Applied Soft Computing 47 (2016) 483–493

Contents lists available at ScienceDirect

Applied Soft Computing

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

Incorporating domain knowledge into the optimization of energy systems

Md Shahriar Mahbub[a,b,*], Markus Wagner[c], Luigi Crema[a]

[a] Fondazione Bruno Kessler, Via Sommarive 18, 38123 Povo, Trento, Italy
[b] University of Trento, Via Sommarive 9, 38123 Povo, Trento, Italy
[c] University of Adelaide, Adelaide, SA 5005, Australia

ARTICLE INFO

Article history:
Received 25 August 2015
Received in revised form 17 May 2016
Accepted 13 June 2016
Available online 17 June 2016

Keywords:
Domain knowledge
Energy system optimization
Initialization
Multi-objective evolutionary algorithm

ABSTRACT

Energy plays a key factor in the advancement of humanity. As energy demands are mostly met by fossil fuels, the world-wide consciousness grows about their negative impact on the environment. Therefore, it becomes necessary to design sustainable energy systems by introducing renewable energies. Because of the intermittent availability of different renewable resources, the designing of a sustainable energy system should find an optimal mix of different resources. However, the optimization of this combination has to deal with a number of possibly contradictory objectives.

Multi-objective evolutionary algorithms (MOEA) are widely used to solve this kind of problems. As optimizing an energy system by using a MOEA is computationally costly, it is necessary to solve the problem efficiently. For this purpose, we propose the incorporation of domain knowledge related to energy systems into different phases (i.e., initialization and mutation) of a MOEA run. The proposed approaches are implemented for two widely used MOEAs and evaluated on the Danish Aalborg test problem. The experimental results show that each approach individually achieves significant improvements of the energy systems, which is expressed in better trade-off sets. Moreover, a state-of-the-art stopping criterion is adapted to detect the convergence in order to save computational resources. Finally, all proposed techniques are merged within two MOEAs with the result that our combined approaches yield significantly better results in less time than generic approaches.

1. Introduction

Most energy generated all over the world is based on fossil fuels [1]. As energy generated by fossil fuels has harmful effects on the environment, recent interest is directed towards the employment of green or renewable sources to generate energy [2]. However, due to their intermittent availability, it is not easy to integrate renewable energy into a larger energy system [3].

In order to solve the integration problem, two optimization phases can be considered [4], i.e., (i) operational optimization and (ii) capacity/sizing optimization. While the day-to-day operations of resources of a given energy system are optimized in the first phase, the second phase is mainly concerned with the design of future energy scenarios to integrate renewable energies. For the first phase, many optimization models such as energy system simulation models are available (e.g., see the comprehensive review article by Connolly et al. [5]). For the second phase, however, only few attempts have been made when the considered energy system consists of inter-connected sub-systems from electric, thermal and transportation sectors [6]. Of the existing approaches, several consider the optimization of the second phase as a multi-objective optimization problem [7–9], where the different objectives can be *total cost, unmet load* and *fuel emission* (i.e., CO_2 emission) [8]. In this article, we focus on the optimization of energy systems in the second phase. Capacity/sizing optimization is an active research topic in the energy domain, where it is possible to leverage synergies between different energy sub-systems [10].

The objectives of real-world problems can often be in conflict with each other. The goal of solving a multi-objective optimization (MOO) problem is to find a (not too large) set of compromise solutions. The Pareto front of a MOO problem consists of the function values representing the different trade-offs with respect to the given objective functions. In practice, it is computationally infeasible to compute the whole Pareto front, and MOO problems

[*] Corresponding author at: Fondazione Bruno Kessler, Via Sommarive 18, 38123 Povo, Trento, Italy.
E-mail addresses: mahbub@fbk.eu (M.S. Mahbub), markus.wagner@adelaide.edu.au (M. Wagner), crema@fbk.eu (L. Crema).

http://dx.doi.org/10.1016/j.asoc.2016.06.013
1568-4946/© 2016 Elsevier B.V. All rights reserved.

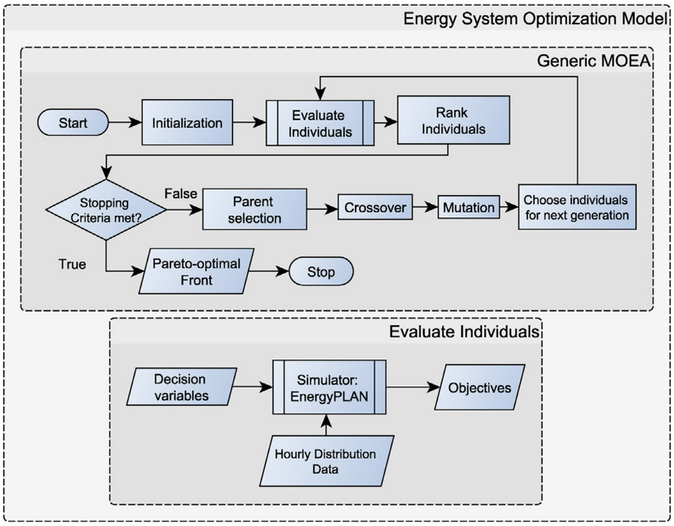

Fig. 1. Energy system optimization model.

often can only be solved approximately by heuristic approaches. Evolutionary algorithms have been widely used to tackle multi-objective problems, and recently, efforts have been made to employ multi-objective evolutionary algorithm (MOEA) to solve the problem of optimizing energy systems [11,12]. In addition, as the energy system optimization problem we want to tackle is non-linear and discontinuous in nature [10], we apply stochastic method such as evolutionary algorithm instead of gradient-based methods (such as used in [13]).

Fig. 1 presents our model for optimizing energy systems. The upper part of the model illustrates the steps undertaken by classical MOEAs such as Non-dominated Sorting Genetic Algorithm (NSGA-II) [14] and Strength Pareto Evolutionary Algorithm (SPEA2) [15]. MOEAs are bio-inspired algorithms, which mimic some fundamental aspects of the neo-Darwinian evolutionary process. They simultaneously search with a *population* of candidate solutions and associate objective scores as fitness values for each candidate solution. The algorithms then select among the population to favor those solutions that are more fit. The next generation (i.e., a new population) consists of replicates of the fitter solutions that have been *genetically mutated* and *crossed over* in a biological metaphor: the decision variables were perturbed such that they inherit characters of their parents, as well as change in random ways. NSGA-II and SPEA2 are nearly identical, but differ in the way they rank solutions within the set of trade-offs, and in the way the individuals for next generations are selected.

The lower part of Fig. 1 shows our steps of evaluating individuals. The core component is the simulator for energy systems, and Connolly et al. [5] provide a detailed review of different computer tools for performing such simulations. The classification of tools mainly depends on the simulated time step and the modeled energy sub-systems. Time steps are important when modeling the intermittent availability of renewable resources. There are several simulation models that consider different time step sizes: HOMER [16,17] for minutes, EnergyPLAN and H_2RES [18,19] for hours, and INFORSE [20] and LEAP [21] for years. However, very few models (e.g., EnergyPLAN, INFORSE) are capable of simulating different levels of penetration of a renewable energy system (electricity, thermal and transportation). We choose EnergyPLAN because it is freely available and it provides fairly detailed (i.e., hourly) analyses of operations of different energy generating sources. It is capable of simulating all the main inter-connected sub-systems of an energy system. In addition, EnergyPLAN also optimizes the operations of a given system (i.e., capacities of different power generations components, demands, efficiencies, and other relevant data) and produces annual indicators (e.g., total emission, primary energy consumption, cost and others).

Generally, simulation models are computationally costly, therefore, we want to optimize energy systems more efficiently. In particular, we want to achieve this by generating a high-quality approximation of the Pareto front [22] at reduced computational cost. To reach this goal, we investigate the incorporation of domain knowledge related to energy systems into the different phases of a MOEA. Firstly, we propose a smart initialization technique and secondly, incorporate a smart mutation [23]; both exploit domain knowledge. Additionally, to detect convergence of the algorithm, we apply the stopping criterion proposed by Mahbub et al. [24] that has proven to work reliably when used in the optimization of a real-world problem (i.e., energy system optimization problem). This way the MOEA stops when no improvements are achieved, which saves computational resources that would otherwise be wasted. We integrate smart initialization, mutation and stopping criterion into MOEAs to form *informed MOEAs* and compare them with generic MOEAs. The results clearly show that all these individual methods work together and have an overall very good impact on the optimization of an energy system. To the best of our knowledge, this is the first attempt to incorporate energy system domain knowledge into different operators of MOEAs.

