[Post-print version] Please cite as : Ishizaka A,Nemery P, Assigning machines to incomparable maintenance strategies with ElectreSort, OMEGA, 2014, advance online publication, 10.1016/j.omega.2014.03.006

Assigning machines to incomparable maintenance strategies with ElectreSort

Alessio Ishizaka[1], Philippe Nemery[2]

[1]Portsmouth Business School, University of Portsmouth, Portland Street – Richmond Building, Portsmouth PO1 3DE, UK, Alessio.Ishizaka@port.ac.uk
tel: 0044 23 92844171

[2] SAP Software Systems – SAP China Labs – Shanghai Pudong Software Park, 1001 Chenhui Road, Shanghai, 201203 P.R. of China, pnemery@gmail.com

Abstract

The increasing demand on productivity and quality requires machines to be constantly available for production. It is therefore crucial to develop an adequate maintenance programme. To facilitate this, several criteria need to be considered, such as: downtime, maintenance frequency, spare parts costs, bottleneck impacts, etc. In the literature, a strategy is selected for each machine with a multi-criteria decision choice method. However, before making an informed decision, each strategy needs to be tested on each machine and then their performances evaluated with a multicriteria decision method. This is time-consuming, inefficient and often unfeasible. As machines' performances are usually systematically collected by industries, a much more practical approach is to assign machines to a maintenance strategy. This is referred to as a sorting problem. However, this problem cannot be solved by existing multi-criteria sorting methods because maintenance strategies cannot always be completely ordered: incomparable strategies exist. Recently, a Decision Making Grid was proposed to allocate machines to incomparable strategies. However, this technique can only be applied to problems with two criteria. In this paper, we have developed ElectreSort, a new sorting method that is able to consider an unlimited number of criteria in order to assign machines to incomparable strategies. A case study illustrates that ElectreSort provides more precise and flexible maintenance strategies than the *Decision Making Grid*.

Keywords: Maintenance, multi-criteria decision making, incomparable strategies, sorting, ELECTRESort

1. Introduction

Maintenance management has become a key strategic task in an organisation as testified in recent state of the art surveys [1-3]. The main objective of maintenance is to keep machine resources running by extending their availability through a combination of technical and administrative actions. Neglected machines may lead to breakdowns causing production or service disruption resulting in delays, planning problems, bottlenecks, high costs or even human casualties. Managers need, therefore, to select the correct maintenance strategy since disruption has a significant impact on budget, technological choices, managerial and organisational procedures, etc. In addition, plants can have thousands of machines (e.g. gauges, pumps, filters, compressors), and their management can be difficult and cumbersome. To choose appropriate maintenance strategies, managers need to consider several criteria such as downtime length, failure frequency, safety, criticality of the whole system, etc [4]. This important and complex problem has often been solved as a choice problem with multi-criteria decision techniques, see Gandhare & Akarte [5] for a review. The goal is to select the single best option among all possible strategies evaluated in regard to each machine. Sorting techniques have never been used. With these techniques, actions (machines) are assigned to ordered and predefined groups, called classes (i.e. our strategies). Selecting the best strategy for each machine has two major methodological drawbacks, which can be remedied if the problem is tackled as a sorting problem of machines to strategies:

- For each machine a choice problem needs to be solved. This is time-consuming when the number of machines is large (defining the parameters of the method for each machine, etc.). One way to overcome this problem is to group machines with same characteristics, e.g. in [6], but the drawback is that the solution is not customised anymore, and approximations are introduced. A better way is to use a sorting approach. When the classes (i.e. strategies) are defined, the assignment can be automated. Moreover, in sorting methods, as all machines are processed by the same algorithm, it creates the possibilities of benchmarking and comparisons studies. This is not possible with separate treatments as in the choice problems.
- In order to have a precise result, each strategy should be tested on each machine and then the performances of the strategies evaluated with a multicriteria choice method on the criteria costs, downtime, frequency… In practice, this is totally inefficient, costly and

often unfeasible, so in practice this approach never happens in industry. Performance data are normally systematically collected from machines (downtime, frequency, etc). Therefore, the machines can be naturally considered as the actions to be assigned to a strategy and not the other way around.

Multi-criteria sorting techniques have not yet been used to assign machines to strategies, probably because they are not adapted for sorting incomparable maintenance strategies. For example, we consider machine A, which has one breakdown with 50 minutes downtime each month, and machine B, which has five breakdowns with 10 minutes downtime each month. Both machines have a total downtime of 50 minutes per month but the causes are different and consequently the resolving maintenance strategy will also be different. One way to solve this problem is to use a “Decision Making Grid” [7], where an aggregation of the criteria is not performed in order to be able to define incomparable strategies. However, the drawback of the Decision Making Grid is that it is only relevant to problems which consider only two criteria.

In this paper, we introduce ElectreSort, a new sorting method able to deal with incomparable strategies while taking into account an unlimited set of criteria.

The structure of this paper is as follows: Section 2 reviews from the literature the main techniques used to select a maintenance strategy and explains their shortcomings. Following this, ElectreSort, the new proposed multicriteria sorting method, is described in Section 3. In Section 4, the ElectreSort method is applied on a case example. The results are then discussed and compared to the Decision Making Grid methodology [7] in Section 5. Finally, conclusions and future research directions are presented.

2. Literature review

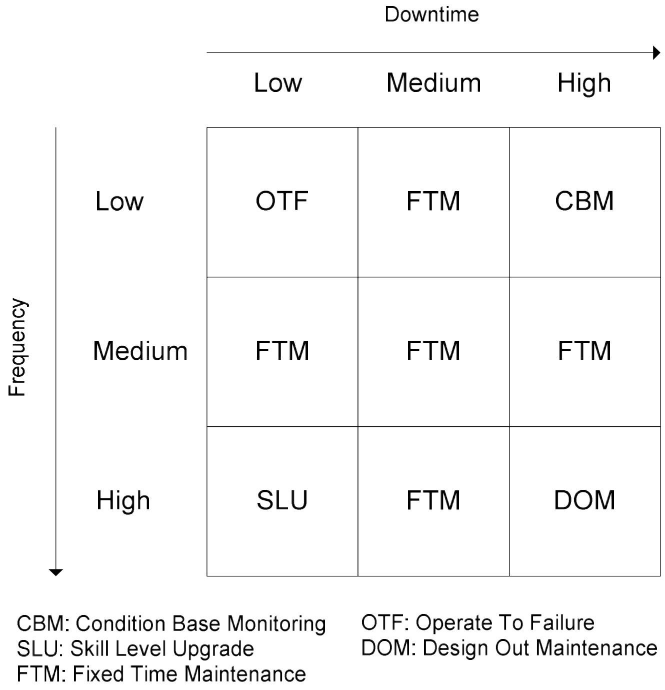
2.1. Selection of a maintenance strategy

The selection of a maintenance strategy is a multicriteria decision making (MCDM) problem, where several MCDM methods have been applied. Fuzzy ELECTRE was adopted to select the best maintenance strategy for a compressor [8]. AHP and ANP were applied concurrently to select the best strategy for all the machines of a printing house [9]. Fuzzy AHP and AHP were used concurrently to find the best strategy for the 70 pieces of equipment (pumps, fans, boilers, etc) of a small thermal power plant [10]. Then, the same problem was solved with TOPSIS and Relative Membership Grade [11]. AHP was used in conjunction with factor analysis for selecting the best maintenance strategy of an industrial unit including lathes, die and press, CNC, welding etc [12]. Shyjith, Ilangkumaran, & Kumanan [13] used AHP to calculate the weight of the criteria and TOPSIS to select the best maintenance strategy for all the machines of a textile plant. Shyjith, Ilangkumaran, & Kumanan [14] extended their work and used Fuzzy AHP for the weight calculation. Fuzzy methods were used in an hypothetical problem [15], and also in an oil industry application for four different machines [16, 17].

These MCDM techniques are generally used to select one maintenance strategy which is applied to all the machines within a plant, because selecting a specific strategy for each machine is very time-consuming. However, there is no guarantee that all machines of the plant have the same breakdown characteristics. This problem has been recognised by Bevilacqua & Braglia [6], who used AHP to select the best maintenance strategy for an important Italian oil refinery, based on four criteria (damages, applicability, added-value, costs). As the number of machines is very high and the decision process is very time-consuming, they first sort the machines into three classes based on a score (called the criticality index) obtained from a weighting sum of five criteria (safety, machine importance for the process, maintenance costs, failure frequency, and downtime length). Then, only one machine from each group is selected at random to determine the maintenance strategy for the entire group of machines. This method has several short-comings:

- The sorting phase is based on a global score which implies that some precious information is lost in the aggregation. For example, machine A scoring 10 on “damages” and 0 on “added-value” will have the same score as a machine B scoring 0 on “damages” and 10 on “added-value”. It is hard to believe that the maintenance strategy should be the same for these two machines, each having very different profiles. In the literature, such profiles are defined as incomparable [18].
- The criteria for sorting the machines into the different classes are different from the ones used for selecting a maintenance strategy. It is therefore difficult to justify that a strategy for one machine is valid for the entire group.

In order to take this incomparable behaviour into account, a Decision Making Grid (Figure 1) was proposed [19-21]. It is a visual two dimensional map of only two characteristics of the machine. The best maintenance strategy is then chosen according to the location of the machine in the grid. It is the first time that the problem is solved as a sorting problem instead of as a choice problem. This technique has been applied in a plant of 130 machines producing roof systems for cars [20, 22], in a disc brake pad manufacturing company [23], in a desalination plant [24] and in a food processing company [25, 26]. Later, it was combined with a fuzzy controller to regulate optimal production according to the recommended maintenance strategy [27].



		Downtime	
Frequency	Low	Medium	High
Low	OTF	FTM	CBM
Medium	FTM	FTM	FTM
High	SLU	FTM	DOM

CBM: Condition Base Monitoring
SLU: Skill Level Upgrade
FTM: Fixed Time Maintenance
OTF: Operate To Failure
DOM: Design Out Maintenance

Figure 1 : Decision Making Grid [19, 23, 28]

The Decision Making Grid is a simple bi-criteria matrix without any aggregation. Nevertheless, Burhanuddin *et al.* [29] recognised that it may be insufficient to consider only two criteria in order to make a wise decision and suggested that, for instance, costs should be considered as well. They therefore developed a two stage process, where the first stage is a simple screening: if the cost of an asset is over a threshold, it is placed in the Decision Making Grid; otherwise the asset is not considered for maintenance. This method is a first attempt to consider several criteria. However, this additional criterion is only used to decide if an asset is worth maintaining. It does not contribute to the selection of the maintenance strategy. For instance, the criticality of its output to the entire production process is important, but not essentially related to cost of the asset. Here a filter criterion may distort any subsequent ranking or strategic choice.

2.2. Classification methods

Several multicriteria methods have been proposed to assign actions to predefined classes. We usually distinguish three main families.

The first set of methods, referred to as *nominal classification methods*, handle classification problems where there is no preference order on the predefined groups. Most of these methods are based on the computation of a similarity, indifference or closeness degree between the actions to be classified and the reference actions or central profiles defining the classes. For example, let us cite amongst others PROAFTN [30], Filtering Procedures [31], TRINOMFC [32], CLOSORT [33]. These relationships are symmetric and do not incorporate the outranking nature of the problem (the classes are not ordered) while ElectreSort has been designed to take into account the outranking relation of an action compared to the reference profiles.

The second family of methods, called the sorting methods, are defined for classification problems where the classes are completely ordered (i.e. there is a complete preference structure on the groups which are by the decision maker from the best to the worst). In this case, the classes are defined either by boundary profiles or by one or several reference actions (also called central profiles). For example, let us cite amongst other Electre-Tri [34], Electre-Tri-nC [35], FlowSort [36], PromSort [37], Theseus Method [38], Rough Sets [39], AHPSort [40]. The main difference between ElectreSort and the aforementioned methods is that the latter ones can only be used when the classes are completely ordered. Incomparable classes cannot be defined.

