Platform for prioritized multi-objective optimization by metamodel-assisted Nash games

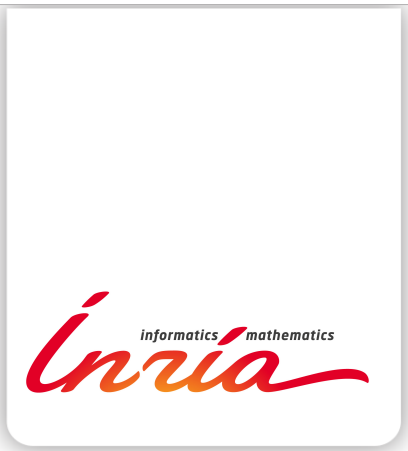
Jean-Antoine Desideri

Platform for prioritized multi-objective optimization by metamodel-assisted Nash games

Jean-Antoine Désidéri

RESEARCH REPORT
N° 9290
September 12, 2019
Project-Team Acumes

ISSN 0249-6399 ISRN INRIA/RR--9290--FR+ENG



Platform for prioritized multi-objective optimization by metamodel-assisted Nash games

Jean-Antoine Désidéri*

Project-Team Acumes

Research Report n° 9290 — September 12, 2019 — 59 pages

* Directeur de Recherche Inria émérite, Équipe Acumes

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Abstract: A multi-objective differentiable optimization algorithm had been proposed to solve problems presenting a hierarchy in the cost functions, $\{f_j(\mathbf{x})\}$ $(j = 1, \ldots, M \geq 2;\ \mathbf{x} \in \Omega_a \subseteq \mathsf{R}^n)$. The first cost functions for which $j \in \{1, \ldots, m\}$ $(1 \leq m < M)$ are considered to be of preponderant importance; they are referred to as the "primary cost functions" and are subject to a "prioritized" treatment, in contrast with the tail ones, for which $j \in \{m+1, \ldots, M\}$, referred to as the "secondary cost functions". The problem is subject to the nonlinear constraints, $c_k(\mathbf{x}) = 0$ $(k = 1, \ldots, K)$. The cost functions $\{f_j(\mathbf{x})\}$ and the constraint functions $\{c_k(\mathbf{x})\}$ are all smooth, say $C^2(\Omega_a)$. The algorithm was first introduced in the case of two disciplines ($m = 1$, $M = 2$), and successfully applied to optimum shape design optimization in compressible aerodynamics concurrently with a secondary discipline [5] [9]. More recently, the theory has been enhanced in both framework and established results, and the new developments will be presented elsewhere in details. In short, an initial admissible point $\mathbf{x}_A^\star$ that is Pareto-optimal with respect to the sole primary cost functions (subject to the constraints) is assumed to be known. Subsequently, a small parameter $\varepsilon \in [0, 1]$ is introduced, and it is established that a continuum of Nash equilibria $\{\bar{\mathbf{x}}_\varepsilon\}$ exists for all small enough ε. The continuum originates from $\mathbf{x}_A^\star$ $(\bar{\mathbf{x}}_0 = \mathbf{x}_A^\star)$. Along the continuum: (i) the Pareto-stationarity condition exactly satisfied by the primary cost functions at $\mathbf{x}_A^\star$ is degraded by a term $O(\varepsilon^2)$ only, whereas (ii) the secondary cost functions initially decrease, at least linearly with ε with a negative derivative provided by the theory. Thus, the secondary cost functions are reduced while the primary cost functions are maintained to quasi Pareto-optimality. In this report, we firstly recall the definition of the different steps in the computational Nash-game algorithm assuming the functions all have known first and second derivatives (here without proofs). Then we show how, in the absence of explicitly known derivatives, the continuum of Nash equilibria can be calculated approximately via the construction of quadratic surrogate functions. Numerical examples are provided and commented.

Key-words: differentiable optimization, multiobjective optimization, Nash game equilibrium

Optimisation multiobjectif priorisée par jeux de Nash assistés de métamodèles

Résumé : Un algorithme d'optimisation différentiable multiobjectif avait été proposé pour résoudre des problèmes présentant une hiérachisation des fonctions coûts, $\{f_j(\mathbf{x})\}$ $(j = 1, \ldots, M \geq 2;\ \mathbf{x} \in \Omega_a \subseteq \mathsf{R}^n)$. Les premières fonctions coûts, pour lesquelles $j \in \{1, \ldots, m\}$ $(1 \leq m < M)$, sont considérées d'importance prépondérante; on y réfère en tant que "fonctions coûts prioritaires" et sont préalablement traitées "en priorité", par opposition aux autres, pour lesquelles $j \in \{m+1, \ldots, M\}$, dites "fonctions coûts secondaires". Le problème est soumis à des contraintes non linéaires $c_k(\mathbf{x}) = 0$ $(k = 1, \ldots, K)$. Les fonctions coûts $\{f_j(\mathbf{x})\}$ et les fonctions de contraintes $\{c_k(\mathbf{x})\}$ sont régulières, disons $C^2(\Omega_a)$. L'algorithme fut d'abord introduit dans le cas de deux disciplines $(m = 1,\ M = 2)$, et appliqué avec succès en conception optimale de forme en aérodynamique compressible en concurrence avec une autre discipline [5] [9]. Plus récemment, la théorie a été élargie en étendant le cadre et les résultats formels, et ces nouveaux développements feront l'objet d'une autre publication détaillée. En bref, on suppose connu un point initial admissible $\mathbf{x}_A^\star$ Pareto-optimal vis-à-vis des seules fonctions coûts prioritaires (sous contraintes). On introduit un petit paramètre $\varepsilon \in [0, 1]$ et on a établi qu'il existe un continuum d'équilbres de Nash $\{\bar{\mathbf{x}}_\varepsilon\}$ pour tout ε suffisamment petit. L'origine du continuum est au point $\mathbf{x}_A^\star$ $(\bar{\mathbf{x}}_0 = \mathbf{x}_A^\star)$. Le long du continuum: (i) la condition de Pareto-stationnarité satisfaite par les fonctions prioritaires en $\mathbf{x}_A^\star$ se dégrade seulement d'un terme en $O(\varepsilon^2)$, alors que (ii) les fonctions secondaires initialement décroissent, au moins linéairement en ε avec une dérivée négative fournie par la théorie, Ainsi, les fonctions coûts secondaires sont réduites alors que les fonctions coûts prioritaires sont quasiment maintenues Pareto-optimales. Dans ce rapport, on rappelle d'abord la définition des différentes étapes de l'algorithme numérique de Nash dans l'hypothèse de fonctions dont on connaît les dérivées premières et secondes (sans les preuves). Puis on montre comment, en l'absence de dérivées connues explicitement, on peut calculer une approximation du continuum des équilibres de Nash par la construction de fonctions quadratiques de substitution. On fournit des exemples numériques commentés.

Mots-clés : optimisation différentiable, optimisation multiobjectif, contraintes nonlinéaires, jeu de Nash, équilibre de Nash

1 Introduction

A multi-objective differentiable optimization algorithm had been proposed to solve problems presenting a hierarchy in the cost functions, $\{f_j(\mathbf{x})\}$ $(j = 1, \ldots, M \geq 2;\ \mathbf{x} \in \Omega_a \subseteq \mathsf{R}^n)$. The first cost functions for which $j \in \{1, \ldots, m\}$ $(1 \leq m < M)$ are considered to be of preponderant importance; they are referred to as the "primary cost functions" and are subject to a "prioritized" treatment, in contrast with the tail ones, for which $j \in \{m+1, \ldots, M\}$, referred to as the "secondary cost functions". The problem is subject to the nonlinear constraints, $c_k(\mathbf{x}) = 0$ $(k = 1, \ldots, K)$.

The cost functions $\{f_j(\mathbf{x})\}$ and the constraint functions $\{c_k(\mathbf{x})\}$ are all smooth, say $C^2(\Omega_a)$. Additionally the cost functions are all assumed to be uniformly strictly positive. This may require a reformulation of the initial setting by the substitution of newly-defined cost functions in place of the former ones, still respecting their sense of variation. This can easily be achieved by application, for example, of an exponential transform with an adequate scaling of the exponent. Note that such a transform modifies the norm of the gradients and Hessians. Hence it can, and should be done in a way that improves the problem numerical conditioning. See for example Section 6.

The theoretical justification for this algorithm will be provided in full details in another publication. The method requires the computation of first and second derivatives which are not always computationally available. For this reason, we are developing here a computational variant in which surrogate approximations of certain functions are constructed, and our Nash-game algorithm is applied to them instead. The corresponding executable file implementing this numerical method is currently being installed in a third chapter of the Inria software platform `http://mgda.inria.fr`. This will soon permit utilizers to operate the code distantly. The purpose of this report is to define precisely the different computational tasks involved in the meta-model-assisted variant of the prioritized approach.

In the next section, we firstly recall the equations defining the standard algorithm when derivatives are computationally available. In the subsequent development, we define one way to proceed by constructing quadratic surrogate functions when only function values are computable. Numerical examples of application of this process are illustrated.

2 Standard algorithm

We assume that a point $\mathbf{x}_A^\star \in \Omega_a$ of Pareto-optimality of the prioritized cost functions alone, $\{f_j(\mathbf{x})\}$ $(1 \leq j \leq m)$, subject to the constraints

$$\mathbf{c} = \mathbf{c}(\mathbf{x}) = \big(c_1(\mathbf{x}), \ldots, c_K(\mathbf{x})\big) = 0 \tag{1}$$

$(K < n)$, has been determined prior to the present development. From the computational viewpoint, such a point $\mathbf{x}_A^\star$ may have been calculated by application of the constrained Multiple-Gradient Descent Algorithm (MGDA) [6] [7] [13] initialized at some appropriate starting point.

Thus, at $\mathbf{x}_A^\star$, the following stationarity condition is satisfied by the projected gradients, here more conveniently expressed as logarithmic gradients:

$$\sum_{j=1}^{m} \alpha_j^\star \mathbf{g}_j^\star = 0 \tag{2}$$

where:

– the superscript * on any symbol indicates an evaluation at $\mathbf{x} = \mathbf{x}_A^\star$;

– the vectors $\{\mathbf{g}_j^\star\}$ $(j = 1, \ldots, m)$ are the projected logarithmic gradients

$$\mathbf{g}_j^\star = \frac{\mathbf{P}\nabla f_j^\star}{f_j^\star} \tag{3}$$

in which ∇ is the symbol for the gradient operator with respect to (w.r.t.) $\mathbf{x}$, and $\mathbf{P}$ is the projection matrix onto the subspace tangent to the constraint manifold, that is, orthogonal to all constraint gradients $\{\nabla c_k^\star\}$;

– the constraint gradients $\{\nabla c_k^\star\}$ $(k = 1, \ldots, K)$ are assumed to be linearly independent, a standard hypothesis for "constraint qualification" [12];

– $\{\alpha_j^\star\}$ $(j = 1, \ldots, m)$ are a set of convex coefficients: $\alpha_j^\star \geq 0$ $(\forall j)$; $\sum_{j=1}^m \alpha_j^\star = 1$.

Generally speaking, the coefficients $\{\alpha_j^\star\}$ associated with the MGDA construction, in case of a point that is Pareto-stationary (critical) or not, can be computed from the following variational definition, an argument extensively made in the theoretical grounds of MGDA: a common descent direction is defined by the vector $\boldsymbol{\omega}^\star$ of minimum Euclidean-norm in the convex hull of the gradients. The choice of the basis to define the metrics is arbitrary, but here, we restrict ourselves to the canonical basis. Hence the vector $\alpha^\star$ solves the following problem:

$$\min_{\alpha} \tfrac{1}{2}\alpha^t \mathbf{G}^t \mathbf{G}\alpha \tag{4}$$

where the superscript ^{t} indicates transposition, and

$$\mathbf{G} = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{g}_1^\star & \cdots & \mathbf{g}_m^\star \\ \vdots & & \vdots \end{pmatrix}, \quad \alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}, \tag{5}$$

and the minimization problem is subject to the following bound and linear constraint:

$$\alpha_j \geq 0 \ (\forall j); \quad \sum_{j=1}^m \alpha_j = 1. \tag{6}$$

Recall that this Quadratic-Programming (QP) problem admits a unique solution for the vector $\omega = \mathbf{G}\alpha$. but not necessarily for the coefficient vector α. The optimality conditions satisfied by the Lagrangian

$$\mathsf{L}(\alpha, \lambda) = \tfrac{1}{2}\alpha^t \mathbf{G}^t \mathbf{G}\alpha + \lambda \left(\sum_{j=1}^m \alpha_j - 1 \right) \tag{7}$$

writes

$$\mathbf{G}^t \mathbf{G}\alpha + \lambda \mathbf{u} = 0 \tag{8}$$

where $[\mathbf{u}]^t = (1, \ldots, 1)$. Hence, the pair $(\alpha^\star, \lambda^\star)$ solves the linear system:

$$\mathbf{\Gamma} \begin{pmatrix} \alpha \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{9}$$

where $\mathbf{\Gamma}$ is the following $(m+1) \times (m+1)$ matrix:

$$\mathbf{\Gamma} = \left(\begin{array}{c|c} [\mathbf{G}^t \mathbf{G})] & [\mathbf{u}] \\ \hline [\mathbf{u}]^t & 0 \end{array} \right). \tag{10}$$

Thus, the question of invertibility of matrix $\mathbf{\Gamma}$ is raised. The following proposition is made:

Proposition 1
If the Jacobian matrix $\mathbf{G}$ *is either of rank* m*, or of rank* $m-1$ *in a situation of Pareto-stationarity, the matrix* $\mathbf{\Gamma}$ *is invertible, and the coefficient vector* $\boldsymbol{\alpha}^\perp$ *of the projected element* $\boldsymbol{\omega}^\perp$ *is unique.*

Proof: Matrix $\mathbf{\Gamma}$ is not invertible if and only if non trivial solutions to the homogeneous system

$$\mathbf{\Gamma} \begin{pmatrix} \boldsymbol{\beta} \\ \mu \end{pmatrix} = 0 \tag{11}$$

($\boldsymbol{\beta} \in \mathsf{R}^m,\ \mu \in \mathsf{R}$) exist, and this is equivalent to the following:

$$(\mathbf{G}^t \mathbf{A}_n \mathbf{G})\boldsymbol{\beta} + \mu \mathbf{e} = 0 \tag{12}$$

$$\mathbf{e}^t \boldsymbol{\beta} = 0. \tag{13}$$

Multiply (12) by $\boldsymbol{\beta}^t$ and use (13) to get $\|\mathbf{G}\boldsymbol{\beta}\|^2 = 0$, that is $\boldsymbol{\beta} \in \ \mathrm{Ker}\,\mathbf{G}$, and two cases are then possible:

- either $\mathbf{G}$ is of rank m and $\mathrm{Ker}\,\mathbf{G} = \{0\}$, and this implies successively: $\boldsymbol{\beta} = 0$ and $\mu = 0$, and only the trivial solution is found;
- or $\mathbf{G}$ is of rank $m-1$ and dim Ker $\mathbf{G} = 1$.

By assumption the latter case occurs in a situation of Pareto stationarity for which there exists a vector $\boldsymbol{\alpha}_0$ such that $\mathbf{G}\boldsymbol{\alpha}_0 = 0$ where the components of α_0 are positive and of sum equal to 1. Then Ker $\mathbf{G} = \{k\boldsymbol{\alpha}_0\}$ ($k \in \mathsf{R}$). Then successively: $\boldsymbol{\beta} = k\boldsymbol{\alpha}_0$, $k = 0$ (by substitution in (13)) (and since $\mathbf{e}^t\boldsymbol{\alpha}_0 = 1$), $\boldsymbol{\beta} = 0,\ \mu = 0$, and again only the trivial solution is found.

In both cases, $\boldsymbol{\Gamma}$ is invertible, and $\boldsymbol{\alpha}^\perp$ is unique. □

We now return to the specific case of the starting point $\mathbf{x}_A^\star$ that is assumed to be Pareto-stationary. At this point, $\boldsymbol{\omega}^\star = \boldsymbol{\omega}_A^\star = 0$, and we make the following

Hypothesis 1
The set of gradients of the primary cost function is of rank $m-1$.

Remark 1
Note that if the rank was inferior to $m-1$, locally, the set of cost functions would be in a sense redundant, not naturally antagonistic, and a new and simpler formulation of the primary problem could be made.

By application of Proposition 1, the coefficient vector $\boldsymbol{\alpha}^\star$ associated with the minimum-norm element $\boldsymbol{\omega}^\star$ is unique and can be calculated by a sequence of projections onto a polytope defined by m vertices in R^n and tests, conducted by a recursive exploration algorithm.

Primary agglomerated cost function. The Gram-Schmidt orthogonalization process is applied to the constraint gradients, technically by the QR factorization. This yields a set of 2 by 2 orthonormal vectors, $\{\mathbf{q}^k\}$ ($k = 1, \ldots, K$), spanning the same subspace. Then

$$\mathbf{P} = \mathbf{I}_n - \sum_{k=1}^{K} \left[\mathbf{q}^k\right] \left[\mathbf{q}^k\right]^t \tag{14}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\left[\mathbf{q}^k\right]$ stands for the column-vector of the components of $\mathbf{q}^k$ in the canonical basis. The Pareto-stationarity condition in (2) can be written equivalently in Lagrangian form as follows:

$$\nabla f_A^\star + \sum_{k=1}^{K} \underline{\lambda}_k \mathbf{q}^k = \nabla f_A^\star + \sum_{k=1}^{K} \lambda_k \nabla c_k^\star = 0 \tag{15}$$

where the $\{\lambda_k\}$ are Lagrange multipliers and the following definition is made

$$f_A(\mathbf{x}) = \sum_{j=1}^{m} \alpha_j^\star \frac{f_j(\mathbf{x})}{f_j^\star}. \tag{16}$$

The function $f_A(\mathbf{x})$ is referred to as the "primary agglomerated cost function".

Convexity fix and primary steering function. The primary agglomerated cost function $f_A(\mathbf{x})$ may not be convex at $\mathbf{x}_A^\star$. The following augmented cost function:

$$f_A^+(\mathbf{x}) = f_A(\mathbf{x}) + \frac{c}{2}\big(\mathbf{x}-\mathbf{x}_A^\star, \mathbf{x}-\mathbf{x}_A^\star\big) \tag{17}$$

where the "convexity-fix constant c" is chosen large enough for the Hessian matrix

$$\mathbf{H}_A^{+,\star} = \nabla^2 f_A^{+,\star} = \nabla^2 f_A^\star + c\mathbf{I}_n \tag{18}$$

to be positive-definite, is referred to as the "primary steering function". We further make the natural hypothesis that the Lagrangian

$$\mathsf{L} = f_A^+(\mathbf{x}) + \sum_{k=1}^{K} \lambda_k c_k(\mathbf{x}) \tag{19}$$

is convex at $\mathbf{x} = \mathbf{x}_A^\star$. This condition is met either because all constraint functions $\{c_k(\mathbf{x})\}$ are locally convex, or by further increase of the convexity-fix constant c.

To accomplish this technically, we propose to proceed as follows.

Let $[h_1, h_n]$ be the range of eigenvalues of matrix $\mathbf{H}_A^\star = \nabla^2 f_A^\star$ and $[\Lambda_1, \Lambda_n]$ the range of eigenvalues of the Hessian of the Lagrangian L. In order to facilitate the iterative convergence of forthcoming procedures, it is preferable to also control the algebraic conditioning. For this, a maximum allowable condition number κ is specified ($\kappa > 1$). Then, for both matrices, we wish to choose c such that the following two conditions are satisfied:

1. the augmented spectrum should be strictly positive;
2. the condition number of either augmented matrix should not exceed κ.

Consider first the case of matrix $\mathbf{H}_A^\star$. The convexity fix shifts the spectrum which becomes $[h_1 + c, h_n + c]$, and the two conditions write:

$$h_1 + c > 0 \tag{20}$$

$$\frac{h_n + c}{h_1 + c} \leq \kappa. \tag{21}$$

If $h_1 > 0$, (20) is satisfied and (21) gives additionally:

$$c \geq c_{11} := \frac{h_n - \kappa h_1}{\kappa - 1} \tag{22}$$

which imposes a limitation only if $h_n - \kappa h_1 > 0$. Inversely, if $h_1 \leq 0$, $c_{11} > 0$ and a strictly-positive c is necessary. Then two sub-cases should be considered. In general $h_n > h_1$, and we propose to replace (20), which does not infer any scale, by the following large inequality

$$h_1 + c \geq \frac{h_n - h_1}{\kappa - 1} \tag{23}$$

which relates the smallest eigenvalue to the eigenvalue range and κ, and requires the additional condition

$$c \geq c_{12} := -h_1 + \frac{h_n - h_1}{\kappa - 1}. \tag{24}$$

But then one observes that with this choice of scale $c_{12} = c_{11}$, so that (22) is suitable regardless the sign of h_1. Now, in the special sub-case of a scalar matrix ($h_n = h_1$), $c_{11} = c_{12} = -h_1$ and if $h_1 < 0$, $c = c_{11}$ is not sufficient. In this special sub-case, we propose to double c_{11}. This precaution being made, in summary, to enforce the positive-definiteness of the matrix $\mathbf{H}_A^\star + c\mathbf{I}_n$, and to limit the condition number to κ, it is sufficient to impose the condition

$$c \geq \max(0, c_{11}) \tag{25}$$

in which c_{11} is doubled in the case of a scalar matrix. Similarly, for the matrix $\nabla^2\mathsf{L}$, one proceeds analogously, and this gives the condition

$$c \geq \max(0, c_{22}) \tag{26}$$

where

$$c_{22} = \frac{\Lambda_n - \kappa\Lambda_1}{\kappa - 1} \tag{27}$$

where again, in the special case of a scalar matrix ($\Lambda_n = \Lambda_1$), the above value of c_{22} is doubled ($c_{22} = -2\Lambda_1$). Ultimately, one sets

$$c = \max(0, c_{11}, c_{22}). \tag{28}$$

The above choice of c contains a certain arbitrariness since the condition number κ is user-specified. In view of (24), if $c \neq 0$, the constant c decreases when κ increases. Hence, a value of κ specified small implies a very good iterative conditioning but at the cost of a large convexity fix. Inversely, a large specified value of κ reduces the convexity fix but may result in iterative difficulties.

As a consequence of this fix, the steering function $f_A^+(\mathbf{x})$ satisfies at $\mathbf{x}_A^\star$ the same necessary conditions for optimality under constraints as does $f_A(\mathbf{x})$. Additionally it is locally convex. Hence, this function is locally minimum at this point.

Territory splitting. After the convexity fix, the reduced Hessian matrix

$$\mathbf{H}'_A = \mathbf{P}\mathbf{H}_A^{+,\star}\mathbf{P} \tag{29}$$

in which

$$\mathbf{H}_A^{+,\star} = \nabla^2 f_A^+(\mathbf{x}_A^\star) \tag{30}$$

is calculated. The matrix $\mathbf{H}'_A$ is symmetric semi-definite positive, since it has exactly K eigenvalues equal to 0 associated with the null space of the projection matrix $\mathbf{P}$. One performs the diagonalization of this matrix

$$\mathbf{H}'_A = \mathbf{\Omega}\,\mathcal{H}\,\mathbf{\Omega}^t \tag{31}$$

where the matrix $\mathbf{\Omega}$ is orthogonal ($\mathbf{\Omega}^t\mathbf{\Omega} = \mathbf{I}_n$), and the matrix

$$\mathcal{H} = \mathbf{Diag}(h'_k) = \begin{pmatrix} \mathcal{H}_\mathbf{u} & 0 \\ 0 & \mathcal{H}_\mathbf{v} \end{pmatrix} \tag{32}$$

is diagonal ($\mathcal{H}_\mathbf{u}$: $(n-p)\times(n-p)$; $\mathcal{H}_\mathbf{v}$: $p\times p$).

