Short communication

A niched genetic algorithm to solve a pollutant emission reduction problem in the manufacturing industry: A case study

L. Grandinetti∗, F. Guerriero, G. Lepera, M. Mancini

Dipartimento di Elettronica, Informatica e Sistemistica Università della Calabria, Rende, Italy

Available online 29 March 2006

Abstract

A multiobjective optimization approach to deal with a pollutant emission reduction problem in the manufacturing industry, through implementation of the best available technical options, is presented in this paper. More specifically, attention is focused on the industrial painting of wood and the problem under investigation is formulated as a bicriteria combinatorial optimization problem. A niched Pareto genetic algorithm based approach is used to determine sets of methods, tools and technologies, applicable both in the design and in the production phase, allowing to simultaneously minimize the total cost and maximize the total pollutant emission reduction.

Keywords: Genetic algorithms; Fitness sharing; Best available options; Multiple criteria analysis

1. Introduction

In the last few decades, environmental problems have received much attention from society and governments. In this context, the European Union has introduced the concept of Best Available Technologies (BAT, for short) as a new integrated approach to deal with environmental problems caused by pollution produced by manufacturing/production industries.

This approach highlights the requirement that the development of new manufacturing/production processes must be balanced by the implementation of best available technical options, that eliminate or at least reduce the amount of pollutants released into the environment.

The BATs are all the methods, tools and technologies that can be applied in the design and production phase, in order to achieve a good reduction of emission level with an acceptable additional cost.

∗ Corresponding author. (L. Grandinetti).
Often, different types of technical options can control a given source. In this context, mathematical programming techniques can be used to select the best combination of removal and control technologies from different alternatives available.

Depending on the specific goal to be achieved, different optimization problems can be formulated [1,2]. Generally, the main objective is to minimize the total cost of policy decisions (i.e., mainly the costs involved in implementation and maintenance of removal and control technologies), whereas, environmental considerations are addressed through integrating emission constraints. Other modelling approaches handle the minimization of the emission as another objective function in addition to the cost. The resulting multiobjective optimization problem is then converted to a single objective problem by a linear combination of the different objectives as a weighted sum.

In this paper, we consider a multiobjective combinatorial linear programming model that handles pollution emissions and control costs simultaneously as competing objectives. The main aim is to determine the Pareto set of combinations of BATs that allow both to maximize emission reductions and to minimize the total control cost.

For the solution of the problem under investigation, a niched Pareto genetic algorithm based approach is defined and implemented. The effectiveness of the developed algorithm is evaluated by using an illustrative case study, that is, industrial painting of wood.

The remainder of the paper is organized as follows. Section 2 gives the general mathematical formulation of the problem we are dealing with. Section 3 presents some basic concepts used in multiobjective evolutionary approaches and gives a general description of the niched Pareto genetic algorithm. Section 4 describes the considered manufacturing process. Section 5 introduces the niched Pareto genetic algorithm specifically designed for the industrial painting of wood. Section 6 reports the experimental results obtained on a real case study. Finally, Section 7 presents the conclusions of this work.

2. Problem statement

In order to describe the problem under investigation, we focus our attention on generic manufacturing plants that can be viewed as decomposed in different production phases, consuming some inputs in order to produce some outputs.

The scheme of a generic manufacturing plan is outlined in Fig. 1, where the production phases are modelled as black boxes, whereas the raw materials represent the inputs for the entire process.

During the execution of successive production phases, additional materials are consumed, pollutions are generated and production costs are incurred.

For each production phase, different candidate sets of control options (i.e., best available technologies) are available. To each option is associated a cost and an emission reduction capability. Thus, the BAT application determines a reconfiguration of energy consumptions, pollutions and costs.

In this context, a key task is selecting the BATs to be applied in each production phase, so that a specific goal is achieved. In what follows, we assume that the main aim is to choose the combination of BATs such that the best possible balance between control cost and pollution emission reduction is obtained.

It is worth observing that a variety of technology interrelationships can exist and these relations should be taken into account in BAT selection. The exact nature of these relations can vary from problem
to problem and it is beyond the scope of this paper to give a complete accounting of every possible relationship between technologies.

However, the most common interdependencies among technologies encountered in practical applications are logical relationships (i.e., if-then-else type relations) [3]. For the sake of simplicity, in what follows we assume that, the only relationship that exists between two technologies, applied either to the same phase or to different production phases, is a two-way exclusivity logical relationship. Indeed, if one technology is used, a second cannot be included and vice-versa. In order to establish and formalize physical compatibility rules and to eliminate combinations that are not physically realizable a compatibility matrix (\(CM\), for short) is used. The generic element \(C_{Mij}\) of \(CM\), is defined as follows:

\[
C_{Mij} = \begin{cases} 
1 & \text{if technology } i \text{ is incompatible with technology } j; \\
0 & \text{if technologies } i \text{ and } j \text{ are independent.} 
\end{cases}
\]

By using the matrix \(CM\) it is possible to define the set of corresponding compatibility constrained to be added to the problem.

