



HAL
open science

Multi-criteria improvement of options

Abdelhak Imoussaten, Jacky Montmain, François Troussset, Christophe Labreuche

► **To cite this version:**

Abdelhak Imoussaten, Jacky Montmain, François Troussset, Christophe Labreuche. Multi-criteria improvement of options. European Society for Fuzzy Logic and Technology, 2011, Annecy, France. pp.1. hal-00808605

HAL Id: hal-00808605

<https://hal.science/hal-00808605>

Submitted on 9 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-criteria improvement of options

Abdelhak Imoussaten¹ Jacky Montmain¹ François Troussel¹ Christophe Labreuche²

¹LGI2P, EMA, Parc Scientifique Georges Besse, 30900 Nîmes, France

²Thales Research & Technology, 1 avenue Augustin Fresnel, 91767 Palaiseau Cedex, France

Abstract

Designing the way a complex system should evolve to better match the customers' requirements provides an interesting class of applications for multi-criteria techniques. The required models to support the improvement design of a complex system must include both preference models and system behavioral models. A MAUT model captures the decisions related to customers' preferences whereas a fuzzy representation is proposed to model the relationships between systems parameters and performances to capture operational constraints. This latter part of the improvement design is supported by a branch and bound algorithm to efficiently compute the most relevant actions to be performed.

Keywords: decision support systems, multiple criteria analysis, bi-capacity, multicriteria improvement, CSP

1. Introduction

To satisfy a fluctuating demand and a high level of quality and services, industrials have to develop and integrate new features in their product market leaders [15]. This increases the complexity in the multi-disciplinary design to fulfill functional, technical, environmental, economic and security requirements. In this context, industrials focus more specifically on optimization and evaluation activities of the design process to improve and adapt complex systems. When designers choose architecture and components of the future system they have to check if their solutions do not violate any constraint and if they satisfy the customers' needs and the technical specifications *w.r.t.* the system characteristics. They have also to identify which and how parameters' values must be changed to achieve the expected characteristics of the new system. Furthermore, these issues are of course not free of budgetary constraints. Such activities are repetitive and tedious due to the large number of possible solutions and design parameters that are sometimes continuous. In order to provide designers with a decision support tool able to assist them when optimizing and evaluating architecture choices, we investigate some mathematical models from multi-criteria decision, operational research and artificial intelligence.

In this brief problem statement, two main stakes appear in the design of improvement of a complex

system: 1) which changes of the system's characteristics would warranty the fulfillment of customers' requirements and material constraints specifications; 2) which adjustments of the parameters should provide with these expected characteristics of the improved system. It may spark off two a priori opposite trends to tackle this preference versus constraints problem. The aim of this paper is to give an overview of our approach that tackles these two issues rather than to focus on a technical detail of our model.

The paper is structured as follows. First, section 2 formally enunciates the problem of interest and provides the main notations in the paper. Next, section 3 models the search of the characteristics to be improved first as a multi criteria optimization problem. Afterwards, section 4 is dedicated to parameters' adjustments that would provide the expected characteristics. It is formulated as a constraints solving problem and a branch and bound algorithm is provided in section 5 to solve it efficiently.

2. Problem of interest and notations

In the design of improvement of complex systems, multiple decision criteria need to be considered [1, 15]. As an example, one can think of a military information architecture [16] or the improvement of the performance of an industrial device [2, 3]. Such a complex system is characterized by input parameters z_1, \dots, z_p , e.g., the precise definition of all entities in the military force and their links, or the control parameters of the industrial device. The set of all possible values of the vector of variables (z_1, \dots, z_p) is denoted by Γ . A system is thus defined by an element $\gamma \in \Gamma$. Not all elements of Γ lead to admissible systems for the customer since some requirements of the customer must usually be fulfilled. The set of elements $\Gamma_A \subseteq \Gamma$ for which the associated system satisfies these requirements are the feasible values of the input parameters.

Yet, all elements of Γ_A are not indifferent to the customer. The company needs to construct a model of the preferences of the customer based on his decision criteria. These criteria are often a refinement of the requirements.

The set of attributes is denoted by X_1, \dots, X_n , and the set of alternatives is $X = X_1 \times \dots \times X_n$. For the military architecture, these attributes quantify the fulfillment of the operational mission and are

obtained by large simulations on architecture-labs [16]. For the industrial process, these attributes are the measures that quantify the quality (purity, concentration,...) [3].

Let $T : \Gamma_A \rightarrow X$ be the transformation that provides the values on the attributes of the system obtained from a vector $\gamma \in \Gamma_A$ of the input parameters. On the examples given above, the determination of $T(\gamma)$ for $\gamma \in \Gamma_A$ is not easy. It requires complex simulations or experiments, and are thus costly and time consuming. For these reasons, one often needs to content himself with a qualitative influence model which links the parameters or actions to the expected characteristics [4, 5, 14]. For instance, for the industrial system, improving the training of the operators should have a positive impact on the reject rate of the manufactured items. This is a qualitative influence rule that may be uncertain. It would be very difficult to quantify the quantitative gain on the reject rate that a better trained operator would yield. All these concepts are summarized in (figure 1 – Labels L_1 to L_4).

The preferences of the customer are usually complex and require an elaborate multi-criteria model. Among the criteria of the customer, one usually have operational and monetary ones. For a customer that aims to possess a complex system, the performance of the product is compulsory. A low cost cannot compensate for bad operational performance. As a result, the operational criteria act as veto. Many other interactions such as conditional relative importance of criteria are most often encountered.

