

A comparative study of multi-objective optimization methodologies for molecular and process design

Ye Seol Lee, Edward J. Graham, Amparo Galindo, George Jackson, Claire S. Adjiman*

Department of Chemical Engineering, Centre for Process Systems Engineering, Institute for Molecular Science and Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom



ARTICLE INFO

Article history:

Received 11 December 2019

Revised 24 February 2020

Accepted 27 February 2020

Available online 29 February 2020

Keywords:

Computer-aided molecular design
Computer-aided molecular and process design

Multi-objective optimization

Mixed-integer nonlinear programming

Global optimization

ABSTRACT

The need to consider multiple objectives in molecular design, whether based on techno-economic, environmental or health and safety metrics is increasingly recognized. There is, however, limited understanding of the suitability of different multi-objective optimization (MOO) algorithm for the solution of such design problems. In this work, we present a systematic comparison of the performance of five mixed-integer non-linear programming (MINLP) MOO algorithms on the selection of computer-aided molecular design (CAMD) and computer-aided molecular and process design (CAMPD) problems. The five methods are designed to address the discrete and nonlinear nature of the problem, with the aim of generating an accurate approximation of the Pareto front. They include: a weighted sum approach without global search phases (SWS), a weighted sum approach with simulated annealing (WSSA), a weighted sum approach with multi level single linkage (WSML), the sandwich algorithm with MSL (SDML) and the non dominated sorting genetic algorithm-II (NSGA-II). The algorithms are compared systematically in two steps. The effectiveness of the global search methods is evaluated with SWS, WSSA and WSML. WSML is found to be most effective and a comparative analysis of WSML, SDML and NSGA-II is then undertaken. As a test set of these optimization techniques, two CAMD and one CAMPD problems of varying dimensionality are formulated as case studies. The results show that the SDML provides the most efficient generation of a diverse set of Pareto points, leading to the construction of an approximate Pareto front close to exact Pareto front.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

The multiple ways in which the selection of a molecule can impact performance have been well documented through the computer aided molecular design (CAMD) and computer aided molecular and process design (CAMPD) literature. This is reflected in the diversity of objective functions considered: property metrics (e.g. Pretel et al., 1994; Gani et al., 2006; Folić et al., 2008; Austin et al., 2018), economics (e.g. Pereira et al., 2011; Gopinath et al., 2016; Ahmad et al., 2018), productivity metrics (e.g. Bardow et al., 2010; Bowskill et al., 2020; Cignitti et al., 2017; Eden et al., 2004; Zhang et al., 2017), or environmental and safety criteria (e.g. Duvedi et al., 1996; Pistikopoulos and Stefanis, 1998; Hostrup et al., 1999; Song and Song, 2008; Papadopoulos et al., 2013; Khor et al., 2017). An in-depth description of these is beyond the scope of our current paper and the reader is referred to Gopinath et al. (2016) and Ng et al. (2015) and the references therein for further details.

The diversity of objectives used highlights that it is beneficial in many cases to consider multiple conflicting objectives—for example, property targets, sustainability targets, societal impact, and economic performance—that cannot easily be combined together in a single metric. Multi-objective optimization (MOO) is thus receiving increasing attention in the area of CAM(P)D. In principle, for a problem with continuous decision variables, the presence of conflicting objectives results in an infinite number of optimal solutions, commonly known as Pareto-frontier solutions. Usually, it is not possible to derive an analytical description of the Pareto frontier (Deb, 2001). Hence, in practical applications, the Pareto frontier is approximated by a finite number of Pareto-optimal solutions (Marler and Arora, 2004).

In a nutshell, the main idea in the development of MOO algorithms is to (1) find non-dominated points that can represent the Pareto-frontier in reasonable computational time; and (2) generate these points so that they are distributed evenly along the Pareto front. The main MOO methods that have been used in molecular design include scalarization methods (e.g. weighted sum (WS) method, sandwich (SD) algorithm), ϵ -constraint methods, and

* Corresponding author.

E-mail address: c.adjiman@imperial.ac.uk (C.S. Adjiman).

metaheuristic methods. Papadopoulos and Linke (2006) proposed a multi-objective molecular design technique linked with a process synthesis framework by using the weighted sum method, extending it to the design of binary working fluid mixtures in Organic Rankine Cycles (ORC) (Papadopoulos et al., 2010) and to the design of solvents for CO₂ capture (Papadopoulos et al., 2016, 2020). The authors adopted the simulated annealing (SA) algorithm proposed by Marcoulaki and Kokossis (2000) to explore the design space directly. Burger et al. (2015) utilized the SD (Bortz et al., 2014) within their mixed-integer nonlinear programming (MINLP) solution strategy to design a solvent for a CO₂ physical absorption process, avoiding the difficulty in assigning weight vectors. The solutions generated by MOO were used as starting points for the solution of the CAMPD MINLP. In this last step, a single (economic) objective was used. Zhou et al. (2019) also applied the SD to identify a list of solvent candidates in their MOO CAMD formulation, which were further optimized using rigorous thermodynamic analysis. In their CAMD problem, selectivity and capacity of the solvents were optimized simultaneously to consider their efficiency within extractive distillation process. Buxton et al. (1999) and Hugo et al. (2004) considered both process economics and environmental impact simultaneously in the formulation of a CAMPD MINLP and proposed the use of the ϵ -constraint method (Haimes et al., 1971) for its solution. The environmental impact metrics were treated as constraints in their formulation, while the economic performance was set as the objective function. Kim and Diwekar (2002) proposed a novel MOO framework based on the ϵ -constraint method to solve the integrated design of the solvent recycling process and environmentally benign solvents for acetic acid removal from water. In their study, a four-objective problem was transformed to single-objective optimization (SOO) problems and the solutions of the problem is obtained using the Hammersley stochastic annealing algorithm by which the number of SOO problems are efficiently sampled. Schilling et al. (2017) solved an integrated working fluid and ORC process design problem with the ϵ -constraint method to identify the trade-off between net power output and total capital investment. The CAMPD problem was formulated as a MINLP and a one-stage continuous-molecular targeting (CoMT) approach was applied. Ng et al. (2014) introduced a metaheuristic method, fuzzy optimization (Liang, 2008), in developing a MOO CAMPD approach to the design of optimal chemical products, by considering both the optimality of product properties and the accuracy of the property prediction models. Venkatasubramanian et al. (1994) employed a genetic algorithm (GA) in the design of a optimal structure of the polymer and refrigerant. They introduced a string representation of the molecular structures as an encoding strategy and used the molecular genetic operators (single-point crossover, chain-mutation, insertion, deletion, and blending) to SOO MINLP problems. Dörgő and Abonyi (2016, 2019) extended the GA approaches to the design of refrigerants for ORC processes. The authors solved the MOO MINLP problems using non-dominated sorting genetic algorithm II (NSGA-II) (Deb, 2001) to investigate trade-offs between several properties of molecules. Xu and Diwekar (2007) developed a multi-objective efficient genetic algorithm (MOEGA) in the integrated design of solvents and solvent recycling process of mixtures of acetic acid and water. The authors considered up to six objectives including acetic acid recovery, process flexibility, two environmental impacts based on LC₅₀, an environmental factor based on bioconcentration factor and the energy consumption of the reboiler.

While several MOO methods have been applied to CAM(P)D, CAM(P)D presents challenges due to the nonconvexity of the search space that arise from the continuous functions and the presence of integer variables. The Pareto front can be discontinuous and nonconvex, so that the efficient identification of a well-

distributed set of points on or near the Pareto front is non-trivial. In particular, WS approaches are highly dependent on the choice of weight vectors. An even distribution of the weights among objective functions does not always leads to an even distribution of solutions on the Pareto front (Das and Dennis, 1997). Therefore, the use of the WS is often time-consuming as a large number of SOO problems need to be solved. Although this drawback has been addressed in the SD algorithm, in which the weight vectors are selected systematically, the WS and SD approaches can only identify convex regions of the Pareto front (Marler and Arora, 2004). This is in fact a limitation of all weighted sum based methods (Bortz et al., 2014). This limitation can be overcome with ϵ -constraint methods, but a challenge is to choose appropriate values of epsilon, ϵ , especially when the problem entails more than two objectives (Mavrotas, 2009). Moreover, the constraints added to the original problem increase the complexity of solving each SOO problem. Furthermore, it may be difficult to find even one point on the Pareto front and convergence to dominated solutions is common. In scalarization and ϵ -constraint methods, this is apparent when SOO problems are solved repeatedly and the mixed-integer and nonconvex nature of most molecular design problems means the solver may converge to a dominated solution rather than a Pareto point. In fuzzy methods and genetic algorithms, a set of solutions that is close to the Pareto front may be identified, but there can be no guarantee that the points found are Pareto-optimal (Deb, 2001). The application of these methods can result in a large number of unsuccessful computations due to the highly constrained nature of such MINLPs. Despite the many challenges faced in applying MOO to MINLPs, there has been no systematic analysis to compare the performance of various algorithms for CAM(P)D applications.

In this work, we present a comparative analysis of the performance of three classes of MINLP MOO approaches, the WS, the SD (Rennen et al., 2011), and the NSGA-II (Deb et al., 2000). The solution of each scalarized problem with WS and SD can be challenging in the presence of nonlinearities. To increase the likelihood of identifying the globally-optimal Pareto front, we introduce SA (Marcoulaki and Kokossis, 2000) and multi-level single linkage (MLSL) (Kucherenko and Sytsko, 2005) as global search methods. We first make use of a SA version of the WS (WSSA) and a MLSL version of the WS (WSML). These two algorithms are compared with the WS method to investigate the effectiveness of global search methods. Based on findings, we also put forward two variants of the algorithms: MLSL with the WS (WSML) and the SD (SDML), and a comparison between WSML, SDML and NSGA-II is undertaken.

The resulting set of algorithms is applied to three case studies: the design of solvents for the chemical absorption of CO₂, the design of working fluids for ORC based on property metrics, and the integrated design of working fluids and ORCs processes. The performance of the different algorithms is compared based on reliability and efficiency criteria.

The main contributions of this work include: 1) the introduction of different global search methods to solve single objective MINLPs within MOO scalarization methods; 2) a systematic comparative study of the performance of different classes of MOO algorithms on solving several literature MOO CAM(P)D problems. In addition, several modifications of the SA, MLSL and NSGA-II approaches are introduced to adapt these algorithms to CAM(P)D problems. The remainder of this article is structured as follows. In Section 2, we describe the optimization algorithms. We present in Section 3 the performance metrics used to compare the algorithms. We formulate the case studies in Section 4. Subsequently, the results of the MOO algorithms when applied to the case studies are reported in Section 5, where the performance of each algorithm is discussed in detail. Finally, we provide the main conclusions in Section 6.

2. Multi-objective optimization methodologies

2.1. Problem formulation

The molecular design problem is often posed as MINLP problem. The generic mathematical formulation of the MINLP MOO problem is:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}, \mathbf{n}} \quad & f_1^o(\mathbf{x}, \mathbf{y}, \mathbf{n}), f_2^o(\mathbf{x}, \mathbf{y}, \mathbf{n}), \dots f_p^o(\mathbf{x}, \mathbf{y}, \mathbf{n}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{n}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{n}) = \mathbf{0} \\ & \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \{0, 1\}^q, \mathbf{n} \in \mathbf{N} \subset \mathbb{Z}^q \end{aligned} \quad (1)$$

where p is the number of objectives, \mathbf{x} is a n -dimensional vector of continuous variables, \mathbf{y} is a q -dimensional vector of binary or integer variables, \mathbf{n} is a q -dimensional vector of integer variables, $\mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{n})$ is a vector of inequality constraints that define design constraints and feasibility constraints and $\mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{n})$ is a vector of equality constraints that include structure-property models, process models and chemical feasibility constraints such as the octet rule (Odele and Macchietto, 1993).

2.2. Scalarization-based methods

2.2.1. Weighted sum method

The WS method is one of the most widely used scalarization-based algorithms in MOO. Multiple objective functions f_i^o , $i=1, \dots, p$ are aggregated into a single objective function using weight coefficients, represented by the normalized vector, \mathbf{w} . Thus, the scalar objective function is given by $\sum_{i=1}^p w_i f_i^o(\mathbf{x}, \mathbf{y}, \mathbf{n})$, where $\sum_{i=1}^p w_i = 1$. The WS method is easy to implement and the problem is of same degree of difficulty as the original MOO problem since there are no additional constraints involved and the feasible region remains unchanged. However, the weights have to be determined *a priori* and an even distribution of weights does not always yield an even distribution of Pareto solutions. The Pareto points generated are strongly dependent on the weight vectors used and a poor choice might lead to points that are clustered. Alternatively, weight vectors can be randomly generated to try and obtain a sufficient coverage of the Pareto front and to overcome these deficiencies; this is the approach taken in our study. However, the use of randomly generated the weight vectors might also be time-consuming as a large number of such vectors may be required to achieve good coverage of the Pareto front. Before formulating the scalarized problem, each objective function is normalized (f_i) with respect to the limits of the objective space to avoid generating search directions that are biased towards specific (larger) objectives. The mathematical formulation of the scalarized problem is as follows:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}, \mathbf{n}} \quad & \sum_{i=1}^p w_i f_i(\mathbf{x}, \mathbf{y}, \mathbf{n}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{n}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}, \mathbf{y}, \mathbf{n}) = \mathbf{0} \\ & f_i(\mathbf{x}, \mathbf{y}, \mathbf{n}) = \frac{f_i^o(\mathbf{x}, \mathbf{y}, \mathbf{n}) - f_{i,\text{nadir}}(\mathbf{x}, \mathbf{y}, \mathbf{n})}{f_{i,\text{utopia}}(\mathbf{x}, \mathbf{y}, \mathbf{n}) - f_{i,\text{nadir}}(\mathbf{x}, \mathbf{y}, \mathbf{n})}, \quad i=1, \dots, p \\ & \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \{0, 1\}^q, \mathbf{n} \in \mathbf{N} \subset \mathbb{Z}^q \end{aligned} \quad (2)$$

where $f_i^o(\mathbf{x}, \mathbf{y}, \mathbf{n})$ is the i th original objective function, $f_i(\mathbf{x}, \mathbf{y}, \mathbf{n})$ is the i th normalized objective function and $f_{i,\text{utopia}}(\mathbf{x}, \mathbf{y}, \mathbf{n})$ is the lowest possible value of i th objective function and can be obtained by minimizing a single objective function, $f_i^o(\mathbf{x}, \mathbf{y}, \mathbf{n})$. $f_{i,\text{nadir}}(\mathbf{x}, \mathbf{y}, \mathbf{n})$ is the highest possible value of the i th objective function in the entire Pareto-optimal set.