On the basis of the previous considerations, the problem under investigation can be mathematically represented as a bicriteria combinatorial optimization problem as follows.

Let \(p\) be the production phase and \(\mathcal{P}\) the set of production phases; \(t\) the technical option, \(\mathcal{T}\) the set of BAT, \(\mathcal{T}_p\) the set of BATs that can be used on the production phase \(p\); \(k\) the single pollutant and \(\mathcal{K}_p\) the set of pollutants generated from production phase \(p\); \(x_{pt}\) a binary decision variable indicating whether technical option \(t\) is used on production phase \(p\) \((x_{pt} = 1)\) or not \((x_{pt} = 0)\); \(C_t\) the cost of technical option \(t\), \(\forall t \in \mathcal{T}\); \(r_{pkt}\) the total reduction of pollutant \(k\) in the production phase \(p\) when control \(i\) is used on \(p\).
Assuming that for each production phase at least one technical option has to be selected, the considered optimization problem may be stated as follows:

\[
\begin{align*}
\text{Min} & \sum_{p \in P} \sum_{t \in T_p} C_t x_{pt}, \\
\text{Max} & \sum_{p \in P} \sum_{k \in x_p} \sum_{t \in T_p} r_{pkt} x_{pt} \\
\text{s.t.} & \sum_{t \in T_p} x_{pt} \geq 1, \quad \forall p \in P; \\
& \mathcal{M}_{ij} (x_{pi} + x_{pj}) \leq 1, \quad \forall i, j \in T, i \neq j, \forall p, \tilde{p} \in P | i \in T_p, \quad j \in T_{\tilde{p}}; \\
& x_{pt} \in \{0, 1\}, \quad \forall p \in \mathcal{P}, \quad t \in T_p.
\end{align*}
\]

The objective function (1) expresses minimisation of the total cost, whereas the objective function (2) represents maximization of the total emission reduction; constraints (3) ensure that, for each production phase, at least one technical option is selected. Constraints (4) guarantee that the set of selected technical options are compatible with physical engineering constraints. Finally, (5) represents the binary constraints.

It is worth observing that, depending on the specific application at hand, different variants of the problem can be considered. For example, in the case in which all the available technologies are independent, the constraints (4) are redundant and thus can be eliminated, consequently the problem can be decomposed in several sub-problems, one for each phase.

In addition, if no assumption is made about the number of technical options to be selected for each phase, the constraints (3) can be eliminated, whereas if exactly either one or \(|p|\) different technical options have to be selected for each phase the constraints (3) should be replaced with \(\sum_{t \in T_p} x_{pt} = 1, \quad \forall p \in \mathcal{P}\) and \(\sum_{t \in T_p} x_{pt} = \tilde{z}_p, \quad \forall p \in \mathcal{P}\), respectively.

It is evident that the problem under investigation turns out to be very difficult to solve, also for a simple process.

Indeed, given a manufacturing process characterized by \(|\mathcal{P}|\) production phases, given that \(T_p\) is the set of \(\mathcal{B}\mathcal{A}\mathcal{T}_s\) that can be used in the production phase \(p\), given a situation in which \(z_p\) different technical options can be selected for each phase and that the technologies are independent, then the total number of different sets of \(\mathcal{B}\mathcal{A}\mathcal{T}_s\) to be evaluated is equal to \(\prod_{p \in \mathcal{P}} (|T_p|!/z_p!(|T_p| - z_p)!)\).

These considerations have motivated the development of a heuristic for solving problem (1)–(5). In particular, attention has been focused on a genetic algorithm. The main reason for using such a solution approach to execute \(\mathcal{B}\mathcal{A}\mathcal{T}_s\) selections and optimization is that the genetic algorithms turn out to be particularly suited to solve multiobjective optimization problems [4–6], since they deal simultaneously with a set of possible solutions (i.e., population) and, thus, allow to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of traditional mathematical programming techniques.
3. Evolutionary approaches to multiobjective optimization

Evolutionary algorithms are search procedures that imitate the process of natural evolution in order to solve complex optimization problems. Since the 1960s, several evolutionary methodologies have been proposed that can be broadly grouped in: genetic algorithms [7–9] evolutionary programming [10] and evolution strategies [11,12].

The main idea of the mentioned approaches is to process a set of candidate solutions, called population, simultaneously. Population is modified by two basic principles of evolution found in nature: selection and variation.

More precisely, each solution (individual) is evaluated, by using a fitness function, that gives a relative measure of the quality of the individual in the population. Selection focuses attention on high-fitness individuals. In other words, the fittest solutions are more likely to be selected to produce the next generation of individuals (i.e., to reproduce their genetic information).

The other principle, variation, imitates natural capability of creating new individuals and is simulated by using operators of crossover and mutation. The crossover implements the search mechanism, whereas the mutation operator is used to maintain an appropriate diversity level among the population, by means of rendering the genetic search random.