Moreover, we denote by $C(\gamma)$ the cost of producing solution γ in practice (respectively, $C(\gamma, \gamma^*)$ is the cost to improve the system γ into γ^*). At the end the company would like to determine the solution that reaches the expected quality e at the lowest cost.

The design of improvement of complex systems is thus a thorny problem as soon as parameters and criteria are numerous. Thus design includes preference models and system behavioral models, quantitative and qualitative knowledge, optimization and combinatory aspects into a multi dimensional assessment context. This paper proposes a formal model of the required knowledge to this problem and computational models to identify efficient improvement actions in practice.

3. Multi-criteria improvement

This section aims to formulate the improvement problem as a multi-criteria optimization problem. Defining an improvement in a multi-criteria context raises some problems.

3.1. The aggregative model

First of all, one must be able to compare any two described situations by means of their elementary

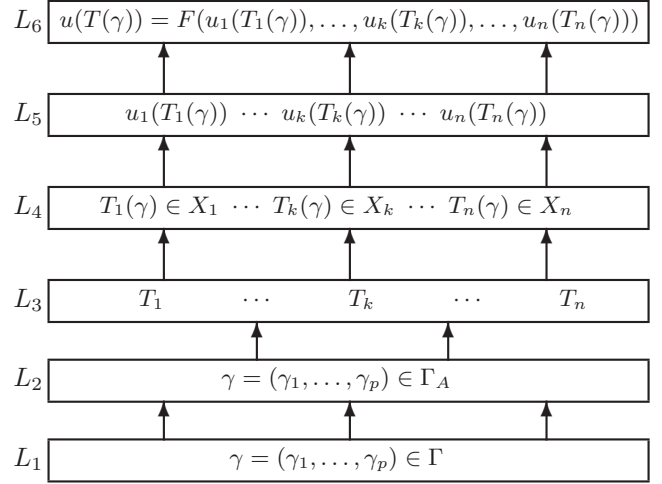


Figure 1: Picture of the concepts used in the evaluation of complex systems. L_1 : space of parameters; L_2 : restrictions with the operational constraints; L_3 : observation of the process; L_4 : space of attributes; L_5 : space of utilities; L_6 : overall satisfaction.

characteristics. The MAUT, i.e., Multi- Attribute Utility Theory [6, 7, 9] provides a possible alternative to tackle this problem in case of quantitative assessments. Usually, it amounts to find a real-valued utility function U such that for any pair of alternatives, x, x' in some set X of alternatives of interest, $x \succ x'$ (x is preferred to x') iff $U(x) \geq U(x')$. When alternatives are n-dimensional, i.e., $X = X_1 \times \dots \times X_n$, a widely studied model is the decomposable model of Krantz *et al.* [10], where U has the form $U(x_1, \dots, x_n) = F(u_1(x_1), \dots, u_n(x_n))$ where the u_i are real-valued utility functions in $[0, 1]$, N the set of n criteria and $F : [0, 1]^n \rightarrow [0, 1]$ is an aggregation operator.

Thus a utility function $u_i : X_i \rightarrow [0, 1]$ is associated to each characteristic X_i of system γ : it quantifies the degree of satisfaction provided by a value x_i regarding the performance of the system relatively to the criterion i . Let us note

$$P_i(\gamma) = u_i(T_i(\gamma)) = u_i(x_i)$$

(figure 1 – labels L_4 and L_5) the performance of system γ with regard to criterion i . $P_i(\gamma)$ will be denoted P_i when there is no ambiguity.

The overall evaluation of a system characterized by $\gamma \in \Gamma_A$ is then defined as:

$$F(u(T(\gamma))) = F(u_1(T_1(\gamma)), \dots, u_n(T_n(\gamma)))$$

where T_1, \dots, T_n are the n components of T (i.e., $\forall i, T_i(\gamma) = x_i$), u_1, \dots, u_n are the utility functions and F is the aggregation function (figure 1 – Labels L_4 to L_6). Then the search of an efficient solution γ^* that reaches the expected quality e at the lowest cost can be stated as the following optimization problem:

$$\underset{\gamma \in \Gamma_A, F(u_1(T_1(\gamma)), \dots, u_n(T_n(\gamma))) = e}{\text{Argmin}} C(\gamma) \quad (1)$$

Let $\gamma \in \Gamma_A$ be the initial system to be improved,

and $a = (u_1(T_1(\gamma)), \dots, u_n(T_n(\gamma))) = (P_1, \dots, P_n)$ the initial elementary performances of γ . Problem (1) can often be refined as the improvement of the initial solution γ . Most of the time the decision-maker wants to know how to improve option a into a new profile b such that the overall evaluation $F(b)$ reaches a given expectation level [12]

The optimization problem 1 may be a very complex operation since we have seen that T is not known explicitly, and it is very complex to perform one computation of T . The qualitative model that is proposed here prevents a more conventional and global approach with Lagrange multipliers because the partial derivatives that would be required with Lagrange multipliers techniques (and thus the short changes it would recommend) would not have any relevancy here. That is the reason why we breakdown the design of the improvement into two steps: defining the most profitable criteria to be improved first and then identifying the parameters to be changed to fulfill this improvement (section 4).

