

Assessing the Uses of NLP-based Surrogate Models for Solving Expensive Multi-Objective Optimization Problems: Application to Potable Water Chains

Florin Capitanescu, Antonino Marvuglia, Enrico Benetto,
Luxembourg Institute of Science and Technology (LIST)
Environmental Research and Innovation (ERIN)
41 rue du Brill, L-4422, Belvaux, Luxembourg

Email: {florin.capitanescu,antonino.marvuglia,enrico.benetto}@list.lu

Aras Ahmadi, Ligia Tiruta-Barna
University of Toulouse
INSA, UPS, INP, LISBP

135 Avenue de Rangueil, F-31077 Toulouse, France
Email: {aras.ahmadi,ligia.barna}@insa-toulouse.fr

Abstract—In practice many multi-objective optimization problems relying on computationally expensive black-box model simulators of industrial processes have to be solved with limited computing time budget. In this context, this paper proposes and explores the uses of an iterative heuristic approach aiming at quickly providing a satisfactory accurate approximation of the Pareto front. The approach builds, in each iteration, a multi-objective nonlinear programming (MO-NLP) surrogate problem model using curve fitting of objectives and constraints. The approximated solutions of the Pareto front are generated by applying the ε -constraint method to the multi-objective surrogate problem, converting it into a desired number of single objective (SO) NLP problems, for which mature and computationally efficient solvers exist. The proposed approach is applied to the cost versus life cycle assessment (LCA)-based environmental optimization of drinking water treatment chains. The paper thoroughly investigates various settings choices of the approach such as: the type of the polynomial function to be fit, the input points, choice of weights in curve fitting, and analytical fit. The numerical simulations results with the approach show that a good quality approximation of Pareto front can be obtained with a significantly smaller computational time than with the popular SPEA2 state-of-the-art metaheuristic algorithm.

I. INTRODUCTION

Nowadays there is a plethora of derivative-free metaheuristic algorithms (e.g. genetic algorithms, particle swarm optimization, differential evolution, ant colony, etc.) [1], [2]. These algorithms are suitable mainly for black-box multi-objective optimization (MOO), since well-established derivative-based mathematical programming methods cannot be applied because either the problem formulation in analytical form is unavailable or the problem features (e.g. non-convex, discontinuous, etc.) make such methods impractical. Metaheuristic algorithms are reliable and can explore complex optimal trade-off Pareto fronts in a single run, but also exhibit some drawbacks such as: poor scalability with problem size, slow convergence close to the optimum, and stopping criterion.

Two main classes of meta-heuristic algorithms can be distinguished:

- *Pareto dominance-based*, including elitism-based algorithms such as: Non-dominated Sorting Genetic Algo-

rithm (NSGA-II) [3], Strength Pareto Evolutionary Algorithm (SPEA2) [4], adaptive evolutionary algorithm [5], etc.

- *non-Pareto dominance-based*, comprising e.g. MOEA/D scalar objective function problem decomposition-based algorithm [6], hybrid algorithms combining global and local search [7], etc.

Many of these meta-heuristic algorithms have been developed aiming to accurately approximate complex Pareto fronts. These algorithms performance is generally evaluated using computationally inexpensive benchmark MOO problems and allowing a large number of functions evaluations. However, in many engineering fields, the simulation of the black-box model can be computationally intensive while the computational time budget may be limited. Under these stringent conditions these well-established algorithms may not converge and hence they may not provide a satisfactorily accurate and/or evenly spread Pareto front. These computationally expensive MOO problems with limited number of black-box model evaluations call for the development of new algorithms with different accuracy/speed trade-offs. In this emerging research field, there exist only a few approaches [8], most of which rely on Gaussian stochastic process modelling [9]–[11].

In this paper we propose a new approach for black-box model expensive MOO problems. The approach builds, in an iterative fashion, a MO-NLP surrogate model based on (objectives and constraints) functions polynomial curve fitting, taking advantage of the existing mature NLP solvers.

The interest of the proposed approach is demonstrated, without loss of generality, for the bi-criteria (cost versus environmental impact) optimization of drinking water production plants. In this work the environmental impact is modeled adopting a life cycle assessment (LCA) approach. LCA is a standardized methodology [12] which analyzes the lifecycle (generally from raw material extraction until the end-of-life of the product) environmental performance of any industrial process but without indicating per se optimal means for reducing the environmental impact.

The extension of various industrial processes optimization

to account for environmental impact has been extensively explored up to now, e.g. [13], [14]. However,

- most of the previous works assume that the process can be (approximately) modeled analytically allowing thereby resorting to mathematical programming-based methods such as: linear programming [15], NLP [16], MILP [17], mixed-integer nonlinear programming [18], etc. As will be explained later on these approaches cannot, or are impractical to be, applied to our black-box MOO problem.
- the LCA optimization of drinking water production plants has received little attention up to the present date, e.g. [19] uses the Nelder-Mead local search method for the SO optimization of a certain environmental impact, [20] relies on global optimizers (e.g. SPEA2 and NSGA-II) for the bi-criteria (cost vs LCA-based environmental impact), and [21] uses a two-step hybrid algorithm combining NSGA-II and the local search algorithm COBYLA for three-criteria optimization (cost vs LCA-based environmental impact vs water quality).

The paper is organized as follows. Section II briefly describes the potable water production plant simulator, called EVALEAU, formulates conceptually the MOO problem and presents the MOO - process engineering - LCA tool. Section III describes the proposed approach for the MOO. Section IV provides optimization results with the approach for a realistic model of a real-world drinking water treatment chain. Section V concludes.

