Computers and Chemical Engineering 82 (2015) 228–239

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

# Some efficient approaches for multi-objective constrained optimization of computationally expensive black-box model problems

F. Capitanescu[a,*], A. Ahmadi[b], E. Benetto[a], A. Marvuglia[a], L. Tiruta-Barna[b]

[a] Luxembourg Institute of Science and Technology (LIST), Environmental Research and Innovation (ERIN) Department, 41 rue du Brill, L-4422 Belvaux, Luxembourg
[b] University of Toulouse, INSA, UPS, INP, LISBP, 135 Avenue de Rangueil, F-31077 Toulouse, France

ARTICLE INFO

*Article history:*
Received 11 March 2015
Received in revised form 14 July 2015
Accepted 19 July 2015
Available online 26 July 2015

*Keywords:*
Multi-objective optimization
Life cycle assessment
Environmental engineering
Expensive black-box model

ABSTRACT

Multi-objective constrained optimization problems which arise in many engineering fields often involve computationally expensive black-box model simulators of industrial processes which have to be solved with limited computational time budget, and hence limited number of simulator calls. This paper proposes two heuristic approaches aiming to build proxy problem models, solvable by computationally efficient optimization methods, in order to quickly provide a sufficiently accurate approximation of the Pareto front. The first approach builds a multi-objective mixed-integer linear programming (MO-MILP) surrogate model of the optimization problem relying on piece-wise linear approximations of objectives and constraints obtained through brute-force sensitivity computation. The second approach builds a multi-objective nonlinear programming (MO-NLP) surrogate model using curve fitting of objectives and constraints. In both approaches the desired number of approximated solutions of the Pareto front are generated by applying the $\varepsilon$-constraint method to the multi-objective surrogate problems. The proposed approaches are tested for the cost vs. life cycle assessment (LCA)-based environmental optimization of drinking water production plants. The results obtained with both approaches show that a good quality approximation of Pareto front can be obtained with a significantly smaller computational time than with a state-of-the-art metaheuristic algorithm.

## 1. Introduction

A wealth of generic mature derivative-free meta-heuristic algorithms (e.g. genetic algorithms, particle swarm optimization, differential evolution, etc.) (Zhou et al., 2011; Deb, 2014) is available nowadays. They are suitable especially for black-box multi-objective optimization (MOO) under constraints, where well-established derivative-based mathematical programming methods cannot be applied due to the unavailability of the problem formulation in analytical form. Meta-heuristic algorithms are reliable, can explore very complex optimal trade-off Pareto fronts (e.g. discontinuous, non-convex, etc.) in a single run but suffer, among other issues, from the curse of dimensionality and slow convergence in the neighbourhood of the optimum.

These algorithms can be classified into two main categories:

- *Pareto dominance-based* e.g. Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002), Strength Pareto Evolutionary Algorithm (SPEA2) (Zitzler et al., 2002), adaptive evolutionary algorithm (Hadka and Reed, 2013), etc.
- *non-Pareto dominance-based* e.g. scalar objective function problem decomposition-based algorithm (Zhang and Li, 2007), or hybrid algorithms combining global and local search (Sindhya et al., 2013).

Many of these meta-heuristic algorithms have been developed aiming to accurately approximate very complex Pareto fronts, their performance being tested using computationally inexpensive (generally unconstrained) benchmark MOO problems from the literature. Most of these algorithms have been also successfully applied for various practical engineering problems in many fields. However, in some engineering fields the black-box model evaluation can be computationally expensive while the computational time budget may be limited. Under these stringent assumptions the above mentioned well-established generic

[*] Corresponding author. Tel.: +352 275 888 1.
*E-mail address:* florin.capitanescu@list.lu (F. Capitanescu).

http://dx.doi.org/10.1016/j.compchemeng.2015.07.013
0098-1354/© 2015 Elsevier Ltd. All rights reserved.

algorithms may not complete and hence they may not provide a satisfactorily accurate and/or evenly distributed Pareto front. This class of computationally expensive MOO problems with limited number of black-box model evaluations opens the way to the development of new algorithms with different accuracy/speed trade-offs. The few existing approaches in this emerging research field rely on Gaussian stochastic process modelling (Jones et al., 1998; Knowles, 2006; Zhang et al., 2010; Zuluaga et al., 2013).

In this paper we propose two new approaches for black-box model expensive MOO problems. These approaches aim to build an approximate model of the black-box MOO problem, which can be efficiently solved, in order to generate potentially promising new solutions. The first approach builds a MO-MILP approximation of MOO problem relying on brute-force sensitivity computation, mimicking thereby first order derivative-based problem modelling. The second approach builds a MO-NLP surrogate model based on (objectives and constraints) functions curve fitting. These problem formulations take advantage of the maturity reached in fields of MILP and NLP, respectively.

The proposed approaches are illustrated, without loss of generality, for the bi-criteria (cost vs. 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 (ISO-14040, 2006) which analyzes the lifecycle (generally from raw material extraction until product end-of-life) environmental performance of any product chain but without intrinsically providing solutions for reducing the environmental impact.

The MOO of various industrial processes to additionally account for environmental constraints has been extensively investigated so far, e.g. Jacquemin et al. (2012) and Pieragostini et al. (2012). 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 (Azapagic and Clift, 1999), NLP (Gebreslassie et al., 2009), MILP (You et al., 2012), mixed-integer nonlinear programming (Guillén-Gosálbez et al., 2008; Guillén-Gosálbez and Grossmann, 2010; Yue et al., 2013; Gong and You, 2014), etc. Most of these works rely on expert solvers for the given class of optimization problem, while some develop in-house approaches (e.g. Gong and You, 2014 combines a branch-and-refine algorithm based on successive piecewise linear approximations and an exact Newton's method-based parametric algorithm). However, the LCA optimization of drinking water production plants has received little attention so far, e.g. Wallace et al. (2014) uses the Nelder-Mead local search method for the single objective optimization of a given environmental impact, (Capitanescu et al., 2015) relies on global optimizers (e.g. SPEA2 and NSGA-II) for the cost vs. LCA-based environmental impact bi-criteria optimization, and (Ahmadi and Tiruta-Barna, 2015) proposes a hybrid algorithm combining global search algorithm NSGA-II and local search algorithm COBYLA for three-criteria optimization (cost vs. LCA-based environmental impact vs. water quality).

The remaining of the paper is organized as follows. Section 2 briefly describes the water production plant simulator, called EVALEAU, formulates conceptually the MOO problem and presents the MOO – process engineering – LCA tool. Section 3 describes the proposed general methodology and the two proposed approaches for the MOO. Section 4 provides optimization results with these approaches for a realistic model of a real-world drinking water production plant. Section 5 concludes and provides directions for future work.

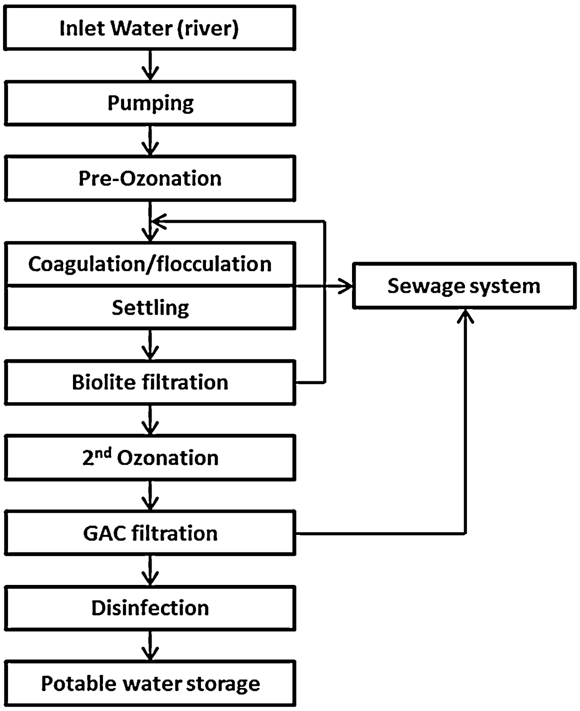


**Fig. 1.** Flowchart of the potable water production plant used in the case study.

## 2. LCA – process engineering – multi-objective optimization problem

### 2.1. EVALEAU simulator

The EVALEAU simulator is a state-of-the-art process modelling – LCA tool for prospective and retrospective simulation of potable water treatment chains (Méry et al., 2013). The simulator comprises a certain number of unit processes (UPs) for water treatment which can be combined to simulate a specific treatment chain (see Fig. 1). The simulator was developed in the LCA software Umberto®, relies on the software PHREEQC (Parkhurst and Appelo, 2013) for the model of chemical reactions in aqueous solutions, and is linked to the Ecoinvent® database (Weidema et al., 2013) for the life cycle inventory (LCI) of background processes.