Although simplistic from a biologist’s viewpoint, genetic algorithms have proven to be useful in a great variety of search and optimization problems arising in engineering, science and commerce [7].

It is worth observing that many real-world problems involve simultaneous optimization of several objective functions, that are, generally, in conflict. In this context, the notion of optimality is replaced by the notion of non-dominance or Pareto-optimality, since in general different solutions exist which can be considered as best solutions of the problem, in the sense that they cannot be improved.

In other words, it is possible to find a set of feasible solutions (i.e., the Pareto-optimal solution set) for which no improvement in any objective function is possible without worsening at least one of the other objective functions.

Evolutionary algorithms seem to be naturally well-suited to solve multiobjective optimization problems, since they deal with populations of individuals and, thus, they allow to find several non-dominated solutions during their iterative search process.

The existing evolutionary approaches for multi-objective optimization problems (surveys can be found in [4–6,13] and references therein) fall into two broad categories: non-Pareto approaches and Pareto-based approaches.

The methods in the former category do not deal with all the objective functions simultaneously and require some simplification process. We cite the vector evaluated genetic algorithm (VEGA) first introduced by Schaffer [14,15] and the lexicographic genetic algorithm proposed by Fourmen [16].

On the other hand, the latter approaches are based on the non-dominance concept of Pareto and use the Pareto optimality in the search process [17,18].

The pioneering Pareto-based evolutionary approaches are the multi-objective optimisation genetic algorithm (MOGA) developed by Fonseca and Fleming [17] and the non-dominated sorting genetic algorithm (NSGA) proposed by Srinivas and Deb [18].

In order to improve the algorithm performance, several selection and diversification strategies have been introduced, like Pareto domination tournaments [19], elitism [7,20–24], niching [19,25,26] and crowding [27,28].
It is worth observing that our cited references represent only a small fraction of the number of published papers which propose evolutionary algorithms for multiobjective optimization: a general bibliography, maintained by Carlos A. Coello. Coello, is available at ⟨http://www.lania.mx/ccoello/EMOO/EMOObib.html⟩.

The next section presents the solution approach we use to deal with the problem (1)–(5). More specifically, we consider a niched Pareto genetic algorithm. The choice of this solution approach is mainly due to the fact that this algorithm is very easy to implement, it is efficient because it does not apply Pareto ranking to the entire population and finally it seems to have a good overall performance [19].

3.1. The niched Pareto genetic algorithm

For solving the problem (1)–(5) we have developed a Niched Pareto Genetic Algorithm (NPGA, for short), that is, an evolutionary multiobjective optimization algorithm developed by Horn et al. [19].

This approach extends the traditional genetic algorithm to multiobjective optimization problem, through the use of Pareto domination ranking and fitness sharing.

The main feature of NPGA is related to the specific strategy used to select the individuals of the current population, that will have their genotypic information passed to the next generation.

The mechanism of selection used in NPGA is a particular tournament selection process based on Pareto dominance principles.

More specifically, at the beginning of each selection process, two random candidates are picked from the population. In order to find the winning solution, a comparison set that contains a specific number \(t_{dom}\) of other individuals in the population is randomly selected.

Then, both candidates are compared with the individuals of the comparison set for domination with respect to objective functions. If only one candidate dominates the comparison set, this individual is selected as the winner. On the other hand, if both are either non-dominated or dominated, a fitness sharing procedure is implemented to determine the tournament winner.

The goal of the fitness sharing method [29,25] is to preserve population diversity and to allow a reasonable representation of the Pareto-optimal front.

The main idea behind sharing is to degrade the fitness of an individual according to the presence of similar individuals in the population.

In particular, for each individual \(\mathcal{I}\), with fitness \(f_{\mathcal{I}}\), the shared fitness \(f_{\mathcal{I}}^{sh}\) is determined by simply dividing \(f_{\mathcal{I}}\) by the niche count \(m_{\mathcal{I}}\), that is:

\[
f_{\mathcal{I}}^{sh} = \frac{f_{\mathcal{I}}}{m_{\mathcal{I}}}.\tag{6}
\]

The niche count \(m_{\mathcal{I}}\) is an estimate of the number of individuals that share with \(\mathcal{I}\) the fitness \(f_{\mathcal{I}}\), and is determined by summing a sharing function \(\mathcal{S}\hat{\mathcal{H}}(d_{\mathcal{I}j})\) over the individuals of the current population, that is:

\[
m_{\mathcal{I}} = \sum_{j=1}^{N} \mathcal{S}\hat{\mathcal{H}}(d_{\mathcal{I}j}); \tag{7}
\]

where \(N\) presents the population size, whereas \(d_{\mathcal{I}j}\) denotes the distance between individuals \(\mathcal{I}\) and \(j\).
The sharing function allows to evaluate the degree of similarity between two individuals of the population. It results as one if the elements are identical, zero if their distance is higher than a dissimilarity threshold value (i.e., niche radius $\sigma_{\text{shared}}$) and an intermediate value otherwise.