The third family are the clustering or unsupervised classification methods. These methods focus on the automatic discovering of groups of similar objects within the dataset. The groups, also called clusters in this context, are thus not predefined. The clusters are such that elements from one cluster are very different (distant) from elements from another cluster. Only few multicriteria clustering methods have been proposed. Let us cite amongst others the following works: [41], [42] and [43].

In the next section, we describe the use of ElectreSort to solve the problem of defining the appropriate maintenance service for machines. The approach is a further development of the Decision Making Grid as it can take into account several criteria and the classes of the maintenance strategies might be incomparable

3. Methodology

3.1. Electre methods

All the ELECTRE methods are based on the same two distinct phases [44, 45]:

- the construction of the outranking relation based on a concordance (section 3.2.2) and a discordance degrees (section 3.2.3).
- the exploitation of the outranking relations (section 3.2.6).

The main difference of the ELECTRE methods lies in the type of problem they solve (Table 1) and therefore different exploitation techniques of the outranking relations are defined.

ELECTRE method	Problem formulations	Special characteristic
ELECTRE I [46]	choice problem	true criteria
Fuzzy ELECTRE I	choice problem	true criteria
ELECTRE II [47]	ranking problem	true criteria
ELECTRE III [48, 49]	ranking problem	pseudo criteria
ELECTRE IV [50]	ranking problem	pseudo criteria and no weighting on the criteria
Hierarchical ELECTRE [51]	ranking/choice problem	criteria are defined in a hierarchy
ELECTRE Tri [52]	sorting problem	pseudo criteria and limiting profiles
ELECTRE Tri-C [35]	sorting problem	pseudo criteria and one central profile per class
ELECTRE Tri-nC [53]	sorting problem	pseudo criteria and several central profiles per class

Table 1 : ELECTRE methods

The particularity of the outranking methods is that four preference relationships between two actions *a* and *b* exist. If we denote by *a* S *b,* the situation where '*a* outranks *b*', we can define the following four preference relationships:

- *a* P *b* (*a* is strictly Preferred to *b*): *a* S *b* and not (*b* S *a*)
- *b* P *a* (*b* is strictly Preferred to *a*): *b* S *a* and not (a S b)
- *a* I *b* (*a* is Indifferent to *b*): *a* S *b* and b S *a*
- *a* R *b* (*a* is incompaRable to *b*): not (*a* S *b*) and not (*b* S *a*)

Electre-Tri B [52], the first sorting method based on the ELECTRE concept and using limiting profiles to define classes, does not accept incomparability amongst the profiles. In order to overcome this limitation, in 2008, Nemery [54] first posited the notion of ELECTRESort, where classes are defined by central limiting profiles, which can also be incomparable. In this paper, we further develop this method and apply it to a maintenance selection strategy. Worth noting at this point is the subsequent development of two other sorting methods: ELECTRE Tri-C [35] and ELECTRE Tri-nC [53]. As with ELECTRESort, these two methods are based on central profiles but their outranking exploitation algorithm does not accept incomparability.

3.2. ElectreSort

The ElectreSort method is a multicriteria sorting method which assigns objects or actions to predefined classes. These classes can be partially ordered, which means that the central profiles defining these classes have a natural ordering on each criterion but that this ordering is not the same for all the criteria. Therefore, classes can be *incomparable* (e.g. Figure 2). This case cannot be handled with any existing sorting method, including Electre-Tri [52] and ELECTRE Tri-C [35], both of which require completely ordered classes (i.e. same order of classes on all criteria).

We now introduce this method by considering the set of *m* actions a_i , i = 1, ..., *m* to be assigned to one of the *k* predefined classes C_j, j = 1, ..$l_{.t}$, $l_{.t+1}$.., *k*, where *t* indicates incomparable classes of the same level *l*. ELECTRESort accepts a partial order structure on the classes and a complete ordered level structure upon it. A class belongs to a higher level than another class if its profile is Pareto superior (i.e. better on all criteria, e.g. in Figure 2, class 1 is on a higher level than class 2.1 and 2.2); otherwise they belong to the same level (e.g. in Figure 2, class 2.1 and 2.2). As levels are completely ordered, it is not possible to have outranking cycles as in ELECTRE III.These classes C$_j$ are characterized by a central profile cp_j defined on the n criteria z_h, h = 1, ..., n by the decision maker.

The outranking relation between each action a_i and all the central profiles cp_j are successively calculated. The action is then assigned to a class according to its preference, indifference and incomparable relations with the profiles. This is the main difference to previous sorting methods that disregarded incomparable relations.

In the next sections, the ElectreSort algorithm is explained in details. The construction of the outranking relations (sections 3.2.1- 3.2.6) is the same as in ELECTRE III.

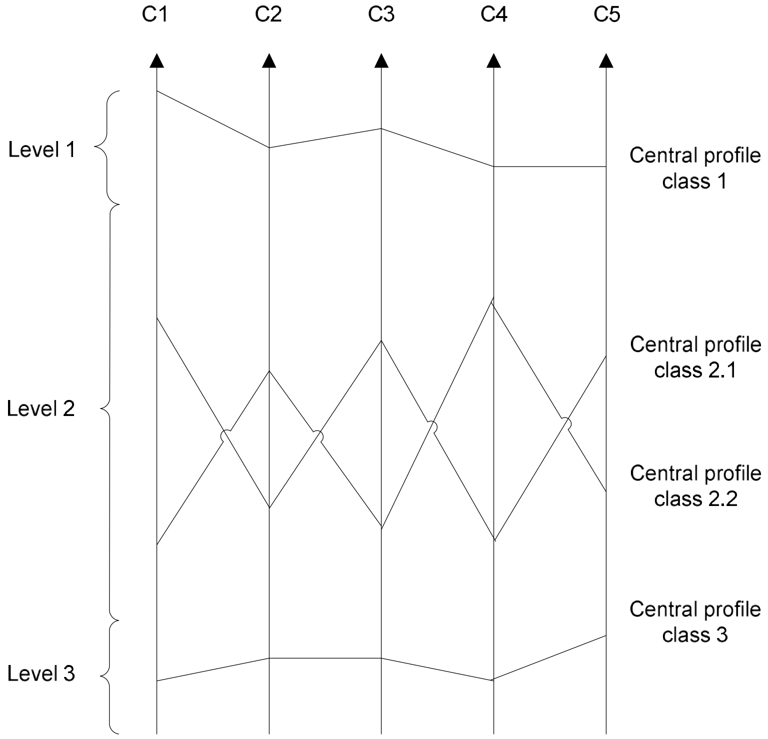


Figure 2: Definition of incomparable classes, i.e. Class 2.1 and Class 2.2, with their central profile performance on each criterion c$_{i.}$

3.2.1. Pseudo-criteria

True criteria used in ELECTRE I and II are the simplest and most traditional form of criterion. They do not have thresholds as only the difference between the scores on the criteria is used to determine which action is preferred. In order to take imprecision, uncertainty and indetermination of complex decision problems into account, all successively developed methods of the ELECTRE family use pseudo-criteria (Table 1). The indifference q

and preference *p* thresholds allow the construction of a pseudo-criterion. We can suppose, without loss of generality that all the criteria have to be maximised. Three relationships between action *a* and central profile *b* can be considered on each criterion *i*:

a) *a* and *b* are **indifferent** if the difference between the performance of the two actions is below the indifference threshold:

$$a \mathbf{I} b \leftrightarrow |z_i(a) - z_i(b)| \leq q \tag{1}$$

where $z_i(x)$: performance of the action *x* on criterion *i*
q: indifference threshold

b) *a* is **weakly preferred** to *b* if the difference between the performance of the two actions is in between the indifference and the preference threshold:

$$a \mathbf{Q} b \leftrightarrow q < z_i(a) - z_i(b) \leq p \tag{2}$$

where $z_i(x)$: performance of the action *x* on criterion *i*
q: indifference threshold
p: preference threshold of criterion *i*

c) *a* is **strictly preferred** to *b* if the difference between the performance of the two actions is higher than the preference threshold:

$$a \mathbf{P} b \leftrightarrow z_i(a) - z_i(b) \geq p \tag{3}$$

where $z_i(x)$: performance of the action *x*
p: preference threshold of criterion *i*

3.2.2. Concordance index

The concordance index C(*a,b*) (4) indicates the truthfulness of the assertion "*a* outranks *b*" noted *a* S *b*. C(*a,b*) = 1 indicates the full truthfulness of the assertion and C(*a,b*) = 0 indicates that the assertion is false. This concordance index aggregates the outranking information of all the criteria and is defined as follows:

$$\mathrm{C}(a,b) = \frac{1}{\mathrm{W}} \sum_{i=1}^{n} w_i \mathrm{c}_i\,(a,b) \tag{4}$$

where $\mathrm{W} = \sum_{i=1}^{n} w_i$ $\quad W = \sum_{i=1}^{n} w_i$

w_i: weight of criterion i

n: number of criteria

The concordance index $c_i(a,b)$ on the criteria i is given by:

$$\mathrm{c}_i(a,b) = \begin{cases} 1 & if\ z_i(b) - z_i(a) \le q_i \\ \frac{p_i(z_i(a)) + z_i(a) - z_i(b)}{p_i(z_i(a)) - q_i(z_i(a))} & if\ q_i < z_i(b) - z_i(a) < p_i \\ 0 & if\ z_i(b) - z_i(a) \ge p_i \end{cases}$$

$$\frac{\left[p_i(z_i(a) + z_i(a)\right] - z_i(b))}{} \; if\ q_i < z_i(b) - z_i(a) < p_i@\mathbf{0} \qquad if\ z_i(b) - z_i(a) \ge p_i$$

(5)

$z_i(x)$: performance of the action x in regard to the criterion i

q_i: indifference threshold for the criterion i

p_i: preference threshold of the action on the criterion i

3.2.3. Discordance index

If the difference of performance between the action a and b, on a criterion i, is higher than the veto threshold v_i, it is cautious to refuse the assertion "*a* outranks *b*". The discordance index for each criterion i is given in (6).

$$D_i(a,b) = \begin{cases} 0 & if\ z_i(b) - z_i(a) \le p_i \\ \frac{z_i(b) - [z_i(a) + p_i]}{v_i - p_i} & if\ p_i < z_i(b) - z_i(a) \le v_i \\ 1 & if\ z_i(b) - z_i(a) \ge v_i \end{cases}$$

$$\mathrm{D}_i(a,b) = \begin{cases} 0 & if\ z_i(b) - z_i(a) \leq p_i \\ \frac{z_i(b)-[z_i(a)+p_i]}{v_i-p_i} & if\ p_i < z_i(b) - z_i(a) \leq v_i \\ 1 & if\ z_i(b) - z_i(a) \geq v_i \end{cases}$$

$$D_i(a,b) = \begin{cases} 0 & if\ z_i(b) - z_i(a) \leq p_i \\ \dfrac{z_i(b) - [z_i(a) + p_i]}{v_i - p_i} & if\ p_i < z_i(b) - z_i(a) \leq v_i \\ 1 & if\ z_i(b) - z_i(a) \geq v_i \end{cases} \qquad (6)$$

where: $z_i(x)$: performance of the action x in regard to the criterion i

p_i: preference threshold of the action on the criterion i

v_i: veto threshold for the criterion i

3.2.4. Degree of credibility

Considering the global concordance (4) and discordance indices (6), the degree of credibility (7) indicates whether the outranking hypothesis is true or not. If the concordance index (4) is higher than or equal to the discordance index of all criteria (6), then the degree of credibility (7) is equal to the concordance index (4). If the concordance index (4) is strictly below the discordance index (6), then the degree of credibility (7) is equal to the concordance index (4) lowered in direct relation to the importance of those discordances.