The general scheme of the WS algorithm is outlined in Algorithm 1.

2.2.2. Sandwich algorithm

The SD algorithm (Solanki et al., 1993; Rennen et al., 2011) has been proposed as a scalarization method with the aim to approx-

Algorithm 1 Weighted sum method for CAM(P)D.

```

1: procedure WEIGHTED SUM ALGORITHM
2:    $W = \emptyset$ ; iteration,  $k = 1$ , where  $W$  is a set of weight vectors
3:   while Stopping criterion is satisfied (i.e.  $k \leq N_{\text{iter}}$ ) do
4:     Generate a random vector  $\nu$  from uniform distribution
       $\nu = \{\nu_{i,k} \mid i = 1, \dots, p\}$ 
5:     Calculate a weight vector  $\mathbf{w}_k = \{w_{i,k} \mid i = 1, \dots, p\}$ ,
       $w_{i,k} = \nu_{i,k} / \sum \nu_{i,k}$ 
6:     if  $\mathbf{w}_k \notin W$  then
7:       Solve problem (2)
8:     end if
9:      $k = k + 1$ ,  $W = W \cup \mathbf{w}_k$ 
10:   end while
11: end procedure

```

imate the (convex) Pareto front with as few optimization runs as possible (Rennen et al., 2011). Within the algorithm, a convex hull (inner approximation) and outer approximation of the Pareto front are constructed sequentially based on the incumbent Pareto points until the approximation error falls below some given tolerance, $\epsilon \geq 0$. The approximation error, d_{max} , is defined as the maximum distance between the inner and outer approximations. The weight vectors are chosen systematically from the set of normal vectors to the facets constructed by inner approximation. The basic scheme of SD is outlined in Algorithm 2. A graphical illustration of the procedure is shown in Fig. 1. The error tolerance is selected for each case study depending on the size of the problem.

Algorithm 2 Sandwich algorithm for CAM(P)D.

```

1: procedure SANDWICH ALGORITHM
2:    $Z = \emptyset$ ;  $d_{\text{max}} = \infty$ ;  $k = 1$ , where  $Z$  is a set of Pareto optimal
   solutions.
3:   Find all anchor points  $Z^A = \{z_i^A \mid i = 1, \dots, p\}$ .
4:   Solve problem (2) to obtain the first point  $z^1$ .
5:   while  $d_{\text{max}} \geq \epsilon$  do
6:     Initialize a set of inner approximation, IPS
       where IPS is obtained by constructing convexhull based
       on the set
        $Z \text{ IPS} = \text{convexhull}(Z), Z = \{z_1^A, \dots, z_p^A, z^1, \dots, z^k\}$ .
7:     Calculate the error,  $d_{\text{error}}$ , for each facet of IPS.
8:     Select the facet,  $FS^*$ , that has largest error,  $d_{\text{max}} = \max(d_{\text{error}})$ .
       Let  $H(\mathbf{w}, \mathbf{b})$  be a supporting hyper plane at  $FS^*$  then,
        $H(\mathbf{w}, \mathbf{b}) : \mathbf{w}^\top \mathbf{z} = b$ 
9:     Determine  $\mathbf{z}^*$  by solving problem (2) with  $\mathbf{w}$ 
10:    if  $(\mathbf{w}^\top \mathbf{z}^* = b)$  then
11:      Set the error of the  $FS^*$  to zero and return to Step 8
12:    end if
13:    Update outer approximation by adding the inequality
        $\mathbf{w}^\top \mathbf{z}^* \leq b$ 
14:    Update  $Z$  by replacing it with  $Z = \{Z, \mathbf{z}^*\}$ 
15:    Set  $k = k + 1$ 
16:  end while
17: end procedure

```

2.2.3. Global phases

A challenge in applying the WS and SD algorithms to MOO problems is ensuring that the global solution of each scalarized problem is identified, as suboptimal solutions are dominated solutions of the MOO problem rather than Pareto-optimal solutions. To increase the likelihood of finding global solutions, two global optimization algorithms are used to solve the scalarized problems at step 7 of Algorithm 1 and step 4 and 9 of

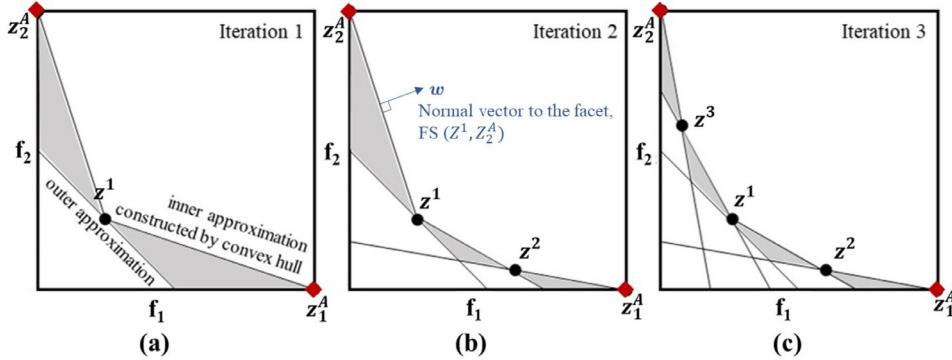


Fig. 1. Illustration of the sandwich algorithm applied to a bi-objective optimization problem: (a) an initial inner approximation is obtained by constructing the convex hull based on anchor points z_1^A, z_2^A (red diamonds) and the Pareto point z^1 (black circle). This gives two facets, with the Pareto front known to lie between the outer and inner approximation. (b) The approximation error (largest distance between the inner and outer approximations) is computed for each facet. The Pareto point z^2 is obtained by solving problem (2) for the facet (inner approximation) that has the largest approximation error. The weight vector w is defined by the normal vector of the facet. (c) The Pareto point z^3 is obtained in the same manner as described in (b) after calculating the approximation error between inner and outer approximations based on z_2^A, z^1 and z^2 . This figure is based on Fig. 1 in Bokrantz and Forsgren (2011).

Algorithm 2. MLSL, a deterministic incomplete global optimization method (Neumaier, 2004), and SA (Marcoulaki and Kokossis, 2000), a stochastic incomplete global optimization method, are applied in our study. These are briefly described in the remainder of this section.

Multi-level single linkage (MLSL)

The MLSL method is derived from the multi-start method, where the aim is to find local minima by performing multiple local minimization procedures from a set of starting points distributed in an appropriate way (e.g., sampled from a uniform distribution) over the decision space (Rinnooy Kan, 1987). A major difference between the multi-start and MLSL methods is that the local search in MLSL is invoked from a reduced set of sample points, N_r , chosen such that there is no other sample point within some critical distance, r_k , that has a lower objective function value. The reduced set is obtained by taking $N_r = \gamma k' N_s$ points from the cumulated sample points that have the lowest function values, where $\gamma \in (0, 1)$ is a control parameter, k' is the number of iterations and N_s is a total number of sample points. Starting points that do not satisfy these conditions are clustered into one of the regions of attraction, C_l , $l=1, \dots, W_{\min}$, where W_{\min} is the number of distinct local minima that have been found up to the current iteration of the algorithm. Here, we denote the cardinality of the set C_l as $|C_l|$. The main algorithmic options for this class of approaches are the choice of an appropriate acceptance/rejection criterion for local search, of the sampling methodology and of a stopping criterion for the overall algorithm. In our study, we modify the algorithm of Kucherenko and Sytsko (2005) to account for the mixed-integer nature of problem (2). We adopt a Sobol' sequence (Sobol' et al., 2011) to generate starting points. Sample points are generated in the space of continuous variables $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{n}}$, where $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{n}}$ are obtained by relaxing the integrality of \mathbf{y} and \mathbf{n} , with the k th point denoted by $(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{n}}^{(k)})$. To enable the evaluation of the objective function and subsequent clustering at each sample point, it is imperative to ensure that the point is feasible. For this purpose, a pure integer problem is solved to find the closest point $(\mathbf{y}^{(k)}, \mathbf{n}^{(k)})$ to $(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{n}}^{(k)})$, as measured by the Euclidean distance, that satisfies the integer constraints. The formulation of this problem at iteration k is given by

$$\begin{aligned} \min_{\mathbf{y}^{(k)}, \mathbf{n}^{(k)}} \quad & \|(\mathbf{y}^{(k)}, \mathbf{n}^{(k)}) - (\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{n}}^{(k)})\|_2 \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{y}^{(k)}, \mathbf{n}^{(k)}) = 0 \\ & \mathbf{y} \in \{0, 1\}^q, \quad \mathbf{n} \in \mathbf{N} \subset \mathbb{Z}^q, \quad n_{k,i} \in \{n_{l,i}, n_{u,i}\}, \quad i = 1, \dots, q \end{aligned} \quad (3)$$

where $\|\cdot\|_2$ is the Euclidean norm and $n_{l,i}$ and $n_{u,i}$ are lower and upper bounds on each integer variable.

In CAMD problems where degrees of freedom are fully specified once the binary and integer variables are fixed, the continuous variables at sample point $(\mathbf{y}^{(k)}, \mathbf{n}^{(k)})$ are obtained by solving the model equations $\mathbf{h}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, \mathbf{n}^{(k)}) = \mathbf{0}$. The objective function of problem (2) can then be evaluated to determine whether a local (MINLP) optimization should be run from this starting point. In CAMPD problems, there are usually the degrees of freedom for continuous variables and the generation of starting points is applied to both integer and continuous variables. Such variables can be generated using a Sobol' sequence restricted to the feasible region based on the method as described in Section 4.3.1. In our study, we focus on generating a set of starting points only for the integer and binary variables, since the outer approximation (OA) algorithm (Viswanathan and Grossmann, 1990) is used as a local search method to solve Problem (2). Within this algorithm only integer and binary variables are specified as the starting points and the continuous variables are specified by solving Primal problems within the OA algorithm.

The stopping rule is given by $W_{\exp} \leq W_{\min,k}$, where W_{\exp} is the number of local minima calculated based on a Bayesian stopping rule (Pál, 2013). The maximum number of generated starting point, N_{\max} , is also used as the stopping criteria and set to $N_{\max} = 10 \times N_s$. The pseudo-code for the MLSL algorithm is outlined in Algorithm 3.

Simulated annealing (SA)

SA algorithms originate from the analogy of the heating and slow cooling of liquid metal so that crystallization into a structure that corresponds to a minimum free energy can be achieved (Metropolis et al., 1953). Markov processes and probability theory are combined in order to move from one state to another based on transition probabilities. The essential algorithmic feature of SA is that it encourages extensive exploration of the search space by accepting "worse-performing" random moves, thereby increasing the likelihood of convergence to a global solution. The choice of algorithmic parameters is a critical aspect of any implementation of SA (Papadopoulos and Linke, 2006; Marcoulaki and Kokossis, 2000). Relevant parameters include: the length of the Markov chain (N_{mc}), cooling schedule, cooling parameter (δ), stopping criterion, initial temperature (T_i), stopping temperature (T_{stop}), and the method used for the generation of perturbation moves. In our study, we employ the method described in Papadopoulos and Linke (2006) to determine appropriate parameter settings. The Markov chain length is selected by examining the standard deviation in the value of the optimal objective. Specifically, ten SA

Table 1
Algorithmic parameters used in the case studies, where CS denotes case study.

Method	Parameter	CS1	CS2	CS3
MLSL	Number of Sobol' points, N_s	128	256	128
	Fraction to define reduced set, γ	0.25	0.25	0.5
	Control parameter, σ	3	3	2
SA	Initial temperature, T_i	10^5	10^5	10^5
	Final temperature, T_{stop}	10^{-3}	10^{-3}	10^{-3}
	Markov chain length, N_{mc}	70	120	80
SD	Cooling parameter, δ	0.8	0.5	0.2
	Error tolerance, ϵ	9.5×10^{-3}	1.5×10^{-2}	5.0×10^{-4}
	Crossover fraction	0.2	0.2	0.2
NSGA-II	Mutation fraction	0.8	0.8	0.8
	Elite gene preservation fraction	0.05	0.05	0.05
	Population size	100	150	50

Algorithm 3 Multi-level single linkage algorithm for CAM(P)D.

```

1: procedure MULTI-LEVEL SINGLE LINKAGE
2:   Initialize  $C_l = \emptyset$ ;  $k' = 1$ ;  $W_{exp} = 0$ ;  $W_{min} = 0$ , Set  $N_s$  and  $\gamma$ .
3:   Set list of objective functions and corresponding sample
   points,  $L = \emptyset$ .
4:   while  $W_{exp} \leq W_{min}$  or  $k'N_s \leq N_{max}$  do
5:     for  $k = (k' - 1)N_s + 1$  to  $k'N_s$  do
6:       Generate  $k$ th Sobol point,  $(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{n}}^{(k)}) \in [0, 1]^q \times$ 
       $[\mathbf{n}_l, \mathbf{n}_u]$ .
7:       Solve Problem (3) to find  $(\mathbf{y}^{(k)}, \mathbf{n}^{(k)})$ .
8:       Generate  $\mathbf{x}^{(k)}$  from  $(\mathbf{y}^{(k)}, \mathbf{n}^{(k)})$ .
9:       Evaluate objective function
    $f^{(k)} = \sum_{m=1}^p w_m f_m(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, \mathbf{n}^{(k)})$ .
10:      Add  $((\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, \mathbf{n}^{(k)}, f^{(k)})$  to  $L$ , so that  $L$  is sorted in
      ascending order of objective functions.
11:    end for
12:    Set  $N_r = \gamma k' N_s$ ;  $N_r$  sample points are selected from  $L$ .
13:    for  $i = 1$  to  $N_r$  do
14:      if  $W_{min} > 0$  and there exist  $j, l$  such that
          $f(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, \mathbf{n}^{(j)}) \geq f(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{n}^{(i)})$  and
          $\|(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, \mathbf{n}^{(j)}) - (\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{n}^{(i)})\| < r_{k'}$ ,
         where  $j \in \{1, \dots, |C_l|\}$  and  $l \in \{1, \dots, W_{min}\}$  then
           Assign the point  $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{n}^{(i)})$  to  $C_l$ .
15:      else
16:        Solve Problem (2) starting from  $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{n}^{(i)})$  us-
           ing a local search method.
17:        if feasible and solution  $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{n}^*) \notin C_l$  then
18:          Assign  $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{n}^*)$  to  $C_{W_{min}+1}$ ;  $W_{min} = W_{min} + 1$ .
19:        end if
20:      end if
21:    end if
22:  end for
23:  Set  $k' = k' + 1$ , Calculate  $W_{exp}$ 
24: end while
25: end procedure

```

simulations are executed with different seeds and starting points and the smallest chain length (N_{mc}) is selected for which the standard deviation is below 0.2%. The cooling parameters and cooling schedule are set according to [Marcoulaki and Kokossis \(2000\)](#) and [Aarts and Laarhoven \(1985\)](#), respectively. All other heuristic parameters used in the case studies are listed in [Table 1](#).

