Optimal Power System Generation Scheduling by Multi-Objective Genetic Algorithms With Preferences

E. Zio*, P. Baraldi, N. Pedroni
Department of Nuclear Engineering, Polytechnic of Milan, Via Ponzio 34/3, 20133 Milan, Italy
*Corresponding author. Phone: +39-2-2399-6340; fax: +39-2-2399-6309
E-mail address: enrico.zio@polimi.it

Abstract

Power system generation scheduling is an important issue both from the economical and environmental safety viewpoints. The scheduling involves decisions with regards to the units start up and shut down times and to the assignment of the load demands to the committed generating units for minimizing the system operation costs and the emission of atmospheric pollutants.

As many other real-world engineering problems, power system generation scheduling involves multiple, conflicting optimization criteria for which there exists no single best solution with respect to all criteria considered. Multi-Objective optimization algorithms, based on the principle of Pareto optimality, can then be designed to search for the set of nondominated scheduling solutions from which the decision maker must a posteriori choose the preferred alternative. On the other hand, often, information is available a priori regarding the preference values of the decision maker with respect to the objectives. When possible, it is important to exploit this information during the search so as to focus it on the region of preference of the Pareto optimal set.

In this paper, ways are explored to use this preference information for driving a Multi-Objective Genetic Algorithm towards the preferential region of the Pareto optimal front. Two methods are considered: the first one extends the concept of Pareto dominance by biasing the chromosome replacement step of the algorithm by means of numerical weights which express the decision maker’s preferences; the second one drives the search algorithm by changing the shape of the dominance region according to linear trade-off functions specified by the decision maker.

The effectiveness of the proposed approaches is first compared on a case study of literature, then, a nonlinear, constrained, two-objective power generation scheduling problem is effectively tackled.

Keywords. Power system generation scheduling; environmental safety; evolutionary algorithm; multi-objective optimization; Pareto optimality; preferences; weights; guided dominance.

Acknowledgment
The authors acknowledge that the work presented in this paper has been partially funded by the Italian Ministry of University and Research (MIUR), within the project PRIN 2005093881-004 (years 2005-2006).

1. INTRODUCTION

There are two tasks considered in power system generation scheduling [44]: one is the unit committment which determines the unit start up and shut down schedules in order to minimize the system fuel expenditure; the other is the economic dispatch which assigns the system load demand to the committed generating units for minimizing the power generation cost. Many studies for power system generation scheduling have addressed this problem with various mathematical algorithms [11], including dynamic programming [35], artificial neural networks [43] and evolutionary algorithms [18].

In addition, the increasing public awareness of environmental protection and the passage of the Clean Air Act Amendments of 1990 have forced the utilities to modify their design and operational strategies to reduce pollution and atmospheric emissions from the power plants. The emission dispatching option is an attractive short-term alternative in which both emission and fuel cost are to be minimized. In recent years, this option has received much attention since it requires only a small modification of the basic economic dispatch problem to include emissions [1], [22].

Such optimization problems involve multiple objectives. These objectives need to be considered simultaneously and are usually conflicting, so that it is not possible to find a single solution which is optimal with respect to all objectives. Instead, there exists a number of so called “Pareto optimal” solutions for which an improvement in any one objective can only be obtained at the expense of degradation in at least one other objective. In the absence of any additional preference information, none of these solutions can be said to be inferior when compared to any other Pareto optimal solution, as it is superior in at least one criterion. It is the decision maker (DM) who needs to solve the dilemma of the final choice by articulating his/her preferences about the objectives. Following the classification proposed in [42], this articulation of preferences may be done either before (a priori), during (progressive) or after (a posteriori) the multi-objective optimization process.

The approaches based on an a priori articulation of the preferences require the decision maker to explicitly weigh the different objectives so as to effectively translate the multi-objective optimization problem into a single-objective one, prior to knowing the possible alternatives [29], [30].
In progressive approaches, decision making and optimization are intertwined. An example of such an approach is given in [41], where an Artificial Neural Network is used to learn the user's preferences during the course of the optimization.

Most Evolutionary Multi-Objective Optimization (EMO) approaches can be classified as a posteriori [9], [19], [23], [28], [42]. They attempt to discover the whole set of Pareto optimal solutions or at least a well distributed set of representatives. The DM then looks at the set (possibly very large) of generated alternatives and makes his/her final decision based on his/her preferences. However, if the number of objectives is large, the search for all Pareto optimal solutions is a time-consuming process and poses high demands on the optimization algorithm. Further, if the Pareto-optimal solutions are numerous their analysis to reach the final decision is quite a challenging and burdensome process for the DM.

Although it is usually unfeasible for a DM to completely specify his or her preferences before any alternatives are known, the DM often has a rough idea of the preferential goals towards which the search should be directed [21], so that he/she may be able to articulate vague, linguistic degrees of importance [16], [36] or give reasonable trade-offs between the different objectives [4], [5]. Such information should be integrated into the multi-objective optimization approach to bias the search towards solutions that are preferential for the DM. This would in principle yield two important advantages: 1) instead of a diverse set of solutions, many of which irrelevant to the DM, a search biased towards the DM's preferences will yield a more fine-grained, and thus more suitable, selection of alternatives; 2) by focusing the search onto the relevant part of the search space, the optimization algorithm is expected to find these solutions more quickly.

In this paper, two methods to account for the DM's preferences during the search for Pareto-optimal solutions are integrated within a Multi-Objective Genetic Algorithm (MOGA) optimization framework [8], [26], [27]. The first method is a modified version of the Weighted Pareto (WP) optimization method [14], [16] in which quantitative weights are combined with the concept of Pareto dominance to drive the MOGA towards relevant regions of the Pareto optimal front. In the original algorithm, the weights are used to bias the chromosome selection step of the MOGA search; however, this way of proceeding has been found to produce a weak and coarse biasing pressure towards the preferred regions of the Pareto-optimal front. For this reason, in this work the algorithm has been modified by shifting the bias to the replacement phase of the MOGA with the effect of increasing the rapidity of convergence and the accuracy of the search.

The second method is the so-called Guided Multi-Objective Genetic Algorithm (G-MOGA), based on the guided domination principle which allows the DM to change the shape of the dominance region specifying maximal and minimal trade-offs between the different objectives,
so as to efficiently guide the MOGA towards Pareto optimal solutions within these boundaries [4], [5].

The proposed preference-guided search schemes are tested on a common problem of literature [16] and then compared with a standard Pareto-based MOGA on a nonlinear, constrained, two-objective power generation scheduling problem [45] which regards the identification of the generating units to be committed (on or off) on a given scheduling horizon in order to minimize operation costs and pollution emissions (NO$_x$). Simulation results show the G-MOGA outperforming both the weighted and the standard Pareto-based MOGA with respect to speed of convergence and quality of the Pareto optimal solutions found.

Furthermore, the final decision-making problem of which solution to choose among those contained in the biased Pareto-optimal fronts identified by the preferential optimization algorithms is tackled by three techniques of literature [33] and a critical comparison of the results thereby obtained is offered.

The paper is organized as follows. In Section 2, a brief overview of preference handling methods in EMO is given. In Section 3, the modified weighted Pareto optimization method and the guided domination principle are described in details. In Sections 4 and 5, the developed approaches are tested and compared on a case study of literature and on a real-world two-objective optimization problem. Finally, some conclusions are proposed in the last Section.