3.2. Criteria improvement indicator

Since solving problem (1) is too complex, an alternative approach is to use algorithms, such as steepest descent to iteratively converge to the optimal solution. In the steepest descent method, one needs to know the direction where it will be more rewarding to change the current vector $\gamma \in \Gamma_A$. Based on the assessment $a := T(\gamma)$ of γ on the criteria, we first need to identify on which criteria the modification of a is the more rewarding.

To solve this problem, an index (called *worth index*) denoted by $\omega_A(F)(a)$ quantifying the worth for the profile a to be improved in criteria among $A \subseteq N$, subject to the evaluation function F , has been proposed in [11]:

$$\omega_A^{\wedge}(F)(a) = \int_0^1 \frac{F((1-\tau)a_A + \tau, a_{N \setminus A}) - F(a)}{c(a, ((1-\tau)a_A + \tau, a_{N \setminus A}))} d\tau \quad (2)$$

where $(d_A, a_{N \setminus A})$ denotes a profile that has component of d on criteria A , and the components of a on the other criteria. This expression gives the mean value of the gain $F(g_A, a_{N \setminus A}) - F(a)$ only for improvement vectors $g_A = (1-\tau)a_A + \tau$ on the diagonal from a_A (for $\tau = 0$) to 1_A (for $\tau = 1$) related to the cost of the improvement $\tau(1 - a_A)$. The reason for considering the diagonal from a_A to 1_A is that it is assumed that the improvements on all criteria are homogeneous. The subset A^* of criteria that maximizes the worth index indicates the performance that are the most profitable to be improved first. After improvement, if γ' is the new system, it is characterized by a performances vector b . We may abusively denote $C(\gamma') = C(b)$.

Note that cost functions in equation 2 may be related to decisive factors that are not necessarily monetary considerations. Indeed, they can be related to risk appraisal, temporal requirement, re-

sources availability, ... In that case, there is no relationship at all between cost functions in equation 2 and $C(\gamma')$.

3.3. Approaching the optimal profile iteratively

As we have seen previously, it may be difficult to reach profile b . We make recommendation to iteratively improve from a to b such that $F(b) \geq e$. We propose the following process

1. Define the expectation level e .
2. Define an initial profile $\gamma \in \Gamma_A$.
3. Compute $a_1 := T_1(\gamma), \dots, a_n := T_n(\gamma)$.
4. Deduce $p = F(u_1(T_1(\gamma)), \dots, u_n(T_n(\gamma))) = F(a)$.
If $p \geq e$ STOP.
5. Compute $A^* = \text{Arg max}_{A \subseteq N} \omega_A^{\wedge}(F)(a)$ as proposed in section 3.2.
6. Based on these figures, the user modifies γ . He tries to act on the criteria in A^* .
7. GOTO Step 3.

The next sections consist in solving step 6.

4. Parameters adjustment

4.1. Objectives and action relationships

The MAUT framework merely captures the preferences of designers without any further considerations regarding the material constraints behind the improvement implementation. Step 5 of the iterative procedure in section 3.3 provides the criteria A^* to be improved first: it provides indications regarding the most efficient improvement to be performed but it says nothing about the way this improvement can be operated (*i.e.*, at this stage no help is provided regarding step 6 of the procedure in section 3.3). These operational constraints however shall not be ignored in designing the implementation component of the improvement project. Sections 4 and 5 provide a decision support for step 6. They aim to search for parameters' values in Γ_A that could improve criteria in A^* .

Two actions are related to a parameter γ_i : its value can be increased or decreased. The two actions “ γ_i increase” and “ γ_i decrease” are mutually exclusive. Thus, there are $2 \times p$ potential actions: a_1, \dots, a_{2p} (p couples of exclusive actions). Adding the potentiality of leaving untouched a parameter, this gives 3^p *global actions* or *action plans* that result from the application of a unique elementary action on each parameter.

Now let us determine for each partial performance $P_i = u_i(T_i(\gamma))$ the set S_i of actions a_j that support an improvement of P_i and the set D_i of actions a_j that distract from P_i . These notations were initially introduced in [4]. We will use the same notations in the algorithm proposed in this section to help clarify the discussion in section 6. For any action $a_j \in S_i$, δ_{ij}^s is the influence degree that an action

a_j may support an improvement with regard to P_i . For any action $a_j \in D_i$, δ_{ij}^d is the influence degree that an action a_j may distract from an improvement with regard to P_i .

To our mind, this qualitative model seems to match the genuine expertise that is generally available with regard to the transformation T in a complex system: the effects of a parameter variation has upon the characteristics X_i of a system can generally be only described in a purely qualitative and non determinist manner. When a characteristic X_i evolves, it then implies a variation of the associated performance P_i . A designer also usually qualitatively knows the way P_i evolves when there is a change in X_i , i.e., the monotony of the function that links P_i and X_i , at least locally.

The set of these fuzzy relationships between actions and system performances provide the necessary representation to support the choice of the actions to be performed to improve all the criteria in A^* with a maximal degree of influence. The way the actions can be selected is proposed in the following.