"The UP modules are mathematical models consisting mainly of a set of equations defining energy and mass balances. They can represent the removal of pollutants via chemical reactions (precipitation, coagulation, and oxidation), separation (settling, filtration, etc.), elimination of microbiologic pathogens, and other treatments such as mineralization and softening to correct water quality. The modelling parameters for the different UPs include engineering design choices (i.e. device hydrodynamics, pipe diameter, efficiencies), technical issues (i.e. pumping height, backwashing schedule, filtering area, etc.), and legal restrictions (i.e. the disinfection requirement criteria)" (Ahmadi and Tiruta-Barna, 2015).

The input of EVALEAU is the inlet (river) water, which quality is described by a set of "more than hundred criteria including parameters such as temperature, pH, turbidity, UV absorbance, dissolved organic carbon, pathogenic microorganisms, inorganic compounds, micro-pollutants, and reaction products" (Ahmadi and Tiruta-Barna, 2015). The outputs of the simulator are (see Fig. 2): the outlet treated water, which quality is described by the same set of parameters as the inlet water, the operation cost of the plant, and the LCI.

EVALEAU-LCA considers the functioning stage of the plant life cycle and neglects the construction and decommissioning steps as

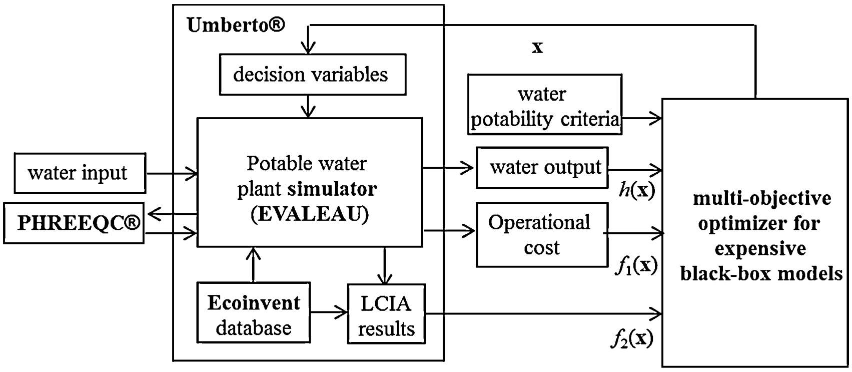

Umberto®
decision variables
x
water potability criteria
water input
Potable water plant simulator (EVALEAU)
water output
h(x)
PHREEQC®
Operational cost
f1(x)
multi-objective optimizer for expensive black-box models
Ecoinvent database
LCIA results
f2(x)

**Fig. 2.** Software architecture of the integrated tool coupling EVALEAU simulator with a hybrid MO optimizer (see Fig. 3).

explained in (Méry et al., 2013). System boundaries correspond to cradle to gate analysis, i.e. all background processes are included. The functional unit chosen is set to 1 m$^3$ of potable water at the plant.

### 2.2. MOO problem conceptual formulation

The MOO problem corresponding to a representative operating scenario of a potable water plant can be conceptually formulated as follows:

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

$$\text{s.t.} g_j(x_1,\ldots,x_n)=0, \quad j=1,\ldots,r \quad (2)$$

$$h_k(x_1,\ldots,x_n) \geq \underline{h}_k, \quad k=1,\ldots,c \quad (3)$$

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

where $\mathbf{x}=[x_1,\ldots,x_n]$ is the vector of decision variables (e.g., design and operational parameters of the various plant unit processes), the objective $f_1$ models the operational cost of the water plant (e.g., costs of raw materials, chemicals, electricity, etc.), the objective $f_2$ models the LCA-based environmental impact of the plant, both objectives expressing plant performances with respect to the functional unit chosen for LCA calculation, i.e. 1 m$^3$ of potable water produced.

Equality constraints (2) represent the set of equations which describe the steady-state input–output mass and energy flow for each unit process in the whole chain. These equations are fully modeled in (and constitute the core of) the EVALEAU simulator (see Fig. 1). Inequality constraints (3) enforce outlet water quality, according to the best water quality class (SEQEau, 2003), modeled by a set of seven major aggregated parameters.[1] Inequality constraints (4) represent physical bounds of the decision variables.

### 2.3. High level software architecture of the LCA – process engineering – multi-objective optimization tool

Note that, approaching the MOO problem (1) to (4) by means of classical mathematical programming methods is impractical if feasible at all due to the following reasons:

- the constraints (2) and (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.);
- the evaluation of all functions involved in the problem (i.e. cost, environmental impact and water quality) requires running the simulator.

To overcome these challenges we propose a solution approach, shown in Fig. 2, which couples the EVALEAU simulator with a multi-objective optimizer according to the methodology described in the next section. The optimization problem is hence split into two manageable blocks which are solved in a loop: (i) solution of equality constraints (2) obtained by running the EVALEAU simulator for a given value of decision variables **x**, and (ii) improvement of the solutions of the MOO problem (1)–(4) by the optimizer, which is fed with the values of functions $f_1(\mathbf{x}), f_2(\mathbf{x})$, and $h_k(\mathbf{x})$ already evaluated by the simulator, and returns to the latter new promising settings for the decision variables **x**.

## 3. Multi-objective constrained optimization: methodology and proposed approaches

### 3.1. General methodology

Fig. 3 shows the main steps of the proposed methodology and the main peculiar features of the two proposed approaches MO-MILP and MO-NLP. The core of the methodology is the way in which the MO surrogate optimization problem is built in both approaches.

The proposed methodology aims to improve the Pareto front estimation accuracy with limited computational budget. It consists mainly in a two-step hybrid approach coupling sequentially a surrogate model based approach (MO-MILP or MO-NLP) with a classical (global, hybrid, or local) meta-heuristic algorithm. The latter has also the role of back-up solution in case the surrogate model does not lead to a satisfactory estimation of the Pareto front. The solutions evaluated by the surrogate model approach are added to the original (randomly generated) initial population of the meta-heuristic algorithm which carries on the computations. The share of the number of functions evaluations between these two steps depends on the expected problem features and computational budget limit.

In order to ensure computational efficiency the MILP approach is to be applied as a one-shot procedure which attempts to take advantage from a small number of functions evaluations and assess whether problem features allow a fast jump close to the Pareto front. On the other hand, the MO-NLP approach can be used either as a one-shot approximated solutions generator or, in an iterative fashion, by accumulating solutions generated in previous iterations until no significant improvement is obtained or the computational

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

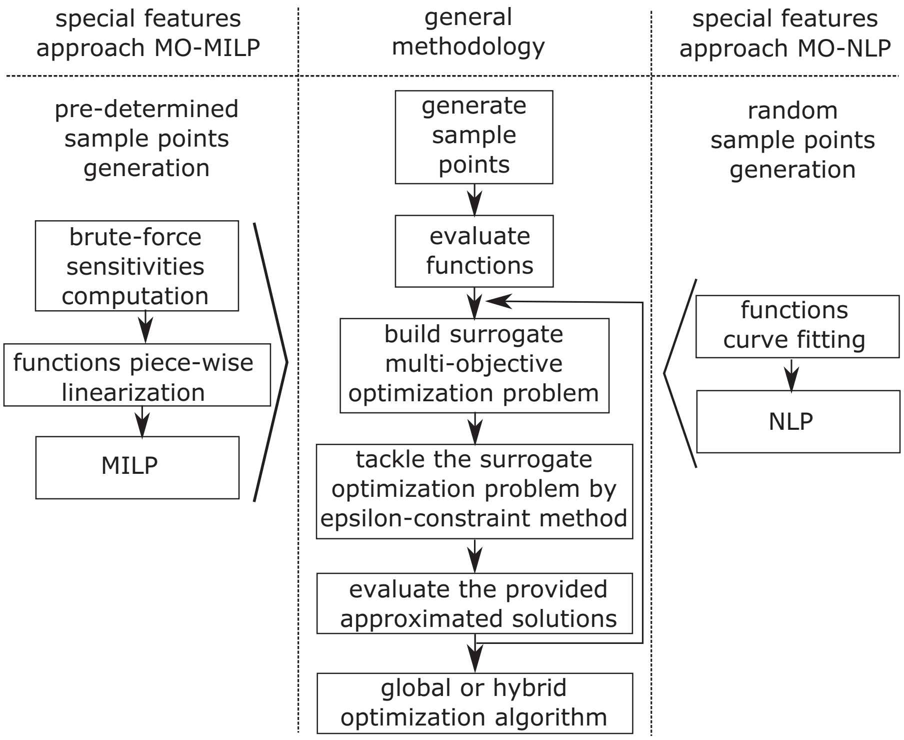

special features approach MO-MILP
general methodology
special features approach MO-NLP
pre-determined sample points generation
generate sample points
random sample points generation
brute-force sensitivities computation
evaluate functions
functions piece-wise linearization
build surrogate multi-objective optimization problem
functions curve fitting
NLP
MILP
tackle the surrogate optimization problem by epsilon-constraint method
evaluate the provided approximated solutions
global or hybrid optimization algorithm