$$S(a,b) = \begin{cases} C(a,b) & if\ D_i(a,b) \leq C(a,b) \forall i \\ C(a,b) \cdot \prod_{i \in J(a,b)} \dfrac{(1 - D_i(a,b))}{(1 - C(a,b))} & otherwise \end{cases}$$

$$S(a,b) = \begin{cases} C(a,b) & if\ D_i(a,b) \leq C(a,b) \forall i \\ C(a,b) \cdot \prod_{i \in J(a,b)} \dfrac{(1 - D_i(a,b))}{(1 - C(a,b))} & otherwise \end{cases}$$

$$S(a,b) = \begin{cases} C(a,b) & if\ D_i(a,b) \leq C(a,b) \forall i \\ C(a,b) \cdot \prod_{i \in J(a,b)} \dfrac{(1 - D_i(a,b))}{(1 - C(a,b))} & otherwise \end{cases}$$

$$S(a,b) = \begin{cases} C(a,b) & if\ D_i(a,b) \leq C(a,b) \forall i \\ C(a,b) \cdot \prod_{i \in J(a,b)} \dfrac{(1 - D_i(a,b))}{(1 - C(a,b))} & otherwise \end{cases}$$

$$S(a,b) = \begin{cases} C(a,b) & \text{if } D_i(a,b) \leq C(a,b) \forall i \\ C(a,b) \cdot \prod_{i \in J(a,b)} \dfrac{(1 - D_i(a,b))}{(1 - C(a,b))} & \text{otherwise} \end{cases}$$

$$S(a,b) = \begin{cases} C(a,b) & \text{if } D_i(a,b) \leq C(a,b) \forall i \\ C(a,b) \cdot \prod_{i \in J(a,b)} \dfrac{(1 - D_i(a,b))}{(1 - C(a,b))} & \text{otherwise} \end{cases} \qquad (7)$$

where J(a,b) $J(a,b)$ is the set of criteria for which $D_i(a,b) > C(a,b)$ $D_i(a,b) > C(a,b)$

3.2.5. Preference relations

The decision-maker needs to choose a credibility level λ, which indicates the minimum level to validate the statement "*a* outranks *b*". The credibility level λ is defined within the range 0.5 – 1, where a higher value means a higher selectivity to satisfy the outranking statement. Four possible relationships between the central profile cp_j of the class *j* and the actions a_i are possible:

- $a_i \gg cp_j$: a_i is preferred to cp_j if and only if $S(a_i, cp_j) \geq \lambda$ and $S(cp_j, a_i) < \lambda$
- $a_i \ll cp_j$: cp_j is preferred to a_i if and only if $S(a_i, cp_j) < \lambda$ and $S(cp_j, a_i) \geq \lambda$
- a_i I cp_j : a_i and cp_j are indifferent if and only if $S(a_i, cp_j) \geq \lambda$ and $S(cp_j, a_i) \geq \lambda$
- a_i R cp_j : a_i and cp_j are incomparable if and only if $S(a_i, cp_j) < \lambda$ and $S(cp_j, a_i) < \lambda$

An outranking graph (e.g. Figure 3) can be drawn based on these outranking relations. An arrow between an action and the central profiles indicates an outranking relationship. In a reduced version, the outranking relation is drawn only between immediate actions. This means that 'transitive' relations are not displayed. For example, in the reduced version, the dashed arrow in Figure 3 would not be represented. In this paper, we will only use the reduced outranking relations between the reference profiles notation for readability reasons. However, in some circumstances, for a better understanding, all the preference relations involving the action to classify are drawn, which leads to a semi-reduced graph (e.g. Figure 14). The assignment of an action to a class is based on this outranking graph, as explained in the next section.

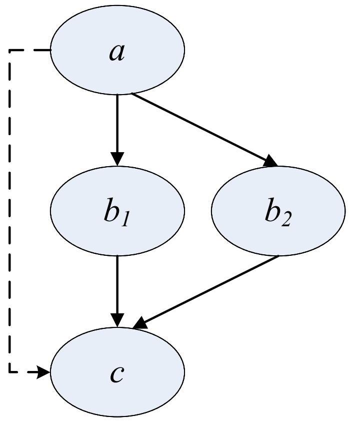


Figure 3 : Outranking graph where *a*→*b* means that *a* S *b*

3.2.6. Exploitation of the outranking relations: the sorting rules

Each action is placed into an outraking graph, where reference classes are ordered into levels. Several reference classes can belong to the same level only if they are incomparable (e.g. class 2.1 and 2.2 in Figure 4). The action can be indifferent to one or several classes, but they will always be on the same or successive level. For example, in Figure 4, it is impossible to have an action indifferent to class 1 and 3, while not being indifferent with class 2.1 or class 2.2. However, it is possible to have an action indifferent to the classes 1, 2.1 and 3. Therefore, we use conjointly two assignment rules: an optimistic (or descending) rule to capture the highest indifferent class and a pessimistic (or ascending) rule to capture the lowest indifferent class. This means that the process captures the two extreme possibilities. The difference between both assignment rules is the starting "point". The descending rule starts from the best class until the action is assigned. The pessimistic rule starts from the worst class.

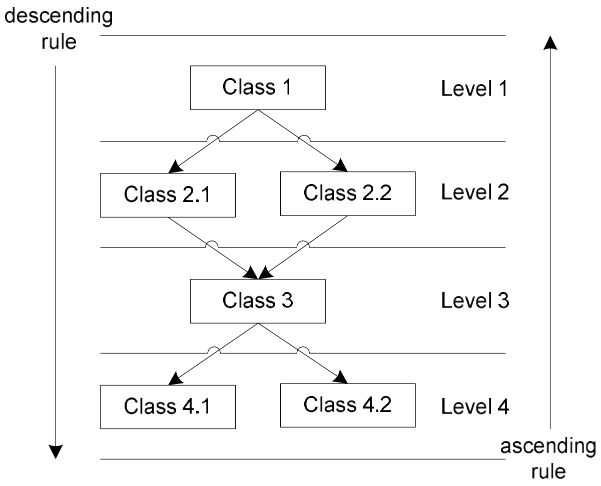


Figure 4 : Illustration of the class hierarchies

When comparing an action to a profile, there are 3 different outranking relations which lead to three assignment possibilities. We now look successively at each class level and:

a) If an action is **indifferent** to at least one profile of a class, then the action is assigned to this class . It is to be noted that an action can be indifferent to several classes of the same level or *successive* levels. If all classes are on the same level (e.g. Figure 5), then the action is assigned to all indifferent classes of the profile (i.e. class 2.1 and 2.2 in the case of Figure 5). If an action is indifferent to classes on different levels (e.g. Figure 6), then the action is assigned only to the indifferent classes of the first encountered level (i.e. in the case of Figure 6, it is assigned to class 1 with the optimistic approach; and to class 2.2 with the pessimistic approach). This justifies that two approaches are used.

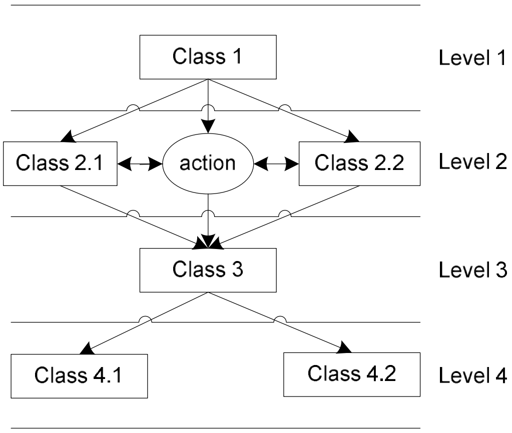


Figure 5: Action is indifferent to several classes of the same level

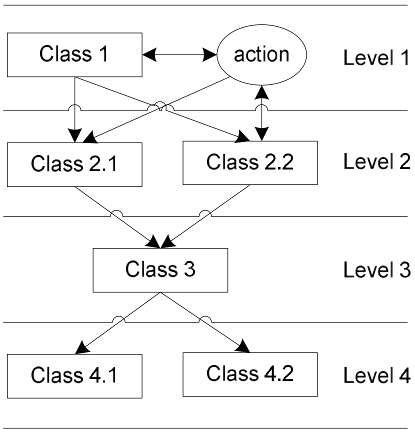


Figure 6: Action is indifferent to several classes of different levels

b) If an action a_i is **incomparable** to at least one central profile (Figure 7) but with no indifference relationship with any profile. Several different processes are possible:

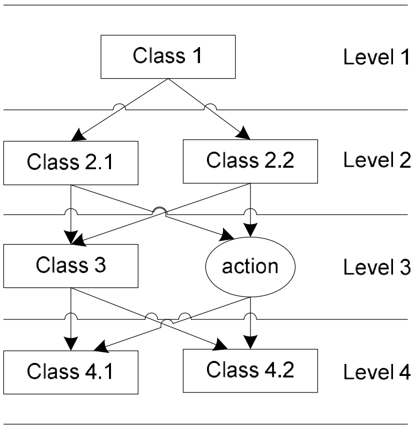


Figure 7 : Action is incomparable to class 3

i. A new class is discovered. The decision maker can use this new information acquired at the end of the exploitation phase, where the incomparable action is used as a reference profile for this newly discovered class. The assignment process is rerun again from the beginning on all the actions compared with all the profiles including this new one. If several incomparable actions on the same level appear, it must be checked whether actions are incomparable between themselves before adding all of them to the set of profiles. Otherwise, there will be two reference profiles for the same class. Adding classes in the middle of the exploitation phase without first terminating the process would imply that the final result is dependent on the order in which the actions are sorted; therefore this possible error must be avoided.

   This treatment fully considers the incomparability and preference relationships between the actions and the profiles. However, it does not always make sense to create a new class when there is no corresponding strategy. Therefore, an assignment which is described in the next paragraph is proposed.

ii. If the decision maker does not want to create a new class, then it is possible to assign the action as follows:

   a. to an existing class of the same level, e.g. class 3 in Figure 7. It is a neutral view, thus we do not consider the incomparability of classes.

   b. to the subsequent higher (worse) class(es). It is a pessimistic view, where we do not consider the outranking relationship between the action and the reference profile of the higher class(es), e.g class 4.1 and 4.2 in Figure 7.

   c. to the preceding lower (better) class(es). It is an optimistic view, where we do not consider the outranking relationship between the action and the reference profile of the lower class(es), e.g. class 2.1 and 2.2 in Figure 7.

   It is worth noting that if a new class is not created, one preference relationship must be violated: incomparability in the case iia, outranking in iib or being outranked in iic.

   In any case, it is important to highlight the fact that a new class might have been discovered. A discussion with the decision-maker is thus recommended in order to adopt the best assignment procedure.

c) If an action is **neither indifferent** nor **incomparable** to any of the central profiles, but is **in between two classes of successive levels** (Figure 8): In the optimistic assignment, the action is allocated to the first class *j* that is preferred over the action (e.g. class 1 in the case of Figure 8). In the pessimistic assignment, it is allocated to the first class *j*+1 over which the action is preferred to (e.g. class 2.1 and 2.2 in the case of Figure 8).