In this study, we focus on the Danish Aalborg energy system [25] to demonstrate the feasibility of our approach. It is a well-understood problem, and the details are readily available. It is important to note that more and more aspects have been investigated in the recent past, giving rise to a large number of optimization problems about renewable energy management and electricity market operation. For instance, energy bidding and reservation [26], economic dispatch [27] and microgrids management [28] have been considered in the last few years. As our proposed improvements are independent of the particular framework used (i.e., as the approach is generic), we conjecture that they can be applied to these problems as well to improve the outcomes.

The paper is organized as follows. Most of Section 2 discusses how domain knowledge is represented and how it can be incorporated into a MOEA through problem-specific initialization. A brief description of smart mutation and stopping criterion is presented in Sections 2.3 and 3. We present our test problem in Section 4. Then, we describe in Section 5 the details of all experiments and the corresponding discussions of the results. Finally, we draw our conclusions in Section 6.

2. Incorporating domain knowledge

In general, a typical MOEA cannot perform well for all classes of problems, as this would be contradictory to the No Free Lunch Theorem [29]. According to this theorem, the average performance of an algorithm over all possible classes of problems is constant. Hence, the good performance of an optimization algorithm on one class of problems is balanced out by the bad performance of the algorithm on another class of problems. However, this also means that problem-specific algorithms with above-average performance are possible. Bonissone et al. [29] define two different ways to achieve this by incorporating domain knowledge: *implicitly* and *explicitly*. Encoding, design of data structures and constraints representation are categorized as an implicit incorporation of domain knowledge. Our article mainly focuses on the explicit incorporation (i.e., smart seeding of initial population, mutation exploiting domain knowledge) for the energy system optimization problem. In the following sections, we will discuss how we represent domain knowledge of energy systems and how we incorporate this knowledge into initialization and mutation.

Table 1
A real-life domain-knowledge representation example.

DVs DK$_{\text{Obj}}$	$W_{C_{\text{off}}}$	$W_{C_{\text{on}}}$	PV_C	PP_C	C_S	O_S	NG_S
$DK_{O_{\text{EM}}}$	True	True	True	Null	False	Null	True
$DK_{O_{\text{AC}}}$	False	False	False	Null	True	Null	False

2.1. Domain knowledge related to energy systems

Typically, experts of a field can provide detailed domain knowledge about the field, and laypeople can at times provide very basic knowledge or “rules of thumb”. In the following, we will use such basic knowledge.

Our goal in this research is to minimize *CO_2 emissions* and the *total annual cost* of a system. In this context, it is obvious that some decision variables have influence on some objectives. For example, increasing the capacities of renewable resources can reduce CO_2 emissions. At the same time, decreasing the capacities of renewable resources can minimize the annual cost. To encode such basic knowledge, we will use the following method: we mark each decision variable with *true*, *false* or *null* for each objective. *True* and *false* indicate *increasing* and *decreasing*, respectively, of the value of a decision variable to minimize an objective. *Null* indicates that there is no domain knowledge available for the decision variable for the objective.

A real-life example of DK representation in the context of energy systems is given in Table 1. $W_{C_{\text{off}}}$ and $W_{C_{\text{on}}}$ represent off-shore and on-shore wind power capacities, respectively. PV_C is the photovoltaic's and PP_C is the power plant's capacity. C_S, O_S and NG_S represent the coal, oil and natural gas shares, respectively, to fire power plants. Finally, $DK_{O_{EM}}$ and $DK_{O_{AC}}$ are the domain knowledge associated with the minimization of emission and annual cost, respectively. The first row of the table can be interpreted as increasing (i.e., off-shore, on-share wind, PV capacities; natural gas share to fuel power plants) and decreasing (i.e., coal share to fuel power plants) of some decision variables that could minimize the objective emissions. The second row can be interpreted in similar fashion.

2.2. Smart initialization

Most experimental studies on MOEAs use random initialization to initialize the first population. There, the initial values of the decision variables are drawn from a uniform distribution within the lower and upper bounds of the variables. In practice, however, optimizers can typically be seeded with good candidate solutions either previously known or created according to some problem-specific method. This seeding has been studied extensively for single-objective problems. For multi-objective problems, however, very little literature is available on the approaches to seeding and their individual benefits and disadvantages.

Friedrich et al. [30] provide a recent overview on MOEA seeding, and also a comprehensive study on 48 artificial MOO problems for five different MOO algorithms. Their generic seeding approaches are based on linear combinations of the objectives, and the individual seeds are computed by a single-objective optimizer that solves a particular linear combination. These linear combinations are quite “evenly spread out” in order to achieve an unbiased but close initial population. They observe that some problems benefit significantly from their seeding strategies, while others profit less. The advantage of seeding also depends on the examined algorithm.

In contrast to their problem-independent seeding approaches, our method initializes a decision variable by using actual domain knowledge. We enable laypersons to provide some domain knowledge in a very basic manner. For example, “decreasing coal share could minimize the objective emissions” is encoded as “false” in Table 1. Our method in the following section then translates this basic knowledge into a diverse set of seeds. We call this process *smart initialization*, and we describe the details in the following.

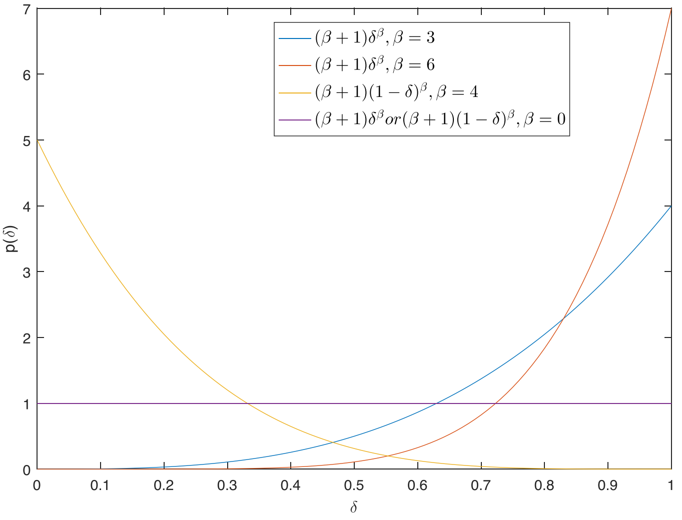

Fig. 2. Example of probability distributions. (For interpretation of the references to color in text near the reference citation, the reader is referred to the web version of this article.)

2.2.1. Methodology

The initial value of a decision variable can be defined as follows:

$$dv_i = lb_i + (ub_i - lb_i) * \bar{\delta} \tag{1}$$

where lb_i and ub_i are the lower and upper bounds of the decision variable, respectively. We calculate $\bar{\delta}$ from the probability distributions described below.

Probability distribution: The following proposed probability distribution is used when a decision variable is needed to be initialized with higher values:

$$p(\delta) = (\beta + 1)\delta^\beta \tag{2}$$

and the following probability distribution is used when a decision variable is needed to be initialized with lower values:

$$p(\delta) = (\beta + 1)(1 - \delta)^\beta \tag{3}$$

These distributions are valid for $\delta \in (0, 1)$. β takes a non-negative value that is used to control the shape of the distribution.

Note that a uniform distribution cannot be used to incorporate any domain knowledge. While Gaussian distributions can be used, it is not as easy as with ours to incorporate domain knowledge. We will see in the following how we can control the shape of the distribution in order to initialize values closer to either the lower or upper bounds of the decision variables.

Fig. 2 illustrates four examples with two different distributions and different β values. The blue and red curves are the plots of Eq. (2), however, the red curve is steeper than the blue curve due to the larger β value. Hence, with larger β value, it is more probable that the initial value will be closer to the upper bound of the decision variable. For lower β values, the distribution will be flatter, hence, increasing the probability of an initial value to fall within the entire range between the upper and lower bounds, rather than near the upper bound. Eventually, when $\beta = 0$ the distribution will be equivalent to the *uniform distribution* (i.e., purple line in Fig. 2) that is commonly used for random initializations. Additionally, the orange curve is an example of a distribution based on Eq. (3). In summary, by controlling the β value and by using the appropriate distribution, one can have a better control over the initial value of a decision variable than with random initialization. Note that, while we could manually set exact values for the initial values, domain knowledge is often a bit "fuzzy", and our distributions allow for random but biased variations of the initial values.