**Fig. 3.** Proposed methodology and specific features of the approaches MO-MILP and MO-NLP.

budget limit is reached. It can also be used as a stand-alone procedure.

Each step will be described in detail hereafter.

### 3.2. *MO-MILP approach*

The main steps of this approach are the following:

1. Brute-force computation of (objectives and constraints) functions' sensitivities with respect to decision variables.
2. Formulation of the MO-MILP problem approximation relying on the sensitivity-based piece-wise linearization of the objective functions and constraints.
3. Solution of the MO-MILP problem by the $\varepsilon$-constraint method (Mavrotas, 2009).

Each step will be described in detail hereafter.

#### 3.2.1. *Piece-wise linearization of a function*

Fig. 4 sketches the piece-wise linearization of a generic function $l_k$ as a function of $i$-th decision variable $x_i$ using $p$ intervals of equal size $\Delta x_i = (\underline{x}_i^{p+1} - \underline{x}_i^1)/p$ and $p+1$ pairs $(x_i^j, l_k(x_i^j))$. In this figure, for the interval $j$, $\underline{x}_i^j$ and $\underline{x}_i^{j+1} = \underline{x}_i^j + \Delta x_i$ are its bounds and

$$s_{i,j}^{l_k} = \frac{\underline{l}_{k,i}^{j+1} - \underline{l}_{k,i}^{j}}{\underline{x}_i^{j+1} - \underline{x}_i^{j}} \tag{5}$$

is its slope.

If the value of the variable lies in this interval, then the value of the function is expressed as $l_k(x_i^j) = \underline{l}_{k,i}^j + s_{i,j}^{l_k}(x_i^j - \underline{x}_i^j)$. Each objective function and operational constraint can be approximated likewise.

#### 3.2.2. *Sensitivities computation*

Brute-force sensitivities computation of objective functions and constraints with respect to decision variables is performed using the following algorithm:

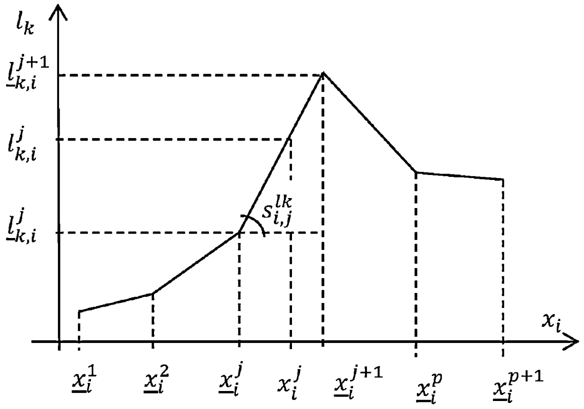


**Fig. 4.** Piece-wise linearization of a function.

1. Input parameter: the desired number $p$ of piece-wise linear approximation intervals.
2. Set decision variables to their initial[2] values ($x_i = x_i^0$, $i = 1, \ldots, n$) and run the simulator to evaluate the initial values of all functions: $f_1^0 = f_1(x_i^0)$, $f_2^0 = f_2(x_i^0)$, $h_k^0 = h_k(x_i^0)$, $k = 1, \ldots, c$.
3. For each decision variable $x_i$, $i = 1, \ldots, n$
   (a) For each point $j = 1, \ldots, p+1$
      - Let $x_i = \underline{x}_i^j = \underline{x}_i^1 + (j-1)\Delta x_i$
      - Run the simulator to evaluate $f_1(x_i)$, $f_2(x_i)$, and $h_k(x_i)$, $k = 1, \ldots, c$.
      - If $j \geq 2$ compute sensitivities: $s_{i,j-1}^{f_1} = (\underline{f}_{1,i}^j - \underline{f}_{1,i}^{j-1})/(\underline{x}_i^j - \underline{x}_i^{j-1})$,

[2] The approach requires providing an initial point $x_i^0$ around which sensitivities are computed and its performances may be sensitive to this choice. Fortunately, in many engineering problems the initial point can be naturally chosen as a realistic operating point of the process under study.

$s_{i,j-1}^{f_2} = (\underline{f}_{2,i}^{j} - \underline{f}_{2,i}^{j-1})/(\underline{x}_i^j - \underline{x}_i^{j-1})$,
$s_{i,j-1}^{h_k} = (\underline{h}_{k,i}^{j} - \underline{h}_{k,i}^{j-1})/(\underline{x}_i^j - \underline{x}_i^{j-1})$, $k = 1, \ldots, c$

- If $j = p + 1$ reset all decision variables to their initial values ($x_i = x_i^0$, $i = 1, \ldots, n$)

### 3.2.3. Formulation of the MO-MILP problem approximation

When seeking for optimal values of decision variables using piece-wise linear functions, the interval which each optimal value will belong to is not known a priori. This requires introducing one additional binary variable $b_i^j$ per decision variable $x_i$ and interval $j$, enforcing that $b_i^j = 1$ if $x_i$ lies on the interval $j$ and $b_i^j = 0$ otherwise. Furthermore, in order to keep the problem formulation linear and take advantage of existing powerful solvers, each decision variable $x_i$ is split into $j$ variables $x_i^j$ such that $x_i = x_i^j$ if $b_i^j = 1$ and $x_i^j = 0$ otherwise.

The brute-force sensitivities computation and the definition of new variables $b_i^j$ and $x_i^j$ allow approximating piece-wisely the objective $f_1$ as follows:

$$f_1(x_i^j, b_i^j) = f_1^0 + \sum_{i=1}^{n} \Delta f_{1,i} = f_1^0 + \sum_{i=1}^{n} \left[ \sum_{j=1}^{p} (b_i^j \underline{f}_{1,i}^j + s_{i,j}^{f_1}(x_i^j - b_i^j \underline{x}_i^j)) - f_1^0 \right] \quad (6)$$

Observe that in this equation if at the optimum the decision variable $x_i$ belongs to interval $q$ then only this term counts in the whole summation, i.e. $\sum_{j=1}^{p} (b_i^j \underline{f}_{1,i}^j + s_{i,j}^{f_1}(x_i^j - b_i^j \underline{x}_i^j)) = \underline{f}_{1,i}^q + s_{i,q}^{f_1}(x_i^q - b_i^q \underline{x}_i^q)$, because for other intervals $j \neq q \rightarrow b_i^j = 0 \rightarrow x_i^j = 0$ and hence $b_i^j \underline{f}_{1,i}^j + s_{i,j}^{f_1}(x_i^j - b_i^j \underline{x}_i^j) = 0$.