Let us denote P_i ($i \in N$) the elementary performance indicators, with the $2 \times p$ associated actions a_j ($j \in \{1, \dots, 2p\}$, actions a_{2j-1} and a_{2j} are mutually exclusive). An action a_j may belong to S_i , to D_i or may exert no influence on P_i . Relations involving actions and performance indicators can be represented through a digraph, such that (see example in Fig. 2): for an action a_j and an indicator P_i , the arc arc_{ij} between a_j and P_i is (see figure 4.1)

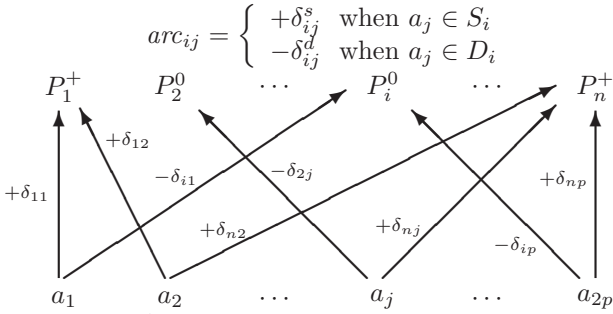


Figure 2: An influence graph

The performances related to criteria in A^* are denoted by P_i^+ otherwise by P_i^0 . For any P_i , we can now consider the sets S_i and D_i to be fuzzy sets, as proposed in [4]. Let us introduce two functions associated with S_i and D_i .

- Support function of performance P_i :
 $s_{P_i}(a_j) = \delta_{ij}^s$ if action a_j supports P_i
with degree δ_{ij}^s , and $s_{P_i}(a_j) = 0$ otherwise.
- Distract function of performance P_i :
 $d_{P_i}(a_j) = \delta_{ij}^d$ if action a_j distracts from P_i
with degree δ_{ij}^d , and $d_{P_i}(a_j) = 0$ otherwise.

For each performance indicator P_i , the two fuzzy sets S_i and D_i are defined by the functions s_{P_i} and d_{P_i} , in their capacity as membership functions. S_i is the fuzzy support set of P_i , whereas D_i is the distraction set of P_i . Note that this model could be

extended: elements of S_i and D_i could be subsets of actions (and not elementary actions as proposed by Felix) that should necessarily be conjointly applied to improve performance indicator P_i (whereas each action of this subset cannot achieve the expected result separately). For example, logical configuration constraints may be added to constraint actions to be conjointly carried out as proposed in [8]. These constraints do not introduce further difficulties in our approach. They might be managed when admissible configurations are filtered and the combinatorial problem is reduced thanks to this additional knowledge.

4.2. Computation of an action plan consistent with the criteria in A^*

Let us suppose that the most suitable criteria A^* to be improved have been determined as proposed in section 3.

Definition 1 An action plan $AP \subseteq \{a_1, \dots, a_{2p}\}$ is a subset of consistent actions, i.e. if the action " γ_i increase" (resp. " γ_i decrease") belongs to AP , then " γ_i decrease" (resp. " γ_i increase") cannot belong to AP (i.e., . actions " γ_i increase" and " γ_i decrease" are mutually exclusive).

There might be additional constraints among the actions. For instance, it may not be possible to perform two actions at the same time. Such constraints exist in the configuration problem [8]. However, we do not consider such constraints explicitly in this paper.

Let us note a_{2q-1} the action "decrease γ " and a_{2q} , the action "increase γ ", $q = 1 \dots p$.

Definition 2 An admissible action plan for A^* is a subset of actions that conjointly improves all the performances indicators in A^* .

Let Ω be the set of the 3^p potential action plans. For any $J \subseteq \{1, \dots, 2p\}$, let $AP_J = \{a_j : j \in J\}$, J_i^+ the subset of indices for actions in $AP_J \cap S_i$, and J_i^- the subset of indices for actions in $AP_J \cap D_i$.

The idea here is to compute the degree of admissibility of action plan AP_J as regards the performances to be improved in A^* .

A possible solution for the degree of admissibility of action plan AP_J with regard to elementary performance P_i may be: $\forall i \in N$

$$s_{P_i}(AP_J) = \begin{cases} \min_{j \in J_i^+} \delta_{ij}^s & \text{if } \min_{j \in J_i^+} \delta_{ij}^s > \max_{j \in J_i^-} \delta_{ij}^d \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The choice of the "min/max" operations in $\min_{j \in J_i^+} \delta_{ij}^s > \max_{j \in J_i^-} \delta_{ij}^d$ is a drastic constraint which leads to a form of veto upon any performance criterion. Thus, it models a cautious viewpoint in the lack of knowledge about the importance of each elementary performance on the overall

one. Other less constaining operators may be envisaged. For example, a more flexible model based on “max/max” operations as stated in the following may be introduced:

$$s_{P_i}(AP_J) = \begin{cases} \max_{j \in J_i^+} \delta_{ij}^s & \text{if } \max_{j \in J_i^+} \delta_{ij}^s > \max_{j \in J_i^-} \delta_{ij}^d \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The choice of the operator semantics is not the subject of this paper and the cautious model (equation 3) has been retained here.

The degree of admissibility of action plan AP_J , relative to the subset of elementary performances P_i in A^* , is then given by:

$$s_{A^*}(AP_J) = \begin{cases} \min_{i \in A^*} s_{P_i}(AP_J) & \text{if } \forall i \in N \setminus A^* \\ & (j \in J \wedge \delta_{ij}^d \text{ exists}) \Rightarrow s_{P_i}(AP_J) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The condition $(j \in J \wedge \delta_{ij}^d \text{ exists}) \Rightarrow s_{P_i}(AP_J) > 0$ for any $i \in N \setminus A^*$ simply means that improving criteria in A^* should not imply a decrease with regard to criteria in $N \setminus A^*$, and when an action a_j damages a performance in $N \setminus A^*$ then there necessarily exists a compensative action a'_j such that $\delta_{ij'}^s > \delta_{ij}^d$. Nevertheless, let us note that when the performances profile is Pareto optimal, improving criteria in A^* will necessary entail a decrease with regard to some other criteria. In this case, the formula in (5) needs to be relaxed.