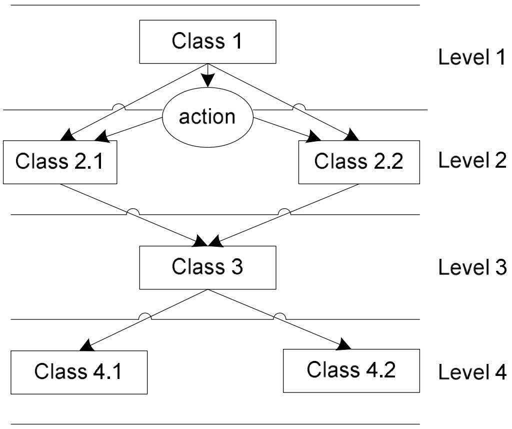


Figure 8: Action in between two levels

The algorithm for the descending assignment will look successively for indifference, incomparability and action between two levels. The creation of a new class in case of incomparability can be replaced by another of the three assignment processes described above in b.ii).

```
FOR ALL action $a$ to be assigned:
    FROM best level i=1 UNTIL the worst level i=K
        IF $a$ indifferent at least to one class of level $i$ THEN:
            Assign $a$ to the all classes of level $i$, that are indifferent to $a$
            Exit
        IF $a$ incomparable at least to one class of level $i$ THEN:
            Method i: define a new class at level $i$ and restart the complete procedure
            Method ii.a : Assign $a$ to all classes of level i
            Method ii.b : Assign $a$ to all classes of level i+1
            Method ii.c : Assign $a$ to all classes of level i-1
            Exit
        IF $a \gg$ profile of level $i+1$ and $a \ll$ profile of level $i-1$ THEN:
            Assign a to the classes of the profiles of level i to which a ≫ profile
            Exit
        IF level i=K:
            Assign a to all classes of level i
            Exit
    END FOR
End For ALL actions.
```

Pseudo-Code 1: Assignment rules in the descending scenario

The algorithm for the ascending assignment is identical except that we start from the worst class moving up to the best class.

3.2.7. Properties of ELECTRE Sort

ELECTRESort has been conceived in order to fulfil the fundamental properties of a sorting method. In the following section, we describe the main properties of the ELECTRESort method. Complete proofs can be found in appendix.

- *Independence*: The assignment of an action a_i does not depend on the assignment of any other action a_j.

  Proof: As an action a_i is compared uniquely to the central profiles and the assignment rules are based solely on these comparisons, the assignment of a_i is thus independent of those of action a_j.

- *Homogeneity*: When two actions are compared similarly to the reference profiles, they are assigned to the same class

  Proof: As the assignment is based solely on the comparison with the reference profiles, if two actions are compared similarly to them, they will be assigned to the same class.

- *Monotonicity*: If action a_i dominates a_j then a_i is assigned at least to the same class that a_j is assigned to.

Proof: If a_i dominates a_j, then, given the dominance condition on the reference profiles, we have that S(a_i, cp_j)≥S(a_j, cp_j) and S(cp_j,a_i)≤S(cp_j,a_j). This ensures us that a_i is preferred at least as much as a_j with regards to the central profile. Therefore, a_i is assigned to the same or better class than a_j is assigned to.

- *Stability*: The fusion or separation of two neighbouring levels does not affect the previous assignment of the actions to the non-modified classes.

  Proof: The fusion or separation of neighbouring levels does not affect the preference relationships of the actions assigned to the 'non-modified' classes and their central profiles. Therefore, the assignment remains unchanged.

- *Conformity*: Each central profile is univocally assigned to its corresponding class with both assignment rules.

  Proof: If the outranking graph of the central profiles does not contain any indifference (which should be avoided), then each central profile is univocally assigned to its corresponding class with both assignment rules

- *Uniqueness*: Every action a_i is assigned to one class according to one of the procedures.
  This property is not respected since an action can be assigned to several classes of a same level if the action is indifferent to several central profiles.

3.3. Comparison with ELECTRE-Tri-C

This section compares ELECTRESort with ELECTRE-Tri-C [35, 55], which are both based on the concept of central reference profiles. As ELECTRE-Tri-C is not able to treat incomparable classes, we will only analyse the cases where the classes are completely ordered. The assignment algorithm of ELECTRESort is described in the section 3.2.6. ELECTRE-TRI-C has the following assignments (see the Theorem 2 in [35]):

- Preference relation ($a \gg cp_h$, $cp_{h+1} \gg a$): a can be assigned to C$_h$ or C$_{h+1}$. Ascending and descending assignment rule does not necessarily assign to the same class.

- Indifference relation ($a \gg cp_{h-1}$, a I cp_h, a I $cp_{h+1},\ldots,$ a I cp_s, $cp_{s+1} \gg a$): For the descending rule, C_h or best adjacent C_{h+1} can be selected. For the ascending rule, C_s or the worst adjacent C_{s-1} can be selected.
- Incomparability relation ($a \gg cp_{h-1}$, a R cp_h, a R $cp_{h+1},\ldots,$ a R cp_s, $cp_{s+1} \gg a$): For the descending rule, C_h or C_{h-1} can be selected. For the ascending rule, C_s or the worst adjacent C_{s+1} can be selected.

In addition to these three situations, we have added six cases that might happen when compared with the best and worst classes (boundary cases). These nine different situations are given in Table 2.

	Situations	ELECTRE-Tri-C	ELECTRESort
I.	$cp_1 \gg a, cp_2 \gg a,\ldots, cp_j \gg a, a\gg cp_{j+1}, a \gg cp_{j+2}, \ldots, a\gg cp_k$	$C_{opt} = C_j$ or C_{j+1} $C_{pess} = C_j$ or C_{j+1}	$C_{opt} = C_j$ $C_{pess} = C_{j+1}$
II.	$cp_1 \gg a, cp_2 \gg a,\ldots, cp_{j-1} \gg a, a$ I $cp_j, a\gg cp_{j+1}, \ldots, a\gg cp_k$	$C_{opt} = C_j$ or C_{j+1} $C_{pess} = C_j$ or C_{j-1}	$C_{opt} = C_j$ $C_{pess} = C_j$
III.	$cp_1 \gg a, cp_2 \gg a,\ldots, cp_{j-1} \gg a, a$ R $cp_j, \ldots, a$ R cp_{j+h-1}, a R $cp_{j+h}, a \gg cp_{j+h+1}, \ldots a\gg cp_k$	$C_{opt} = C_j$ or C_{j-1} $C_{pess} = C_{j+h}$ or C_{j+h+1}	$C_{opt\ 1}$ = define a new classes for each incomparable relation $C_{opt\ 2} = C_j$ $C_{opt\ 3} = C_{j+1}$ $C_{opt\ 4} = C_{j-1}$ $C_{pess\ 1}$ = define a new classes for each incomparable relation $C_{pess\ 2} = C_{j+h}$ $C_{pess\ 3} = C_{j+h+1}$ $C_{pess\ 4} = C_{j+h-1}$
IV.	$a \gg cp_1, \ldots, a\gg cp_j,$	$C_{opt} = C_1$ or C_2 $C_{pess} = C_1$ or C_2	$C_{opt} = C_1$ $C_{pess} = C_1$

	…, $a \gg cp_k$		
V.	a R cp_1, …, a R cp_{j-1}, a R cp_j, …, $a \gg cp_k$	$C_{opt} = C_1$ $C_{pess} = C_j$ or C_{j+1}	$C_{opt\ 1}$ = define a new classes for each incomparable relation $C_{opt\ 2} = C_1$ $C_{opt\ 3} = C_2$ $C_{pess\ 1}$ = define a new classes for each incomparable relation $C_{pess\ 2} = C_j$ $C_{pess\ 3} = C_{j+1}$ $C_{pess\ 4} = C_{j-1}$
VI.	a I cp_1, $a \gg cp_2$, …, $a \gg cp_j$, …, $a \gg cp_k$	$C_{opt} = C_1$ or C_2 $C_{pess} = C_1$	$C_{opt} = C_1$ $C_{pess} = C_1$
VII.	$cp_1 \gg a$, $cp_2 \gg a$,…, $cp_j \gg a$, …, $a \gg cp_k$	$C_{opt} = C_k$ $C_{pess} = C_k$	$C_{opt} = C_k$ $C_{pess} = C_k$
VIII.	$cp_1 \gg a$, $cp_2 \gg a$,…, $cp_{j-1} \gg a$, a R cp_j, …, a R cp_k	$C_{opt} = C_j$ or C_{j-1} $C_{pess} = C_k$	$C_{opt\ 1}$ = define a new classes for each incomparable relation $C_{opt\ 2} = C_j$ $C_{opt\ 3} = C_{j+1}$ $C_{opt\ 4} = C_{j-1}$ $C_{pess\ 1}$ = define a new classes for each incomparable relation $C_{pess\ 2} = C_k$ $C_{pess\ 3} = C_{k-1}$
IX.	$cp_1 \gg a$, $cp_2 \gg a$,…, $cp_{k-1} \gg a$, cp_k I a	$C_{opt} = C_k$ $C_{pess} = C_k$ or C_{k-1}	$C_{opt} = C_k$ $C_{pess} = C_k$

Table 2 : Table of comparison ELECTRE-Tri-C/ELECTRESort

Let us at the outset highlight that the 'ELECTRE-Tri-C column' gives *possible* assignments (see Theorem 2 in [35]) based on the preference relation and not the outranking degrees (as the method would do). This explains why when using the ELECTRE-Tri-C the assignment of each procedure in Table 2 may not be univocal, in the sense that two possible classes may be possible and additional technical parameters are needed for the assignment.

When comparing the assignment between ELECTRESort and ELECTRE-TRI-C, we may conclude that a possible drawback of using outranking degrees is that some results might be difficult to explain to a decision maker. For instance, in situation II, although an action is indifferent to exactly one central profile, ELECTRE-Tri-C might not assign it to that class (this depends on the outranking degree and the lambda value).

Another difficult case to justify might be the following scenario: Considering the problem where suppliers are sorted into three classes: 'good', 'medium' and 'bad'. If we suppose that supplier A and B are both preferred to the central profile of the class 'medium', it seems difficult to explain to a decision maker that ELECTRE-Tri-C may assign a supplier A to class ‘good’ and another supplier B to class ‘medium’.

Furthermore, one may notice that the incomparability is handled in different ways, according to the different procedures of ELECTRESort (as explained in Section 3.2.6). However, each procedure follows the same behaviour in the nine situations as seen in Table 2.

4. Illustrative example

We consider the manufacturing company referred to in the introduction ([19, 20, 22], which operates 130 different machines. Only the machines producing 89% of the downtime have been reported in the original paper. The others are considered to be unproblematic and a sophisticated maintenance strategy was thus not needed. The next paragraph will firstly present different maintenance strategies. Then, the criteria and the reference profiles used for sorting the machines are described. Finally, the results will be discussed.

4.1. Maintenance strategies

Maintenance strategies can be divided into reactive and proactive maintenance [56]. Reactive maintenance strategies require an action only when the failure or breakdown has occurred. Therefore, it leads to more reparative actions. In a proactive maintenance approach, breakdowns are expected to be avoided through preventive maintenance activities. This means thus that maintenance activities are organised and performed even if the machine is working smoothly.

Described below, the most widely adopted maintenance strategies are used to define our classes:

Reactive Maintenance:

- *Corrective Maintenance (CM)*: It is also named Operate To Failure (OTF), failure based maintenance, or breakdown maintenance. In this strategy, corrective maintenance is conducted only when a facility breaks down. This is the original strategy that was adopted in industry [57]. Today, this strategy is appropriate for assets with rare breakdown, those that are easy to repair or replace, and those with minimal impact consequences (e.g. light bulb). The downtime, the spare parts costs, and the bottleneck impacts are low. In this post-failure strategy, the maintenance team needs to repair the facility as urgently as possible.

- *Skill Level Upgrade (SLU)*: This strategy requires upgrading the skills of the operator, (e.g. through training), or upgrading the machine (e.g with visual signs, displays, etc) in order to make the maintenance service and repair of the machine easier. This strategy is used for frequent failures, which can be fixed quickly (e.g. paper jam in a photocopier machine, where advanced assistance is provided through electronic displays and coloured levers). The spare parts costs are very low and the machine does not have a critical role in the production process.

