation approach based on the Pareto Theory and we present the implemented evolutionary algorithm. In section 3 we describe the problem of the start-up optimisation of a combined cycle, in section 4 and 5 we respectively show the experimentations we made for this work and the results. Finally, in section 6 we comment the results and we give an orientation of our future work.

2 Multiobjective Optimisation

Optimisation techniques for solving Single-Objective Problems (SOPs) are largely developed and well known. However the modelisation of many real world problems leads to more than one and often conflicting objective functions. A problem dealing with two or more objectives is called Multiobjective Problem (MOP).

Since in MOPs objectives can be conflicting, such problems may lead to a set of solution instead of a single solution. Solutions belonging to this set are the result of a trade off between conflicting objectives and in order to define a set of good trade off solution the concept of dominance is introduced. Formally a solution dominates another solution if it is better at least in one objective and it is not worse in all objectives than the other solution. A set composed of all non-dominated solutions is called *Pareto optimal set* or *Pareto front*.

In MOPs, besides finding a set of solutions as close as possible to Pareto front, we need to mantain the solutions as diverse as possible. This because a well spaced set of solutions leaves the decision maker a wider choice of trade off between objectives.

Point-by-point methods have several disadvantages with respect to population based methods on complex problems with non-linear and non-convex spaces. Thus, in multiobjective optimisation Evolutionary Algorithms are largely used.

2.1 Multiobjective Evolutionary Algorithms

Multiobjective Evolutionary Algorithms (MOEAs) allow to find a set of non-dominated solution in each optimisation run. Because of their stochastic nature they are robust enough to tackle non-linear problems. The first algorithm that uses the non-dominated classification is the Multiobjective Genetic Algorithm (MOGA) proposed by Fonseca and Fleming in 1993 [2]. They proposed to assign a rank to each solution based on the number of solutions that dominates that one. This rank allows in some cases to compare two solutions without any fix-up like weights or other parameters. Subsequently many algorithms used non-dominated classification as Non-Dominated Sorting Genetic Algorithm (NSGA) proposed by Deb in 1994 [3] and then upgraded with elitism in 2000 with the name of Elitist

Non-Dominated Sorting Genetic Algorithm (NSGA-II) [4, 5]. Since NSGA-II is a well known, efficent algorithm we use this MOEA for our work and we analyze it in detail. A survey about MOEAs can be found in [6].

In order to compare the quality of two solutions NSGA-II uses the ranking level approach. Given a population, this approach assignes rank level 1 to all non-dominated solution of the entire population, then it assignes rank 2 to all non-dominated solution of the population without solution of rank 1 and so on until it is assigned a rank to all solutions.

In order to mantain a diverse set of solution, it is assigned a crowding distance to each solution. This value is high for isolated solutions and low for solutions with many neighbors of the same rank. Extreme solutions are always taken, with an infinity crowding value, and other solutions are compared to their nearest neighbors (see figure 1).

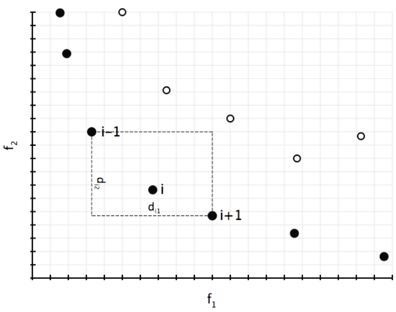

Fig. 1. Crowding Distance

3 Start-up optimisation of a combined cycle power plant

Gas and steam turbines are an established technology available in sizes ranging from several hundred kilowatts to over several hundred megawatts. Industrial turbines produce high quality heat that can be used for industrial or district heating steam requirements. Alternatively, this high temperature heat can be recuperated to improve the efficiency of power generation or used to generate steam and drive a steam turbine in a combined-cycle plant.

The gas-steam combined cycle produces electricity more efficiently than either gas or steam turbine alone because it performs a very good ratio of transformed electrical power per CO_2 emission. CC power plants are characterized as the 21^{st} century power generation by their high efficiency and possibility to operate on different load conditions by reason of the variation in consumer load.

CC plants are highly complex systems but with the availability of high powerful processors and advanced numerical solutions there is a great opportunity to develop high performance simulators for modeling energy systems in order to consider various aspects of the system. This is a complex task including several limitations that have to be fulfilled simultaneously like the maximum allowed thermal stress caused by temperature gradients. In literature one of the most studied problems of CC operation is the start-up optimisation.

As example, in [7] through a parametric study, the start-up time is reduced while keeping the life-time consumption of critically stressed components under control. In [8] an optimum start up algorithm for CC, using a model predictive control algorithm, is proposed in order to cut down the start-up time keeping the thermal stress under the imposed limits. In [9] a study aimed at reducing the start-up time while keeping the life-time consumption of the more critically stressed components under control is presented. In this work the optimisation of the start-up procedure for a CC power plant has been studied by means of a system simulator.