It can finally be imagined that a subset of criteria B^* included in $N \setminus A^*$ for which a minimal decrease is tolerated is defined. Constraint “ $(j \in J \wedge \delta_{ij}^d \text{ exists}) \Rightarrow s_{P_i}(AP_J) > 0$ ” is maintained in equations (5) only for performance indicators in $N \setminus (A^* \cup B^*)$. For performances in B^* :

$$s_{P_i}(AP_J) = \begin{cases} \min_{j \in J^+} \delta_{ij}^s & \text{if } \min_{j \in J^+} \delta_{ij}^s > \max_{j \in J^-} \delta_{ij}^d \\ -\max_{j \in J^-} \delta_{ij}^d & \text{otherwise} \end{cases} \quad (6)$$

Then, the solution will have to conjointly maximize $s_{A^*}(AP_J)$ and $s_{B^*}(AP_J)$. A threshold can also be introduced to quantify admissible decreases.

The degree of admissibility in (5) can also be “smoothed” if elementary performances are not assigned the same relative importance. Using the “min” operator, a right of veto is conferred upon any criterion. For a more flexible attitude, we could assign a weighting distribution to the performance indicators. Let us denote w_i the weight relative to P_i , which enables computing the following:

$$s_{A^*}(AP_J, w) = \begin{cases} \min_{i \in A^*} (\max(1 - w_i, s_{P_i}(AP_J))) & \text{if } \forall i \in N \setminus A^* s_{P_i}(AP_J) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

In this paper, the weights distribution in (7) is derived from A^* : $w_i = 1$ for $i \in A^*$ and $w_i = 0$ otherwise. Relations (5) and (7) are equivalent in

this simple case. However, we have already proposed further methods to build weights distributions in the framework of multicriteria performances improvement when objectives are more precisely defined [13]. Hence, the interest of (7) is justified.

As a final step, we define $\tilde{\Omega}$ as a fuzzy set of action plans AP_J by the function $s_{A^*}(AP_J, w)$, as a membership function, before considering the α -cut of $\tilde{\Omega}$ (an α -cut of a fuzzy set is the subset of elements whose degree of membership is greater than or equal to α).

Definition 3 *An action plan is said to be α -admissible for A^* when its degree of admissibility is equal to or greater than α .*

Definition 4 *An α -admissible set of action plans is ultimately defined as a subset of action plans API_α , such that:*

$$API_\alpha = \{AP_J : s_{A^*}(AP_J, w) \geq \alpha\}. \quad (8)$$

We can now conclude this section by seeking an efficient α -admissible action plan. A cost c_j is associated with each action a_j : $c_j(a_j)$. The cost of action plan $AP_J \in API_\alpha$ can thus be computed as follows:

$$c(AP_J) = \sum_{j \in J} c_j(a_j). \quad (9)$$

An efficient, α -admissible action plan AP_J is then given by:

$$\text{Arg min}\{c(AP_J) : AP_J \in API_\alpha\}. \quad (10)$$

When the number of actions is high this computation clearly raises a combinatory problem; section 5 presents an efficient resolution approach.

This result concludes the parameters design stage. An efficient α -admissible action plan can thus be planned. It provides a decision support to step 6 in the iterative improvement procedure defined in section 3.3.

This computation only qualitatively defines the actions to be performed: we merely know whether the values of selected parameters are to be increased or decreased. The designer must then proceed by trial and error, from experience when applying the required actions. Applying an efficient α -admissible action plan to initial profile a provides a new performance profile b such that $F(b) \geq F(a)$. If $F(b)$ remains lower than the expected quality e , further improvement iterations are required as indicated in section 3.3. The new improvement always requires the same following steps: first, compute the most profitable criteria to be improved, and then identify the admissible action plan consistent with these performances improvements requirements. Improvement is carried on until objective e is reached.

5. The constraint solving problem for performance improvement

When the number of actions is high the computation of admissible action plans clearly raises a

combinatory problem. Our objective is to determine the set of admissible action plans with minimal cost and maximal admissibility degree. In practice, $\forall \alpha \in]0, 1]$, $\text{Argmin}\{c(AP_J) / AP_J \in \text{APII}_\alpha\}$ is computed. This section presents a Branch and Bound algorithm with appropriate heuristics to solve this problem efficiently.

5.1. Properties and definitions

Since the order the actions are carried out is not considered, identifying an admissible action plan in the set of actions \mathcal{A} , consists in searching a subset $AP \subseteq \mathcal{A}$ which satisfies admissibility and cost properties. Thus the search space is isomorphic to $2^{\mathcal{A}}$. If we take into account mutually exclusive actions (p couples of actions $(\gamma_{increase}, \gamma_{decrease})$) then $|\mathcal{A}| = 2p$ and the search may be reduced to $3^p < 2^{|\mathcal{A}|}$.