Now, the analytical formula for $\bar{\delta}$ (used in Eq. (1)) can be found by calculating the inverse of the cumulative distribution function [31] (also known as inversion method [32]).[1]

$$\int_0^{\bar{\delta}} p(\delta)d\delta = t \tag{4}$$

where t is a random number within [0, 1] drawn from a uniform distribution. When Eq. (2) is used as probability distribution, the analytical formula for $\bar{\delta}$ is

$$\bar{\delta} = t^{1/(\beta+1)}. \tag{5}$$

When lower values are preferred (Eq. (3)), the formula is

$$\bar{\delta} = 1 - (1-t)^{1/(\beta+1)} \tag{6}$$

2.2.2. Generating initial populations

As it is not possible to know in advance which combinations of β values for the decision variables of an individual will be suitable to generate the initial biased populations, we will consider all possible combinations. If the problem at hand has d decision variables and if we consider b different values for β, then the total number of combinations is $n_c = b^d$. Therefore, for each of the o objectives, there are n_c combinations of β values, and for each combination we can generate k individuals to achieve some diversity among the biased solutions. Hence, we generate $n_{DK} = o * n_c * k$ individuals in total.

Most evolutionary algorithms, however, only work with populations that are significantly smaller than n_{DK} for real-world problems. For example, with $o = 3$, $b = 3$, $d = 5$, and $k = 10$ the initial population size would be $n_{DK} = 7290$, which is about two orders of magnitude above what is typically used in studies. Populations of this size tend to slow down the actual optimization algorithm as the computational complexity of the algorithm is typically dependent on the population size. To solve this problem, we provide a procedure for reducing the number of individuals down to a fixed number. Algorithm 1 presents our overall process of generating initial populations that uses domain knowledge.

Algorithm 1. Algorithm for generating initial populations.

```
for all domain-knowledge arrays $DK_{o_i}$ do
  for all combinations ($c$) from $n_c$ combinations do
    Generate $k$ individuals using Algorithm 2
  end for
end for
Reduce number of individuals from $n_{DK}$ to $n_p$
```

Algorithm 2. Algorithm for generating a single individual.

```
Require: a combination $c$ of $d$ numbers for $\beta$ values
for all $dv_j$ do
  if $DK_{o^i}^{dv_j}$ is true then
    Use Eqs. (5) and (1) to generate the value of the decision variable; value of $\beta$ is taken from the combination $c$
  else if$DK_{o^i}^{dv_j}$ is false then
    Use Eqs. (6) and (1) to generate the value of the decision variable; value of $\beta$ is taken from the combination $c$
  else if $DK_{o^i}^{dv_j}$ is null then
    Generate a random value within upper and lower bound from uniform distribution (i.e., use Eqs. (5) or (6) with $\beta = 0$)
  end if
end for
```

Reducing the number of individuals: Due to combinatorial explosion, our previously proposed methodology generates many individuals. As we want to reduce the number of individuals, we adapt the concept of *decision space diversity* to cover the space efficiently using a fixed number n_p of individuals. Ulrich et al. [33] categorizes diversity measurements depending on the way they are calculated. The categorization is based on (i) relative abundances, (ii) taxonomy, (iii) aggregating the dissimilarities, and (iv) utility of solutions. In the first class (relative abundance), diversity is calculated by measuring the relative abundance of each solution within a population set, with one example being the Shannon entropy [34] metric. The metrics in the second class use the path length within a taxonomy tree, where the solutions are arranged in a tree that reflects the taxonomic classification of the solutions. Clustering metrics belong to this class, and the calculation of metrics from this class typically suffers from high computational cost. The metrics of the third kind are computed by summing up all the dissimilarities between all the individuals. For example, Shir et al. [35] use a metric from this class to enhance diversity in a MOEA. The last class is based on measuring the utility of solutions. Solow et al. [36] propose a metric that uses a utilitarian view on solutions. There, each solution has a pre-defined utility value, and the key idea is that the total utility of a population does not increase by adding duplicate individuals. Ulrich et al. [33] has shown that this Solow and Polasky metric (a metric of the forth class) is the only one to fulfill all the three basic requirements of a diversity measure (i.e., monotonicity in varieties, twinning and monotonicity in distance). In our article, we will use the Solow and Polasky metric to measure the diversity of an initial population.

Let us consider a population P with y individuals (A_1, A_2, ..., A_y), and let $d(A_i, A_j)$ be the Euclidean distance in the decision space between A_i and A_j (i.e., $d(A_i, A_j) = \sqrt{\Sigma_{l=1}^{z}(dv_{A_i}^{l} - dv_{A_j}^{l})^2}$, where $dv_{A_i}^{l}$, $dv_{A_j}^{l}$ are the l^{th} decision variable of A_i and A_j individuals, respectively and z is the number of decision variables). Then each element $m_{i,j}$ of a $y \times y$ matrix (M) can be defined as $m_{i,j} = exp(-\theta * d(A_i, A_j))$, where θ is a normalizing parameter between distance and number of individuals. Finally, the Solow and Polasky metric is the summation of all the elements of the M^{-1} matrix. Algebraically, it can be written as follows:

$$SP(P) = vM^{-1}v^T \tag{7}$$

where $v = (1, 1, \ldots, 1)$ is a row vector of size $1 \times y$ and v^T represents the transpose of v. Intuitively, the metric measures the number of different individuals present in the population. The individuals that are close to each other are considered as the same individuals, which can be adjusted via the value of θ.

The initial population P_{DK} from our previous approach contains $n_{DK} = o * n_c * k$ individuals. As we need to reduce the number of individuals from initial population, the idea is to select a subset P_I of size n_p from P_{DK} that maximizes the population's diversity. The problem is formulated as follows:

$$\underset{P_I \subseteq P_{DK}}{\arg\max}\ SP(P_I) \tag{8}$$

As it is not practical to consider all possible subsets, we use the greedy approach proposed in [33]. This approach iteratively removes the individual that contributes the least to the Solow and Polasky metric.

2.3. Smart mutation

While we expect the smarter initialization to give the optimization a "head start", we also want to speed up the actual optimization process itself. Mahbub et al. [23] developed the idea of *smart mutation*, which is based on the same concept of domain knowledge utilization as is our initialization strategy. In that article, the polynomial mutation as proposed by Deb [22] is modified to create two additional mutation operators called renewable energy favor

[1] A process for generating random variates from a specific probability distribution.

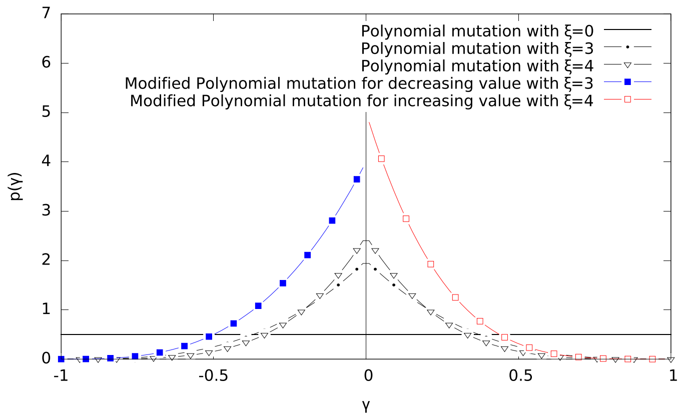

Fig. 3. Probability distributions for polynomial and modified polynomial mutation. (For interpretation of the references to color in text near the reference citation, the reader is referred to the web version of this article.)

mutation (REFM) and conventional energy favor mutation (CEFM). Polynomial mutation mutates an offspring so that the values of the decision variables can be increased or decreased depending on a randomly generated value. On the other hand, both REFM and CEFM use the same domain knowledge representation as described in Section 2.1. For example, the first and second row of Table 1 can be used for REEM and CEFM, respectively. In addition, REEM mutates the decision variables to be increased or decreased based on domain knowledge represented by the first row of Table 1 and in the same way, the second row can be interpreted for CEFM. In the following, we will briefly outline the details of smart mutation.

In polynomial mutation, a probability distribution is used to perturb a decision variable of an offspring solution.

$$\bar{\mu} = \mu + (\mu^U - \mu^L)\bar{\gamma} \tag{9}$$

where $\bar{\mu}$ is the mutated offspring decision variable, μ is the offspring decision variable that will be perturbed. μ^U and μ^L are the upper and lower bounds of μ, respectively. $\bar{\gamma}$ will be calculated from the following probability distribution.

$$p(\gamma) = 0.5(\xi + 1)(1 - |\gamma|)^{\xi} \tag{10}$$

where ξ is a parameter that takes only non-negative values, called distribution index. This distribution is valid for $\gamma \in (-1, 1)$. Three black lines of Fig. 3 illustrate the distributions with different distribution index values.