In general, most studies on CC are based on simulators and the goal is to minimise the start-up time alone in single-objective approach managing all other important aspects of the problem, like thermal stress or pollutant emissions, as constraints. In this way the global operations are not optimised because an effective start-up optimisation would be multiobjective.

The start-up scheduling is as follows (see figure 2). From zero to time t_0 ($\approx$ 1200 sec) the rotor engine velocity of the gas turbine is set to 3000 rpm. From time t_0 to t_1 the power load is set to 10 MW and then the machine keeps this regime up to time t_2. All this initial sequence is fixed. From time t_2 to t_3($\approx$ 3600 sec) the machine must achieve a new power load set point which has to be set optimal and then the machine has to keep this regime up to time t_4. The time lag $t_4 - t_3$ has to be optimised and during this interval the steam turbine starts with the rotor reaching the desired velocity. Then the turbines have to reach at time t_5 the normal power load regime (270 MW for the gas turbine) according to two load gradients which need optimising.

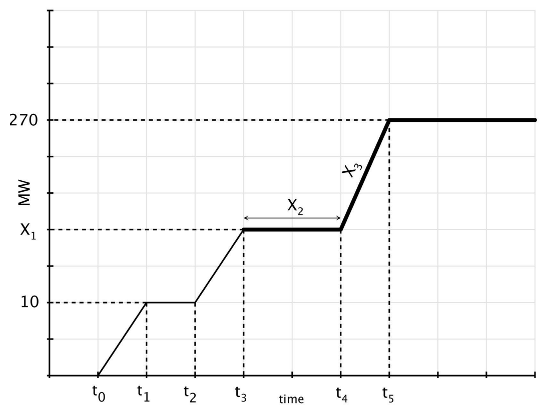

Fig. 2. Start-up sequence

Therefore, the process variables to be optimised with their operating ranges are visible in Table 3

Table 1. Process Variables

Name	Description	Range	Measure Unit
X_1	Intermediate power load set point	$[20, 120]$	MW
X_2	Intermediate waiting time $t_4 - t_3$	$[7500, 10000]$	sec
X_3	Gas turbine load gradient	$[0.01, 0.2]$	MW/s
X_4	Steam turbine load gradient	$[0.01, 0.2]$	$\%/s$

In order to optimise the overall start-up operations, the following objectives should be fulfilled:

1. minimise fuel consumption (Kg)
2. minimise time (s)
3. maximise energy production (KJ)
4. minimise pollutant emissions ($\frac{mg \cdot s}{Nm^3}$)
5. minimise thermal stress

It is therefore obvious that a multiobjective approach would improve the overall performance of such a power plant with a remarkable positive effect on the environment. Thus, we have approached the start-up problem in a multiobjective way by optimising the mentioned criteria with multiobjective genetic algorithms. Experimentations has been carried out on data obtained by means of a software simulator provided by AnsaldoEnergia.

4 Experimentations

We applied our implementation of the NSGA-II algorithm on the multiobjective optimisation of the problem described in Section 3 and we compared it to the following algorithms:

1. RAND: A random search algorithm
2. WSGA: Weighted-Sum Genetic Algorithm
3. NSGA-II: Non-Dominated Sorting Genetic Algorithm

The RAND algorithm is a trivial random search in the input space, with the same number of overall fitness evaluations of the other algorithms. At the end of this sampling, all the non-dominated solutions are considered inside the Pareto Front.

The WSGA applies a weighted sum of all objectives in order to reduce the original MOP to a single objective one. At each run a Genetic Algorithm is executed with a different random convex combination of the weights of the fitness function.

All the algorithms were executed 10 times and the resulting non-dominated set of the union of the Pareto fronts obtained at the end of each run was taken. In order to fairly compare the algorithms, each one is run over the same number of fitness evaluations.

Each input variable (see Table 3) can assume 21 different values, we encoded the decision variables with a Gray code binary string, whose minimal length can be obtained by:

$$\log_2 21^4 \approx 18 \tag{1}$$

The encoding is simple, we enumerated all the solutions (with numbers from 1 to 21^4) assigning each value of the 18-bit string to a solution.

Table 4 describes the algorithms' parameters used during our tests.

Table 2. Algorithm Parameters

	NSGA-II	WSGA
Population Size	100	50
Generations	50	30
Selection	Binary Tournament	
Crossover	Single Point	
Crossover Probability	0.75	
Mutation	Bitwise	
Mutation Probability	1/18	

To evaluate the performance of the different methodologies we used the following metrics:

- Dominance Ratio
- Spacing
- Hypervolume

Dominance Ratio metric was suggested by Zitzler in 1999 [10]. It compares two fronts and returns the percentual of solutions of the first one dominated by the second one, with respect to all the solutions of the first front.