The bi-objective constrained MILP approximation of the problem can be formulated as follows:

$$\min_{x_i^j, b_i^j} \{f_1(x_i^j, b_i^j); f_2(x_i^j, b_i^j)\} \quad (7)$$

$$\text{s.t.} f_1(x_i^j, b_i^j) = f_1^0 + \sum_{i=1}^{n} \left[ \sum_{j=1}^{p} (b_i^j \underline{f}_{1,i}^j + s_{i,j}^{f_1}(x_i^j - b_i^j \underline{x}_i^j)) - f_1^0 \right] \quad (8)$$

$$f_2(x_i^j, b_i^j) = f_2^0 + \sum_{i=1}^{n} \left[ \sum_{j=1}^{p} (b_i^j \underline{f}_{2,i}^j + s_{i,j}^{f_2}(x_i^j - b_i^j \underline{x}_i^j)) - f_2^0 \right] \quad (9)$$

$$f_1(x_i^j, b_i^j) \leq f_1^{\max} \quad (10)$$

$$f_2(x_i^j, b_i^j) \leq f_2^{\max} \quad (11)$$

$$h_k(x_i^j, b_i^j) = h_k^0 + \sum_{i=1}^{n} \left[ \sum_{j=1}^{p} (b_i^j \underline{h}_{k,i}^j + s_{i,j}^{h_k}(x_i^j - b_i^j \underline{x}_i^j)) - h_k^0 \right] \geq \underline{h}_k, \quad k = 1, \ldots, c \quad (12)$$

$$b_i^j \underline{x}_i^j \leq x_i^j \leq b_i^j \underline{x}_i^{j+1}, \quad i = 1, \ldots, n, j = 1, \ldots, p \quad (13)$$

$$\sum_{j=1}^{p} b_i^j = 1, \quad i = 1, \ldots, n \quad (14)$$

$$b_i^j \in \{0, 1\}, \quad i = 1, \ldots, n, j = 1, \ldots, p \quad (15)$$

In this formulation constraints (8) and (9) are piece-wise linearizations of objectives $f_1$ and $f_2$, respectively, constraints (12) are piece-wise linearizations of inequality constraints, constraints (10) and (11) impose upper bounds[3] on objectives $f_1$ and $f_2$, constraint (13) imposes bounds on decision variables $x_i^j$ (i.e. if $b_i^j = 1$ then $\underline{x}_i^j \leq x_i^j \leq \underline{x}_i^{j+1}$) and enforces the variable to be set to zero if the optimal value does not belong to its corresponding interval (i.e. $b_i^j = 0 \rightarrow x_i^j = 0$), constraint (14) models the fact that each decision variable can belong to only one linearization interval, and constraint (15) models the binary variables.

Note that, compared to the MOO conceptual problem formulation (1)–(4), the above MO-MILP approximation dropped the equality constraints (2), which significantly relieve the MILP formulation. This choice is not only due to the reasons previously mentioned (e.g. unavailability in analytical form) but also to the fact as, in our case (as well as many applications[4]), the equality constraints are satisfied whatever the combinations of decision variables values within their physical range.

The MILP optimization problem (7)–(15) contains $n \times p$ binary variables $b_i^j$ and the same number of continuous variables $x_i^j$.

### 3.3. Solution of the MO-MILP problem by the ε-constraint method

To solve this bi-objective MILP problem we use the ε-constraint method (Mavrotas, 2009), which, in order to generate the desired number of solutions $m$, solves a series of $m$ single objective slightly modified MILP problems, according to the following algorithm:

1. Solve the single objective MILP problem $f_1^{\min} = \min f_1$ subject to (8), (9), (12)–(15). Set $f_2^{\max}$ to the value of the objective $f_2$ obtained at the solution of this optimization problem. Store the optimal solution.
2. Solve the single objective MILP problem $f_2^{\min} = \min f_2$ subject to (8), (9), (12)–(15). Set $f_1^{\max}$ to the value of the objective $f_1$ obtained at the solution of this optimization problem. Store the optimal solution.
3. Select the objective that will be modeled as constraint 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 distribution of the solutions on the Pareto front. Assume, without loss of generality, that the objective $f_2$ will be modeled as constraint.
4. For $i = 3, \ldots, m$, do:
   (a) Set $f_2^{\max} \leftarrow f_2^{\min} + (i-2)(f_2^{\max} - f_2^{\min})/(m-1)$ in constraint (11).
   (b) Solve the single objective MILP problem $\min f_1$ subject to (8)–(15). Store the optimal solution.

According to the methodology proposed in Fig. 3 the $m$ stored solutions obtained with this algorithm, representing the front approximation, are further checked with the EVALEAU simulator. If constraints violation occur, the right-hand-sides of violated constraints are updated proportionally with the violation amount so as to ensure a security margin preventing new violations, and the MILP problems are run again.

Note that the degree of linearity error (e.g. the ratio between functions estimated and real variations) provides useful information regarding the approximation quality and thereby indicates to what extent further optimizing the current approximated front by a meta-heuristic algorithm is expected to improve the current front.

[3] These constraints, which are not necessary for this general MILP formulation, have been included to facilitate the presentation of the MO problem approach via the ε-constraint method.

[4] Admittedly more research is needed to harness potential cases where equality constraints might not be satisfied for some combinations of decision variables values.

**Table 1**
Minimum number of evaluation points needed according to the type of function to fit (linear, simple quadratic, simple cubic, quadratic and cubic with higher order terms).

| $f(\mathbf{x}) =$ | Number of points |
|---|---|
| $\gamma + \sum_{i=1}^{n} \alpha_i x_i$ | $n+1$ |
| $\gamma + \sum_{i=1}^{n} (\alpha_i x_i^2 + \beta_i x_i)$ | $2n+1$ |
| $\gamma + \sum_{i=1}^{n} (\alpha_i x_i^3 + \beta_i x_i^2 + \delta_i x_i)$ | $3n+1$ |
| $\gamma + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{ij} x_i x_j + \sum_{i=1}^{n} \beta_i x_i$ | $0.5n^2+1.5n+1$ |
| $\gamma + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_{ijk} x_i x_j x_k + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} x_i x_j + \sum_{i=1}^{n} \delta_i x_i$ | $n^2+4n+2$ |

### 3.4. MO-NLP approach

The main steps of this approach are as follows:

1. Curve fitting of objective functions and constraints.
2. Formulation of the MO-NLP problem approximation relying on the curve fitting of objective functions and constraints.
3. Solution of the MO-NLP problem by the $\varepsilon$-constraint method described in Section 3.3.

#### 3.4.1. Functions curve fitting

A crucial aspect of curve fitting of a function is the beforehand choice of the analytical function model to be fit as a compromise between the number of required trial points (and hence the computational effort to generate these points) and expected approximation quality (because the real function is not known a priori, modeling higher order terms does not necessarily guarantee a consistent increase in accuracy). As a consequence, we seek only a reasonable approximation of every function to be used afterwards in the surrogate MOO problem. Furthermore, our time consuming simulator drastically limits the type of function to fit using a reasonably small number of trial points such as: linear, quadratic, or cubic polynomial functions (see Table 1). Accordingly, as a linear function fit may be inaccurate due to its simplicity while a cubic function requires a significant number of evaluations and may not be afforded in our context, as a compromise solution we choose to fit second order quadratic functions of the form:

$$f(\mathbf{x}) = \gamma + \sum_{i=1}^{n} (\alpha_i x_i^2 + \beta_i x_i), \tag{16}$$

where optimal values of the coefficients $\gamma$, $\beta_i$'s and $\alpha_i$'s are to be found.

Assuming a dataset ($\mathbf{x}^t$, $y^t = f(\mathbf{x}^t)$) is generated for $t$ trial points $\mathbf{x}^1, \ldots, \mathbf{x}^t$, the coefficients of the function can be found solving the following unconstrained nonlinear optimization problem:

$$\min_{\gamma, \alpha_i, \beta_i} \chi^2 = \sum_{j=1}^{t} \left( \frac{y^j - \gamma - \sum_{i=1}^{n} \left[ \alpha_i (x_i^j)^2 + \beta_i x_i^j \right]}{\sigma^j} \right)^2 \tag{17}$$

which minimizes the weighted least squares of the distances between the analytical function and the real trial values, where $\sigma^j$ is the weighting factor (mimicking the uncertainty) associated to the trial point $j$. As we will prove with numerical experiments in the next section, the choice of these weights is essential in order to obtain good results. To this end, we choose $\sigma^j = y^j$, $j = 1, \ldots, t$ so as to force the curve to better fit the best solutions at hand to the detriment of less good solutions, and thereby explore the most promising areas of the search space.