When it is required to increase the value of a decision variable (e.g., require for REFM and CEFM), a modified probability distribution function is used instead of Eq. (10).

$$p(\gamma) = 0.5(\xi + 1)(1 - \gamma)^{\xi} \tag{11}$$

The distribution is valid for $\gamma \in (0, 1)$. This distribution insures that the value of an offspring decision variable only can be increased or stay same as before. The black line with downward triangle in Fig. 3 illustrates the probability distribution used in a polynomial mutation with distribution index 4. Whereas, the red line with empty rectangles in the figure shows the modified distribution with same distribution index. The closed/analytical form of $\bar{\gamma}$ for the above distribution is as follows:

$$\bar{\gamma} = 1 - [1 - r + r(1 - \gamma^{\max})^{\xi+1}]^{1/(\xi+1)} \tag{12}$$

where r is a random number within a range of [0, 1], drawn from an uniform distribution. And $\gamma^{max} = \min\{(\mu - \mu^L), (\mu^U - \mu)\}/(\mu^U - \mu^L)$. This closed form of $\bar{\gamma}$ is used in Eq. (9) to calculate the new mutated value.

In the following, the modified probability distribution and the closed form of $\bar{\gamma}$ are presented, when the value of a decision variable of an offspring needs to be decreased:

$$p(\gamma) = 0.5(\xi + 1)(1 + \gamma)^{\xi}, \quad \text{valid for} \quad \gamma \in (-1, 0) \tag{13}$$

$$\bar{\gamma} = [r + (1 - r)(1 - \gamma^{\max})^{\xi+1}]^{1/(\xi+1)} - 1 \tag{14}$$

At the beginning of an optimization, the probabilities of the problem-specific mutations (i.e., REFM and CEFM) are maximal and these decrease over time. As generations pass, the probability of generic mutation (i.e., polynomial mutation) is increased. This dynamic adjustment of the mutation probabilities ensures that modified mutations are used more in the early stage to better explore the search space (generate some extreme individuals) and polynomial mutation is used more in the later stage to maintain the usual exploration and exploitation behavior of a generic mutation. Mahbub [23] demonstrated the successful application of this problem-specific mutation for an energy system optimization problem, which is why it forms an important component in our research as well.

3. Stopping criterion

We use EnergyPLAN (version 11.0, released in September 2013) [37] to assess solutions within a MOEA. As function evaluations using EnergyPLAN are in general computationally costly, we need to find ways to reduce the computational time. One way to minimize the computational cost is to stop a MOEA in the appropriate moment (i.e., when it is converged) to save costly function evaluations. Interestingly, this is in stark contrast to the usual MOEA use, where they are stopped after a certain amount of function evaluations.

Setting this particular number efficiently for real-world optimization problems (where no knowledge specific to the convergence is available) is a difficult task. Therefore, we adapt the stopping technique that was recently proposed by Mahbub et al. [24] in this case. The proposed method is based on simultaneously monitoring the population of the MOEA in the objective and decision space. The *average Hausdorff distance* and *diversity* are used to monitor the objective and decision spaces, respectively. To detect the stability of these two metrics, a statistical hypothesis test (i.e., two-sided *t*-test) is performed on the time-value slope of the metrics. Mahbub et al. [24] report that the convergence (i.e., absence of progress) of a MOEA is detected more reliably by this combination than the way when only objective space metrics are used.

Another advantage of the technique is the simplicity of setting this stopping criterion. The first parameter α specifies the significance level of the statistical test. The next parameter *nGenLT* is the number of previous generations, for which the metric values are considered to determine the slope. The parameter *nGenUnCh* is the number of successive generations for which no significant improvement is obtained (i.e., no significant change in metric values). The final parameter *MaxGen* is the number of maximum allowable generations for evolving a MOEA. Basically, this last parameter specifies the available computational limit for a problem that always needs to be set.

4. Case study: Aalborg energy system

To study the effect of our proposed techniques to use of domain knowledge in initialization and mutation, and our use of the stopping criterion, we choose the Aalborg energy system problem [25]

Table 2
Lower and upper bounds of different decision variables for Aalborg energy system problem.

DVs Bounds	dv_{CHP}	dv_{HP}	dv_{ONW}	dv_{OFW}	dv_{PV}
Lower (MW)	0	0	0	0	0
Upper (MW)	1000	1000	1500	1500	1500

as a test problem.[2] The reason behind choosing this particular problem is that it is extensively analyzed, therefore, all necessary data is available.

Aalborg is the third most populous city in Denmark with approximately 200,000 residents and an area of around 1100 km^2. The energy demand of Aalborg in 2007 was basically met by wind power, fossil fuel-based power plants (i.e., coal, natural gas), and individual heating. Moreover, the transportation sector was completely based on fossil fuels. The 100% renewable energy system based on local renewable resources was developed by the researchers of Aalborg university [38,39,25] and Aalborg is expected to become self-sufficient before the year of 2050. According to [38,39,25], the projected electricity demand for the 100% renewable energy scenario will be 1.47 TWh/year (electric – 0.7 TWh/year, transport – 0.46 TWh/year, heat pump – 0.12 TWh/year, hydrogen generation – 0.19 TWh/year). Additionally, the projected thermal demand is 1.45 TWh/year and mainly met by central district heating, combined heat and power (CHP), and heat pumps. More details on the Aalborg energy system can be found in [25].

As a case study, we optimize the Aalborg energy system by keeping the same electric and thermal demands of the 100% renewable system. We optimize five decision variables (i.e., capacities in MW of: combined heat and power (CHP), heat pump (HP), on-shore wind (ONW), off-shore wind (OFW) and photo-voltaic (PV)) in order to minimize CO_2 emission (EM) and annual cost (AC). The lower and upper bounds for the decision variables are presented in Table 2. The computational cost of about three seconds per evaluation through EnergyPLAN make this a challenging real-world problem.[3]

5. Comprehensive experiments and results

We divide our experiments into three parts. The first part reports the impact of our smart initialization by comparing it against the commonly used random initialization. In the second part, we investigate the influence of the smart mutation by comparing it with the commonly used polynomial mutation. The third part of the experiments assesses our combined approach that uses all techniques (i.e., smart mutation, smart initialization, and stopping criterion) into NSGA-II and SPEA2.

5.1. General experimental settings

In this section, we present all the experimental settings that are used for all three phases of the experiments. Specific experimental settings related to a specific phase will be presented in the corresponding section.

Table 3
General parameter settings for NSGA-II and SPEA2.

	NSGA-II	SPEA2
Population size	100	100
Archive size	–	100
Crossover probability	0.9	0.9
Mutation probability	0.1	0.1
Distribution index	10	10
Maximum evaluations	7000	7000

Table 4
Domain knowledge related to different decision variables and objectives for Aalborg energy system problem.

DVs DK_{Obj}	dv_{CHP}	dv_{HP}	dv_{ONW}	dv_{OFW}	dv_{PV}
$DK_{o_{EM}}$	True	True	True	True	True
$DK_{o_{AC}}$	False	False	False	False	False

All the proposed methodologies are implemented in jMetal [40], a JAVA-based multi-objective meta-heuristic framework.[4] Table 3 shows the general parameter settings that are used in the experiments. Additionally, we use simulated binary crossover [22], polynomial mutation [22] and binary tournament selection [22] for both algorithms. For each phase of the experiments, the algorithms are run independently 30 times to facilitate a statistical analysis of the results. Table 4 presents the domain knowledge used in the experiments to optimize Aalborg energy system. This knowledge is based on intuitive understanding. The first row of the table describes that the increasing capacities of CHP, HP On-, off-shore wind and PV can decrease the emission of Aalborg energy system. In addition, the second row represents that the decreasing of the capacities can decrease the annual cost of the system.

5.2. Evaluation metrics

Evaluation metrics are necessary to measure the quality of the found set of trade-off solutions. To evaluate our proposed initialization method, we will use four metrics that are commonly used, namely the hypervolume (HV) [41], the inverted generational distance (IGD) [42], the additive epsilon approximation [43], and the spread [14]. The HV metric measures the volume covered by a set of solutions in the objective space with respect to a pre-defined reference point. IGD is the average distance of all the solutions in the true Pareto front (tPF) to the nearest solution of given set of solutions. The concept of epsilon approximation is that one determines the minimum distance a found solution set in the objective space needs to be translated to in order to dominate the tPF. Deb at al. [14] introduced an indicator, called spread, to understand how well the front is distributed. Typically, a tPF (or an approximation thereof) is required to calculate the values of all the metrics except for the hypervolume. The reference point can be chosen either arbitrarily or based on a reference set: assuming we are minimizing, the coordinates of the point can be the maximum values per objective that the reference set in the objective space attains.