More specifically, the sharing function $SH(d_{IJ})$ is a decreasing function of the distance $d_{IJ}$, that can be represented mathematically as follows:

$$SH(d_{IJ}) = \begin{cases} 1 - \left( \frac{d_{IJ}}{\sigma_{\text{shared}}} \right)^z & \text{if } d_{IJ} \leq \sigma_{\text{shared}} \\ 0 & \text{otherwise} \end{cases}$$

where $z$ is a parameter, whose value determines the shape of the sharing function.

Typically, $z$ is set equal to one and the resulting sharing function is referred to as the **triangular sharing function**. Individuals within $\sigma_{\text{shared}}$ distance of each other degrade each other’s fitness, since they are in the same niche.

In order to determine the distance $d_{IJ}$ between individuals $I$ and $J$ a distance metric is required. The distance can be computed either in the decision space (i.e., chromosomal representation of an individual) or in the objective space (i.e., fitness of individual).

In our study, we have considered a phenotypic distance that is directly linked to real parameters of the search space.

It is worth observing that in NPGA, the non-dominance is computed by comparing an individual with a randomly chosen population set of size $t_{\text{dom}}$. Thus, the success of this algorithm highly depends on the parameter $t_{\text{dom}}$. The effect of the dominance pressure on the behaviour of the NPGA has been investigated by Horn and Nafpliotis in [30], where some empirically derived guidelines for setting its value are given. More specifically, let $N$ be the population size, the computational experiments reported in [30] underline that:

- if $t_{\text{dom}} \approx 1\%$ of $N$ then NPGA determines too many dominated solutions;
- if $t_{\text{dom}} \approx 10\%$ of $N$ then NPGA yields a tight and complete distribution;
- if $t_{\text{dom}} \approx 20\%$ of $N$ then NPGA prematurely converges to a small portion of the front.

On the basis of the previous considerations, it is evident that if a proper size is not chosen, Pareto-optimal points may not be found.

The effectiveness of the sharing principle mainly depends on the parameter $\sigma_{\text{shared}}$. More specifically, the lower the value of $\sigma_{\text{shared}}$, the worse the diversification. On the other hand, the higher the value of $\sigma_{\text{shared}}$, the lower the progression speed towards the pareto front. Consequently, the niche radius must be set carefully and various empirical formulas have been proposed to set this dissimilarity threshold [17,30].

4. The manufacturing process

We have focused our attention on the industrial painting of wood. Our choice has been motivated by the fact that the directive 1999/13/CE of the European Commission [31,32] applies to the companies working in this productive sector. Such a directive focuses on the emissions of volatile organic compounds (i.e., $VOC_s$) into the environment by manufacturing industries and, in order to prevent or reduce the direct and
indirect effects of such emissions, the directive has established limit values for emission of VOC$_s$ that cannot be exceeded.

The industries that violate such limits (i.e., operate above the solvent consumption thresholds defined in [31]) should consider the implementation of BAT$_s$ in order to meet the requirements of this norm. In general, the following strategies can be adopted to satisfy the constraints imposed by the directive:

- consider specific systems that reduce the quantity of VOC$_s$ produced (i.e., abatement equipments);
- use potentially less harmful substitutes in the production process.

In what follows, we describe the wood painting process and we present alternative approaches that can be followed to set and achieve the emission limits for VOC$_s$ in this manufacturing process, in accordance with the BAT$_s$ principle.

It is worth observing that for the process under consideration, the BAT$_s$ could be paints and special dyes, specific application systems which allow to apply paint ensuring the minimum level of pollution emissions and also abatement equipments (i.e., removal and suction plants). The main environmental problems, related to the use of painted products, come from the quantity of solvents contained in their formulation. They not only turn into VOC emissions in the atmosphere, but they also influence the waste typology emitted by the wood painting cycle (water from painting chamber, smudges, paint boxes, diluent residuals used for cleaning equipment, wood wastes, etc.).

The wood painting process can be viewed as characterized by the following three main production phases:

- dye laying off;
- primer application;
- final coating.

Raw materials, production plants, abatement equipment and costs are associated to each phase. In particular, we can distinguish among the following raw materials:

- dyes;
- paints;
- solvents.

It is important to point out that it is possible to use chemical reticulation products which emit less solvents (i.e., polyurethane, acrylic and polyester paints). There is also the opportunity to use water paints even if, in this case, different factors should be taken into account. Indeed, many difficulties have been found in obtaining a convenient reticulation of the resins used in these formulations. This implies that tendentially water paints give lower performances (mechanical, chemical and sometimes also esthetical resistances) in comparison with the traditionally used solvent paints.

Consequently, the raw materials for the painting of wood can be better classified as follows:

- water dye;
- solvent dye;
- polyurethane;
The main application systems that can be used in the process under consideration are described in what follows.

