Computers and Chemical Engineering 67 (2014) 137–147

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

MILP-based decomposition algorithm for dimensionality reduction in multi-objective optimization: Application to environmental and systems biology problems

Pedro J. Copado-Méndez, Gonzalo Guillén-Gosálbez *, Laureano Jiménez

Departament d'Enginyeria Quimica, Universitat Rovira i Virgili, Avinguda Països Catalans 26, 43007 Tarragona, Spain

ARTICLE INFO

Article history:
Received 8 January 2014
Received in revised form 30 March 2014
Accepted 7 April 2014
Available online 18 April 2014

Keywords:
Dimensionality reduction
Multi-objective optimization
Enviromental system biology
Divide-and-conquer
MILP
MOO

ABSTRACT

Multi-objective optimization has recently gained wider interest in different domains of engineering and science. One major limitation of this approach is that its complexity grows rapidly as we increase the number of objectives. This work proposes a computational framework to reduce the dimensionality of multi-objective optimization (MOO) problems that identifies and eliminates in a systematic manner redundant criteria from the mathematical model. The method proposed builds on a mixed-integer linear programming (MILP) formulation introduced in a previous work by the authors. We illustrate the capabilities of our approach by its application to two case studies: the design of supply chains for hydrogen production and the multi-objective optimization of metabolic networks. Numerical examples show that our method outperforms other existing algorithms for dimensionality reduction.

1. Introduction

Multi-objective optimization (MOO) is widely used in many areas of science and engineering for simultaneously optimizing several objective functions subject to some equality and inequality constraints. With the recent advances in software packages and optimization theory, MOO has gained wider interest, being nowadays increasingly used in process systems engineering (PSE). MOO has been applied to several fields like the mean-risk multi-stage capacity investment problem (Claro, 2012), portfolio selection (Ustun, 2012), laser cutting of thin sheets of aluminium alloys (Sharma, 2012), power planning problems in power networks (Alonso, 2012) and reconfiguration problems in distribution feeders (Niknam and Fard, 2012), among others.

Unfortunately, the complexity of MOO grows rapidly in size with the number of objectives due to two main reasons (see Deb and Saxena (2005), for further details). First, the solution techniques available for MOO are rather sensitive to the number of objectives. Second, even if we could generate a sufficiently large number of Pareto solutions in an efficient manner, there is still the issue of visualizing and analysing them. To avoid these limitations, most MOO models restrict the optimization task to two or three objectives (López Jaimes et al., 2009). These approaches either omit objectives, or aggregate them into single metrics defined by attaching weights to them. Both approaches are inadequate, as they change the dominance structure of the problem in a manner such that they might eliminate solutions that are optimal in the original space of objectives. Dimensionality reduction techniques aim at overcoming these limitations by identifying redundant objectives that can be omitted while still preserving the problem structure to the extent possible. Deb and Saxena (2005) proposed a method based on principal component analysis for reducing the number of objectives in MOO. Brockhoff and Zitzler (2006) introduced the concept of delta error, an approximation error that arises after removing objectives in a multi-objective optimization problem. These authors formally stated the following two problems: (1) computing a minimum objective subset (MOSS) of a multi-objective problem that does not exceed a certain approximation error (denoted as the δ-MOSS problem); and (2) identifying a minimum objective subset of size k with minimum approximation error (k-MOSS problem). The authors presented both an exact and an approximation algorithm to tackle these problems and applied them to several case studies, showing that substantial dimensionality reductions are possible while still preserving to a large extent the problem structure. Based on these ideas, Guillén-Gosálbez (2011a) introduced an approach for

* Corresponding author.
E-mail addresses: pedrojesus.copado@urv.cat (P.J. Copado-Méndez), gonzalo.guillen@urv.cat (G. Guillén-Gosálbez), laureano.jimenez@urv.cat (L. Jiménez).

http://dx.doi.org/10.1016/j.compchemeng.2014.04.003
0098-1354/

dimensionality reduction based on an MILP that solves both the k-MOSS and δ-MOSS problems, and which takes advantage of the powerful branch-and-cut algorithms available for MILPs. More recently, Thoai (2011) proposed an approach to reduce the number of criteria as well as the dimension of a linear MOO problem using the concept of representative and extreme criteria, while López Jaimes et al. (2009) introduced two algorithms for objective reduction in MOO based on a feature selection technique.

In this paper, we present a new decomposition approach for objective reduction that is shown from numerical examples to outperform other existing algorithms Brockhoff and Zitzler (2006). We apply our algorithm to MOO problems arising in systems biology and sustainable engineering, showing how significant reductions in problem size can be attained without changing too much the problem structure. The article is organized as follows. In Section 2, we introduce the main concepts and theory behind objective reduction. We then describe the MILP formulation for dimensionality reduction proposed in a previous contribution and discuss its computational limitations (Section 3). Our computational framework, which is based on this MILP, follows (Section 4). The capabilities of our approach are tested next (Section 5), while in the final section (Section 6) the conclusions of the work are drawn.

2. Background

In the next subsections, we present the main concepts behind dimensionality reduction in MOO. We follow the work by Brockhoff and Zitzler (2006a), in which the reader will find further details on this topic.

Let $MO(x)$ be a multi-objective minimization problem of the following form:

$$MO(X) = \min_x \{f(x) := (f_1(x), \ldots, f_k(x)) : x \in X\} \quad (1)$$

with k objective functions $f_i := X \longrightarrow \mathbb{R}$, $1 \le i \le k$, where each objective function f_i maps a solution $x \in X$ to a value of the function vector $f := (f_1, \ldots, f_k)$.

We consider the weak Pareto dominance relationship defined as follows: $\preceq_{F'} := \{(x, y) \mid x, y \in X \wedge \forall f_i \in F' : f_i(x) \le f_i(y)\}$, where F' is a set of objectives with $F' \subseteq F := \{f_1, \ldots, f_k\}$. The following definitions are used in our analysis.

Definition. x weakly dominates y ($x \preceq_{F'} y$) with respect to the objective set F' if $(x, y) \in \preceq_{F'}$.

Definition. $x^* \in X$ is called Pareto optimal if there is no other $x \in X$ that dominates x^* with respect to the set of all objectives.

In this work we present an algorithm to reduce the dimension (i.e., number of objectives) of $MO(X)$. Given a set of objectives $F := (f_1, \ldots, f_k)$, our goal is to determine a subset F' of F ($F' \subseteq F$) of given cardinality such that the error of removing the objectives not included in F' is minimum. The main ideas underlying dimensionality reduction are illustrated next by means of a simple example.

2.1. Illustrative example: objective reduction in environmental problems

Consider the 3 Pareto optimal solutions depicted in Fig. 1 that minimize 4 different objectives: (f_1, f_2, f_3, f_4). This figure is a parallel coordinates plot (Purshouse and Fleming, 2003) that depicts in the x axis the set of objectives and in the y axis the normalized value attained by each solution. Each line in the figure represents a different Pareto solution. As seen, all of these lines intersect in at least one point, since no solution is dominated by any of the others.