As the tPF of Aalborg energy system problem is not known in advance, we merge all the found sets of solutions (i.e., solution sets of 240 different individual runs), and we take only the non-dominated solutions from the merged fronts. We use this approximation of the tPF to calculate all the metric values. Please note that higher HV values indicate better results, whereas lower values for the other metrics indicate better results.

[2] Our framework can be applied to identify future optimization scenarios for other cities, states and countries. With the availability of specific domain knowledge, we conjecture that our proposed improvements can solve these problems more efficiently as well.

[3] The simulation was performed on a machine having two 2.6 GHz CPUs with six cores each and 96 GB RAM.

[4] Our complete source code is available online: https://github.com/shaikatcse/EnergyPLANDomainKnowledgeEAStep1.

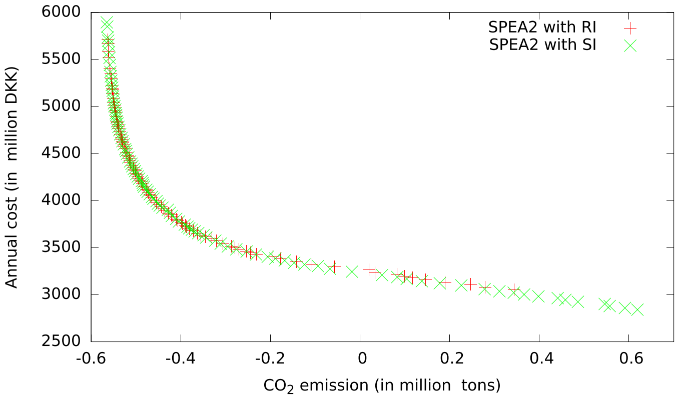

Fig. 4. Pareto fronts generated by SPEA2 with random initialization (RI) and with smart initialization (SI). (For interpretation of the references to color in text near the reference citation, the reader is referred to the web version of this article.)

Table 5
Mean and standard deviation (in subscript) for different quality indicators.

		Mean and standard deviation	
Algorithm	Evaluation metrics	Random Initialization	Smart Initialization
NSGA-II	HV	$8.35e-01_{3.6e-04}$	$8.35e-01_{2.9e-04}$
	IGD	$7.01e-03_{4.0e-04}$	$6.81e-03_{3.5e-04}$
	Epsilon	$6.70e+00_{4.8e+00}$	$2.37e+00_{1.5e+00}$
	Spread	$4.38e-01_{3.6e-02}$	$4.29e-01_{3.5e-02}$
SPEA2	HV	$8.32e-01_{2.4e-03}$	$8.35e-01_{5.1e-04}$
	IGD	$2.51e-02_{2.0e-02}$	$1.00e-02_{1.8e-03}$
	Epsilon	$7.06e+01_{5.5e+01}$	$2.63e+01_{1.2e+01}$
	Spread	$5.98e-01_{4.2e-02}$	$5.45e-01_{2.3e-02}$

5.3. *Influence of smart initialization*

To investigate the impact of our smart initialization (SI), we will compare it with random initialization (RI) that is typically used in MOEA experiments. Random initialization is the process of selecting the value of a decision variable uniformly at random within the lower and upper bound of the decision variable. For the smart initialization process, we set $\theta = 6.0$, $k = 4$ and $\beta \in [0, 1, 2]$ based on preliminary experiments.[5]

Fig. 4 presents two Pareto fronts for Aalborg energy system problem, generated by the SPEA2 algorithms; the red one is generated using RI and the green one uses SI. The *X*-axis represents CO_2 emission in million tons and the *Y*-axis represents annual cost in million Danish Krone (DKK) for Aalborg energy system. It is noticeable that some of the solutions (i.e., scenarios) have negative emissions. This is simply because these scenarios export electricity generated by green sources from within the system to outside partners of the system. The net amount of emission of the system is adjusted by the electricity generation mix of imported electricity. Details of this aspect can be found in the article by Lund [44].

Moreover, the green set of solutions clearly has a better spread, hence, produces more optimized energy scenarios towards the corners that can be interesting to energy planners. In addition, sometimes it produces better solutions than some of the solutions of the Pareto front generated by using RI. However, as previously mentioned, it is required to perform statistical analyses to understand the performance of the algorithms. Therefore, Fig. 5 shows the results as boxplots for the four different evaluation metrics. The means and standard deviations of four metrics for two MOEAs are presented in Table 5. The dark gray shade indicates better results.

Table 6
Mann–Whitney *U*-tests: *p*-values for different metrics when comparing smart initialization (SI) with the common random initialization (RI).

	p-Value	
Evaluation metrics	Compare NSGA-II: with SI and RI	Compare SPEA2: with SI and RI
HV	0.02247	$9.083e^{-12}$
IGD	0.087783	$2.2e^{-16}$
Epsilon	$5.92e^{-05}$	$1.886e^{-04}$
Spread	0.3986	$2.841e^{-07}$

It is very clear from the figure and the table that SI outperforms RI on all the metrics for the two algorithms. Moreover, we perform Mann–Whitney *U*-tests [45][6] on all the metric values of the different runs. The test is performed to test the null hypothesis against an alternative hypothesis to determine whether two samples come from same population or not. In our context, we want to reject the null hypothesis, as the evaluation metric values of applying SI and RI should be significantly different to each other. We consider that the null hypothesis will be rejected if the corresponding *p*-value is less than 0.05. Table 6 presents *p*-values for all the metrics of the two different MOEAs. All the *p*-values except *spread and IGD for NSGA-II* are less than 0.05. The test results shows that there are statistically significant differences for the metrics (except spread for NSGA-II) when SI is used instead of RI. In summary, for most of the metrics, significant amounts of improvements are achieved by using our proposed smart initialization technique.

While it is necessary to compare the final solutions, we are also interested in the actual effect that SI has on the optimization. As SPEA2 benefits more than NSGA-II from the use of SI, we are showing the development of the indicator values over time in Fig. 6. As we can see, our smart initialization strategy results in significantly better starting points for the optimization process than random initialization does. In addition, SI also appears to provide better populations for the subsequent optimization, as the progress over time is "steeper" in comparable parts of the optimization. For example in the case of additive epsilon approximation, it takes six generations to improve from approximately 380 to 200 when SI is used, whereas twice as much time is needed when random initialization is used. Similar trends can be observed for development of the other indicator values. We conjecture that the bias of the initial populations as generated by SI has favorable properties that are exploited in the subsequent optimization, whereas RI with its uniform initialization is not problem-specific.

As a final note, the chosen approach to initialize the populations based on the maximization of Solow and Polasky diversity metric is computationally relatively expensive, taking approximately 40 minutes each time on our machine. While it is possible to maximize Solow and Polasky metric in slightly different approach that reduce the computational time within few seconds; we continue to use the approach proposed in [33] and described in Section 2.2.2, as we do not intend to compare different approaches to maximize Solow and Polasky diversity metric.

5.4. *Influence of smart mutation*

In this section, we investigate the impact of smart mutation on the Aalborg energy system problem. To do so, we compare polynomial mutation (PM) with our smart mutation (SM) using the same evaluation metrics as above for NSGA-II and SPEA2 as the underlying MOEAs. For each MOEA, the same initial population is used for each run to ensure a fair comparison between PM and SM. By using

[5] We conjecture that additional performance gains are possible, however, a tuning of these parameters is beyond the scope of this article.

[6] A non-parametric statistical test, also known as Wilcoxon rank-sum test.