**Table 2**
Computational effort of MILP and NLP approaches for bi-criteria optimization

| Number of functions evaluations | |
|---|---|
| MO-MILP | MO-NLP |
| $n \times p + m$ | $\geq 2 \times n + 1 + m$ |
| Number of variables | |
| MILP problem | NLP approach |
| $n \times p$ binary<br>$n \times p$ continuous | $(2n+1) \times (c+2)$ continuous for curve fitting NLP<br>$n$ continuous for surrogate NLP |
| Number of constraints | |
| MILP problem | NLP approach |
| $(n+1) \times p + c + 4$ | 0 for curve fitting NLP<br>$n + c$ for surrogate NLP |

Note that the value of the objective $\chi^2$ in (17) provides a measure of the fit accuracy. If this value exceeds some predefined threshold (e.g. the simplest metric is the number of degrees of freedom $t - 2n - 1$) then the curve fit is not trusted and rejected.

#### 3.4.2. Formulation of the MO-NLP problem approximation

Once trusted approximations for objectives $f_1, f_2$, and inequality constraints $h_k$, $k = 1, \ldots, c$ have been obtained, the bi-objective constrained MO-NLP surrogate problem can be formulated in the following way:

$$\min_{x_1, \ldots, x_n} \left\{ \gamma^{f_1} + \sum_{i=1}^{n} (\alpha_i^{f_1} x_i^2 + \beta_i^{f_1} x_i); \gamma^{f_2} + \sum_{i=1}^{n} (\alpha_i^{f_2} x_i^2 + \beta_i^{f_2} x_i) \right\} \tag{18}$$

$$\text{s.t.} \gamma^{h_k} + \sum_{i=1}^{n} (\alpha_i^{h_k} x_i^2 + \beta_i^{h_k} x_i) \geq \underline{h}_k, \quad k = 1, \ldots, c, \tag{19}$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \qquad i = 1, \ldots, n \tag{20}$$

where new notations are self-explanatory.

The MO-NLP problem (18)–(20) can be reduced to a series of $m$ single objective NLP problems, via the $\varepsilon$-constraint method of Section 3.3, likewise for the MO-MILP problem. Furthermore, given the specific choice of quadratic functions fit, the NLP problem takes a particular case of quadratically constrained quadratic program (QCQP) which exhibit some computational advantages (e.g. all second order derivatives are constant) which can be better exploited by a solver.

### 3.5. Summary of problems size and computational effort of both approaches

Table 2 provides a summary of the computational effort of both MILP and NLP approaches as well as the size of involved optimization problems.

Note that the major computational effort of the MILP approach[5] is $n \times p + m$, i.e. grows almost proportionally with the number of decision variables $n$ for reasonable values of the number of linearization intervals $p$. The choice of the number of intervals $p$ used in piece-wise linear approximation is critical and should be done as a trade-off between the computation budget and the expected quality of approximation (the larger is $p$ the better the linearization

[5] For expensive black-box simulators evaluation the computational time is generally much larger than the MILP or NLP solution.

quality but the larger the computational time). For instance, using more than say 10 intervals is not expected to further significantly improve the solution quality. Furthermore, for large scale problems, the parameter $p$ should be kept small also in order to limit the combinatorial space of the MILP problem and ensure its solution in negligible time. Therefore, it would be reasonable choosing $p$ so as $2 \leq p \leq 5$, while the upper bound of this interval could be further limited by the overall computational budget and the share of this budget between the MILP approach and the meta-heuristic algorithm.

The computational time of NLP approach scales also basically linearly with the decision variables number, and in general requires a comparable number of evaluations with MILP approach provided that the latter uses a small number of linearization intervals $p$. The effort of NLP problem (both curve fitting and NLP surrogate) could cause issues for large scale problems, especially if the number of functions to be fit $c+2$ becomes significant.

## 4. Numerical experiments

### 4.1. Brief description of the studied water plant and simulation tools

The proposed optimization approaches are applied to a realistic model of an existing French water treatment plant. 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 2.70 GHz/8 GB 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 metric 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 LCIA method Goedkoop et al. (2009).

The functional unit is 1 $m^3$ of treated water while monetary units (m.u.) stand for euro.

### 4.2. Experiments assumptions

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

- In order to assess the various approaches scaling with the problem size we use two sets of decision variables:
  - the *original set* which comprising 18 continuous variables;
  - a *subset* of the original set, comprising 6 most efficient variables.
- One requires approximating the Pareto front by 24 solutions.
- As baseline for performance comparison of the proposed approaches we use the well-known benchmark elitist global meta-heuristic evolutionary algorithm SPEA2 (Zitzler et al., 2002). The algorithm is run with the default values proposed on PISA platform (Bleuler et al., 2003), using a population of 24 individuals. Because, given the different features of evolutionary algorithms and proposed approach, a comparison of approximated front after exactly the same number of evaluations is generally not possible, we will use for the SPEA2 algorithm a slightly larger number of evaluations so as to enable it completing the evaluation of a certain generation.
- Three comparison criteria[6] 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 approaches, according to the methodology in Fig. 3.
- MILP problems are solved via the GLPKv4.55 solver (Makhorin, 2014).
- curve fitting unconstrained optimization problem (17) is solved via the "curve_fit" Python function while NLP problem is solved by the IPOPT solver (Wächter and Biegler, 2006).

### 4.3. MO-MILP approach

#### 4.3.1. Performances of the approach and impact of the number of linearization intervals

The MILP approach depends on the beforehand chosen number of linearization intervals. The latter may reasonably vary between 2 and 10 per decision variable, which means 3–11 functions evaluations[7] per variable. In order to assess the influence of the number of linearization intervals on the front approximation quality, we performed the MILP approach for 2–5, and 10 intervals used for linearization. We observed that, as somewhat expected, the larger the number of intervals, the better the front quality. However, because in this example the improvement obtained as the number of linearization intervals grows is not significant, for the sake of figures readability, the following figures plot only the two extreme cases, obtained for 2 intervals and 10 intervals, respectively.

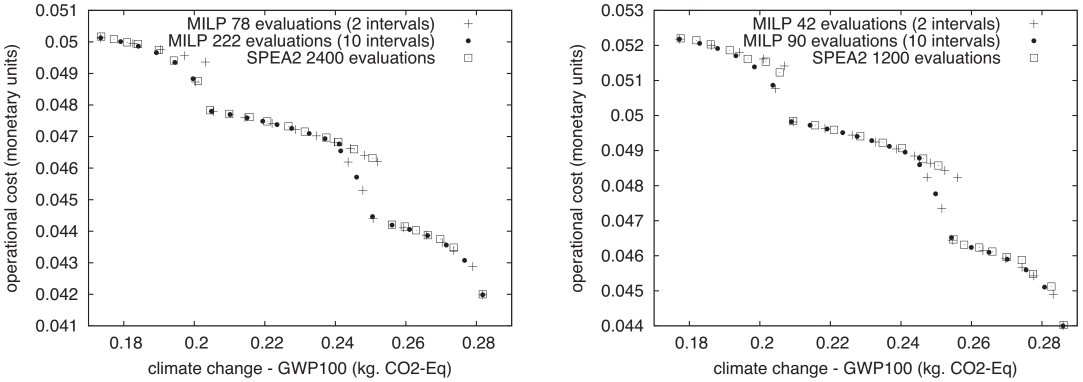
To aid ascertaining the trade-off between the degree of sub-optimality and the computational effort of the proposed approach, Fig. 5 shows the Pareto front approximations obtained with both the MILP approach and SPEA2 for the two sets of decision variables considered. One can observe that the MILP approach with 10 linearization intervals produces in both cases only non-dominated solutions which are furthermore well distributed. It also slightly outperforms SPEA2 in terms of both solutions dominance and even spread. The figures also show that the MILP approach with 2 linearization intervals provides mostly Pareto-optimal solutions with well spread but also a few dominated solutions. Its performances are only slightly inferior to those provided by MILP approach with 10 linearization intervals but comparable with SPEA2. An explanation why the MILP approach works very well in this case is that the Pareto set contains solutions where most variables reach a physical bound.

In terms of computational effort the MILP approach with 2 intervals and 10 intervals, respectively, is roughly around 30 and 10, respectively, times faster than the SPEA2 algorithm for a comparable front approximation quality, which justifies it as a promising approach for MOO of computationally intensive black-box model problems.

Note that, compared to one EVALEAU simulator run for functions evaluation which takes in average around 120 s, the overall effort of solving to optimality the MO-MILP formulation for optimization part is negligible (e.g. it takes less than 10 s to produce 24 solutions on the front). As it will be shown later SPEA2 algorithm produces already a good number of non-dominated Pareto solutions after around 720 and 480 evaluations for the set of 18 and

[6] Note that front comparison metrics such as hypervolume indicator or generational distance are not used as the mere front visualization allows easily concluding.