As observed, two objectives can be omitted (i.e., objectives 2 and 3) without changing the dominance structure. This is because $x \preceq_{f_1, f_4} y$ is satisfied if and only if $x \preceq_{f_1 f_2 f_4 f_4} y$ is satisfied. Further reductions are not possible without modifying the dominance structure. For instance, if we remove f_1 and f_4, then $x_2 \preceq_{f_2, f_3} x_1$ is satisfied although $x_2 \not\preceq_{f_1, f_2, f_3, f_4} x_1$. As seen, solution x_2 would dominate x_1 in the original 4-dimensional Pareto space $\{f_1, f_2, f_3, f_4\}$ if it showed the same value of f_4 than x_1. The difference between the true value of f_4 in x_2 and that required to dominate x_1 in the original space of objectives can be used as a measure to quantify the change in the dominance structure. For this example, this value, referred to as δ value by Brockhoff and Zitzler, is 0.9. The delta value quantifies the change in the dominance structure of a MOO problem that takes place after removing objectives. Our goal is to determine subsets of objectives of a MOO model that minimize the δ value. This problem

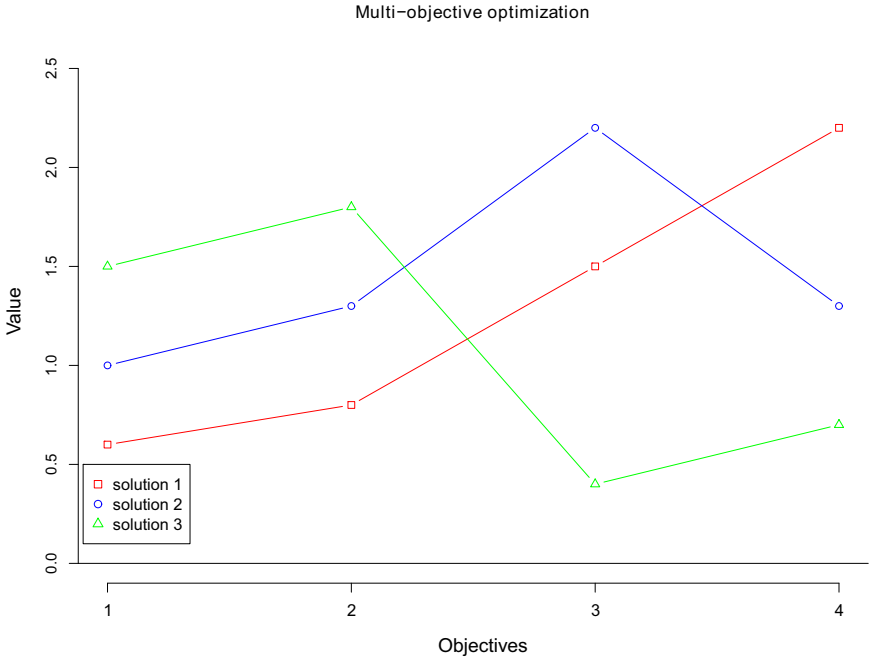

Fig. 1. Illustrative example of dominance structure.

was formally defined in a pioneering work by Brockhoff and Zitzler (2006a). Alternatively, we may also be interested in calculating the minimum number of objectives for a given allowable approximation error. We will refer to these problems as δ-MOSS and k-MOSS problems, respectively. The sections that follow describe in detail our algorithmic approach to tackle them.

3. Mathematical model

A computational framework is proposed for the efficient solution of the k-MOSS and δ-MOSS problems. Our approach builds upon the MILP for dimensionality reduction introduced by Guillén-Gosálbez (2011a). We provide first an overview of this MILP before presenting our methodology for dimensionality reduction (further details on this MILP can be found in the original article).

Given a *MOO* problem where a set of k objective functions is minimized, we aim at determining an objective subset of given size (i.e., $|F'| = j$) such that the dominance structure is preserved with a minimal value of δ. For this subset F', the dominance structure of both, the original and reduced sets of objectives, will be the same except for an error equal to δ.

We use the following notation. The parameter $OF(s, i)$ denotes the value of the i objective in solution s. The binary parameter $YP(s', s, i)$ takes the value of 1 if solution s' is better than solution s in objective i (i.e., $OF(s, i) \geq OF(s', i)$) and 0 otherwise. The binary variable $ZO(i)$ is 1 if objective i is removed and 0 otherwise, while the binary variable $ZD(s, s')$ takes the value of 1 if solution s' dominates solution s in the space resulting from removing the objectives i for which $ZO(i) = 1$ (reduced Pareto space) and it is 0 otherwise.

Fig. 2 illustrates an example of this notation. Variable $ZD(s, s')$ is determined by means of the following constraints:

$$\left(k - \sum_i ZO(i)\right) - k(1 - ZD(s, s')) \leq \sum_i YP(s', s, i)(1 - ZO(i)) \leq \left(k - \sum_i ZO(i)\right) + k(1 - ZD(s, s')) \quad \forall s \neq s' \tag{2}$$

$$\sum_i YP(s', s, i)(1 - ZO(i)) \leq \left(k - \sum_i ZO(i)\right) - 1 + kZD(s, s') \quad \forall s \neq s' \tag{3}$$

Solution s' dominates s in the reduced space, if and only if it is better than s in all of the objectives kept. Therefore, if s' dominates s, then $YP(s', s, i)$ will be equal to 1 for all of the objectives for which $ZO(i) = 0$, and the summation of $YP(s', s, i)$ will equal the number of objectives kept in the reduced space. By adding constraint (2), we ensure that this will hold if $ZD(s, s')$ is 1. On the other hand, if solution s' does not dominate s, then there will be objectives in which s will be better than s' and others in which the opposite will hold. Consequently, the term $YP(s', s, i)(1 - ZO(i))$ will be necessarily lower than the cardinality of the set of objectives kept, and constraint (3) will force the binary variable $ZD(s, s')$ to take the value of 0.

Constraint (4) specifies the total number of objectives to be omitted:

$$\sum_i (ZO(i)) = OB \tag{4}$$

Note that the approximation error increases with larger values of OB. The continuous variables $\delta(s, s', i)$ quantifies the error of removing objectives. We define this variable via constraint (5):

$$(OF(s', i) - OF(s, i))ZOD(i, s, s') = \delta(s, s', i) \quad \forall i, s \neq s' \tag{5}$$

In which $ZOD(i, s, s')$ is defined via the following constraints:

$$ZOD(i, s, s') \leq ZO(i) \quad \forall i, s \neq s' \tag{6}$$

$$ZOD(i, s, s') \leq ZD(s, s') \quad \forall i, s, s', s \neq s' \tag{7}$$

$$ZOD(i, s, s') \geq ZO(i) + ZD(s, s') - 1 \quad \forall i, s \neq s' \tag{8}$$

As observed, the value of $\delta(s, s', i)$ is determined only for those solutions s dominated by at least another solution s' in the reduced space of objectives, and only for the omitted objectives i. On the other hand, constraint (5) forces variable $\delta(s, s', i)$ to take a 0 value when s is Pareto optimal in the reduced space and i is a non-omitted objective. Note that in the latter case variable $ZOD(i, s, s')$ will take a 0 value, making $\delta(s, s', i)$ equal to zero.

The model seeks to minimize the maximum error of omitting objectives. The overall mathematical formulation can therefore be expressed as follows:

$$\text{(MOR)} \quad \min \max_{s,s',i} \delta \quad \text{s.t.} \quad \text{constraints 2, 3 and 4 to 8} \tag{9}$$

We can slightly modify model (MOR) in order to calculate the smallest possible set of objectives that preserves the original dominance structure except for an error of δ. This is accomplished by replacing constraint (4) by Eq. (10), which imposes an upper bound on the maximum allowable error.

$$\delta(s, s', i) \leq \delta \tag{10}$$

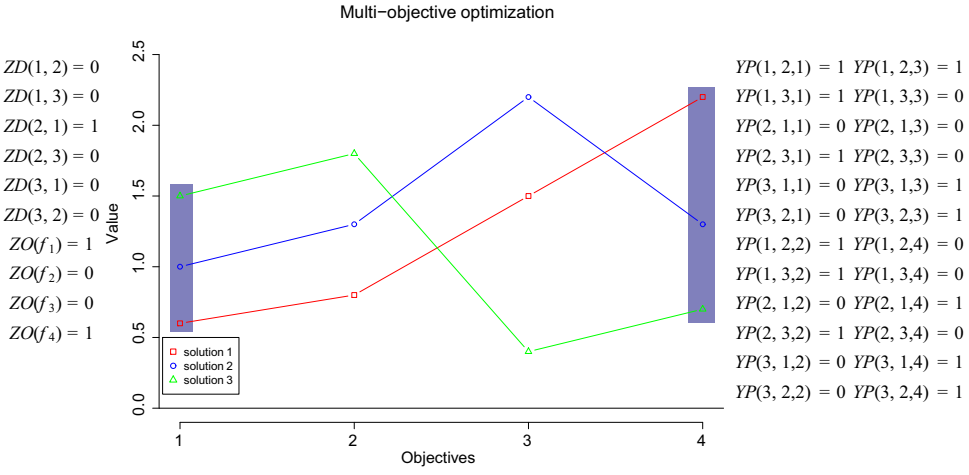

Fig. 2. Example of the notation used.