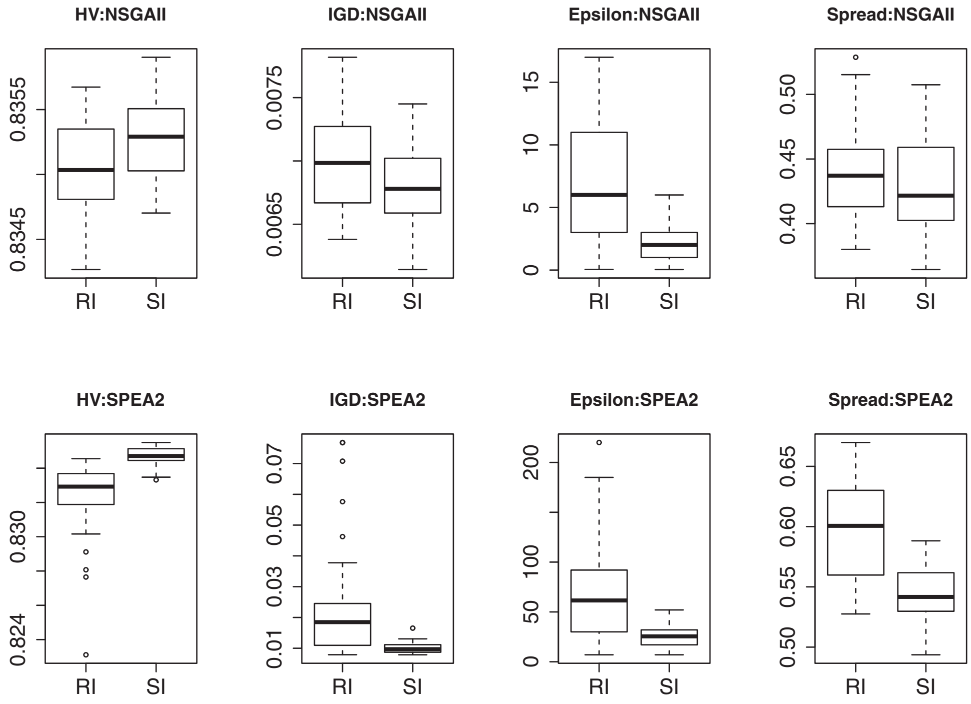

Fig. 5. Boxplots for four metrics comparing random initialization (RI) with smart initialization (SI) for NSGA-II and SPEA2.

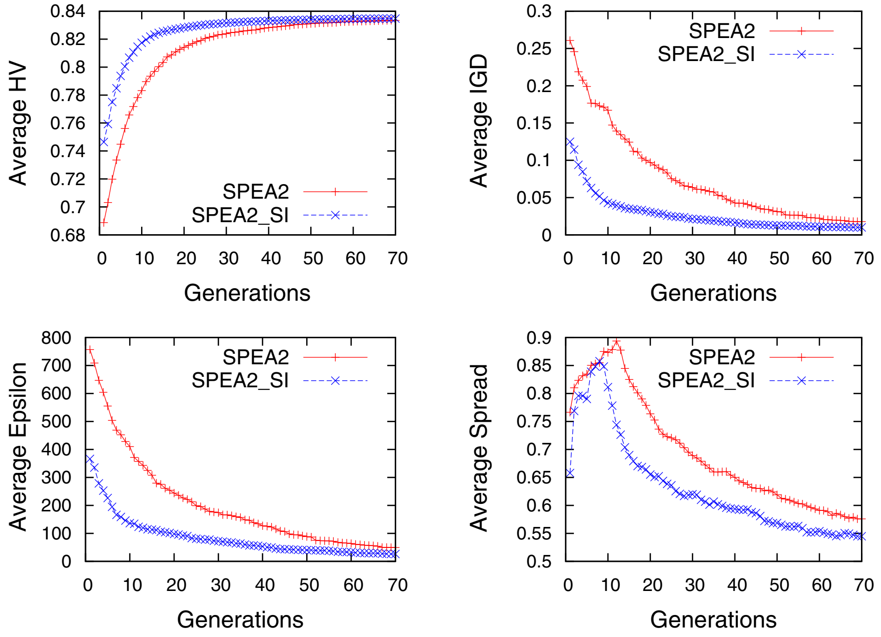

Fig. 6. Development of the indicators during the optimization runs for SPEA2.

the same initial populations we cut down any advantage gain by a MOEA in the initial phase. Therefore, the settings ensure that if any MOEA performs better than any other, then this is strictly due to the use of our specific mutation SM.

Fig. 7 shows two Pareto fronts generated by SPEA2 using PM and SM. The Pareto front with SM has better spread and produces better solutions. Fig. 8 presents the comparison of PM and SM as boxplots. For all the metrics, SM performs better than PM. However, the statistical test (Table 7) shows that not all the metrics are not statistically significantly different for NSGA-II. Nevertheless, significant improvement is achieved for SPEA2. From the boxplots and the statistical test, it can be concluded that SM provides a good performance improvement for both MOEAs.

Table 7
Mann–Whitney *U*-tests: *p*-values for different metrics when comparing our smart mutation (SM) with the common polynomial mutation (PM).

	p-Value	
Evaluation metrics	Compare NSGA-II: with SM and PM	Compare SPEA2: with SM and PM
HV	0.0906	$1.51e^{-05}$
IGD	0.2539	$1.20e^{-16}$
Epsilon	0.01664	0.07721
Spread	0.7082	$1.34e^{-04}$

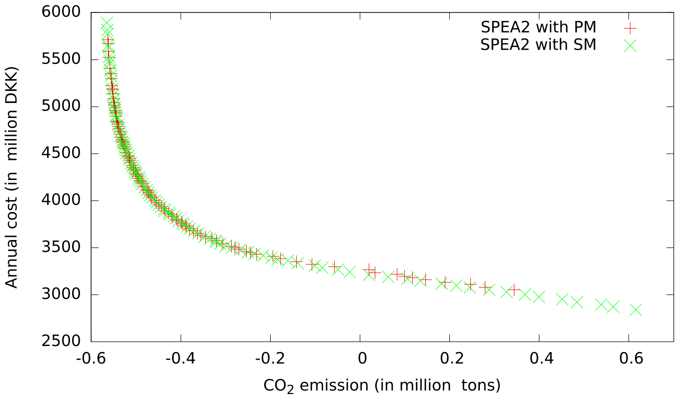

Fig. 7. Pareto fronts generated by SPEA2 with polynomial mutation (PM) and with smart mutation (SM).

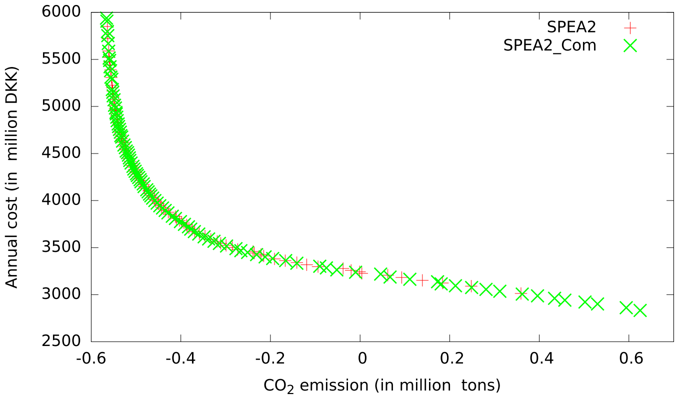

Fig. 9. Pareto fronts generated by generic SPEA2 and SPEA2_Com.

5.5. Combined approach

Finally, we want to compare a combined approach (i.e., smart initialization, smart mutation and stopping criterion are integrated) with a typical approach (i.e., random initialization, polynomial mutation and fixed maximum generations) for NSGA-II and SPEA2 on the Aalborg energy system problem.

For this experiment, we use the following parameters for the stopping criterion: $nGenLT = 20$, $nGenUnCH = 5$, $\alpha = 0.05$ and $MaxGen = 70$. The values are chosen in such a way that the stopping happens early, which we prefer as we are using costly simulations. All other parameters remain unchanged.

First, Fig. 9 presents an example of the comparison of two Pareto fronts generated by the generic SPEA2 and by SPEA2 with the combined approach (SPEA2_Com). Again, SPEA2_Com clearly performs better than the generic approach. In addition, Table 8 shows the results of the four different MOEAs, where a darker cell color indicates a better result. Fig. 10 shows the results as boxplots to compare NSGA-II and NSGA-II with the combined approach (NSGA-II_Com), and similarly for SPEA2. Table 9 presents the *p*-values for all the metrics. From the boxplots and Table 8, it is clear that our combined approach performs very similarly to a typical approach for NSGA-II, however, it outperforms a typical approach on every chosen metric for SPEA2. Nevertheless, it should be noted that our combined approach achieves the similar results for NSGA-II with fewer function evaluations (Table 10). Hence, it can be concluded that our combined technique has a positive impact on MOEAs. For NSGA-II, our technique does not provide significantly better results in terms of metrics values, however, on average almost one-forth of total function evaluations (Table 10) are saved. On the other hand, for SPEA2, a significant improvement of the metric values are obtained, compared to the typical method.

Lastly, we list in Table 10 the total number and average percentage of saved function evaluations for each MOEA for 30 runs. Note that all the generic simulations without our stopping criterion technique use 7000 function evaluations (i.e., 70 generations). We calculate the number of saved function evaluations (SFE) for each run by $SFE = (70 - Gen_{sc}) * PS$, where Gen_{sc} is stopping generation number of the stopping criterion and PS is the population size. Then, we sum up all SFE values of the 30 runs to find the total SFE.