II. LCA - PROCESS ENGINEERING - MULTI-OBJECTIVE OPTIMIZATION PROBLEM

A. EVALEAU Simulator

The EVALEAU simulator is a state-of-the-art process modelling - LCA tool for prospective and retrospective simulation of drinking water treatment chains [22]. The simulator comprises a certain number of unit processes for water treatment which can be combined to simulate a specific treatment chain (see e.g. Fig. 1). The simulator was developed in the LCA software Umberto[®] and is linked to the Ecoinvent[®] database [23] for the life cycle inventory (LCI) of background processes. It relies on the software PHREEQC [24] for water chemistry calculation.

B. MOO Problem Abstract Formulation

The MOO problem corresponding to a relevant operating scenario of a potable water production plant can be abstractly formulated as follows:

$$\min_{x_1, \dots, x_n} \{f_1(x_1, \dots, x_n); f_2(x_1, \dots, x_n)\} \quad (1)$$

$$\text{s.t.} \quad g_q(x_1, \dots, x_n) = 0, \quad q = 1, \dots, n_g \quad (2)$$

$$h_r(x_1, \dots, x_n) \geq \underline{h}_r, \quad r = 1, \dots, n_h \quad (3)$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, n, \quad (4)$$

where: $\mathbf{x} = [x_1, \dots, x_n]$ is the vector of decision variables (e.g. design and operational parameters of the various unit

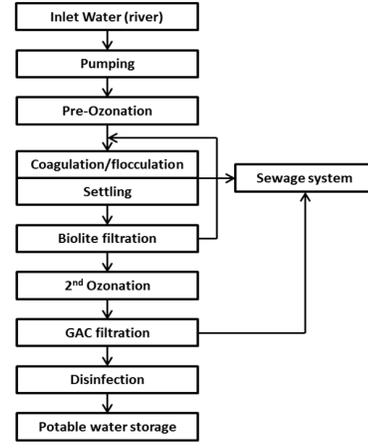


Fig. 1: Flowchart of the potable water production plant used in the case study.

processes of the treatment chain), the goal f_1 represents the operational cost of the water plant (e.g. costs of raw materials, chemicals, electricity, etc.), the goal f_2 models the LCA-based environmental impact of the water production plant.

Constraints (2) describe the input-output mass flow for each unit process in the whole chain. Constraints (3) enforce outlet water drinkability quality, according to the best water quality class in France [25], modeled by a set of seven major aggregated parameters¹. Finally, constraints (4) represent physical bounds of the decision variables.

C. High Level Software Architecture of the LCA - Process Engineering - Multi-Objective Optimization Tool

Note that, tackling the MOO problem (1)-(4) by means of classical mathematical programming methods is impractical due to several reasons, such as: the constraints (2)-(3) possess challenging features (e.g. non-linearity, non-convexity, bi-level as they rely on the expert program PHREEQC for simulating chemical reactions equilibrium in some unit process, being thereby very difficult to fully express in analytical form, large scale size, etc.) and the evaluation of all functions involved in optimization depends on the outcome of the simulator run.

To overcome these challenges we adopt a solution, shown in Fig. 2, which couples the EVALEAU simulator with a multi-goal optimizer. The optimization problem is hence decoupled into two manageable sub-problems which are solved iteratively: (a) solution of equality constraints (2) obtained by running the simulator for a given vector of control variables \mathbf{x} , and (b) improvement of the MOO problem solutions by the optimizer, which is fed with the values of functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, and $h_r(\mathbf{x})$ just evaluated by the simulator, and returns to the latter new promising values for decision variables \mathbf{x} .

¹E.g. total coliforms, total trihalomethanes, total organic carbon, Escherichia coli, faecal streptococci, turbidity, and conductivity.

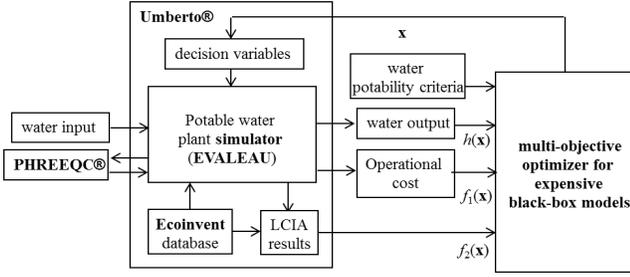


Fig. 2: Software architecture of the integrated tool coupling EVALEAU simulator with a MO optimizer.

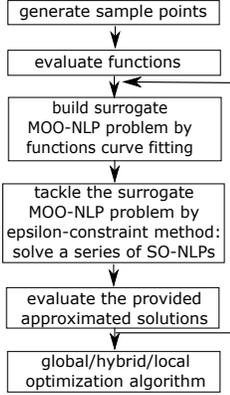


Fig. 3: Flowchart of the proposed approach.

III. THE PROPOSED APPROACH FOR EXPENSIVE MOO PROBLEMS

A. Flowchart of the Approach

Figure 3 shows the flowchart of the proposed MO-NLP approach. The core of the approach is the way in which the surrogate MO optimization problem is built.

The proposed approach has three main steps:

- 1) Curve fitting of objective functions and constraints.
- 2) Formulation of the MO-NLP problem approximation relying on the functions fitting in the previous step.
- 3) Tackling the MO-NLP problem by the ϵ -constraint method described in Section III-D, which converts the problem into a desired number of SO-NLP problems.

The method can work with any set of input points at hand. The input points can be generated randomly, e.g. as in most meta-heuristic algorithms, or in a pre-determined way, e.g. as for a brute-force sensitivity computation of functions with respect to decision variables.