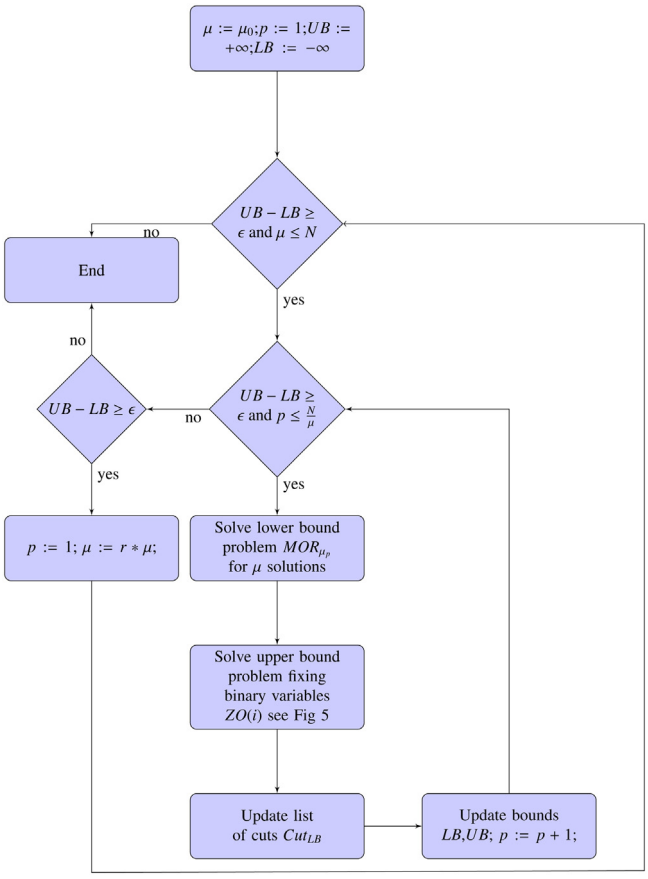

Fig. 3. Flowchart of the decomposition algorithm.

The modified model (MOR2) can then be expressed as follows:

$$\text{(MOR2)} \quad \max \sum_i (ZO(i)) \quad \text{s.t.} \quad \text{constraints 2, 3, 5 to 8, and 10} \tag{11}$$

The complexity of these MILPs grows rapidly with the number of solutions and objectives. We present in the section that follows a decomposition strategy to expedite their solution.

4. Proposed methodology

The pivotal idea of our method is to decompose the original full space MILP for dimensionality reduction into two sub-problems: a lower bounding and upper bounding sub-problem, between which our algorithm iterates until a termination criterion is satisfied. Furthermore, the solution of these sub-problems, which can be both solved faster than the full-space MILP, are used to construct a set of cutting planes that are used in order to tighten the relaxation of the upper bounding sub-problem, thereby expediting its solution. A brief outline of the algorithm is given below.

4.1. Solution strategy

The proposed decomposition algorithm (see Figs. 3–5) solves in each iteration a lower bounding problem (MOR_{μ_p}) and an upper bounding problem until the difference between the lower and upper bounds falls within an epsilon tolerance ε.

The lower bounding problem is very similar to the full-space MILP, but as oppose to it, it is defined only for a subset of the Pareto solutions. To this end, we divide the Pareto set of N solutions into subsets containing μ solutions each, with the property that the intersection of these sub-sets contains the original set. Because of the way in which they are constructed, the lower bounding MILPs contain fewer binary variables than the original MILP, and for this reason their combinatorial complexity and computational burden is lower. Furthermore, in each lower level problem we add a set of cutting planes obtained from previous sub-problems of size μ', where $\mu' < \mu$. The lower level problem (MOR_{μ_p}) serves two major purposes. First, it yields a lower bound LB on the global optimum of the full space MILP. Second, it identifies sets of objectives that are likely to yield a good approximation error after being removed from the original MILP. This information is used in the upper bounding problem to expedite its solution.

The upper bounding problem consists of a customized algorithm that calculates the delta error for a given set of objectives removed. This set of objectives is given by the lower bounding problem. The upper bounding problem provides therefore an upper bound UB on the global optimum of the original MILP. Note that calculating the delta error with the customized algorithm is faster than solving the full space MILP. We describe next each of the levels of the algorithm in more detail.

The algorithmic steps of the inner loop are summarized in Section 4.4. The notation used here is as follows. Variable p denotes the lower bounding problem solved, while μ represents the size of this sub-problem, which is denoted by variable p (hence, MOR_{μ_p} is the problem of size μ solved in iteration p). At the end of each algorithmic iteration, the value of variable μ is kept, while p is increased by one until the value N/μ is attained, thereby finishing the inner loop. In the outer loop, the value of variable μ is multiplied by r, where $r^*\mu$ represents the size of the lower bounding problems in the next iterations. When μ is equal to N, then the outer loop finishes and the algorithm stops giving a solution. At the end of each iteration p, the list of cuts $Cuts_{LB}$ are updated. The list keeps the best cuts derived from the lower bounding problems. These cuts are sorted according to the delta values attained by the corresponding lower bounding problems. Since the space is limited, we replace the worst cuts be new ones as iterations proceeds.

4.2. Lower level

As previously mentioned, the lower bounding problem corresponds to the original MILP that is defined only for a subset of solutions of size μ. To expedite the solution of these sub-problems, we add cutting planes that are obtained from previous sub-problems already solved. These cutting planes take the following form:

$$a_j \geq b_j \qquad \forall j \in CUT_{\mu_p} \tag{12}$$

where CUT_{μ_p} is the set of cuts used in iteration p to solve problems of size μ, and a_j is an auxiliary continuous variable defined as follows:

$$\delta'(s,s') \leq \delta_{\mu_p}, \qquad \forall s, s' \in S_{\mu_p} \tag{13}$$

$$\delta'(s,s') \leq a_j \qquad \forall s, s' \in S_{\mu_p} \qquad \forall j \in CUT_{\mu_p} \tag{14}$$

$$\delta(s,s',i) \leq \delta'(s,s') \qquad \forall s, s' \in S_{\mu_p} \qquad \forall i \tag{15}$$

The solution of each sub-problem MOR_{μ_p} solved at iteration p provides a lower bound on the global optimum of the full space MILP. This is because these sub-problems are defined over a reduced number of solutions (i.e., they are relaxations of the full space MILP, and are therefore guaranteed to produce a rigorous lower bound). This property is quite appealing, as it allows approximating the solution of the full space MILP without having to solve