- **Normal or assisted spray gun**: (most commonly known as air compressed system) It represents the oldest spraying system. It combines compressed air and paint, producing small atomized particles. This approach allows to have an easy management of the paintings and an optimal atomisation. In this way, it is then possible to obtain excellent final coats, perfectly smooth and flat.
- **Normal or assisted airless**: It is without doubt the fastest spraying system. As a matter of fact, this approach allows to spray in the same time unit the double quantity of material than the air compressed system. This is the main reason why the airless one is the most favourable system for high speed painting lines or for great dimension details.
- **Normal or assisted airmix**: The airmix systems have been developed to increase the transfer efficiency and the productivity of the air compressed conventional systems; in fact, an efficiency increase of about 30% has been verified.
- The airmix allows to obtain an excellent esthetical final coat, very similar to the one obtained by using the spray gun and better in comparison with the one obtained with the airless systems.
- **Normal or assisted High Volume and Low Pressure (HVLP)**: HVLP is similar to the traditional air compressed system, except that the air and fluid nozzle are designed to reduce the spray velocity. Higher air volumes are used at lower pressures. This approach allows an overspray reduction of 30–35%. Such a system is very efficient because of its capacity to atomize paint over high air volume, produced at very low pressure. In addition, this approach sprays well into recesses and cavities. On the other hand, the main disadvantage of HVLP is that it sometimes has difficulty in reaching high quality atomization at higher rates of application, which affects the quality of the finish.
- **Immersion**: Immersion painting is the best choice when a high productivity is required on geometrically simple details. This technique is very cost effective. However, the quality of the finish is very poor and thus this technique is usually used in primer application.

In order to reduce emissions of VOCs into the environment, the following suction plans and abatement equipments can be used:

- dry suction;
- wet suction;
- incinerator;
- active carbons.

On the basis of the previous considerations, the generic model depicted in Fig. 1, has been specialised for the specific production process under investigation, as shown in Fig. 2, where, for the sake of simplicity, only the raw materials and the BATs are reported.
5. Design of NPGA for industrial painting of wood

The first step in designing a genetic algorithm for a particular problem is to devise a suitable scheme to represent the individuals. In the NPGA, proposed for solving the problem under investigation, the set of technical options available for each phase is viewed as an individual or member of the population. Consequently, a binary representation is an obvious choice for the problem since it represents the underlying 0 – 1 integer variables. Hence, in our representation, we use a n-bit binary string, where $n = \sum_{p \in P} |T_p|$ is the number of variables in the problem.

This binary representation of an individual’s chromosome is illustrated in Fig. 3.

In addition, each individual $I$ is characterized by the two-dimensional fitness vector $f^j(I)$, with $j = c, e$, where $f^c(I)$ represents the total cost associated to the solution, whereas $f^e(I)$ indicates the corresponding total emission.

Given a population of $N$ individuals, the selection of a candidate is made by using a tournament selection procedure (see Section 3.1), repeated $n_{cycle}$ times, which is illustrated in the following.
Tournament Selection Procedure

Step 1
Set \( count = 1 \).

Step 2
Pick randomly two candidates for selection \( I_1 \) and \( I_2 \).
Select randomly a comparison set \( \mathcal{I} \) of other individuals in the population, such that \( |\mathcal{I}| = l_{\text{dom}} \).

Step 3
Compare each candidate \( I_1 \) and \( I_2 \) against each individual belonging to \( \mathcal{I} \) for domination.

Step 4
If \( I_1 \) [or \( I_2 \)] is dominated by the individuals in \( \mathcal{I} \) while \( I_2 \) [or \( I_1 \)] is not, then select \( I_2 \) [or \( I_1 \)] for reproduction and go to Step 6.

Step 5
If neither or both candidates are dominated by the comparison set \( \mathcal{I} \), then implement a fitness sharing procedure to determine the tournament winner.

Step 6
If \( count = N \) then Stop; else set \( count = count + 1 \) and go to Step 2.

For the candidate \( I_1 \) [or \( I_2 \)], the fitness sharing procedure is performed in the following way.

Fitness Sharing Procedure

Step 1
Set \( J = 1 \).