Depending on the context and computational budget the approach can be used either stand-alone or serving as input for classical (global, hybrid, or local) meta-heuristic algorithms, aiming to improve the accuracy/speed trade-off of the latter. Also, the approach can be used either as a one-shot generator of trial solutions, or, in an iterative fashion, by accumulating points generated in previous stages until no significant improvement is obtained.

B. Functions Curve Fitting

The polynomial function to be fit can be expressed in the following general form²:

$$f(\mathbf{x}) = a^0 + \sum_{i=1}^n \sum_{k=1}^d a_i^k x_i^k, \quad (5)$$

where, for the polynomial to be fit, d is its degree and a^0 , a_i^k 's are its coefficients.

Assuming t trial points $\mathbf{x}^1, \dots, \mathbf{x}^t$ and their corresponding values ($\mathbf{x}^t, y^t = f(\mathbf{x}^t)$), the coefficients of the function (5) can be found solving the following unconstrained nonlinear optimization problem which minimizes the weighted least squares of the distance between the analytical function and the real trial values:

$$\min_{a^0, a_i^k} \chi^2 = \sum_{j=1}^t \left(\frac{y^j - a^0 - \sum_{i=1}^n \sum_{k=1}^d a_i^k (x_i^j)^k}{\sigma^j} \right)^2, \quad (6)$$

where σ^j is the weight associated to the trial point j .

The value of the objective χ^2 in (6) yields a measure of the fit accuracy. If this value exceeds some threshold, e.g. the simplest metric is the number of degrees of freedom $t - n \times d - 1$, then the curve fit is rejected.

Two chief aspects of the curve fitting procedure require special attention:

- the a priori choice of the analytical function model to be fit, which requires trading-off the number of required trial points (and hence the computational burden to generate them) and the expected approximation quality. The facts that (i) we seek only a reasonable approximation of functions needed by the surrogate MOO problem and (ii) our simulator is time consuming limiting the type of function to fit to those requiring a reasonably small number of trial points such as: linear, quadratic, or cubic polynomial functions, which requires to evaluate at least $n+1$, $2n+1$, and $3n+1$ trial points, respectively. Linear functions are discarded as they may be too inaccurate and we will focus hereafter only on quadratic and cubic functions.
- the choice of the weights σ^j 's will also be investigated aiming to force the curve to better fit the best solutions at hand to the detriment of less good solutions, and hence explore the most promising areas of the search space.

1) *A Particular Case: Fitting Analytically Quadratic Polynomials*: An interesting particular case of curve fitting arises if one choses to fit a quadratic polynomial function, by superposing n individual quadratic functions, each one expressed as a function of a single decision variable, i.e.

$$f(\mathbf{x}) = \sum_{i=1}^n f(x_i) = \sum_{i=1}^n (a_i^0 + a_i^1 x_i + a_i^2 x_i^2), \quad (7)$$

²Note that fitting more complex polynomial functions may not be computationally affordable in our context. For instance, adding products of variables $x_i x_j, i \neq j$ in a say quadratic polynomial function, will lead to the evaluation of $0.5(n^2 - n)$ additional terms, whereas increasing the degree of a polynomial leads only to n extra terms.

and where the trial points are selected in a pre-determined way, by using only 3 points per decision variable i which stem from the shift of only the variable i (e.g. using minimum bound, maximum bound, and the half of the physical range).

In these assumptions one can easily compute analytically the $3n$ unknown coefficients and build up the quadratic curve.

C. Formulation of the MO-NLP Problem Approximation

Once trusted functions approximation of the objectives and constraints have been built, the bi-objective constrained MO-NLP surrogate problem can be formulated as follows:

$$\min_{x_1, \dots, x_n} \left\{ f_1(x_1, \dots, x_n); f_2(x_1, \dots, x_n) \right\} \quad (8)$$

$$\text{s.t. } f_1(x_1, \dots, x_n) = (a^0)^{f_1} + \sum_{i=1}^n \sum_{k=1}^d (a_i^k)^{f_1} x_i^k \quad (9)$$

$$f_2(x_1, \dots, x_n) = (a^0)^{f_2} + \sum_{i=1}^n \sum_{k=1}^d (a_i^k)^{f_2} x_i^k \quad (10)$$

$$f_1(x_1, \dots, x_n) \leq f_1^{\max} \quad (11)$$

$$f_2(x_1, \dots, x_n) \leq f_2^{\max} \quad (12)$$

$$h_r(x_1, \dots, x_n) = (a^0)^{h_r} + \sum_{i=1}^n \sum_{k=1}^d (a_i^k)^{h_r} x_i^k \geq \underline{h}_r, \quad (13)$$

$$r = 1, \dots, n_h, \quad (13)$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, n \quad (14)$$

where new notations are self-explanatory.

Note that, with respect to the original problem formulation (1)-(4) the equality constraints (2) have been dropped as they are satisfied whatever the combination of decision variables values within their physical range. This fact also significantly relieves the MO-NLP formulation.

The MO-NLP problem (8)-(14) can be converted into a series of m SO-NLP problems, via the ε -constraint method described in Section III-D.

Furthermore, if quadratic functions fit is chosen, the problem becomes a quadratically constrained quadratic program (QCQP) which exhibits some computational advantages (e.g. all second order derivatives are constant) which can be better exploited by a solver.