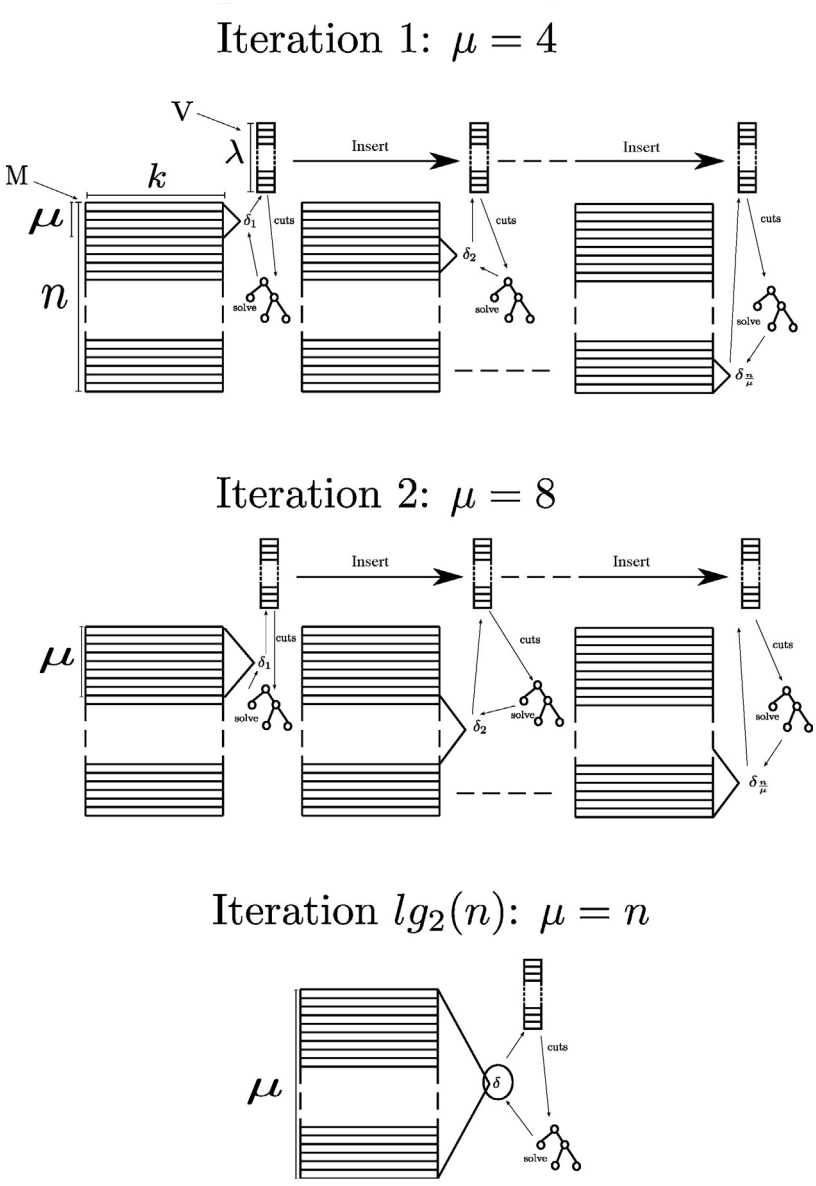

Fig. 4. Illustrative example of how the algorithm works. In each iteration we solve a problem of size μ. The solution of this problem is located in the first position in the list $Cuts_{LB}$. The $Cuts_{LB}$ list is sorted in a descendent order of delta values. Finally, the cuts are generated and added to the model.

it explicitly for problems involving a very large number of solutions and objectives (see proof in Appendix A).

4.3. Upper bounding

The upper bounding level determines the delta error for the combination of objectives predicted by the lower bounding MILP. In this level we apply a customized algorithm that calculates the delta error for a given combination of objectives. This algorithm is described in detail in Fig. 5.

4.4. Algorithmic steps

The detailed steps of the proposed decomposition strategy as follows:

1 Set problem count $p := 1$, upper bound $UB := +\infty$, lower bound $LB := -\infty$, problem size $\mu := \mu_0$, and tolerance to zero ($\epsilon := 0$).

2 Outer loop: while gap condition is satisfied ($UB - LB \geq \epsilon$) and problem size is not exceeded ($\mu \leq N$), otherwise algorithm stops.

(a) Inner loop: while gap condition is satisfied ($UB - LB \geq \epsilon$) and problem counter is not exceeded ($p \leq N/\mu$), otherwise if gap condition is satisfied ($UB - LB \geq \epsilon$) set $p := 1$ and $\mu := r^* \mu$ and Go to step 2.

(b) Solve lower bounding problem MOR_{μ_p} pointed out by p with size μ using cutting planes in list Cut_{LB}. A lower bound on the delta value (δ_{LB}) and a set of objectives to be removed are obtained ($ZO(i)$).

(c) Solve upper bounding problem fixing binary variables ($ZO(i)$) and obtain an upper bound on the delta value(δ_{UB}).

(d) List of cuts Cut_{LB} is updated: A new cut δ_{LB} is then inserted in the corresponding position in the list of cuts Cut_{LB}, replacing a previous cut when the list is filled. The list Cut_{LB} is sorted in descending order of delta value for all of the lower bounding problems solved in previous iterations.

Require: $PS : Matrix[N, K]$ of $\mathbb{R}$, $ZO : Array[i]$ of $Boolean$

Ensure: : δ is the maximum delta error of the Pareto set of solutions PS

1: **function** CUSTOMIZEDDELTAERROR($PS : Matrix[N, K]$ of $\mathbb{R}, ZO : Array[i]$ of $Boolean$) **return** $\delta : \mathbb{R}$

2: $\delta, \delta', \delta'' : \mathbb{R}$

3: $i, j, k : \mathbb{N}$

4: $dominance : Matrix[N, N] of Boolean$;

5: $dominance := FALSE$;

6: $\delta := \delta' := \delta'' := 0$;

7: **for** $i := 1; i \leq N; i := i + 1$ **do**

8: **for** $j := i + 1; j \leq N; j := j + 1$ **do**

9: $dominance[i, j] := \bigwedge_{ZO[k]} (PS[i, k] \leq PS[j, k])$ and $\overline{\bigwedge_{ZO[k]} (PS[i, k] \geq PS[j, k])}$;

10: $dominance[j, i] := \bigwedge_{ZO[k]} (PS[j, k] \leq PS[i, k])$ and $\overline{\bigwedge_{ZO[k]} (PS[j, k] \geq PS[i, k])}$;

11: **end for**

12: **end for**

13: **for** $i := 1; i \leq N; i := i + 1$ **do**

14: **for** $j := i + 1; j \leq N; j := j + 1$ **do**

15: **for** $k := 1; k \leq K; k := k + 1$ **do**

16: $\delta' := (PS[j, k] - PS[i, k]) * 1_{dominance[i,j]} * 1_{ZO[k]}$;

17: $\delta'' := (PS[i, k] - PS[j, k]) * 1_{dominance[j,i]} * 1_{ZO[k]}$;

18: **if** $\delta \leq \delta'$ **then**

19: $\delta := \delta'$;

20: **end if**

21: **if** $\delta \leq \delta''$ **then**

22: $\delta := \delta''$;

23: **end if**

24: **end for**

25: **end for**

26: **end for**

27: **end function**

Fig. 5. Algorithm to solve the upper level problem, where lines 7–12 determine the dominance relationship ($dominance[i, j]$) according to fixed variables ZO. Lines 13 to 26 compute the delta error δ.

(e) The LB and UB are updated: if $LB < \delta_{LB}$ then $LB := \delta_{LB}$ and if $UB > \delta_{UB}$ then $UB := \delta_{UB}$.
(f) Increase problem counter p by one ($p := p + 1$) and Go to step 2.

Remarks

- As we discussed above, our approach is based on the MILP introduced in a previous work (Guillén-Gosálbez, 2011b), but incorporates some differentiating issues. On the one hand, we decompose the original MILP via a bi-level algorithm, where the lower level solves sub-problems using the original MILP (that are smaller in size than the full-space MILP), and the upper level, which employs the results obtained by the lower level, is computed by a customized algorithm (see Fig. 5). On the other hand, our method differs from the one presented in Guillén-Gosálbez (2011b) in the inclusion of cuts, which are implemented through Eqs. (12)–(16).
- The lower bounding MILPs can be either solved to global optimality or stopped when an optimality gap is reached. When the second option is selected (which expedites the solution of the sub MILPs), the cutting planes are constructed considering the best possible bound obtained by the branch and cut algorithm (instead of the delta value of the best integer solution).
- Note that we have slightly modified the MILP introduced by Guillén-Gosálbez (2011a) in order to calculate the approximation error in the same manner as proposed by Brockhoff and Zitzler (2006a). Hence, to determine the delta error, we consider any pair of solutions such that one dominates the other in the reduced