2. PREFERENCES IN EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION: RELATED WORK

In a multi-objective optimization problem, several possibly conflicting objective functions $f_i(\cdot)$, $i = 1, 2, \ldots, n_f$, must be evaluated in correspondence of each decision variable vector $U$ in the search space. The final goal is to identify the solution vector $U^*$ which gives rise to the best compromise among the various objective functions. The comparison of solutions may be achieved in terms of the concepts of Pareto optimality and dominance [38]: with reference to a minimization problem, solution $U_a$ is said to dominate $U_b$ ($U_a > U_b$) if

$$\forall i \in \{1,2,\ldots,n_f\}; f_i(U_a) \leq f_i(U_b)$$

and

$$\exists j \in \{1,2,\ldots,n_f\}; f_j(U_a) < f_j(U_b).$$

The decision variable vectors which are not dominated by any other of a given set are called nondominated with respect to this set; the decision variable vectors that are nondominated
within the entire search space are said to be *Pareto optimal* and constitute the so called *Pareto optimal set* or *Pareto optimal front*.

There has been a number of attempts to account for the decision maker preferences in EMO so as to focus the search to the preferred subset of Pareto optimal solutions [10].

The approaches described in [21] and [24] exploit goal programming to allow specification of the DM goals in terms of a vector of desired characteristics \( g = (g_1, g_2, \ldots, g_n) \). In [24], a higher importance in the search is given to those objectives still not satisfying the goal. In [21], the goal is used to modify the optimization criteria: if the goal with respect to criterion \( i \) is to find a solution \( U \) such that \( f_i(U) \leq g_i \), then \( f_i(U) = \max\{0, f_i(U) - g_i\} \) is considered during the search to take into account how far solution \( U \) takes us from the goal. If set appropriately, this approach may indeed restrict the search space to the preferred region of the Pareto front. However, the problem is to decide on the goal a priori, i.e. before the Pareto optimal front is known. For instance, if the goal is too ambitious, none of the solutions will reach the goal, even for a single objective, and thus the search is not restricted at all; on the other hand, if the goal is set within the feasible region, it prevents the search for improved solutions.

The weighted Pareto optimization method, introduced in [14], [16] and further developed in [12], [15], extends the Pareto principle by specifying the relative importance of the different objectives through the definition of numerical weights. For a given weight-vector \( w = (w_1, w_2, \ldots, w_n) \), whose elements sum to one, and a real number \( 0 < \tau \leq 1 \), the solution vector \( U_a \) is said to \( (w, \tau) \)-dominate the solution \( U_b \) \( (U_a \succ_w^\tau U_b) \) if and only if

\[
\sum_{i=1}^n w_i \cdot I_i(U_a, U_b) \geq \tau
\]  

(2)

where

\[
I_i(U_a, U_b) = \begin{cases} 
1, & f_i(U_a) \leq f_i(U_b) \\
0, & f_i(U_a) > f_i(U_b)
\end{cases}, \quad i = 1, 2, \ldots, n_f
\]  

(3)

with inequalities holding in at least one case. The standard definition of dominance could be obtained by setting \( \tau = 1 \) and uniform weights \( w_1 = w_2 = \ldots = w_n = 1/n_f \).

With this Pareto dominance definition, a GA population of solutions to the optimization problem can be sorted as follows: a) the elements are first sorted according to the dominance criterion (2); b) if two elements are nondominated, they are sorted according to their ranks [17], [23], [40]; c) if two elements are both nondominated and have the same rank they are sorted according to their fitness, calculated as a weighted sum of the objective values.
However, since for every objective the dominance scheme only considers whether one solution is better than another solution, and not by how much it is better, this approach allows only a very coarse guidance and is difficult to control.

In practical applications, it is expected that the DM's preferences be expressed in qualitative terms, which then need to be translated into numerical weights. To facilitate the specification of the weights, in [13] a method is suggested to turn fuzzy, linguistic preferences into specific quantitative weights; for completeness, this method is summarized in Appendix A at the end of the paper.

A way to convert fuzzy preferences into weight intervals is also proposed in [31] and combined with a dynamic weighted aggregation GA to obtain the corresponding Pareto optimal solutions. This approach converts the multi-objective optimization problem into a single-objective one by weighted aggregation, but varies the weights dynamically during the optimization run within the relevant boundaries.

Finally, in the biased sharing approach the sharing method is altered such that the density of solutions produced by the GA is higher in preferred regions and lower elsewhere [20]. The preferred regions are specified by weighting the different criteria. Since with this method usually the whole Pareto front is outlined, only with different densities, it may give the decision maker some idea as to how the Pareto optimal front looks like, even away from the region specified as preferred. As drawbacks, the method requires to know whether the Pareto optimal front is convex or not and does not allow to focus on a given area with similar weights on all criteria (for that setting, the whole Pareto front would be covered equally, just as in the standard approach). A new biased crowding approach based on the idea of biased sharing has been successively developed to overcome these drawbacks [3].

In the present work, two approaches have been considered: a modified form of the weighted Pareto optimization method and the guided domination approach. A thorough discussion is given in the following section.

3. TWO APPROACHES TO PREFERENTIAL OPTIMIZATION

3.1 Modified Weighted Pareto-based Multi-Objective Genetic Algorithm (WP-MOGA)

In the WP-MOGA, the DM's preferences are expressed by a priori assigning to each criterion a weight \( w_i \), \( i = 1, 2, \ldots, n_f \). As mentioned in the previous Section, in practice these weights will be obtained by translating linguistic preferences elicited from the DM into specific quantitative weights, while ensuring consistency [13].
Once the DM’s weights are specified, the WP-MOGA proceeds to use such information on the DM’s preferences within the concept of Pareto dominance underpinning the selection step of the MOGA search; this allows driving the search towards preferred regions of the Pareto optimal front, according to the following computational flow [15]:

i. The GA population is sorted as follows: a) the elements of the population are first sorted according to the dominance criterion (2); b) if two elements are nondominated, they are sorted according to their ranks [17], [23], [40]; c) if two elements are both nondominated and have the same rank, they are sorted according to their fitness, calculated as a weighted sum of the objective values;

ii. Two parents are selected from the sorted population by using binary domination tournaments; two candidate parents are randomly selected and compared: if one dominates the other (or if it has a higher rank or fitness), it wins and it is selected as the first parent. The same procedure is repeated for choosing the second parent;

iii. The parents are crossed to obtain two children;

iv. The children are kept in the population;

v. Return to step i. above.

However, it has been found experimentally that the biasing pressure applied to the selection phase of the MOGA search (steps i. and ii. above) by the extended Pareto dominance criterion (2) and by the weighted sum of objective values is too weak to allow an effective guidance of the algorithm towards preferred regions of the Pareto optimal front. To improve the search, in this work the GA computational flow has been modified by shifting the pressure of the DM’s preferences to the replacement phase of the MOGA search (steps iii. and iv. below):

i. Two parents are randomly selected from the population, sorted according to a standard nondomination ranking procedure in [40]: individuals are ranked by iteratively determining the nondominated solutions in the population (nondominated front), assigning the next best rank to those individuals and removing them from the population;

ii. The parents are crossed to obtain two children;

iii. The four individuals involved (two parents and two children) are sorted according to the above nondomination ranking procedure [40]; if two elements have the same rank, they are sorted according to their fitness, calculated as a weighted sum of the objective values;

iv. The two fittest individuals are kept in the population.
v. Return to step i. above.