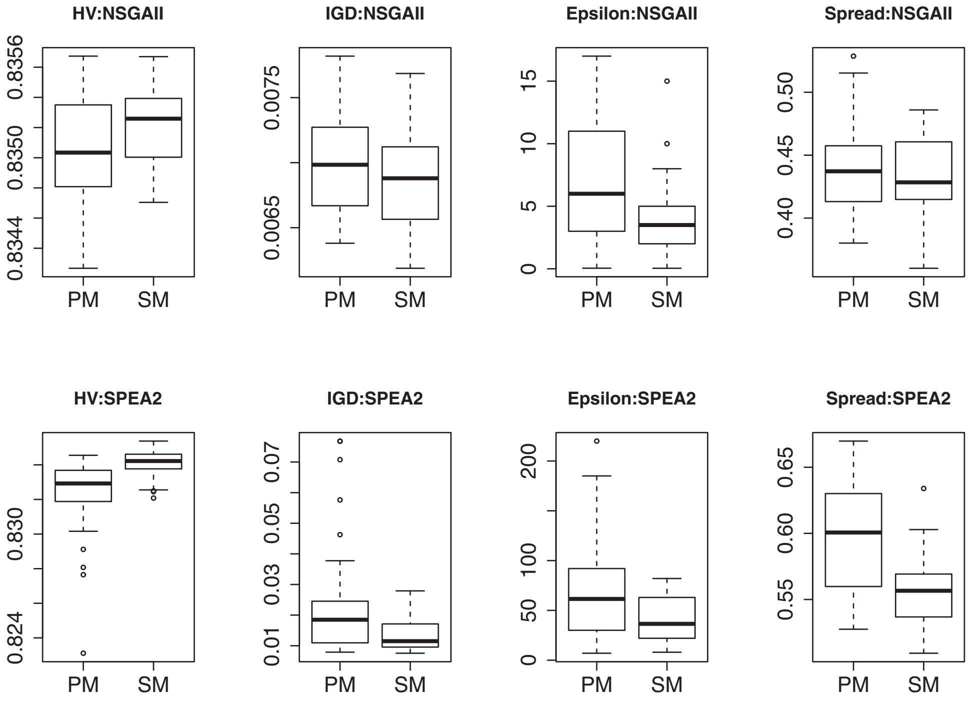

Fig. 8. Boxplots for four metrics comparing PM with SM for NSGA-II and SPEA2.

Table 8
Mean and standard deviations of four different metrics.

Metric	NSGA-II	NSGA-II_Com	SPEA2	SPEA2_Com
HV	$8.35e-01_{3.6e-04}$	$8.35e-01_{3.5e-04}$	$8.32e-01_{2.4e-03}$	$8.35e-01_{7.5e-04}$
IGD	$7.01e-03_{4.0e-04}$	$6.85e-03_{4.3e-04}$	$2.51e-02_{2.0e-02}$	$9.96e-03_{2.9e-03}$
Epsilon	$6.70e+00_{4.8e+00}$	$6.57e+00_{4.9e+00}$	$7.06e+01_{5.5e+01}$	$2.03e+01_{1.4e+01}$
Spread	$4.38e-01_{3.6e-02}$	$4.27e-01_{3.1e-02}$	$5.98e-01_{4.2e-02}$	$5.51e-01_{2.5e-02}$

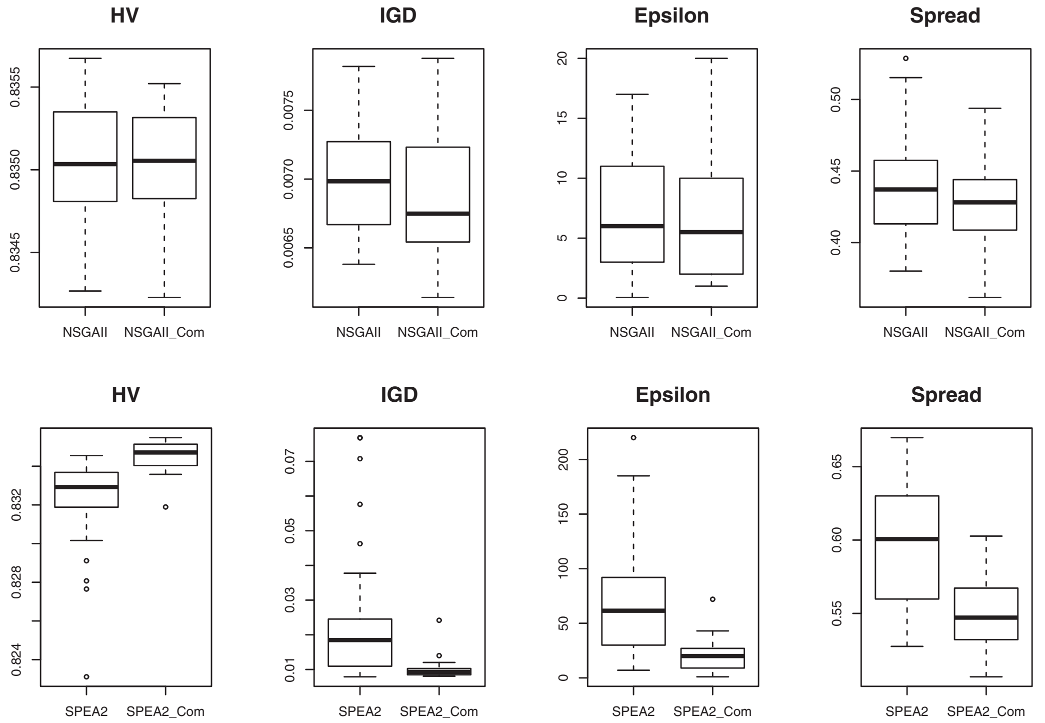

Fig. 10. Boxplots for four metrics comparing NSGA-II and SPEA2 with combined approaches.

Table 9
Mann–Whitney *U*-tests: *p*-values for different metrics when comparing our combined approaches (Com) with the generic approaches.

	p-Value	
Evaluation metrics	Compare NSGA-II: with combined and generic approach	Compare SPEA2: with combined and generic approach
HV	0.7191	$1.63e^{-09}$
IGD	0.1727	$2.20e^{-16}$
Epsilon	0.8704	$6.932e^{-06}$
Spread	0.3136	$4.106e^{-09}$

Table 10
Number and percentage of saved function evaluations for different MOEAs.

MOEA	Total saved function evaluation	Average percentage of saved function evaluations
NSGA-II_Com	45,400	22%
SPEA2_Com	27,400	13%

The average percentage of SFE is calculable by $\frac{SFE^T*100}{FE^T}$, where SFE^T is the total SFE and $FE^T = 210{,}000$ as this is the total number of function evaluations required when the stopping criterion is not used. According to the results, the modification of NSGA-II has saved the largest number of function evaluations. On average 22% function evaluations are saved, while also yielding a similar performance of the final solutions (see Table 8 and Fig. 10).

The poor results of generic SPEA2 can be also explained by the number of saved evaluations. According to the table, fewer evaluations are saved compared to NSGA-II. Therefore, it is clear that SPEA2 is not able to converge as quickly as NSGA-II on this problem.

6. Conclusion

Current and future energy systems will include more and more renewable energy sources. To accurately plan such systems, complex and computationally costly simulations are typically used to assess configurations according to different objectives, for example, based on their cost and their emissions.

General purpose multi-objective evolutionary algorithms are often used to solve such problems, however, the simulation cost result in time-consuming optimizations. In this article, we present and combine different techniques to improve both solution quality and speed of an optimization. First and foremost, we incorporate basic domain knowledge about energy systems into different operators of such algorithms in order to increase the solution quality. In addition, we also adapt a recently-developed stopping criterion to save simulations.

Our results on the Danish Aalborg energy system problem reveal that our problem-specific approaches achieve significant improvements over generic state-of-the-art approaches, both in terms of solution quality and optimization speed. It is noteworthy that this was achieved with rather basic domain knowledge. It remains to be seen how much solution quality can be improved further by using more detailed knowledge.

References

[1] IEA, 2014 Key World Energy Statistics, Tech. Rep., International Energy Agency, 2014.

[2] P. Østergaard, K. Sperling, Towards sustainable energy planning and management, Int. J. Sustain. Energy Plan. Manage. 1 (2014) 1–5, http://dx.doi.org/10.5278/ijsepm.2014.1.1.

[3] P.A. Østergaard, Reviewing optimisation criteria for energy systems analyses of renewable energy integration, Energy 34 (9) (2009) 1236–1245, http://dx.doi.org/10.1016/j.energy.2009.05.004.