The difficulty in applying SA to CAM(P)D is in ensuring the random moves satisfy the structural feasibility constraints. Note that the random moves should not be biased, so an optimization problem is formulated and solved to find the nearest point $(\mathbf{y}'_k, \mathbf{n}'_k)$ to the newly generated points $(\mathbf{y}_k, \mathbf{n}_k)$, which satisfies all structure-related constraints. The formulation of this problem at iteration k

is given by

$$\begin{aligned} \min_{\mathbf{y}'_k, \mathbf{n}'_k} & \|(\mathbf{y}_k, \mathbf{n}_k) - (\mathbf{y}'_k, \mathbf{n}'_k)\|_2 \\ \text{s.t. } & \mathbf{h}(\mathbf{y}'_k, \mathbf{n}'_k) = 0 \\ & \mathbf{y}' \in \{0, 1\}^q, \quad \mathbf{n} \in \mathbf{N} \subset \mathbb{Z}^q, \quad n'_{k,i} \in \{n_{l,i}, n_{u,i}\}, \quad i = 1, \dots, q \end{aligned} \quad (4)$$

Due to the difficulty of handling design constraints in SA, these are handled via a penalty function ([Marcoulaki and Kokossis, 2000](#)), $F_k(\mathbf{x}_k, \mathbf{y}'_k, \mathbf{n}'_k; M)$. The molecular feasibility constraints are not included in the penalty function F_k , since only the structurally feasible molecules are generated within the algorithm. The penalty function used to combine the objective function and the constraints is given by:

$$\begin{aligned} F_k(\mathbf{x}_k, \mathbf{y}'_k, \mathbf{n}'_k; M) = & \sum_{i=1}^p w_i f_i(\mathbf{x}_k, \mathbf{y}'_k, \mathbf{n}'_k) \\ & + M \left(\sum_{i=1}^{m_g} [\max(0, g_i(\mathbf{x}_k, \mathbf{y}'_k, \mathbf{n}'_k))] \right. \\ & \left. + \sum_{i=1}^{m_h} |h_i(\mathbf{x}_k, \mathbf{y}'_k, \mathbf{n}'_k)| \right) \end{aligned} \quad (5)$$

where M is a penalty weight imposed on inequality and equality constraints, and m_g and m_h are the number of inequality and equality constraints, respectively.

The SA algorithm applied to CAM(P)D is outlined in [Algorithm 4](#).

2.3. Non-dominated sorting genetic Algorithm-II (NSGA-II)

The NSGA-II algorithm ([Deb et al., 2000](#)) is a particular form of genetic algorithm (GA) and is one of the most prominent methods for evolutionary-based stochastic search. The NSGA-II is directly applicable to MOO as it uses an explicit diversity-preserving mechanism based on a crowding distance metric to generate uniformly distributed Pareto points. Furthermore, it makes use of elite-preserving operators to give "elite genes" an opportunity to survive in the next generation. In order to adapt the NSGA-II for CAM(P)D problems, one needs to encode molecular structure in an operable and interpretable form, and to specify a suitable set of heuristic parameters. In our work, the dynamic tree structure proposed by [Zhou et al. \(2017\)](#) is implemented to represent molecules. The construction of the tree-like structure guarantees the generation of structurally feasible molecules and allows the application of the crossover and mutation operators over molecules to be more tractable. For CAM(P)D problems, a penalty-based ([Zhou et al., 2017](#)) fitness vector ($\mathbf{F}_{fit,k}$) is assigned to an individual solution, $f_{i,k}$, to measure their relative merit: $F_{fit,i,k} = f_{i,k}(\mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_k) \times P_k$. The fitness function is chosen to be proportional to the

Algorithm 4 Simulated annealing algorithm for CAM(P)D.

```

1: procedure SIMULATED ANNEALING
2:   Initialization  $\mathbf{y}_0, \mathbf{n}_0, \mathbf{x}_0 \leftarrow$  initial guesses;  $T = T_i$ ; Set  $M$ .
3:   Calculate initial objective function value,  $F_0(\mathbf{x}_0, \mathbf{y}_0, \mathbf{n}_0; M)$ .
4:   while  $T \leq T_{stop}$  do
5:     for  $k = 1$  to  $N_{mc}$  do
6:       Generate random vector  $\Delta\mathbf{y}_k, \Delta\mathbf{n}_k$  and corresponding move
7:        $\mathbf{y}_k = \mathbf{y}'_{k-1} + \Delta\mathbf{y}_k, \mathbf{n}_k = \mathbf{n}'_{k-1} + \Delta\mathbf{n}_k$ 
8:       Solve Problem (4) to find  $\mathbf{y}'_k, \mathbf{n}'_k; \mathbf{x}_k = \mathbf{x}(\mathbf{y}'_k, \mathbf{n}'_k)$ 
9:        $\Delta F_k = F_k(\mathbf{x}_k, \mathbf{y}'_k, \mathbf{n}'_k; M) - F_{k-1}(\mathbf{x}_{k-1}, \mathbf{y}'_{k-1}, \mathbf{n}'_{k-1}; M)$ .
10:      if  $\Delta F_k \geq 0$  then
11:        Sample random number,  $P_{r,k}$  from uniform distribution function and calculate probability of accepting the random move,  $P_{accept}$ , where  $P_{accept} = \exp(-\Delta F_k/T)$ .
12:        if  $P_{accept} \leq P_{r,k}$  then
13:          Discard the value of  $\mathbf{n}'_k$  and  $F_k$  and assign the values from the iteration  $k-1$ ;  $F_k = F_{k-1}$  and
14:           $\mathbf{y}'_k = \mathbf{y}'_{k-1}, \mathbf{n}'_k = \mathbf{n}'_{k-1}$ 
15:        end if
16:      end if
17:    end for
18:    Set  $k = k + 1$ 
19:  Reduce the Temperature  $T$  based on cooling schedule.
end while
end procedure

```

magnitude of an aggregate constraint violation, P_k : $P_k = 1000 \times (\sum_{j=1}^{m_g} \max[0, g_j(\mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_k)] + \sum_{j=1}^{m_h} |h_j(\mathbf{x}_k, \mathbf{y}_k, \mathbf{n}_k)|)$. The tournament selection technique is employed as a selection strategy to choose parents for the next generation, since it can avoid premature convergence and stagnation (Deb et al., 2000). In tournament selection, four population members are randomly chosen to compete with each other and the best one out of the pool of members is selected to be a parent. All other parameters used in the case studies are specified in Table 1.

2.4. Algorithmic combinations

The combination of global phase and MOO methods results in four scalarization algorithms: weighted sum with multi-level single linkage (WSML), weighted sum with simulated annealing (WSSA), sandwich algorithm with multi-level single linkage (SDML) and sandwich algorithm with simulated annealing (SDSA). This set of four combinations is augmented with the weighted sum without global phase (SWS) and non-dominating sorting genetic algorithm II (NSGA-II), which is not based on scalarization. These approaches are summarized in Table 2. Convergence to the global optimal solution of the single objective optimization problem, Problem 1, is especially critical in the SD algorithm since the algorithm uses the incumbent solution set to generate the convex hulls and weight vectors. On the basis of this observation, the comparison of the performance of the algorithms is divided into two stages: a com-

parative study of the performance of two global search methods with the WS approach is first carried out; in the second stage, only the global search method that gives better performance is used in further comparisons. The methods compared in the first stage include the SWS, WSML, and WSSA. As will be shown, MLSL provides the best performance for this investigation so that WSML, SDML and NSGA-II are compared in the second stage.

3. Quality measures for multi-objective optimization

Since no single metric can represent the performance of the algorithms, a series of appropriate metrics is used to assess performance in the specific domain of molecular design. In particular, in developing the metrics, we take into account the fact that some of the case studies involve only discrete decision variables, in which case a full enumeration of the solution is possible and provides further insights. Other case studies include both discrete and continuous variables and such an enumeration is not possible. Thus, we define the following two quantities:

1. The exact number of Pareto points (N_{true}): For some CAMD problems, the exact set of Pareto points, P_T , can be obtained by complete enumeration of all possible combinations of the functional groups. This set contains N_{true} Pareto points, i.e. $N_{true} = |P_T|$ where $|A|$ denotes the cardinality of set A .
2. The number of best-known Pareto points (N_{BP}): For some CAMD and CAMPD problems, the best-known Pareto front, P_{BP} , contains a set of non-dominated points (i.e. $N_{BP} = |P_{BP}|$) obtained by evaluating dominance across all points generated from several runs for a given problem. It is an approximation of the true Pareto front, which may contain some points that are not Pareto-optimal and/or may be missing Pareto points.

The following quantitative metrics are used to measure the quality of the solution set obtained following a MOO run:

1. The number of unique non-dominated solutions (N_{ung}): The set P_U contains all the unique non-dominated solutions obtained in a given run with $N_{ung} = |P_U|$. This number captures the diversity of the solution set.
2. The number of non-dominated points (N_{PF}): The set P_{PF} contains the N_{PF} non-dominated points generated in a given run that lie on the true Pareto front (or the best-known Pareto front), i.e. $P_{PF} = P_U \cap P_T$ (or $P_{PF} = P_U \cap P_{BP}$). N_{PF} is a measure of the extent of convergence to the Pareto front.
3. The number of supporting non-dominated points (N_{SPF}): The P_{SPF} contains the non-dominated points that have been identified during a SOO in which the weight vector supports the Pareto front, i.e. $P_{SPF} \subseteq P_{PF}$. Specifically, if a point is in P_{SPF} , this indicates that the point was identified following termination of the SOO algorithm at a global solution rather than at a local solution. If a point in P_{PF} is not in P_{SPF} , this point was identified following convergence of the SOO algorithm to a local solution, indicating a fortunate outcome to what is essentially a failure of the global SOO solver.
4. Hypervolume (HV): The hypervolume (Zitzler et al., 2003) of a set P_{PF} of non-dominated solutions is the volume of the p -dimensional region in the objective space enclosed by the non-dominated solutions obtained and a reference point, $f_{ref,i}$: $f_{ref,i}=1, i = 1, \dots, p$. The larger HV, the better the P_{PF} in terms of convergence to the true Pareto front and/or in terms of diversity of the solutions.
5. CPU time: Fast convergence to the Pareto frontier is a critical aspect for computationally-expensive MOO problems. Both the average CPU time to generate a true (or best known) solution ($T_{CPU,a}$) and the total CPU time to generate all solutions ($T_{CPU,t}$) are reported.

Table 2

Algorithmic used in this work.

Algorithm name	Pseudo code
SWS	Algorithm 1 + local MINLP solver for Step 7
WSML	Algorithm 1 + Algorithm 3 for Step 7
WSSA	Algorithm 1 + Algorithm 4 for Step 7
SDML	Algorithm 2 + Algorithm 3 for Step 9
NSGA-II	-

4. Case studies

The optimization methodologies presented are applied to three case studies to assess their performance and to examine the applicability of each method. The case studies are selected so that different levels of complexity are explored in terms of problem size and numerical difficulty.

4.1. Case study 1: solvent design for chemical absorption of carbon dioxide (CAMD)

A CAMD application focused on the design of solvents for the chemical absorption of CO₂ is chosen as a first example. The formulation is based on the first problem of the recent approach proposed by Papadopoulos et al. (2016) who considered an extensive list of property criteria for the design of solvents for the CO₂ capture. Four solvent properties are selected as objective functions in this case study, namely, the liquid density (ρ), heat capacity (C_p), saturated vapour pressure (P_{vap}) and relative energy difference (RED). With four objectives and only discrete decision variables, this case study provides an opportunity to enumerate the entire search space and gain insights into some of the key features of CAMD MOO problems. Several other performance criteria are considered as constraints: a) the normal melting point (T_m), to ensure that the solvent is in the liquid state at the lowest process operating temperature; b) the normal boiling point (T_b), to avoid excessive evaporation at absorber operating conditions; c) the viscosity (μ), to ensure ease of transport; and d) the surface tension (σ), to promote mass transfer performance. The property targets and performance criteria are summarized in Table 3. Here, we focus on the first stage of the method of Papadopoulos et al. (2016). Solving their first subproblem provides an initial set of solutions that must be analyzed further using other criteria such as reactivity (Papadopoulos et al., 2016) and synthesizability (Zhang et al., 2015). Nevertheless, it is useful to investigate the performance of MOO algorithms on this restricted subproblem as it allows us to consider an example with four objective functions.