D. Solution of the MO-NLP Problem by the ε -Constraint Method

The bi-objective MO-NLP problem is tackled using the ε -constraint method [26], which, in order to generate the desired number of solutions m , solves a series of m single objective slightly modified NLP problems, according to the following algorithm:

- 1) Solve the SO-NLP problem $f_1^{\min} = \min f_1$ subject to (9), (10), (13), and (14). Set f_2^{\max} to the value of the objective f_2 given by (10) at the solution of this optimization problem.
- 2) Solve the SO-NLP problem $f_2^{\min} = \min f_2$ subject to (9), (10), (13), and (14). Set f_1^{\max} to the value of

the objective f_1 given by (9) at the solution of this optimization problem.

- 3) Assuming, without loss of generality, that the objective f_2 will be modeled as constraint³:

For $l = 3, \dots, m$, do:

- a) Set $f_2^{\max} \leftarrow f_2^{\min} + (l-2)(f_2^{\max} - f_2^{\min})/(m-1)$ in constraint (12).
- b) Solve the SO-NLP problem $\min f_1$ subject to (9)-(14).

IV. NUMERICAL RESULTS

A. Brief Description of the Case Study and Simulation Tools

The proposed optimization approach is illustrated to a real-world model of a French potable water treatment chain. The treatment chain of the inlet river water contains the major unit process shown in Fig. 1 such as: pumping, two ozonation phases, coagulation/flocculation, settling, biolite filtration, granular activated carbon (GAC) filtration, and bleach disinfection.

The simulations are run on a 2.70GHz/8GB computer. The EVALEAU simulator runs in Umberto[®]5.6 environment and is linked to the Ecoinvent[®] v2.2 LCI database for background processes.

The environmental impact objective function focuses only on climate change impact category (Global Warming Potential - GWP - factors based on 100 years-time horizon) and is computed using the Midpoint ReCiPe life cycle impact assessment method [27]. The functional unit is $1m^3$ of treated water while monetary units (m.u.) stand for euro.

B. Experiments Assumptions

The numerical experiments and comparisons conducted hereafter rely on the following assumptions:

- We consider a set of 6 decision variables.
- One requires approximating the Pareto front by 24 solutions.
- As baseline for performance comparison of the proposed approach we use the benchmark elitist global meta-heuristic evolutionary algorithm SPEA2 [4]. The algorithm is run with the default values proposed on PISA platform [28], using a population of 24 individuals.
- Three comparison criteria⁴ are used:
 - *criterion C1*: quality of approximated fronts after (ideally) the same number of evaluations;
 - *criterion C2*: computational effort (i.e. number of evaluations) to reach a quasi-equivalent approximated front quality;
 - *criterion C3*: the reduction of computational effort of an evolutionary algorithm via the hybridization with the proposed approach, the latter serving as an input.

³The objective that will be modeled as constraint can be selected as the one with the largest normalized interval variation by comparing the ratios: f_1^{\max}/f_1^{\min} and f_2^{\max}/f_2^{\min} in order to obtain a more even spread of the Pareto front.

⁴Note that fronts comparison metrics (e.g. hypervolume indicator) are not used here as the mere front visualization allows easily concluding.

	pre-determined points		random points	
	quadratic	cubic	quadratic	cubic
t	30	30	78	73
p	13	19	13	19
$\chi^2(f_1)$	0.01413	0.001169	0.005353	0.0003572
$\chi^2(f_2)$	0.16970	0.027966	0.060445	0.0045585

TABLE I: Quality of the fit of two types of curves (quadratic and cubic) by means of the χ^2 value.

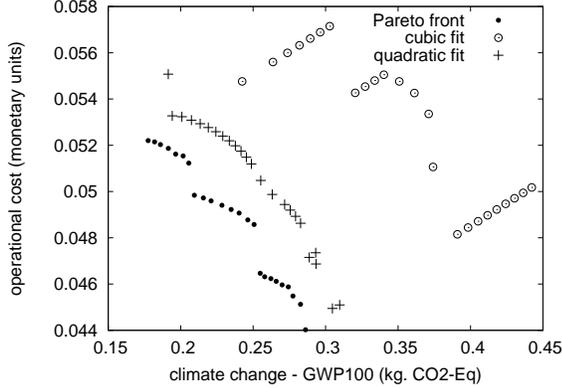


Fig. 4: Quadratic vs cubic (curve fitting): fronts obtained with the proposed approach, in one shot, for 30 pre-determined input points.

- curve fitting unconstrained optimization problem (6) is solved via the “curve_fit” Python function while NLP problem is solved by the IPOPT solver [29].

C. Choice of the Type of Curve to be Fit

Table I provides the quality of curve fitting for both goal functions f_1 and f_2 according to the χ^2 objective (6), for two types of curve fit (quadratic and cubic), and two sets of trial points (pre-determined and random). In this table t is the number of trial points used by the curve fitting procedure and p the number of coefficients to be found by the curve fitting procedure. The 30 pre-determined points are chosen in such a way that each decision variable takes five values (minimum bound, maximum bound, and then: a quarter, a half, and three quarters of its physical range) while the other variables are kept to their initial value. One can observe that according to the curve fitting objective function χ^2 , the cubic polynomial function fits better the given datasets (as more degrees of freedom are allowed) than the quadratic counterpart.