Step 2
Compute the distance \( d_{I_1 \mathcal{I}} \) [or \( d_{I_2 \mathcal{I}} \)] between the candidate \( I_1 \) [or \( I_2 \)] and another individual \( \mathcal{I} \), as follows:

\[
d_{I_1 \mathcal{I}} = (d_{1(I_1)}^{1/p} + d_{2(I_1)}^{1/p})^p,
\]
\[ d_{I_1 J} = \left( d_{I_1 J}^{1/p} + d_{I_2 J}^{1/p} \right)^p, \]

where

\[ d_{I_1 J} = \text{abs}(f^c(I_1) - f^c(J)), \]

\[ d_{I_2 J} = \text{abs}(f^c(I_2) - f^c(J)); \]

and

\[ d_{I_1 J} = \text{abs}(f^e(I_1) - f^e(J)), \]

\[ d_{I_2 J} = \text{abs}(f^e(I_2) - f^e(J)). \]

**Step 3**

Compare the distance \( d_{I_1 J} \) [or \( d_{I_2 J} \)] with a predetermined niche radius \( \sigma_{\text{shared}} \) and compute the following fitness sharing function.

\[
    \mathcal{SH}(d_{I_1 J}) = \begin{cases} 
        1 - (d_{I_1 J} / \sigma_{\text{shared}})^2 & \text{if } d_{I_1 J} \leq \sigma_{\text{shared}}, \\
        0 & \text{otherwise}, 
    \end{cases}
\]

or

\[
    \mathcal{SH}(d_{I_2 J}) = \begin{cases} 
        1 - (d_{I_2 J} / \sigma_{\text{shared}})^2 & \text{if } d_{I_2 J} \leq \sigma_{\text{shared}}, \\
        0 & \text{otherwise}. 
    \end{cases}
\]

**Step 4**

Set \( J = J + 1 \). If \( J \leq N \), go to Step 2; else calculate the niche count \( m_{I_1} \) [or \( m_{I_2} \)] for the candidate \( I_1 \) [or \( I_2 \)] as follows:

\[
    m_{I_1} = \sum_{J=1}^{N} \mathcal{SH}(d_{I_1 J});
\]

or

\[
    m_{I_2} = \sum_{J=1}^{N} \mathcal{SH}(d_{I_2 J}).
\]

**Step 5**

Compare \( m_{I_1} \) and \( m_{I_2} \). If \( m_{I_1} < m_{I_2} \), then select the first candidate \( I_1 \), else choose the second candidate \( I_2 \).

We have adopted for the \( NPGA \) a uniform crossover; in uniform crossover two parents have a single child. Each bit in the child solution is created by copying the corresponding bit from one or the other
In order to deal with premature convergence, a mutation operator is randomly activated during the genetic cycles and modifies some randomly selected bits of the individual string, that is the chosen bits changes from 0 to 1 or viceversa. The rate of mutation is generally set to be a small value (i.e., 1 or 2 bits per string).

Clearly, the child solution generated by the crossover and mutation operator may not be feasible because the set of the chosen technical options may be incompatible with physical engineering constraints. In order to guarantee feasibility, a *legality operator* is applied [33].

6. Numerical results and discussion

In this section, we present the results obtained from the application of the *NPGA* on the problem described in Section 4, by considering as case study the manufacturing process of an Italian company specialized in the industrial painting of wood.

The main activity of the company in question is the manufacturing of chairs. In this case, the threshold value for the consumption of solvents, established by the 1999/13/CE norm, is fixed at 15 tons per year. The consumptions are determined by considering the materials used in production as well as the solvents used in diluting, in cleaning instruments and machinery and for the maintenance of factory plants.

A study carried out by the CATAS S.p.A. [34], in order to evaluate the impact of the *EEC*’s solvent norm on companies in the “Chairs Triangle” manufacturing district, has shown that the production of a single chair determines an average amount of 226.29 g of organic solvent, including those used in cleaning operations.

The company considered in our study produces 1,000,000 chairs per year. Thus, on the basis of the previous considerations, it is evident that this company operates above the solvent consumption thresholds established by the *ECC*’s norm and, consequently, it is required to find the combinations of *BAT*s that allow to achieve the emission limits for *VOC*’s. It is important to point out that the company was interested in finding all the non-dominated solutions. The choice has been the one that not only allows to satisfy the emission limits but also corresponds better to the owner’s preferences. To meet this goal the proposed *NPGA* has been used.
The computational experiments have been carried out by normalizing cost and emission function values, since the wide range variability of the related results could adversely affect the quality of the Pareto-front. We performed a sensitivity analysis on the population size $N$, on the parameters $t_{dom}$ and $\sigma_{shared}$, in order to evaluate their impact on the final Pareto-front.

More specifically, we have considered three values for the population size, that is $N = 30$, 60 and 90, the sharing parameter $\sigma_{shared}$ has been chosen equal to 0.1, 0.01, 0.001, 0.0001, whereas the parameter $t_{dom}$ has been set equal to 1, 2, and 3 for $N = 30$, 2, 4, and 6 for $N = 60$, and 3, 6, and 9 for $N = 90$.

In all the computational experiments, the number of generations has been set equal to 90. The related results are plotted in Figs. 5–10, where for the most meaningful combinations of the parameter values, we report the actual cost $f_c$ and the corresponding actual emission $f_e$.

Regarding the quality and the diversity of the solutions found by NPGA, the performance of the algorithm depends mainly on two parameters: population size and sigma value.
In particular, a value of 0.01 for $\sigma_{shared}$ seems to be the best compromise for convergence rate and quality and diversity of solutions, instead in relation to the population size, the best results have been obtained by considering $N = 90$. On the other hand, the computational results show that the performance of the method is not affected by the $t_{dom}$ values in the considered range.