The ordering of the eigenvectors, column-vectors of matrix $\mathbf{\Omega}$, is chosen such that the corresponding eigenvalues, $\{h'_k\}$, are such that

$$h'_1 = \cdots = h'_K = 0;\ h'_{K+1} \geq h'_{K+2} \geq \cdots \geq h'_n > 0. \tag{33}$$

Then, the following change of variables (or "territory splitting") is introduced:

$$\mathbf{x} = \mathbf{x}_A^\star + \mathbf{\Omega}\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} := \mathbf{X}(\mathbf{u}, \mathbf{v}) \tag{34}$$

where $\mathbf{u} \in \mathsf{R}^{n-p}$ and $\mathbf{v} \in \mathsf{R}^p$ and the integer p is chosen such that $1 \leq p < n-K$, so that $n-p > K$. One also defines the following block matrix structures:

$$\mathbf{\Omega} = (\mathbf{\Omega}_\mathbf{u}\ \mathbf{\Omega}_\mathbf{v}), \quad \mathbf{\Omega}_\mathbf{u} = \begin{pmatrix} \mathbf{\Omega}_{\mathbf{uu}} \\ \mathbf{\Omega}_{\mathbf{vu}} \end{pmatrix}, \quad \mathbf{\Omega}_\mathbf{v} = \begin{pmatrix} \mathbf{\Omega}_{\mathbf{uv}} \\ \mathbf{\Omega}_{\mathbf{vv}} \end{pmatrix}, \tag{35}$$

where the diagonal blocks $\mathbf{\Omega}_{\mathbf{uu}}$ and $\mathbf{\Omega}_{\mathbf{vv}}$ are $(n-p)\times(n-p)$ and $p\times p$ respectively, and the following partial gradients are defined

$$\nabla_\mathbf{u}(.) = \mathbf{\Omega}_\mathbf{u}^t\nabla(.), \quad \nabla_\mathbf{v}(.) = \mathbf{\Omega}_\mathbf{v}^t\nabla(.). \tag{36}$$

In particular, the following $p \times p$ matrix

$$\mathbf{S} = \nabla^2_{\mathbf{vv}} f_A^{+,\star} \tag{37}$$

is positive-definite and turns out to be the tail $p \times p$ diagonal block of $\mathcal{H}$:

$$\mathbf{S} = \mathcal{H}_{\mathbf{v}}. \tag{38}$$

Secondary steering function. One defines the new variable

$$\mathbf{w} = \mathbf{S}^{\frac{1}{2}} \mathbf{v}. \tag{39}$$

and the corresponding partial gradient

$$\nabla_{\mathbf{w}}(.) = \mathbf{S}^{-\frac{1}{2}} \nabla_{\mathbf{v}}(.) = \mathbf{S}^{-\frac{1}{2}} \mathbf{\Omega}^t_{\mathbf{v}} \nabla(.). \tag{40}$$

Then one computes the following logarithmic gradients of the secondary cost functions

$$\mathbf{g}^\star_j = \frac{\nabla_{\mathbf{w}} f^\star_j}{f^\star_j} = \frac{\mathbf{S}^{-\frac{1}{2}} \mathbf{\Omega}^t_{\mathbf{v}} \nabla f^\star_j}{f^\star_j} \quad (j = m+1, \ldots, M). \tag{41}$$

By applying the MGDA construction, one identifies coefficients $\{\alpha^\star_j\}$ for which the convex combination

$$\boldsymbol{\omega}^\star_B = \sum_{j=m+1}^{M} \alpha^\star_j \mathbf{g}^\star_j \tag{42}$$

has minimal Euclidean norm. Then, the "secondary steering function" is defined as follows

$$f_B = \sum_{j=m+1}^{M} \alpha^\star_j \frac{f_j}{f^\star_j}. \tag{43}$$

Consequently:

$$\nabla f^\star_B = \boldsymbol{\omega}^\star_B, \quad \left(\nabla f^\star_B, \boldsymbol{\omega}^\star_B\right) = \sigma_B = \left\|\boldsymbol{\omega}^\star_B\right\|^2 \geq 0. \tag{44}$$

If $\sigma_B = 0$, or equivalently $\boldsymbol{\omega}^\star_B = 0$, the secondary gradient vectors $\{\mathbf{g}^\star_j\}$ $(j = m+1, \ldots, M)$ are in a configuration of Pareto-stationarity and no improvement of the secondary cost functions can be achieved in the subspace spanned by the subvector $\mathbf{v}$. Thus, if σ_B is too small, the entire process is abandoned. Otherwise, one proceeds with the Nash games.

Nash games. A continuation parameter ε $(0 \leq \varepsilon \leq 1)$ is introduced and the following auxiliary function, convex combination of the two steering functions, is defined:

$$f_{AB} = f_A + \varepsilon \left(f_B - f_A\right). \tag{45}$$

For fixed ε, a Nash equilibrium point

$$\overline{\mathbf{x}}_\varepsilon = \mathbf{X}(\overline{\mathbf{u}}_\varepsilon, \overline{\mathbf{v}}_\varepsilon) \tag{46}$$

is sought as the solution of the following coupled system:

- the sub-vector $\overline{\mathbf{u}}_\varepsilon$ minimizes $f_A^+\big(\mathbf{X}(\mathbf{u}, \overline{\mathbf{v}}_\varepsilon)\big)$ w.r.t. $\mathbf{u}$ subject to the constraints $\mathbf{c}\big(\mathbf{X}(\mathbf{u}, \overline{\mathbf{v}}_\varepsilon)\big) = 0$;
- the sub-vector $\overline{\mathbf{v}}_\varepsilon$ minimizes $f_{AB}\big(\mathbf{X}(\overline{\mathbf{u}}_\varepsilon, \mathbf{v})\big)$ w.r.t. $\mathbf{v}$ subject to no constraints.

Conclusions. It has been established that the above Nash equilibrium point exists for all ε sufficiently small. In particular, for $\varepsilon = 0$:

$$\overline{\mathbf{x}}_0 = \mathbf{x}_A^\star \tag{47}$$

(consistency). If $\sigma_B > 0$, as ε varies:

- The primary steering function $f_A^+\big(\mathbf{X}(\overline{\mathbf{u}}_\varepsilon, \overline{\mathbf{v}}_\varepsilon)\big)$ increases from the value 1, with an initial derivative equal to 0. Hence the equilibrium point departs from the Pareto-optimality of the primary cost functions $\{f_j\}$ $(j = 1, \ldots, m)$ by a term $O(\varepsilon^2)$ only.
- The secondary steering function $f_B\big(\mathbf{X}(\overline{\mathbf{u}}_\varepsilon, \overline{\mathbf{v}}_\varepsilon)\big)$ decreases linearly with ε; its initial derivative is equal to $(-\sigma_B)$. The secondary cost functions $\{f_j\}$ $(j = m+1, \ldots, M)$ decrease at this rate or faster.

3 Metamodel-assisted variant

When first or second derivatives are not available, the algorithm of Section 2 is not directly applicable. In this case, we propose to construct quadratic metamodels, as surrogates to the constraints and the two steering functions, and to apply the standard algorithm based on these functions instead. In this section, the numerical options adopted to implement the successive steps in this construction are described.

3.1 Game preparation

1. **Local evaluation of function values and differentials**
   The following elements are calculated at $\mathbf{x} = \mathbf{x}_A^\star$:

   - cost functions and gradients: $\{f_j^\star\}$, $\{\nabla f_j^\star\}$, $(j = 1, \ldots, M)$;
   - constraint functions and gradients: $\{c_k^\star\}$, $\{(\partial c_k/\partial x_i)^\star\}$ $(k = 1, \ldots, K,\ i = 1, \ldots, n)$;
   - diagonals of Hessians: $\{(\partial^2 f_j/\partial x_i^2)^\star\}$, $\{(\partial^2 c_k/\partial x_i^2)^\star\}$ $(\forall i, \forall j, \forall k)$.

   All differential elements are approximated by central differencing, using the specified step-size 'hfdiff'. Note that the calculation of diagonals of Hessians necessitates no more information than required by the other calculations. If $\mathbf{x}_A^\star$ has been properly specified, the constraint values are equal to 0. Hence, their evaluation provides an accuracy verification.

2. **Geometrical database about $\mathbf{x}_A^\star$ for global information**
   For every hyperplane (x_i, x_j) $(i = 1, \ldots, n-1,\ j = i+1, \ldots, n)$, one considers 8 points for which $x_i = \pm h/\sqrt{2}$ and $x_j = \pm h/\sqrt{2}$, and 8 points for which $x_i = \pm h$ and $x_j = \pm h$. The stepsize h is given the value of the specified parameter 'hbox'. In this way, in each coordinate direction, as well as in each coordinate bisector, 5 symmetrical points centered at $\mathbf{x}_A^\star$ are avaialble, permitting a satisfactory global approximation of a second derivative. In total, the database is made of $8n(n-1)$ points. For example with 6 variables, this gives 240 points. (See Appendix A for details.)

3. **Initialization of constraint metamodels**
   Every constraint function $c_k(\mathbf{x})$ $(k = 1, \ldots, K)$ is evaluated over the geometrical database, and is meta-modeled by the following quadratic surrogate function:

$$\tilde{c}_k(\mathbf{x}) = \big(\nabla c_k^\star, \mathbf{x} - \mathbf{x}_A^\star\big) + \tfrac{1}{2}\big(\mathbf{x} - \mathbf{x}_A^\star, \mathbf{H}_{c_k}^\star\,(\mathbf{x} - \mathbf{x}_A^\star)\big) \tag{48}$$

   where the gradient and the diagonal of the Hessian have been calculated at Step 1. The remaining elements, that is, the off-diagonal elements of the Hessian are adjusted to best

approximate the constraint values over the database in the sense of least squares. (See Appendix B for details.) [1, 2]

In this way:

$$\forall k: \quad \tilde{c}_k^\star = 0, \quad \nabla\tilde{c}_k^\star = \nabla c_k^\star, \quad \text{and} \quad \forall \mathbf{x}, \quad \tilde{c}_k(\mathbf{x}) \doteq c_k(\mathbf{x}). \tag{49}$$

4. **Orthogonalization of constraint Jacobian $\{\nabla\mathbf{c}^\star\}$**
Note that $\nabla\mathbf{c}^\star = \nabla\tilde{\mathbf{c}}^\star$. The QR factorization of the constraint Jacobian $\nabla\mathbf{c}^\star$ is accomplished by a call to the Lapack procedure DGEQRF

$$\nabla\mathbf{c}^\star = \begin{pmatrix} \vdots & & \vdots \\ \nabla c_1^\star & \dots & \nabla c_K^\star \\ \vdots & & \vdots \end{pmatrix} = \mathbf{QR}. \tag{50}$$

The column vectors $\{\mathbf{q}^k\}$ of matrix $\mathbf{Q}$ constitute a set of orthogonal vectors spanning the same subspace as the constraint gradients $\{\nabla c_k^\star\}$ $(k = 1, \dots, K)$.

The projection matrix $\mathbf{P}$ is formally expressed as follows:

$$\mathbf{P} = \mathbf{I}_n - \sum_{k=1}^{K} \left[\mathbf{q}^k\right] \left[\mathbf{q}^k\right]^t. \tag{51}$$

However, its application to a vector $\mathbf{x} \in \mathsf{R}^n$ is more directly performed by the computation of scalar products:

$$\mathbf{P}\mathbf{x} = \mathbf{x} - \sum_{k=1}^{K} (\mathbf{x}, \mathbf{q}^k)\mathbf{q}^k. \tag{52}$$

5. **Projection of logarithmic gradients of prioritized cost functions**
(This step is omitted if $m = 1$.)
Compute the projected vectors

$$\mathbf{g}_j^\star = \mathbf{P}\frac{\nabla f_j^\star}{f_j^\star} = \frac{\nabla f_j^\star}{f_j^\star} - \sum_{k=1}^{K} (\frac{\nabla f_j^\star}{f_j^\star}, \mathbf{q}^k)\mathbf{q}^k \qquad (j = 1, \dots, m) \tag{53}$$

and the norm

$$G = \max_{j=1}^{m} \left\|\mathbf{g}_j^\star\right\|. \tag{54}$$

6. **Calculation of the convex coefficients $\{\alpha_j^\star\}$ $(j = 1, \dots, m)$**
If $m = 1$, one has trivially $\alpha_m^\star = 1$. Otherwise $(m > 1)$, the calculation is carried out by the exploration of a polytope of R^n with m vertices by a recursive algorithm alternating projections and tests.

7. **Definition of the agglomerated primary cost function**

$$f_A(\mathbf{x}) = \sum_{j=1}^{m} \alpha_j^\star \frac{f_j(\mathbf{x})}{f_j^\star} \tag{55}$$

and development of the associated computer procedure.

Computation of the gradient ∇f_A^* and the second derivatives $\{(\partial^2 f_A/\partial x_i^2)^*\}$ $(i = 1, \dots, n)$ as convex combinations of analogous differential elements calculated at Step 1 by central differencing.

[1] In all what follows, the "tilde" over a symbol refers to the associated metamodel.

[2] Note that the construction combines information on local differentials with global information on cross second derivatives.

8. **Metamodel $\tilde{f}_A$ surrogate for the agglomerated primary cost function**

   - Computation of approximate Lagrange multipliers $\{\lambda_k\}$ $(k = 1, \ldots, K)$ to best fit the gradient ∇f_A^* by the vector $\left(-\sum_k^K \lambda_k \nabla c_k^\star\right)$ in the sense of least-squares over the geometrical database.
   - Approximation of the diagonal elements of the Hessian matrix $\mathbf{H}_A^\star$ by convex combinations of analogous differential elements calculated at Step 1.
   - Approximation of the off-diagonal elements of the Hessian matrix $\mathbf{H}_A^\star$ by the same procedure as for constraints at Step 3.

   As a result, the following metamodel is defined:

$$\tilde{f}_A(\mathbf{x}) = 1 - \big(\sum_{k=1}^{K} \lambda_k \nabla c_k^\star, \mathbf{x} - \mathbf{x}_A^\star\big) + \tfrac{1}{2}\big(\mathbf{x} - \mathbf{x}_A^\star, \mathbf{H}_A^\star\,(\mathbf{x} - \mathbf{x}_A^\star)\big). \tag{56}$$

   (See Appendix B for details.)

   Consequently to this construction, the Pareto-stationarity condition (under constraints) is exactly satisfied by the metamodel:

$$\nabla \tilde{f}_A^\star + \sum_{k=1}^{K} \lambda_k \nabla c_k^\star = 0. \tag{57}$$

9. **Convexity fix and primary steering function**
   The eigenvalues range $[h_1, h_n]$ of the Hessian matrix $\mathbf{H}_A^\star$, and the eigenvalue range $[\Lambda_1, \Lambda_n]$ of the Hessian of the Lagrangian L are calculated by calls to the Lapack procedure DSYEV. Then, the convexity-fix constant c is calculated according to (28). This permits to complete the definition of the "primary steering function $f_A^+(\mathbf{x})$" from the metamodel $\tilde{f}_A(\mathbf{x})$:

$$f_A^+(\mathbf{x}) = \tilde{f}_A(\mathbf{x}) + \frac{c}{2}\left\|\mathbf{x} - \mathbf{x}_A^\star\right\|^2. \tag{58}$$

   This function is convex at $\mathbf{x}_A^\star$, as well as the associated Lagrangian and satisfies the necessary optimality conditions under (metamodeled) constraints. Hence the user-specified point $\mathbf{x}_A^\star$ is an exact solution to the problem:

$$\min_{\mathbf{x}} f_A^+(\mathbf{x}) \text{ subject to the constraints: } \tilde{c}_k(\mathbf{x}) = 0 \ (k = 1, \ldots, K). \tag{59}$$

10. **Territory splitting**

    - Computation of the matrix

$$\mathbf{H}'_A = \mathbf{P}\,(\mathbf{H}_A^\star + c\mathbf{I}_n)\,\mathbf{P} \tag{60}$$

    - Diagonalization

$$\mathbf{H}'_A = \mathbf{\Omega}\mathcal{H}\mathbf{\Omega}^t, \quad \mathcal{H} = \begin{pmatrix} \mathcal{H}_\mathbf{u} & 0 \\ 0 & \mathcal{H}_\mathbf{v} \end{pmatrix} \tag{61}$$

      in compliance with the eigen-mode ordering convention.

    - Identification of matrix blocks $\mathbf{\Omega_u}$ et $\mathbf{\Omega_v}$:

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Omega_{uu}} & \mathbf{\Omega_{uv}} \\ \mathbf{\Omega_{vu}} & \mathbf{\Omega_{vv}} \end{pmatrix}, \quad \mathbf{\Omega_u} = \begin{pmatrix} \mathbf{\Omega_{uu}} \\ \mathbf{\Omega_{vu}} \end{pmatrix}, \quad \mathbf{\Omega_v} = \begin{pmatrix} \mathbf{\Omega_{uv}} \\ \mathbf{\Omega_{vv}} \end{pmatrix}. \tag{62}$$

11. **Identification of scaling matrix S, new variable w and associated partial gradient**

$$\mathbf{S} = \mathcal{H}_{\mathbf{v}}, \quad \mathbf{w} = \mathbf{S}^{\frac{1}{2}}\mathbf{v}, \quad \nabla_{\mathbf{w}}(.) = \mathbf{S}^{-\frac{1}{2}}\mathbf{\Omega}_{\mathbf{v}}^{t}\nabla(.), \tag{63}$$

calculation of logarithmic partial gradients of secondary cost functions:

$$\mathbf{g}_j^\star = \frac{\nabla_{\mathbf{w}} f_j^\star}{f_j^\star} = \mathbf{S}^{-\frac{1}{2}}\mathbf{\Omega}_{\mathbf{v}}^{t}\frac{\nabla f_j^\star}{f_j^\star} \quad (j = m+1, \ldots, M). \tag{64}$$

These gradients are assumed to be linearly independent.

(Possibility of a redefinition of the secondary cost functions for a better balance of these gradients.)

12. **Secondary steering function** $f_B(\mathbf{x})$
The minimum Euclidean-norm element $\boldsymbol{\omega}_B^\star$ in the convex hull of above gradients

$$\boldsymbol{\omega}_B^\star = \sum_{j=m+1}^{M} \alpha_j^\star \mathbf{g}_j^\star \tag{65}$$

is identified by calculating the convex coefficients $\{\alpha_j^\star\}$ $(m+1 \leq j \leq M)$ by a procedure similar to that of Step 6: if $M - m > 1$, the calculation is carried out by the exploration of a polytope of R^n with $M - m$ vertices by a recursive algorithm alternating projections and tests.

The directional derivative

$$\sigma_B = \left(\mathbf{g}_j^\star, \boldsymbol{\omega}_B^\star\right) \tag{66}$$

which is positive and independent of j for all j such that $\alpha_j \neq 0$, is also calculated.

Possibility of abandon of the process if σ_B is too small. Otherwise, proceed with the next step.

13. **Agglomerated secondary cost function**

$$f_B(\mathbf{x}) = \sum_{j=m+1}^{M} \alpha_j^\star \frac{f_j(\mathbf{x})}{f_j^\star} \tag{67}$$

and construction of the associated quadratic metamodel $\tilde{f}_B$

$$\tilde{f}_B(\mathbf{x}) = 1 + \Big(\sum_{j=m+1}^{M} \alpha_j \frac{\nabla f_j^\star}{f_j^\star}, \mathbf{x} - \mathbf{x}_A^\star \Big) + \tfrac{1}{2}\big(\mathbf{x} - \mathbf{x}_A^\star, \mathbf{H}_B(\mathbf{x} - \mathbf{x}_A^\star)\big) \tag{68}$$

by adjustment of the elements of the matrix $\mathbf{H}_B$ by least-squares for best curve-fit of function values. Subsequently, the function $\tilde{f}_B(\mathbf{x})$ is referred to as the **"secondary steering function"**.

Lastly, one defines the auxiliary cost function

$$f_{AB} = (1 - \varepsilon) f_A^+ + \varepsilon \tilde{f}_B \tag{69}$$

where $\varepsilon \in [0, 1]$ is the continuation parameter.

Program realization of these functions.

14. **Computations preliminary to Nash games**
Recall that the determination of a Nash game, for a given ε, is the result of the coordination of a constrained optimization of the subvector $\mathbf{u}$ at fixed subvector $\mathbf{v}$ with the unconstrained optimization of subvector $\mathbf{v}$ for fixed subvector $\mathbf{u}$. These two types of optimization are described in Appendix C and the global iterative algorithm is sketched in the flowchart of Figure 24. To facilitate these two processes a number of elements are computed once for all:

- **Q**-matrices: $\mathbf{Q}_{\mathbf{u}}$, $\mathbf{Q}_{\mathbf{v}}$, $\mathbf{Q}_{c_k}$ $(k = 1, \ldots, K)$;
- **H**-matrices: $\mathbf{H}_{A\mathbf{u}}$, $\mathbf{H}_{B\mathbf{u}}$, $\mathbf{H}_{A\mathbf{v}}$, $\mathbf{H}_{B\mathbf{v}}$;
- vectors ϕ_A and ϕ_B.

Additionally, from these elements is deduced, by the solution of a generalized eigenvalue problem, the maximum value $\varepsilon_{\max}$ below which the the optimization of subvector $\mathbf{v}$ is convex. Subsequently, the parameter ε is incremented in the interval $[0, \varepsilon_{\max}]$.

15. **Realization of procedures defining functions in terms of $(\mathbf{u}, \mathbf{v})$ instead of $\mathbf{x}$**
In the most recent version of the code, only the following procedure is defined and used:
    - Procedure for the mapping $\mathbf{x} = \mathbf{X}(\mathbf{u}, \mathbf{v})$

3.2 Step-by-step determination of the continuum of Nash equilibria

Before defining inner-loop iterations, the following outer-loop algorithm is proposed to determine step by step the continuum of Nash equilibria by incrementation of the continuation parameter ε.

For $\ell = 1, 2, \cdots \leq \ell_{\max}$ (at most) set $\varepsilon_\ell = \ell\Delta\varepsilon \leq 1$ (at most), and proceed as follows:

- **Initialization**
  - the very first time ($\ell = 1$, $\varepsilon_1 = \Delta\varepsilon$) the initialization is made in compliance with the known asymptotics:
$$\overline{\mathbf{u}}^0_{\varepsilon_1} := 0, \quad \overline{\mathbf{v}}^0_{\varepsilon_1} := -\varepsilon_1 \mathbf{S}^{-1} \nabla_{\mathbf{v}} f_B^\star; \tag{70}$$
  - subsequently ($\ell > 1$), the previously found equilibrium point is used:
$$\overline{\mathbf{u}}^0_{\varepsilon_\ell} := \overline{\mathbf{u}}_{\varepsilon_{\ell-1}}, \quad \overline{\mathbf{v}}^0_{\varepsilon_\ell} := \overline{\mathbf{v}}_{\varepsilon_{\ell-1}}. \tag{71}$$
- **Coordination loop**
For $\lambda = 1, 2, \ldots$ do:
  1. Minimize $f_A^+\big(\mathbf{X}(\mathbf{u}, \mathbf{v})\big)$ w.r.t. $\mathbf{u}$ under the constraints $\tilde{\mathbf{c}}\big(\mathbf{X}(\mathbf{u}, \mathbf{v})\big) = 0$ for fixed $\mathbf{v} = \overline{\mathbf{v}}_\ell^{\lambda-1}$ by a few iterations of Newton's method. Denote $\overline{\mathbf{u}}_\ell^\lambda$ the converged vector $\mathbf{u}$. (See Appendix C.)
  2. Minimize $\tilde{f}_B\big(\mathbf{X}(\mathbf{u}, \mathbf{v})\big)$ w.r.t. $\mathbf{v}$ under no constraints for fixed $\mathbf{u} = \overline{\mathbf{u}}_\ell^\lambda$, by solving the linear system of optimality conditions. Denote $\overline{\mathbf{v}}_\ell^\lambda$ the solution $\mathbf{v}$, and compute $\Delta\overline{\mathbf{v}}^\lambda = \big\|\overline{\mathbf{v}}_\ell^\lambda - \overline{\mathbf{v}}_\ell^{\lambda-1}\big\|$. (See Appendix C.)
  3. Convergence test: if $\Delta\overline{\mathbf{v}}^\lambda > TOL$, return to 1 (incrementation of λ); otherwise: end of coordination loop.
- **Nash equilibrium point**
Compute
$$\mathbf{x}_\ell = \mathbf{x}_A^\star + \mathbf{\Omega}_{\mathbf{u}} \overline{\mathbf{u}}_\ell + \mathbf{\Omega}_{\mathbf{v}} \overline{\mathbf{v}}_\ell \tag{72}$$
(in which the superscript λ has now been omitted) and the corresponding values of the functions of interest, store the results, and if if $\ell < \ell_{\max}$, increment ℓ and ε and proceed.