Lemma 5 *Let \mathcal{A} , SAP and \mathcal{B} be three sets of actions such that (SAP, \mathcal{B}) is a partition of \mathcal{A} . Let $\mathcal{Z} \subseteq \mathcal{B}$ be such that no admissible action plan of \mathcal{A} containing SAP and any action of \mathcal{Z} can be built. Obviously, the set of admissible action plans of \mathcal{A} containing SAP is identical to the set of admissible action plans of $\mathcal{A}' = \mathcal{A} \setminus \mathcal{Z}$ containing SAP .*

This lemma means that if any action in \mathcal{Z} cannot be added to SAP to build an admissible action plan then, if there exists an admissible action plan AP containing SAP , then the actions in $AP \setminus SAP$ must necessary be searched in $\mathcal{A} \setminus \mathcal{Z}$.

Let now consider the constraints related to α -admissibility of an action plan. They will allow us to derive properties to reduce the search.

Definition 6 Restricting actions (RA):

$$\forall C \subseteq N, \forall \alpha \in [0 \dots 1], \\ \mathbf{RA}(\mathcal{A}, C, \alpha) = \{a_j \in \mathcal{A}, \exists i \in C / \delta_{ij}^s < \alpha\}$$

Definition 7 Locking actions (LA):

$$\forall C \subseteq N, \mathbf{LA}(\mathcal{A}, C) = \{a_j \in \mathcal{A}, \exists i \in C / \delta_{ij}^d \geq \max_{k/a_k \in \mathcal{A}} \delta_{ik}^s\}$$

Definition 8 Incompatible actions (INC): *An action a_j is said to be incompatible with another action $a_{j'}$ if and only if any of the three following conditions is verified $\forall C \subseteq N$:*

- *i* : $\exists i \in C$, a_j supports P_i with degree δ_{ij}^s , $a_{j'}$ distracts from P_i with degree $\delta_{ij'}^d$, and $\delta_{ij'}^d \geq \delta_{ij}^s$.
- *ii* : $\exists i \in C$, a_j distracts from P_i with degree δ_{ij}^d , $a_{j'}$ supports P_i with degree $\delta_{ij'}^s$, and $\delta_{ij}^d \leq \delta_{ij'}^s$.
- *iii* : a_j and $a_{j'}$ are mutually exclusive actions.

$\mathbf{INC}(\mathcal{A}, C, \mathbf{a}) = \{a' \in \mathcal{A}, a' \text{ is incompatible with } a\}$.

Property 9 *An admissible action plan cannot contain two incompatible actions.*

Definition 10 $\forall C \subset N$ we say that a subset of action AP_J is an α -admissible action plan for C if all of the following conditions are fulfilled:

- AP_J improve all criteria in C .
 - $\forall j \in J, \mathbf{INC}(AP_J, N \setminus C, a_j) = \emptyset$
- (11)

Property 11 *Let $a \in \mathbf{RA}(\mathcal{A}, C, \alpha)$ then a cannot belong to any α -admissible action plan for C .*

Remark 12 *Providing C in previous definitions of **LA** and **INC** allows them to be very general concepts. But due to the strong constraint in equation 5, $C = N$ in the following. General definitions (with $C \neq N$) will be useful when equation 5 will be relaxed as mentioned in section 4.2.*

Property 13 *Let SAP and \mathcal{B} be two disjoint sets of actions. let consider $a_k \in \mathbf{LA}(SAP \cup \mathcal{B}, N) \cap \mathcal{B}$ then no action plan containing a_k and SAP can be built.*

Property 14 *Let SAP and \mathcal{B} two disjoint sets of actions and $a \in \mathcal{B}$.*

Let $a_k \in \mathbf{INC}(\mathcal{B} \setminus \{a\}, N, a)$ then no α -admissible action plan for A^ containing a_k and $SAP \cup \{a\}$ can be built.*

5.2. General principle

The search space is represented as a binary tree which is explored in depth first. The basic idea is that each node is associated to an action. From each node start two branches denoted \xrightarrow{a} and $\xrightarrow{-a}$ where a is the action associated to the node. A path originated from the root of the tree is a potential action plan where branches $\xrightarrow{-a}$ mean that actions a do not belong to the action plan and branches \xrightarrow{a} mean that actions a belong to the action plan.

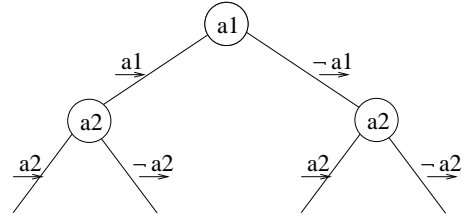


Figure 3: The search space is a binary tree

Remark 15 *In any path originated from the root of the tree there cannot be two nodes associated to the same action.*

The search for the least costly α -admissible action plan requires exploring the binary tree. Heuristics based on the previous properties are then to be introduced to reduce the search combinatory. An appropriate management of constraints enables to relevantly cut branches and significantly reduce the search as much as possible.

At each node the selected action is the one that generates the most drastic new constraints when considering the path from the root of the tree to the current node. This allows eliminating as much actions as possible as soon as possible

Remark 16 *Eliminating an action means to associating this action to the next node and only exploring the branch $\xrightarrow{-a}$ (the branch \xrightarrow{a} is cutted)*

Definition 17 Considering the path P ending at any node, we can define two sets of actions SAP and \mathcal{B} such that:

$$SAP = \{ \text{actions } a / \text{branch } \xrightarrow{\alpha} \text{ belong to } P \}$$

$$\mathcal{B} = \{ \text{actions } a / \text{branch } \xrightarrow{\alpha} \text{ or } \xrightarrow{\neg\alpha} \text{ does not belong to } P \}$$

SAP is an α -admissible sub-action plan improving criteria $C \subset A^* \subseteq N$ and \mathcal{B} is the set of remaining actions potentially able to complete SAP and improve all criteria in A^* .