Figures 4 and 5 provide, for the cost vs. environmental impact optimization, the approximations of the Pareto front with both quadratic and cubic curves fitting and both pre-determined and random points. The graphs in Figure 5 follow the proposed approach, starting from the 25 random points and iterating, accumulating newly generated points to the existing ones, until no significant improvement in the front approximation is obtained. Observe that, the cubic function provides unexpectedly (according to results in Table I) poor

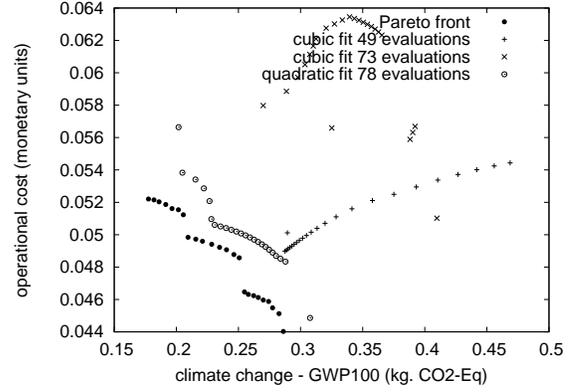


Fig. 5: Quadratic vs cubic (curve fitting): final fronts obtained with the proposed approach, in iterative fashion, starting from 25 random points.

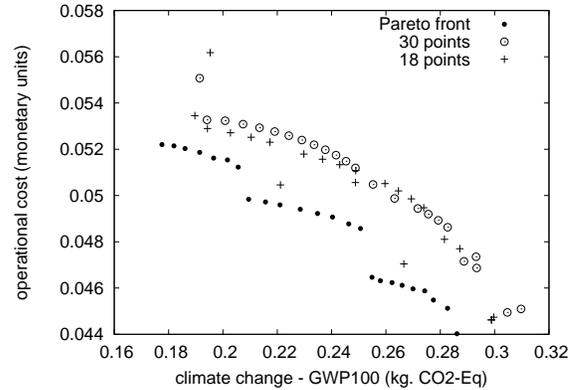


Fig. 6: Front approximation with quadratic fit for different number of pre-determined input points.

quality fronts (in terms of sub-optimality and distribution of solutions), whereas the quadratic approximation leads in our example to much better approximations of the Pareto front.

This finding, corroborated by the fact that the larger the polynomial degree to be fit the larger the computational effort, suggests that, in the absence of a reasonable beforehand guess of the real functions to be fit, one should keep the polynomial function degree as small as possible. Consequently, the quadratic curve fitting appears as a good candidate solution for the trade-off between the approximation quality and the computational effort. Furthermore, as Figure 6 shows for the quadratic fit, adding more points does not necessarily improve the quality of the results.

Figure 7 plots, assuming quadratic polynomial functions, the fronts obtained with the curve fitting procedure using 25 points, and with the analytical formula (see subsection III-B1), using 18 points (i.e. 3 points per decision variable). One can observe that the use of the analytical approach produces in this case slightly worse results.

Based on the results presented so far, we will proceed

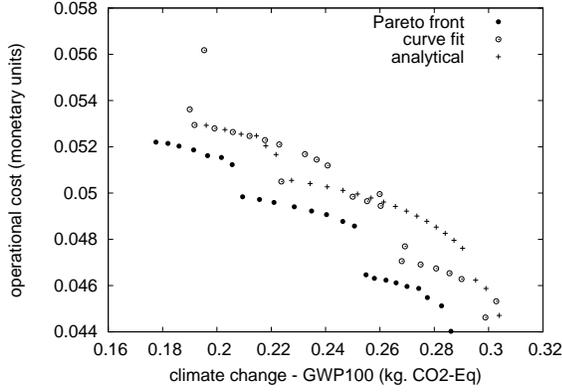


Fig. 7: Front approximation with quadratic functions fit vs analytical fit.

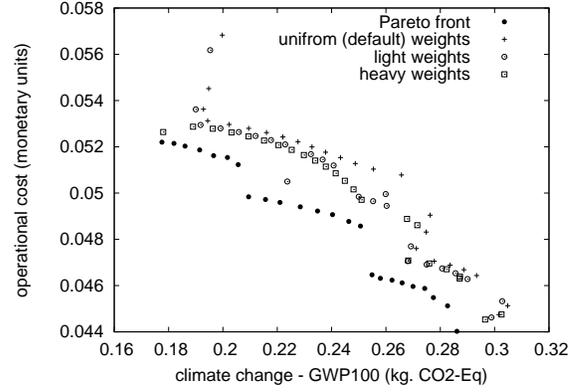


Fig. 8: Impact of weights for pre-determined points.

hereafter assuming only quadratic curves to be fit.

D. Choice of Curve Fitting Weights

An essential aspect for improving NLP approach performances is the choice of weights associated with the t trial points according to the curve fitting optimization problem (6). To this end we consider three sets of weights in (6):

- uniform weights (default) i.e. $\sigma^j = 1, j = 1, \dots, t$
- light weights i.e. $\sigma^j = y^j, j = 1, \dots, t$
- heavy weights i.e. $\sigma^j = (y^j)^2, j = 1, \dots, t$

The purpose of light and heavy weights is to force the curve fitted to stay closer (the heavier the weights the closer the fit) to the given data, according to the objective value (i.e. the better the objective, the closer the distance between the point and the curve).

Let us first consider as input the set of 18 pre-determined points obtained such that each decision variable takes three values (minimum bound, maximum bound, and the half of its physical range) while the other variables are kept to their initial value.

Figure 8 shows that, given the restricted computational budget, the fronts obtained with both light and heavy weights lead to comparable good approximations of the Pareto front, which are systematically better, in terms of optimality and number of solutions provided, than the front obtained using the default weights.

Figure 9 plots, for the sake of readability but without loss of generality, only the non-dominated solutions obtained with the proposed approach for the three sets of weights and starting from 25 random points. The figure clearly shows that the fronts obtained with both light and heavy weights outperform the front obtained with default weights, the heavy weights leading to slightly superior accuracy than the light weights.