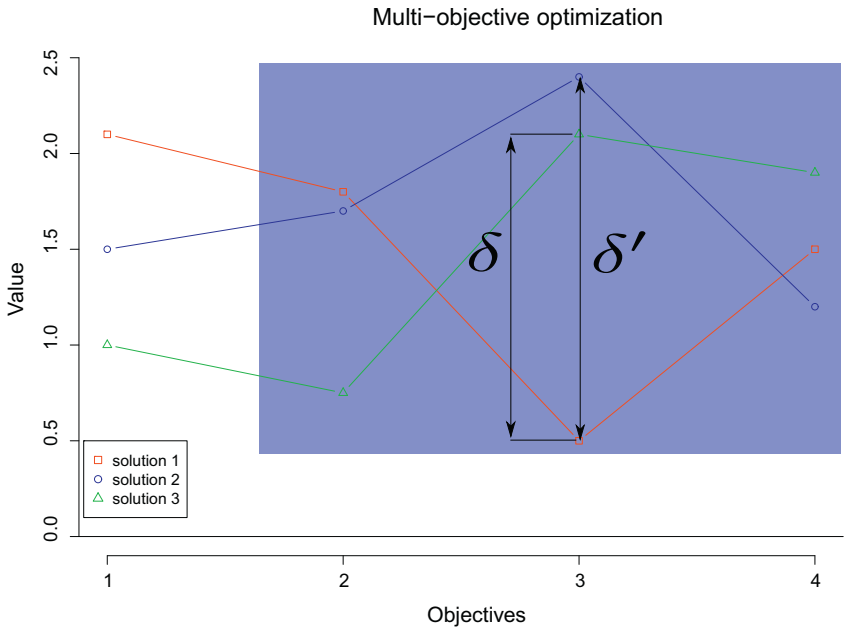

Fig. 6. Illustrative example on how to define the delta error (according to either Brockhoff and Zitzler (Brockhoff and Zitzler, 2006a), or to Guillén-Gosálbez (Guillén-Gosálbez, 2011a)). In this example, we remove objectives two, three and four. According to Brockhoff and Zitzler (Brockhoff and Zitzler, 2006a), the error would be δ', as we consider the error between any two solutions such that one dominates the other in the reduced space. However, according to Guillén-Gosálbez (Guillén-Gosálbez, 2011a), the error would be δ.

Table 1
Comparison for the hydrogen SC: MILP FS (full-space MILP), Cut-MOSS (our algorithm), exhaustive method and greedy algorithm (Brockhoff and Zitzler, 2006a) for 16 objectives removing 1–15. The CPU time associated with the Cut-MOSS includes the time spent in generating the cuts. The time displayed for every algorithm considers a 0% optimality gap, except for the greedy method which offers no guarantee of global optimality. The greedy gap is obtained from the solution provided by the greedy algorithm and the global optimal solution determined by the exact methods.

Obj	MILP FS	Cut-MOSS		Exhaustive method	Greedy	
Removed	CPU times(s)	Iterations	CPU times(s)	CPU times(s)	CPU times(s)	Optimality gap (%)
1	8.33×10^3	1	7.22	1.78×10^3	1.00×10^{-5}	0
2	8.23×10^3	1	7.31	1.74×10^3	1.34×10^{-5}	0
3	8.42×10^3	1	7.44	1.86×10^3	2.45×10^{-5}	0
4	8.12×10^3	1	7.46	1.86×10^3	3.65×10^{-5}	0
5	8.20×10^3	1	7.36	1.86×10^3	6.70×10^{-5}	0
6	8.20×10^3	1	7.39	1.87×10^3	8.23×10^{-5}	0
7	8.22×10^3	1	7.40	1.86×10^3	2.13×10^{-5}	0
8	8.33×10^3	1	7.61	1.86×10^3	9.83×10^{-4}	0
9	8.13×10^3	1	7.52	2.17×10^3	2.31×10^{-4}	0
10	8.92×10^3	1	7.58	1.96×10^3	3.10×10^{-4}	0
11	8.12×10^3	1	7.75	2.13×10^3	8.11×10^{-4}	0
12	8.34×10^3	8	5.62×10	1.20×10^3	1.32×10^{-4}	0
13	8.12×10^3	6	1.25×10^2	4.46×10^2	2.40×10^{-4}	0
14	8.22×10^3	8	1.13×10^2	9.50×10	5.65×10^{-4}	0
15	8.43×10^3	8	6.12×10	3.05×10	1.11×10^{-4}	0

space regardless of whether both solutions are Pareto optimal in such a reduced domain. Fig. 6 provides an illustrative example on this issue.

- The Pareto solutions must be normalized before the application of the algorithm. There are different alternatives to perform this step. The method followed in this work is described in detail in Appendix B.

5. Computational results

We illustrate the capabilities of our approach using two case studies, which have been solved following the steps summarized in Fig. 7. The first addresses the design of hydrogen supply chains (a problem in the area of green engineering), while the second deals with the multi-objective optimization of metabolic networks (taken from the field of systems biology). In both cases, our method was compared against the full space MILP model introduced by Guillén-Gosálbez (2011a), and the exact and greedy methods developed by Brockhoff and Zitzler (2006a). All the numerical experiments were conducted on a computer Intel(R) Core (TM) i7-3612QM CPU@ 2.10GHz 2.10GHz and 6GB of memory RAM. We describe next in detail the numerical results obtained in each case.

5.1. Design of hydrogen supply chains for vehicle use

This example deals with the optimal design of a hydrogen SC for vehicle use in Spain taking into account economic and environmental concerns. The problem, which was first proposed by Almansoori and Shah (2009), considers different technologies for production, storage and transportation of hydrogen to be established in a set

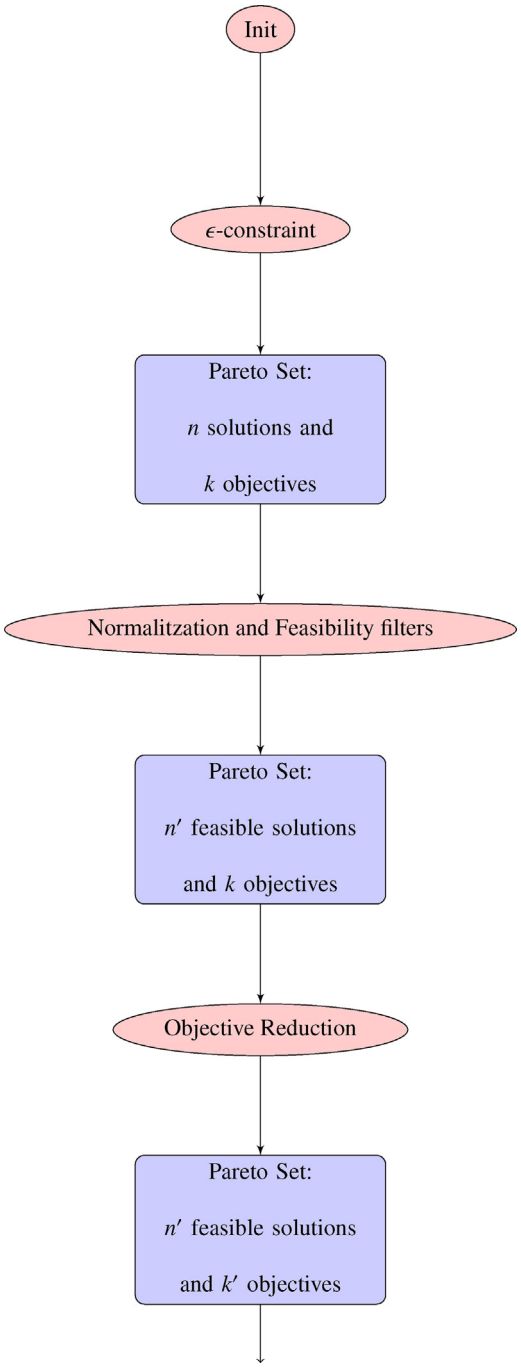

Fig. 7. Proposed framework to Pareto set generation.

of geographical regions distributed all over the country (see Fig. 8). The goal is to determine the optimal network configuration in terms of its economic and environmental performance. The problem can be formulated as a multi-objective MILP that seeks to minimize the total cost of the network and its environmental impact. In this formulation, integer variables indicate the number of plants and storage facilities to be opened in a specific region (i.e., grid), whereas binary variables are employed to denote the existence of transportation links connecting the SC entities. The environmental impact was calculated through 16 life cycle assessment indicators based on the Eco-indicator 99 methodology. Further details on this case study can be found in Sabio et al. (2010).