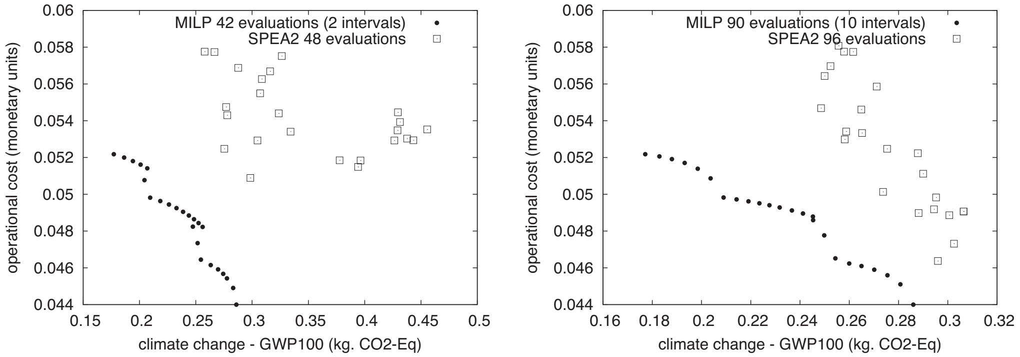
[7] Recall that e.g. 78 evaluations in the MILP approach (see Fig. 5(a)) correspond to 18 (variables) × 3 (evaluation points for 2 intervals) + 24 (required Pareto solutions).



(a) 18 decision variables

(b) 6 decision variables

**Fig. 5.** Cost vs. environmental impact optimization: MILP approximation vs. SPEA2.



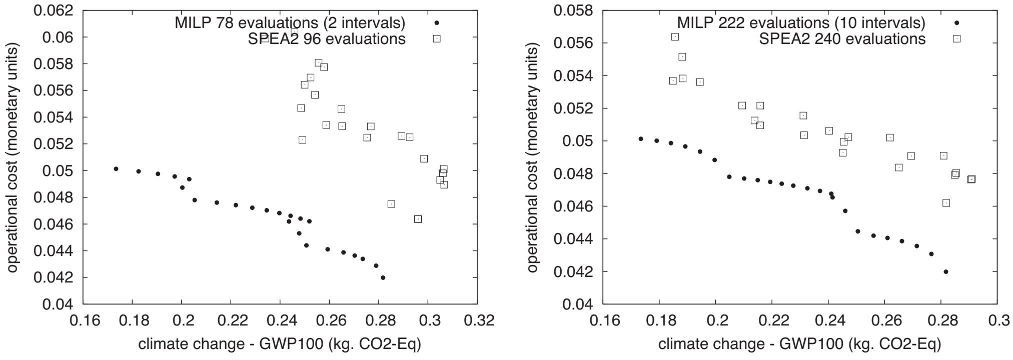
(a) 2 intervals

(b) 10 intervals

**Fig. 6.** MILP approximation vs. SPEA2: front approximation according to criterion C1 (6 decision variables).

6 decision variables, respectively, the remaining evaluations being used mostly to improve the solutions distribution.

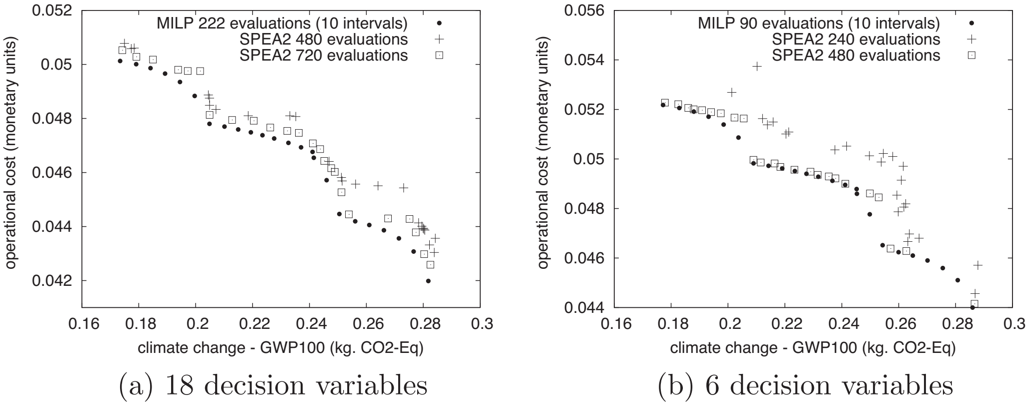
The initial water plant operating point corresponds to a cost of 0.058 m.u. per 1 $m^3$ of water treated and an environmental impact of 0.216 kg. $CO_2$-Eq per 1 $m^3$ of water treated. In terms of water plant performance with respect to the initial operating point, which corresponds to a cost of 0.058 m.u. per 1 $m^3$ of water treated and an environmental impact of 0.216 kg. $CO_2$-Eq per 1 $m^3$ of water treated, the optimal solution with the lowest environmental impact (see Fig. 5(a)) leads to a reduction of operational cost and GWP100 indicator of around 13% and 20%, respectively. Assuming that the water plant treats 50,000 $m^3$ per day the cost



(a) 2 intervals

(b) 10 intervals

**Fig. 7.** MILP approximation vs. SPEA2: front approximation according to criterion C1 (18 decision variables).



(a) 18 decision variables

(b) 6 decision variables

**Fig. 8.** MILP approximation (with 10 linearization intervals) vs. SPEA2 according to criterion C2.

difference between the initial operating point and the least cost operating point is roughly 0.058−0.042 = 0.016 euro/$m^3$ of treated water, which represents an important cost reduction of 28% and corresponds to savings of 800 euro per day and 292,000 euro per year.

### *4.3.2. MILP approach vs. SPEA2*

Figs. 6 and 7 show, for both the original set and its subset of decision variables, the front approximations obtained with both the MILP approach (for 2 and 10 linearization intervals) and SPEA2, according to criterion C1. Note that, despite a slightly larger number of evaluations, due to its evolutionary algorithm nature, SPEA2 is at an early stage of the search, its solutions looking more like "cloud points" and lagging far away behind the very good front approximations provided by the MILP approach.

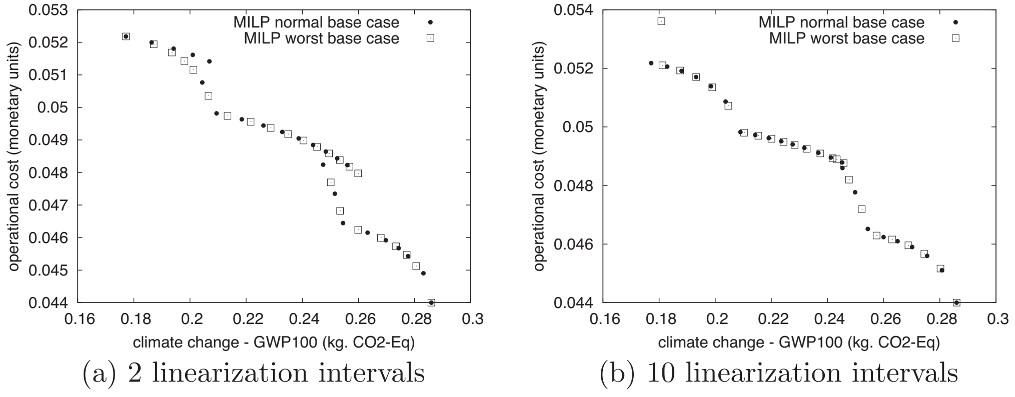
Fig. 5 allows also comparing according to criterion C2 (i.e. when both generated fronts have quasi-equivalent quality) the front approximations obtained with the MILP approach and SPEA2. One can observe that increasing three times the size of the decision variables set leads to an increase of roughly 1.8–2.5 times of the number of evaluations in MILP approach and of around two times for the SPEA2 algorithm. Fig. 8 displays additional supporting information with SPEA2 solutions evolution, showing that, the most computational effort is in ensuring a good solutions spread, since from the Fig. 8(b) one sees that after 480 evaluations most SPEA2 solutions are Pareto-optimal but their spread is not satisfactory. Also, for an evolutionary algorithm, the larger the number of decision variables, the slower the convergence to the Pareto front (e.g. after 480 evaluations, the SPEA2 front is much closer to the Pareto front for 6 variables than for 18). Fig. 8(a) shows that after 720 evaluations SPEA2 solutions are consistently suboptimal.