The SAFT- γ Mie group contribution-equation of state (Papaioannou et al., 2014) is used to predict ρ , C_p , P_{vap} , and T_b . The property-prediction method of Hukkerikar et al. (2012) is used for T_m , Hsu et al. (2002) is used to predict σ , RED and μ . A set of 13 functional groups ($N = 13$), occurring n_i ($i = 1, \dots, N$) times in the molecule including 8 amine groups is selected as building blocks based on the applicability of the property prediction methods. The set is shown in Table 4 and includes eight amine groups. The total number of functional groups in the

molecule and the total number of groups with amine functionality are constrained by bounds n_{tot} and $n_{tot,A}$, respectively. The set of amine group is given by $G_A = \{\text{CH}_2\text{N}, \text{CH}_3\text{N}, \text{CHNH}, \text{CH}_2\text{NH}, \text{CH}_3\text{NH}, \text{CNH}_2, \text{CHNH}_2, \text{CH}_2\text{NH}_2\}$. In this case study, only acyclic molecules are considered. The resulting mathematical formulation is as follows:

$$\begin{aligned} \min_{\mathbf{n}} \quad & C_p, P_{vap}, \text{RED} \\ \max_{\mathbf{n}} \quad & \rho \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{n}) \leq \mathbf{0} \\ & \sum_{i=1}^N (2 - v_i)n_i - 2 = 0, \\ & \sum_{j \in G_A} n_j - n_{tot,A} \leq 0, \\ & \sum_{i=1}^N n_i - n_{tot} \leq 0 \\ & n_i \in \{n_{l,i}, n_{u,i}\}, i = 1, \dots, q \end{aligned} \quad (6)$$

where $\mathbf{g}(\mathbf{n})$ is a vector of inequality constraints on the physical properties (cf. Table 3) and v_i is the valence of group i .

4.2. Case study 2: working fluid design for ORC (CAMD)

In the second case study, we consider the design a working fluid for an ORC following the CAMD formulation of Papadopoulos et al. (2010) and Palma-Flores et al. (2015). The Pareto front generated by solving such a problem is useful in identifying high-performance ORC fluids as part of a two-stage CAMPD methodology (Papadopoulos et al., 2010). The purpose of this case study is to investigate the performance of each algorithm for a higher problem dimension. The objective functions for the design of the working fluid are expressed as five thermodynamic properties: the liquid density (ρ_L), which is indicative of the size of process equipment and should be maximized; the latent heat of vaporization (H_V), which is a measure of how much heat can be added to the ORC system; the thermal conductivity (λ_L), for which a higher value is desirable to obtain a larger heat transfer coefficient in the heat exchangers; the liquid heat capacity ($C_{p,L}$); and the viscosity (μ) to increase the heat transfer coefficient and achieve reduced energy consumption. In addition to the properties aforementioned, the melting temperature (T_m) and critical temperature (T_c) are also considered as property constraints in order to make sure that the working fluid is in the vapor-liquid coexistence across the range of operating conditions of the ORC. The predictive thermodynamic models used for each property and the relevant bounds are listed in Table 5. The molecular design space is generated from 23 functional groups, including hydrocarbon, ether, fluoro, amine, formate, aldehyde, and hydroxyl groups (Table 6). The design space of possible working fluids consists of compounds generated by linear combination of functional groups containing up to 6 functional groups. The resulting optimization problem is as follows:

$$\begin{aligned} \max_{\mathbf{n}} \quad & \rho_L, H_V, \lambda_L \\ \min_{\mathbf{n}} \quad & C_{p,L}, \mu \\ \text{s.t.} \quad & T_m(\mathbf{n}) \leq T_{\min,op}, T_{\max,op} \leq T_c(\mathbf{n}) \\ & \sum_{i=1}^N (2 - v_i)n_i - 2 = 0, \\ & \sum_{j \in G_E} n_j - n_{tot,E} \leq 0, \\ & \sum_{i=1}^N n_i - n_{tot} \leq 0 \\ & n_i \in \{n_{l,i}, n_{u,i}\}, i = 1, \dots, q \end{aligned} \quad (7)$$

where $T_{\min,op}$, $T_{\max,op}$ are the minimum and maximum operating temperature of the ORC system, respectively, n_{tot} is the maximum allowable number of groups in a molecule, $n_{tot,E}$ is the maximum

Table 3
Property constraints for case study 1.

Physical properties, $\mathbf{g}(\mathbf{n})$	Bounds
ρ (g/cm ³) at 25°C and 1 atm	[0.6, 1.5]
RED	[10 ⁻⁵ , 6.5]
T_b (K) at 1 atm	[393, 550]
T_m (K) at 1 atm	[273, 313]
σ (dyn/cm) at 25°C	[25, 60]
μ (cP) at 40°C	[10 ⁻⁵ , 60]

Table 4
Solvent design space for case study 1.

Functional groups	Bounds
CH ₂ N, CH ₃ N,	
CHNH, CH ₂ NH	
CH ₃ NH, CNH ₂ ,	$n_{tot} = 13$
CNH ₂ , CH ₂ NH ₂ ,	$n_{tot,A} = 5$
CH ₃ , CH ₂ ,	
CH, C, and OH	

Table 5

Property constraints for case study 2, ($T_{\max,op} = 353.15$ K, $T_{\min,op} = 308.15$ K, $T_{op} = 330$ K, T_b is normal boiling point of working fluid).

Physical property	Bounds	Reference
ρ_L (g/cm ³) at T_{op} and 1 atm	[0.2, 1.86]	Poling et al. (2001)
H_f (kJ/mol) at T_b	[0.2, 2.5]	Constantinou and Gani (1994)
λ_L (W/m-K) at T_{op}	[0.06, 1]	Sastri and Rao (1999)
Cp,L (J/mol K) at T_{op}	[110, 500]	Sahinidis and Tawarmalani (2000)
μ (cP) at 30°C and 1 atm	[$10^{-5}, 10$]	Hsu et al. (2002)
T_m (K) at 1 atm	[273, $T_{\min,op}$]	Constantinou and Gani (1994)
T_c (K)	[$T_{\max,op}$, 313]	Joback and Reid (1987)

Table 6

Solvent design space for case study 2.

Functional groups	Bounds
$\text{CH}_3, \text{CH}_2, \text{CH}_2=\text{CH}, \text{CH}=\text{CH},$	
$\text{CH}_3\text{O}, \text{CH}_2\text{O}, \text{FCH}_2\text{O}, \text{CF}_3, \text{CF}_2,$	
$\text{CH}_2\text{NH}_2, \text{CHNH}_2, \text{CH}_3\text{NH}, \text{CH}_2\text{NH}, \text{CH}_3\text{N},$	$n_{tot} = 6$
$\text{CH}_3\text{COO}, \text{CH}_2\text{COO}, \text{COO}, \text{COOH},$	$n_{tot,E} = 2$
$\text{CH}_3\text{CO}, \text{CH}_2\text{CO}, \text{CHO}, \text{HCOO}, \text{and OH}$	

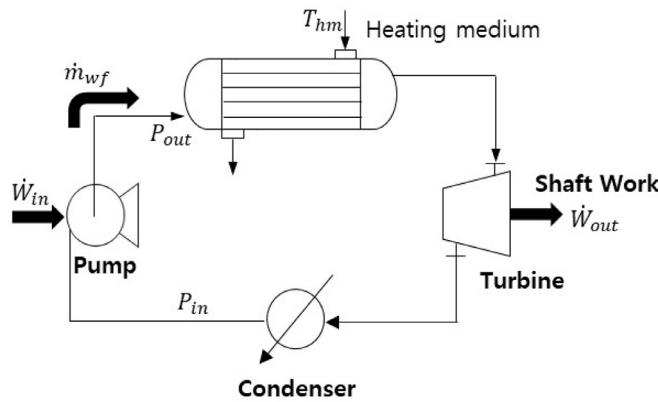


Fig. 2. Schematic of the organic Rankine cycle used in Case Study 3.

allowable number of end groups to ensure the molecule is a linear combination of the functional groups and $G_E = \{\text{CH}_3, \text{CH}_2=\text{CH}, \text{CH}_3\text{O}, \text{FCH}_2\text{O}, \text{CF}_3, \text{CH}_2\text{NH}_2, \text{CH}_3\text{NH}, \text{CH}_3\text{COO}, \text{COOH}, \text{CH}_3\text{O}, \text{CH}_3\text{O}, \text{CHO}, \text{HCOO}, \text{OH}\}$. Note that once a molecular structure has been specified, all of the continuous variables are fully determined.

4.3. Case study 3: integrated working fluid and ORC process design (CAMPD)

In this case study, we consider the integrated design of the working fluid and the ORC process. The aim of the optimization is to determine the optimal molecular structure of the working fluid as well as the process operating conditions that maximize the net power output (P_{net}) extracted from the ORC and to minimize the total cost of investment (TCI), for a specified heat source and heat sink (Schilling et al., 2017). The ORC system is defined as a single-stage, subcritical, non-recuperated cycle and comprises a turbine, a condenser, a pump, and an evaporator. A schematic of this process configuration is shown in Fig. 2. As a basis for the selection of the working fluid, the following nine function groups are selected: $\text{CH}_3, \text{CH}_2, \text{CH}_2=\text{CH}=, \text{eO}$ (end group oxygen), cO (central oxygen), and OH . eO represents an oxygen atom connected to one CH_3 and one CH_2 group, and cO describes an oxygen atom bonded to two CH_2 groups. As well as the functional groups, four ORC process variables (m_{WF} : mass flow of working fluid, $P_{in,pump}$, $P_{out,pump}$: pump inlet, outlet pressure, $\Delta T_{superheat}$: extent of superheating) can also be optimized. The optimization variables and

Table 7

Property constraints, $\mathbf{g}(\mathbf{x}, \mathbf{n})$ for case study 3. The bounds of pump inlet and outlet pressure (P_{\min}, P_{\max}) are determined by feasibility tests.

Physical properties, $\mathbf{g}(\mathbf{x}, \mathbf{n})$	Bounds
Mass flowrate of working fluid, m_{WF} (kg/s)	[0, 1000]
Extent of superheating, $\Delta T_{superheat}$ (K)	[0, 500]
Pump inlet pressure, $P_{in,pump}$ (Pa)	[P_{\min}, P_{\max}]
Pump outlet pressure, $P_{out,pump}$ (Pa)	[$P_{in,pump}, P_{\max}$]
Minimum approach temperature, ΔT_{\min} (K)	[10, 10 ⁵]

their upper and lower bounds are listed in Table 7. In solving the CAM(P)D problem, the feasible range of process variables is found to vary depending on the molecule used in the process; it therefore becomes challenging to provide reasonable starting points and bounds for the process variables, especially as many of the relevant bounds are typically expressed as implicit constraints (e.g., a two-phase system is expected in the evaporator and condenser) (Gopinath et al., 2016). As a result, these assumptions are readily violated when a new ORC fluid is selected and this usually leads to numerical failure in the calculation of phase behaviour. To avoid these failures, Bowskill et al. (2020) have extended the modified outer-approximation algorithm of Gopinath et al. (2016) and introduced feasibility tests to recognize the feasible domain for a specific choice of molecule. Here, we adopt the same optimization framework and modeling assumptions. For a detailed description of the optimization strategy and assumptions used, see Bowskill et al. (2020) and case study 2 (Table 1) therein. The generic formulation of case study 3 is as follows:

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{n}} \quad P_{net}, \\ & \max_{\mathbf{x}, \mathbf{n}} \quad TCI \\ & \text{s.t.} \quad \mathbf{g}(\mathbf{x}, \mathbf{n}) \leq 0, \\ & \quad \mathbf{h}(\mathbf{x}, \mathbf{n}) = 0, \end{aligned} \quad (8)$$

where $\mathbf{g}(\mathbf{x}, \mathbf{n})$ is a vector of inequality constraints (see Table 7) and $\mathbf{h}(\mathbf{x}, \mathbf{n})$ is a vector of equality constraints. The detailed equations for \mathbf{g} and \mathbf{h} are described in the supplementary information of Bowskill et al. (2020).

4.3.1. Remarks on the application of SA and NSGA-II to case study 3

For the application of the SA and NSGA-II algorithm to the solution of a single objective CAMPD problem, both integer variables and continuous variables need to be sampled. However, the feasible domain is not defined unless the molecular structure is selected. Here, we combine NSGA-II, to optimize integer (molecular) variables, with a gradient-based deterministic algorithm, to optimize the continuous nonlinear variables at the given molecular structure proposed by NSGA-II. For the application of SA, we divide the generation of random moves of SA approach in two parts: (1) the generation of a new molecule that can pass the feasibility tests of Bowskill et al. (2020); (2) the generation of continuous variables based on the feasible region derived for the new molecule.

Table 8

Problem size of the model for each case study (CS)

	CS1	CS2	CS3
Number of objectives	4	5	2
Number of total model variables	98	105	275
Number of discrete variables	13	24	9
Number of continuous design variables	85	81	266
Number of constraints	93	96	282

5. Results and discussion

In this section, we compare the relative performance of all algorithms. Table 8 summarizes the comparison of the problem size for the model defined in each case study. The feasibility tests are not considered in the problem size of case study 3. All MOO methods are implemented with common subfunctions using the same programming language in Matlab 2018a and all runs are performed on a single Intel(R) Xeon(R) Gold 5122 CPU @ 3.60GHz processor with 384 GB of RAM. For the local solution of MINLPs, the single-objective optimization problem is solved using an in-house implementation of the outer approximation algorithm with augmented penalty (Viswanathan and Grossmann, 1990) that interfaces with gPROMS ModelBuilder 5.1.1 (process model and NLP solver) and Gurobi 8.1 MILP solver to solve the primal and master problems iteratively. The SAFT- γ Mie thermodynamic calculations are carried out using gSAFT via a Foreign Object interface. We carry out two types of runs: fixed number of iterations and fixed CPU time. For the scalarization-based methods (WS and SD), the number of iterations is defined as the number of weight vectors that are explored. For NSGA-II, the number of iterations is defined as the number of generations. Where a limit is imposed on CPU time, this is selected based on the time taken for SDML to reach a pre-defined error tolerance for each case study.