Regarding the convergence rate of $NPGA$, from Fig. 11 it is evident that around the 50th iteration, the algorithm determines a set of non-dominated solutions, which seems to approximate the Pareto-front.

Applying the proposed $NPGA$ based approach by using the appropriate values for the corresponding parameters (that is, $N = 90$, $\sigma_{shared} = 0.01$, $t_{dom} = 9$), the distribution of non-dominated solutions is reported in Fig. 12.
In order to show the effectiveness of the proposed solution approach, we have also carried out computational experiments by treating the problem as a single-objective optimization problem, where a linear combination of cost and emission objective functions has been considered. The related problem assumes the following form:

\[
\min \lambda \sum_{p \in P} \sum_{t \in T_p} c_{t} x_{pt} + (1 - \lambda) \sum_{p \in P} \sum_{k \in X} \sum_{t \in T_p} r_{pk} t x_{pt}
\]  

subject to constraints (3)–(5), where \( \lambda \) is a weighting factor.

To generate non-dominated solutions, the single-objective optimization problem has been solved for different values of \( \lambda \), by using a single objective genetic algorithm (SOGA, for short). More specifically,
10 different problems have been defined and solved, by varying the value of $\lambda$ in the range $[0, 1]$ with stepsize 0.1; the related results are reported in Fig. 13.

By observing Fig. 13, the advantage of the use of a NPGA based approach is evident. It is worth noting that the solutions obtained by SOGA using different weights for $\lambda$ are very close to the ones generated by the NPGA based approach. However, by using appropriate values for the population size and the sharing parameter, NPGA has found two non-dominated solutions (i.e., $S_1 = (74.71, 6.03)$ and $(S_2 = 101.89, 0)$) that the other approach was not able to determine.

In addition, solution $S_1$ represents the best compromise between achievement of the emission limit (i.e., 15 tons) and minimization of the total cost. On the other hand, it is worth observing that solution $S_0 = (47.74, 13.96)$ (found by both methods) allows to satisfy the emission limit. However, since the evaluation of the emission reductions is usually obtained on the basis of simulation models (i.e., it can
be affected by errors) and, in addition, the total emission (i.e., 13.96 tons) obtained by considering $S_0$ is very close to the limit (i.e., 15 tons), it seems reasonable to select solution $S_1$.

On the basis of the previous considerations, it is evident that, in the case in which only the objective aggregation approach is available, the user is forced to select solution $S_3 = (109.04, 0)$ (that is dominated by the solution $S_2 = (101.89, 0)$ found by the $NPGA$-based approach), with a consequent useless waste of resources.

It is worth observing that, even though, given both the discrete nature and the small size of the practical problem considered in the computational experiments, the total number of non-dominated solutions is very limited, the improvement in terms of quality of the solutions of the proposed $NPGA$ with respect to the objective aggregation approach can be considered satisfactory.

Since large real instances were not available, in order to better assess the effectiveness of the proposed $NPGA$ based approach and to verify if it displays a trend similar to the one observed in the real case study when solving large problems, computational experiments have also been carried out by considering a set of randomly generated test problems.

A random problem generator has been implemented and used for this task.

The total number of best technologies available for each phase, the cost of each technical option, the total reduction of each pollutant when a technical option is used on a given phase, the set of pollutants generated from each production phase have all been chosen randomly according to a uniform distribution, for all the test problems considered.

In Table 1, we report the characteristics of the considered test problems (i.e., number of phases, number of compatibility constraints and number of $BAT$ range).

For each test problem, given the number $|\mathcal{F}_p|$ of $BAT$ that can be used in phase $p$, the value of $z_p$ has been chosen, according to a uniform distribution, within the range $[1, \lceil |\mathcal{F}_p|/2 \rceil]$.

Also for this set of test problems, a sensitivity analysis has been carried out on the number of generations, the population size, the parameters $\sigma_{\text{shared}}$ and $t_{\text{dom}}$, with the aim of assessing the impact of their value on the final Pareto front.