In the coordination loop, the solution for $\overline{\mathbf{u}}_\ell^\lambda$, in item 1 above, is conducted by application of Newton's method. The accuracy tolerance is taken to be one hundredth of the coordination accuracy tolerance TOL. Usually, only very few iterations (<5) are sufficient. In item 2, the solution for $\overline{\mathbf{v}}_\ell^\lambda$ is conducted by solving the linear system of optimality. Since $\varepsilon < \varepsilon_{\max}$, the inversion is possible by convexity. (See Appendix C for details.)

Remark 2
At completion of the very first computation of the Nash equilibrium point ($\ell = 1$), the converged subvector $\overline{\mathbf{v}}_{\varepsilon_1}$ is printed for comparison with the asymptotics $\overline{\mathbf{v}}^0_{\varepsilon_1}$ used to initialize. Both vectors are expected to be very close, and this should be verified in practice.

4 Analytical testcases solved using the Nash-MGDA Platform

4.1 Test-case TC4

This test-case has been introduced to illustrate the prioritized multi-objective method in the simplest situation in which all calculations can be carried out analytically, and this is done first here.

Test-case TC4: analytical developments. Are considered: $n = 4$ variables, a single primary cost function $f_A = f_1$ ($m = 1$), one scalar constraint ($K = 1$), and two secondary cost functions f_2 and f_3 ($M = 3$ disciplines in total).

We consider the case of a concave primary cost function f_A to emphasize the necessity to substitute an augmented convex cost function to it in preparation of the Nash games. The problem is defined as follows:

$$\begin{cases} \mathbf{x} = (x_1, x_2, x_3, x_4)^t \\ f_A(\mathbf{x}) = f_1 = 3 - \left(\sum_{i=1}^{4} x_i^2 + x_1 \right) \\ \mathbf{c}(\mathbf{x}) = c_1 = \sum_{i=1}^{4} x_i^2 - 1 \\ f_2(\mathbf{x}) = (x_3 - 1)^2 + \dfrac{4(x_4 - 1)^2}{r} - \dfrac{4}{r} + \dfrac{1 - x_1}{5} \\ f_3(\mathbf{x}) = -r(x_3 - 1)^2 + (x_4 - 1)^2 + r + 1 - x_1 \end{cases} \tag{73}$$

The free parameter r ($r > 0$) will be chosen later. Consequently: $\mathbf{x}_A^\star = (1, 0, 0, 0)^t$ and $f_A^\star = f_1^\star = f_2^\star = f_3^\star = 1$.

Since $f_A(\mathbf{x})$ is not convex at $\mathbf{x}_A^\star$ (in fact, uniformly concave), it is augmented by the convexity-fix term to give:

$$f_A^+(\mathbf{x}) = f_A(\mathbf{x}) + \frac{c}{2}(\mathbf{x} - \mathbf{x}_A^\star, \mathbf{x} - \mathbf{x}_A^\star) \tag{74}$$

where here the constant c must be greater than 2. Setting for example $c = 4$, one gets:

$$f_A^+(\mathbf{x}) = 5 + \sum_{i=1}^{4} x_i^2 - 5x_1 \tag{75}$$

Suppose we consider a split that assigns 2 variables to each player ($n - p = p = 2$). Here, one has:

$$\mathbf{H}_A^{+\star} = \nabla^2 f_A^{+\star} = 2\mathbf{I}_4 \tag{76}$$

and since $\nabla \mathbf{c}^\star = (2, 0, 0, 0)^t$:

$$\mathbf{P} = \mathbf{I}_4 - \frac{[\nabla \mathbf{c}^\star][\nabla \mathbf{c}^\star]^t}{[\nabla \mathbf{c}^\star]^t[\nabla \mathbf{c}^\star]} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{77}$$

and:

$$\mathbf{H}_A^{+\prime} = \mathbf{P}\mathbf{H}_A^{+\star}\mathbf{P} = 2\mathbf{P}^2 = 2\mathbf{P}, \quad \mathcal{H}_\mathbf{u} = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, \quad \mathcal{H}_\mathbf{v} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2\mathbf{I}_2. \tag{78}$$

Consequently, *Player A*'s strategy must be made of x_1 complemented by any other variable whose axis is orthogonal; one chooses x_2. Then, *Player B*'s strategy can be made of any two variables

spanning the (x_3, x_4) plane and whose axes are orthogonal. One chooses (x_3, x_4). Hence:

$$\mathbf{\Omega} = \mathbf{I}_4, \quad \mathbf{u} = \begin{pmatrix} x_1 - 1 \\ x_2 \end{pmatrix} \quad \mathbf{v} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}. \tag{79}$$

Here $\mathbf{S} = 2\mathbf{I}_2$ and $\mathbf{w} = \sqrt{2}\mathbf{v}$. Thus, to define f_B, one examines the following reduced gradients:

$$\nabla_{\mathbf{w}} f_2^\star = \frac{1}{\sqrt{2}} \nabla_{\mathbf{v}} f_2^\star = \begin{pmatrix} \sqrt{2}(x_3^\star - 1) \\ \frac{4\sqrt{2}}{r}(x_4^\star - 1) \end{pmatrix} = \begin{pmatrix} -\sqrt{2} \\ -\frac{4\sqrt{2}}{r} \end{pmatrix} \tag{80}$$

$$\nabla_{\mathbf{w}} f_3^\star = \frac{1}{\sqrt{2}} \nabla_{\mathbf{v}} f_3^\star = \begin{pmatrix} -\sqrt{2}r(x_3^\star - 1) \\ \sqrt{2}(x_4^\star - 1) \end{pmatrix} = \begin{pmatrix} \sqrt{2}r \\ -\sqrt{2} \end{pmatrix}. \tag{81}$$

Evidently, for large r, the angle between these two vectors is less than π but close to it, and this results in the unfavorable near Pareto-stationarity situation. In the limit $r \to \infty$, f_2 and f_3 could not be reduced simultaneously. Let us set $r = 4$ so that:

$$\nabla_{\mathbf{w}} f_2^\star = \begin{pmatrix} -\sqrt{2} \\ -\sqrt{2} \end{pmatrix}, \quad \nabla_{\mathbf{w}} f_3^\star = \begin{pmatrix} 4\sqrt{2} \\ -\sqrt{2} \end{pmatrix} \tag{82}$$

a) Proper split implementation

Clearly the minimum-norm element in the convex hull of the reduced gradients is:

$$\boldsymbol{\omega}_B^\star = \begin{pmatrix} 0 \\ -\sqrt{2} \end{pmatrix} = \frac{4}{5} \nabla_{\mathbf{w}} f_2^\star + \frac{1}{5} \nabla_{\mathbf{w}} f_3^\star. \tag{83}$$

It is associated with the common directional derivative

$$\sigma_B = \boldsymbol{\omega}_B^\star . \nabla_{\mathbf{w}} f_2^\star = \boldsymbol{\omega}_B^\star . \nabla_{\mathbf{w}} f_3^\star = 2. \tag{84}$$

Hence one is led to define:

$$f_B(\mathbf{x}) = \frac{4}{5} f_2(\mathbf{x}) + \frac{1}{5} f_3(\mathbf{x}) = (x_4 - 1)^2 + \frac{9}{25}(1 - x_1) \tag{85}$$

and the auxiliary cost function is formed:

$$f_{AB}(\mathbf{x}) = (1 - \varepsilon) f_A^+(\mathbf{x}) + \varepsilon f_B(\mathbf{x}) = (1 - \varepsilon) \left(5 + \sum_{i=1}^{4} x_i^2 - 5x_1 \right) + \varepsilon (x_4 - 1)^2 + \frac{9\varepsilon}{25}(1 - x_1). \tag{86}$$

On one hand, according to *Player B*'s strategy, the function $f_{AB}(\mathbf{x})$ must be minimized w.r.t. (x_3, x_4) for fixed (x_1, x_2) subject to no constraints. The result is immediate:

$$x_3 = \bar{x}_{3,\varepsilon} = 0, \quad x_4 = \bar{x}_{4,\varepsilon} = \varepsilon. \tag{87}$$

On the other hand, according to *Player A*'s strategy, (x_1, x_2) should be optimized to minimize $f_A^+(\mathbf{x})$ subject to the constraint and for fixed (x_3, x_4). This is realized by setting x_2 to 0 and x_1 to the largest value left possible for the satisfaction of the constraint after the assignment of (x_3, x_4), and that is:

$$x_1 = \bar{x}_{1,\varepsilon} = \sqrt{1 - \varepsilon^2}, \quad x_2 = \bar{x}_{2,\varepsilon} = 0. \tag{88}$$

In summary, along the continuum, one has $\varepsilon \leq 1$, and:

$$\begin{cases} \bar{\mathbf{x}}_\varepsilon = (\sqrt{1-\varepsilon^2}, 0, 0, \varepsilon) \\ \phi_A(\varepsilon) = f_A(\bar{\mathbf{x}}_\varepsilon) = 2 - \sqrt{1-\varepsilon^2} = 1 + \dfrac{\varepsilon^2}{2} + O(\varepsilon^4) \\ \phi_A^+(\varepsilon) = f_A^+(\bar{\mathbf{x}}_\varepsilon) = 6 - 5\sqrt{1-\varepsilon^2} = 1 + \dfrac{5\varepsilon^2}{2} + O(\varepsilon^4) \\ \phi_B(\varepsilon) = f_B(\bar{\mathbf{x}}_\varepsilon) = (1-\varepsilon)^2 + \dfrac{9}{25}(1 - \sqrt{1-\varepsilon^2}) = 1 - 2\varepsilon + \dfrac{59}{50}\varepsilon^2 + O(\varepsilon^4) \\ \phi_2(\varepsilon) = f_2(\bar{\mathbf{x}}_\varepsilon) = (1-\varepsilon)^2 + \dfrac{1}{5}(1 - \sqrt{1-\varepsilon^2}) = 1 - 2\varepsilon + \dfrac{11}{10}\varepsilon^2 + O(\varepsilon^4) \\ \phi_3(\varepsilon) = f_3(\bar{\mathbf{x}}_\varepsilon) = (1-\varepsilon)^2 + (1 - \sqrt{1-\varepsilon^2}) = 1 - 2\varepsilon + \dfrac{3}{2}\varepsilon^2 + O(\varepsilon^4) \end{cases} \tag{89}$$

As ε varies along the continuum, we observe the following facts that confirm our theoretical findings:

- $\mathbf{u} = O(\varepsilon^2)$ and $\mathbf{v} = O(\varepsilon)$;
- $\phi_1(0) = \phi_2(0) = \phi_3(0) = 1$;
- $\phi_1'(0) = 0$, $\phi_A(\varepsilon) = \phi_1(\varepsilon) = 1 + O(\varepsilon^2)$;
- $\phi_2'(0) = \phi_3'(0) = \phi_B'(0) = -\sigma_B = -2$.

This test-case is illustrated on Figure 1a.

b) Improper split implementation

Let us now reconsider the above test-case, but with an improper definition of the cost function f_B obtained by modifying slightly the coefficients $\{\alpha_2, \alpha_3\}$; let us now set

$$f_B(\mathbf{x}) = \frac{3}{4} f_2(\mathbf{x}) + \frac{1}{4} f_3(\mathbf{x}) = -\frac{1}{4}(x_3 - 1)^2 + (x_4 - 1)^2 + \frac{1}{4} + \frac{2}{5}(1 - x_1) \tag{90}$$

so that:

$$\begin{aligned} f_{AB}(\mathbf{x}) &= (1-\varepsilon) f_A^+(\mathbf{x}) + \varepsilon f_B(\mathbf{x}) \\ &= (1-\varepsilon)\left(5 + \sum_{i=1}^4 x_i^2 - 5x_1\right) + \varepsilon\left[\frac{1 - (x_3-1)^2}{4} + (x_4-1)^2 + \frac{2}{5}(1-x_1)\right] \end{aligned} \tag{91}$$

which should again be minimized w.r.t. (x_3, x_4) for fixed (x_1, x_2). This gives:

$$\begin{cases} 2(1-\varepsilon)x_3 - \dfrac{\varepsilon}{2}(x_3 - 1) = 0 \\ 2(1-\varepsilon)x_4 + 2\varepsilon(x_4 - 1) = 0 \end{cases} \tag{92}$$

and one gets:

$$x_3 = \bar{x}_{3,\varepsilon} = \frac{-\varepsilon}{4 - 5\varepsilon}, \quad x_4 = \bar{x}_{4,\varepsilon} = \varepsilon. \tag{93}$$

Then:

$$x_1 = \bar{x}_{1,\varepsilon} = \sqrt{1 - \varepsilon^2\left[1 + \frac{1}{(4-5\varepsilon)^2}\right]}, \quad x_2 = \bar{x}_{2,\varepsilon} = 0. \tag{94}$$

Now the limitation on ε imposed by the radical in $\bar{x}_{1,\varepsilon}$ is more stringent

$$\varepsilon \leq \varepsilon_{\lim} \approx 0.635\,. \tag{95}$$

These results are injected into the expressions of the cost functions and new graphs are plotted (see Figure 1b). In this setting, the derivatives $\phi_2'(0)$ and $\phi_3'(0)$ are not equal. Thus the minimization of f_B is not as effective on the two secondary cost functions f_2 and f_3. Evidently f_2 is not reduced as much.

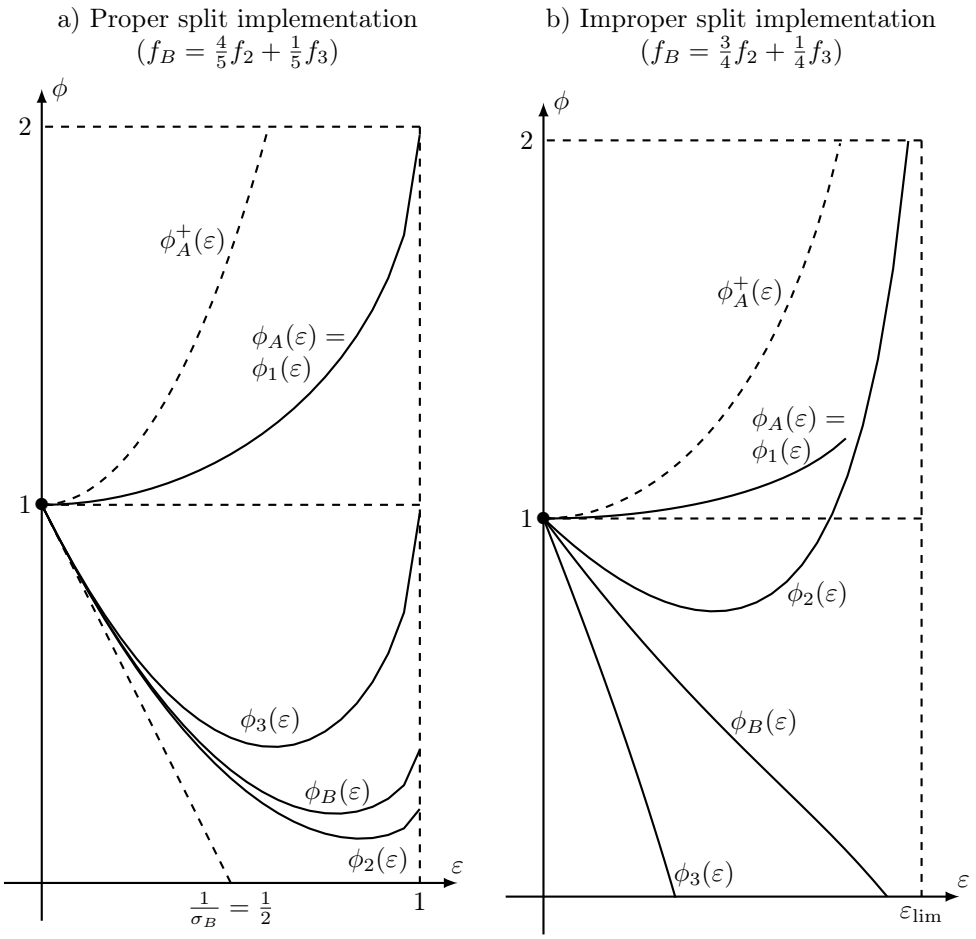


Figure 1: Test-case 4: One primary and two secondary cost functions

Utilization of the Nash-MGDA Platform. To compute the same test-case using the software platform, two files are prepared first:

- '`my_testcase.dat`': file for the specification of optional elements necessary to the program assembly, compilation and execution;
- '`my_functions.f`': file containing the subroutines defining in Fortran (F77 or F90) the primary, secondary and cost functions to be appended to the platform software.

The files specifically used to compute Test-Case TC4 are given in Tables 1 and 2 to demonstrate their format. The procedure to launch the numerical process from a site outside of Inria is omitted here. It will be described on the Inria software platform

`http://mgda.inria.fr`

in the near future.

The '`my_testcase.dat`' (Table 1) contains a series of elements necessary to the numerical procedure. Each element is prescribed in three lines as follows:

1. the first line is textual; for the first element, the text is formatted in 80 characters, and is intended to facilitate the private test-case classification by the utilizer; for the subsequent elements, it indicates the name of a variable in the code;
2. the second line is the value of the variable in the code to be read by the software;
3. the third line is blank for separation with the next element.

For the specific `my_testcase.dat` file of Table 1, the symbols read as follows:

- '`ndim`': dimension n of admissible set for vector $\mathbf{x}$;
- '`np`': dimension p of subspace V used to optimize the secondary cost functions after the territory split;
- '`mfun`': number m of primary cost functions;
- '`mtot`': total number M of cost functions;
- '`kc`': number K of constraints (all scalar and of equality type);
- '`xa_star`': components of the Pareto-optimal starting point $\mathbf{x}_A^\star$;
- '`hfdiff`': step-size used to approximate derivatives by centered finite differences;
- '`hbox`': half-size h of the hypercube used to define the geometrical database about $\mathbf{x}_A^\star$ (see Appendix A);
- '`Bkappa`': specified upper-bound κ on the condition number of Hessian matrices to enforce convexity (see Section 2, paragraph on the convexity fix);
- '`lstepmax`': number of sub-divisions of $[0, \varepsilon_{\max}]$ in the incrementation of the continuation parameter ε in the Nash games;
- '`TOL`': accuracy tolerance in the Nash-game $(\mathbf{u}, \mathbf{v})$ coordination; the accuracy tolerance in the calculation of subvector $\mathbf{u}$ for a given $\mathbf{v}$ is $TOL/100$;
- '`Lambdamax`': $\lambda_{\max}$, maximum allowable coordination iterations to achieve above accuracy tolerance;
- '`mumax`': $\mu_{\max}$, maximum allowable iterations in the procedure *ustarProcess* (3 iterations are usually sufficient).

```
testcase (character*80) :
Test-case TC4 : one prime and two secondary cost functions, and one constraint

ndim
4

np
2

mfun
1

mtot
3

kc
1

xa_star
            1.d0
            0.d0
            0.d0
            0.d0

hfdiff
1.d-4

hbox
1.d-3

Bkappa
10.d0

lstepmax
1000

TOL
1.d-4

Lambdamax
10

mumax
5
```

Table 1: The '`my_testcase.dat`' file used to compute Test-Case TC4

```
c     ------------------------------------------------------------------
      SUBROUTINE PRIME_FUNCTIONS (xv, ndim, fun, mfun)
      implicit double precision (a-h,o-z)
      dimension xv(ndim), fun(mfun)
c     primary cost function generic values

c     TEST CASE TC4

      sum = 0.d0
      do i = 1,4
         sum = sum + xv(i)**2
      end do

      fun(1) = 3.d0 - (sum+xv(1))

      RETURN
      END

c     ------------------------------------------------------------------
      SUBROUTINE SECOND_FUNCTIONS ( xv, ndim, fun, mfun, mtot)
      implicit double precision (a-h,o-z)
      dimension xv(ndim), fun(mtot)
c     secondary cost function generic values

c     TEST CASE TC4

      fun(2) = (xv(3)-1.d0)**2 + (xv(4)-1.d0)**2 - 1.d0
     ~     + 0.2d0 * (1.d0-xv(1))

      fun(3) = -4.d0 * (xv(3)-1.d0)**2 + (xv(4)-1.d0)**2 + 5.d0 -xv(1)

      RETURN
      END

c     ------------------------------------------------------------------
      SUBROUTINE CONSTRAINTS ( xv, ndim, cfun, kc)
      implicit double precision (a-h,o-z)
      dimension xv(ndim), cfun(kc)

c     TC4 : unit sphere

      sum = 0.d0
      do i = 1,4
         sum = sum + xv(i)**2
      end do

      cfun(1) = sum - 1.d0

      RETURN
      END
```

Table 2: The '`my_functions.f`' file used to compute Test-Case TC4

The numerical process is executed in four steps by a shell file:

1. **Reading:** the file '`my_testcase`' is read and this creates the files: '`my_dimensions.dat`' (for array dimension specification) and '`my_other_info.dat`' (to make the remaining part of '`my_testcase.dat`' available);

2. **Assembling and compiling:** the code is assembled using the specified dimensions for arrays, and the user-specified subroutines, with a possible link to a user depository directory of object files, and compiled with a link to the Lapack Library. This step creates the executable file '`nash_mgda.f.out`'.

3. **Executing:** The execution of '`nash_mgda.f.out`' consists of the 15 preparation steps of Subsection 3.1 followed by the step-by-step computation of the Nash equilibria described in Subsection 3.2. At completion of execution, the following files have been created:

   - '`meta_nash_mgda_run_report.txt`': formatted output of the global execution run;
   - '`nash-equilibria.dat`': numerical data defining the continuum of Nash equilibria, one line per equilibrium point, according to the following write statement:

```
      write (88,*) 'step-index=', lstep,
     &        'epsilon=', vepsi,
     &        'x-vector=', (xv(i), i=1,ndim),
     &        'functions...f_j:f_j*=', (fun(j)/fun_star(j), j=1,mtot),
     &        'constraints...c_k=', (cfun(k), k=1,kc),
     &        'fa...faplus...fb...fbtilde=', fa, faplus, fb, fbtilde,
     &        'ubar=', (u1(iu), iu=1,ndim1),
     &        'vbar=', (v1(iv), iv=1,np)
```

     where the code nomenclature is as follows:

     – '`lstep`': ε-incrementation index ℓ;
     – '`vepsi`': $\varepsilon = \varepsilon_\ell$;
     – '`xv`': vector $\overline{\mathbf{x}}_\varepsilon$;
     – '`fun_star(j)`': $f_j(\mathbf{x}_A^\star)$;
     – '`fa, faplus, fb, fbtilde`': $f_A(\overline{\mathbf{x}}_\varepsilon)$, $f_A^+(\overline{\mathbf{x}}_\varepsilon)$, $f_B(\overline{\mathbf{x}}_\varepsilon)$, $\tilde{f}_B(\overline{\mathbf{x}}_\varepsilon)$;
     – '`u1`': $\overline{\mathbf{u}}_\varepsilon$ (dimension $n-p$);
     – '`v1`': $\overline{\mathbf{u}}_\varepsilon$ (dimension p).