Proactive Maintenance:

- *Fixed Time Maintenance (FTM)*: In this strategy, maintenance is scheduled in advance to prevent a failure. A series of checks and component replacements are performed at a determined date. This strategy assumes that component life is predictable, for example with constant erosion or corrosion. The time unit used to plan a maintenance schedule is either the calendar time (e.g. annual gas boiler check) or the component running time (e.g. tyres of a car). This strategy facilitates the management of maintenance activities and inventory control. However, in spite of preventive maintenance, some failures may occur because interventions are planned according to a failure probability distribution and not according to the individual machines [58]. It may be an uneconomical strategy because still serviceable parts are replaced prior to the expiry of their usable life.
- *Condition Based Maintenance (CBM)*: This strategy implies the existence of data acquisition systems, generally sensors, which measure machine performance (e.g. vibration, ultrasonic test, thermal monitoring, etc) in real time. This continuous monitoring can detect abnormal situations, alerting the maintenance team to perform a control or a replacement activity. It is therefore the condition of the machine that triggers an intervention. This strategy infers a high reliability to the system and reduces the maintenance costs because only necessary services are operated (unlike in the FTM strategy). However, it has high infrastructure installation costs and unpredictable maintenance periods [59]. The machine does not often break down but typically needs a long time to be repaired, the spare parts have a rather high cost and the intervention causes

a bottleneck. This preventive maintenance has generated considerable interest in reliability centred maintenance [60], where fault is not tolerable, e.g. nuclear plant, airplanes, etc.

- *Design-Out Maintenance (DOM)*: This approach aims to eliminate causes of failure or to minimise the need of maintenance. It is appropriate for machines with frequent failures that are difficult to repair, with high maintenance costs, and having a critical role in production. In this case, the intervention is at the design stage of the machine's development and not during its utilisation [61].

4.2. Criteria definition

The original maintenance selection strategy was based on only two criteria [19, 20]:

- *Downtime*: time of unavailability of the machine in hours.
- *Frequency*: number of maintenance calls.

In order to make a more informed decision, two other criteria are added for evaluation:

- *Spare part costs*: cost of ordering, stocking and replacing a part.
- *Bottleneck*: number of batch loadings missed at the end of the production line, i.e. a higher demanded machine will have a higher bottleneck.

The data used in [19, 20] can be found in the first two columns of Table 3. Adding to this, we have completed the table with plausible spare parts costs and bottleneck data.

	Downtime [hours]	Frequency [#]	Spare parts costs [£]	Bottleneck [# loadings]
Machine A	30	9	62.5	6
Machine B	20	3	4.2	27
Machine C	20	16	25	12
Machine D	17	12	20.5	15
Machine E	16	8	21	7.5
Machine F	12	4	21	22
Machine G	7	27	12.5	10
Machine H	6	2	100	29
Machine I	6	8	8.3	51
Machine J	4	7	12.5	19

Machine K	3	8	50.4	27

Table 3 : Data Breakdown

4.3. Profile definition

Five maintenance strategies were presented in Section 4.1. The most effective strategy is to *Design-Out Maintenance,* which definitively eliminates the problem. However, this strategy may imply high costs. Therefore, a more efficient strategy can be recommended. According to the descriptions in Section 4.1, a typical efficient example for each class is defined and presented in Table 4. In ElectreSort they are called central profiles.

	Downtime [hours]	Frequency [#]	Spare parts costs [£]	Bottleneck [# loadings]
OFT	4	2	10	7
DOM	25	25	70	30
CBM	25	2	50	30
FTM	14	16	20	7
SLU	4	20	10	20
Weights	1	1	1	1
Indifference threshold	4	2	5	7
Preference threshold	15	16	20	15
Veto threshold	20	20	40	25

Table 4 : Central profiles

We note here that the maintenance profiles of Table 4 are *naturally* ordered on each criterion but that this order is not the same for all criteria: they are partially ordered. The traditional ELECTRE-Tri, which requires a complete order, cannot be used in this case. The profile of the *Operate to Failure* strategy is the best and the *Design Out Maintenance* strategy is the worst. The performance of the other profiles cross each other as can be seen in Figure 9.

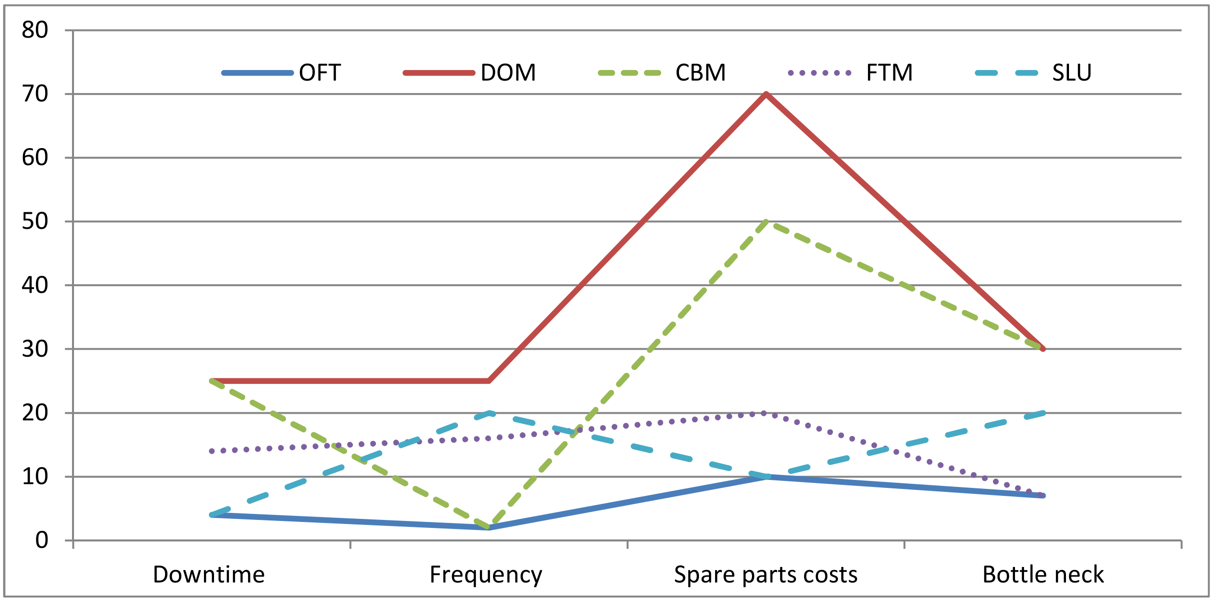


Figure 9 : Typical profiles of the classes

The credibility matrix of the profiles is given in Table 5. According to the different credibility level λ, different reduced outranking graphs can be drawn (Figure 10 to Figure 12). As the credibility index is a technical parameter, it is important to examine, often in conjunction with the decision-maker, which value produces the most realistic outranking graph. The outranking graphs with a $\lambda \leq 0.78$ (Figure 12 and Figure 13) show that the strategies Skill Level Upgrade and Fixed Time Maintenance are indifferent. This is not possible as these two strategies are clearly not interchangeable. Figure 11 indicates that the Skill Level Upgrade is incomparable to the Fixed Time Maintenance and the Condition Based Maintenance; and the Fixed Time Maintenance outranks the Condition Based Maintenance. This is not impossible and it could be good reasons to consider FTM>>CBM, whereas SLU is incomparable to both. However, this case is only valid for a credibility level of exactly 0.79. Therefore, the most plausible outranking graph is one with a threshold of the credibility level $\lambda = 0.8$ (or higher) represented in Figure 10. From this graph, it can be seen that the strategies Skill Level Upgrade, Fixed Time Maintenance and Condition Based Maintenance are incomparable. This observation was also evidenced in the Decision Making Grid (Figure 1).

It is to note that if no satisfactory outranking graph is found (e.g. profile having a cycle), the profiles needs to be reviewed by the decision-maker. Another way to change the outranking graph in the direction sought by the decision-maker is to fine tune the parameter values (weights, thresholds).

	OTF	FTM	DOM	CBM	SLU
OTF	1	**1**	**1**	**1**	**1**
FTM	0.57	1	**1**	0.79	0.78
DOM	0	0	1	0	0
CBM	0	0.079	**1**	1	0.41
SLU	0.56	0.78	**1**	0.75	1

Table 5 : Credibility matrix of the profiles (bold are credibility degrees above the credibility level λ = 0.8)

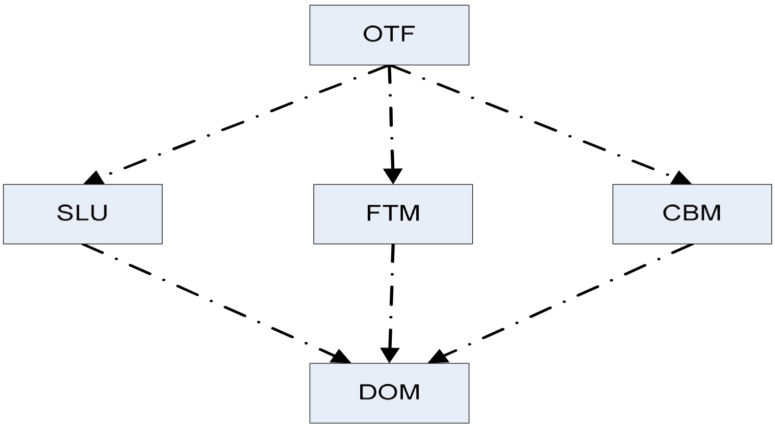


Figure 10 : Reduced graph of the partially ordered classes representing the different maintenance strategies with a credibility level λ ≥ 0.8

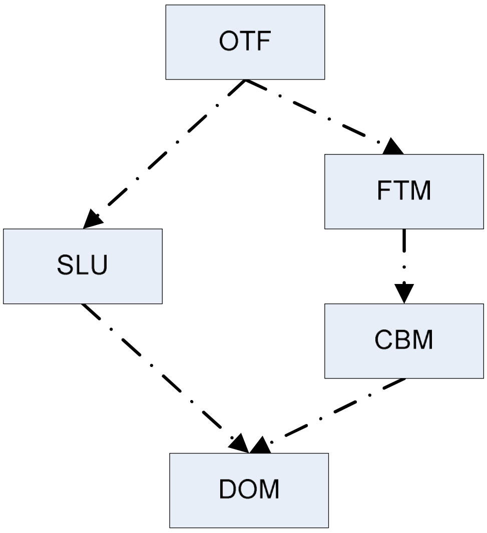


Figure 11 : Reduced graph of the partially ordered classes representing the different maintenance strategies with a credibility level $\lambda = 0.79$

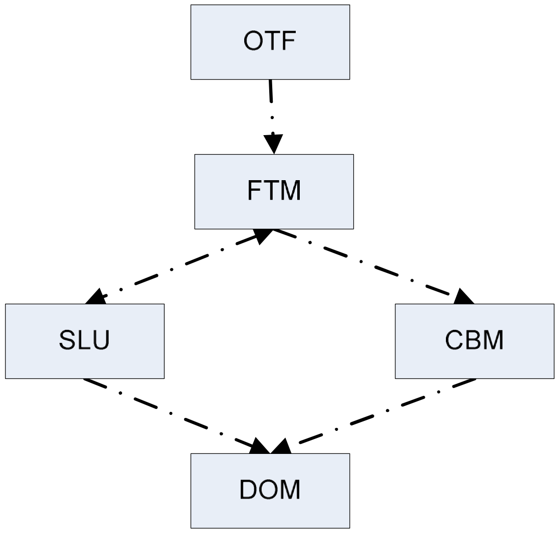


Figure 12: Reduced graph of the partially ordered classes representing the different maintenance strategies with a credibility level $\lambda \in [0.76\text{-}.078]$

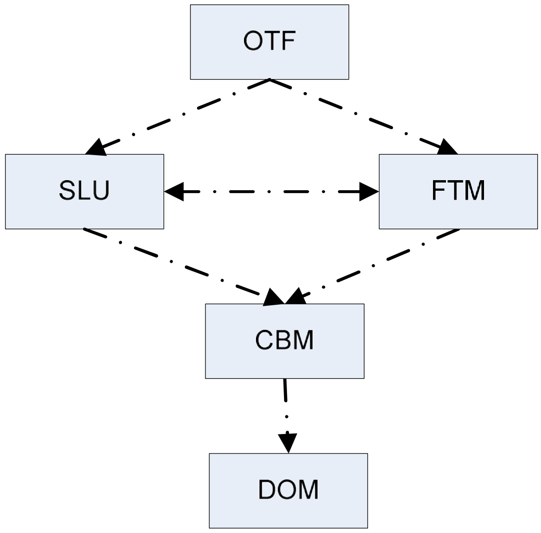


Figure 13: Reduced graph of the partially ordered classes representing the different maintenance strategies with a credibility level $\lambda \leq 0.75$.