These results highlight the importance of choosing weights properly. We proceed hereafter assuming light weights only.

E. Starting from Randomly Generated Solutions

A major advantage of the proposed approach is the ability to efficiently take advantage of a given set of (random) solutions

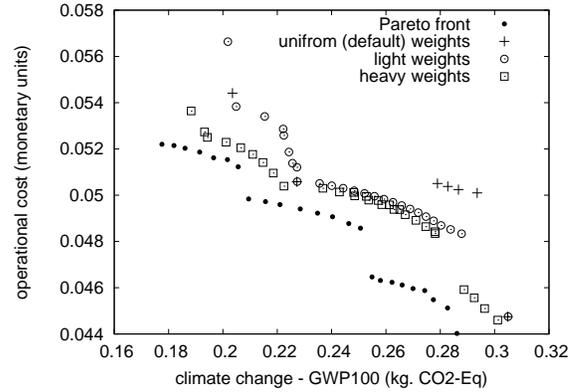


Fig. 9: Impact of weights for random points.

at hand. We illustrate this feature of the approach in an iterative fashion starting from 25 points (the base case and the 24 random points generated by the SPEA2 algorithm in the initial population).

For the sake of readability the solutions generated with the NLP approach is limited to three iterations and are presented in Figures 10 and 11.

Figure 10 displays the initial random points generated by SPEA2, the Pareto front, and the first front approximation of the MO-NLP approach. Note that, although the MO-NLP produces initially only five different solutions instead of 24, they are practically containing the initial random points and are located closer to the Pareto front.

Figure 11 shows the Pareto front approximations obtained with the NLP approach during the first three iterations. Note that, thanks to the newly added five solutions at the first iteration, in the second iteration the MO-NLP approach generates 24 distinct solutions, progressing quite well toward the Pareto front. In the third iteration the new front produced by the MO-NLP approach improves generally in both the optimality and distribution spread. However, although not shown on the figure for readability reasons, from the fourth iteration on the front improves little emphasizing the limitation of the NLP

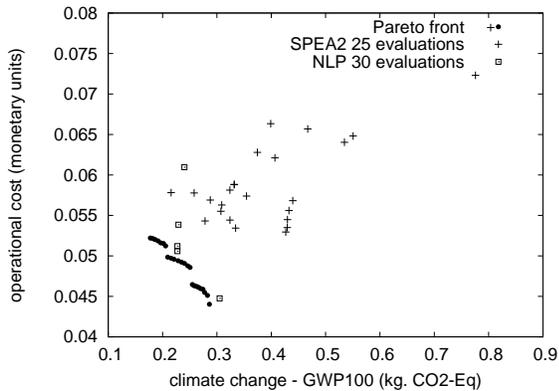


Fig. 10: Approximated front for random starting solutions.

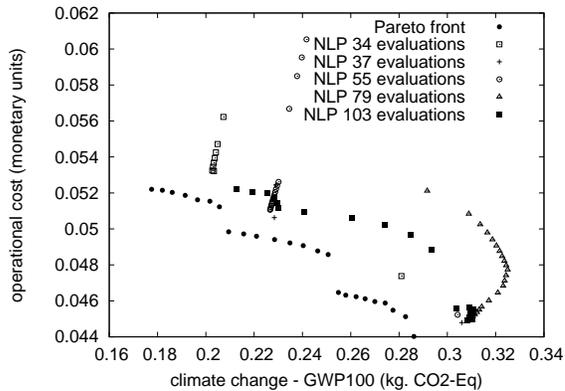


Fig. 12: Approximated front in the first five iterations for SPEA2' 24 input points.

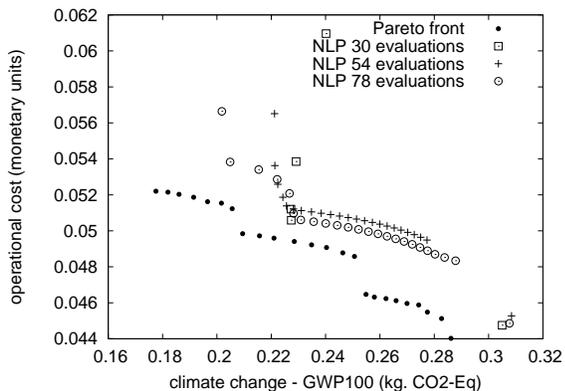


Fig. 11: Approximated front in the first three iterations.

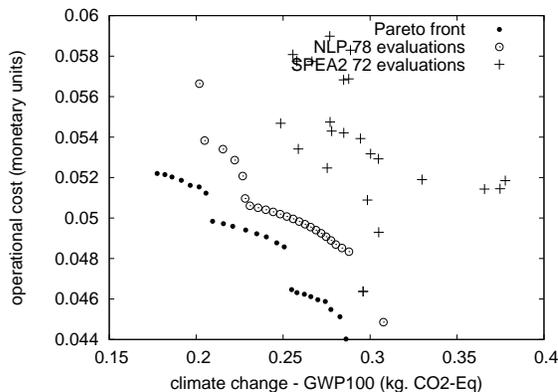


Fig. 13: MO-NLP vs. SPEA2: front approximation according to criterion C1.

approach as a stand-alone technique.