   - '`nash.gnu`' is an executable `gnuplot` program;

4. **Plotting:** the elaboration of plots is conducted automatically by execution of the `gnuplot` program '`nash.gnu`'. This execution creates the files
   '`nash-points.pdf`', '`nash-functions.pdf`', '`nash-constraints.pdf`',
   (and analogs in postscript format) corresponding to the plot of the components of the equilibrium point $\overline{\mathbf{x}}_\varepsilon$ as ε increases, and the analogous plots of the functions and constraints respectively. Usually, the plots are interrupted at some value of ε inferior to the limit of convexity $\varepsilon_{\max}$.

Remark 3
In the calculation, no assumptions are made on the intervals of variation of the components of vector $\mathbf{x}$. *However, in the plotting of results, whenever a component* x_i *falls outside of the interval* $[-\pi/2, \pi/2]$, *it is represented by symmetry w.r.t. the limit. This is convenient particularly when sine functions are used in the parameterization. This is the case in particular of* x_3 *in Figure 14 as well as in Figure 20. Should this convention be inadequate to the representation of the variable* $\mathbf{x}$, *the utiliizer is invited to devise one's own graphics from the data of '*`nash-equilibria.dat`*'.*

In the specific example of Test-Case TC4, the plots of the variables, functions and (single) constraint are given in Figures 2, 3 and 4. These computed results match very accurately the analytical results of (89) and Figure 1. In particular, recall that $\bar{x}_1(\varepsilon) = \sqrt{1-\varepsilon^2}$, and

$$\frac{d\bar{x}_1}{d\varepsilon} = \infty \tag{96}$$

at $\varepsilon = 1$. However, this singularity does seem to cause a serious difficulty to the numerical process, up to the last discretization point before the limit $\varepsilon_{\max} = 1$, except near the limit, for enforcing the constraint which is subject to the accuracy tolerance.

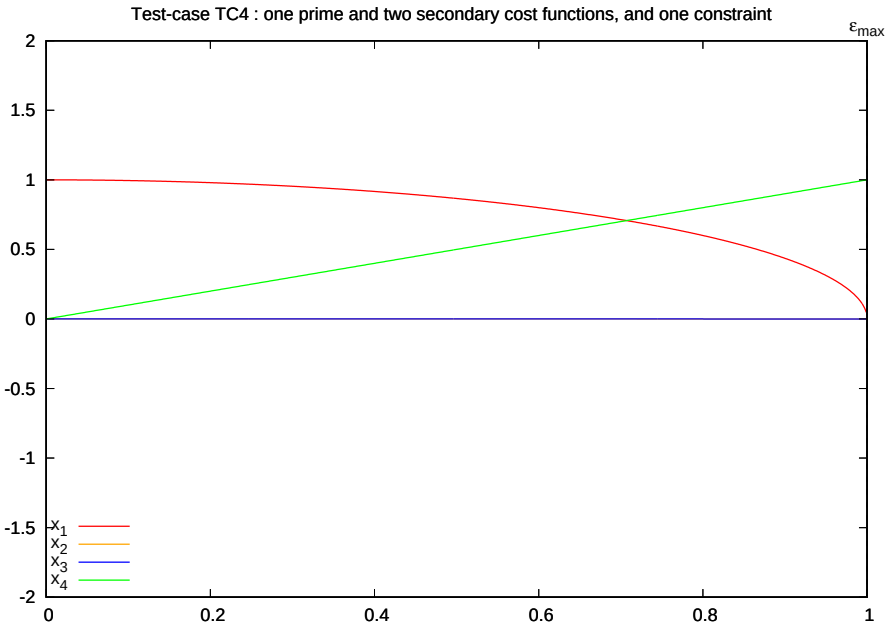


Figure 2: Test-case TC4: variables along the continuum of Nash equilibria

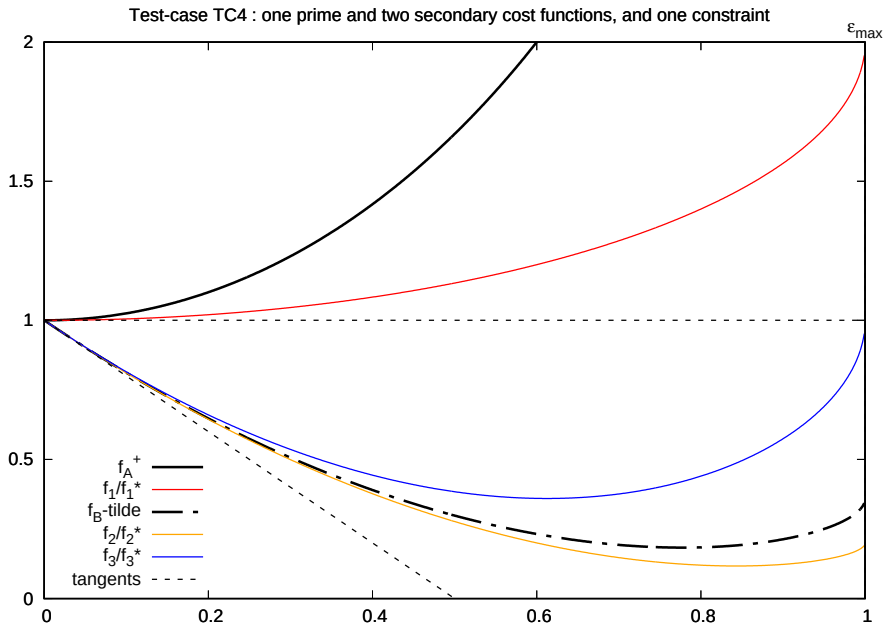


Figure 3: Test-case TC4: cost functions along the continuum of Nash equilibria

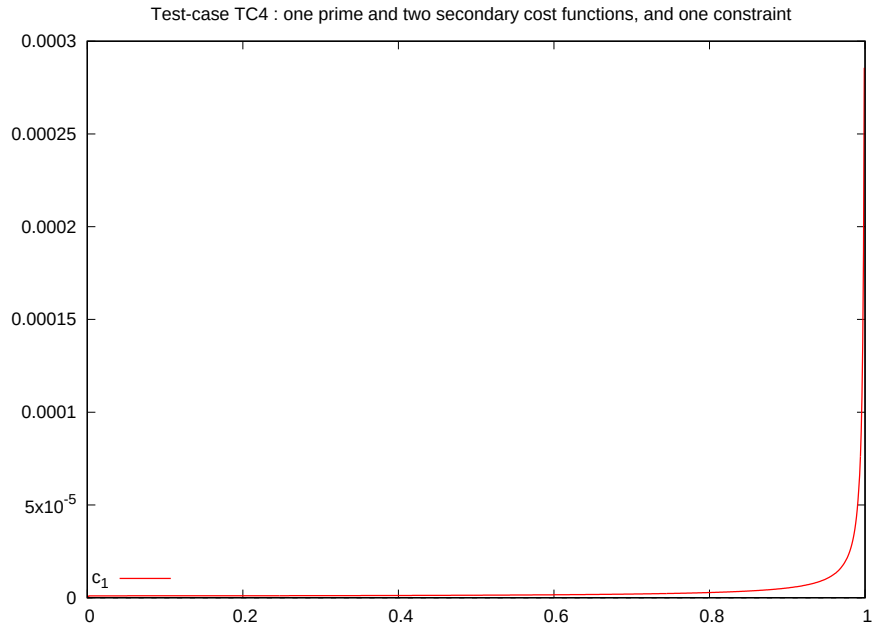


Figure 4: Test-case TC4: cost functions along the continuum of Nash equilibria

4.2 Variations about the classical Fonseca-Fleming test-case

The definition of the classical Fonseca-Fleming test-case can be found in [10] and [16]. For $n = 2$ variables, it consists of the minimization of the two functions:

$$\begin{cases} f_1(\mathbf{x}) = 1 - \exp\left[-\sum_{i=1}^{2}\left(x_i - \frac{1}{\sqrt{2}}\right)^2\right] \\ f_2(\mathbf{x}) = 1 - \exp\left[-\sum_{i=1}^{2}\left(x_i + \frac{1}{\sqrt{2}}\right)^2\right] = f_1(-\mathbf{x}) \end{cases} \tag{97}$$

subject to the following bounds on the variables

$$-4 \leq x_1 \leq 4, \quad -4 \leq x_2 \leq 4. \tag{98}$$

(Here the questions related to larger n are not considered.)

For purpose of demonstration of the utilization of the MGDA Platform in a constrained problem, in [8], the problem was enhanced to 4 variables by the introduction of 2 additional slack variables x_3 and x_4 subject to the nonlinear constraints

$$\begin{cases} c_1(\mathbf{x}) = x_1 - 4\sin x_3 = 0 \\ c_2(\mathbf{x}) = x_2 - 4\sin x_4 = 0. \end{cases} \tag{99}$$

One feature of this problem is that the Pareto front in the (f_1, f_2) plane which is given in parametric form by the following

$$x_1(t) = x_2(t) = \frac{t}{\sqrt{2}}, \tag{100}$$

$$f_1(t) = 1 - \exp\left[-(t-1)^2\right], \quad f_2(t) = 1 - \exp\left[-(t+1)^2\right], \tag{101}$$

$$-1 \leq t \leq 1, \tag{102}$$

is concave except in two small portions, at the extremities, in the neighborhood of the minimum of either cost function [16] (see Figure 5). Hence in the intermediate concave portion, at least one function is not convex, and this has permitted to demonstrate that MGDA, with a proper setting of the optional parameters, would nevertheless converge.

In the present experiment, we further extend the problem dimension to $n = 6$ in order to introduce two new cost functions and consider a prioritized optimization. We now redefine

$$\begin{cases} f_1(\mathbf{x}) = 1 - \exp\left[-\sum_{i=1}^{2}\left(x_i - \frac{1}{\sqrt{2}}\right)^2\right] + \Theta_1 \\ f_2(\mathbf{x}) = 1 - \exp\left[-\sum_{i=1}^{2}\left(x_i + \frac{1}{\sqrt{2}}\right)^2\right] + \Theta_2 \end{cases} \tag{103}$$

where

$$\begin{cases} \Theta_1 = 200x_1^2(x_5^2 + x_6^2) \\ \Theta_2 = 200x_2^2(x_5 + x_6)^2. \end{cases} \tag{104}$$

Additionally, we let:

$$\begin{cases} f_3 = \ln\left(2(x_5 - x_3)^2\right) + 10 \\ f_4 = \ln\left(\frac{1}{2}(x_6 - x_4)^2\right) + 10. \end{cases} \tag{105}$$

Now, the pair (f_1, f_2) is considered as prioritized cost functions subject to the constraints $c_1(\mathbf{x}) = c_2(\mathbf{x}) = 0$, and the pair (f_3, f_4) as secondary cost functions. Clearly, on the Pareto set $x_5 = x_6 = 0$. Hence the trace of this set in the (f_1, f_2) plane remains the same.

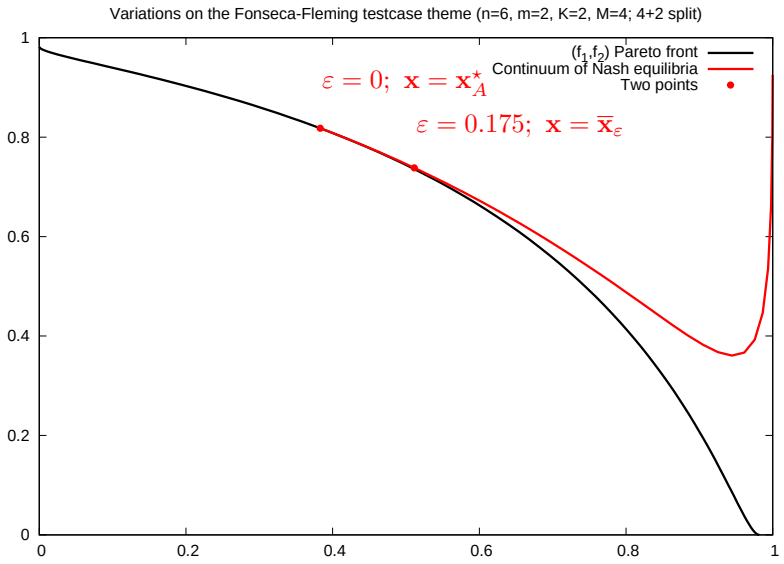


Figure 5: Trace in the (f_1, f_2) plane of the Pareto front associated with the Fonseca-Fleming test-case (black solid line) - the figure also represents the trace of the continuum of Nash equilbria that originates from point $\mathbf{x}_A^\star$ (red solid line)

We picked the following initial Pareto-optimal point:

$$\mathbf{x}_A^\star = (0.215730, 0.215730, 0.053959, 0.053959, 0, 0) \tag{106}$$

visualized on Figure 5 by the symbol • ($\varepsilon = 0$). Four variables were retained for the primary function strategy and two for the secondary function strategy ($p = 2$). The other parameters were set according to Table 3.

The variation with ε of the variables, functions and constraints is illustrated on Figures 6, 7 and 8 respectively, in view of which the following observations are made:

- The two primary cost functions f_1 and f_2 are separated by the primary steering function f_A^+. At the start, f_1 increases linearly with ε and f_2 decreases while their steering function f_A^+ increases but with an initial null derivative, thus preserving the Pareto-stationarity condition to second-order in ε.

- The two secondary cost functions f_3 and f_4 first decrease linearly with ε following very closely their steering function over about half of the ε interval; then non monotonic behavior is observed. The initial derivative is correctly calculated.

- The constraint c_2 is quadratic and preserved throughout since its associated meta-model is exact. The constraint c_1 meta-model is only approximate, but accurate, and the constraint remains very small over some 80% of the ε interval.

```
testcase (character*80) :
Variations about the FF testcase; constraint with quartic term 0.25*(x1-x2)^4

ndim
6

np
2

mfun
2

mtot
4

kc
2

xa_star
            0.215730
            0.215730
            0.053959
            0.053959
    0.
    0.

hfdiff
1.d-2

hbox
0.5d0

Bkappa
20.d0

lstepmax
1000

TOL
1.d-4

Lambdamax
20

mumax
20
```

Table 3: The '`my_testcase.dat`' file used to compute the Fonseca-Fleming testcase.

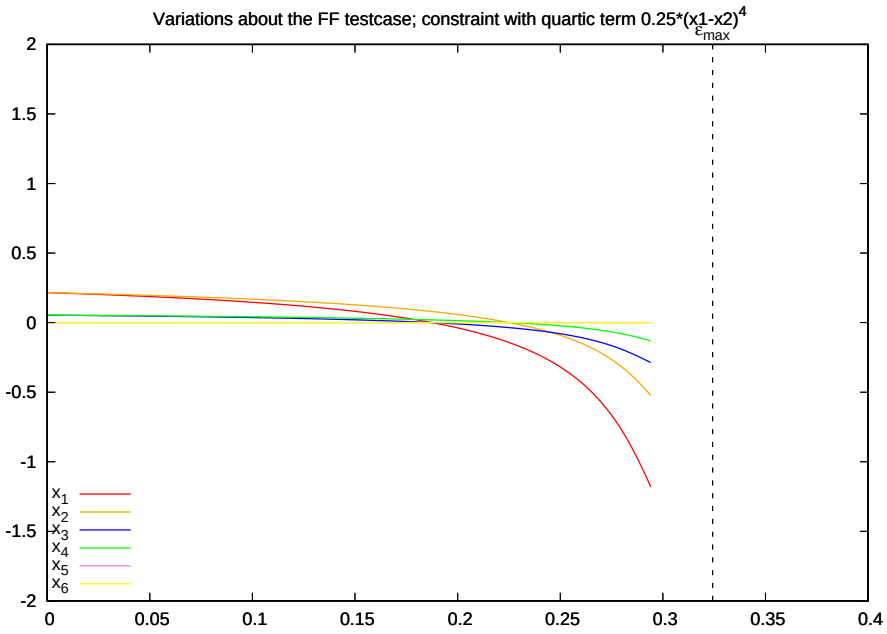


Figure 6: Variations on the Fonseca-Fleming test-case involving six variables, two prime functions, two secondary cost functions, and two constraint functions - optimization variables in terms of ε

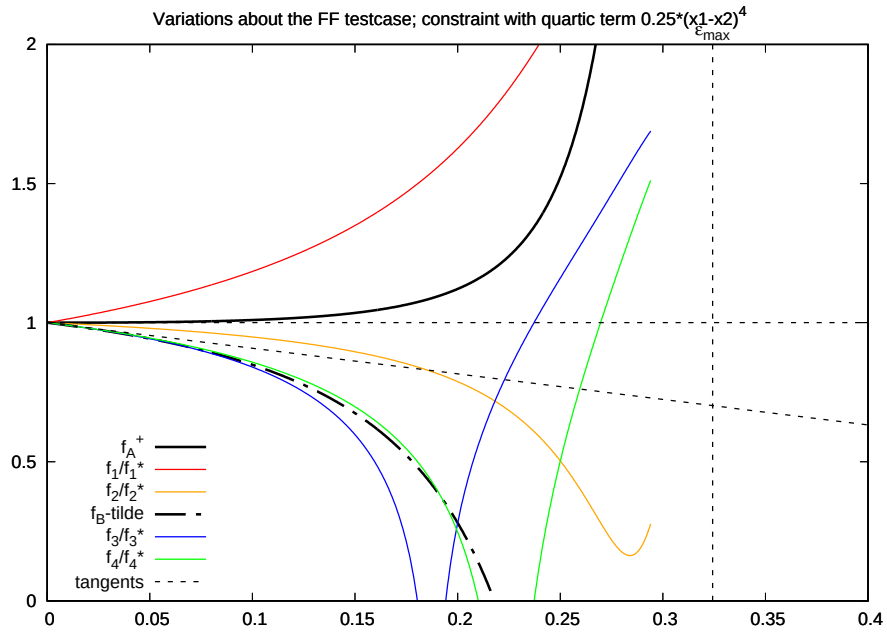


Figure 7: Variations on the Fonseca-Fleming test-case involving six variables, two prime functions, two secondary cost functions, and two constraint functions - functions of interest in terms of ε

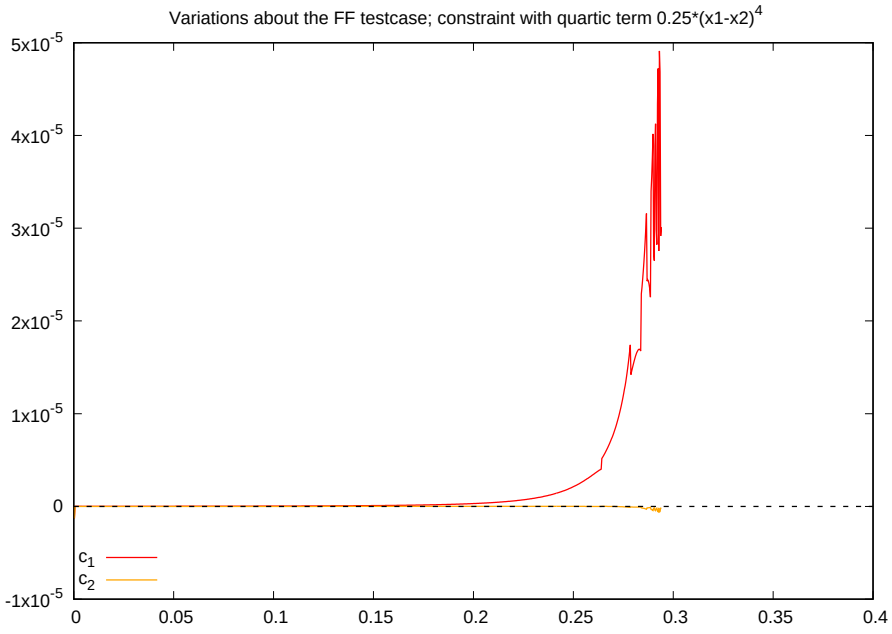


Figure 8: Variations on the Fonseca-Fleming test-case involving six variables, two prime functions, two secondary cost functions, and two constraint functions - constraints in terms of ε

5 A test-case in structural analysis: design of a sandwich panel

Mechanical problem, critical forces and design variables. The test-case is inspired by the mechanical design of a sandwich panel considered in [14], but in a modified setting. The structural element is a three-layer non-symmetric sandwich panel with aluminum skins and a regular hexagonal honeycomb core. We refer to [14] for a detailed description of the physical model and general geometry.

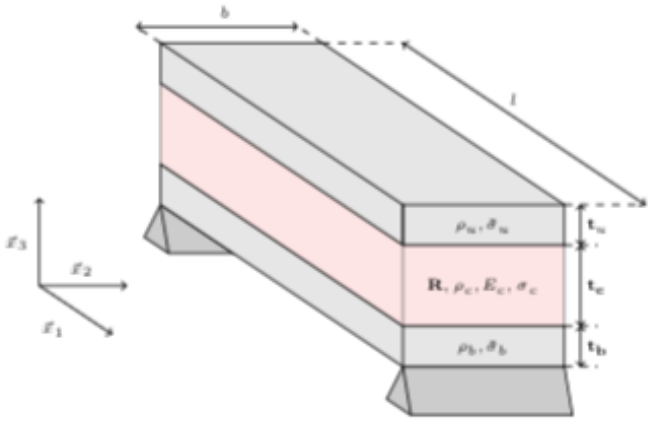


Figure 9: Geometrical definition of the sandwich panel.

The objective of the testcase is to optimize the thicknesses (t_u, t_b, t_c) and the ratio thickness/length ($\mathbf{R} = t/\ell$) of the cell walls for best mechanical properties.

The sandwich mass is proportional to the quantity

$$M = \rho_u \mathbf{t}_u + \rho_R \mathbf{t}_c + \rho_b \mathbf{t}_b \tag{107}$$

where:

$$\rho_u = \rho_b = \rho = 2700 \text{ kg/m}^3, \quad \rho_R = \frac{2\mathbf{R}}{\sqrt{3}}\rho. \tag{108}$$

Gibson et Ashby [11] have provided formulas expressing certain mechanical characteristics of an equivalent element. Generally speaking, one wishes to minimize mass and maximize several, hereafter two, critical forces of failure in flexion.