A detailed accounting of the experimental results is reported in [35] and includes, for each test problem, the solutions determined by $NPGA$ and the corresponding execution time, for each combinations of the
Table 1
Characteristics of test problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of phases</th>
<th>Number of compatibility constraints</th>
<th>Number of $\mathcal{A} \mathcal{I}$ range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>20</td>
<td>10</td>
<td>[1, 10]</td>
</tr>
<tr>
<td>Test 2</td>
<td>25</td>
<td>12</td>
<td>[1, 10]</td>
</tr>
<tr>
<td>Test 3</td>
<td>30</td>
<td>15</td>
<td>[1, 15]</td>
</tr>
<tr>
<td>Test 4</td>
<td>35</td>
<td>17</td>
<td>[1, 15]</td>
</tr>
<tr>
<td>Test 5</td>
<td>40</td>
<td>20</td>
<td>[1, 20]</td>
</tr>
<tr>
<td>Test 6</td>
<td>45</td>
<td>22</td>
<td>[1, 20]</td>
</tr>
<tr>
<td>Test 7</td>
<td>50</td>
<td>25</td>
<td>[1, 25]</td>
</tr>
<tr>
<td>Test 8</td>
<td>55</td>
<td>27</td>
<td>[1, 25]</td>
</tr>
<tr>
<td>Test 9</td>
<td>60</td>
<td>30</td>
<td>[1, 30]</td>
</tr>
<tr>
<td>Test 10</td>
<td>65</td>
<td>32</td>
<td>[1, 30]</td>
</tr>
</tbody>
</table>

Fig. 14. Comparison between SOGA and NPGA based approach for problem Test 3.

parameter values, and the results obtained by SOGA, when treating the problem as a single objective optimization problem.

The results obtained were qualitatively comparable for all considered instances. Hence, for the sake of brevity, in what follows we report only the results related to some of the test problems (i.e., Test 3, Test 5, Test 9 and Test 10), that will be used as the target of our discussion. In particular, Figs. 14–17 show the computational results obtained by the NPGA (i.e., the solutions of the first and the last generation of NPGA), on the basis of appropriate values of the aforementioned parameters, and the solutions determined by SOGA.

The first thing to observe is that, for all test problems, the solutions obtained by SOGA, using different weights, are very close to the approximated front generated by NPGA. However, the obtained outcomes are dominated by at least one of the solutions determined by NPGA. This behaviour is better underlined
by the results plotted in Figs. 18–21, in which we compare the approximated Pareto front obtained by 
NPGA and the set of non-dominated solutions determined by SOGA.

The results collected also indicate that, with a limited increase in computational effort, compared to the 
execution time of a single run of SOGA, NPGA determines a diverse set of non-dominated solutions and provides a robust outcome to the decision maker analysis. In particular, the execution time required by NPGA to generate the approximated Pareto set is on average 2.5% more than the computational cost required by SOGA to determine only one solution [35].

To illustrate the diversity of solutions found by NPGA, Table 2 reports several different combinations of 
BATs obtained by applying NPGA to the test problems considered.

In particular, for each test problem, solution \( s_1 \) is the best in terms of cost, solution \( s_2 \) represents a good compromise between cost and pollutant emissions, whereas \( s_3 \) is the best considering only the pollutant emissions.
Fig. 17. Comparison between SOGA and NPGA based approach for problem Test 10.

Fig. 18. Comparison between the set of non-dominated solutions found by SOGA and the approximated Pareto front determined by NPGA for the problem Test 3.

Fig. 19. Comparison between the set of non-dominated solutions found by SOGA and the approximated Pareto front determined by NPGA for the problem Test 5.
Fig. 20. Comparison between the set of non-dominated solutions found by SOGA and the approximated Pareto front determined by NPGA for the problem Test 9.

Fig. 21. Comparison between the set of non-dominated solutions found by SOGA and the approximated Pareto front determined by NPGA for the problem Test 10.

Table 2
Some non-dominated solutions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solutions ($f^c$ (K Euros); $f^c$ (Tons))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$s_1$</td>
</tr>
<tr>
<td>Test 3</td>
<td>(115.12; 180.21)</td>
</tr>
<tr>
<td>Test 5</td>
<td>(84.11; 158.00)</td>
</tr>
<tr>
<td>Test 9</td>
<td>(155.03; 128.07)</td>
</tr>
<tr>
<td>Test 10</td>
<td>(125.12; 1750.11)</td>
</tr>
</tbody>
</table>

With the non-dominated solutions in hand, the user can choose for each phase the appropriate combination of $BIF_s$, that allows to reach the best compromise between achievement of the emission limit and minimization of the total cost.
7. Concluding remarks

In this paper, a \( NPGA \) based approach has been applied to the pollutant emission reduction problem in the manufacturing industry. The problem under investigation has been formulated as a bicriteria optimization problem with competing cost and environmental impact objectives. The sensitivity of the proposed solution approach to the parameters that control the behaviour of the algorithm itself have been assessed, namely population size, tournament size and niche radius.

The computational results, collected on a real case study and on a set of randomly generated instances, are satisfactory and show that: (1) the proposed solution approach turned out to be more effective than treating the problem as a single objective optimization problem; (2) it can be used to effectively address the optimal selection of best available technologies, to maximize the total pollutant removal and minimize the overall cost; as a matter of fact, one of the solutions determined by the proposed \( NPGA \) based approach has been implemented by the Italian company, considered as a case study.

References


[34] Norms regarding solvents, a survey carried out by the CATAS organisation to see the impact of the EEC’s solvents norms on companies in the Chairs Triangle manufacturing district, 2002. http://www.timberandmore.com/news/