In order to study the MO-NLP performance for another set of input points, we remove the base case⁵ from the initial set, and run the algorithm with only the set of 24 random points generated by SPEA2 in the initial population. Figure 12 yields the Pareto front approximations obtained with the NLP approach during iterations for the new set of input points. One can observe that the NLP approach needs 103 evaluations (distributed over five iterations) until the front approximation starts taking an acceptable shape. Comparing Figs. 11 and 12 one can note that, using a slightly better set of input points leads in less evaluations to a much better front approximation. This experiment highlights the important influence of the starting points set quality on the performances of the approach.

F. MO-NLP versus SPEA2

Figures 13 and 14 allow comparing the approximated fronts produced by MO-NLP and SPEA2, according to criteria C1 and C2. One can easily observe that MO-NLP clearly outperforms SPEA2 in both cases. In particular it allows

⁵The base case is closer to the Pareto front than most other points from the initial set.

saving roughly 2.5 evaluations in order to achieve a front of comparable quality according to criterion C2.

G. Hybrid Algorithm: MO-NLP as an Input for SPEA2

Figure 15 plots the approximated fronts obtained with both SPEA2 algorithm, initialized with default settings (see Fig. 10), and the proposed hybrid MO-NLP-SPEA2, where MO-NLP approach starts from the same initial population as SPEA2 and, after performing three iterations, provides its final population as an initial population for SPEA2. One can remark that, after the same number of evaluations, most solutions provided by the hybrid MO-NLP-SPEA2 belong or are very close to the Pareto front, while the original SPEA2 algorithm solutions are still lagging behind. This example clearly demonstrates the interest of using a hybrid algorithm, with MO-NLP as an input for an evolutionary algorithm to speed up the convergence of the latter.

V. CONCLUSION

This paper has proposed an approach tailored especially for expensive black-box model multi-objective constrained optimization problems. The approach aims to quickly provide

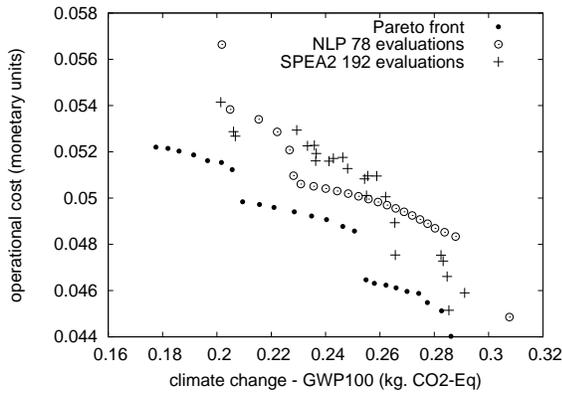


Fig. 14: MO-NLP vs. SPEA2: front approximation according to criterion C2.

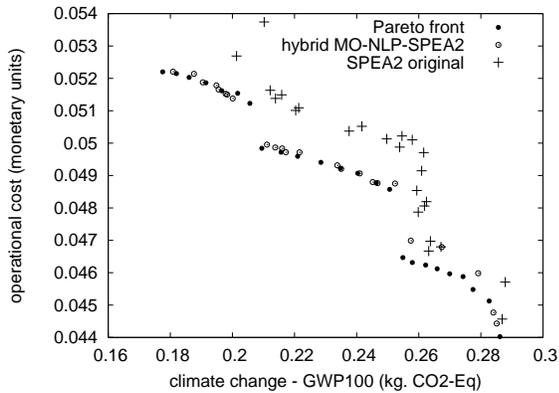


Fig. 15: SPEA2 vs hybrid MO-NLP-SPEA2: front approximation (after 240 evaluations) according to criterion C3.

a sufficiently accurate approximation of the Pareto front and relies on an NLP surrogate model, taking thereby advantage of powerful existing NLP solvers, which are able to solve such proxy models with negligible computational effort compared to a single model simulation.

Numerical results have shown that the proposed approach outperforms a classical state-of-the-art multi-objective algorithms (e.g. SPEA2) in terms of front quality approximation with limited computational time budget.

Depending on the computational time budget, the proposed approach can be used as stand-alone or to assist a classical meta-heuristic algorithm serving as an input for the latter aimed to accelerate its convergence. The proposed NLP approach is an anytime versatile technique which can take advantage of the set of solutions at hand. However, its performance crucially depends on the quality of the input points. The conducted numerical experiments suggest that a significant improvement can be expected if the input trial solutions are located far from the Pareto front. On the other hand, due to the approximations used, and similarly to the evolutionary algorithms, if the trial solutions are located close to the Pareto

front the improvement is expectedly marginal. Therefore the approach is suitable in early stages of the solution space search. We plan future work to integrate the NLP approach within a state-of-the-art metaheuristic algorithm, using it as an independent operator.

Although the proposed method has been applied to the bi-criteria (e.g. cost versus life cycle assessment-based environmental impact) optimization of drinking water production chains at planning stage, it is generic to other application fields addressing (expensive) multi-goal optimization problems, with regard to the problem formulation and operational context (e.g. from operational planning until close to on-line).

Last but not least, it is worthy remarking that, as regards the treatment of the LCA-based environmental metric within MOO problem, the proposed approach represents an important leap forward with respect to the state-of-the-art, since the calculation of the objective scores is based on a fully-fledged LCA (based on the simulation of the operational chain) and not just on the simplified calculation of an environmental metric (e.g. CO₂-Eq) based on the link between the optimization algorithm and an inventory table.

ACKNOWLEDGMENT

The authors acknowledge the funding from Luxembourg National Research Fund (FNR) in the framework of the OASIS project (CR13/SR/5871061).