The first failure mode under consideration is illustrated by Figure 10. It corresponds to the core indentation under the critical load $F_{c,1}$. The second failure mode is illustrated by Figure 11. It occurs when the elastic resistance of the lower skin is exceeded under the critical load $F_{c,2}$. We admit that these critical forces are given by the following formulas:

$$F_{c,1} = 2b\mathbf{t}_u\sqrt{\sigma_R\sigma_u} + ab\sigma_R, \quad F_{c,2} = \frac{4}{\ell}b\mathbf{t}_b\left(\mathbf{t}_c + \frac{\mathbf{t}_u + \mathbf{t}_b}{2}\right)\sigma_b, \tag{109}$$

where:

$$a = 0.01 \text{ m}, \quad b = 0,03 \text{ m}, \quad \sigma_u = \sigma_b = \sigma = 3500 \text{ MPa}, \quad \sigma_R = 5.6\mathbf{R}^{\frac{5}{3}}\sigma, \quad \ell = 0.1 \text{ m}. \tag{110}$$

The panel should be sized to realize an adequate trade-off between the minimization of mass and the maximization of the critical failure forces. The sizing is subject to the following bound constraints:

$$\begin{gathered}\mathbf{t}_u,\ \mathbf{t}_b \in [0.03, 0.14] \\ \mathbf{t}_c \in [0.05, 0.19] \\ \mathbf{R} \in [0.01, 0.20].\end{gathered}$$

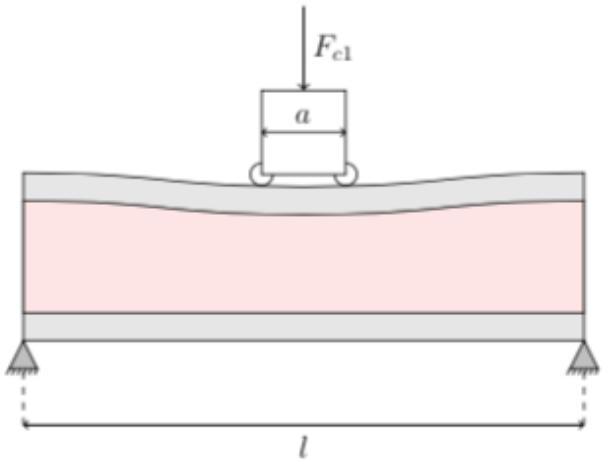


Figure 10: First failure mode: core indentation.

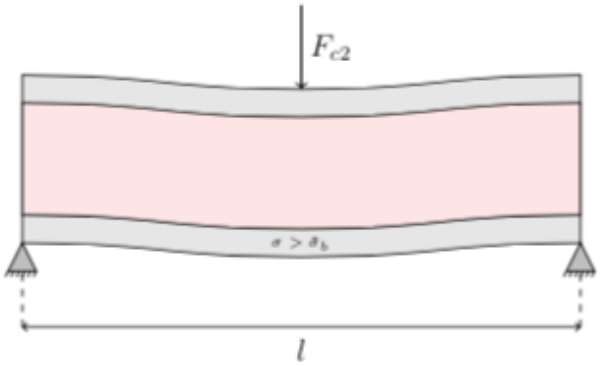


Figure 11: Second failure mode: exceeding the elastic resistance of the lower skin

Additionally, the total thickness is to be limited by:

$$e = \mathbf{t}_u + \mathbf{t}_b + \mathbf{t}_c \leq 0.25. \tag{111}$$

(All lengths are expressed in meters.)

Mathematical formulation and primary optimization problem In order to satisfy the bound constraints automatically, the following new variables $(x_1, \ldots, x_4)$ (initially all in $[-\pi/2, \pi/2]$) are introduced, and one lets:

$$\begin{aligned}
\mathbf{t}_u &= 0.085 + 0.055 \sin x_1 \\
\mathbf{t}_b &= 0.085 + 0.055 \sin x_2 \\
\mathbf{t}_c &= 0.120 + 0.070 \sin x_3 \\
\mathbf{R} &= 0.105 + 0.095 \sin x_4.
\end{aligned}$$

The bound constraint on the total thickness is then expressed as the following equality constraint:

$$c_1(\mathbf{x}) = 0.29 + 0.055 \left(\sin x_1 + \sin x_2\right) + 0.070 \sin x_3 - 0.25 + x_5^2 = 0 \tag{112}$$

where x_5 is a slack variable. In this formulation, the optimization is conducted in R^5 to optimize the vector

$$\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) \tag{113}$$

under the above equality constraint.

In order to scale the mass-related cost function adequately, the following median mass is considered

$$\overline{M} = \overline{\rho_u}\,\overline{\mathbf{t}_u} + \overline{\rho_{\mathbf{R}}}\,\overline{\mathbf{t}_c} + \overline{\rho_b}\,\overline{\mathbf{t}_b} = \rho\left(\overline{\mathbf{t}_u} + \overline{\mathbf{t}_b}\right) + \frac{2\overline{\mathbf{R}}}{\sqrt{3}}\rho\overline{\mathbf{t}_c} \tag{114}$$

and this gives: $\tau = \overline{M}/\rho = 0.170 + \frac{2}{\sqrt{3}} \times 0.120 \times 0.105 \doteq 0.1845$ m. Hence, the scaled mass-related cost function is defined as follows:

$$\begin{aligned} f_1(\mathbf{x}) = \frac{M}{\overline{M}} &= \frac{0.170 + 0.055(\sin x_1 + \sin x_2) + (0.120 + 0.070 \sin x_3)\,\frac{2\mathbf{R}}{\sqrt{3}}}{\tau} \\ &= \frac{0.170 + 0.055(\sin x_1 + \sin x_2) + (0.120 + 0.070 \sin x_3)\,\dfrac{0.21 + 0.19 \sin x_4}{\sqrt{3}}}{\tau}. \end{aligned} \tag{115}$$

Secondly, concerning the critical forces, note that in [14], the purpose was to demonstrate the potential of the stochastic extension SMGDA of MGDA to handle nonconvex problems including variability in the data. For this, the smaller critical force

$$F_c = \min\left(F_{c,1}, F_{c,2}\right) \tag{116}$$

was maximized. This led to a non-differentiable formulation well handled by SMGDA. However here the objective is different. We wish to provide another example of the prioritized multi-objective approach in a deterministic, differentiable optimization context. For this, the following cost function is considered instead:

$$f_2(\mathbf{x}) = \frac{ab\sigma}{2}\left(\frac{1}{F_{c,1}} + \frac{1}{F_{c,2}}\right) = \frac{1}{4\dfrac{\mathbf{t}_u}{a}\sqrt{\dfrac{\sigma_R}{\sigma}} + 2\dfrac{\sigma_R}{\sigma}} + \frac{a\ell}{8\mathbf{t}_b\left(\mathbf{t}_c + \dfrac{\mathbf{t}_u + \mathbf{t}_b}{2}\right)} \tag{117}$$

where:

$$\frac{\sigma_R}{\sigma} = 5.6\mathbf{R}^{\frac{5}{3}}. \tag{118}$$

In this way, the minimization of $f_2(\mathbf{x})$ permits to avoid excessive values of either critical force, and this serves the practical purpose.

In summary, the primary multi-objective optimization problem consists in minimizing the cost functions $\big(f_1(\mathbf{x}), f_2(\mathbf{x})\big)$ of (115)-(117) in R^5 under the equality constraint (112).

Primary optimization experiment. An experiment was devised to establish a fairly accurate Pareto front associated with the minimization of (f_1, f_2) under the constraint $c_1(\mathbf{x}) = 0$.

Initialization

The variable x_1 was discretized uniformly by 10 values in the interval $[-\pi/2, \pi/2]$; so were the variables x_2 and x_3 independently. Out of the 1000 corresponding triplets, only 307 permitted to satisfy the constraint (112) by adequate adjustment of the slack variable x_5. Each admissible triplet was completed by one of 10 similar discrete values of x_4, yielding a total of 3070 potential starting points $\mathbf{x}$ filling rather well the admissible domain.

Convergence paths

For each starting point, the Quasi-Riemannian MGDA iteration [8] was launched providing a convergence trajectory ending up at a certain limit point. The following assignments were made

in the numerical procedure.

$$\begin{cases} \texttt{iscale = logmode} = 0 \\ \texttt{eps0} = 1 \\ \texttt{itermax} = 10 \\ \texttt{eps\_Hdiag} = 10^{-10} \\ \texttt{TOL} = 10^{-4} \\ \texttt{boundc} = 5.10^{-4} \\ \texttt{maxiter} = 12. \end{cases} \tag{119}$$

This numerical setting was not equally satisfactory in all cases, and the insufficiently-converged paths as well as those ending up at a point dominated in efficiency were removed. This resulted in a set of 1174 endpoints that provided the fairly accurate representation of the Pareto front of Figure 12.

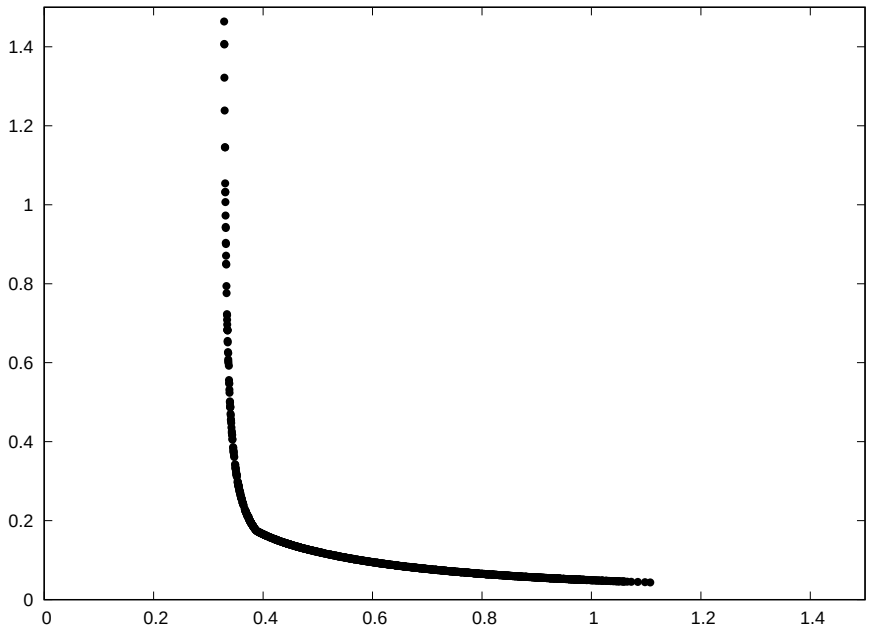


Figure 12: Sandwich panel optimization - Numerical Pareto front associated with the constrained minimization of (f_1, f_2)

Preliminary experiment of prioritized optimization. In order to proceed with an exercise of prioritized optimization, a single previously-obtained Pareto-optimal solution was elected as a starting point. We picked the point corresponding to:

$$\begin{cases} x_1 = x_1^\star = 0.26494773361885992635 \\ x_2 = x_2^\star = -0.77734301336741629118 \\ x_3 = x_3^\star = -0.55331172659353800025 \\ x_4 = x_4^\star = 1.53051540327955315490 \\ x_5 = x_5^\star = 0.14477446385936240669 \end{cases} \tag{120}$$

for which

$$\begin{cases} f_1 = f_1^\star = 0.89426570728178023106 \\ f_2 = f_2^\star = 0.05666958030488201958. \end{cases} \tag{121}$$

(The variable x_5 was slightly readjusted in order to satisfy the constraint exactly.)

Remark 4
The primary optimization experiment was conducted very redundantly with a large set of points, although only one was sufficient to the following. This was done here for purpose of demonstration of the link between the Pareto front and the continuum of Nash equilibria. In a more complex experiment only one, or a few Pareto optimal points are necessary to launch the prioritized optimization sequence, which in a sense, is meant to replace the prior identification of a segment of the front.

Then, in order to exercise our technique of prioritized optimization, the primary cost functions f_1 and f_2 were replaced by the following ones for a better scaling:

$$\begin{cases} \mathsf{f}_1 = \exp\left(10\left(\dfrac{f_1}{f_1^\star} - 1\right)\right) \\ \mathsf{f}_2 = \exp\left(5\left(\dfrac{f_2}{f_2^\star} - 1\right)\right) \end{cases} \tag{122}$$

and the following secondary cost function was considered:

$$\mathsf{f}_3(\mathbf{x}) = \exp\left(\frac{x_1}{x_1^\star} - 1\right) + \frac{1}{100} f_1(\mathbf{x})^2 \tag{123}$$

in an attempt to define designs of smaller thicknesses $\mathbf{t}_u$ and smaller masses.

Three variables were assigned to the constrained optimization of $\mathbf{u}$ for fixed $\mathbf{v}$, and thus two variables to the unconstrained) optimization of $\mathbf{v}$ for fixed $\mathbf{u}$ ($p = 2$).

The obtained results are in qualitative compliance with the theory. The pattern of cost functions is shown on Figure 13. The variation of f_1 (in red) indicates that mass is only very moderately diminished, and this occurs only for $\varepsilon < 0.25$, but x_1 (in blue) does decrease of some 60% at this point. The primary steering function $f_A^\star$ separate the two primary functions over a significant interval. The function f_3 follows very well its quadratic meta-model, and the initial tangent is correctly calculated. Much further beyond, as the limit of convexity is approached, the trends cease to be meaningful.

The variation with ε of the 5 optimization variables is shown on Figure 14. The kink in the plot of x_3 is not associated with a derivative discontinuity; it only reflects a plotting artifact: when the argument x_i of a sine function used in the parameterization, nothing is changed in the numerical calculation, but in the corresponding plot, x_i is conventionally represented by symmetry; in this way the plots all fall in the interval $[-\pi/2, \pi/2]$.

Figure 15 provides the variation of the constraint with ε. Evidently, the quadratic constraint meta-model which is upgraded after every Nash equilibrium point has been determined permits to satisfy the constraint very accurately over most of the continuum.

Lastly, the path associated with the continuum is depicted (in red) over a relevant part of the Pareto front in the (f_1, f_2) plane on Figure 16. The symbols • indicate the starting point ($\varepsilon = 0$) and the point at which $\varepsilon/\varepsilon_{\max} = 0.25$, where mass has just begun to increase again. At the latter, (f_1, f_2) have essentially not changed, but the new solution is such that x_1 has been reduced significantly (more than 50%). The most important observation to be made is that the path of the continuum is tangent to the Pareto front, as it should be, and it fact, very close to it over a small segment sufficient to permit the reduction of f_3.

These results do not make great physical sense from the structural analysis viewpoint, since the secondary cost function was defined without real physical relevance. This test-case will be revisited by considering more meaningful secondary cost functions related to vibrational modes or deformations under specified loads.

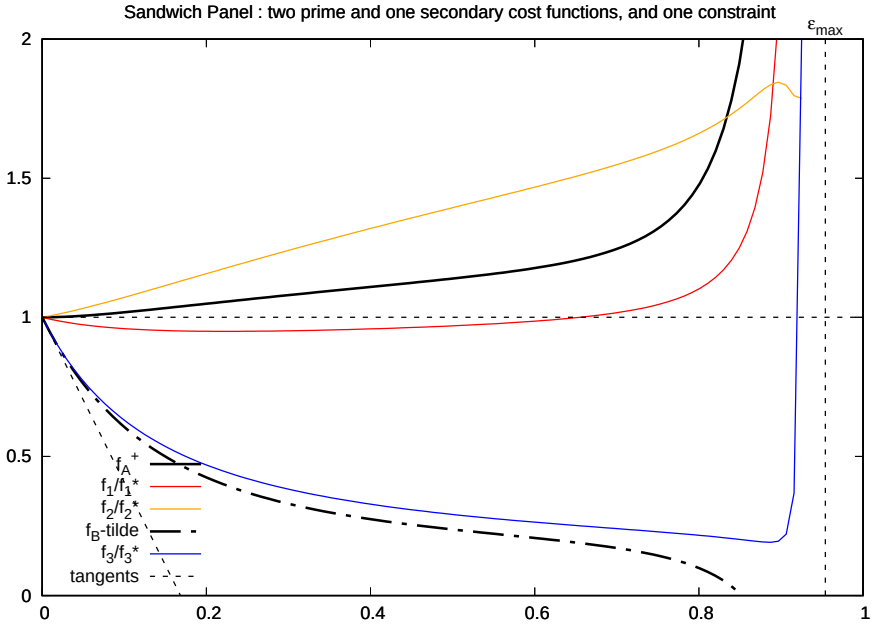


Figure 13: Sandwich panel prioritized optimization - Functions along the continuum

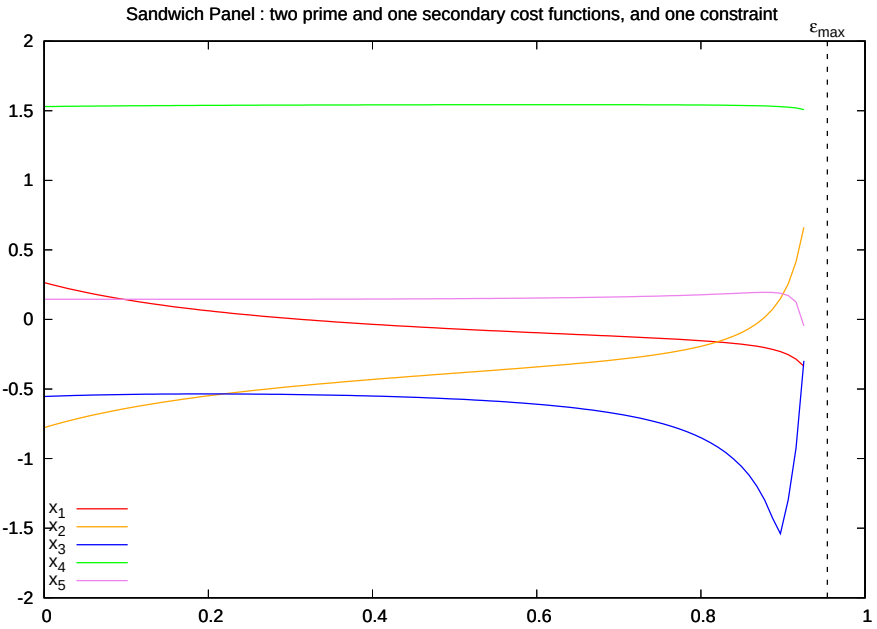


Figure 14: Sandwich panel prioritized optimization - Variables along the continuum

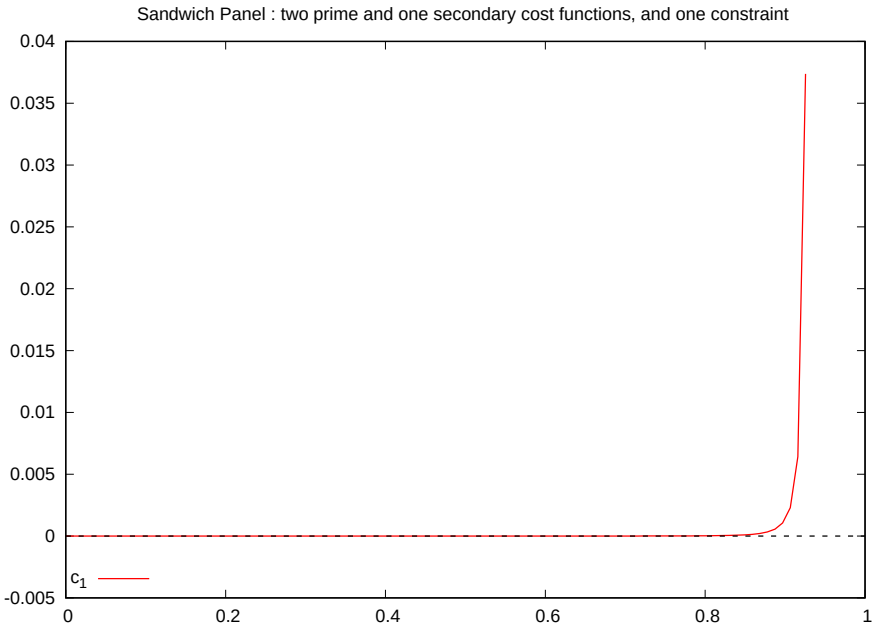


Figure 15: Sandwich panel prioritized optimization - Functions along the continuum

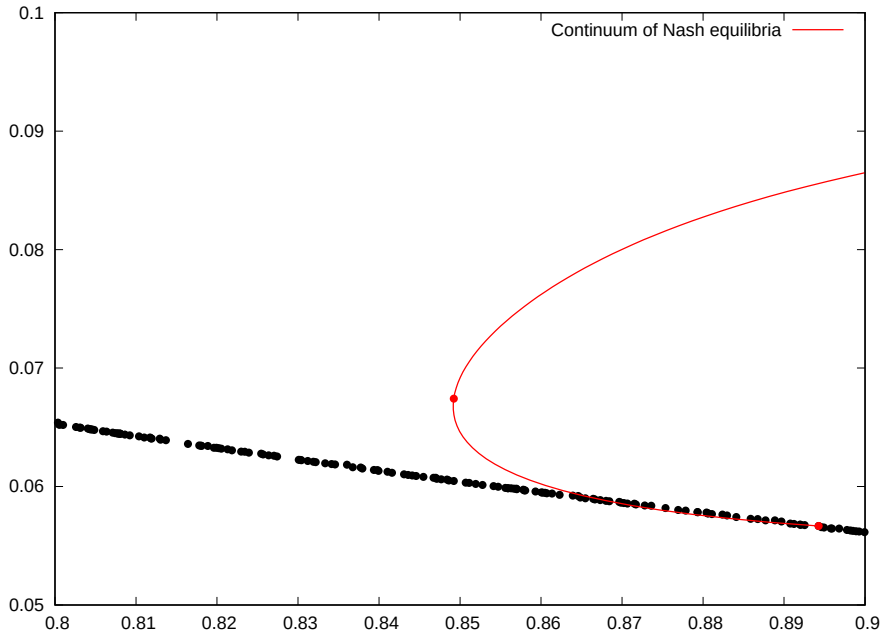


Figure 16: Sandwich panel prioritized optimization - Path of the continuum (in red) compared to Pareto front (black symbols) in the (f_1, f_2) plane; the symbols • indicate the starting point ($\varepsilon = 0$) and the point at which $\varepsilon/\varepsilon_{\max} = 0.25$ of the continuum.

6 A test-case in flight mechanics

This section was developed jointly with R. Duvigneau (Inria)[3]. It results from a collaboration with M. Ravachol, Dassault Aviation.

Flight-mechanics software, variables and functions. During the period 2006-2009, the ANR, the French Research Agency, has supported the project *"Optimisation Multidisciplinaire"* (Multi-Disciplinary Optimization, MDO) in which fourteen partners from Industry and Academics contributed to a variety of MDO problematics and case studies [2] [3]. The project had a sequel involving new partners and permitting to investigate new problems [4]. In the course of these initiatives, Dassault Aviation had provided partners with a software permitting to exercise numerical procedures for multidisciplinary or multi-objective problems in flight-mechanics design [15]. This software has been used again here for purpose of illustration of our method for prioritized multi-objective optimization without claiming to contribute to the technical and technological field of airplane design.

The software is made of several procedures that calculate "performance quantities" according to classical laws of flight-mechanics, the Breguet laws in particular, as functions of 15 design variables $\{X_i\}$ $(i = 1, \ldots, 15)$, all subject to prescribed interval constraints $X_{i,\min} \leq X_i \leq X_{i,\max}$. For the design of a SuperSonic Business Jet (SSBJ) these variables and their bounds are described in Table 4.

symbol (X_i)	significance (unit)	lower bound $X_{i,\min}$	upper bound $X_{i,\max}$
z	cruise altitude (m)	8000	18500
xmach	cruise Mach number	1.6	2.0
S	wing reference surface (m^2)	100	200
phi0w	wing leading-edge sweep angle (^{o})	40	70
phi100w	wing trailing-edge sweep angle (^{o})	-10	20
xlw	wing taper ratio	0.05	0.50
t_cw	wing relative thickness	0.04	0.08
phi0t	vertical-tail leading-edge sweep angle (^{o})	40	70
phi100t	vertical-tail trailing-edge sweep angle (^{o})	0	10
xlt	vertical-tail taper ratio	0.05	0.50
t_ct	vertical-tail relative thickness	0.05	0.08
dfus	fuselage diameter (m)	2.0	2.5
wfuel	fuel mass (kg)	15,000	40,000
alpha	landing maximum angle of attack (^{o})	10	15
xfac	mlw/tow, landing to take-off mass ratio	0.85	0.95

Table 4: Physical design variables in the flight-mechanics test-case and their specified bounds

In order to respect the prescribed interval constraints, we have used, for each design variable X_i, a transformation of the following type

$$X_i = \frac{X_{i,\,\max} + X_{i,\min}}{2} + \frac{X_{i,\,\max} - X_{i,\min}}{2} \sin x_i \tag{124}$$

and used the dimensionless vector $\mathbf{x} = \{x_i\}$ as the optimization variable.