[4] A. Pina, C.A. Silva, P. Ferrão, High-resolution modeling framework for planning electricity systems with high penetration of renewables, Appl. Energy 112 (2013) 215–223, http://dx.doi.org/10.1016/j.apenergy.2013.05.074.

[5] D. Connolly, H. Lund, B.V. Mathiesen, M. Leahy, A review of computer tools for analysing the integration of renewable energy into various energy systems, Appl. Energy 87 (4) (2010) 1059–1082, http://dx.doi.org/10.1016/j.apenergy.2009.09.026.

[6] C. Dong, G. Huang, Y. Cai, Y. Liu, An inexact optimization modeling approach for supporting energy systems planning and air pollution mitigation in Beijing city, Energy 37 (1) (2012) 673–688, http://dx.doi.org/10.1016/j.energy.2011.10.030.

[7] A. Alarcon-Rodriguez, G. Ault, S. Galloway, Multi-objective planning of distributed energy resources: a review of the state-of-the-art, Renew. Sustain. Energy Rev. 14 (5) (2010) 1353–1366, http://dx.doi.org/10.1016/j.rser.2010.01.006.

[8] M. Sharafi, T.Y. ELMekkawy, Multi-objective optimal design of hybrid renewable energy systems using PSO-simulation based approach, Renew. Energy 68 (2014) 67–79, http://dx.doi.org/10.1016/j.renene.2014.01.011.

[9] Q. Zhang, B.C. Mclellan, T. Tezuka, K.N. Ishihara, An integrated model for long-term power generation planning toward future smart electricity systems, Appl. Energy 112 (2013) 1424–1437, http://dx.doi.org/10.1016/j.apenergy.2013.03.073.

[10] M.S. Mahbub, M. Cozzini, P.A. Østergaard, F. Alberti, Combining multi-objective evolutionary algorithms and descriptive analytical modelling in energy scenario design, Appl. Energy 164 (2016) 140–151, http://dx.doi.org/10.1016/j.apenergy.2015.11.042.

[11] M. Fadaee, M. Radzi, Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: a review, Renew. Sustain. Energy Rev. 16 (5) (2012) 3364–3369, http://dx.doi.org/10.1016/j.rser.2012.02.071.

[12] Y. Katsigiannis, P. Georgilakis, E. Karapidakis, Multiobjective genetic algorithm solution to the optimum economic and environmental performance problem of small autonomous hybrid power systems with renewables, Renew. Power Gener. IET 4 (5) (2010) 404–419, http://dx.doi.org/10.1049/iet-rpg.2009.0076.

[13] M. Esmaili, M. Rajabi, Optimal charging of plug-in electric vehicles observing power grid constraints, IET Gener. Transm. Distrib. 8 (April 2014) 583–590, http://dx.doi.org/10.1049/iet-gtd.2013.0628.

[14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[15] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Tech. Rep., TIK-Report No. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, 2001.

[16] H. Energy, The HOMER Team – Hybrid Renewable and Distributed Generation System Design Software. http://www.homerenergy.com/ (online; accessed 04.06.15).

[17] T. Lambert, P. Gilman, P. Lilienthal, Micropower system modeling with HOMER, Integr. Altern. Sources Energy 1 (1) (2006) 379–385.

[18] H2RES. http://h2res.fsb.hr/ (online; accessed 04.06.15).

[19] H. Lund, N. Duić, G. Krajac, M. da Graça Carvalho, et al., Two energy system analysis models: a comparison of methodologies and results, Energy 32 (6) (2007) 948–954, http://dx.doi.org/10.1016/j.energy.2006.10.014.

[20] Sustainable Energy Vision 2050. http://www.inforse.org/europe/Vision2050.htm (online; accessed 04.06.15).

[21] Long-range Energy Alternatives Planning System. http://www.energycommunity.org/ (online; accessed 04.06.15).

[22] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, vol. 16, John Wiley & Sons, 2001.

[23] M.S. Mahbub, A domain knowledge-based multi-objective evolutionary algorithm for optimizing energy systems, in: MENDEL 2014, vol. 20, 2014.

[24] M.S. Mahbub, T. Wagner, L. Crema, Improving robustness of stopping multi-objective evolutionary algorithms by simultaneously monitoring objective and decision space, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO'15, ACM, 2015, pp. 711–718, http://dx.doi.org/10.1145/2739480.2754680.

[25] P.A. Østergaard, B.V. Mathiesen, B. Möller, H. Lund, A renewable energy scenario for Aalborg municipality based on low-temperature geothermal heat, wind power and biomass, Energy 35 (12) (2010) 4892–4901, http://dx.doi.org/10.1016/j.energy.2010.08.041.

[26] T. Dai, W. Qiao, Trading wind power in a competitive electricity market using stochastic programming and game theory, IEEE Trans. Sustain. Energy 4 (July 2013) 805–815, http://dx.doi.org/10.1109/TSTE.2013.2251917.

[27] C. Li, X. Yu, W. Yu, T. Huang, Z.-W. Liu, Distributed event-triggered scheme for economic dispatch in smart grids, IEEE Trans. Ind. Inform. PP (99) (2015) 1, http://dx.doi.org/10.1109/TII.2015.2479558.

[28] A. Dizqah, A. Maheri, K. Busawon, A. Kamjoo, A multivariable optimal energy management strategy for standalone dc microgrids, IEEE Trans. Power Syst. 30 (Sept 2015) 2278–2287, http://dx.doi.org/10.1109/TPWRS.2014.2360434.

[29] P.P. Bonissone, R. Subbu, N. Eklund, T.R. Kiehl, Evolutionary algorithms + domain knowledge = real-world evolutionary computation, IEEE Trans. Evol. Comput. 10 (3) (2006) 256–280, http://dx.doi.org/10.1109/TEVC.2005.857695.

[30] T. Friedrich, M. Wagner, Seeding the initial population of multi-objective evolutionary algorithms: a computational study, Appl. Soft Comput. 33 (2015) 223–230, http://dx.doi.org/10.1016/j.asoc.2015.04.043.

[31] J.E. Gentle, Random Number Generation and Monte Carlo Methods, Springer Science & Business Media, 2006.

[32] L. Devroye, Non-uniform Random Variate Generation, Springer-Verlag, 1986.

[33] T. Ulrich, Exploring Structural Diversity in Evolutionary Algorithms (Ph.D. thesis), ETH, Zurich, 2012.

[34] L. Masisi, V. Nelwamondo, T. Marwala, The use of entropy to measure structural diversity, in: IEEE International Conference on Computational Cybernetics, ICCC 2008, IEEE, 2008, pp. 41–45, http://dx.doi.org/10.1109/ICCCYB.2008.4721376.

[35] O.M. Shir, M. Preuss, B. Naujoks, M. Emmerich, Enhancing decision space diversity in evolutionary multiobjective algorithms, in: Evolutionary Multi-criterion Optimization, Springer, 2009, pp. 95–109, http://dx.doi.org/10.1007/978-3-642-01020-0_12.

[36] A.R. Solow, S. Polasky, Measuring biological diversity, Environ. Ecol. Stat. 1 (2) (1994) 95–103.

[37] D. of Development and A. U. Planning, EnergyPLAN | Advanced energy system analysis computer model.

[38] P.A. Østergaard, Baggrundsrapport for Energivision for Aalborg Kommune 2050, Institut for Samfundsudvikling og Planlægning, Aalborg Universitet, 2010.

[39] P.A. Østergaard, H. Lund, F. Hvelplund, B. Möller, B.V. Mathiesen, A. Remmen, L.M. Odgaard, Energivision for Aalborg Kommune 2050, Institut for Samfundsudvikling og Planlægning, Aalborg Universitet, 2010.

[40] J.J. Durillo, A.J. Nebro, jMetal: a Java framework for multi-objective optimization, Adv. Eng. Softw. 42 (2011) 760–771, http://dx.doi.org/10.1016/j.advengsoft.2011.05.014.

[41] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271, http://dx.doi.org/10.1109/4235.797969.

[42] C.A.C. Coello, M.R. Sierra, A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm, in: MICAI 2004: Advances in Artificial Intelligence, Springer, 2004, pp. 688–697, http://dx.doi.org/10.1007/978-3-540-24694-7_71.

[43] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. Da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132, http://dx.doi.org/10.1109/TEVC.2003.810758.

[44] H. Lund, EnergyPLAN: Advanced Energy System Analysis Computer Model, Tech. Rep., Aalborg University, Denmark, 2014.

[45] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60.