5.1. True Pareto fronts

For case study 2 and case study 3, the true Pareto front is generated using exhaustive enumeration of a finite set of molecular structures in order to provide a benchmark for the quality of the solutions produced by each algorithm. For case study 3, it is impossible to construct a complete set of Pareto points as there is an infinite number of solutions due to the presence of continu-

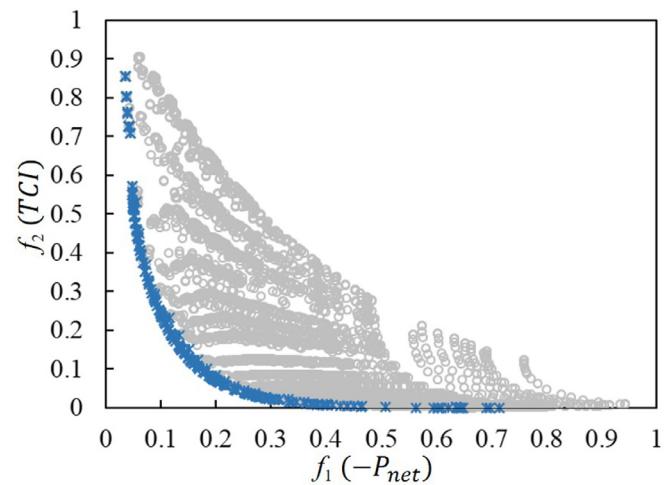


Fig. 4. Best-Known Pareto front (blue marker, \times) and feasible points (gray marker, \circ) of the normalized net power output, $-P_{\text{net}}$ (f_1), versus the normalized total capital investment, TCI (f_2), resulting from the enumeration of all possible working fluids in case study 3.

ous variables. As an alternative, 23 NLP optimizations corresponding to different weight vectors are conducted for each of the 267 feasible working fluids, resulting in 6141 NLPs. The resulting set of the solutions is combined with the solutions obtained with each method across all runs during all computational studies carried out in Sections 5.2 and 5.3 with the different algorithms. The results of these extensive calculations can be used as a reasonable approximation of the true Pareto front for graphical comparison. Information on the true/best-known Pareto frontier obtained for each case study is shown in Table 9. A graphical representation of the Pareto points in the domain of the objective function is given in Figs. 3 and 4. Within the 203 Pareto points identified in case study 3, there are only 12 distinct molecular structures, but a range of values of the continuous variables.

5.2. Comparisons between global search methods

We begin by comparing the effectiveness of MSL and SA in identifying global solutions using the same scalarization method

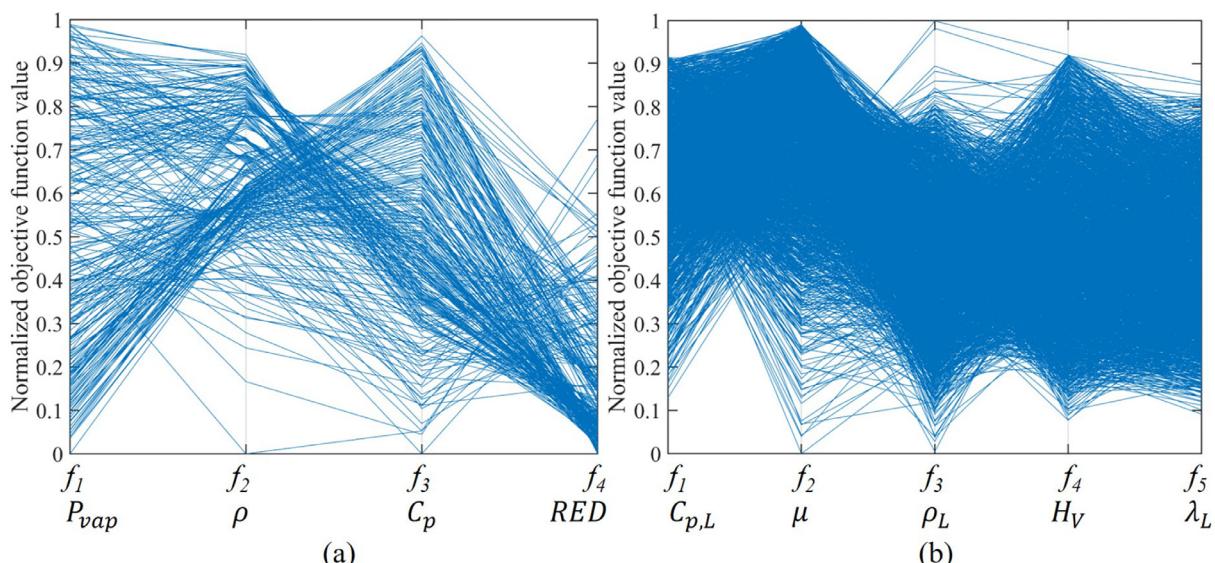


Fig. 3. Parallel coordinate plots of the set of true Pareto points on the: (a) four-objective space of case study 1; and (b) five-objective space of case study 2, obtained by the enumeration of all possible combinations of molecules.

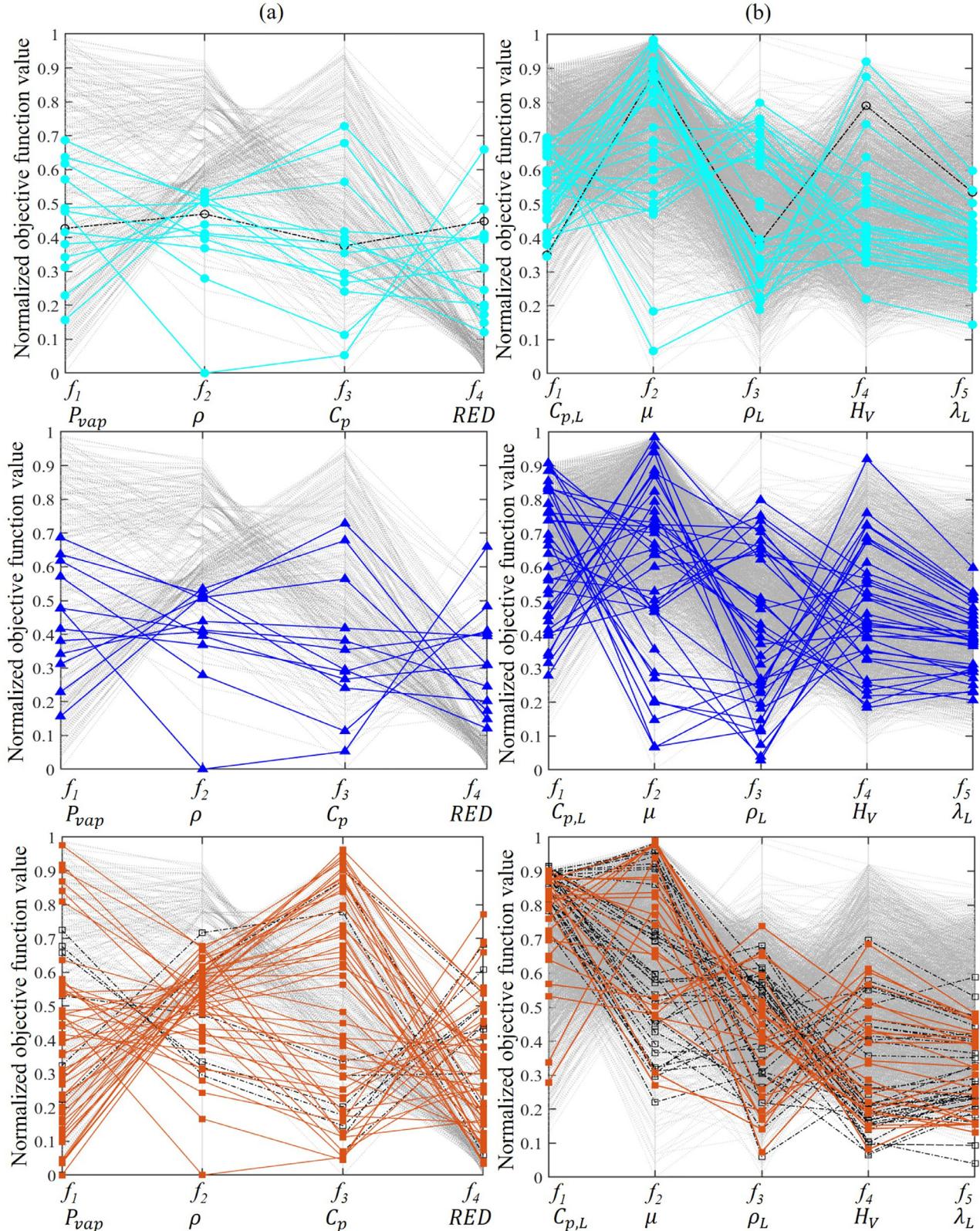


Fig. 5. Parallel coordinate plots of true Pareto fronts (P_{PF}) generated by SWS (—●—), WSML (—▲—), and WSSA (—■—) for (a) case study 1 and (b) case study 2, and true Pareto fronts (P_T) (—·—·—). The unique Pareto fronts (P_U) that do not intersect the P_{PF} , i.e. $P_U = (P_{PF} \cap P_U)$, are described as dash-dotted lines (—·—·—) and empty markers in each plot.

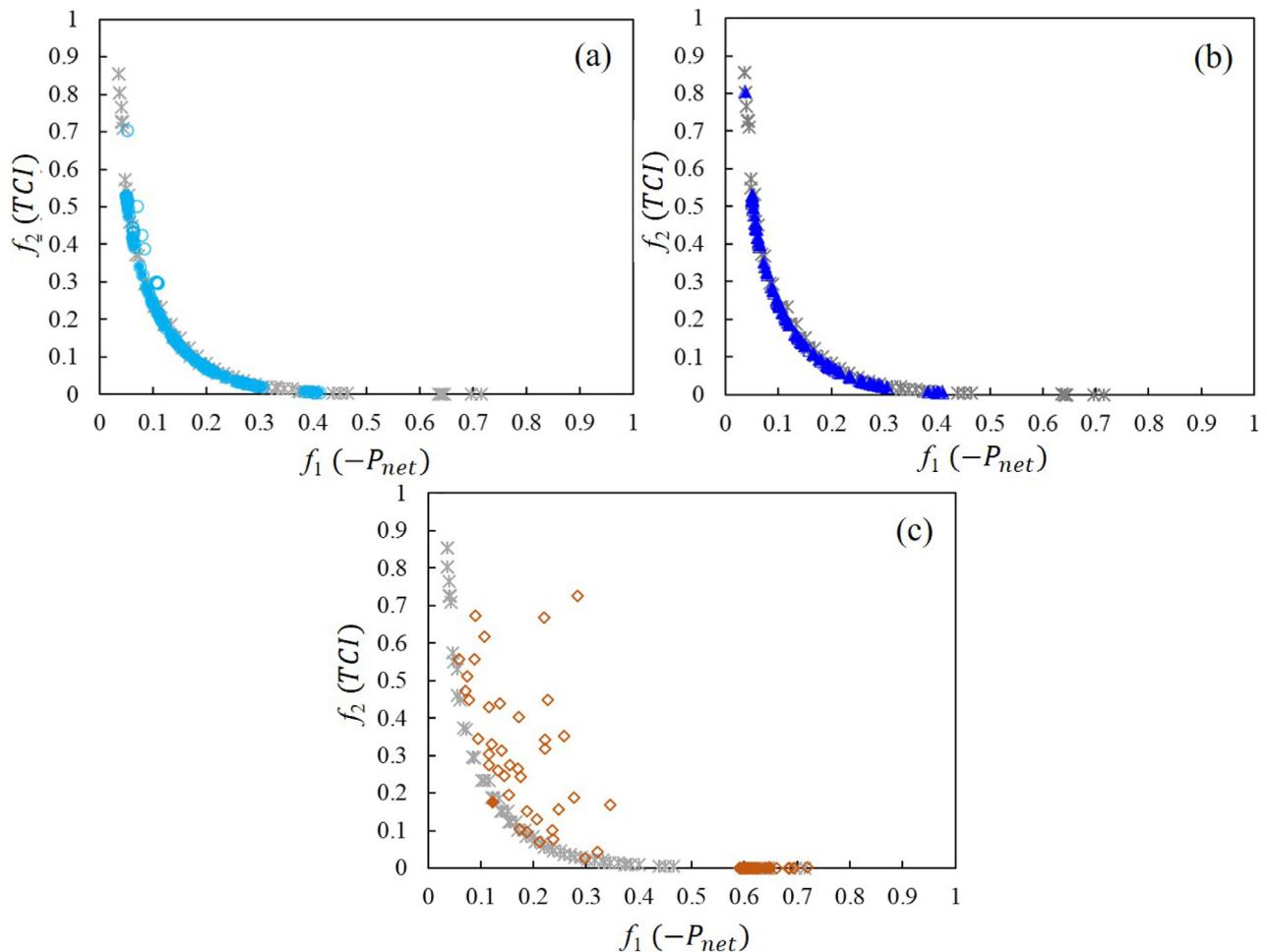


Fig. 6. Best-known Pareto fronts (P_{BP}) for case study 3 generated by (a) SWS (●), (b) WSM (▲), and (c) WSSA (◆). Gray markers (*) indicate the best-known Pareto points (P_B) obtained by full enumeration. The empty markers in each case study indicate the unique Pareto fronts (P_U) that do not intersect the best-known Pareto front, i.e. $P_U - (P_U \cap P_{BP})$.

Table 9

Enumeration of all solutions for all possible combinations of functional groups in each case study. $N_{structure}$: the number of molecules that satisfy structural feasibility, $N_{feasible}$: the number of molecules that satisfy all constraints, N_{true} : the number of exact Pareto Points, N_{BP} : the number of best-known Pareto Points, HV: hypervolume, $T_{CPU,t}$: total CPU time to enumerate the space in seconds. The value in (.) of N_{true} of case study 3 is the number of different molecular structures within true points.