Therefore, the smaller the value the better the first front respect to the second is and a value of one implies that the first front is completely dominated by the second.

Spacing metric, proposed by Schott in 1995 [11], evaluates relative distance between consecutive solutions belonging to non-dominated set. It is related to the spread of each non-dominated set independently. The lower spacing value is the more uniform the distribution of solutions is. Since in MOPs we need to mantain the set of solutions as diverse as possible, as mentioned earlier in section 2, a uniform distribution of solutions is highly preferred.

Hypervolume metric was proposed by Zitzler and Thiele in 1999 [10]. It evaluates both dominance and spreading of solutions. This metric calculates the area covered by the hypervolume whose vertices are the solutions set and a reference point, a vector of worst values each objective function can assume. Since no scaling is used, a good spread of high magnitude solutions in an objective implies a better performance value with respect to a good spread of low magnitude solutions in another objective. In order to reduce the computational load of such metric we used a Monte Carlo estimation of the hypervolume, considering the percentage of random points which are dominated by the Pareto front we are measuring. As reference point we considered the worst values among all the solutions of the Pareto fronts considered.

Even if in real-world problems the real optimal Pareto front usually is not available, we computed, for a complete comparison of the selected algorithms, the fitness values of all the points inside the solution space. Despite it was computationally expensive (it took several days on a cluster with 1024 CPUs) we have the real optimal Pareto front

In order to show graphically the behavior of the algorithms we tested the problem firstly for only two of the five objectives described in section 3. We considered two clearly conflicting objectives: maximising energy production while minimising pollutant emissions. Pareto fronts' graph and relative metrics are presented. Subsequently we considered the problem with all five objectives and we present only the related performance metrics results since the plot of Pareto Fronts is not possible.

5 Results

We show results of two and five objectives optimisation obtained with 10 run of all algorithms but RAND (see Section 4).

We performed a multiobjective optimisation considering two objectives and five objectives. In the first case we considered the maximisation of energy production and minimisation of pollutant emissions and the overall number of fitness evaluations is 15300. In the second case we have the same number of evaluations.

Figure 3 shows that the NSGA-II Pareto front is overlapping with the real one and therefore it dominates all the solutions of the other algorithms while, as expected, the RAND front is dominated by both. The metrics values presented in Tables 3 and 4 reflect this situation. The columns labeled "2D" are related to the experimentations with 2 objectives and, similarly, for the problem with 5 objectives. The last line of Table 4 shows the number of solutions for different Pareto fronts.

For the 5-objectives problem we can't plot directly the Pareto Fronts and so we have to establish the comparison between the algorithms on the metrics' values. We can observe that the size of fronts of the algorithms shows an evident variability: from 10 (WSGA) to 2435 (RAND) and the same we can assert the same for spacing, WSGA shows that the solutions in its front cover a larger space than other two algorithms.

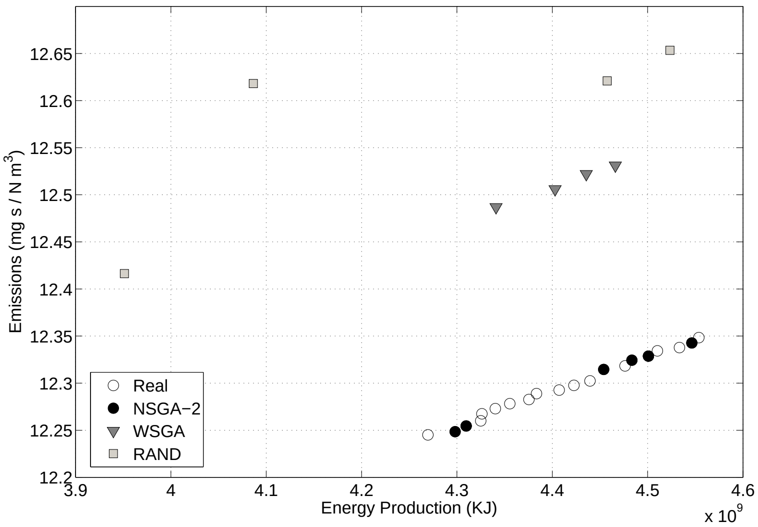

Fig. 3. Pareto Fronts plot on the 2-objectives problem

Table 3. Dominance Ratio

	Real		NSGA-II		RAND		WSGA	
Dimensions	2D	5D	2D	5D	2D	5D	2D	5D
Real	-	-	0	0	0	0	0	0
NSGA-II	0	0.659	-	-	0	0.406	0	0.008
RAND	1	0.637	1	0.014	-	-	0.5	0.001
WSGA	1	0.5	1	0.1	0	0.1	-	-

5.1 Discussion

In two dimensions the results we obtain aren't much different from the ones we expected: the ability of Evolutionary Computation based algorithms like NSGA-II permits to explore effectively the solution space and find the best solutions, achieving a Pareto front far better than those obtained with WSGA or random search.