The nondominated sorting procedure ensures convergence to the Pareto optimal front while the biased replacement guided by the weighted aggregation of objectives is expected to more effectively and precisely drive the MOGA towards the preferred regions of the Pareto optimal front itself.

Finally, it seems worthwhile to stress that the use of weighted sums of the objective functions to rank the GA population (step i. above) requires that the objectives be normalized in the range [0, 1]: this is not always trivial [17]. On the other hand, failing to perform the normalization would result in increased importance of those objectives with large values and decreased importance of those objectives with small values (the so called scaling effect [6]). One method may be that of obtaining the positive ideal solution, i.e., an optimal value for each objective single-optimized separately, and use this as reference point for normalization [25]: however, this way of proceeding can be expensive and even infeasible if the number of objectives is large. In the present work, the effective range method has been adopted which dynamically changes the normalization factors as improved solutions are obtained [37]: being \( f_i \) the generic \( i \)-th objective, \( i = 1, 2, \ldots, n_f \), and \( f_{\text{max}}(t) \) and \( f_{\text{min}}(t) \) its maximum and minimum values in the first \( t \) generations, respectively, the transformed objective \( f^*_i = \frac{(f_i - f_{\text{min}}(t))}{(f_{\text{max}}(t) - f_{\text{min}}(t))} \) at generation \( t \) is automatically normalized to [0, 1].

### 3.2 Guided Multi-Objective Genetic Algorithm (G-MOGA)

In the G-MOGA, user preferences are taken into account by modifying the definition of dominance [4], [5].

For each criterion, a weighted utility function of the objective vector \( f = (f_1, f_2, \ldots, f_{n_f}) \) is defined as follows:

\[
\Omega_i(f(U)) = f_i(U) + \sum_{i \neq j} a_{ij} \cdot f_j(U), i = 1, 2, \ldots, n_f
\]

where \( a_{ij} \) the amount of loss in the \( i \)-th objective function that the DM is disposed to accept for a gain of one unit in the \( j \)-th objective function. Obviously, \( a_{ii} = 1 \).

A new domination concept is then defined: with reference to a minimization problem for example, a solution \( U_a \) dominates another solution \( U_b \) if \( \Omega_i(f(U_a)) \leq \Omega_i(f(U_b)) \) for all \( i = 1, 2, \ldots, n_f \) and the strict inequality is satisfied for at least one objective.

The basic idea behind the G-MOGA is that although the decision maker may be unable to define exactly his/her preferences in the form of weights on the different objectives, he/she usually has some idea about reasonable trade-offs. This knowledge can be exploited to guide
the search towards the preferred region of solutions. In particular, the decision maker is asked to specify maximally acceptable trade-offs for each pair of objectives. For instance, in the case of two objectives, the decision maker could define that a gain of one unit in objective \( f_2 \) is worth a degradation of objective \( f_1 \) by at most \( a_{12} \) units: this corresponds to identifying a straight line on the \((f_1, f_2)\)-plane with slope \(-1/a_{12}\), leading to a weighted utility function \( \Omega_1(f_1, f_2) = f_1 + a_{12} f_2 \). Similarly, a gain in objective \( f_1 \) by one unit could be regarded worth a loss of at most \( a_{21} \) units on objective \( f_2 \): this specification identifies a straight line on the \((f_1, f_2)\)-plane with slope \(-a_{21}\), leading to a weighted utility function \( \Omega_2(f_1, f_2) = a_{21} f_1 + f_2 \). The right graph in Figure 1 shows the contour lines corresponding to the above linear functions passing through a solution \( A \) in the objective functions space. All solutions in the hatched regions are dominated by \( A \) according to the definition of domination given in (4). It is interesting to note that in the usual definition of domination (eq. (1)), the region of domination by \( A \) is marked by the horizontal and vertical lines incident on \( A \) (Figure 1, left): the original dominance criterion can thus be considered a special case of the guided dominance criterion with \( a_{12} = a_{21} = \infty \). It can be seen that the modified definition of domination allows a larger region to become dominated by any solution than the standard definition of eq. (1).

**Figure 1. Representation of the standard (left) and preference-guided (right) dominance regions in the objective functions space.**

As a further illustration, the depicted solutions \( A, B, C, D \) in Figure 2 left would be non-dominated with the ordinary dominance scheme, whereas when the preference-guided dominance scheme is adopted, solution \( A \) is dominated by solution \( B \). This shows that the Pareto optimal front found with the original domination definition of eq. (1) may not be non-dominated according to the new definition of domination (4). Figure 2 right reports the non-dominated frontier (bold solid curve) for the same values of the trade-off matrix \( \alpha \) in the two objective functions illustration of Figure 1 right: this region is bounded by the solutions where the trade-off functions are tangent to the Pareto optimal front (grey region). Thus, by choosing appropriate trade-off values, it is possible to focus on any part of the convex Pareto...
optimal front. Yet, since the approach implicitly assumes linear utility functions, it may not be possible to focus on all parts of a concave Pareto optimal front.

Also, when the guided domination principle is extended to \( n_f > 2 \) objectives, the decision maker has to specify an increasing number of trade-offs, i.e. \((n_f^j - n_f^i)/2\), and the dominance calculations become more complex.

Further, it is worth noting that the use of linear maximal and minimal trade-off boundaries might sometimes limit the applicability of the G-MOGA, namely when the DM's trade-off is strongly dependent not only on the relative positions of the two solutions to be compared, but also on their absolute positions, i.e. when the area dominated by a solution depends on the solution’s location in the objective functions space [4].

Finally, note that scaling of the objectives in a common range, e.g. [0, 1], may be performed also in this preferential optimization approach (for instance, by the methods of positive ideal solution or effective range described in the previous Section). However, differently from the WP-MOGA, in the case of G-MOGA the normalization is not mandatory due to the dimensional nature of the aggregating coefficients \( a_{ij}, i, j = 1, 2, \ldots, n_f \). In fact, the generic coefficient \( a_{ij} \) is defined as the amount of loss in the \( i \)-th objective function that the DM is willing to accept for a gain of one unit in the \( j \)-th objective function: as a consequence, though the \( i \)-th weighted utility function \( \Omega_i, i = 1, 2, \ldots, n_f \), in (4) is an aggregation of all the \( n_f \) non commensurable objectives, it has the same “unit of measurement” as the objective function \( f_i \).

It follows that the extended domination concept involving the \( \Omega_i \)'s (Figure 1, right) does not need scaling of the objectives, exactly in the same way as the standard domination relation (Figure 1, left).

![Diagram](image)

**Figure 2.** Effect of the guided dominance scheme used by the G-MOGA: solution A, which is nondominated according to the standard definition of dominance (eq. 1), becomes dominated by B according to the guided definition (eq. 4) (left); the complete Pareto optimal front resulting
from the original domination definition (thin solid curve) may not be nondominated according to the new definition of domination (right).

4. CASE STUDY 1: A TEST PROBLEM OF LITERATURE

The approaches presented in Section 3 are first tested and compared on a simple test problem introduced in [16], regarding the maximization of the following two objective functions:

\[
\begin{align*}
    f_1(x_1, x_2) &= \sin(x_1^2 + x_2^2 + 1) \\
    f_2(x_1, x_2) &= \sin(x_1^2 + x_2^2 - 1)
\end{align*}
\]

by choosing the values of the two decision variables \(x_1, x_2 \in [0, 3\pi/4]\).