When exploring a branch $\xrightarrow{\alpha}$ the search is reduced by eliminating some actions in \mathcal{B} as follows:

$$\mathcal{B}' = \mathcal{B} \setminus (LA(\mathcal{B} \setminus INC(\mathcal{B}, N, a), N))$$

and when exploring a branch $\xrightarrow{\neg\alpha}$:

$$\mathcal{B}' = \mathcal{B} \setminus (LA(\mathcal{B} \setminus \{a\}, N))$$

From properties 13 and 14 on one hand, and lemma 5 on the other hand these reductions do not impact the space of solutions.

The actions in $\mathcal{B} \setminus \mathcal{B}'$ will then be eliminated which means (as seen in remark 16) that these actions will be associated to next nodes in the path but only $\xrightarrow{\neg\alpha}$ branches will be explored.

Remark 18 The complexity of computing $LA(\mathcal{B})$, $INC(\mathcal{B}, N, a)$ is linear to $|\mathcal{B}|$, while the dimension of the suppressed branches in the search space is related to $2^{|\mathcal{B}|}$.

5.3. Preferences on the action plans

To be able to identify the most interesting action plans, we need to define an order of preference on them. For that we are going to consider the Pareto non-dominated action plans with respect to the two metrics: degree of admissibility $s_{A^*}(AP_j)$ and cost $c(AP_j)$. Since the preference over these two metrics are opposite, we will use the Pareto order with regard to the couple $(\theta, \phi) = (s_{A^*}(AP_j), -c(AP_j))$.

Definition 19 We define \prec_{Pareto} on \mathbb{R}^2 by

$$(\theta, \phi) \prec_{\text{Pareto}} (\theta', \phi')$$

$$\iff \theta < \theta' \wedge \phi \leq \phi', \quad \theta = \theta' \wedge \phi < \phi'$$

and \triangleleft on $\mathbb{R}^2 \times (\mathbb{R}^2)^p$ (for any p) by

$$(\theta, \phi) \triangleleft \{(\theta_1, \phi_1), \dots, (\theta_p, \phi_p)\}$$

$$\iff \exists j \in \{1, \dots, p\} (\theta, \phi) \prec_{\text{Pareto}} (\theta_j, \phi_j).$$

We denoted \mathcal{D} the set of the Pareto non-dominated admissible action plans.

Remark 20 One may replace the partial order \triangleleft by a complete one (for instance, using only one of the two metrics). In this case, the vector \mathcal{D} will be composed of only one single action plan and the pruning will be more efficient.

5.4. Algorithm

The objective of the algorithm is to determine the set \mathcal{D} from the set of actions \mathcal{A} . First, we eliminate $RA(\mathcal{A}, A^*, \alpha)$. As seen in property 11 and by applying lemma 5 these reductions do not impact the space of solutions.

The selected action at each node is such that:

$$a = \text{Arg max}_{a' \in \mathcal{B}} \{ |INC(\mathcal{B}, N, a')| \}$$

The heuristics behind this decision criterion is: the more restricting the variable, the more decisional in the search process because the current path will be forsaken as soon as possible if necessary.

- The algorithm computes $\mathcal{D} = \text{Algo}(\mathcal{A}, \emptyset, \emptyset, \emptyset)$ which is the set of non-dominated action plans in the sense of \triangleleft .
- **Function** $\text{Algo}(\mathcal{B}, C, SAP, \mathcal{D}')$:


```

If ( $\mathcal{B} \neq \emptyset$ ) then
  - select  $a$  such that :  $|INC(\mathcal{B}, N, a)|$ 
  is maximal;
  -  $SAP' \leftarrow SAP \cup \{a\}$ ;
  -  $\mathcal{B}' \leftarrow \mathcal{B} \setminus INC(\mathcal{B}, N, a)$ ;
  -  $C' \leftarrow C$  and new criteria improved
  by  $a$ ;
  If ( $SAP'$  is an action plan) then
    ① -  $\mathcal{B}' \leftarrow \emptyset$  //cut the branch;
    If  $((s_{A^*}(SAP'), -c(SAP')) \not\triangleleft \mathcal{D}')$ 
    then
      - store  $SAP'$  in  $\mathcal{D}'$ ;
      - remove all dominated elements
      of  $\mathcal{D}'$ ;
    end If
  else
    ② -  $AP_t \leftarrow$  element of
       $\mathcal{D}' / S_{A^*}(AP_t) = S_{C'}(SAP')$ ;
    If  $((AP_t \text{ does not exist } ) \text{ or } (c(SAP') < c(AP_t)))$  then
      -  $\text{Algo}(LA(\mathcal{B}', N), C', SAP', \mathcal{D}')$ ;
    else
      -  $\mathcal{B}' \leftarrow \emptyset$ 
      //cut the branch;
    end If
  end If
  -  $\text{Algo}(LA(\mathcal{B} \setminus \{a\}, N), C, SAP, \mathcal{D}')$ ;
  else
    | ③ //cut the branch;
  end If
return  $\mathcal{D}'$ ;
End

```

Algorithm 1: Algorithm for the determination of \mathcal{D} .