Note that, a comparison MILP vs. SPEA2 according to C3 criterion is not necessary since the MILP converges practically to the Pareto front.

The analyses of Figs. 5–8 allow us to safely conclude that the MILP approach is a promising candidate, with a very good accuracy/speed trade-off in our case, for MOO problems with expensive functions evaluations.

### *4.3.3. Influence of the base case*

Fig. 9 illustrates, for the subset of 6 decision variables, the impact of the initial operating point on the quality of results obtained with the MILP approach. To this end the base case with the worst values of both objectives was chosen, corresponding to 0.093 m.u. and 1.12 kg. $CO_2$-Eq, respectively. The figure shows that, for our problem, the influence of the initial point is not significant, for both number of intervals. For the variant with 10 intervals the solution corresponding to the minimum environmental impact is



(a) 2 linearization intervals

(b) 10 linearization intervals

**Fig. 9.** Impact of initial point in MILP: normal base case vs. worst base case.

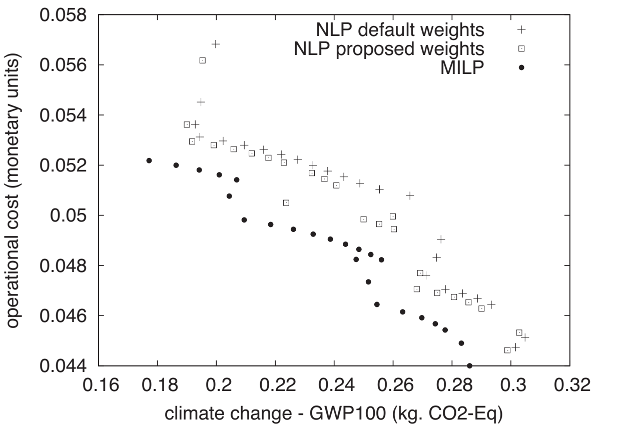


**Fig. 10.** MO-NLP: impact of weights and comparison with MILP (2 intervals, 42 evaluations).

less good for the worst base case due to a major shift in the MILP solution compared to other solutions.

### 4.4. *MO-NLP approach*

In this section we restrict ourselves to the subset of 6 decision variables.

#### 4.4.1. *Choice of curve fitting weights and comparison with MILP approach*

Let us first consider as input the 18 points obtained with MILP approach which considers 2 linearization intervals. The MILP approach provides already a good approximation of the Pareto front, as shown in Fig. 5(b), and displayed in Fig. 10 as a base of comparison.

A key aspect for improving the performance of the NLP approach is the choice of weights associated with the $t$ trial points according to the curve fitting optimization problem (17).

Fig. 10 shows that, compared to the MILP approach, a systematically less good approximation of the Pareto front is obtained using the default weights, i.e. $\sigma^j = 1$, $j = 1, \ldots, t$ in (17). Note, however, that NLP approximation is still better than SPEA2 results after the same number of evaluations (see Fig. 6(a)). On the other hand, the use of proposed set of weights, i.e. $\sigma^j = y^j$, $j = 1, \ldots, t$ leads to a certain improvement of the NLP approximation, the results being only slightly inferior to the MILP approach.

#### 4.4.2. *Starting from randomly generated solutions*

The main advantage of the NLP approach compared to the MILP one is the ability to computationally efficiently[8] take advantage of a given set of (random) solutions at hand. We illustrate this feature of the NLP approach in an iterative fashion starting 24 random points of the initial generation of SPEA2 algorithm.

For the sake of readability the solutions generated with the NLP approach is limited to three iterations and are presented in Figs. 11 and 12.

Fig. 11 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 distinct solutions instead of 24, they are practically enveloping the initial random points and are located closer to the Pareto front.

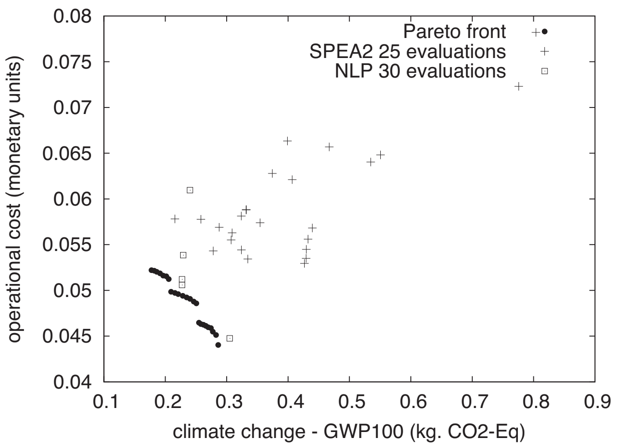


**Fig. 11.** MO-NLP approximated front for random starting solutions.

Fig. 12 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 the sake of readability, from the fourth iteration on the front improves little, emphasizing the limitation of the NLP approach as a stand-alone technique.

#### 4.4.3. *MO-NLP vs. SPEA2*

Fig. 13(a) and (b) allow comparing the MO-NLP and SPEA2 results, according to criteria C1 and C2. One can observe that MO-NLP outperforms SPEA2 in both cases, e.g. allowing saving around 2.5 evaluations in order to achieve a front of comparable quality according to criteria C2.

### 4.5. *MO-NLP as an input for SPEA2*

Fig. 14 plots the approximated fronts obtained with both SPEA2 algorithm, initialized with default settings (see Fig. 11), and the proposed hybrid MO-NLP-SPEA2, where MO-NLP approach starts from the same initial population as SPEA2 and, after performing

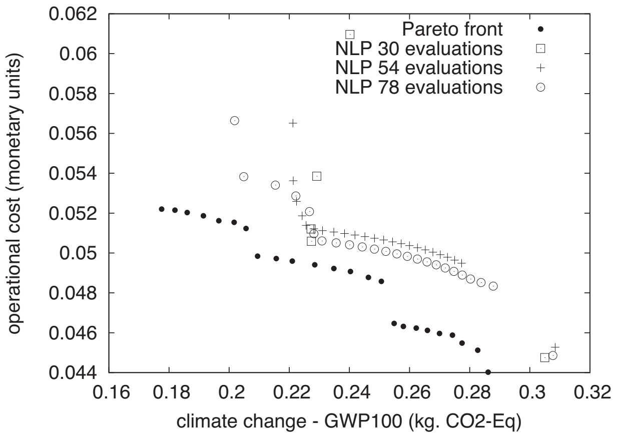
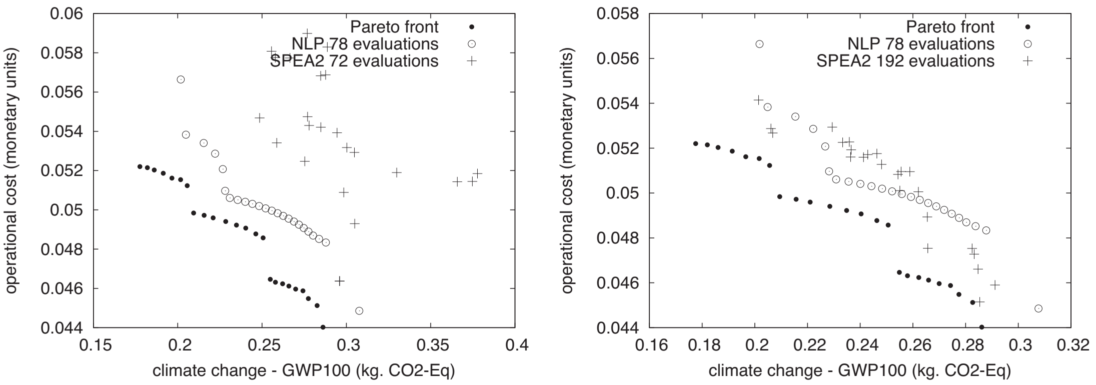


**Fig. 12.** MO-NLP: approximated front in the first three iterations.

[8] Indeed, applying the fastest MILP variant (i.e. with 2 intervals) for 24 random initial solutions would require a computational effort of 24 × 42 = 1008 evaluations, which may not be afforded in some computational budget context.