Three different preferential optimizations have been performed by using both WP-MOGA and G-MOGA, with population size \(n_p = 50\), crossover probability \(p_c = 1.0\), mutation probability \(p_m = 0.02\) and number of generations before stopping the search \(n_g = 15\). The weightings \(w_i\) and trade-off coefficients \(a_{ij}\) used to drive the search of the respective algorithms towards the preferred regions of the Pareto optimal fronts are given in Table 1 for the three different cases considered.

Notice that the numerical weights for the WP-MOGA have been selected arbitrarily and such to allow to clearly show the effects of different preferential relations on the structure of the Pareto-optimal front. Obviously, in a real application the decision-maker is expected to follow a rigorous procedure for the elicitation of the weights.

Table 1. Preferences and their coding into numerical weights and linear trade-offs for Case Study 1.

<table>
<thead>
<tr>
<th>Preference</th>
<th>WP-MOGA weights</th>
<th>G-MOGA linear trade-offs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f_1) much more important than (f_2)</td>
<td>(w = (0.9, 0.1))</td>
<td>(a_{12} = 0), (a_{21} = 1.5)</td>
</tr>
<tr>
<td>(f_1 \gg f_2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(f_1) as important as (f_2)</td>
<td>(w = (0.5, 0.5))</td>
<td>(a_{12} = 0.8), (a_{21} = 0.8)</td>
</tr>
<tr>
<td>(f_1 \approx f_2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(f_1) less important than (f_2)</td>
<td>(w = (0.4, 0.6))</td>
<td>(a_{12} = 1.0), (a_{21} = 0.4)</td>
</tr>
<tr>
<td>(f_1 \prec f_2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figures 3 and 5 present the Pareto optimal fronts found by the WP- and G-MOGA in the three different cases, respectively. The WP-MOGA turns out to be quite difficult to guide and steer towards the preferred region as it tends to outline the whole Pareto front, albeit with different densities (Figure 3). To highlight this characteristic, the Pareto optimal front in the objective functions space has been divided into 10 equal sectors and the number of Pareto optimal solutions found in each sector has been counted, for different weight values. Figure 4 shows how changing the weight vector, the degree of bias towards the preferred region and the density of solutions along the different regions of the Pareto optimal front change. The Figure clearly shows that when $w_1 = w_2 = 0.5$, most solutions concentrate in the central part of the Pareto front (Figure 4, squares). On the contrary, when $w_1 = 0.1$ and $w_2 = 0.9$, more solutions cluster in the region where $f_2$ is maximal (Figure 4, triangles-up) and when $w_1 = 0.9$ and $w_2 = 0.1$, an opposite trend emerges (Figure 4, triangles-down). This may provide the DM some idea as to how the Pareto front looks like, even away from the region specified as preferred.

![Figure 3](image_url)

*Figure 3. The different regions of the Pareto optimal front found by the WP-MOGA for different preferences: $f_1 \gg f_2$ (top, left), $f_1 \approx f_2$ (top, right), $f_1 < f_2$ (bottom).*
Figure 4. Density of solutions along the Pareto optimal front for different combinations of numerical weights.

On the contrary, it is evident that the definition of maximal and minimal trade-offs allows the G-MOGA to identify a precise section of the Pareto front which turns out to be very densely covered (Figure 5).
Figure 5. The different parts of the Pareto optimal front found by the G-MOGA for different preferences: \( f_1 \gg f_2 \) (top, left), \( f_1 \approx f_2 \) (top, right), \( f_1 < f_2 \) (bottom).

5. CASE STUDY 2: UNIT COMMITMENT AND EMISSION DISPATCHING IN POWER SYSTEM GENERATION SCHEDULING

The two optimization approaches previously described have been applied on the test case T54.4 of the INGEnet – thematic European network on evolutionary algorithms applications of industrial interest (www.ingenet.ulpgc.es/functional/databases/ceani/index.html), titled “Optimal Scheduling of power generation using evolutionary algorithms”.

5.1 Problem formulation

The unit commitment problem concerns the selection of the generating units to be committed during the scheduling horizon, to minimize the system operation costs and the emission of atmospheric pollutants, in particular nitrogen oxides, \( \text{NO}_x \), while satisfying several equality and inequality constraints.

The list of symbols used for this case study is as follows [45]:

- \( T_{\text{max}} \): number of hours considered (scheduling horizon);
- \( t = 1, 2, ..., T_{\text{max}} \): hourly time index;
- \( N \): number of generating units;
- \( i = 1, 2, ..., N \): generating unit index;
- \( P_t^i \): power generated by unit \( i \) at time \( t \);
- \( P_{\text{max}}^i \): rated upper generation limit of unit \( i \);
- \( P_{\text{min}}^i \): rated lower generation limit of unit \( i \);
- \( u_t^i \): binary commitment state of unit \( i \) at time \( t \) \((= 1 \text{ if unit is committed at time } t; = 0, \text{ otherwise})\);
- \( LC_t^i \): fuel cost at time \( t \), for the actual power level \( P_t^i \); it is computed as follows: \( LC_t^i = A_i (P_t^i)^2 + B_i P_t^i + C_i \), where \( A_i, B_i \) and \( C_i \) are fuel cost coefficients related to the \( i \)-th generator;
- \( x_t^i \): time duration during which the unit \( i \) has been off, at hour \( t \);
- \( S_t(x_t^i) \): start-up cost of the unit \( i \) at hour \( t \); it is computed as a function of the number of hours that the unit has been down up to hour \( t \), \( S_t(x_t^i) = a_i (1 - e^{-c x_t^i}) + b_i \); \( a_i, b_i \), and \( c_i \) are the start-up cost coefficients related to unit \( i \);
- \( D \): shut-down cost of unit \( i \);
- \( L \): load demand at time \( t \);
- \( R_t \): system spinning reserve requirement at time \( t \);
- \( T_t^{\text{on/off}} \): minimum up/down time of unit \( i \);
\( (NO_x)_t' \): mg/Nm\(^3\) of NO\(_x\) produced by unit \( i \) at time \( t \); it is computed as \( (NO_x)_t' = D_i (P_t')^2 + E_i P_t' + F_i \), where \( D_i \), \( E_i \), and \( F_i \) are the emission coefficients related to the \( i \)-th generator.

In the problem considered, the number of generating units \( N \) is set to 10, the time horizon period \( T_{\text{max}} \) is 24 hours and the spinning reserve \( R_t \) is 10\% of \( L_t \), \( t = 1, 2, \ldots, T_{\text{max}} \). The data related to each generating unit and to the load demand are given in Appendix B. More details about this case study may be found in [45].

The first objective of the optimization is to minimize the \( N \)-units system operation cost \( f_1 \), in arbitrary monetary units [m.u.], which includes the fuel cost for generating power, the start up cost and the shut down cost, over the entire time horizon:

\[
f_1 = \sum_{i=1}^{N} \sum_{t=1}^{T_{\text{max}}} \left[ u_t' \cdot LC_x + u_t' \cdot (1-u_t'^{\text{on}}) \cdot S_t(x_t') + (1-u_t') \cdot D_t \right], \text{[m.u.]}.
\] (7)

The second objective of the problem is to minimize the quantity of nitrogen oxides NO\(_x\) released in the atmosphere, \( f_2 \):

\[
f_2 = \sum_{i=1}^{N} \sum_{t=1}^{T_{\text{max}}} (NO_x)_t', \text{[mg/Nm}\(^3\)].
\] (8)

The constraints for the optimization are:

1. System power balance: the total power generation at time \( t \) must cover the total demand \( L_t' \). Hence,

\[
\sum_{i=1}^{N} P_t' = L_t', t = 1, 2, \ldots, T_{\text{max}}.
\] (9)

2. System spinning reserve requirements: a reserve is necessary to face in real time possible sudden load increases due to a demand increase or to a failure of any of the working units. Hence,

\[
\sum_{i=1}^{N} u_t' P_{\text{max}_t} \geq L_t' + R_t, t = 1, 2, \ldots, T_{\text{max}}.
\] (10)

3. Unit minimum up/down times:

Minimum up time: \( \sum_{i=1}^{N} u_t' \geq T_t'^{\text{on}}, t = 1, 2, \ldots, N \).