Let $SAP \cup \{a\}$ be the current α -admissible sub-action plan improving the criteria in $C \subset A^*$ and \mathcal{B} the set of remaining actions after eliminating incompatible and locking actions. The branch $\xrightarrow{\alpha}$ is cut in the following cases:

- 1• If $SAP \cup \{a\}$ is an admissible action plan for A^* : any additional action cannot improve the admissibility degree and it would necessarily increase the improvement cost
- 2• If $SAP \cup \{a\}$ is not an action plan for A^* and $\exists AP \in \mathcal{D}$ such that : $c(SAP \cup \{a\}) \geq c(AP)$ and $S_{A^*}(AP) \geq S_C(SAP \cup \{a\})$
- 3• If $SAP \cup \{a\}$ is not an action plan for A^* and $\mathcal{B} = \emptyset$

In condition $\bullet 1\bullet$, if $(s_{A^*}(SAP \cup \{a\}), -c(SAP \cup \{a\})) \not\in \mathcal{D}$ we store $SAP \cup \{a\}$ in \mathcal{D} and we remove all dominated elements from \mathcal{D} .

All the solutions given in \mathcal{D} are α -admissible action plans. We can still decide if we prefer action plans with the highest admissibility degree whatever their costs or the ones with the lowest cost as soon as they are α -admissible.

6. Discussion

This paper is devoted to propose a tradeoff between the managerial and implementation aspects of industrial improvements. The MAUT model discussed in sections 3 enables synthesizing manager preferences with respect to the partial performances to be improved first. Manager preferences are indeed recorded in an analytical form that facilitates the search for strategic improvements in terms of optimization problems, and this therefore provides a powerful artifact for recording overall company performance and deriving a rationale from a managerial perspective. In our opinion, the MAUT model is incompatible with operational constraints, thus requiring it to be complemented with other models that take into account the operational context. More specifically, a model of relationships between the objectives and potential improvement action plans is needed to successfully complete the implementation component of the improvement. It is explained herein how both models must be used in an iterative procedure to design an efficient improvement of a system. Many choices have been made at different stages of the decision making procedure. These local choices clearly impact the computation of the best action plan. The choice of the min operator in equations 3 and 5 is the most influential because the branch & bound heuristics depend on it. All these choices require further semantic analyses to propose a more robust model. This is the concern of our futur works.

In conclusion, the way in which these models are conjointly used within our entire design procedure is intended to prove that both models must be used in tandem in order to address the managerial and implementation issues involved in an improvement project. The challenge consists of developing the consensual transition from motivation to action, between managerial decisions and operational capabilities. This challenge constituted the source of our proposal.

References

[1] A. Baykasoglu, A. Oztas, and E. Ozbay. Prediction and multi-objective optimization of high strength concrete parameters via soft computing approaches. *Expert Systems with Applications*, 36:6145–6155, 2002.

[2] L. Berrah, G. Mauris, A. Haurat, and L. Foulloy. Global vision and performance indicators

for an industrial improvement approach. *Computers in Industry*, 43:211–225, 2000.

[3] L. Berrah, G. Mauris, and J. Montmain. Diagnosis and improvement indexes for a multicriteria industrial performance synthesized by a choquet integral aggregation. *Int. J. of Management Science, OMEGA*, 36:340–351, 2008.

[4] R. Felix. Relationships between goals in multiple attribute decision making. *Fuzzy sets and systems*, 67:47–52, 1994.

[5] R. Felix. Multicriteria decision making (mcdm): Management of aggregation complexity through fuzzy interactions between goals or criteria. In *IPMU*, Malaga, Spain, 2008.

[6] PC. Fishburn. *Utility Theory for Decision-Making*. John Wiley & Sons, New York, 1970.

[7] PC. Fishburn. *The foundations of expected utility*. Dordrecht: Reidel, 1982.

[8] F. Fleurey, V. Delhen, N. Bencomo, B. Morin, and J.M. Jézéquel. Modeling and validating dynamic adaptation. In *3rd Int. workshop on Models @Runtime (MRT'08), at MoDELS'08*, Toulouse, France, 2008.

[9] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives Preferences and Value Tradeoffs*. Cambridge University Press, 1976.

[10] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Additive and Polynomial Representations*, volume 1 of *Foundations of measurement*. Academic Press, 1971.

[11] Ch. Labreuche. Determination of the criteria to be improved first in order to improve as much as possible the overall evaluation. In *IPMU*, pages 609–616, Perugia, Italy, 2004.

[12] J. Montmain. Interprétation de la contribution d'un critère dans le contexte d'une amélioration multicritère basée sur une intégrale de Choquet. In *LFA*, Nîmes, France, 2007.

[13] J. Montmain and S. Sahraoui. How to improve the overall industrial performance in a multicriteria context. In *IPMU*, Malaga, Spain, 2008.

[14] J. Montmain and F. Trouset. The translation of will into act: achieving a consensus between managerial decisions and operational abilities. In *13th symposium on information control in manufacturing*, Moscow, Russia, 2009.

[15] O. Mouelhi, P. Couturier, and T. Redarce. A hybrid search algorithm for multicriteria optimization and evaluation in mechatronic products design. In *Advanced Intelligent Mechatronics*, Singapore, 2009.

[16] J.P. Pignon and Ch. Labreuche. A methodological approach for operational and technical experimentation based evaluation of systems of systems architectures. In *ICSSEA*, Paris, France, 2007.