REFERENCES

- [1] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art", *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32-49, 2011.
- [2] K. Deb, "Multi-objective optimization", *Search methodologies*, pp. 403-449, Springer, 2014.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA: Improving the strength pareto evolutionary algorithm for multiobjective optimization", *Evolutionary Methods for Design, Optimization, and Control*, pp. 95-100, 2002.
- [5] D. Hadka and P. Reed, "BORG: An auto-adaptive many-objective evolutionary computing framework", *Evolutionary Computation*, vol. 21, no. 2, pp. 231-259, 2013.
- [6] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition", *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.
- [7] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization", *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 495-511, 2013.
- [8] L.V. Santana-Quintero, A.A. Montano, and C.A. Coello Coello, "A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-Objective Optimization", In: *Computational intelligence in expensive optimization problems*. Springer Berlin Heidelberg, pp. 29-59, 2010.
- [9] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems", *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50-66, 2006.
- [10] Q. Zhang, W. Liu, E. Tsang, and B. Virginias, "Expensive multiobjective optimization by MOEA/D with Gaussian process model", *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456-474, 2010.
- [11] M. Zuluaga, G. Sergent, A. Krause, and M. Püschel, "Active learning for multi-objective optimization", *Proceedings of the 30th International Conference on Machine Learning*, pp. 462-470, 2013.
- [12] International Organization for Standardization, *Environmental Management: Life Cycle Assessment: Principles and Framework*, ISO-14040, 1997.

- [13] L. Jacquemin, P.-Y. Pontalier, and C. Sablayrolles, "Life cycle assessment (LCA) applied to the process industry: a review", *The International Journal of Life Cycle Assessment*, vol. 17, no. 8, pp. 1028-1041, 2012.
- [14] C. Pieragostini, M.C. Mussati, and P. Aguirre, "On process optimization considering LCA methodology", *Journal of Environmental Management*, vol. 96, no. 1, pp. 43-54, 2012.
- [15] A. Azapagic and R. Clift, "The application of life cycle assessment to process optimization", *Computers & Chemical Engineering*, vol. 10, pp. 1509-1526, 1999.
- [16] B.H. Gebreslassie, G. Guillén-Gosálbez, L. Jiménez, and D. Boer, "Design of environmentally conscious absorption cooling systems via multi-objective optimization and life cycle assessment", *Applied Energy*, vol. 86, no. 9, pp. 1712-1722, 2009.
- [17] F. You, L. Tao, D.J. Graziano, and S.W. Snyder, "Optimal design of sustainable cellulosic biofuel supply chains: multiobjective optimization coupled with life cycle assessment and input-output analysis", *AIChE Journal*, vol. 58, no. 4, pp. 1157-1180, 2012.
- [18] G. Guillén-Gosálbez and I. Grossmann, "A global optimization strategy for the environmentally conscious design of chemical supply chains under uncertainty in the damage assessment model", *Computers & Chemical Engineering*, vol. 34, no. 1, pp. 42-58, 2010.
- [19] R.J. Wallace, A. Marvuglia, E. Benetto, and L. Tiruta-Barna, "A New Approach to Optimization with Life Cycle Assessment: Combining Optimization with Detailed Process Simulation", In: B. Murgante et al. (Eds.): ICCSA 2014, Part III, LNCS 8581, *Computational Science and Its Applications - ICCSA 2014*, pp. 707-720, 2014.
- [20] F. Capitanescu, E. Igos, A. Marvuglia, and E. Benetto, "Coupling Multi-objective Constrained Optimization, Life Cycle Assessment, and Detailed Process Simulation for Potable Water Treatment Chains", *Journal of Environmental Accounting and Management*, to appear, 2015.
- [21] A. Ahmadi and L. Tiruta-Barna, "A Process Modelling - Life Cycle Assessment - MultiObjective Optimization (PM-LCA-MOO) tool for the eco-design of conventional treatment processes of potable water", *Journal of Cleaner Production*, vol. 100, no. 1, pp. 116-125, 2015.
- [22] Y. Méry, L. Tiruta-Barna, E. Benetto, and I. Baudin, "An Integrated "Process Modelling-Life Cycle Assessment" Tool for the Assessment and Design of Water Treatment Processes", *Int. J. Life Cycle Assessment*, vol. 18, no. 5, pp. 1062-1070, 2013.
- [23] B.P. Weidema et al., "Overview and methodology: Data quality guideline for the ecoinvent database version 3", Swiss Centre for Life Cycle Inventories, 2013.
- [24] D.L. Parkhurst and C.A.J. Appelo, "Description of Input and Examples for PHREEQC Version 3—a Computer Program for Speciation, Batch-reaction, One-dimensional Transport, and Inverse Geochemical Calculations", 2013.
- [25] Water Quality Evaluation System in France (version 2), available on-line <http://sierm.eaurmc.fr/eaux-superficielles/fichiers-telechargeables/grilles-seq-eau-v2.pdf>, (in French), 2003.
- [26] G. Mavrotas, "Effective implementation of the ε -constraint method in multi-objective mathematical programming problems", *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455-465, 2009.
- [27] M. Goedkoop, R. Heijungs, M. Huijbregts, A. De Schryver, J. Struijs, and R. van Zelm, "ReCiPe 2008, A life cycle impact assessment method which comprises harmonised category indicators at the midpoint and the endpoint level", RIVM report, 2009.
- [28] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "PISA - A platform and programming language independent interface for search algorithms", *Evolutionary multi-criterion optimization*, pp. 494-508, 2003.
- [29] A. Wächter and L.T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming", *Mathematical programming*, vol. 106, no. 1, pp. 25-57, 2006.