Minimum down time: \( T_{\text{max}} - \sum_{i=1}^{N} u_t' \geq T_t'^{\text{off}}, t = 1, 2, \ldots, N \).
4. Unit generation limits: for stable operation, the power output of each generator is restricted by lower and upper limits as follows:

\[ P_{\text{min}} \leq P'_i \leq P_{\text{max}}, i = 1, 2, ..., N. \]  

(13)

Notice that in a basic unit commitment problem, when a unit starts up, its generating capability is assumed to increase immediately from zero to \( P_{\text{max}} \); likewise, when a unit shuts down, its generating capability jumps from \( P_{\text{max}} \) to zero.

5.2 Application of the MOGA approaches

The configuration of the chromosome for the simple unit commitment problem described is quite straightforward (Figure 6): the chromosome is subdivided into \( T_{\text{max}} \) genes (one for each hour of the scheduling horizon), each containing \( N \) bits (one for each generating unit). If the \( i \)-th bit of the \( t \)-th gene is 1, then \( u'_i \), the commitment state of unit \( i \) at time \( t \), is 1, i.e. the unit \( i \) is working at time \( t \); viceversa, if the bit is 0. Since \( N = 10 \) and \( T_{\text{max}} = 24 \) hours, each individual chromosome contains 240 bits; thus, the genetic search has to be conducted in a \( 2^{240} \)-dimensional space. Note that contrary to other GA applications, in this case of unit commitment the binary chromosome does not encode real-valued control factors: the information regarding the state of the units is included in the bits themselves so that no decoding is necessary.

![Figure 6. Configuration of the chromosome for the unit commitment problem.](image)

\( u'_i = 0 \), unit 1 is not working at hour 1

\( u'_i = 1 \), unit 1 is working at hour 2

The different MOGA search schemes applied to this Case Study are summarized in Table 2.

<table>
<thead>
<tr>
<th>Configuration number</th>
<th>Search algorithm</th>
<th>Population size, ( n_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard MOGA</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>Standard MOGA</td>
<td>600</td>
</tr>
</tbody>
</table>
For a fair comparison, all simulations have been run for $n_g = 1000$ generations with crossover probability $p_c = 1.0$ and mutation probability $p_m = 0.0042$, calculated as the inverse of the number of bits constituting a chromosome, according to [7], [34] and [39]. Moreover, in all the different genetic search schemes summarized in Table 2, at each generation two parents are uniformly *randomly* sampled with replacement among the entire sorted population; then, the parents are crossed to obtain two children; finally, out of the four individuals (two parents and two children) involved in the crossover procedure, the *fittest* two (i.e., those with the best rank) replace the parents.

For the implementation of the constraints (9)-(13), the approach adopted in this paper is simply to restrict the search within the feasible region [1]. To this aim, a procedure has been implemented to check the feasibility, with respect to the constraints, of the chromosomes of the initial population and of those generated as children during the GA search operations: those chromosomes which do not respect the assigned constraints are discarded from the population and the algorithm proceeds to create new ones. This ensures the feasibility of the nondominated solutions.

Table 3 shows the weights and the maximum and minimum trade-offs arbitrarily chosen to drive the WP-MOGA and G-MOGA searches towards the preferred regions of the Pareto optimal front, respectively.
Table 3. Preferences and their coding into numerical weights and linear trade-offs for Case Study 2.

<table>
<thead>
<tr>
<th>Preference</th>
<th>WP-MOGA weights</th>
<th>G-MOGA linear trade-offs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$ much more important than $f_2$</td>
<td>$w = (0.9, 0.1)$</td>
<td>$a_{12} = 0$, $a_{21} = 4$</td>
</tr>
<tr>
<td>$f_1 &gt;&gt; f_2$</td>
<td></td>
<td>$a_{12} = 0.25$, $a_{21} = 15$</td>
</tr>
<tr>
<td>$f_1$ as important as $f_2$</td>
<td>$w = (0.5, 0.5)$</td>
<td></td>
</tr>
<tr>
<td>$f_1 = f_2$</td>
<td></td>
<td>$a_{12} = 0.05$, $a_{21} = 0$</td>
</tr>
<tr>
<td>$f_1$ much less important than $f_2$</td>
<td>$w = (0.1, 0.9)$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7 depicts the Pareto optimal fronts obtained for each simulation run.

![Pareto optimal fronts obtained for different MOGA configurations](image)

Figure 7. Representation in the objective functions space of the Pareto optimal frontiers for Configurations 1, 2, 3, 4 of the MOGA.

It can be seen that, in general, the articulation of a preference, which drives the search towards interesting regions of the Pareto optimal front, produces a faster convergence of the
search algorithm. In fact, by increasing the population size from 200 (Configuration 1, circles in Figure 7) to 600 (Configuration 2, dots in Figure 7), i.e. by improving the genetic diversity, the standard MOGA succeeds in finding Pareto solutions comparable to those of the G-MOGA (Configuration 4, triangles in Figure 7) at least with respect to the second objective function, \( f_2 \); on the other hand, the increased population size almost triplicates the computational time necessary to obtain the convergence of the search algorithm.

Further, both the WP-MOGA (Configurations 3) and the G-MOGA (Configurations 4) perform much better than the standard MOGA, run with the same population size and for the same number of generations (Configuration 1), with respect to the quality of the Pareto solutions obtained.

Finally, the fine guidance determined by the specification of maximum and minimum linear trade-off functions allows the G-MOGA to explore the preferred regions of the high-dimensional search space effectively also in terms of diversity and number of individuals found: the G-MOGA succeeds in reaching extreme portions of the Pareto optimal front (Configurations 4, \( f_1 >> f_2 \) and \( f_1 << f_2 \), squares and triangles in Figure 7, respectively) which are not found by the standard MOGA and the WP-MOGA.

5.3 Effect of the MOGA parameters on the structure of the Pareto-optimal front

In general, the final structure of the Pareto-optimal front achieved by a MOGA search may strongly depend on the parameters of the MOGA. In what follows, a general discussion of this issue is provided, based on both literature findings and previous experimentation performed by the authors [32], [46].

The parameters and procedures which are recognized to most affect the performance of a MOGA search (and of the structure and quality of the Pareto-optimal front thereby identified) are the selection and replacement procedures, the probability of mutation, \( p_m \), and the population size, \( n_p \) [32].