Designing efficient hydrogen supply chains requires the simultaneous assessment of several alternatives and identification of the best combination of technologies. These technologies might differ in capital investments, required feedstocks, production cost (Balat and Kirtay, 2010) and environmental performance (Koroneos et al., 2004; Spath and Mann, 2001, 2001). In this context, optimizing exclusively the economical performance may lead to solutions that do not fully exploit the environmental benefits of moving toward a hydrogen-based energy system (De-León Almaraz et al., 2013). It is therefore clear that environmental concerns must be accounted for along with economic criteria in the optimization of hydrogen supply chains. Several works have been presented and we review briefly these. Hugo et al. (2005) developed a model that identifies the optimal infrastructure taking into account different hydrogen pathways in Germany and considering investment and environmental criteria. This model has been extended and considered as a basis for other works (Lin et al., 2008; Ingason et al., 2008; Qadrdan et al., 2008; Guillén-Gosálbez et al., 2010). Murthy Konda et al. (2011) presented a multi-period optimization framework for the design of spatially explicit and time-evolutionary hydrogen supply networks. Finally, Sabio et al. (2012) addressed the strategic planning of hydrogen supply chains by minimizing the total cost along with a set of life cycle assessment (LCA) impacts. The authors performed as well a post-optimal analysis of the results using principal component analysis (PCA) in order to facilitate the interpretation and selection of final alternatives.

We first generated 300 Pareto solutions of the supply chain design problem using the ϵ-constraint method (Haimes et al., 1971), which solves a set of single-objective problems in which one objective is kept as main objective while the others are transferred to auxiliary constraints. These solutions were normalized (see Appendix B) and then used for dimensionality reduction. Particularly, we solved different instances of the dimensionality reduction problem, in each of which the goal was to eliminate a given number of objectives ranging between 1 and 15. The MILP for dimensionality reduction contains 4,485,034 variables and 7,445,102 equations.

Table 1 shows the numerical results obtained with each of the algorithms (i.e., the full space MILP, the proposed approach, and the two algorithms of Brockhoff and Zitzler, 2006a). Particularly, the table provides the CPU time, the number of iterations (only for the case of our approach) and the optimality gap (only for the case of the greedy algorithm). The MILP and our approach are solved to global optimality (i.e., with an optimality gap of 0%; note that the exhaustive method also provides a solution with a zero gap). On the other hand, the greedy algorithm provides no information regarding the gap of the final solution found. However, for this later algorithm we determine the optimality gap through comparison with the optimal solution calculated with the other exact approaches.

As observed, our method identifies the optimal solution faster than the MILP (Guillén-Gosálbez, 2011a) and than the exhaustive method proposed by Brockhoff and Zitzler (Brockhoff and Zitzler, 2006a) in almost all of the cases solved. Particularly, we get reductions of CPU time in between 1 and 3 orders of magnitude when compared to the other methods that also guarantee global optimality (i.e., the full space MILP and the exhaustive method). On the other hand, the greedy algorithm is always the fastest approach. Note that despite yielding the global optimum in all of the cases, there is indeed no theoretical guarantee of reaching the global optimum when using the greedy algorithm.

The high performance of the greedy algorithm is due largely to its strategy, which starts with a reduced set of one objective, and then adds to this set the objective that leads to the minimum error, and keeps on doing this until the desired number of objectives kept is reached. This strategy performs very well in practice, but offers no guarantee of convergence to the global optimum. Note also that the exhaustive method is the best algorithm only for those cases with a small number of objectives kept (i.e., when removing more than 12). Recall that the exhaustive method calculates the error

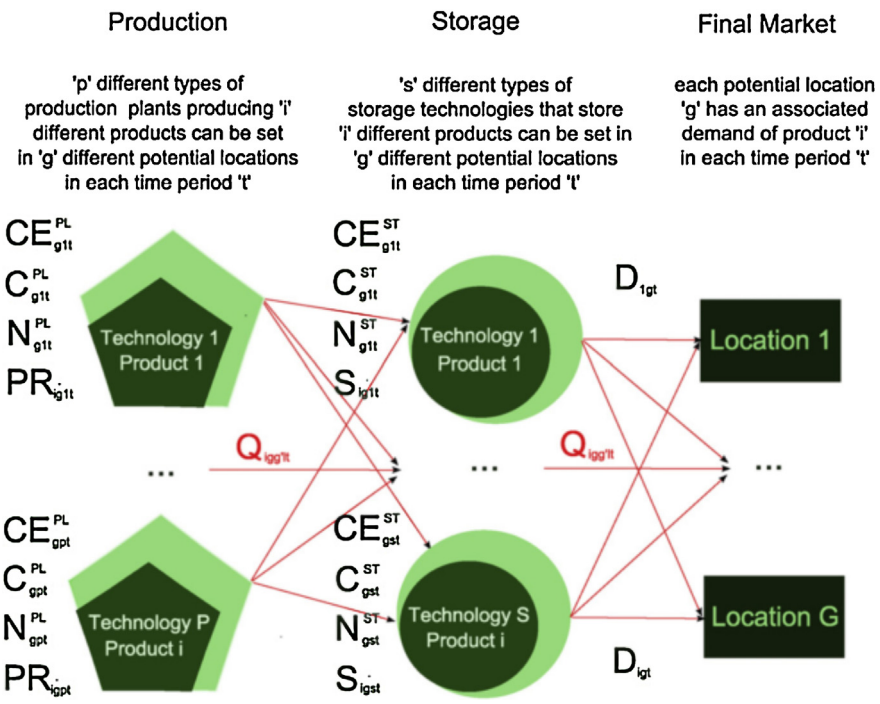

Fig. 8. Superstructure for the hydrogen supply chain design problem, which is formulated as an MILP. We derive an MILP for the optimal design of the network that optimizes simultaneously the total cost and environmental performance (quantified in terms of 16 environmental impacts). The MILP provides the optimal location of production and storage facilities, the technologies to be implemented and the transportation links between the SC entities.

for every possible combination of objectives kept. Hence, when the number of objectives kept is small, and so is the number of potential combinations, the algorithm performs well, since it can determine the delta error of each such combination quite fast. On the contrary, for a large number of potential combinations, we expect the exhaustive algorithm to perform worst than our method.

5.2. Multi-objective optimization of metabolic networks

This example deals with metabolic optimization problems arising in systems biology studies (Pozo and Guillén, 2012). Given a metabolic network (see Fig. 9) described by a GMA model, the goal is to determine the set of preferred enzymatic profiles that optimize the synthesis rate of a metabolite at minimum cost (minimum number of changes in the enzyme activities, i.e., minimum change in gene expression) and minimum increase in the concentration of intermediate metabolites during the fermentation of Saccharomyces cerevisiae for ethanol production. We consider 15 objectives: Objectives 1–8 correspond to changes in the enzyme expressions that should be minimized so as to make it easier to manipulate genetically the strain of interest and maintain cell's homeostasis (enzymes $K1$–$K8$). K1: Hexose transporters, $K2$: Glucokinase/Hexokinase, $K3$: Phosphofructokinase, $K4$: Trehalose 6-phosphate syntase complex (+Glycogen production), $K5$: Glyceraldehyde-3-phosphate dehydrogenase, $K6$: GOL (Glycerol production), $K7$: Pyruvate kynase, $K8$: ATPase; objective 9 is the total cost (the cost of changing the enzyme activities, that should be also minimized); objective 10 is the synthesis rate of ethanol (that should be maximized); and the remaining 5 objectives represent the concentration of metabolites $X1$–$X5$ that should all be minimized and kept as close as possible to those in the basal state to ensure as well cell's homeostasis. $X1$: Internal glucose, $X2$: Glucose-6- phosphate, $X3$: Fructose-1,6-diphosphate, $X4$: Phosphoenolpyruvate, $X5$: Adenosine triphosphate (Pozo and Guillén, 2012).