With 5 dimensions the situation changes drastically. The RAND algorithm becomes the best algorithm, achieving a Pareto front which dominates about the 40% of solutions of the NSGA-II's front and the nearest hypervolume to the optimal one. A probable explanation of this situation should be found in the last line of Table 4, where we can observe that the size of the real optimal Pareto front is about 800 times larger than the optimal one with two objectives. This

Table 4. Spacing, hypervolume and size of the real optimal Pareto front and the ones obtained by the considered algorithms

	Real		NSGA-II		RAND		WSGA	
	2D	5D	2D	5D	2D	5D	2D	5D
Spacing	0.015	0.007	0.002	0.07	0.013	0.023	0.004	0.376
Hypervolume	0.93	0.394	0.898	0.348	0.069	0.37	0.338	0.129
Size	20	15608	11	261	4	2435	4	10

means that random search is more effective because it's simpler to find randomly good solutions than in the 2D space.

It's an interesting observation the fact that the solution proposed from the plant manager results dominated in both the problem spaces, in 2 and 5 dimensions, by all the algorithms we tested. Therefore, all the solutions provided by the algorithms should be considered "better" (from a multiobjective point of view) than the real used ones.

6 Conclusion and Future Work

We underlined the capability of multiobjective optimisation techniques of providing a set of feasible solutions among which a decision can be taken. We made our experimentations on a precise software simulator of a combined cycle plant considering two and five objectives functions.

Considering only a subset of the objectives (maximisation of energy output and minimisation of pollutant emissions) we observe that NSGA-II algorithm works far better than a random search and a combined single-objective algorithm, finding solutions on the real optimal Pareto front. With all the objectives the situation changes and the results of a random search outperform the Evolutionary Computation based approach.

Despite these results seem inconsistent, we think that it is not simple to estimate the performances of a set of algorithms when increasing the number of considered objectives, because in real problems the objectives function to minimise (or maximise) are heterogeneous, i.e. the relation between results in low and high dimensional space is not straightforward. In the real case we considered, a deeper study of objective functions is needed, in order to explore mutual relations between them.

Although the primary goal of this paper is to highlight the application of multiobjective optimisation to a real world problem, comparisons can be extended also to other MOEA for a more complete overview.

Moreover this work raises the issue of reducing the computational load of stochastic algorithms such the ones we used of real problems, where the evaluation of a solution is based on the execution of a software simulator, which reflects the complexity of the problem it simulates. We think that such problem can be coped with by considering an algorithm which uses both the real fitness function and an approximated one, in order to lower the number of executions of the computationally expensive software simulator.

References

1. Cutello, V.: A multi-objective evolutionary approach to the protein structure prediction problem, Journal of The Royal Society Interface, vol. 3, pp 139–151 (2006)
2. Fonseca, C. M., Fleming, P.J.: Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. Proceedings of the Fifth International Conference on Genetic Algorithms, pp 416-423, (1993).
3. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation, vol. 2, no. 3, pp 221–248, (1994).
4. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, In: Lecture Notes in Computer Science, Springer, pp. 849–858 (2000)
5. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, (2001).
6. Coello, C. A.: An updated survey of ga-based multiobjective optimization techniques, ACM Comput. Surv., vol. 32, no. 2, pp. 109–143 (2000)
7. Alobaida, F., Postlera, R., Ströhlea, J., Epplea, B., Kimb, H-G.: Modeling and investigation start-up procedures of a combined cycle power plant, Applied Energy, vol. 85, Issue 12, pp. 1173–1189 (2008)
8. Tetsuya, F.: An optimum start up algorithm for combined cycle, Transactions of the Japan Society of Mechanical Engineers, vol. 67, no. 660, pp. 2129–2134 (2001)
9. Casella, F., Pretolani, F.: Fast Start-up of a Combined-Cycle Power Plant: a Simulation Study with Modelica, In: Proceedings 5th International Modelica Conference, Vienna, Austria, September 6-8, pp. 3–10 (2006)
10. Knowles, J., Corne, J.: On metrics for comparing non-dominated sets, Procediings of the 2002 Congress on Evolutionary Computation (CEC 2002), vol. 1, pages 711–716, (2002).
11. Schott, J.: Fault tolerant design using single and multicriteria genetic algorithm optimization. Masters thesis Department of Aeronautics and Astronautics. Massachusetts Institute of Technology, (1995).