(a) criterion C1

(b) criterion C2

**Fig. 13.** MO-NLP vs. SPEA2: front approximation according to criteria C1 and C2.

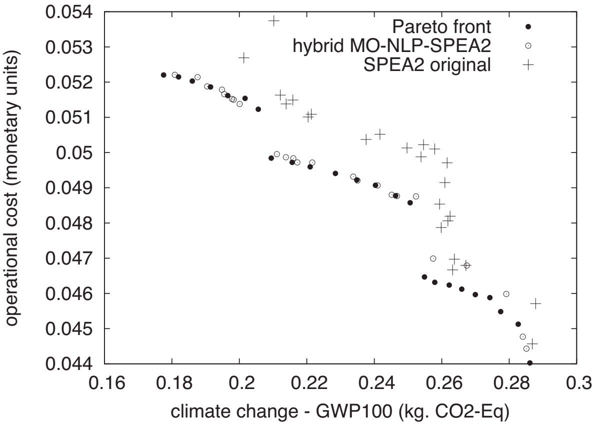


**Fig. 14.** SPEA2 vs. hybrid MO-NLP-SPEA2: front approximation (after 240 evaluations) according to criterion C3.

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 evolutionary algorithms to speed up the convergence of the latter.

## 5. Conclusions and future work

This paper has proposed and compared the performances of two approaches for solving expensive black-box model multi-objective constrained optimization problems. These approaches, aimed to quickly provide a sufficiently accurate approximation of the Pareto front, rely on MILP and NLP surrogate models, taking thereby advantage of powerful existing solvers able to solve such proxy models with negligible computational effort compared to a single simulator evaluation.

Numerical results have shown that both approaches compare favorably with classical state-of-the-art meta-heuristic algorithm SPEA2 in terms of front quality approximation with limited computational time budget. In other words, compared to SPEA2, the proposed approaches allow saving a significant number of evaluations to achieve sub-optimal solutions of comparable accuracy. In our tests, the MILP approach outperforms the NLP one in terms of solutions quality, providing solutions most of which are either practically Pareto-optimal or are located very close to the Pareto front. However, as in order to ensure computational efficiency, the MILP approach neglects potential nonlinear interactions between decision variables, it may be less efficient in a context where such interactions are strong. Another slight drawback of the MILP approach is the fact that it generates trial solutions in a rigid predetermined way so as to approximate sensitivities in a brute-force fashion. On the other hand, the NLP approach is an anytime iterative versatile technique which can take advantage of the set of solutions at hand and hence can take into account implicitly such nonlinear coupling effects. Therefore, future work is planned to couple the NLP approach with a state-of-the-art metaheuristic algorithm. Depending on the computational time budget and expected problem nonlinear features, the proposed approaches can serve as an input for a classical meta-heuristic algorithm aimed to speed up its convergence. Supported by the obtained results one expects a significant improvement if the current trial solutions are located far from the Pareto front. On the other hand, due to the approximations used, and likewise for evolutionary algorithms, if the trial solutions are located close to the Pareto front the improvement is expectedly insignificant. Therefore the approaches are suitable in early stages of the solution space search.

Although the proposed approaches have been tested for the cost vs. (life cycle assessment-based) environmental impact optimization of drinking water production plants at planning stage, they are generic to other application fields dealing with expensive multi-objective optimization problems, as concerns the problem formulation and operation context (e.g. from operation planning until close to real-time).

## Acknowledgement

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

## References

Ahmadi A, Tiruta-Barna L. A process modelling – life cycle assessment – multi-objective optimization (PM-LCA-MOO) tool for the eco-design of conventional treatment processes of potable water. J Clean Prod 2015;100:116–25.

Azapagic A, Clift R. The application of life cycle assessment to process optimization. Comput Chem Eng 1999;10:1509–26.

Bleuler S, Laumanns M, Thiele L, Zitzler E. PISA – a platform and programming language independent interface for search algorithms. In: Evolutionary multi-criterion optimization. Springer; 2003. p. 494–508.

Capitanescu F, Igos E, Marvuglia A, Benetto E. Coupling multi-objective constrained optimization, life cycle assessment, and detailed process simulation for potable water treatment chains. J Environ Account Manag 2015;3:213–24.

Deb K. Multi-objective optimization. In: Search methodologies. Springer; 2014. p. 403–49.

Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 2002;6:182–97.

Gebreslassie BH, Guillén-Gosálbez G, Jiménez L, Boer D. Design of environmentally conscious absorption cooling systems via multi-objective optimization and life cycle assessment. Appl Energy 2009;86:1712–22.

Goedkoop M, Heijungs R, Huijbregts M, De Schryver A, Struijs J, van Zelm R. Recipe 2008. In: A life cycle impact assessment method which comprises harmonised category indicators at the midpoint and the endpoint level 1; 2009.

Gong J, You F. Global optimization for sustainable design and synthesis of algae processing network for $CO_2$ mitigation and biofuel production using life cycle optimization. AIChE J 2014;60:3195–210.

Guillén-Gosálbez G, Caballero JA, Jiménez L. Application of life cycle assessment to the structural optimization of process flowsheets. Ind Eng Chem Res 2008;47:777–89.

Guillén-Gosálbez G, Grossmann I. A global optimization strategy for the environmentally conscious design of chemical supply chains under uncertainty in the damage assessment model. Comput Chem Eng 2010;34:42–58.

Hadka D, Reed P. Borg: an auto-adaptive many-objective evolutionary computing framework. Evolut Comput 2013;21:231–59.

ISO-14040. Environmental management – life cycle assessment – principles and framework. Geneva, Switzerland: International Organization for Standardization; 2006.

Jacquemin L, Pontalier P-Y, Sablayrolles C. Life cycle assessment (LCA) applied to the process industry: a review. Int J Life Cycle Assess 2012;17:1028–41.

Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Glob Optim 1998;13:455–92.

Knowles J. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evolut Comput 2006;10:50–66.

Makhorin A. GLPK (gnu linear programming kit) reference manual version 4.55; 2014.

Mavrotas G. Effective implementation of the $\varepsilon$-constraint method in multi-objective mathematical programming problems. Appl Math Comput 2009;213:455–65.

Méry Y, Tiruta-Barna L, Benetto E, Baudin I. An integrated "process modelling-life cycle assessment" tool for the assessment and design of water treatment processes. Int J Life Cycle Assess 2013;18:1062–70.

Parkhurst DL, Appelo C. Description of input and examples for PHREEQC Version 3-a computer program for speciation, batch-reaction, one-dimensional transport and inverse geochemical calculations; 2013.

Pieragostini C, Mussati MC, Aguirre P. On process optimization considering LCA methodology. J Environ Manag 2012;96:43–54.

SEQEau. Water quality evaluation system in France (version 2). Technical report; 2003 (in French) http://sierm.eaurmc.fr/eaux-superficielles/fichiers-telechargeables/grilles-seq-eau-v2.pdf

Sindhya K, Miettinen K, Deb K. A hybrid framework for evolutionary multi-objective optimization. IEEE Trans Evolut Comput 2013;17:495–511.

Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Progr 2006;106:25–57.

Wallace RJ, Marvuglia A, Benetto E, Tiruta-Barna L. A new approach to optimization with life cycle assessment: combining optimization with detailed process simulation. In: Computational science and its applications-ICCSA 2014. Springer; 2014. p. 707–20.

Weidema BP, Bauer C, Hischier R, Mutel C, Nemecek T, Reinhard J, et al. Overview and methodology: data quality guideline for the ecoinvent database version 3. Technical report Swiss Centre for Life Cycle Inventories; 2013.

You F, Tao L, Graziano DJ, Snyder SW. Optimal design of sustainable cellulosic biofuel supply chains: multiobjective optimization coupled with life cycle assessment and input–output analysis. AIChE J 2012;58:1157–80.

Yue D, Kim MA, You F. Design of sustainable product systems and supply chains with life cycle optimization based on functional unit: general modeling framework, mixed-integer nonlinear programming algorithms and case study on hydrocarbon biofuels. ACS Sustain Chem Eng 2013;1:1003–14.

Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evolut Comput 2007;11:712–31.

Zhang Q, Liu W, Tsang E, Virginas B. Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE Trans Evolut Comput 2010;14:456–74.

Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q. Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evolut Comput 2011;1:32–49.

Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Evolutionary methods for design, optimization, and control; 2002. p. 95–100.

Zuluaga M, Sergent G, Krause A, Püschel M. Active learning for multi-objective optimization. In: Proceedings of the 30th international conference on machine learning; 2013. p. 462–70.