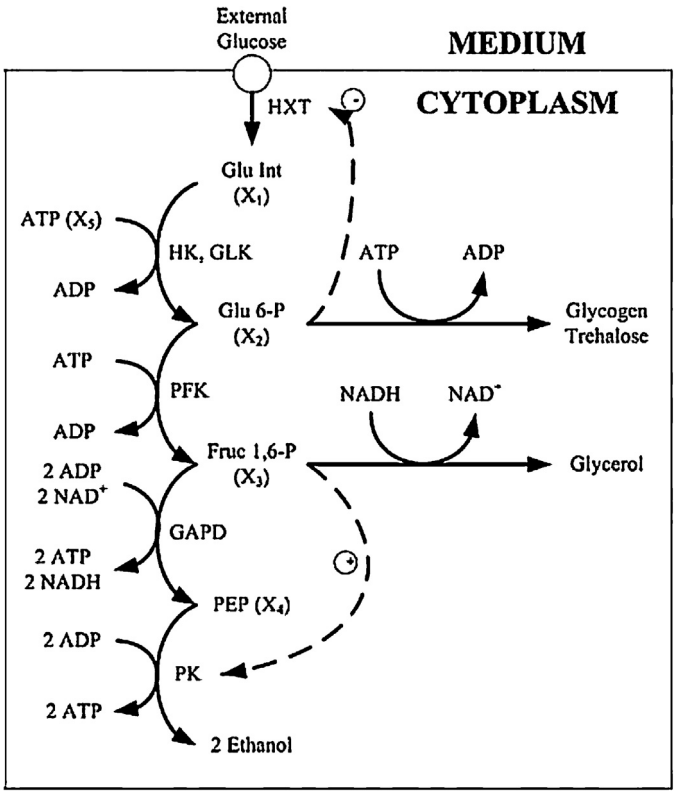

Fig. 9. Metabolic pathway of the fermentation of *Saccharomyces cerevisiae*. The identification of the optimal enzymatic profiles in terms of several biological criteria is posed as an mixed-integer nonlinear programming (MINLP) problem based on a generic kinetic representation of the network. The objectives considered include the minimization of the enzymatic manipulations, the maximization of the ethanol synthesis rate and the minimization of several intermediate metabolites.

Table 2
Comparison for the metabolic network problem: MILP FS (full space MILP), Cut-MOSS (our algorithm), exhaustive method and greedy algorithms (Brockhoff and Zitzler, 2006a) for 14 objectives removing 1–14. The CPU time associated with the Cut-MOSS includes the time spent in generating the cuts. The time displayed for every algorithm considers a 0% optimality gap, except for the greedy method which offers no guarantee of global optimality. The greedy gap is obtained from the solution provided by the greedy algorithm and the global optimal solution determined by the exact methods.

Obj	MILP FS	Cut-MOSS		Exhaustive method	Greedy	
Removed	CPU times(s)	Iterations	CPU times(s)	CPU times(s)	CPU times(s)	Optimality gap (%)
1	8.33×10^3	1	7.55	1.58×10^3	4.84×10^{-5}	0
2	8.19×10^3	1	7.32	1.58×10^3	2.92×10^{-5}	0.91
3	8.18×10^3	1	7.44	1.57×10^3	8.84×10^{-5}	0.91
4	8.49×10^3	1	7.46	1.59×10^3	9.44×10^{-5}	0.91
5	8.51×10^3	1	7.36	1.50×10^3	2.66×10^{-5}	0.91
6	8.20×10^3	1	7.40	1.50×10^3	1.90×10^{-5}	0.91
7	7.62×10^3	1	7.40	1.50×10^3	8.87×10^{-5}	0.91
8	8.33×10^3	11	7.61	1.50×10^3	4.62×10^{-5}	0.91
9	7.65×10^3	1	7.52	1.58×10^3	5.49×10^{-5}	0.91
10	7.37×10^3	1	7.58	1.48×10^3	9.28×10^{-5}	0.91
11	7.32×10^3	9	7.75	1.17×10^3	1.60×10^{-4}	0.91
12	No solution	13	4.69×10^2	5.30×10^2	9.81×10^{-4}	0.91
13	No solution	12	1.28×10^2	2.00×10^2	5.86×10^{-4}	0.90
14	No solution	5	1.13×10^2	4.56	2.24×10^{-4}	0.74

The importance of multi-objective optimization in metabolic studies has been pointed out by several authors like Vera et al. (2003), Liu and Wang (2008), and Wu et al. (2011). Pozo and Guillén (2012) combined multi-objective global optimization and Pareto filters as a manner to identify optimal genetic manipulations in metabolic models. de Hijas-Liste et al. (2014) apply multi-objective optimization to metabolic pathways. Wang and Wu (2013) introduce a generalized fuzzy multi-objective optimization approach for finding optimal enzyme effects on metabolic network systems. Higuera et al. (2012) apply a multi-objective optimization approach to the allosteric regulation of enzymes using a model of a metabolic substrate-cycle.

For this case, we proceeded similarly as in the previous example. We first generated 300 solutions using the ϵ-constraint method (Haimes et al., 1971). The solutions were next normalized (see Appendix B), and we then solved the dimensionality reduction problem, where the goal was to minimize the delta error eliminating a number of objectives ranging between 1 and 14. This model contains 1,421,528 variables and 3,529,528 equations.

Table 2 shows the numerical results obtained with each of the algorithms (i.e., the full space MILP, the proposed approach and the two algorithms of Brockhoff and Zitzler, 2006a). The MILP and our approach are both solved with an optimality gap of 0%. As observed in this case, our method identifies the optimal solution much faster than the MILP (Guillén-Gosálbez, 2011a) and the exhaustive method (Brockhoff and Zitzler, 2006a). Particularly, we get reductions of CPU time of approximately 3 orders of magnitude when compared to the other exhaustive methods that guarantee global optimality. In this case, the greedy algorithm is again always the fastest approach, but recall that it offers no theoretical guarantee of global optimality.

Note that the greedy algorithm also shows an exceptional performance in this case, but as already mentioned before, we should emphasize that: (i) it offers no guarantee of convergence to the global optimum; (ii) it provides in this case solutions that are suboptimal. The exhaustive method is the best algorithm only for the case with 1 objective kept (14 objectives removed). Note that the MILP method fails to provide a solution for those cases with a small number of objectives kept (i.e, more than 11 objectives removed).

6. Conclusions

This article has proposed a novel approach for reducing the number of objectives in MOO that is based on an MILP formulation for dimensionality reduction introduced in a previous work by the authors. The method presented relies on decomposing the aforementioned MILP formulation into a set of sub MILPs whose solution is used to construct cutting planes for the original full space MILP. Numerical results show that the method proposed works efficiently, outperforming in complex problems with a large number of objectives and/or solutions the stand alone MILP and the exhaustive and greedy algorithms for dimensionality reduction introduced by Brockhoff and Zitzler (2006a). Our tool aims to ameliorate the numerical difficulties arising in the solution of *MOO* problems with a large number of objectives. Future work will focus on integrating this tool within MOO algorithms in order to enhance their numerical performance.

Acknowledgements

Pedro J. Copado wishes to acknowledge support of this research work from the Spanish Ministry of Education and Science (DPI2008-04099/DPI). The authors wish also to acknowledge support from the Spanish Ministry of Education and Science (Projects CTQ2012-37039-C02, DPI2012-37154-C02-02 and ENE2011-28269-C03-03).