4.4. Results

Each machine is compared individually to all central profiles simultaneously (Table 4) with the ElectreSort algorithm described in Section 3.2. Table 6 contains the credibility outranking degrees between the reference profiles and the machines.

Machine	S(M,DOM)	S(DOM,,M)	S(M,CBM)	S(CBM,M)	S(M,SLU)	S(SLU,M)	S(M,FTM)	S(FTM,M)	S(M,OTF)	S(OTF,M)
A	**0.98**	0.085	0.76	0.3	0	0.62	0	**0.91**	0	**1**
B	**1**	0	**1**	0	0.75	0.74	0.7	0.62	0.5	**0.99**
C	**1**	0	0.79	0.48	0.58	**0.93**	**0.95**	**1**	0.37	**1**
D	**1**	0	**0.86**	0.36	0.7	**0.89**	**0.97**	**0.96**	0.53	**1**
E	**1**	0	**0.93**	0.14	0.72	0.65	**1**	**0.89**	0.65	**1**
F	**1**	0	**1**	0.51	**0.81**	0.75	0.75	**0.82**	0.56	**1**
G	**1**	0	0	0.015	**0.91**	**0.91**	**0.84**	**0.89**	0	**1**
H	0.75	0	0	0.6	0	0.75	0	0.69	0	**1**
I	0.75	0	0.68	0	0	**0.82**	0	0.69	0	**1**
J	1	0	**0.95**	0	**1**	**0.8**	**0.84**	0.7	0.79	**1**
K	**1**	0	**0.93**	0	0	**0.82**	0.48	0.73	0	**1**

Table 6 : Credibility matrix (bold are credibility degrees above the credibility level λ = 0.8)

In Table 7, the preference relation between the machines and the reference profiles and the resulting assignments are presented (see notation in section 3.2.5).

Machine	DOM	CBM	SLU	FTM	OTF	C_{opt}	C_{pess}	Original strategy in [19]
A	≫	R	R	≪	≪	OTF	FTM	FTM
B	≫	≫	R	R	≪	OTF	CBM	CBM
C	≫	R	≪	I	≪	FTM	FTM	DOM
D	≫	≫	≪	I	≪	FTM	FTM	FTM
E	≫	≫	R	I	≪	FTM	FTM	FTM
F	≫	≫	≫	≪	≪	CBM SLU	FTM	FTM
G	≫	R	I	I	≪	SLU FTM	SLU FTM	SLU
H	R	R	R	R	≪	DOM	OTF	OTF

I	R	R	≪	R	≪	DOM	FTM	FTM
J	R	≫	I	≫	≪	SLU	SLU	N/A
K	≫	≫	≪	R	≪	CBM	SLU	N/A

Table 7 : Preference relation and assignments

The results of Table 7 should be seen as the basis for an interactive discussion with the decision-maker. In the following paragraphs, we will discuss some specific cases.

The machine A (Figure 14) is in between two levels: OTF and FTM. As it has a low breakdown frequency and bottleneck (Table 3) and if we believe that the high downtime and costs are supportable, then we operate until failure. Otherwise, if we estimate that the high downtime and spare parts costs are too high, we can adopt the FTM strategy.

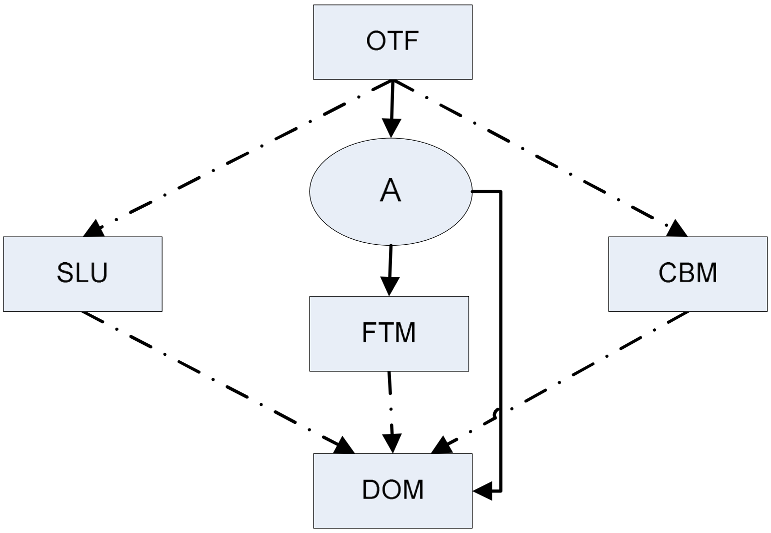


Figure 14 : Preference relation of machine A (arrow means outranks)

Machine B (Figure 15) is also between two strategies:OTF and CBM. The breakdown frequency and the spare parts costs are low but the downtime and the bottleneck are high (Table 3). If we consider the high downtime and bottleneck being not problematic, then we can adopt a OTF strategy. At the contrary, if we believe that they are critical, then a CBM strategy should be adopted.

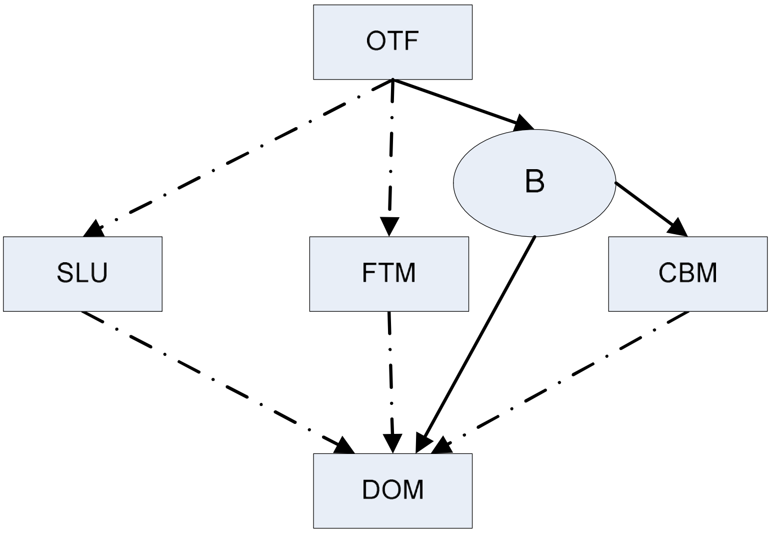


Figure 15 : Preference relation of machine B (arrow means outranks)

Machine G (Figure 16) is indifferent to two strategies: SLU and FTM, which suggests that a mixed strategy should be applied. As the frequency of breakdown is high (Table 3), a regular control should be done (FTM strategy). As the downtime is low, the machine could be repaired by upgrading the skill level.

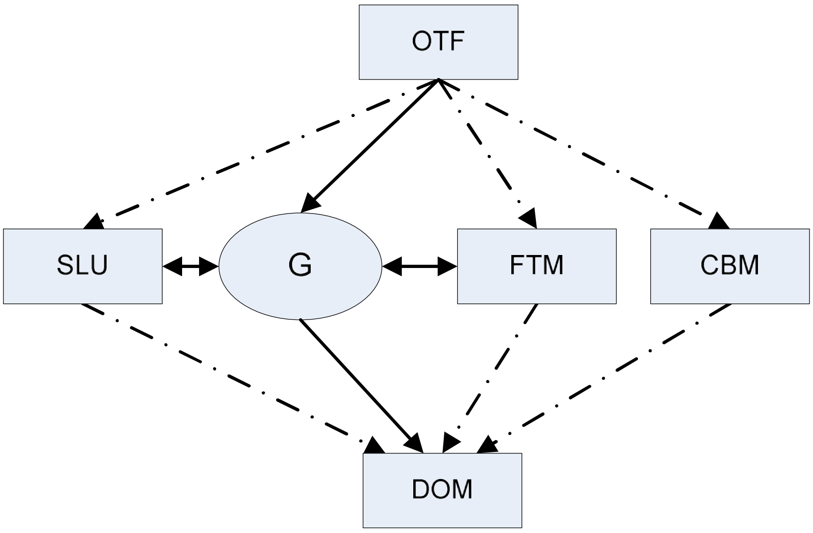


Figure 16 : Preference relation of machine G (arrow means outranks)

As in any other decision problem, a sensitivity analysis must be performed by varying the parameters of the method such as weights and thresholds. Therefore, we changed the values of the weights, indifference, preference and veto threshold up to 20% from their initial value. In our case, we have found that overall the results are robust to those changes (Table 8, Table 9, Table 10, Table 11). The assignments of the machines A, B, D, E, G and H are identical after the variations in the weights and thresholds . Changes in the indifference threshold have only little effect for machine J, while variations on the veto threshold do not have any implication (Table 11). On the other hand, Machine J is the most sensitive to weight variations. From Table 10, it seems that the results are the most dependent on the preference threshold since the changes impact several machines.

The robustness of ELECTRESort may not be surprising as the construction of the outranking relations is identical than in ELECTRE III (only its exploitation algorithm is different) and it has been already seen in previous works that the methodology leads to robust results [62].

	Weight Downtime -20%	Weight Downtime +20%	Weight Frequency -20%	Weight Frequency +20%	Weight Spare parts costs -20%	Weight Spare parts costs + 20%	Weight Bottleneck -20%	Weight Bottleneck +20%
A	no	no	no	no	no	no	no	no
B	no	no	no	no	no	no	no	no
C	no	**yes**	**yes**	no	no	**yes**	no	**yes**
D	no	no	no	no	no	no	no	no
E	no	no	no	no	no	no	no	no
F	no	no	no	no	no	no	no	no
G	no	no	no	no	no	no	no	no
H	no	no	no	no	no	no	no	no
I	no	no	no	no	no	no	no	no
J	**yes**	**yes**	no	**yes**	**yes**	**yes**	**yes**	**yes**
K	no	no	no	no	no	no	no	no

Table 8 : Classification changes with ±20% variation of the weights ("no" means that classification does not change)

	Indifference Downtime -20%	Indifference Downtime +20%	Indifference Frequency -20%	Indifference Frequency +20%	Indifference Spare parts costs	Indifference Spare parts costs	Indifference Bottleneck -20%	Indifference Bottleneck +20%

					-20%	+ 20%		
A	no	no	no	no	no	no	no	no
B	no	no	no	no	no	no	no	no
C	no	no	no	no	no	no	no	no
D	no	no	no	no	no	no	no	no
E	no	no	no	no	no	no	no	no
F	no	no	no	no	no	no	no	no
G	no	no	no	no	no	no	no	no
H	no	no	no	no	no	no	no	no
I	no	no	no	no	no	no	no	no
J	no	no	no	**yes**	no	no	no	**yes**
K	no	no	no	no	no	no	no	no

Table 9 : Classification changes with ±20% variation of the indifference threshold