Many ways exist to perform the selection step, e.g.: i) standard roulette selection, in which the probability of choosing an individual as parent is proportional to its rank; ii) random selection (adopted in this work), in which the parents are uniformly (i.e. regardless of their rank) randomly selected with replacement, among the entire sorted population; iii) fit-fit selection, in which the ordered population is scanned and each individual is parent-paired with the next fittest one; iv) fit-weak selection, in which, as in the preceding case, the ranked population is scanned but this time each individual is parent-paired with the one symmetrically positioned with respect to the mid of the ordered rank list [32].
Similarly, different procedures of replacement are: i) fittest individuals (adopted in this work), in which out of the four individuals (two parents and two chromosomes) involved in the crossover procedure, the fittest two (i.e., those with the best rank) replace the parents; ii) children-parents, in which the children simply replace the parents; iii) weakest individuals, in which the children replace the two weakest individuals in the entire population, parents included, provided that the children rank is higher; iv) random replacement, in which the children replace two individuals randomly chosen in the entire population, parents included [32].

From the results of experimentation performed by the authors in the present paper and in previous works, general considerations can be drawn about the effect of different genetic operators on the convergence of the MOGA population towards the Pareto front and on its structure [46]:

- **Probability of mutation, \( p_m \):** low probabilities of mutation (e.g., less than 0.001) favour the uniform convergence of the MOGA population towards the optimal Pareto front; higher values of \( p_m \) have a negative impact when combined with fitness-ignoring replacement procedures (e.g., children-parents and random), reducing the extension and quality of the Pareto front; on the contrary, in strongly fitness-oriented searches (e.g., fittest and weakest replacement procedures), larger values of \( p_m \) positively improve genetic diversity, favouring a deep exploration of the search space and producing a wide Pareto optimal front;

- **Replacement:** fitness-guided replacement procedures (e.g., fittest or weakest) efficiently move the population towards solutions on the optimal Pareto front; on the other hand, when they are combined with fitness-guided selection methods (e.g. standard roulette or fit-fit), genetic diversity must be ensured by means of a relatively high mutation probability to avoid entrapment in local optima and to obtain an extended Pareto front;

- **Selection:** selection is less determining than replacement; however, it is interesting to observe that the combination of a uniform sampling of the population (random selection) with a fitness-guided replacement procedure (e.g. fittest) helps improving genetic diversity and finding a wide optimal Pareto front (that is the reason why this configuration has been adopted in this work);

- **Population size, \( n_p \):** small population sizes reduce the computational cost, but also genetic diversity, thus limiting the exploration of the search space and reducing the quality of the Pareto solutions; on the contrary, large populations, though
computationally expensive, positively improve genetic diversity producing wide Pareto optimal fronts (MOGA Configuration 1 vs 2 in Section 5.2).

5.4 Choosing a compromise solution from the biased Pareto-optimal front

Once a set of (biased) Pareto optimal solutions is obtained, higher-level decision making considerations must come into play to choose a solution among those identified. The following methods can be adopted [33].

5.4.1 “Min-max” method

A popular criterion for choosing a “single best compromise solution” is the so called “min-max method” [2]. Let \( f = \{f_1, f_2, \ldots, f_n\} \) denote a generic point on the \( n_f \)-dimensional Pareto surface, and \( f_{i,\text{max}} \), \( i = 1, 2, \ldots, n_f \), the maximum value of the \( i \)-th objective function on such surface. For each point \( f \) we calculate the relative deviations \( z_i = (f_{i,\text{max}} - f_i)/f_{i,\text{max}} \), \( i = 1, 2, \ldots, n_f \) and take as a representative value \( z_f = \min_i(z_i) \). By definition, the best compromise solution is the point \( f^* \) on the Pareto surface corresponding to the maximum \( z_f \). In other words, the chosen compromise solution is the one for which the minimal relative deviation from the extremes of the optimal Pareto front is largest: thus, the user should adopt this method when he/she desires a solution that is representative of the “center” of the Pareto-optimal frontier. Figure 8 shows the chosen solution in the Pareto-optimal set (circles) for a two-objective minimization problem.

![Figure 8. Best compromise solution \( f^* \) chosen from the Pareto-optimal front (circles) according to the “min-max” method for a two-objective minimization problem](image)

5.4.2 Marginal rate of return method

The marginal rate of return is defined as the amount of improvement in one objective function which can be obtained by sacrificing a unit decrement in performance in any other objective function [33]. The solution having the maximal marginal rate of return is the one chosen by this method. Since pair-wise comparison have to be made with all \( n_f \) objectives and for each
Pareto-optimal solution, this method may be computationally expensive. Figure 9 shows the preferred “knee” point $f^*$, where the marginal rate of return is maximum among a set of obtained Pareto-optimal solutions (circles) with reference to a two-objective minimization problem.

![Figure 9. Best compromise solution $f^*$ chosen from the Pareto-optimal front (circles) according to the marginal rate of return method for a two-objective minimization problem](image)

5.4.3 Weighted average method

A simple strategy would be to choose a solution closer to the optimum solution corresponding to a particular user specified weighted average of the objective functions [33]. Figure 10 shows the chosen solution $f^*$ with the weight-vector $w = (w_1, w_2) = (0.85, 0.15)$ for a two-objective minimization problem. This procedure is different from the classical weighted average scheme in that here a solution is chosen after many Pareto-optimal solutions have been found. In the classical weighted average scheme, only one solution optimizing the weighted average of the objective would be found. A different solution can be found with a different weight vector.

![Figure 10. Best compromise solution $f^*$ chosen from the Pareto-optimal front (circles) according to the weighted average method with weight-vector $w = (w_1, w_2) = (0.85, 0.15)$ for a two-objective minimization problem](image)
5.4.4 Application

The three approaches above described have been applied to choose a single solution among those contained in each one of the biased Pareto optimal frontiers of Figure 7 (Configurations 3 and 4 of the MOGA); the results are shown in Table 4; notice that the weight vectors adopted in the weighted average method are $w = (0.9, 0.1)$, $w = (0.5, 0.5)$ and $w = (0.1, 0.9)$ for the preference relations $f_1 >> f_2$, $f_1 \approx f_2$ and $f_1 << f_2$, respectively, as in Table 3.

**Table 4. Compromise solutions chosen from the biased Pareto optimal frontiers of Figure 7 (Configurations 3 and 4 of the MOGA) according to the “min-max”, marginal rate of return and weighted average methods**

<table>
<thead>
<tr>
<th>Configuration number</th>
<th>Min-Max</th>
<th>Marginal rate of return</th>
<th>Weighted average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fuel (x 10000) [m.u.]</td>
<td>NOx (x 10000) [mg/Nm³]</td>
<td>Fuel (x 10000) [m.u.]</td>
</tr>
<tr>
<td>3, $f_1 &gt;&gt; f_2$</td>
<td>79.894</td>
<td>50.278</td>
<td>79.894</td>
</tr>
<tr>
<td>3, $f_1 &lt;&lt; f_2$</td>
<td>80.672</td>
<td>47.696</td>
<td>81.208</td>
</tr>
<tr>
<td>4, $f_1 &gt;&gt; f_2$</td>
<td>77.875</td>
<td>56.502</td>
<td>78.221</td>
</tr>
<tr>
<td>4, $f_1 \approx f_2$</td>
<td>78.796</td>
<td>50.189</td>
<td>78.897</td>
</tr>
<tr>
<td>4, $f_1 &lt;&lt; f_2$</td>
<td>80.785</td>
<td>46.393</td>
<td>80.997</td>
</tr>
</tbody>
</table>

It is worth underlying that the solutions selected by the three a posteriori techniques are not comparable, since the concepts motivating the selection are completely different in each of these methods: the “min-max” method typically finds solutions that are representative of the “center” of the Pareto-optimal frontier (for instance, see Configuration 4, $f_1 << f_2$ in Table 4 and Figure 7); the marginal rate of return method typically identifies solutions located at “knee” points of the Pareto frontier (for instance, see Configuration 3, $f_1 << f_2$ and $f_1 >> f_2$ in Table 4 and Figure 7); finally, the weighted average method chooses the Pareto-optimal solution which is closer to an ideal user-specified weighted average of the objective functions: in other words, it finds the solution that best represents the portion of Pareto front preferred by the decision-maker (for instance, see Configuration 4, $f_1 << f_2$, $f_1 \approx f_2$ and $f_1 >> f_2$ in Table 4 and Figure 7).
6. CONCLUSIONS

In this paper, the problem of optimizing the operation of power plants has been considered with respect to the minimization of both the costs of power generation and the atmospheric emissions. In realistic cases, the problem is a combinatorial multi-objective optimization problem which can be effectively tackled by evolutionary algorithms, such as the genetic algorithms, within a Pareto optimality scheme of search of the nondominated solutions.