Appendix A. Proof of Theorem A.1

Theorem A.1 shows that the solution of each sub-problem MOR_{μ_p} solved at iteration p is a relaxation of the full space MILP, and it in turn provides a rigorous lower bound on the global optimum of the full space MILP. We demonstrate this property below:

Theorem A.1. *Let δ_{μ_p} be the optimal solution of sub-problem MOR_{μ_p} solved at iteration p and size μ of the algorithm (which is defined for the subset of solutions S_{μ_p} contained in S), and b_j be a parameter used in cut j that corresponds to the optimal solution of an instance of problem MOR that is used to construct cut j (i.e., instance MOR_j defined for solutions S_j). The inequalities from (12)–(15) are a valid cut for problem MOR_{μ_p}.*

Proof. The proof is by contradiction. We claim that inequalities (12) and (15) do not chop off any feasible solution of problem *MOR*. Assume that there is a feasible solution of *MOR* such that for this solution (i.e., combination of objectives omitted) the value of the auxiliary variable a_j (denoted by $\bar{a}_j$) is strictly lower than b_j. From Eqs. (13)–(15), it follows that for such combination of objectives omitted, the value of the error (denoted by $\overline{\delta(s, s', i)}$) in every

objective and pair of solutions in the set S_j will be strictly lower than b_j, that is:

$$\overline{\delta(s, s', i)} < b_j \qquad \forall s, s' \in S_j, \qquad \forall j \in CUT_{\mu_p}, \qquad \forall i \tag{16}$$

This contradicts the fact that b_j is the global optimum of problem MOR_j ■.

□

Appendix B. Normalization of the Pareto optimal solutions

A normalization step is applied to the Pareto set of solutions in order to make them comparable in all of the objectives. Several methods are available for this purpose (Cloquell and Santamarina, 2001). We have used in our case the following expression:

$$v'_i = \frac{v_i - v_{MIN}}{v_{MAX} - v_{MIN}}$$

where v'_i is the normalized value, and v_i is the original value. This method covers exactly the range [0, 1], where zero is the best value and one the worst value (we consider that we aim to minimize all of the objectives simultaneously).

References

Almansoori A, Shah N. Design and operation of a future hydrogen supply chain: Multi-period model. Int J Hydrogen Energy 2009];34(19):7883–97.

Alonso MAHA-OC. A multiobjective approach for reactive power planning in networks with wind power generation. Renew Energy 2012];37(1):180–91.

Balat M, Kirtay E. Major technical barriers to a "hydrogen economy". Energy Source Part A 2010];32(9):863–76.

Brockhoff D, Zitzler E. Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In: Parallel Problem Solving from Nature-PPSN IX; 2006a]. p. 533–42.

Brockhoff D, Zitzler E. On Objective Conflicts and Objective Reduction in Multiple Criteria Optimization. In: TIK Report 243; 2006].

Claro JPDSJ. A multiobjective metaheuristic for a mean-risk multistage capacity investment problem with process flexibility. Comput Operat Res 2012];39(4):838–49.

Cloquell V, Santamarina MHA. In: XVII Congreso Nacional de Ingeniera de Proyectos. Murcia. Nuevo procedimiento para la normalizacin de valores numricos en la toma de decisiones 2001].

de Hijas-Liste GM, Klipp E, Balsa-Canto E, Banga JR. Global dynamic optimization approach to predict activation in metabolic pathways. BMC Syst Biol 2014];8:1.

De-León Almaraz S, Azzaro-Pantel C, Montastruc L, Pibouleau L, Senties OB. Assessment of mono and multi-objective optimization to design a hydrogen supply chain. Int J Hydrogen Energy 2013];38(33):14121–45.

Deb K, Saxena DK. On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: KanGal Report Number 2005011; 2005].

Guillén-Gosálbez G. A novel MILP-based objective reduction method for multi-objective optimization: application to environmental problems. Comput Chem Eng 2011a].

Guillén-Gosálbez G. A novel MILP-based objective reduction method for multi-objective optimization: application to environmental problems. Comput Chem Eng 2011b];35(8):1469–77.

Guillén-Gosálbez G, Mele FD, Grossmann IE. A bi-criterion optimization approach for the design and planning of hydrogen supply chains for vehicle use. AIChE J 2010];56(3):650–67.

Haimes YY, Lasdon LS, Wismer D. Integrated Optimization. IEE Trans System Man Cybernet 1971];47(JULy):296–7.

Higuera C, Villaverde AF, Banga JR, Ross J, Morán F. Multi-criteria optimization of regulation in metabolic networks. PLoS ONE 2012];7(7):e41122.

Hugo A, Pistikopoulos E. Environmentally conscious long-range planning and design of supply chain networks. J Clean Prod 2005];13(15):1471–91.

Ingason H, Pallingolfsson H, Jensson P. Optimizing site selection for hydrogen production in Iceland. Int J Hydrogen Energy 2008];33(14):3632–43.

Koroneos C, Dompros A, Roumbas G, Moussiopoulos N. Life cycle assessment of hydrogen fuel production processes. Int J Hydrogen Energy 2004];29(14):1443–50.

Lin Z, Ogden J, Fan Y, Chen CW. The fuel-travel-back approach to hydrogen station siting. Int J Hydrogen Energy 2008];33(12):3096–101.

Liu P-K, Wang F-S. Inference of biochemical network models in S-system using multiobjective optimization approach. Bioinformatics (Oxford, England) 2008];24(8):1085–92.

López Jaimes A, Coello C, Urías Barrientos J. Online objective reduction to deal with many-objective problems. In: Evolutionary multi-criterion optimization; 2009]. p. 423–37.

Murthy Konda N, Shah N, Brandon NP. Optimal transition towards a large-scale hydrogen infrastructure for the transport sector: the case for the Netherlands. Int J Hydrogen Energy 2011];36(8):4619–35.

Niknam T, Fard AKSA. Distribution feeder reconfiguration considering fuel cell/wind/photovoltaic power plants. Renew Energy 2012];37(1):213–25.

Pozo C, Guillén G. Identifying the Preferred Subset of Enzymatic Profiles in Nonlinear Kinetic Metabolic Models via Multiobjective Global Optimization and Pareto Filters. PLoS ONE 2012];7(9):1–11.

Purshouse R, Fleming P. Conflict, harmony, and independence: relationships in evolutionary multi-criterion optimisation. In: Evolutionary multi-criterion optimization; 2003]. p. 67.

Qadrdan M, Saboohi Y, Shayegan J. A model for investigation of optimal hydrogen pathway, and evaluation of environmental impacts of hydrogen supply system. Int J Hydrogen Energy 2008];33(24):7314–25.

Sabio N, Gadalla M, Guillén-Gosálbez G, Jiménez L. Strategic planning with risk control of hydrogen supply chains for vehicle use under uncertainty in operating costs: a case study of Spain. Int J Hydrogen Energy 2010];35(13):6836–52.

Sabio N, Kostin A, Guillén-Gosálbez G, Jiménez L. Holistic minimization of the life cycle environmental impact of hydrogen infrastructures using multi-objective optimization and principal component analysis. Int J Hydrogen Energy 2012];37(6):5385–405.

Sharma AYV. Modelling and optimization of cut quality during pulsed Nd:YAG laser cutting of thin Al-alloy sheet for straight profile. Opt Laser Technol 2012];44(1):159–68.

Spath M, Mann P. Life cycle assessment of renewable hydrogen production via wind/electrolysis; 2001], Technical report, Department of Energy's Hydrogen Program.

Spath P, Mann M. Life cycle assessment of hydrogen production via natural gas steam reforming; 2001], Technical report, National Renewable Energy Laboratory.

Thoai NV. Criteria and dimension reduction of linear multiple criteria optimization problems. J Glob Opt 2011]:1–10.

Ustun OKR. Combined forecasts in portfolio optimization: a generalized approach. Comput Operat Res 2012];39(4):805–19.

Vera J, de Atauri P, Cascante M, Torres NV. Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by Saccharomyces cerevisiae. Biotechnol Bioeng 2003];83(3):335–43.

Wang FS, Wu WH. Multi-objective optimization in chemical engineering. Oxford, UK: John Wiley & Sons Ltd; 2013].

Wu W-H, Wang F-S, Chang M-S. Multi-objective optimization of enzyme manipulations in metabolic networks considering resilience effects. BMC Syst Biol 2011];5(1):145.