	Preference Downtime -20%	Preference Downtime +20%	Preference Frequency -20%	Preference Frequency +20%	Preference Spare parts costs -20%	Preference Spare parts costs + 20%	Preference Bottleneck -20%	Preference Bottleneck +20%
A	no	no	no	no	no	no	no	no
B	no	no	no	no	no	no	no	no
C	no	no	no	**yes**	no	no	no	no
D	no	no	no	no	no	no	no	no
E	no	no	no	no	no	no	no	no
F	**yes**	no	**yes**	**yes**	no	no	no	**yes**
G	no	no	no	no	no	no	no	no
H	no	no	no	no	no	no	no	no
I	no	no	**yes**	no	no	no	no	no
J	no	no	no	**yes**	no	no	**yes**	**yes**
K	no	no	**yes**	no	no	no	no	no

Table 10 : Classification changes with ±20% variation of the preference threshold

	Veto Downtime	Veto Downtime	Veto Frequency	Veto Frequency	Veto Spare parts costs	Veto Spare parts costs	Veto Bottleneck	Veto Bottleneck

	-20%	+20%	-20%	+20%	-20%	+ 20%	-20%	+20%
A	no	no	no	no	no	no	no	no
B	no	no	no	no	no	no	no	no
C	no	no	no	no	no	no	no	no
D	no	no	no	no	no	no	no	no
E	no	no	no	no	no	no	no	no
F	no	no	no	no	no	no	no	no
G	no	no	no	no	no	no	no	no
H	no	no	no	no	no	no	no	no
I	no	no	no	no	no	no	no	no
J	no	no	no	no	no	no	no	no
K	no	no	no	no	no	no	no	no

Table 11 : Classification changes with ±20% variation of the veto threshold

5. Discussion

The maintenance of machines may have incomparable strategies (see section 4.3). In order to take this preference relationship into account, the literature has proposed the use of a Decision Making Grid [7, 19] to assign appropriate maintenance strategies to machines. Apart from being restricted to *two* criteria, the Decision Making Grid has another major disadvantage: it is built with four straight rigid lines, which leads to nine identical squares used for defining classes. This inherent assumption is inappropriate and unrealistic. The Decision Making Grid leads itself into trying to manipulate the modelling of the problem in order to "fit it" into the method instead of vice versa. As a result, the four top squares of the Decision Making Grid have been filled with clear distinct strategies but all other squares have the common strategy: Total Productive Maintenance (TPM) in [20, 22] and FTM in [19, 23, 28]. This change of strategy by the same author has never been justified. It is also surprising that a machine with low downtime and medium breakdown frequency has the same strategy as a machine that has a high downtime and a medium breakdown frequency. As a result, most of the machines are assigned to the FTM strategy (Table 7), which seems to be a holdall strategy.

ElectreSort, the method introduced in this paper, has a different approach, which in our opinion is more logical. The strategies and then their typical profile are first defined. Therefore, the number of strategies and the criteria depends only on the problem and are not fixed by the method as in the Decision Making Grid. Moreover, ElectreSort has a more flexible and interactive approach. Several or even a mix of strategies can be recommended. The early involvement of the maintenance manager is fundamental [63, 64]. Different views in the assignment can be proposed and then discussed, even on a case by case basis, with the maintenance manager. For example, if an action is assigned to several classes, a discussion with the decision-maker must be conducted to decide which of the strategy must be selected or if a mix of strategies can be implemented. Clearly some strategies are not compatible, it is therefore important to have this discussion with the decision-maker.

The main drawback of ELECTRESort is its complexity. In case, where a decision needs to be taken quickly without the presence of the decision-maker, then the Decision Making Grid (that can be expressed as business rules) is sufficient for an approximate result.

Finally, ElectreSort is a generic method that can be applied in other sectors for sorting, where incomparability exists. For example, it can be used for sorting employees with different skills, and therefore incomparable, on a pay scale. Two classes of employees having different skills can be on the same pay scale. Another application could be on the sorting of products in classes of importance. Two classes of products can be on the same importance of level but for different reasons: the first class has a large market share but a low benefit due to its low margin revenue, the second class has a low market share but generate a high benefit.

6. Conclusion

The pressure on manufacturing organisations to be competitive and to provide timely deliveries of quality products has been recently intensified under the spotlight of austerity and the need to remain competitive. Adopting the right maintenance policy is a key element to achieve these objectives. It contributes to reducing costs, minimising machine downtime, and improving quality and productivity whilst ensuring safety. With this motivation, a decision support method that is able to utilise the available information on previous failures is needed.

A practical maintenance policy selection method should be applicable in the industry. Typically, a production line consists of hundreds of machines with thousands of components. Designing the right maintenance policy is therefore time consuming, and a complex problem. It needs to take into account a large number of criteria, and more importantly, the knowledge of experts, which is often expressed on a qualitative dimension. Assigning a machine to a maintenance strategy is a sorting problem and several multicriteria decision making (MCDM) methods have been proposed. However, classes may be partially ordered and cannot be solved with traditional MCDM methods. A first attempt to solve this problem was proposed using a Decision Making Grid. Unfortunately, this technique considers only two criteria at a time and requires the definition of nine strategies, and as we have shown, may rely on catchall categorisation when class definitions are unsuitable.

In this paper, we presented the development of ElectreSort, a sorting method which is able to analyse several criteria such as downtime, frequency, spare parts costs and bottleneck status. The dual optimistic and pessimistic approach adds richness to the model because it can indicate that an action is in between two classes or is incomparable to certain profiles. In the former case, a mix of strategy derived from both classes is suggested. Different views in the assignment of machines to strategies can be proposed and then discussed interactively with the maintenance manager.

Maintenance is presented as an iterative process, where improvement must be refined towards the achievement of Total Productive Maintenance. The condition of the machines evolves and the maintenance strategies must be adapted. It is important that the decision module is coupled with a data collection system for redefining the reference profiles and the assignments of machines to strategies.

Maintenance usually involves several stakeholders that may have different strategic considerations, e.g. production, finance, quality, human resources, etc. We aim in a future research to extend ELECTRESort for group decisions to take into account different structural and managerial point of views.

Finally, it might be interesting to adapt ElectreSort to solve clustering problem, where classes are not defined a priori.

ACKNOWLEDGMENT

The authors wish to thank Ian Stevens for his proofreading and comments.

7. References

[1] Hong D, Kamaruddin S, Azid I. Maintenance policy selection: a review towards building proper selection model. International Journal of Industrial and Systems Engineering. 2012;10:355-75.

[2] Smadi H, Kamrani A. Maintenance planning and management: a state of the art survey. International Journal of Collaborative Enterprise. 2011;2:227-50.

[3] Garg A, Deshmukh S. Maintenance management: literature review and directions. Journal of Quality in Maintenance Engineering. 2006;12:205-38.

[4] Van Horenbeek A, Pintelon L. Development of a maintenance performance measurement framework—using the analytic network process (ANP) for maintenance performance indicator selection. Omega. 2014;42:33-46.

[5] Gandhare B, Akarte M. Maintenance Strategy Selection. Ninth AIMS International Conference on Management. Pune2012. p. 1330-6.

[6] Bevilacqua M, Braglia M. The analytic hierarchy process applied to maintenance strategy selection. Reliability Engineering and System Safety. 2000;70:71-83.

[7] Aslam-Zainudeen N, Labib A. Practical application of the Decision Making Grid (DMG). Journal of Quality in Maintenance Engineering. 2011;17:138 - 49.

[8] Li C, Xu M, Guo S. ELECTRE III Based on Ranking Fuzzy Numbers for Deterministic and Fuzzy Maintenance Strategy Decision Problems. In: Wang Z, Gu J, editors. Automation and Logistics, 2007 IEEE International Conference on. Jinan2007. p. 309-12.

[9] Zaim S, Turkyilmaz A, Acar M, Al-Turki U, Demirel O. Maintenance strategy selection using AHP and ANP algorithms: a case study. Journal of Quality in Maintenance Engineering. 2012;18:16 - 29.

[10]Wang L, Chu J, Wu J. Selection of optimum maintenance strategies based on a fuzzy analytic hierarchy process. International Journal of Production Economics. 2007;107:151-63.

[11]Peng A, Wang Z. Selecting optimum maintenance approach based on relative membership grade under fuzzy environment. Control and Decision Conference (CCDC), 2011 Chinese. Mianyang: Research Publishing Services; 2011. p. 1748-52.

[12]Zaeri M, Shahrabi J, Pariazar M, Morabbi A. A combined multivariate technique and multi criteria decision making to maintenance strategy selection. In: Herlander M, Xie

M, Jiao R, Tan K, editors. Industrial Engineering and Engineering Management, 2007 IEEE International Conference on. Singapore2007. p. 621-5.

[13]Shyjith K, Ilangkumaran M, Kumanan S. Multi-criteria decision-making approach to evaluate optimum maintenance strategy in textile industry. Journal of Quality in Maintenance Engineering. 2008;14:375 - 86.

[14]Ilangkumaran M, Kumanan S. Selection of maintenance policy for textile industry using hybrid multi-criteria decision making approach. Journal of Manufacturing Technology Management. 2009;20:1009 - 22.

[15]Jafari A, Jafarian M, Zareei A, Zaerpour F. Using Fuzzy Delphi Method in Maintenance Strategy Selection Problem. Journal of Uncertain Systems. 2008;2:289-98.

[16]Mechefske C, Wang Z. USING FUZZY LINGUISTICS TO SELECT OPTIMUM MAINTENANCE AND CONDITION MONITORING STRATEGIES. Mechanical Systems and Signal Processing. 2003;17:305-16.

[17]Mechefske C, Wang Z. Using fuzzy linguistics to select optimum maintenance and condition monitoring strategies. Mechanical Systems and Signal Processing. 2001;15:1129-40.

[18]Vincke P. Multicriteria Decision-Aid. Chichester: John Willey & Sons; 1992.

[19]Labib A. A decision analysis model for maintenance policy selection using a CMMS. Journal of Quality in Maintenance Engineering. 2004;10:191-202.

[20]Labib A. World-class maintenance using a computerised maintenance management system. Journal of Quality in Maintenance Engineering. 1998;4:66 - 75.

[21]Niraj M, Kumar P. Selection of maintenance practice through fuzzy logic based simulation in TPM. International Journal of Scientific & Engineering Research. 2011;2:1-5.

[22]Labib A, Williams G, O'Connor R. An intelligent maintenance model (system): an application of the analytic hierarchy process and a fuzzy logic rule-based controller. Journal of the Operational Research Society. 1998;49:745-57.

[23]Fernandez O, Labib A, Walmsley R, Petty D. A decision support maintenance management system: Development and implementation. International Journal of Quality & Reliability Management. 2003;20:965 - 79.

[24]Ramachandran K, Al Hinai A. Decision mapping and optimal inspection models for plant maintenance: some case studies. The 4th International Multi-Conference on Engineering and Technological Innovation. Orlando2011. p. accessed: 25th May 2012.

[25]Tahir Z, Burhanuddin M, Ahmad A, Halawani S, Arif F. Improvement of decision making grid model for maintenance management in small and medium industries. In: Atputharajah A, editor. Fourth International Conference on Industrial and Information Systems. Sri Lanka: IEEE Conference Publications 2009. p. 598-603.

[26]Burhanuddin M, Ahmad A, Desa M. Maintenance decision making model using multiple criteria analysis for small medium industries. Jurnal Teknologi Maklumat. 2007;19:35-46.

[27]Yuniarto M, Labib A. Fuzzy adaptive preventive maintenance in a manufacturing control system: a step towards self-maintenance. International Journal of Production Research. 2006;44:159-80.

[28]Labib A. A supplier selection model: a comparison of fuzzy logic and the analytic hierarchy process. International Journal of Production Research. 2011;49:6287-99.

[29]Burhanuddin M, Halawani S, Ahmad A. A costing analysis for decision making grid model in failure-based maintenance. Advances in Decision Sciences. 2011;2011:1-15.