	Case Study 1	Case Study 2	Case Study 3
$N_{structure}$	284,964	5,196,075	3175
$N_{feasible}$	6200	905,168	267
N_{true}	283	2748	-
N_{BP}	-	-	203 (12)
HV	0.4054	0.1435	0.9160
$T_{CPU,t}$ (s)	3.18×10^4	1.03×10^6	1.06×10^4

(WS). To compare SWS, WSM, and WSSA, an identical set of randomly generated weight vectors is used for all algorithms. We note that in all cases the number of iterations used falls far below the number of true or best known Pareto points. Thus these runs can only provide a partial view of the Pareto front and we therefore assess the efficiency with which Pareto points are generated. As can be seen in Tables 10, 11 and Fig. 5, WSSA allows one to generate more diverse solutions (larger N_{ung}) for case studies 1 and 2 compared to MSL. However, only 55–65% of solutions produced by SA are found to be true Pareto optimal points, and the solutions iden-

tified as true Pareto optimal points are not necessarily supported by the corresponding weight vector, and are thus obtained as a fortuitous result of the convergence of the SA algorithm to a local solution. For example, 34 of the solutions that are identified by WSSA as true Pareto points in case study 1 are sub-optimal solutions; hence, only four solutions lie on the convex Pareto front. Therefore, an algorithm with SA might mislead *a priori* articulation of preferences (weight vectors) of a decision-maker, since some solutions solved from specified weight vectors correspond to different weight vectors as a results of premature convergence to the Pareto front. This trend becomes more evident in case study 3. As shown in Fig. 6, many of the points identified by WSSA are not on the true Pareto front, and are therefore dominated by other solutions, resulting in the lowest value of N_{PF} for this case study. In addition, a comparison of the average CPU times in Tables 12 and 13 for WSM and WSSA suggests that the lowest value is obtained systematically with WSM, indicating that the use of MSL for SOO is more efficient. Furthermore, in all cases, we find that for WSM, $N_{PF} = N_{SPF}$, indicating that the MSL algorithm always converges to the global (or best-known) solution of Problem (2).

For the problems considered here, the SWS algorithm performs nearly as well as WSM. While the number of non-dominated solutions found is similar for both approaches, some of the points identified by SWS are local solutions ($N_{SPF} \leq N_{PF}$). However, the fact that similar values of HV are obtained for SWS and WSM indicates that these local solutions are close to the Pareto front. This

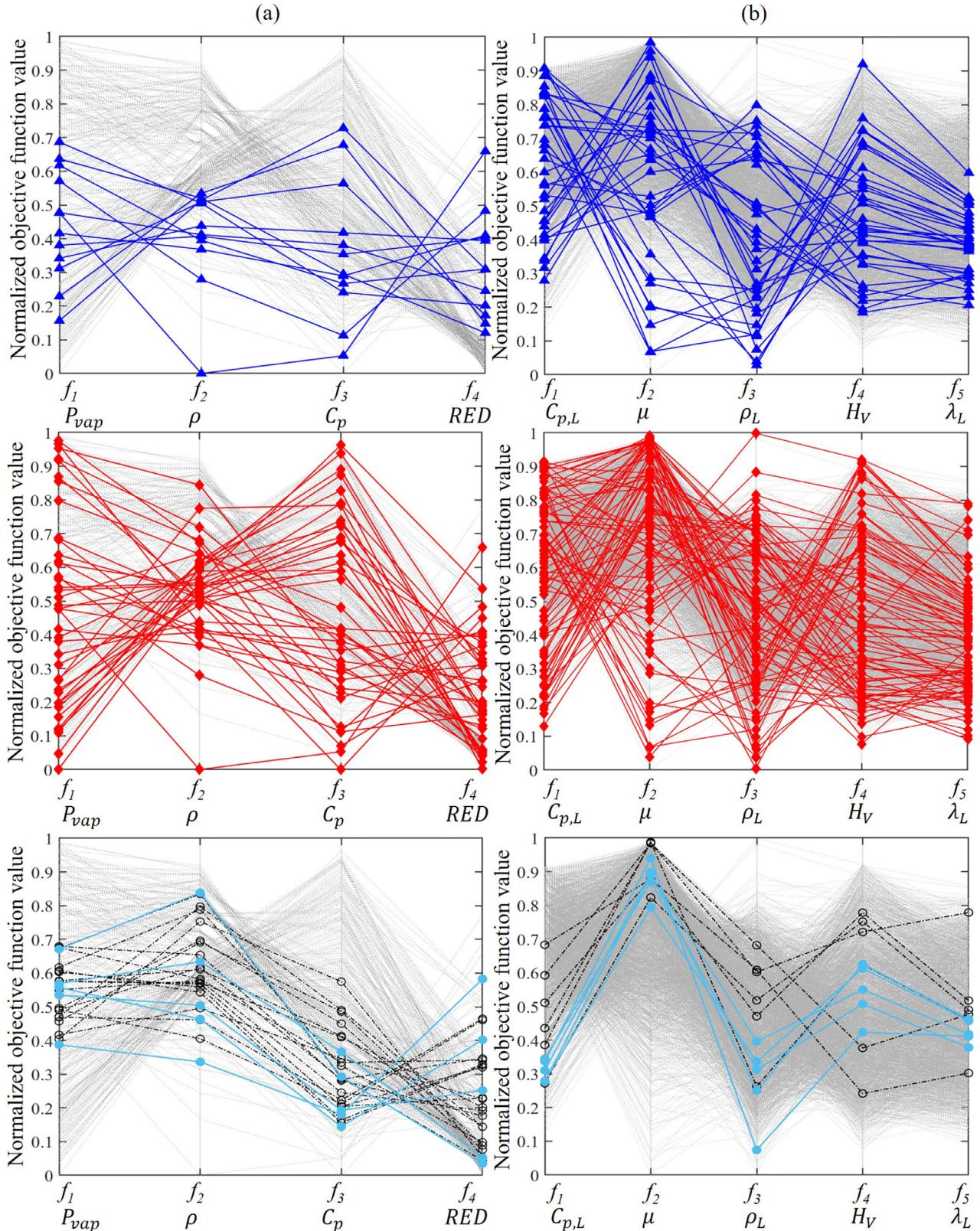


Fig. 7. Parallel coordinate plots of the exact true Pareto front (P_T) (---) and true Pareto fronts (P_{BF}) generated by WSML (—▲—), SDML (—◆—), and NSGA-II (—●—) for (a) case study 1 and (b) case study 2 for a fixed number of iterations. The unique Pareto fronts (P_U) that do not intersect the P_{PF} , i.e. $P_U - (P_{PF} \cap P_U)$, are described as dash-dotted lines (—·—) and empty makers in each plot.

Table 10

Performance metrics for case studies 2 and 3 using a fixed number of iterations and the weighted sum approach with three SOO approaches (SWS, MLSL and SA).

	Case Study 1			Case Study 2		
	SWS	WSML	WSSA	SWS	WSML	WSSA
N_{iter}	100	100	100	150	150	150
N_{unq}	14	12	58	30	35	69
N_{PF}	14	12	38	28	35	40
N_{SPF}	12	12	4	20	26	23
HV	0.3584	0.3592	0.3865	0.0745	0.1081	0.1003
$T_{CPU,a}$ (s)	1.41	20.26	800.07	2.74	16.68	304.11
$T_{CPU,t}$ (s)	1.98×10^1	2.47×10^2	3.04×10^4	7.67×10^1	5.84×10^2	1.22×10^4

Table 11

Performance metrics for case study 3 for a fixed number of iterations and the weighted sum approach with three SOO approaches (SWS, MLSL and SA).

	Case Study 3		
	SWS	WSML	WSSA
N_{iter}	100	100	100
N_{unq}	98	100	100
N_{PF}	83	95	7
N_{SPF}	52	95	1
HV	0.9084	0.9103	0.8872
$T_{CPU,a}$ (s)	1.76	88.93	4268.89
$T_{CPU,t}$ (s)	1.50×10^2	7.38×10^3	3.24×10^4

is also evident from Figs. 5 and 6. The greater confidence in the quality of the solutions afforded by WSML comes at a computational cost, as is always the case when global optimization is used.

Overall, since MLSL is found to show much better performance than WSSA across all case studies, only MLSL is be used as the global search method with WS and SD in the subsequent simulations.

5.3. Comparison between MOO methods

Having selected the MLSL algorithm to solve SOO problems to global optimality, we now focus on comparing the performance of the WSML, the SDML and the NSGA-II algorithms. The results of the three case studies with these algorithms for a fixed number of iterations are summarized in Tables 12, 13, Figs. 7 and 8. The results for a fixed CPU time are reported in Tables 14, 15, Figs. 9 and 10.

In case studies 1 and 2, each point in the feasible region corresponds to a distinct molecular structure. As a result, the Pareto set and set of dominated solutions are entirely disjoint in terms of the molecular structures they represent. In case study 3, it is possible for some molecular structures to appear both on the Pareto front and in the set of dominated solutions, as these may differ in terms of the continuous variables alone. However, in our experience, the dominated solutions generated by the NSGA-II algorithm

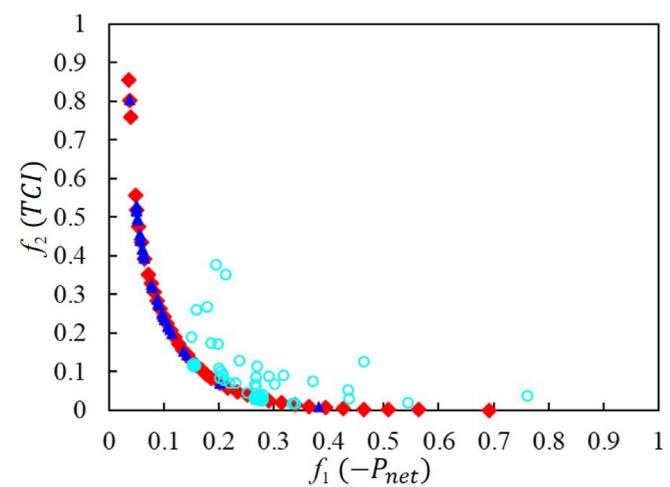


Fig. 8. Best-known Pareto fronts (P_{BF}) generated by WSML (\blacktriangle), SDML (\blacklozenge), and NSGA-II (\circ) for case study 3 for a fixed number of iterations. The empty markers in each case study indicate the unique Pareto fronts (P_U) that do not intersect the best-known Pareto fronts, i.e. $P_U - (P_U \cap P_{BF})$.

Table 12

Performance metrics for case studies 1 and 2 using WSML, SDML and NSGA-II for a fixed number of iterations.

	Case Study 3		
	WSML	SDML	NSGA-II
N_{iter}	40	40	40
N_{unq}	40	40	37
N_{PF}	39	40	3
HV	0.8617	0.9138	0.8261
$T_{CPU,a}$ (s)	50.47	55.32	325.03
$T_{CPU,t}$ (s)	1.97×10^3	2.31×10^3	9.75×10^2

correspond to different molecules from those on the “true” Pareto front. The set of optimal solutions (P_T , P_U , P_{PF}) is given in full in the Supporting Information (See the data statement).

Table 13

Performance metrics for case studies 1 and 2 using WSML, SDML and NSGA-II for a fixed number of iterations.

	Case Study 1			Case Study 2		
	WSML	SDML	NSGA-II	WSML	SDML	NSGA-II
N_{iter}	100	100	100	150	150	150
N_{unq}	12	40	24	35	83	11
N_{PF}	12	40	6	35	83	5
HV	0.3529	0.3917	0.2558	0.1081	0.1186	0.0353
$T_{CPU,a}$ (s)	20.60	8.83	43.98	37.49	7.58	546.45
$T_{CPU,t}$ (s)	2.47×10^2	3.53×10^2	2.64×10^2	5.84×10^1	6.29×10^2	1.64×10^3