To render the optimization search more efficient, any information regarding the preference values of the decision maker with respect to the objectives must be exploited to focus the search on the region of preference of the Pareto optimal set of nondominated solutions. Following an experimental survey of some recent approaches to this problem, two methods have been considered and implemented to include the DM’s preferences over the different objectives into the optimization process of a MOGA.

In the WP-MOGA, the concept of Pareto dominance has been integrated in the replacement step of the algorithm with a biasing action enforced through a priori fixed weights encoding the DM’s preferences.

In the G-MOGA, the search has been finely guided by introducing minimum and maximum acceptable trade-offs explicitly set by the DM.

These approaches have been tested and compared on a simple case study of literature and then applied on a real-world, high-dimensional, nonlinear, constrained two-objective power system generation scheduling problem.

The results obtained show that both methods outperform the standard MOGA because by focusing the search onto the relevant part of the search space they are capable of yielding a more finely-grained and suitable selection of alternatives, while also speeding up convergence and, thus, saving computational time.

The G-MOGA turns out to be more promising than the WP-MOGA for at least two reasons: first, reasonable trade-offs between different objectives may be easier to specify than exact weights; second, the guided dominance scheme allows a more precise control of the focus on the region of interest and, thus, a faster convergence. Actually, the WP-MOGA tends to still outline the whole Pareto optimal front, only with different densities of solutions, thus still providing a wide range of solutions, many of them possibly of little relevance to the DM.

On the other hand, the G-MOGA is expected to encounter practical difficulties when more than two objective functions have to be considered, as the required number of trade-offs to be specified increases dramatically with the number of objectives. Further work is needed to verify this issue on power system generation scheduling problems involving more than two
objectives and possibly further research is required to address the issue in a more effective way, in general.

Another issue worth of further consideration concerns whether the guided dominance scheme may be used during the optimization process to identify possible inconsistencies in the preferences of the decision-maker.

Finally, to facilitate the preferential guiding of the MOGA search, two interesting possibilities may be explored in future research: a method similar to the one presented in [13] could support the specification of minimum and maximum trade-off functions and the guided domination scheme could be combined with an approach similar to that given in [41] that ‘learns’ the DM’s preferences during the optimization run.

7. REFERENCES


8. APPENDIX A: TRANSLATING LINGUISTIC PREFERENCES INTO NUMERICAL WEIGHTS

The method presented in [13] to translate fuzzy (linguistic) preferences into specific quantitative weights is hereafter summarized.

Letting \( f = \{f_1, f_2, \ldots, f_n\} \) be the set of conflicting objectives, the algorithm for computing the corresponding preference weight-vector \( w = \{w_1, w_2, \ldots, w_n\} \) is as follows:

1. For every pair of objectives specify one of the following characterizations: a) less important (\(<\)); b) much less important (\(<<\)); c) equally important (\(\approx\)); d) more important (\(>\)); e) much more important (\(>>\)); f) don’t care (\(\neg\)).

2. Use the following valuation \( v \) [valuation is a method of assigning semantic values (in this case, numbers) to syntactic categories]:

   - If \( a << b \) then \( v(a) = \alpha \) and \( v(b) = \beta \);
   - If \( a < b \) then \( v(a) = \gamma \) and \( v(b) = \delta \); \hspace{1cm} (A1)
   - If \( a \approx b \) then \( v(a) = v(b) = 1/2 \).

where \( \alpha, \beta, \gamma \) and \( \delta \) are arbitrary [0, 1] real numbers.

Notice that taking into account the intended meaning of these relations, it can be assumed that \( \alpha < \gamma < 1/2 < \delta < \beta \).
It is also assumed that $\alpha + \beta = \gamma + \delta = 1$ (so called probabilistic relation): as a consequence, the decision-maker has to define only two of these parameters (e.g., $\alpha$ and $\gamma$), thus reducing the arbitrary nature of the valuation assignments (A1). Further, it has been demonstrated that the influence of parameters $\alpha$ and $\gamma$ on the values of the numerical weights drastically decreases as the number of conflicting objectives increases [15].

3. Initialize two matrices $R$ (fuzzy preference matrix) and $R_a$ (relative preference matrix) of size $n_f \times n_f$ to the identity matrix. They will be used as follows:

$$f_i << f_j \Leftrightarrow R(i, j) = \alpha, R(j,i) = \beta \Leftrightarrow R_a(i, j) = 0, R_a(j,i) = 2$$

$$f_i < f_j \Leftrightarrow R(i, j) = \gamma, R(j,i) = \delta \Leftrightarrow R_a(i, j) = 0, R_a(j,i) = 1$$

$$f_i = f_j \Leftrightarrow R(i, j) = 1/2, R(j,i) = 1/2 \Leftrightarrow R_a(i, j) = 1, R_a(j,i) = 1$$

(A2)

This evaluation gives the idea of how to generalize preferences to have $s$ levels instead of only five (from “much less important” to “much more important”). If $f_i$ is $s^*$ levels more important than $f_j$, the assignments are $R_a(i, j) = s^*$ and $R_a(i, j) = 0$.

4. Perform the following steps:

Step 1) For all $i \leq n_f$ and for all $j \leq n_f$ such that $j \neq i$ do:

Step 1a) If $R_a(i, j) + R_a(j,i) = 0$, then:

- ask whether $f_i << f_j, f_i < f_j, f_i << f_j$, or $f_j < f_i$;

- using equations (A2), set $R_a(i, j)$ and $R_a(i, j)$ accordingly.

Step 1b) Compute the transitive closure of $R_a$:

for $k = 1, 2, \ldots, n_f$

for $j = 1, 2, \ldots, n_f$

for $i = 1, 2, \ldots, n_f$

$$R_a(i, j) = \min(2, \max(\{R_a(i, j), R_a(i, k) \cdot R_a(k, j)\}))$$

(A3)

The rationale behind this formula is the following: if either $R_a(i, j)$ or $R_a(k, j)$ is 0 (no path between $i$ and $k$ or between $k$ and $j$) then $R_a(i, j)$ does not change; the term $\min(2, \cdot)$ is used so that $R_a(i, j) \in \{0, 1, 2\}$. Notice that Step 1b) is very
important as it protects against contradictions in the sense that if, e.g., the preferences \( i \succ j \) and \( j \approx k \) are already specified, the algorithm automatically infers that \( i \succ k \) and skips the \( i?k \) question preventing eventual contradictions.