[30]Belacel N. Multicriteria assignment method PROAFTN: Methodology and medical application. European Journal of Operational Research. 2000;125:175-83.

[31]Perny P. Multicriteria filtering methods based on concordance and non-discordance principles. Annals of Operations Research. 1998;80:137-65.

[32]Léger J, Martel J-M. A multicriteria assignment procedure for a nominal sorting problematic. European Journal of Operational Research. 2002;138:349-64.

[33]Fernandez E, Navarro J, Duarte A. Multicriteria sorting using a valued preference closeness relation. European Journal of Operational Research. 2008;185:673-86.

[34]Yu W. ELECTRE TRI: Aspects methodologiques et manual d'utilisation. Universite Paris-Daupline; 1992. p. 80p.

[35]Almeida-Dias J, Figueira J, Roy B. Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions. European Journal of Operational Research. 2010;204:565-80.

[36]Nemery P, Lamboray C. FlowSort: a flow-based sorting method with limiting or central profiles. TOP. 2008;16:90-113.

[37]Araz C, Ozkarahan I. Supplier evaluation and management system for strategic sourcing based on a new multicriteria sorting procedure. International Journal of Production Economics. 2007;106:585-606.

[38]Fernandez E, Navarro J. A new approach to multi-criteria sorting based on fuzzy outranking relations: The THESEUS method. European Journal of Operational Research. 2011;213:405-13.

[39]Greco S, Matarazzo B, Slowinski R. Rough sets methodology for sorting problems in presence of multiple attributes and criteria. European Journal of Operational Research. 2002;138:247-59.

[40]Ishizaka A, Nemery P, Pearman C. AHPSort: an AHP based method for sorting problems. International Journal of Production Research. 2012;50:4767-84.

[41]Meyer P, Olteanu A-L. Formalizing and solving the problem of clustering in MCDA. European Journal of Operational Research. 2013;227:494-502.

[42]Rocha C, Dias L. MPOC: an agglomerative algorithm for multicriteria partially ordered clustering. 4OR-Q J Oper Res. 2013;11:253-73.

[43]De Smet Y, Nemery P, Selvaraj R. An exact algorithm for the multicriteria ordered clustering problem. Omega. 2012;40:861-9.

[44]Giannoulis C, Ishizaka A. A Web-based decision support system with ELECTRE III for a personalised ranking of British universities. Decision Support Systems. 2010;48:488-97.

[45]Roy B. Multicriteria methodology for decision aiding. Dordrecht: Kluwer Academic Publisher; 1996.

[46]Roy B. Classement et choix en présence de points de vue multiples (la méthode Electre). Revue Francaise d'Informatique et de Recherche Opérationnelle. 1968;2:57-75.

[47]Roy B, Bertier P. La méthode ELECTRE II: Une application au media-planning. In: Ross M, editor. Operational Research 19721973. p. 291-302.

[48]Roy B, Present M, Silhol D. A programming method for determining which Paris metro stations should be renovated. European Journal of Operational Research. 1986;24:318-34.

[49]Roy B. ELECTRE III: algorithme de classement base sur une présentation floue des préférences en présence de critères multiples. Cahiers du CERO. 1978;20:3-24.

[50]Hugonnard J, Roy B. Ranking of suburban line extension projects for the Paris metro system by a multicriteria method. Transportation Research. 1982;16A:301-12.

[51]Corrente S, Greco S, Słowiński R. Multiple Criteria Hierarchy Process with ELECTRE and PROMETHEE. Omega. 2013;41:820-46.

[52]Yu W. Aide multicritère à la décision dans le cadre de la problématique du tri: concepts, méthodes et applications. Paris: Université Paris-Dauphine; 1992.

[53]Almeida-Dias J, Figueira J, Roy B. A multiple criteria sorting method where each category is characterized by several reference actions: The Electre Tri-nC method. European Journal of Operational Research. 2012;217:567-79.

[54]Nemery P. On the use of multicriteria ranking methods in sorting problems [PhD dissertation]. Bruxelles: Université Libre de Bruxelles; 2008.

[55]Andriosopoulos D, Gaganis C, Pasiouras F, Zopounidis C. An application of multicriteria decision aid models in the prediction of open market share repurchases. Omega. 2012;40:882-90.

[56]Mishra P, Anand G, Kodali R. Strengths, weaknesses, opportunities, and threats analysis for frameworks of world-class maintenance. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture. 2007;221:1193-208.

[57]Waeyenbergh G, Pintelon L. A framework for maintenance concept development. International Journal of Production Economics. 2002;77:299-313.

[58]Mann L, Saxena A, Knapp G. Statistical-based or condition-based preventive maintenance? Journal of Quality in Maintenance Engineering. 1995;1:46-59.

[59]Williams J, Drake P, Davies A. Condition-Based Maintenance and Machine Diagnostics London: Chapman & Hall; 1994.

[60]Carnero Moya MC. The control of the setting up of a predictive maintenance programme using a system of indicators. Omega. 2004;32:57-75.

[61]Markeset T, Kumar U. Design and development of product support and maintenance concepts for industrial systems. Journal of Quality in Maintenance Engineering. 2003;9:376 - 92.

[62]Buchanan J, Vanderpooten D. Ranking projects for an electricity utility using ELECTRE III. International Transactions in Operational Research. 2007;14:309-.

[63]Hipkin I, De Cock C. TQM and BPR: lessons for maintenance management. Omega. 2000;28:277-92.

[64]Crespo Marquez A, Gupta J. Contemporary maintenance management: process, framework and supporting pillars. Omega. 2006;34:313-26.

Appendix: Proof of the properties of ELECTRESORT

a) Independence.

From Pseudo-Code 1, where the assignment rules are given for the descending case, one can notice that an action ***a*** is only compared to the profiles defining different classes. Thus, ***a*** is not compared to any other action, which ensures the property of Independence.

b) Homogeneity

Section 3.2.5 describes the preference relations between an action to be assigned and the central profiles. The assignment rules, given in Pseudo-Code 1, are solely based on the preference relations between the action and the central profiles. Therefore, if two actions ***a*** and ***b*** are compared similarly to the reference profiles (i.e. the preference relations between ***a*** and the central profiles are equivalent to the preference relations between ***b*** and the central profiles), the assignment rules will lead to the same outcome.

c) Monotonicity

We need to prove that if an action ***a*** dominates an action ***b***, action ***a*** is assigned to a class of the same level or a better level as action ***b***.

The dominance relation between ***a and b*** means that for each criterion z_i (with *i*=1,…,*n*), we have that $z_i(b) \leq z_i(a)\ \forall i$, if we suppose that every criterion has to be maximized.
This leads, based on the definition of the concordance index (Section 3.2.2 – equation (5)) and the discordance index (Section 3.2.3 – equation (6))) on each criterion *i* to:

$$\mathrm{c}_i(a, cp_j) \geq \mathrm{c}_i(b, cp_j), \forall cp_j, i \tag{A.1}$$

$$\mathrm{D}_i(a, cp_j) \leq \mathrm{D}_i(b, cp_j), \ \forall cp_j, i \tag{A.2}$$

And therefore, with [A. 1], the global concordance index:

$$\mathrm{C}(a, cp_j) = \frac{1}{\mathrm{W}} \sum_{i=1}^{n} w_i * \mathrm{c}_i(a, cp_j) \geq \frac{1}{\mathrm{W}} \sum_{i=1}^{n} w_i * \mathrm{c}_i(b, cp_j) = \mathrm{C}(b, cp_j) \tag{A.3}$$

Based on [A.3], [A.2] and the definition of the degree of credibility (7), we have that:

$$\mathrm{S}(a, cp_j) \geq \mathrm{S}(b, cp_j), \forall cp_j \qquad [A.4]$$

Similarly, based on the dominance between ***a*** and ***b**,* and the definition of the concordance and the discordance index (5), we have that:

$$\mathrm{c}_i(cp_j, a) \leq \mathrm{c}_i(cp_j, b), \forall cp_j, i \qquad [A.5]$$

$$\mathrm{D}_i(cp_j, a) \geq \mathrm{D}_i(cp_j, b), \forall cp_j, i \qquad [A.6]$$

Based on [A.5] and [A.6] we have finally that:

$$\mathrm{S}(cp_j, a) \leq \mathrm{S}(cp_j, b), \forall cp_j \qquad [A.7]$$

From the credibility indices [A.4], [A.7] and the preference relations given in Section 3.2.5, we can draw the following conclusions regarding the preference relations between the actions ***a*** and ***b*,** and the reference profiles:

- $\boldsymbol{b} \gg cp_j \Longrightarrow \boldsymbol{a} \gg cp_j$
- $\boldsymbol{b} \ll cp_j \not\Leftrightarrow \boldsymbol{a} \ll cp_j$
- $\boldsymbol{b}$ I cp_j $\Longrightarrow$ $\boldsymbol{a}$ I cp_j or $\boldsymbol{a} \gg cp_j$
- $\boldsymbol{b}$ R cp_j $\Longrightarrow$ $\boldsymbol{a}$ I cp_j or $\boldsymbol{a} \gg cp_j$ or $\boldsymbol{a}$ R cp_j

Based on these preference relations and the assignment rules given in Pseudo-Code 1, we can conclude that action ***a*** will be assigned to class of a better level (or at least the same level) as compared to the class to which ***b*** is assigned.

d) Conformity

We need to prove that a central profile cp_j is univocally assigned to its corresponding class with both assignment rules.

Based on the definitions of concordance (4), (5) and discordance (6) indices, we have the degree of credibility of $\mathrm{S}(cp_j, cp_j) = 1$. This reflects the fact that a central profile cp_j is always indifferent to itself: cp_j I cp_j . This is the only case where we have an indifference relation. In fact, as mentioned in Section 3.2, there is a complete ordered structure for the classes. This implies that higher-level classes have central profiles which are *Pareto Superior to* central profiles of lower-level classes: $cp_j \gg cp_i, \forall j < i$ and $cp_j \ll cp_h, \forall j > h$. Moreover, when classes are on the same level, we have the following preference relation between the central profiles:-

$$cp_{j_m} \text{ R } cp_{j_n}, \forall j, m \neq n$$

Based on these preference relations and the descending assignment rules given in Pseudo-Code 1, we conclude that cp_j will be assigned to its corresponding class. This equally holds true when using the ascending rules.

e) Stability

We need to prove that the fusion of two categories of consecutive levels does not alter the assignments of actions, which have been assigned to other categories.

Let us consider the fusion of the categories C_j and C_{j+1} and that an action ***a*** was initially assigned to a better category C_h (where $h < j$). Based on the assignment rules given in Pseudo-Code 1, a has been assigned to C_h due to one of the following situations:

i. $a\ I\ cp_h$: ***a*** is indifferent to the central profile C_h

ii. $a\ R\ cp_{h_p}$ with p = 1,…,*n*: ***a*** is incomparable to at least one central profile of level h

iii. $a \ll cp_h$ and $a \gg cp_{h+1}$: ***a*** is preferred to the central profile of level h and being preferred by the central profile of level *h-1*.

After fusion of the categories C_j and C_{j+1}, we can notice that relations *i* and *ii* remain identical (as $h < j$) as well as the preference relation $a \ll cp_h$.
Let us assume cp^*_{h+1} is the new central profile resulting from the fusion of two categories of adjacent levels. Given the Pareto dominance condition on the reference profiles ($cp_{h+1} \gg cp^*_{h+1}$), the preference relation $a \gg cp^*_{h+1}$ must hold. This guarantees the stability of assignment of the actions assigned to a better category. The preference relations remain unchanged for those actions as well for those assigned in a lower category than the fused one. This proves that the stability condition is fulfilled for the assignment rules in the descending case. The proof is analogous in the ascending case.