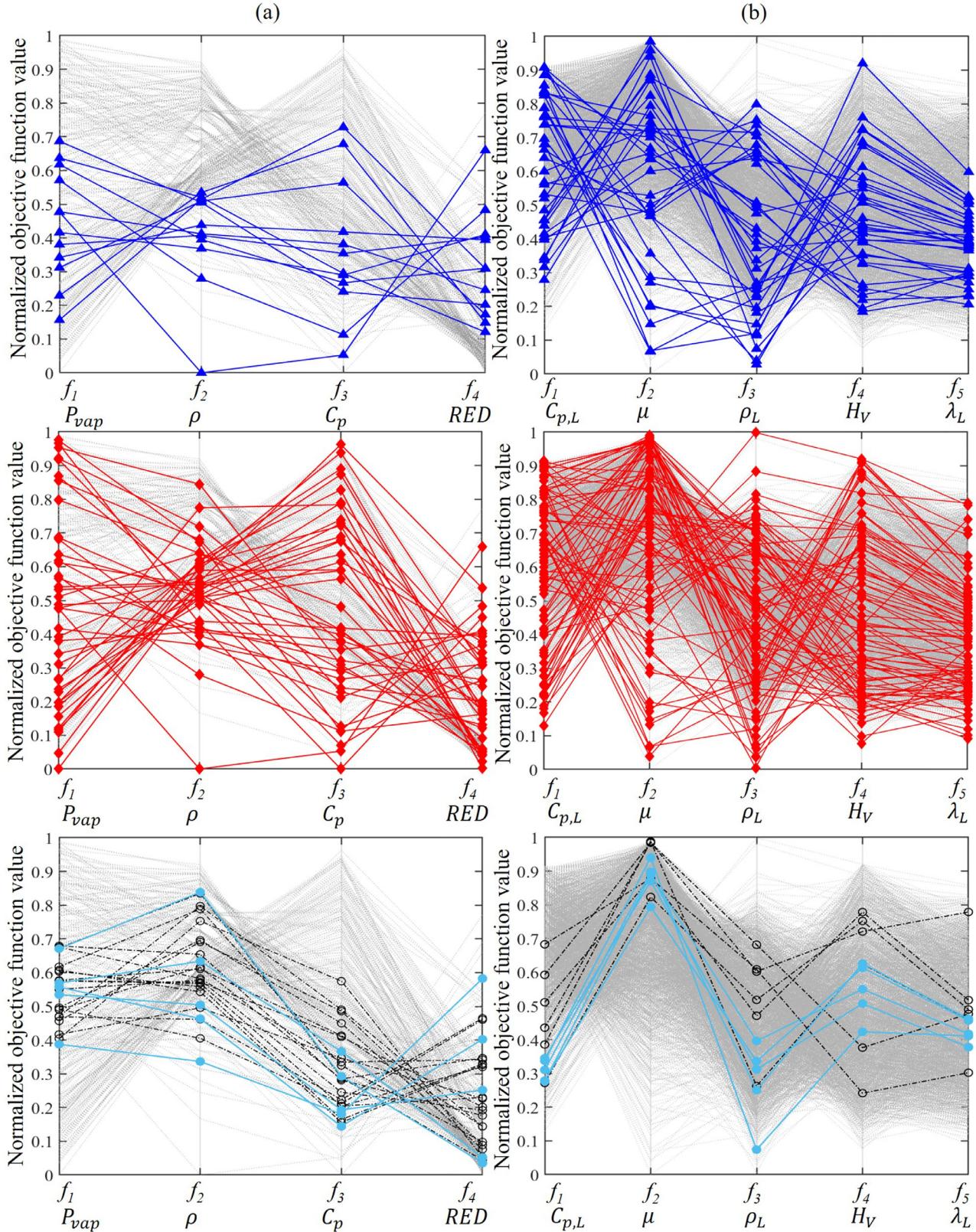


Fig. 9. Parallel coordinate plots of the exact true Pareto front (P_T) (---) and true Pareto fronts (P_{BF}) generated by WSML (—▲—), SDML (—◆—), and NSGA-II (—●—) for (a) case study 1 and (b) case study 2 for a fixed number of iterations. The unique Pareto fronts (P_U) that do not intersect the P_{PF} , i.e. $P_U - (P_{PF} \cap P_U)$, are described as dash-dotted lines (—·—) and empty makers in each plot.

Table 14

Performance metrics for case studies 1 and 2 using WSM, SDML and NSGA-II for a fixed CPU time.

	Case Study 1			Case Study 2		
	WSML	SDML	NSGA-II	WSML	SDML	NSGA-II
$T_{CPU,t}$ (s)	380	380	380	1000	1000	1000
N_{iter}	150	177	140	266	225	128
N_{unq}	13	46	28	51	108	22
N_{PF}	13	46	6	51	108	12
HV	0.3673	0.3928	0.3232	0.1121	0.1239	0.0798

Table 15

Performance metrics for case study 3 using WSM, SDML and NSGA-II for a fixed CPU time.

	Case Study 3		
	WSML	SDML	NSGA-II
$T_{CPU,t}$ (s)	1800	1800	1800
N_{iter}	35	31	135
N_{unq}	35	31	38
N_{PF}	34	31	2
HV	0.9038	0.9132	0.8888

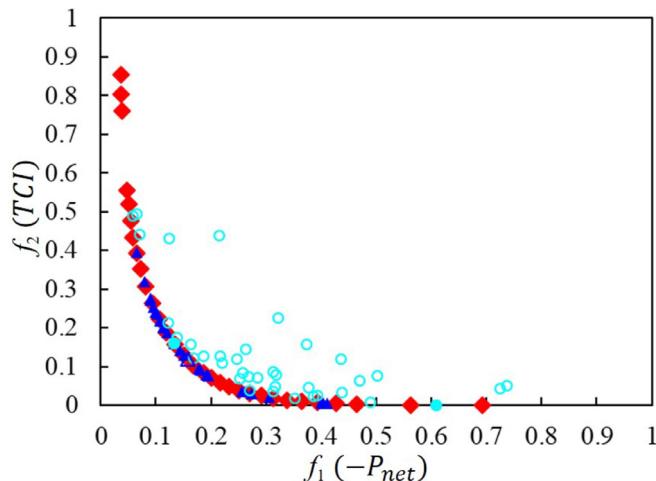


Fig. 10. Best-known Pareto fronts (P_{BF}) generated by WSM (▲), SDML (◆), and NSGA-II (●) for case study 3 for a fixed number of iterations. The empty markers in each case study indicate the unique Pareto fronts (P_U) that are not mutual to the best-known Pareto fronts, i.e. $P_U - (P_U \cap P_{BF})$.

Focusing first on a fixed number of iterations, we consider the values of HV and N_{PF} obtained with each approach as shown in Tables 12 and 13. The highest values of HV and N_{PF} for all cases are obtained when using SDML. These values of the HVs (0.3917 for case study 1, 0.1186 for case study 2, and 0.9138 for case study 3) are the closest to the HVs of the true Pareto fronts (0.4054 for case study 1, 0.1435 for case study 2, and 0.9160 for case study 3, see Table 9). From visual inspection of Figs. 7 and 8, it can be observed that SDML performs best in generating a diverse set of solutions close to the true Pareto front.

The performance of WSM is similar to that of SDML in terms of HV, but fewer points are identified in the Pareto front, indicating a less diverse set of solutions. The similar trends can be seen in the case of a fixed CPU time. The highest HV and N_{PF} values for case studies 1 and 2 are achieved with SDML for a given time as described in Table 14, showing 6–55% higher values of the hypervolume and 2.1–7.7 times higher values of N_{PF} , respectively. Although the highest N_{PF} can be obtained by WSM in the case study 3, the highest HV and similar N_{PF} are achieved with SDML

as can be seen in Table 15. This is because the direction of the weight vector in the SDML approach is updated deterministically in a sequence of iterations that increase the likelihood of finding Pareto points that are evenly distributed. The results confirm that diverse molecular structures and properties are found in the Pareto-optimal set generated by SDML.

In contrast, only a small number of chemical structures is identified with WSM in case studies 1 and 2. This indicates that randomly generated weight vectors do not always lead to different solutions. One may wonder why some of the solutions generated by SDML are non-unique for the pure-integer problem (i.e. case studies 1 and 2). This is mainly due to the fact that the mapping of the weights onto objective space assumes that the feasible region is continuous. As a result, the assigned weights may converge to one of the Pareto points previously identified if there are no supported solutions in the integer domain.

In Tables 12–15, it appears that NSGA-II exhibits a lower HV than other methods, although a larger number of unique points (N_{unq}) are generated for both stopping criteria. This suggests that the reliability of the Pareto front generated is not satisfactory when using NSGA-II with the parameters in Table 1. It can be observed that the use of the mutation and crossover operators make it challenging for the algorithm to generate feasible offspring when constraints are involved. Accordingly, this forces premature convergence to a sub-optimal front. This issue is likely to be especially acute in molecular design problems where many combinations of the integer variables are infeasible.

6. Conclusions

In this paper, we have compared several MOO algorithms by assessing their performance on molecular design applications. The algorithms include two types of scalarization-based methods and one evolutionary algorithm. In order to avoid premature convergence to a suboptimal front, two global search algorithms were combined with one of the scalarization methods (weighted sum) and tested for reliability and efficiency. Two CAMD case studies and one CAMPD case study, each with a different size of design space and numerical complexity, were employed to evaluate the performance of the algorithms.

For the global search phase, comparative results highlighted the robustness of the MSL algorithm in terms of computational efficiency and success in reaching a global solution. Furthermore, the findings from the case studies have provided clear evidence of the effectiveness of the SDML to solve CAM(P)D problems, relative to WSM and NSGA-II. SDML can be used to generate well-distributed Pareto fronts in comparatively few solves, requiring low computational effort. Although stochastic approaches have been successfully implemented in many practical problems, SA and NSGA-II have been found to encounter difficulties in converging to the Pareto optimal front for the case studies investigated. This is mainly attributed to the fact that the gene operators in NSGA-II and the random moves in SA cannot guarantee the generation of molecules that satisfy property constraints. Based on the re-

sults of our current work, SDML offers a very promising route to solve CAMD and CAMPD problems formulated as MOO and its performance should be investigated on additional case studies. The Pareto-solutions generated by SDML lie on convex portions of the Pareto front. Further work could be directed at testing the MOO algorithms on more case studies to derive general conclusions of their performance for the CAM(P)D problems. Combining the hyper-boxing algorithm suggested by [Bortz et al. \(2014\)](#) with SDML to explore non-convex regions of the Pareto front could also be considered in future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ye Seol Lee: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resource, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Edward J. Graham:** Software, Data curation, Investigation. **Amparo Galindo:** Writing - review & editing, Visualization, Supervision. **George Jackson:** Writing - review & editing, Supervision. **Claire S. Adjiman:** Conceptualization, Visualization, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgements

The authors gratefully acknowledge financial support from the Engineering and Physical Sciences Research Council ([EPSRC](#)) of the UK (grant [EP/J003840](#)), European Union's [Horizon 2020](#) research and innovation program under the grant agreement [727503](#) - ROLINCAP and Centre for Process Systems Engineering Research Committee of Imperial College London via Roger Sargent scholarship.

Data statement: Data underlying this article can be accessed on Zenodo at doi.org/10.5281/zenodo.3568383, and used under the Creative Commons Attribution license.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.compchemeng.2020.106802](#).

References

- Aarts, E., Laarhoven van, P., 1985. Statistical cooling: a general approach to combinatorial optimization problems. *Philips J. Res.* 40 (4), 193–226.
- Ahmad, M.Z., Hashim, H., Mustaffa, A.A., Maarof, H., Yunus, N.A., 2018. Design of energy efficient reactive solvents for post combustion CO₂ capture using computer aided approach. *J. Cleaner Prod.* 176, 704–715. doi:[10.1016/j.jclepro.2017.11.254](#).
- Austin, N.D., Sahinidis, N.V., Konstantinov, I.A., Trahan, D.W., 2018. Cosmo-based computer-aided molecular/mixture design: a focus on reaction solvents. *AIChE J.* 64 (1), 104–122. doi:[10.1002/aic.15871](#).
- Bardow, A., Steur, K., Gross, J., 2010. Continuous-molecular targeting for integrated solvent and process design. *Ind. Eng. Chem. Res.* 49 (6), 2834–2840. doi:[10.1021/ie901281w](#).
- Bokrantz, R., Forsgren, A., 2011. A dual algorithm for approximating Pareto sets in convex multi-criteria optimization. *Technology* 60 (August), 68. doi:[10.1118/1.3481566](#).
- Bortz, M., Burger, J., Asprion, N., Blagov, S., Böttcher, R., Nowak, U., Scheithauer, A., Welke, R., Küfer, K.H., Hasse, H., 2014. Multi-criteria optimization in chemical process design and decision support by navigation on Pareto sets. *Comput. Chem. Eng.* 60, 354–363. doi:[10.1016/j.compchemeng.2013.09.015](#).
- Bowskill, D.H., Tropp, U.E., Gopinath, S., Jackson, G., Galindo, A., Adjiman, C.S., 2020. Beyond a heuristic analysis: integration of process and working-fluid design for organic rankine cycles. *Mol. Syst. Des. Eng.* 5, 493–510. doi:[10.1039/C9ME00089E](#).
- Burger, J., Papaioannou, V., Gopinath, S., Jackson, G., Galindo, A., Adjiman, C.S., 2015. A hierarchical method to integrated solvent and process design of physical CO₂ absorption using the SAFT- γ mie approach. *AIChE J.* 61 (10), 3249–3269. doi:[10.1002/aic.14838](#).
- Buxton, A., Livingston, A.G., Pistikopoulos, E.N., 1999. Optimal design of solvent blends for environmental impact minimization. *AIChE J.* 45 (4), 817–843. doi:[10.1002/aic.690450415](#).
- Cignitti, S., Andreassen, J.G., Haglind, F., Woodley, J.M., Abildskov, J., 2017. Integrated working fluid-thermodynamic cycle design of organic Rankine cycle power systems for waste heat recovery. *Appl. Energy* 203, 442–453. doi:[10.1016/j.apenergy.2017.06.031](#).
- Constantinou, L., Gani, R., 1994. New group contribution method for estimating properties of pure compounds. *AIChE J.* 40 (10), 1697–1710. doi:[10.1002/aic.690401011](#).
- Das, I., Dennis, J.E., 1997. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct. Optim.* 14 (1), 63–69.
- Deb, K., 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. John Wiley & Sons.
- Deb, K., Pratab, A., Moitra, S., 2000. Mechanical component design for multiple objectives using elitist non-dominated sorting GA. In: *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 859–868. doi:[10.1007/3-540-45356-3_84](#).
- Dörgö, G., Abonyi, J., 2016. Group contribution method-based multi-objective evolutionary molecular design. *Hung. J. Ind. Chem.* 44 (1), 39–50. doi:[10.1515/hjic-2016-0005](#).
- Dörgö, G., Abonyi, J., 2019. Hierarchical representation based constrained multi-objective evolutionary optimisation of molecular structures. *Period. Polytech. Chem.* Eng. 63 (1), 210–225. doi:[10.3311/PPch.12021](#).
- Duveedi, A.P., Achenie, L.E., et al., 1996. Designing environmentally safe refrigerants using mathematical programming. *Chem. Eng. Sci.* 51 (15), 3727–3739. doi:[10.1016/0009-2509\(96\)00224-2](#).
- Eden, M.R., Jørgensen, S.B., Gani, R., El-Halwagi, M.M., 2004. A novel framework for simultaneous separation process and product design. *Chem. Eng. Process.* 43 (5), 595–608. doi:[10.1016/j.cep.2003.03.002](#).
- Folić, M., Adjiman, C.S., Pistikopoulos, E.N., 2008. Computer-aided solvent design for reactions: maximizing product formation. *Ind. Eng. Chem. Res.* 47 (15), 5190–5202. doi:[10.1021/ie0714549](#).
- Gani, R., Jimenez-Gonzalez, C., ten Kate, A., Crafts, P.A., Jones, M., Powell, L., Atherton, J.H., Cordiner, J.L., 2006. A modern approach to solvent selection: although chemists' and engineers' intuition is still important, powerful tools are becoming available to reduce the effort needed to select the right solvent. *Chem. Eng.* 113 (3), 30–44.
- Gopinath, S., Jackson, G., Galindo, A., Adjiman, C.S., 2016. Outer approximation algorithm with physical domain reduction for computer-aided molecular and separation process design. *AIChE J.* 62 (9), 3484–3504. doi:[10.1002/aic.15411](#).
- Haimes, Y.Y., Lasdon, L.S., WISMER, D.A., 1971. On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Trans. Syst. Man Cybern.* (3) 296–297. doi:[10.1109/TSMC.1971.4308298](#).
- Hostrup, M., Harper, P.M., Gani, R., 1999. Design of environmentally benign processes: integration of solvent design and separation process synthesis. *Comput. Chem. Eng.* 23 (10), 1395–1414. doi:[10.1016/S0098-1354\(99\)00300-2](#).
- Hsu, H.-C., Sheu, Y.-W., Tu, C.-H., 2002. Viscosity estimation at low temperatures ($T_r < 0.75$) for organic liquids from group contributions. *Chem. Eng. J.* 88, 27–35. doi:[10.1016/S1385-8947\(01\)00249-2](#).
- Hugo, A., Ciamei, C., Buxton, A., Pistikopoulos, E.N., 2004. Environmental impact minimization through material substitution: a multi-objective optimization approach. *Green Chem.* 6 (8), 407–417. doi:[10.1039/b401868k](#).
- Hukkerikar, A.S., Sarup, B., Ten Kate, A., Abildskov, J., Sin, G., Gani, R., 2012. Group-contribution + (GC+) based estimation of properties of pure components: Improved property estimation and uncertainty analysis. *Fluid Phase Equilibil.* 321, 25–43. doi:[10.1016/j.fluid.2012.02.010](#).
- Joback, K.G., Reid, R.C., 1987. Estimation of pure-component properties from group-contributions. *Chem. Eng. Commun.* 57 (1–6), 233–243. doi:[10.1080/00986448708960487](#).
- Khor, S.Y., Liam, K.Y., Loh, W.X., Tan, C.Y., Ng, L.Y., Hassim, M.H., Ng, D.K., Chemmangattuvalappil, N.G., 2017. Computer aided molecular design for alternative sustainable solvent to extract oil from palm pressed fibre. *Process Saf. Environ. Prot.* 106, 211–223. doi:[10.1016/j.pses.2016.12.023](#).
- Kim, K.-J., Diwekar, U.M., 2002. Integrated solvent selection and recycling for continuous processes. *Ind. Eng. Chem. Res.* 41 (18), 4479–4488. doi:[10.1021/ie010777g](#).
- Kucherenko, S., Sytsko, Y., 2005. Application of deterministic low-discrepancy sequences in global optimization. *Comput. Optim. Appl.* 30 (3), 297–318. doi:[10.1007/s10589-005-4615-1](#).
- Liang, T.-F., 2008. Fuzzy multi-objective production/distribution planning decisions with multi-product and multi-time period in a supply chain. *Comput. Ind. Eng.* 55 (3), 676–694. doi:[10.1016/j.cie.2008.02.008](#).
- Marcoulaki, E., Kokossis, A., 2000. On the development of novel chemicals using a systematic optimisation approach. part ii. solvent design. *Chem. Eng. Sci.* 55 (13), 2547–2561. doi:[10.1016/S0009-2509\(99\)00523-0](#).
- Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* 26 (6), 369–395. doi:[10.1007/s00158-003-0368-6](#).
- Mavrotas, G., 2009. Effective implementation of the ε -constraint method in multi-objective mathematical programming problems. *Appl. Math. Comput.* 213 (2), 455–465. doi:[10.1016/j.amc.2009.03.037](#).

- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21 (6), 1087–1092. doi:[10.1063/1.1699114](https://doi.org/10.1063/1.1699114).
- Neumaier, A., 2004. Complete search in continuous global optimization and constraint satisfaction. *Acta Numer.* 13, 271–369. doi:[10.1017/S0962492904000194](https://doi.org/10.1017/S0962492904000194).
- Ng, L.Y., Chemmangattuvalappil, N.G., Ng, D.K., 2014. A multiobjective optimization-based approach for optimal chemical product design. *Ind. Eng. Chem. Res.* 53 (44), 17429–17444. doi:[10.1021/ie502906a](https://doi.org/10.1021/ie502906a).
- Ng, L.Y., Chong, F.K., Chemmangattuvalappil, N.G., 2015. Challenges and opportunities in computer-aided molecular design. *Comput. Chem. Eng.* 81, 115–129. doi:[10.1016/j.compchemeng.2015.03.009](https://doi.org/10.1016/j.compchemeng.2015.03.009).
- Odele, O., Macchietto, S., 1993. Computer aided molecular design: a novel method for optimal solvent selection. *Fluid Phase Equilib.* 82, 47–54. doi:[10.1016/0378-3812\(93\)87127-M](https://doi.org/10.1016/0378-3812(93)87127-M).
- Pál, L., 2013. Comparison of multistart global optimization algorithms on the BBOB noiseless testbed. In: Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation. ACM, pp. 1153–1160. doi:[10.1145/2464576.2482693](https://doi.org/10.1145/2464576.2482693).
- Palma-Flores, O., Flores-Tlacuahuc, A., Canseco-Melchor, G., 2015. Optimal molecular design of working fluids for sustainable low-temperature energy recovery. *Comput. Chem. Eng.* 72, 334–349. doi:[10.1016/j.compchemeng.2014.04.009](https://doi.org/10.1016/j.compchemeng.2014.04.009).
- Papadopoulos, A.I., Badr, S., Chremos, A., Forte, E., Seferlis, P., Papadokonstantakis, S., Galindo, A., Jackson, G., Adjiman, C.S., 2016. Computer-aided molecular design and selection of CO₂ capture solvents based on thermodynamics, reactivity and sustainability. *Mol. Syst. Des. Eng.* 1 (3), 313–334. doi:[10.1039/C6ME00049E](https://doi.org/10.1039/C6ME00049E).
- Papadopoulos, A.I., Linke, P., 2006. Multiobjective molecular design for integrated process-solvent systems synthesis. *AIChE J.* 52 (3), 1057–1070. doi:[10.1002/aic.10715](https://doi.org/10.1002/aic.10715).
- Papadopoulos, I.A., Shavalieva, G., Papadokonstantakis, S., Seferlis, P., Perdomo, A.F., Galindo, A., Jackson, G., Adjiman, C.S., 2020. An Approach for Simultaneous Computer-Aided Molecular Design with Holistic Sustainability Assessment: Application to Phase-change CO₂ Capture Solvents. *Comput. Chem. Eng.* 135, 10679. doi:[10.1016/j.compchemeng.2020.106769](https://doi.org/10.1016/j.compchemeng.2020.106769).
- Papadopoulos, A.I., Stijepovic, M., Linke, P., 2010. On the systematic design and selection of optimal working fluids for organic rankine cycles. *Appl. Therm. Eng.* 30 (6–7), 760–769. doi:[10.1016/j.applthermaleng.2009.12.006](https://doi.org/10.1016/j.applthermaleng.2009.12.006).
- Papadopoulos, A.I., Stijepovic, M., Linke, P., Seferlis, P., Voutetakis, S., 2013. Toward optimum working fluid mixtures for organic rankine cycles using molecular design and sensitivity analysis. *Ind. Eng. Chem. Res.* 52 (34), 12116–12133. doi:[10.1021/ie400968j](https://doi.org/10.1021/ie400968j).
- Papaioannou, V., Lafitte, T., Avendaño, C., Adjiman, C.S., Jackson, G., Müller, E.A., Galindo, A., 2014. Group contribution methodology based on the statistical associating fluid theory for heteronuclear molecules formed from Mie segments. *J. Chem. Phys.* 140 (5). doi:[10.1063/1.4851455](https://doi.org/10.1063/1.4851455).
- Pereira, F.E., Keskes, E., Galindo, A., Jackson, G., Adjiman, C.S., 2011. Integrated solvent and process design using a SAFT-VR thermodynamic description: high-pressure separation of carbon dioxide and methane. *Comput. Chem. Eng.* 35 (3), 474–491. doi:[10.1016/j.compchemeng.2010.06.016](https://doi.org/10.1016/j.compchemeng.2010.06.016).
- Pistikopoulos, E., Stefanis, S., 1998. Optimal solvent design for environmental impact minimization. *Comput. Chem. Eng.* 22 (6), 717–733.
- Poling, B.E., Prausnitz, J.M., O'Connell, J.P., et al., 2001. *The Properties of Gases and Liquids*, vol. 5. McGraw-Hill New York.
- Pretel, E.J., López, P.A., Bottini, S.B., Brignole, E.A., 1994. Computer-aided molecular design of solvents for separation processes. *AIChE J.* 40 (8), 1349–1360. doi:[10.1002/aic.690400808](https://doi.org/10.1002/aic.690400808).
- Rennen, G., Van Dam, E.R., Den Hertog, D., 2011. Enhancement of sandwich algorithms for approximating higher-dimensional convex Pareto sets. *INFORMS J. Comput.* 23 (4), 493–517. doi:[10.1287/ijoc.1100.0419](https://doi.org/10.1287/ijoc.1100.0419).
- Rinnooy, K., Timmer, A.H.G., 1987. Stochastic global optimization methods part i: clustering methods. *Math. Program.* 39, 27–56. doi:[10.1007/BF02592070](https://doi.org/10.1007/BF02592070).
- Sahinidis, N.V., Tawarmalani, M., 2000. Applications of global optimization to process and molecular design. *Comput. Chem. Eng.* 24 (9–10), 2157–2169. doi:[10.1016/S0098-1354\(00\)00583-4](https://doi.org/10.1016/S0098-1354(00)00583-4).
- Sastri, S., Rao, K., 1999. A new temperature–thermal conductivity relationship for predicting saturated liquid thermal conductivity. *Chem. Eng. J.* 74 (3), 161–169. doi:[10.1016/S1385-8947\(99\)00046-7](https://doi.org/10.1016/S1385-8947(99)00046-7).
- Schilling, J., Tillmanns, D., Lampe, M., Hopp, M., Gross, J., Bardow, A., 2017. From molecules to dollars: integrating molecular design into thermo-economic process design using consistent thermodynamic modeling. *Mol. Syst. Des. Eng.* 2, 301–320. doi:[10.1039/C7ME00026J](https://doi.org/10.1039/C7ME00026J).
- Sobol', I.M., Asotsky, D., Kreinin, A., Kucherenko, S., 2011. Construction and comparison of high-dimensional Sobol' generators. *Wilmott* 2011 (56), 64–79. doi:[10.1002/wilm.10056](https://doi.org/10.1002/wilm.10056).
- Solanki, R.S., Appino, P.A., Cohon, J.L., 1993. Approximating the noninferior set in multiobjective linear programming problems. *Eur. J. Oper. Res.* 68 (3), 356–373. doi:[10.1016/0377-2217\(93\)90192-P](https://doi.org/10.1016/0377-2217(93)90192-P).
- Song, J., Song, H.-H., 2008. Computer-aided molecular design of environmentally friendly solvents for separation processes. *Chem. Eng. Technol.* 31 (2), 177–187. doi:[10.1002/ceat.200700233](https://doi.org/10.1002/ceat.200700233).
- Venkatasubramanian, V., Chan, K., Caruthers, J., 1994. Computer-aided molecular design using genetic algorithms. *Comput. Chem. Eng.* 18 (9), 833–844. doi:[10.1016/0098-1354\(93\)E0023-3](https://doi.org/10.1016/0098-1354(93)E0023-3).
- Viswanathan, J., Grossmann, I.E., 1990. A combined penalty function and outer-approximation method for MINLP optimization. *Comput. Chem. Eng.* 14 (7), 769–782. doi:[10.1016/0098-1354\(90\)87085-4](https://doi.org/10.1016/0098-1354(90)87085-4).
- Xu, W., Diwekar, U.M., 2007. Multi-objective integrated solvent selection and solvent recycling under uncertainty using a new genetic algorithm. *Int. J. Environ. Pollut.* 29 (1/3), 70. doi:[10.1504/IJEP.2007.012797](https://doi.org/10.1504/IJEP.2007.012797).
- Zhang, J., Qin, L., Peng, D., Zhou, T., Cheng, H., Chen, L., Qi, Z., 2017. Cosmo-descriptor based computer-aided ionic liquid design for separation processes: part ii: task-specific design for extraction processes. *Chem. Eng. Sci.* 162, 364–374. doi:[10.1016/j.ces.2016.12.023](https://doi.org/10.1016/j.ces.2016.12.023).
- Zhang, L., Cignitti, S., Gani, R., 2015. Generic mathematical programming formulation and solution for computer-aided molecular design. *Comput. Chem. Eng.* 78, 79–84. doi:[10.1016/j.compchemeng.2015.04.022](https://doi.org/10.1016/j.compchemeng.2015.04.022).
- Zhou, T., Song, Z., Zhang, X., Gani, R., Sundmacher, K., 2019. Optimal solvent design for extractive distillation processes: a multiobjective optimization-based hierarchical framework. *Ind. Eng. Chem. Res.* 58 (15), 5777–5786. doi:[10.1021/acs.iecr.8b04245](https://doi.org/10.1021/acs.iecr.8b04245).
- Zhou, T., Zhou, Y., Sundmacher, K., 2017. A hybrid stochastic-deterministic optimization approach for integrated solvent and process design. *Chem. Eng. Sci.* 159, 207–216. doi:[10.1016/j.ces.2016.03.011](https://doi.org/10.1016/j.ces.2016.03.011).
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G., 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* 7 (2), 117–132. doi:[10.1109/TEVC.2003.810758](https://doi.org/10.1109/TEVC.2003.810758).