Step 2) Using (A2), calculate matrix \( R \) from \( R_a \) as follows:

- If \( R_a(i, j) = 1 \) and \( R_a(j, i) = 1 \), set \( R(i, j) = 1/2 \) and \( R(j, i) = 1/2 \).
- If \( R_a(i, j) = 0 \) and \( R_a(j, i) = 1 \), set \( R(i, j) = \gamma \) and \( R(j, i) = \delta \).
- If \( R_a(i, j) = 1 \) and \( R_a(j, i) = 0 \), set \( R(i, j) = \delta \) and \( R(j, i) = \gamma \).
- If \( R_a(i, j) = 0 \) and \( R_a(j, i) = 2 \), set \( R(i, j) = \alpha \) and \( R(j, i) = \beta \).
- If \( R_a(i, j) = 2 \) and \( R_a(j, i) = 0 \), set \( R(i, j) = \beta \) and \( R(j, i) = \alpha \).

Step 3) For each \( f_i, i = 1, 2, \ldots, n_f \), compute the leaving score, \( S_L(f_i, R) \) as

\[
S_L(f_i, R) = \sum_{j=1}^{n_f} R(i, j). \quad (A4)
\]

Step 4) For each \( f_i, i = 1, 2, \ldots, n_f \), compute the weight as a normalized leaving score:

\[
w_i = \frac{S_L(f_i, R)}{\sum_{j=1}^{n_f} S_L(f_j, R)}. \quad (A5)
\]

9. **APPENDIX B: DATA RELATIVE TO THE GENERATING UNITS OBJECT OF THE CASE STUDY OF SECTION 5 [45].**

Table B1. Data for the evaluation of fuel costs, start up costs and shut down costs in monetary units (m. u.).

<table>
<thead>
<tr>
<th>Unit</th>
<th>( P_{max} ) [MW]</th>
<th>( T_{on} ) [h]</th>
<th>( T_{off} ) [h]</th>
<th>( I_s ) [h]</th>
<th>( a_i ) [m.u.]</th>
<th>( b_i ) [m.u.]</th>
<th>( c_i ) [m.u.]</th>
<th>( D_i ) [m.u.]</th>
<th>( A_i ) [m.u./MW^2]</th>
<th>( B_i ) [m.u./MW]</th>
<th>( C_i ) [m.u.]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>3</td>
<td>1</td>
<td>-1</td>
<td>85</td>
<td>20.59</td>
<td>0.20</td>
<td>15</td>
<td>0.0454</td>
<td>9.9214</td>
<td>159.33</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>3</td>
<td>1</td>
<td>-1</td>
<td>101</td>
<td>20.59</td>
<td>0.20</td>
<td>25</td>
<td>0.0356</td>
<td>10.375</td>
<td>222.16</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>114</td>
<td>22.57</td>
<td>0.20</td>
<td>40</td>
<td>0.0254</td>
<td>15.031</td>
<td>262.87</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>94</td>
<td>10.65</td>
<td>0.18</td>
<td>32</td>
<td>0.0212</td>
<td>15.031</td>
<td>315.44</td>
</tr>
<tr>
<td>5</td>
<td>150</td>
<td>5</td>
<td>3</td>
<td>-7</td>
<td>113</td>
<td>18.64</td>
<td>0.18</td>
<td>29</td>
<td>0.0106</td>
<td>20.765</td>
<td>485.69</td>
</tr>
<tr>
<td>6</td>
<td>280</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>176</td>
<td>27.57</td>
<td>0.15</td>
<td>42</td>
<td>0.0253</td>
<td>18.995</td>
<td>1236.9</td>
</tr>
<tr>
<td>7</td>
<td>520</td>
<td>8</td>
<td>4</td>
<td>-5</td>
<td>267</td>
<td>34.75</td>
<td>0.09</td>
<td>75</td>
<td>0.0085</td>
<td>19.566</td>
<td>4437.2</td>
</tr>
<tr>
<td>8</td>
<td>150</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>282</td>
<td>45.48</td>
<td>0.09</td>
<td>49</td>
<td>0.0116</td>
<td>22.251</td>
<td>300.86</td>
</tr>
</tbody>
</table>
Table B2. Coefficients for the evaluation of NO\textsubscript{x} emissions.

<table>
<thead>
<tr>
<th>Unit</th>
<th>(D_i \text{ [mg/Nm}^3\text{/MW]})</th>
<th>(E_i \text{ [mg/Nm}^3\text{/MW]})</th>
<th>(F_i \text{ [mg/Nm}^3\text{]})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2.286</td>
<td>235.92</td>
<td>-1890</td>
</tr>
<tr>
<td>2</td>
<td>-1.793</td>
<td>246.71</td>
<td>-2636</td>
</tr>
<tr>
<td>3</td>
<td>0.0193</td>
<td>1.7774</td>
<td>230.49</td>
</tr>
<tr>
<td>4</td>
<td>0.016</td>
<td>1.7774</td>
<td>276.59</td>
</tr>
<tr>
<td>5</td>
<td>-0.005</td>
<td>16.287</td>
<td>-321.4</td>
</tr>
<tr>
<td>6</td>
<td>-0.069</td>
<td>36.931</td>
<td>-1626</td>
</tr>
<tr>
<td>7</td>
<td>-0.245</td>
<td>154.16</td>
<td>-1154.6</td>
</tr>
<tr>
<td>8</td>
<td>0.1686</td>
<td>-20.0</td>
<td>1361.8</td>
</tr>
<tr>
<td>9</td>
<td>-0.002</td>
<td>16.414</td>
<td>-691.1</td>
</tr>
<tr>
<td>10</td>
<td>0.1313</td>
<td>-20.77</td>
<td>1885.6</td>
</tr>
</tbody>
</table>

Table B3. Electrical hourly load demand, \(L_t\) (MW).

<table>
<thead>
<tr>
<th>Hour</th>
<th>Load demand [MW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1459</td>
</tr>
<tr>
<td>2</td>
<td>1372</td>
</tr>
<tr>
<td>3</td>
<td>1299</td>
</tr>
<tr>
<td>4</td>
<td>1280</td>
</tr>
<tr>
<td>5</td>
<td>1271</td>
</tr>
<tr>
<td>6</td>
<td>1314</td>
</tr>
<tr>
<td>7</td>
<td>1372</td>
</tr>
<tr>
<td>8</td>
<td>1314</td>
</tr>
<tr>
<td>9</td>
<td>1271</td>
</tr>
<tr>
<td>10</td>
<td>1242</td>
</tr>
<tr>
<td>11</td>
<td>1197</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>12</td>
<td>1182</td>
</tr>
<tr>
<td>13</td>
<td>1154</td>
</tr>
<tr>
<td>14</td>
<td>1138</td>
</tr>
<tr>
<td>15</td>
<td>1124</td>
</tr>
<tr>
<td>16</td>
<td>1095</td>
</tr>
<tr>
<td>17</td>
<td>1066</td>
</tr>
<tr>
<td>18</td>
<td>1037</td>
</tr>
<tr>
<td>19</td>
<td>993</td>
</tr>
<tr>
<td>20</td>
<td>978</td>
</tr>
<tr>
<td>21</td>
<td>963</td>
</tr>
<tr>
<td>22</td>
<td>1022</td>
</tr>
<tr>
<td>23</td>
<td>1081</td>
</tr>
<tr>
<td>24</td>
<td>1459</td>
</tr>
</tbody>
</table>