The software permits to compute in particular

- mass at take-off, M (to be minimized),
- range, R (to be maximized),
- approach speed, V (to be minimized),

[3] regis.duvigneau@inria.fr

- take-off distance, D (to be minimized, or maintained below a specified upper bound).

The take-off distance D was subject to the following inequality constraint:

$$D \leq 1828 \text{ m}. \tag{125}$$

This led us to introduce an additional variable, x_{16}, as a slack variable, and to replace the above inequality by the following equality constraint:

$$c_1(\mathbf{x}) = \frac{D}{1828} - 1 + x_{16}^2 = 0 \tag{126}$$

thus increasing by one the dimension of the vector $\mathbf{x}$.

Three cost functions were defined in relation to mass, range and approach speed:

$$f_1(\mathbf{x}) = \exp\left[\gamma\left(\frac{M}{M^\star} - 1\right)\right] \tag{127}$$

$$f_2(\mathbf{x}) = \exp\left[\gamma\left(\frac{R^\star}{R} - 1\right)\right] \tag{128}$$

$$f_3(\mathbf{x}) = \exp\left[\gamma\left(\frac{V}{V^\star} - 1\right)\right] \tag{129}$$

where starred quantities correspond to initial values at $\mathbf{x} = \mathbf{x}_A^\star$ and are case-dependent as well as the positive numerical parameter γ. In this setting, the parameter γ appears as a scaling factor in the expression of the initial gradients of the cost functions, since it permits to render their magnitudes close to 1. The three functions $f_1,\ f_2,\ f_3$ are uniformly positive and equal to 1 initially:

$$f_1^\star = f_2^\star = f_3^\star = 1. \tag{130}$$

Problematics, Pareto front and prioritized optimization. We decided to prioritize the cost functions f_1 and f_2 representing mass and range, and to consider f_3 representing approach speed as a secondary cost function, all three functions being subject to the constraint $c_1(\mathbf{x}) = 0$ related to take-off distance.

In a first experiment, not involving the slack-variable x_{16}, the mass-range Pareto front associated with the pair (f_1, f_2) subject to the equality constraint $c_1(\mathbf{x}) = 0$ handled by penalization was determined approximately by applying the *Pareto Archived Evolution Strategy* (PAES) [1]. This task revealed more laborious than we had anticipated due to the small scales affecting the design variables. Indeed, in the exploration conducted by PAES, the specification of small increments in the optimization variables leads to numerical stiffness and to a small portion of the front being investigated. Inversely, larger scales result in the elimination of an excessive proportion of the tested points by violation of the constraint. The trade-off was found delicate to adjust, and the convergence of the front was improved by successive steps conducted somewhat locally. After some 100,000 evaluations of the functions, eliminating all the points violating the constraint (125) and all the dominated points, resulted in the approximate Pareto front of Figure 17.

Five points A, B, C, D and E associated with mass-range Pareto-optimal solutions subject to the distance constraint were selected from the right-most point of the approximate discrete Pareto front (largest mass and range), to the left-most point (smallest mass and range). Each of these points was used as a starting point $\mathbf{x}_A^\star$ of a particular continuum of Nash equilibria conducted to diminish the approach speed. In this way, we obtained five different paths initiated from the front and represented on Figure 17 in different colors.

These paths were generated by setting the parameters of the numerical method according Table 5. These parameters were only very roughly optimized case by case. Nevertheless the table reflects some trends. The parameter γ was chosen to separate satisfactorily the plots of the different cost functions in the the horn-shaped pattern. It is clearly related to the local physical scales. The parameters 'hdiff' and 'hbox' are the local and global $\mathbf{x}$ scales used to build the metamodels.

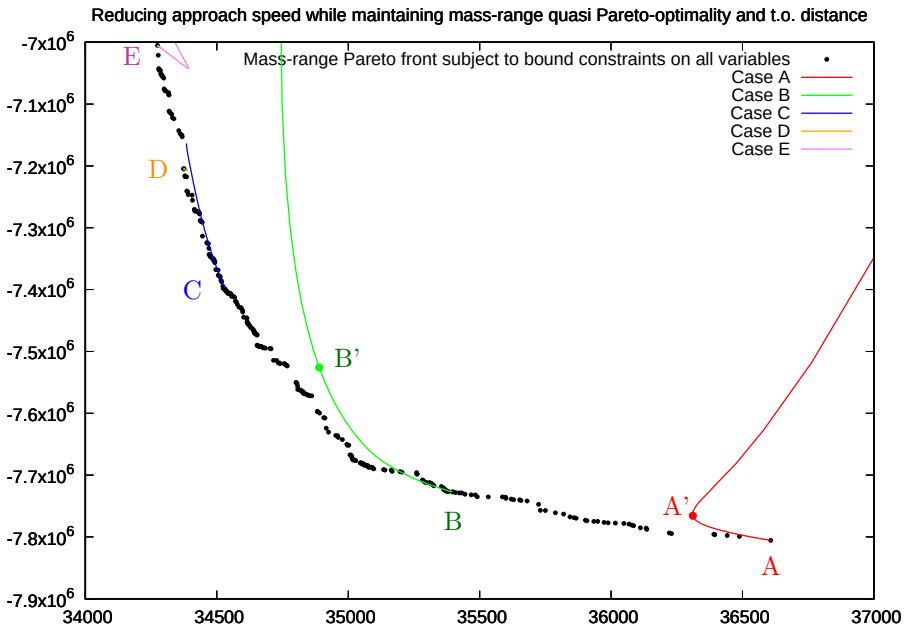


Figure 17: Discrete mass-range Pareto front and five continua of Nash equilibria (abscissa: mass in kg; ordinate: -range in m)

case	γ [a]	hdiff [b]	hbox [c]	κ [d]	$\ell_{\max}$ [e]	TOL [f]	$\lambda_{\max}$ [g]	$\mu_{\max}$ [h]
A	10	10^{-2}	0.5	10	100 ; 96	10^{-4}	15	5
B	10	10^{-2}	0.5	5	100 ; 94	10^{-4}	15	5
C	10	10^{-4}	0.5×10^{-2}	10	100 ; 99	10^{-4}	15	5
D	10^3	10^{-5}	0.5×10^{-3}	10	100 ; 99	10^{-4}	15	5
E	40	10^{-4}	0.5×10^{-2}	1.5	100 ; 99	10^{-4}	15	5

[a] constant to control magnitude of initial gradients
[b] stepsize in central differencing
[c] box size for global metamodels
[d] parameter controling the convexity fix
[e] number of discretization subintervals of $[0, \varepsilon_{\max}]$
[f] accuracy tolerance in $(\mathbf{u}, \mathbf{v})$ coordination outer loop
[g] maximum allowable number of $(\mathbf{u}, \mathbf{v})$ coordination iterations
[h] maximum number of iterations by Newton's method in each $\mathbf{u}$ subproblem

Table 5: Parameter setting for the five paths generated in the flight-mechanics test-case

The smaller scales are necessary in the small-mass design cases, to the left of the Pareto front. The parameter κ indirectly sizes the added convexity-fix term. Reducing this parameter increases convexity artificially. This was also found necessary for paths on the left side of the front where the physical scales are very small. In the column for $\ell_{\max}$ two figures are indicated: 100, the specified number of subintervals used to discretize the interval $[0, \varepsilon_{\max}]$, and the number of Nash equilibria actually determined, before the numerical process interrupts. The two numbers were found very close and this confirmed that when the optional numerical parameters are set adequately, the

process continues almost to the theoretical limit of convexity. The tolerance TOL is the accuracy tolerance in the coordination process between the $\mathbf{u}$ and $\mathbf{v}$ subvectors for given ε (outer loop). It also controls the accuracy tolerance on the inner iteration used to determine the subsector $\mathbf{u}$ for given subvector $\mathbf{v}$ by Newton's method which was set to $TOL/100$. This parameter was not found really critical. Hardly more important is the maximum allowable number of coordination iterations $\lambda_{\max}$ which was found to affect slightly the convergence at the tail of the continuum only. The parameter $\mu_{\max}$ bounds the number of Newton's method iterations used to solve for subvector $\mathbf{u}$ for given $\mathbf{v}$. The specified values for $\lambda_{\max}$ and $\mu_{\max}$ were largely in excess except at the tail of the continuum (see below Figure 22).

Let us first examine the most standard case corresponding to the green path initiated at B. In this experiment, the parameter γ was set to 10 to better visualize the variations of the functions. Evidently, it is confirmed that the path is tangent to the Pareto front and constitutes a satisfactory approximation of it over a significant segment. A value of κ somewhat smaller than in case A was found necessary to damp out a slight oscillation observed in the solution.

A great part of the path remains relatively close to the front. For example, the point B' at which $\varepsilon/\varepsilon_{\max} = 0.8$, where $\varepsilon_{\max}$ is the theoretical upper bound for convexity given by (206), is indicated by the symbol •. Thus, the portion of the path beyond this point only represents the 20% tail of the continuum.

The variation with ε of the three cost functions, and two steering functions f_A^+ and $\tilde{f}_B$ is given on Figure 18 on which the values at point B' are indicated by the symbol •.

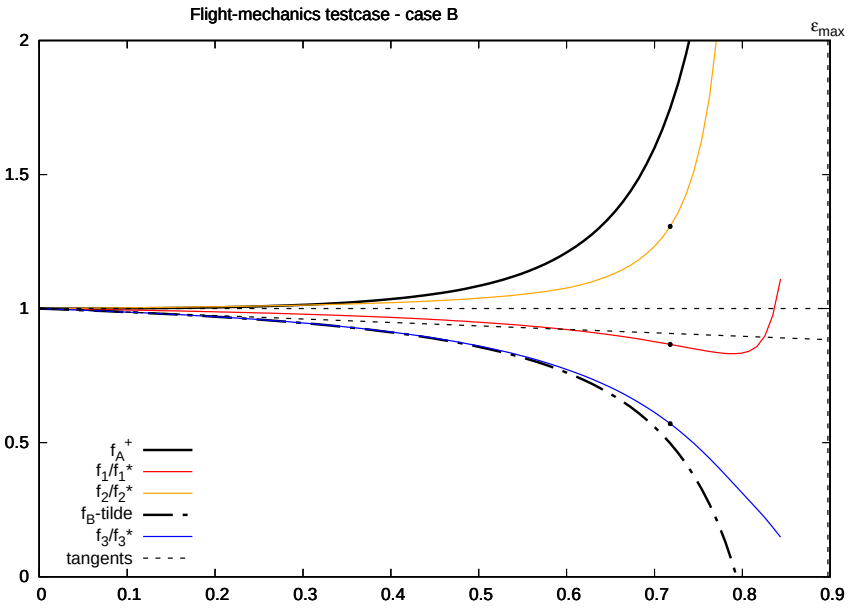


Figure 18: Variation with ε of the cost functions f_1, f_2, f_3 and the two steering functions f_A^+ and $\tilde{f}_B$ in Case B. The values at point B' are indicated by the symbol •.

Several observations can be made from this plot:

- The general horn-shaped pattern is again obtained.
- The primary steering function f_A^+ has indeed a null initial derivative. The two primary cost

functions vary inversely: f_1 diminishes, and f_2 increases. For example, at B':

$$f_1 \approx 0.866,\ f_2 \approx 1.307 \Longrightarrow M/M^\star \approx 0.986,\ R/R^\star \approx 0.974; \tag{131}$$

hence mass was reduced of 1.4%, a gain, and range diminished 2.6%, a loss. The function f_A^+ has a more rapid increase due to the convexity-fix term.

- The initial derivative of the secondary steering function $\tilde{f}_B$ is well given by the theoretical prediction indicated by a dashed tangent line. The function f_3, calculated a posteriori, follows accurately its metamodel $\tilde{f}_B$. At B':

$$f_3 \approx 0.571 \Longrightarrow V/V^\star \approx 0.944. \tag{132}$$

Hence the approach speed was reduced of some 5.6% This is confirmed by the plot of V as a function of mass along the continuum given by Figure 19, over which the value at B' is indicated by the symbol •.

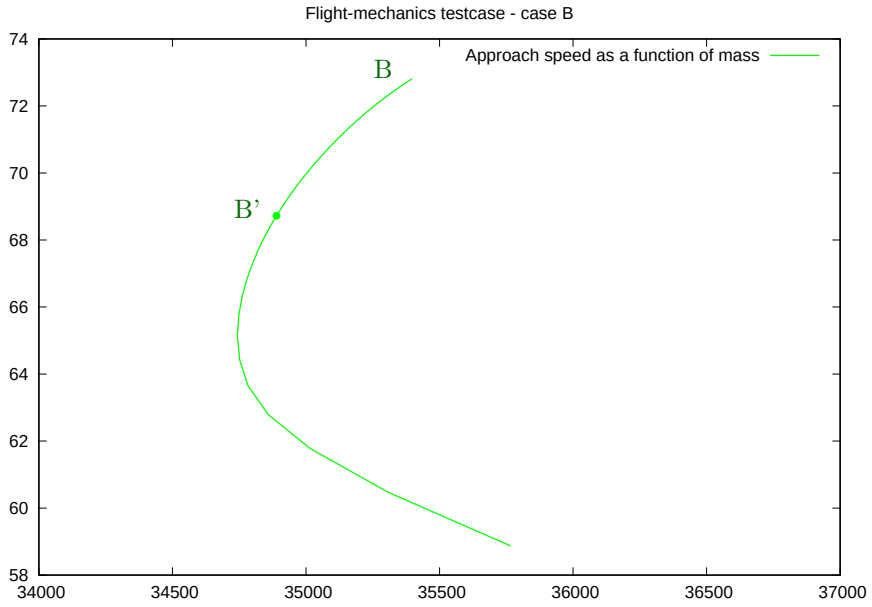


Figure 19: Approach speed as a function of mass along the continuum. The value at point B' is indicated by the symbol •.

Correspondingly, the variations with ε of the optimization variables $\{x_i\}$ $(i = 1, \ldots, 16)$ are given by Figure 20 over which the values at B' are indicated by the symbols •. Only the first four of these variables have a definite variation, the other ones remain nearly constant. Also note that the angular point in the plot of $x_3(\varepsilon)$ does not reflect a derivative discontinuity, but a convention adopted in the graphics representation. Since only $\sin x_i$ matters for the physical variable X_i, whenever x_i fell outside the range $[-\pi/2, \pi/2]$, x_i was replaced by symmetry by $sin^{-1}(sinx_i)$ in the graphics, but not in the numerical process or output datafile.

In Figure 21, we present the variation with ε of the constraint c_1 related to the take-off distance. Over a broad portion of the interval, the constraint is satisfied with very high accuracy thanks to

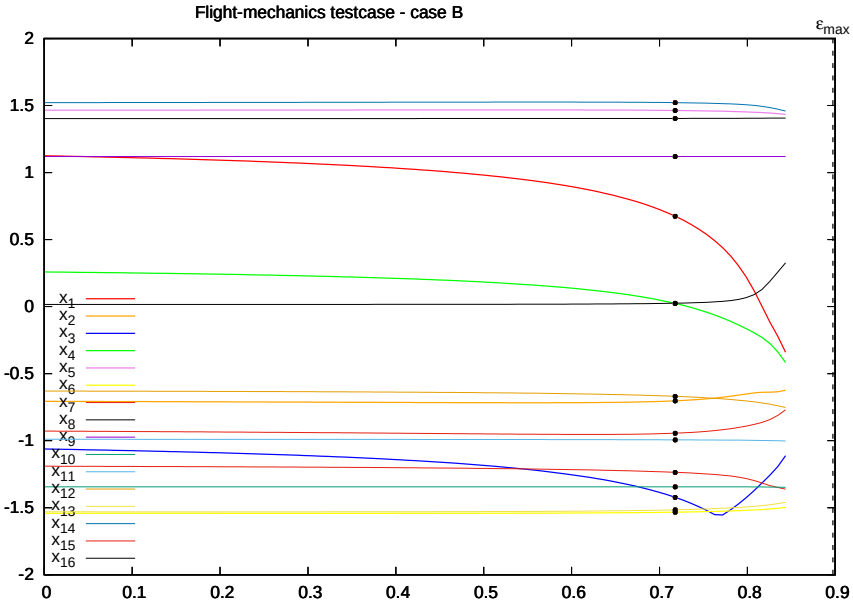


Figure 20: Variation with ε of the optimization variables $\{x_i\}$ $(i = 1, \ldots, 16)$. The values at B' are indicated by the symbols •.

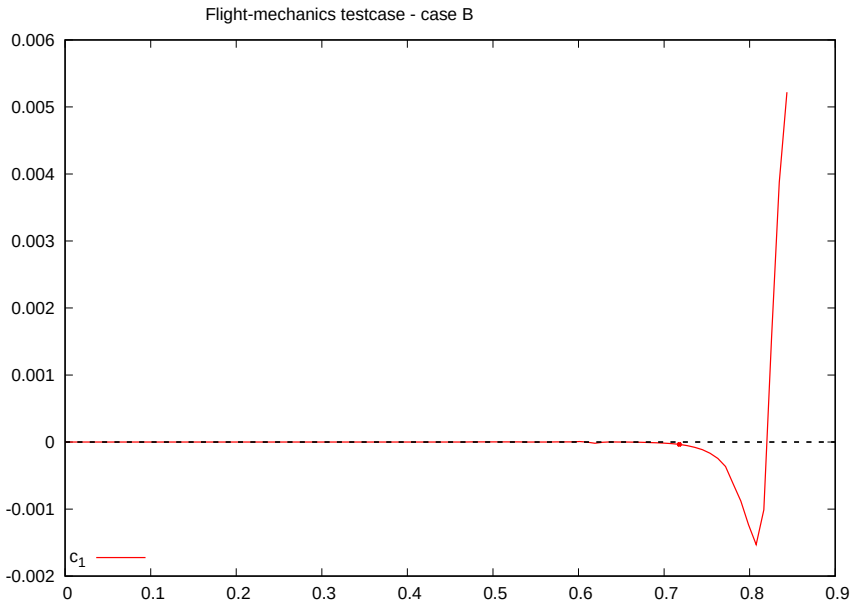


Figure 21: Variation with ε of the landing distance constraint along the continuum. The value at B' is indicated by the symbol •.

the local constraint metamodel, upgraded prior to the calculation of each new Nash equilibrium point. For example, at point B', the value is $c_1 \approx -3.7 \times 10^{-5}$, while $TOL = 10^{-4}$. Beyond this point, the constraint value ceases to be negligible rapidly. These values are firstly negative which implies that an unnecessarily severe strict inequality is satisfied, that is, some significant distance away from the Pareto front, the approach speed is overly reduced at the cost of an unduly mass increase and range reduction. Further away, when approaching $\varepsilon = \varepsilon_{\max}$, an instability is triggered, rendering the tail of the continuum meaningless. Perhaps a more sophisticated constraint metamodel could permit to prolongate the continuum further. This point is currently being examined.

We finish this description of Case B by some information relative to the number λ of outer iterations necessary to coordinate the two subvectors $\mathbf{u}$ and $\mathbf{v}$ at a given ε for the specified tolerance $TOL = 10^{-4}$, and the number μ of Newton's method inner iterations to solve for $\mathbf{u}$ for fixed $\mathbf{v}$ with accuracy tolerance of $TOL/100 = 10^{-6}$. Recall that over the hundred ε steps, 94 were successful to determine the Nash equilibrium. In all these cases, μ never exceeded 3, while the termination value of λ in given by Figure 22 in terms of ε. Over more than 60% of the continuum, λ remains at most equal to 2. Then when approaching the limit of convexity it increases, and finally the process diverges, or more rigorously speaking, is interrupted. These results reflect some arbitrariness related to the setting of numerical parameters; they would change with a different setting of the accuracy tolerances, or with the specification of a different maximum allowable number of iterations in the inner or outer loop.

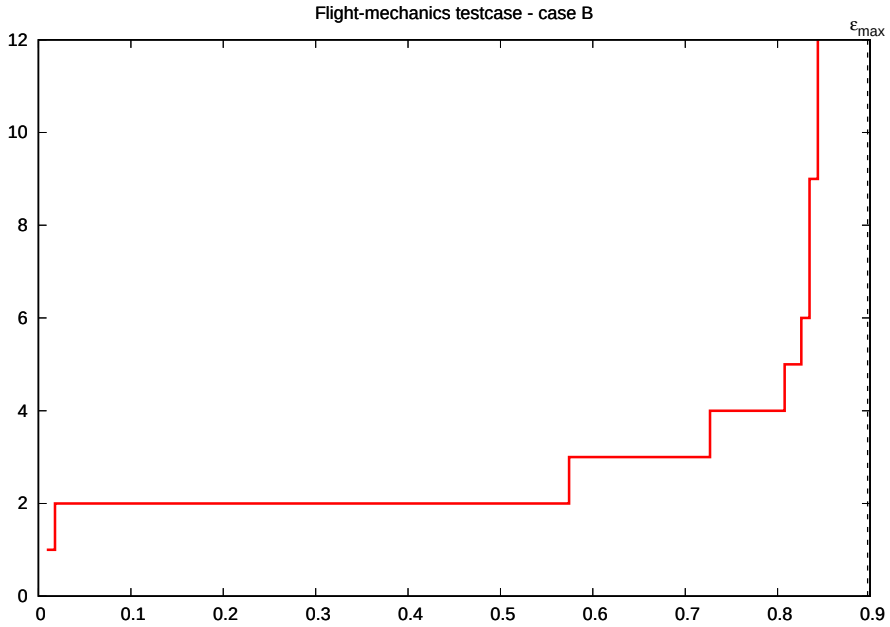


Figure 22: Number λ of outer iterations necessary to coordinate the subvectors $\mathbf{u}$ and $\mathbf{v}$ at a given ε as a function of ε. (The process was interrupted whenever the convergence test still was not met after the 15th iteration.)

We now discuss the results related to the continuum originating from point A, at the extreme right of Figure 17 corresponding to the case of largest mass and range. These results are similar to those of Case B except that the portion of the continuum close to the Pareto front extends

somewhat less. Again, a lighter aircraft could permit a smaller approach speed, but at the cost of a smaller range. For example, we have again spotted the point A' at which $\varepsilon/\varepsilon_{\max} = 0.8$. By examining the discrete Pareto front in the area of point A', we suspect the existence of zones of multi-modality, and this could explain the abrupt change in the direction of the continuum beyond point A'.

Thirdly, the continuum initiated at point C remains very close to the discrete Pareto front, and is here an excellent approximation of it over the entire segment.

Fourthly, the continuum initiated at point D is very small and obtained with difficulty using the very large value of 1000 for the parameter γ. In this area, there seems to occur an inversion of trend, more visible from the continuum initiated at point E, using $\gamma = 40$, and a very low upper bound on the condition number ($\kappa = 1.5$) in which results a large convexity-fix constant. The continuum there makes a loop, indicating a form of instability.

Evidently, the results in Cases D and E do not bring improved practical designs.

Lastly, recall that each continuum is supposed to be initiated from a point on the Pareto front. In order to appreciate how this condition was met for points A through E, we defined for each point the following scaled measure $\|\boldsymbol{\omega}_A^\star\| \,/\, \max_{j\leq m} \left\|\mathbf{g}_j^\star\right\|$. For points exactly on the Pareto front, this dimensionless measure is equal to 0. For the points A through E, we found 0.137, 0.391, 0.210, 0.130 and 0.156 respectively. Surprisingly, these numbers are not so small, and this indicates a relatively poor degree of convergence to the front, or some inconsistency between local physical gradients and numerical approximations by finite differences. We observe, however, that this does not affect the numerical procedure severely.

In conclusion of this case study, we emphasize the following points:

- the test-case is of actual technical interest;
- it involves 15 design variables subject to interval constraints that were handled by a change of variables, and one functional constraint handled by the introduction of a slack variable; many of these variables however remained almost constant along the continuum;
- the scales in the design variables were found very small and made the establishment of the Pareto front by the evolutionary strategy laborious.

Nevertheless, we were able to identify a continuum of Nash equilibria originating from a large portion of the Pareto front. This continuum complies with the theoretical findings, in particular with respect to tangency with the Pareto front. Along the continuum, the cost functions exhibit a horn-shaped pattern bounded on top by the primary steering function that increases by $O(\varepsilon^2)$ with the continuation parameter ε and below by the secondary steering function that decreases, initially linearly, and in compliance with the theoretical results. The constraint is very well satisfied over a large part of the interval $[0, \varepsilon_{\max}]$, where $\varepsilon_{\max}$ is the known theoretical limit of convexity of the metamodel-based problem. Some extreme cases required an unusual adjustment of the parameter κ (which controls the added convexity term), and γ which controls the magnitudes of initial gradients.

In these experiments, we have determined a priori an approximate but entire Pareto front by PAES. This was for purpose of numerical illustration and verification of the theoretical relationship between the front and the continuum. However, only a few Pareto-optimal points were actually necessary to initiate the continuums. In practice, these initial points could have been obtained by other means, more efficiently, and the Pareto front roughly sketched by the continuums themselves alone, and this would have been an alternative to the evolutionary strategy.

Lastly, to establish one such continuum typically required about 45 sec of real time on a standard laptop for program assembly and compilation, execution (mostly in the 15 phases of preparation of the Nash games), and the production of plots using the gnuplot software. In this test-case that involved 16 variables, the database contained 1920 points about each $\mathbf{x}_A^\star$. Consequently, most of the computational work was devoted to the establishment of the metamodels (Steps 3 and 13) and the evaluation of the primary agglomerated function f_A (Step 7), and relatively less to the calculation of the Nash equilibria per se. For example, in Case B, 94 equilibrium

points were calculated. The numerical experiment required a total of 2048 calls to the evaluation procedures of the primary and secondary cost functions, and 5055 calls to the procedure of constraint evaluation.

7 Conclusions

This report has provided the detailed technical description of the numerical procedures being incorporated in the chapter on prioritized multi-objective optimization of the Inria software platform `http:mgda.inria.fr`. The potential utilizer of our technique will find the instructions for running our code on the platform.

Some proofs were included for the sake of clarity, but the full description of the mathematical setting is being prepared for submission to a journal.

Acknowledgements

The author wishes to express his warmest thanks to his colleagues from Inria and the University Côte d'Azur, R. Duvigneau, A. Habbal, L. Hascoët, L. Monasse for very fruitful scientific discussions on Nash games and polytope exploration in dimension n, as well as software implementation of algorithms.

We are also indebted to M. Ravachol from Dassault Aviation for very useful discussions on the role of sensitivity evaluation in aircraft design. He also provided the flight-mechanics software.

Finally, we thank the Inria Service of Experimentation and Development (SED) for the development of the web interface of the Nash-MGDA Platform, and T. Kloczko and J. Wintz in particular.

References

[1] J. Knowles, D. Corne, *The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation*, Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, January 1999, doi = 10.1109/CEC.1999.781913.

[2] R. Filomeno Coelho, P. Breitkopf (ed.), *Optimisation multidisciplinaire en mécanique 1 - démarche de conception, stratégies collaboratives et concourantes, multiniveaux de modèles et de paramètres*, Mécanique et Ingénierie des Matériaux, Hermes Lavoisier, Paris, 2009, `www.hermes-science.com` `www.lavoisier.fr`.

[3] R. Filomeno Coelho, P. Breitkopf (ed.), *Optimisation multidisciplinaire en mécanique 2 - réduction de modèles, robustesse, fiabilité, réalisations logicielles*, Mécanique et Ingénierie des Matériaux, Hermes Lavoisier, Paris, 2009, `www.hermes-science.com` `www.lavoisier.fr`.

[4] P. Breitkopf, R. Filomeno Coelho (ed.), *Multidisciplinary design optimization in computational mechanics*, Wiley-ISTE, May 2010, `https://www.wiley.com/en-fr/Multidisciplinary+Design+Optimization+in+Computational+Mechanics-p-9781848211384`.

[5] J.-A. Désidéri, *Split of Territories in Concurrent Optimization*, Research Report 6108, INRIA, October 2007, `https://hal.inria.fr/inria-00127194`.

[6] ———, *Multiple-gradient descent algorithm (MGDA) for multiobjective optimization*, Comptes Rendus de l'Académie des Sciences Paris **350** (2012), 313–318.

[7] ———, *Révision de l'algorithme de descente à gradients multiples (MGDA) par orthogonalisation hiérarchique*, Research Report 8710, INRIA, April 2015, `https://hal.inria.fr/hal-01139994`.

[8] ———, *Quasi-Riemannian Multiple Gradient Descent Algorithm for Constrained Multiobjective Differential Optimization*, Research Report 9159, INRIA, March 2018, `https://hal.inria.fr/hal-01740075`.

[9] J.-A. Désidéri, R. Duvigneau, and A. Habbal, *Computational Intelligence in Aerospace Sciences, V. M. Becerra and M. Vassile eds.*, Progress in Astronautics and Aeronautics, T. C. Lieuwen Ed.-in-Chief, vol. 244, ch. Multi-Objective Design Optimization Using Nash Games, American Institute for Aeronautics and Astronautics Inc., Reston, Virginia, 2014.

[10] C. M. Fonseca and P. J. Fleming, *An Overview of Evolutionary Algorithms in Multiobjective Optimization*, Evolutionary Computation **3** (1995), no. 1, 1–16, DOI:10.1162/evco.1995.3.1.1.

[11] L. J. Gibson and M. F. Ashby, *Cellular solids : Structure and properties*, Cambridge University Press, 1999.

[12] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*, Academic Press, London, 1986.

[13] J.-A. Désidéri et al, *MGDA Platform*, March 2017, `http://mgda.inria.fr`.

[14] Q. Mercier, F. Poirion, and J.-A. Désidéri, *Nonconvex mutiobjective optimization under uncertainty: a descent algorithm. Application to sandwich plate design and reliability*, Engineering Optimization (to appear).

[15] M. Ravachol, *Multidisciplinary design optimization in computational mechanics*, ch. Multilevel Multidisciplinary Optimization in Airplane Design, Wiley-ISTE, May 2010, `https://www.wiley.com/en-fr/Multidisciplinary+Design+Optimization+in+Computational+Mechanics-p-9781848211384`.

[16] Wikipedia The Free Encyclopedia, *Test functions for optimization*, `https://en.wikipedia.org/wiki/Test_functions_for_optimization`.

A Geometrical database

The optimization variable $\mathbf{x} \in \mathsf{R}^n$ is viewed as a point with geometrical coordinates. The metamodels are constructed to approximate functions in the neighborhood of the optimality point $\mathbf{x}_A^\star$. The metamodels are defined in a hypercube $[-h, h]^n$ centered at $\mathbf{x}_A^\star$. An adequate value of the parameter h is to be provided by the user.

In all what follows, the superscript * indicates an evaluation at $\mathbf{x} = \mathbf{x}_A^\star$.

It is assumed that the cost functions $\{f_j(\mathbf{x})\}$ ($j = 1, \ldots, m$ for the prime functions, and $j = m+1, \ldots, M$ for the secondary functions) as well as the constraint functions $\{c_k(\mathbf{x})\}$ ($k = 1, \ldots, K$) are defined by procedures of evaluation that do not include gradient information. Second-order accurate approximations of the gradients, $\{\nabla f_j^\star\}$ and $\{\nabla c_k^\star\}$, are firstly calculated by central differencing using a step-size of $h/10$. To complete this local information, the metamodels are meant to approximate the functions more globally in the hypercube, in the sense of least squares of deviations of the metamodel from the exact function values over a discrete set of datapoints $\{\mathbf{x}_\nu\}$ ($\nu = 1, \ldots, \nu_{\max}$) spread in the hypercube. Thus the datapoints should be defined to permit a good global approximation of second derivatives. To this aim, we proceed as follows: for each pair of variables (x_i, x_j) ($i = 1, \ldots, n-1$; $j = i+1, \ldots, n$), we consider a stencil in the corresponding hyperplane, made of 8 points on the square of side $2h$ and 8 points on the square of side $h\sqrt{2}$ as illustrated on Figure 23. In this way, one expects enough information is present in the geometrical database to correctly approximate all types of second derivatives, $\partial^2/\partial x_i^2$, $\partial^2/\partial x_j^2$ and $\partial^2/\partial x_i/\partial x_j$. Thus, the ratio between the sides of the inner and outer squares is equal to $1/\sqrt{2}$. This is justified by the following one-dimensional analysis.

As a model representative of the present situation, one considers the "best approximation" of a function $f(x)$ by interpolation of function values $\{y_{-1}, y_{-\theta}, y_0, y_\theta, y_1\}$ at $\{-1, -\theta, 0, \theta, 1\}$ and of the first derivative y_0' at $x = 0$. The Hermite-type interpolant is expressed as follows:

$$h(x) = y_0 + xy_0' + x^2 g(x) \tag{133}$$

where $g(x)$ should satisfy the following conditions:

$$\begin{cases} y_{-1} &= y_0 - y_0' + g(-1) \\ y_{-\theta} &= y_0 - \theta y_0' + \theta^2 g(-\theta) \\ y_\theta &= y_0 + \theta y_0' + \theta^2 g(\theta) \\ y_1 &= y_0 + y_0' + g(1) \end{cases} \tag{134}$$

Clearly, the admissible polynomial $g(x)$ of lowest degree is unique; it is the Lagrange interpolation polynomial corresponding to these conditions. The interpolation error is given by the following

Proposition 2
For all $x \in (-1, 1)$, there exists $\xi \in (-1, 1)$ such that

$$f(x) = h(x) + (x+1)(x+\theta)x^2(x-\theta)(x-1)\frac{f^{(6)}(\xi)}{6!}. \tag{135}$$

Proof: Let $x_0 \in (-1, 1)$ be given.
Either $x_0 = -1,\ -\theta,\ 0,\ \theta$ or 1, and the statement is trivially verified by $x = x_0$. Or x_0 is distinct from all the interpolation points. Then let

$$\lambda = \frac{f(x_0) - h(x_0)}{(x_0^2 - 1)(x_0^2 - \theta^2)x_0^2} \tag{136}$$

and

$$\phi(x) = f(x) - h(x) - \lambda(x^2 - 1)(x^2 - \theta^2)x^2. \tag{137}$$

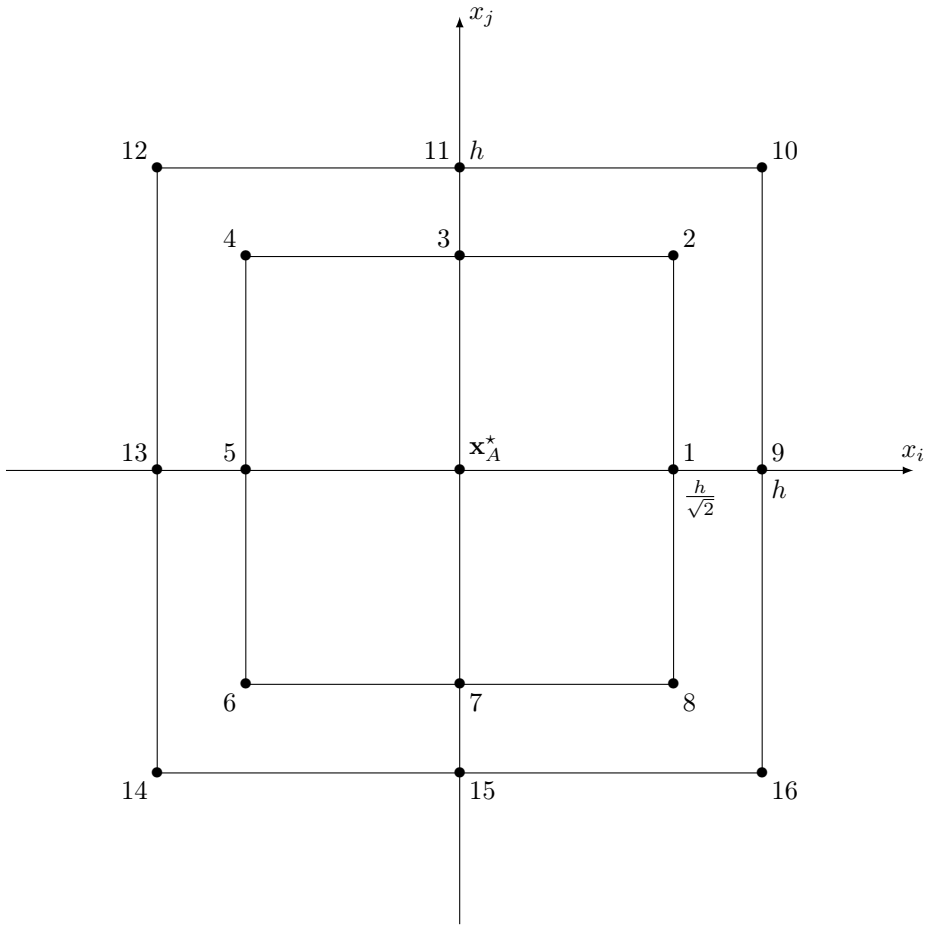


Figure 23: Stencil of the 16 datapoints in the hyperplane (x_i, x_j) about $\mathbf{x}_A^\star$; the side of the outer square is $2h$, and the side of the inner square is $h\sqrt{2}$.

Then $\phi(x)$ admits (at least) 6 distinct zeroes: $\pm 1,\ \pm\theta,\ 0$ and x_0. Additionally $\phi'(0) = 0$. Suppose for example that x_0 falls between 0 and θ. Then, by virtue of Rolle's theorem, one gets a zero of $\phi'(x)$ in each of the disjoint open intervals: $]-1, -\theta[,\ ]-\theta, 0[,\ ]0, x_0[,\ ]x_0, \theta[,\ ]\theta, 1[$. None of these 5 zeroes is located at $x = 0$; hence 0 is a 6th distinct zero of $\phi'(x)$. Therefore, $\phi''(x)$ admits (at least) 5 distinct zeroes, and so on. We conclude to the existence of a zero ξ of $\phi^{(6)}(x)$, and since $h^{(6)}(\xi) = 0$ (since $h(x)$ is of degree 5 at most), this gives

$$\lambda = \frac{f^{(6)}(\xi)}{6!}. \tag{138}$$

□

Hence, for the interpolation problem, the optimal value for θ is the one that minimizes the maximum value of the interpolation error:

$$\theta^\star = \arg\min_\theta \max_{x\in(-1,1)} |e(x)|\,, \quad e(x) = (x^2-1)(x^2-\theta^2)x^2. \tag{139}$$

The interpolation error is an even function of x, monotone increasing over the interval $(0, \theta)$ over which it reaches a positive maximum $e_{\max}$ at x_1, and monotone decreasing over the interval $(\theta, 1)$ over which it reaches a negative minimum $e_{\min}$ at x_2, where x_1 and x_2 are the zeroes of the derivative

$$e'(x) = 2x.(x^2-\theta^2).x^2 + (x^2-1).2x.x^2 + (x^2-1).(x^2-\theta^2).2x. \tag{140}$$

Hence x_1 and x_2 are the roots of the following equation:

$$(x^2 - \theta^2)x^2 + (x^2 - 1)x^2 + (x^2 - 1).(x^2 - \theta^2) = 0 \tag{141}$$

that is:

$$3x^4 - 2(\theta^2 + 1)x^2 + \theta^2 = 0 \tag{142}$$

and this gives:

$$x_1,\ x_2 = \frac{\theta^2 + 1 \pm \sqrt{1 - \theta^2(1 - \theta^2)}}{3} \tag{143}$$

The optimal $\theta^\star$ corresponds to the satisfaction of the condition:

$$e_{\min} = -e_{\max} \tag{144}$$

and one can verify that this is achieved when $\theta^2 = \frac{1}{2}$, that is:

$$\theta^\star = \frac{1}{\sqrt{2}}. \tag{145}$$

In conclusion, the geometrical database is constructed by considering 16 datapoints in each of the $n(n-1)/2$ hyperplanes (x_i, x_j). This gives a total of $8n(n-1)$ datapoints $\{\mathbf{x}_\nu\}$ ($\nu = 1, \ldots, \nu_{\max} = 8n(n-1)$).

B Metamodels

B.1 Quadratic metamodels for the nonlinear constraints

Metamodels at the starting point $\mathbf{x}_A^\star$. For the constraint $c_k(\mathbf{x})$ $(k = 1, \ldots, K)$, recall that $c_k(\mathbf{x}_A^\star) = 0$, and consider the following quadratic metamodel:

$$\tilde{c}_k(\mathbf{x}) = \big(\nabla c_k^\star, \delta\mathbf{x}\big) + \tfrac{1}{2}\big(\delta\mathbf{x}, \mathbf{H}_{c_k}^\star \delta\mathbf{x}\big) \tag{146}$$

where $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_A^\star$. The gradient $\nabla c_k^\star$ as well as the diagonal of the Hessian $\mathbf{H}_{c_k}^\star$ have been determined by central differencing. Thus here, one adjusts the off-diagonal, say the strictly-upper-diagonal elements of the symmetric matrix $\mathbf{H}_{c_k}^\star = \{h_{c_{i,j,k}}\}$ to best fit the values of the constraint c_k over the geometrical database in the sense of least squares:

$$\min_{\{h_{c_{i,j,k}}\}} \sum_\nu \tfrac{1}{2}\left[\big(\nabla c_k^\star, \delta\mathbf{x}_\nu\big) + \tfrac{1}{2}\big(\delta\mathbf{x}_\nu, \mathbf{H}_{c_k}^\star \delta\mathbf{x}_\nu\big) - c_k(\mathbf{x}_\nu)\right]^2 \tag{147}$$

where $2 \le j \le n$, $1 \le i \le j-1$. Thus we have a total of $n(n-1)/2$ unknown elements $h_{c_{i,j,k}}$ to optimize for each k.

The optimality w.r.t. to the element $h_{c_{i,j,k}}$ writes:

$$\sum_\nu \left[\big(\nabla c_k^\star, \delta\mathbf{x}_\nu\big) + \tfrac{1}{2}\big(\delta\mathbf{x}_\nu, \mathbf{H}_{c_k}^\star \delta\mathbf{x}_\nu\big) - c_k(\mathbf{x}_\nu)\right](\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j = 0. \tag{148}$$

The construction is made using a geometrical database that is symmetrical w.r.t. the central point $\mathbf{x}_A^\star$. Thus, by symmetry, in the above sum, the first-order term in $\delta\mathbf{x}_\nu$ disappears, and the equation simplifies to:

$$\sum_\nu \left[\tfrac{1}{2}\big(\delta\mathbf{x}_\nu, \mathbf{H}_{c_k}^\star \delta\mathbf{x}_\nu\big)\right](\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j = \sum_\nu c_k(\mathbf{x}_\nu)(\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j. \tag{149}$$

The above equation is indexed (i, j). By symmetry, the element $h_{c_{l,m,k}}$ affects two equal terms and this results in the coefficient $(\delta\mathbf{x}_\nu)_l(\delta\mathbf{x}_\nu)_m(\delta\mathbf{x}_\nu)_j i(\delta\mathbf{x}_\nu)_j$ since $l \neq m$ $(2 \le m \le n,\ 1 \le l \le m-1)$. The linear system is thus given by:

$$\forall (i,j) \text{ such that } 2 \le j \le n \text{ and } 1 \le i \le j-1 :$$
$$\sum_{m=2}^{n} \sum_{l=1}^{m-1} h_{l,m,k} \sum_\nu (\delta\mathbf{x}_\nu)_l(\delta\mathbf{x}_\nu)_m(\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j = \sum_\nu c_k(\mathbf{x}_\nu)(\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j. \tag{150}$$

The matrix element

$$A_{i,j,l,m} = \sum_\nu (\delta\mathbf{x}_\nu)_l(\delta\mathbf{x}_\nu)_m(\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j \tag{151}$$

is the coefficient of $h_{l,m,k}$ in the equation indexed (i, j). Let $(l.m)$ be fixed. Then for a given i there are two values of l associated with opposite values of $(\delta\mathbf{x}_\nu)_l$. Hence the above element is equal to 0 unless $l = i$, and for the same reason unless $m = j$. Consequently the matrix A is diagonal and

$$A_{i,j,i,j} = \sum_\nu (\delta\mathbf{x}_\nu)_i^2 (\delta\mathbf{x}_\nu)_j^2. \tag{152}$$

Furthermore, this element is a constant since the geometrical configuration of the database is the same for all pairs (i, j):

$$A_{i,j,i,j} = A_h = 5(hbox)^4 \tag{153}$$

(see Figure 23), and finally for off-diagonal elements $(i \neq j)$:

$$h_{i,j,k} = \frac{1}{A_h} \sum_\nu c_k(\mathbf{x}_\nu)(\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j. \tag{154}$$

Updated metamodels at subsequent points. The metamodels $\{\tilde{c}_k(\mathbf{x})\}$ for constraint $\{c_k(\mathbf{x})\}$ are updated after each determination of a Nash equilibrium point. We propose to let the constraint quadratic metamodels evolve along the continuum to avoid the accumulation of modelling errors, and for this purpose to account for updated values of:

- the constraint itself by direct evaluation, since we have just enforced that $\tilde{c}_k(\overline{\mathbf{x}}) = 0$, but $\bar{c}_k = c_k(\overline{\mathbf{x}}) \neq 0$ in general,
- the constraint gradient by new local central differencing about $\overline{\mathbf{x}}$, $\nabla\bar{c}_k = \nabla c_k(\overline{\mathbf{x}})$,
- the diagonal of the constraint Hessian, $\bar{\mathbf{H}}_{c_k}$, by new local central differencing about $\overline{\mathbf{x}}$.

The off-diagonal elements of the Hessian, usually less important, are not refreshed and numerical experiments tend to confirm that this is sufficient.

These updated metamodels are described in details in Subsection C.1.

B.2 Quadratic metamodel for the prime steering function

A local gradient, $\nabla f_A^\star$, has been calculated by central differencing. A Hessian matrix, $\widehat{\mathbf{H}}_A^\star$ has been calculated by combining local central differencing for the diagonal elements, and global least-square approximation for the off-diagonal elements. The corresponding quadratic metamodel is then slightly modified by computing the Lagrange multipliers $\lambda_k^\star$ $(k = 1, \ldots, K)$ for which the gradient is best approximated by the linear combination $-\sum_{k=1}^K \lambda_k^\star \nabla c_k^\star$. Hence the problem for the vector $\lambda^\star$ is:

$$\min_{\lambda^\star} \tfrac{1}{2} \left\| \sum_{k=1}^K \lambda_k^\star \nabla c_k^\star + \nabla f_A^\star \right\|^2 \tag{155}$$

in which the vectors $\nabla f_A^\star$ and $\{\nabla c_k^\star\}$ $(\forall k)$ are known. The optimality condition results in the following $K \times K$ symmetric, positive-definite linear system for the Lagrange multipliers:

$$\forall j: \quad \sum_k \big(\nabla c_j^\star, \nabla c_k^\star\big)\lambda_k^\star = -\big(\nabla c_j^\star, \nabla f_A^\star\big). \tag{156}$$

The system is assembled, factorized by the Lapack procedure DPOTRF, and solved by DPOTRS. One then lets:

$$\nabla \tilde{f}_A^\star = -\sum_{k=1}^K \lambda_k^\star \nabla c_k^\star. \tag{157}$$

and the following quadratic global metamodel for f_A is expressed:

$$\tilde{f}_A(\mathbf{x}) = 1 + \big(\mathbf{x} - \mathbf{x}_A^\star, \nabla \tilde{f}_A^\star + \tfrac{1}{2}\widetilde{\mathbf{H}}_A^\star(\mathbf{x} - \mathbf{x}_A^\star)\big). \tag{158}$$

Note that the construction is such that the function $\tilde{f}_A(\mathbf{x})$ satisfies exactly the stationarity condition at $\mathbf{x} = \mathbf{x}_A^\star$:

$$\nabla \tilde{f}_A^\star + \sum_{k=1}^K \lambda_k^\star \nabla c_k^\star = 0. \tag{159}$$

Lastly, the Hessian matrix $\widetilde{\mathbf{H}}_A^\star$ is augmented of the scalar matrix $c\mathbf{I}_n$ to insure convexity[4]:

$$\mathbf{H}_A^{+,\star} = \widetilde{\mathbf{H}}_A^\star + c\mathbf{I}_n \tag{160}$$

and this provides the primary steering function global quadratic metamodel:

$$f_A^+(\mathbf{x}) = 1 + \big(\mathbf{x} - \mathbf{x}_A^\star, \nabla \tilde{f}_A^\star + \tfrac{1}{2}\mathbf{H}_A^{+,\star}(\mathbf{x} - \mathbf{x}_A^\star)\big). \tag{161}$$

In conclusion, the function $f_A^+(\mathbf{x})$ is a convex quadratic that satisfies the stationarity conditions at $\mathbf{x} = -\mathbf{x}_A^\star$ where it admits a global minimum.

[4]See paragraph on convexity fix in Section 2.

B.3 Quadratic metamodel for the secondary steering function

The secondary steering function $f_B(\mathbf{x})$ is expressed as follows:

$$f_B(\mathbf{x}) = \sum_{j=m+1}^{M} \alpha_j^\star \frac{f_j(\mathbf{x})}{f_j^\star} \tag{162}$$

once the coefficients $\{\alpha_j^\star\}$ $(j \geq m+1)$ are determined by the MGDA process. In particular $f_B^\star = 1$. An approximate gradient at $\mathbf{x}_A^\star$, $\nabla \tilde{f}_B^\star$, and approximate second partial derivatives, $\{(\partial^2 f_B/\partial x_i^2)^\star\}$ $(i = 1, \ldots, n)$ are calculated by central differencing. The function is also evaluated over the geometrical database to determine the off-diagonal elements of the hessian:

$$\left(\tilde{\mathbf{H}}_B^\star\right)_{i,j} = \frac{1}{A_h} \sum_{\nu} f_B(\mathbf{x}_\nu)(\delta\mathbf{x}_\nu)_i(\delta\mathbf{x}_\nu)_j \quad (\forall i, \forall j \neq i). \tag{163}$$

Once these elements have been computed, the metamodel for $f_B(\mathbf{x})$ is expressed as follows:

$$\tilde{f}_B(\mathbf{x}) = 1 + \left(\mathbf{x} - \mathbf{x}_A^\star, \nabla \tilde{f}_B^\star + \tfrac{1}{2}\tilde{\mathbf{H}}_B^\star(\mathbf{x} - \mathbf{x}_A^\star)\right). \tag{164}$$

C Optimization of u and v in the Nash game

C.1 Constrained optimization of $\mathbf{u} \in \mathsf{R}^{n-p}$ for fixed $\mathbf{v} = \mathbf{v}_0 \in \mathsf{R}^p$

Recall that in the metamodel-assisted algorithm, the subvector $\mathbf{u} \in \mathsf{R}^{n-p}$ is optimized for fixed $\mathbf{v} = \mathbf{v}_0 \in \mathsf{R}^p$ to minimize the cost function f_A^+ under the constraint $\tilde{\mathbf{c}} = \{\tilde{c}_k\} = 0$ $(1 \le k \le K)$, where all these metamodels are constructed to be quadratic.

Global metamodel for $f_A^+(\mathbf{x})$. The primary agglomerated cost function f_A is metamodeled by $\tilde{f}_A$ and augmented by the convexity-fix term to yield the primary steering function $f_A^+(\mathbf{x})$ according to (161).

Since the optimization of subvector $\mathbf{u}$ is conducted at fixed subvector $\mathbf{v} = \mathbf{v}_0$:

$$\mathbf{x} - \mathbf{x}_A^\star = \mathbf{\Omega}_\mathbf{u}\mathbf{u} + \mathbf{\Omega}_\mathbf{v}\mathbf{v}_0 \tag{165}$$

where the matrices $\mathbf{\Omega}_\mathbf{u}$ and $\mathbf{\Omega}_\mathbf{v}$ are fixed. Hence:

$$\begin{aligned} f_A^+ &= 1 + \left(\mathbf{\Omega}_\mathbf{u}\mathbf{u} + \mathbf{\Omega}_\mathbf{v}\mathbf{v}_0, \nabla\tilde{f}_A^\star + \tfrac{1}{2}\mathbf{H}_A^{+,\star}(\mathbf{\Omega}_\mathbf{u}\mathbf{u} + \mathbf{\Omega}_\mathbf{v}\mathbf{v}_0)\right) \\ &= \underbrace{1 + \left(\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0, \nabla\tilde{f}_A^\star + \tfrac{1}{2}\mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0\right)}_{\text{(independent of } \mathbf{u})} \\ &\quad + \underbrace{\left(\mathbf{\Omega}_\mathbf{u}\mathbf{u}, \nabla\tilde{f}_A^\star + \tfrac{1}{2}\mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0\right) + \left(\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0, \tfrac{1}{2}\mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{u}\mathbf{u}\right)}_{\text{(linear in } \mathbf{u})} \\ &\quad + \underbrace{\left(\mathbf{\Omega}_\mathbf{u}\mathbf{u}, \tfrac{1}{2}\mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{u}\mathbf{u}\right)}_{\text{(quadratic in } \mathbf{u})}. \end{aligned} \tag{166}$$

Thus let:

$$\begin{cases} \phi_0 = 1 + \left(\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0, \nabla\tilde{f}_A^\star + \tfrac{1}{2}\mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0\right) & (1) \\ f_0 = \mathbf{\Omega}_\mathbf{u}^t\left(\nabla\tilde{f}_A^\star + \mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{v}\mathbf{v}_0\right) & (n_1 \times 1) \\ \mathbf{Q}_\mathbf{u} = \mathbf{\Omega}_\mathbf{u}^t\mathbf{H}_A^{+,\star}\mathbf{\Omega}_\mathbf{u} & (n_1 \times n_1) \end{cases} \tag{167}$$

where the different dimensions of these elements are indicated to the right, between parentheses, and the notation $n_1 = n - p$ is used. In this way:

$$f_A^+(\mathbf{u}) = \phi_0 + f_0^t\mathbf{u} + \tfrac{1}{2}\mathbf{u}^t\mathbf{Q}_\mathbf{u}\mathbf{u}. \tag{168}$$

From the computational viewpoint, note that:

- the matrix $\mathbf{Q}_\mathbf{u}$ is fixed and calculated once for all;
- the vector f_0 depends on $\mathbf{v}_0$ and is updated at each coordination iteration;
- the vector $\phi_0 = \phi(\mathbf{v}_0)$ depends on $\mathbf{v}_0$, but plays no role in the optimization of $\mathbf{u}$ at fixed $\mathbf{v}_0$; hence in practice, it is not computed.

Local constraint metamodels $\{\tilde{c}_k(\mathbf{x})\}$. In the determination of $\mathbf{u}_1$ by the solution of the optimality conditions by a procedure described in the next paragraph, the constraints are accounted for by means of metamodels. Hence, the actual constraints are only enforced approximately. If global constraint metamodels were used, the systematic modeling error would tend to accumulate at the risk of a severe violation of the admissible set, thus "changing the rules". To avoid this, we propose to correct the constraint metamodels after each determination of a Nash equilibrium point.

Let ℓ be the index of the discrete Nash equillibria $\{\overline{\mathbf{x}}_\ell\}$ as the determination of the continuum proceeds by incrementation of the continuation parameter $\varepsilon = \varepsilon_\ell$. Let us consider the situation where the Nash equilibrium $\overline{\mathbf{x}}_{\ell-1}$, shortly denoted $\overline{\mathbf{x}}$ hereafter, has just been determined on the

basis of the constraint metamodels $\{\tilde{c}_k^{(\ell-1)}(\mathbf{x})\}$ that differ slightly from the actual constraints $\{c_k(\mathbf{x})\}$, defined by the user's specification. To the accuracy at which the optimality conditions have been satisfied:

$$\tilde{c}_k^{(\ell-1)}(\overline{\mathbf{x}}) = 0 \ (\forall k), \quad \text{but } c_k(\overline{\mathbf{x}}) \neq 0 \text{ (in general)}. \tag{169}$$

Then, to prepare the determination of the next equilibrium point, $\overline{\mathbf{x}}_\ell$, we wish to construct a revised metamodel $\tilde{c}_k^{(\ell)}(\mathbf{x})$, shortly denoted $\tilde{c}_k(\mathbf{x})$ hereafter, that incorporates the following local novel information:

$$\forall k: \quad \overline{c}_k = c_k(\overline{\mathbf{x}}), \quad \nabla\overline{c}_k, \quad \frac{\partial^2 c_k}{\partial x_i^2}(\overline{\mathbf{x}}) \ (\forall i), \tag{170}$$

in which the values $\{\overline{c}_k\}$ are provided by an exact evaluation of the actual constraints, and the derivatives are approximated by accurate central differencing about $\overline{\mathbf{x}}$. The off-diagonal elements of the constraint Hessian are not revised for simplicity. This gives:

$$\forall k: \quad \tilde{c}_k(\mathbf{x}) = \overline{c}_k + \big(\mathbf{x}-\overline{\mathbf{x}}, \nabla\overline{c}_k + \tfrac{1}{2}\bar{\mathbf{H}}_{c_k}(\mathbf{x}-\overline{\mathbf{x}})\big). \tag{171}$$

In this way, the revised metamodel matches the actual constraint function $c_k(\mathbf{x})$ at the freshly-computed nearby point $\overline{\mathbf{x}}$. It is therefore a locally very accurate representation of the actual constraint. As a result, if the constraint metamodel is locally quadratic, it is not quadratic globally, since the quadratic form is upgraded at each step.

The above expression is then cast into a form similar to (168) by using again (165). This gives:

$$\mathbf{x} - \overline{\mathbf{x}} = \mathbf{x}_1 + \mathbf{\Omega}_\mathbf{u}\mathbf{u} \tag{172}$$

where

$$\mathbf{x}_1 = \mathbf{x}_A^\star - \overline{\mathbf{x}} + \mathbf{\Omega}_\mathbf{v}\mathbf{v}_0 \tag{173}$$

and

$$\nabla\overline{c}_k + \tfrac{1}{2}\tilde{\mathbf{H}}_{c_k}(\mathbf{x}-\overline{\mathbf{x}}) = \mathbf{g}_1 + \tfrac{1}{2}\bar{\mathbf{H}}_{c_k}\mathbf{\Omega}_\mathbf{u}\mathbf{u} \tag{174}$$

where

$$\mathbf{g}_1 = \nabla\overline{c}_k + \tfrac{1}{2}\bar{\mathbf{H}}_{c_k}\mathbf{x}_1 \tag{175}$$

so that:

$$\tilde{c}_k(\mathbf{u}) = \overline{c}_k + \big(\mathbf{x}_1 + \mathbf{\Omega}_\mathbf{u}\mathbf{u}, \mathbf{g}_1 + \tfrac{1}{2}\bar{\mathbf{H}}_{c_k}\mathbf{\Omega}_\mathbf{u}\mathbf{u}\big) = c_{0,k} + \gamma_{0,k}^t\mathbf{u} + \tfrac{1}{2}\mathbf{u}^t\mathbf{Q}_{c_k}\mathbf{u} \tag{176}$$

where the following definitions are made:

$$\begin{cases} c_{0,k} = \overline{c}_k + \big(\mathbf{x}_1, \mathbf{g}_1\big) & (1\times K) \\ \gamma_{0,k} = \mathbf{\Omega}_\mathbf{u}^t\left(\mathbf{g}_1 + \tfrac{1}{2}\bar{\mathbf{H}}_{c_k}\mathbf{x}_1\right) & (n_1\times 1\times K) \\ \mathbf{Q}_{c_k} = \mathbf{\Omega}_\mathbf{u}^t\,\bar{\mathbf{H}}_{c_k}\,\mathbf{\Omega}_\mathbf{u} & (n_1\times n_1\times K). \end{cases} \tag{177}$$

From the computational viewpoint, note that:

- For given ε (fixed by the outer-loop):
  - Set $\overline{\mathbf{x}}$ and $\{\overline{c}_k, \nabla\overline{c}_k, \bar{\mathbf{H}}c_k\}$ $(\forall k)$: at the very first time ($\ell = 1$), $\overline{\mathbf{x}} = \mathbf{x}_A^\star$, so that the constraints $\overline{c}_k = 0$ $(\forall k)$, while the gradients $\{\nabla\overline{c}_k\}$ have been set at Step 1, and the Hessians $\{\bar{\mathbf{H}}c_k\}$ at Step 3; at subsequent steps ($\ell > 1$), the constraint values are calculated by function calls, while constraint gradients and Hessians are computed locally by central deifferencing about $\overline{\mathbf{x}}$.
  - Update the matrices $\{\mathbf{Q}_{c_k}\}$ $(\forall k)$.
- For given $\mathbf{v}_0$ (fixed by the coordination inner loop): the vectors $\mathbf{x}_1$, $\mathbf{g}_1$ and $\{c_{0,k}, \gamma_{0.k}\}$ $(\forall k)$ are updated.

Then, all the elements are prepared for the determination of the subvector $\mathbf{u}_1$ resulting from $\mathbf{u}_0$.

Optimality conditions. These conditions express the stationarity of the following Lagrangian w.r.t. the state vector $\mathbf{u} \in \mathsf{R}^{n-p}$ and the vector of Lagrange multipliers $\lambda \in \mathsf{R}^K$:

$$\mathsf{L} = \mathsf{L}(\mathbf{u}, \lambda) = \phi_0 + f_0^t \mathbf{u} + \tfrac{1}{2}\mathbf{u}^t \mathbf{Q}_\mathbf{u} \mathbf{u} + \sum_{k=1}^{K} \lambda_k \left(c_{0,k} + \gamma_{0,k}^t \mathbf{u} + \tfrac{1}{2}\mathbf{u}^t \mathbf{Q}_{c_k} \mathbf{u} \right). \tag{178}$$

This gives:

$$\begin{cases} \nabla_\mathbf{u} \mathsf{L} = \mathbf{Q}_\mathbf{u} \mathbf{u} + f_0 + + \sum_{k=1}^{K} \lambda_k \left(\mathbf{Q}_{c_k} \mathbf{u} + \gamma_{0,k} \right) = 0 \\ \dfrac{\partial \mathsf{L}}{\partial \lambda_k} = c_{0,k} + \gamma_{0,k}^t \mathbf{u} + \tfrac{1}{2}\mathbf{u}^t \mathbf{Q}_{c_k} \mathbf{u} = 0 \quad (\forall k) \end{cases} \tag{179}$$

which is a system of $n - p + K$ quadratic equations for as many unknowns. The system is solved by Newton's method:

$$\mathbf{Q}_N \begin{pmatrix} \mathbf{u}^{\mu+1} - \mathbf{u}^\mu \\ \lambda^{\mu+1} - \lambda^\mu \end{pmatrix} = - \begin{pmatrix} \nabla_\mathbf{u} \\ \nabla_\lambda \end{pmatrix} \mathsf{L}^\mu \tag{180}$$

where μ is the iteration index, and

$$\mathbf{Q}_N = \nabla^2 \mathsf{L} = \begin{pmatrix} \nabla_\mathbf{u} \\ \nabla_\lambda \end{pmatrix} \begin{pmatrix} \nabla_\mathbf{u}^t \mathsf{L} & \nabla_\lambda^t \mathsf{L} \end{pmatrix} = \begin{pmatrix} \nabla_\mathbf{u}^t \nabla_\mathbf{u} \mathsf{L} & \nabla_\mathbf{u} \nabla_\lambda^t \mathsf{L} \\ \nabla_\lambda \nabla_\mathbf{u}^t \mathsf{L} & \nabla_\lambda \nabla_\lambda^t \mathsf{L} \end{pmatrix} \tag{181}$$

is of dimension $(n_1 + K) \times (n_1 + K)$, and

$$\nabla_\mathbf{u}^t \nabla_\mathbf{u} \mathsf{L} = \nabla_{\mathbf{uu}}^2 \mathsf{L} = \mathbf{Q}_\mathbf{u} + \sum_{k=1}^{K} \lambda_k \mathbf{Q}_{c_k} \tag{182}$$

is of dimension $n_1 \times n_1$, and

$$\nabla_\mathbf{u} \nabla_\lambda^t \mathsf{L} = \begin{pmatrix} \left(\mathbf{Q}_{c_1} \mathbf{u} + \gamma_{0,1}\right)_1 & \dots & \left(\mathbf{Q}_{c_K} \mathbf{u} + \gamma_{0,K}\right)_1 \\ \vdots & & \vdots \\ \left(\mathbf{Q}_{c_1} \mathbf{u} + \gamma_{0,1}\right)_{n_1} & \dots & \left(\mathbf{Q}_{c_K} \mathbf{u} + \gamma_{0,K}\right)_{n_1} \end{pmatrix}, \quad \nabla_\lambda \nabla_\mathbf{u}^t \mathsf{L} = \left(\nabla_\mathbf{u} \nabla_\lambda^t \mathsf{L}\right)^t, \tag{183}$$

are of dimension $n_1 \times K$ and $K \times n_1$ respectively, and and

$$\nabla_\lambda \nabla_\lambda^t \mathsf{L} = 0 \tag{184}$$

is the null matrix of dimension $K \times K$.

An estimate for vector $\mathbf{u}$ is known either from asymptotics for the very first equilibrium, or as the converged vector at the previous equilibrium. This estimate is used for $\mathbf{u}^0$.

Concerning λ^0, for the very first equilibrium, the Lagrange multipliers calculated at STEP 8, (57), are used. For subsequent equilibria, λ^0 is set to the converged vector at the previous equilibrium.

The iteration is interrupted either when a convergence criterion is met or when μ exceeds a specified number $\mu_{\max}$, often set to 10. The convergence criterion is $\left\|\mathbf{u}^{\mu+1} - \mathbf{u}^\mu\right\| < s$ (Euclidean norm) where the tolerance s is set to $TOL/100$, where TOL is the tolerance on the $(\mathbf{u}, \mathbf{v})$ coordination outer loop measured by variations in vector $\mathbf{v}$.

Very few iterations are frequently sufficient, except at the maximum allowable ε, $\varepsilon_{\max}$, for which the convexity of the $\mathbf{v}$-subproblem (see next subsection) is insured.

C.2 Unconstrained optimization of $\mathbf{v} \in \mathsf{R}^p$ for fixed $\mathbf{u} = \mathbf{u}_1 \in \mathsf{R}^{n-p}$

For fixed $\mathbf{u} = \mathbf{u}_1 \in \mathsf{R}^{n-p}$, the subvector $\mathbf{v} \in \mathsf{R}^p$ is optimized to realize the unconstrained minimization of the auxiliary cost function

$$f_{AB} = (1 - \varepsilon) f_A^+ + \varepsilon \tilde{f}_B \tag{185}$$

where $f_A^{+,\star} = \tilde{f}_B^{\star} = 1$ by construction, and here

$$\mathbf{x} - \mathbf{x}_A^{\star} = \mathbf{\Omega_u} \mathbf{u}_1 + \mathbf{\Omega_v} \mathbf{v} \tag{186}$$

and $\mathbf{u}_1$ is the end result of the previous section, maintained fixed in this optimization. Hence $\mathbf{v}$ is the solution of the stationarity equation:

$$(1-\varepsilon)\nabla_{\mathbf{v}} f_A^{+} + \varepsilon \nabla_{\mathbf{v}} \tilde{f}_B = 0 \tag{187}$$

where

$$\nabla_{\mathbf{v}} = \mathbf{\Omega}_{\mathbf{v}}^t \nabla \tag{188}$$

is the symbol for the partial gradient w.r.t. $\mathbf{v}$ (for constant $\mathbf{u}$).

Differentiating (16) yields:

$$\nabla f_A^{+} = \nabla \tilde{f}_A^{\star} + \mathbf{H}_A^{+,\star}(\mathbf{x} - \mathbf{x}_A^{\star}) = \nabla \tilde{f}_A^{\star} + \mathbf{H}_A^{+,\star}(\mathbf{\Omega_u} \mathbf{u}_1 + \mathbf{\Omega_v} \mathbf{v}). \tag{189}$$

Similarly:

$$\nabla \tilde{f}_B = \nabla f_B^{\star} + \mathbf{H}_B^{\star}(\mathbf{x} - \mathbf{x}_A^{\star}) = \nabla f_B^{\star} + \mathbf{H}_B^{\star}(\mathbf{\Omega_u} \mathbf{u}_1 + \mathbf{\Omega_v} \mathbf{v}). \tag{190}$$

Hence $\mathbf{v}$ is the solution of the $p \times p$ linear system:

$$\mathbf{\Omega}_{\mathbf{v}}^t \left\{ (1-\varepsilon) \left[\nabla \tilde{f}_A^{\star} + \mathbf{H}_A^{+,\star}(\mathbf{\Omega_u} \mathbf{u}_1 + \mathbf{\Omega_v} \mathbf{v}) \right] + \varepsilon \left[\nabla f_B^{\star} + \mathbf{H}_B^{\star}(\mathbf{\Omega_u} \mathbf{u}_1 + \mathbf{\Omega_v} \mathbf{v}) \right] \right\} = 0 \tag{191}$$

that is

$$\mathbf{Q_v} \mathbf{v} + \phi_{AB\mathbf{u}} = 0 \tag{192}$$

where

$$\begin{cases} \mathbf{Q_v} = \mathbf{\Omega}_{\mathbf{v}}^t \left[(1-\varepsilon)\, \mathbf{H}_A^{+,\star} + \varepsilon\, \mathbf{H}_B^{\star} \right] \mathbf{\Omega_v} \\ \phi_{AB\mathbf{u}} = \mathbf{\Omega}_{\mathbf{v}}^t \left\{ (1-\varepsilon) \nabla \tilde{f}_A^{\star} + \varepsilon \nabla f_B^{\star} + \left[(1-\varepsilon) \mathbf{H}_A^{+,\star} + \varepsilon \mathbf{H}_B^{\star} \right] \mathbf{\Omega_u} \mathbf{u}_1 \right\} \end{cases} \tag{193}$$

The following vectors

$$\phi_A = \mathbf{\Omega}_{\mathbf{v}}^t\, \nabla \tilde{f}_A^{\star}, \quad \phi_B = \mathbf{\Omega}_{\mathbf{v}}^t\, \nabla f_B^{\star}, \qquad (p \times 1) \tag{194}$$

and matrices

$$\begin{cases} \mathbf{H}_{A\mathbf{u}} = \mathbf{\Omega}_{\mathbf{v}}^t\, \mathbf{H}_A^{+,\star}\, \mathbf{\Omega_u}, \quad \mathbf{H}_{B\mathbf{u}} = \mathbf{\Omega}_{\mathbf{v}}^t\, \mathbf{H}_B^{\star}\, \mathbf{\Omega_u} & (p \times n_1) \\ \mathbf{H}_{A\mathbf{v}} = \mathbf{\Omega}_{\mathbf{v}}^t\, \mathbf{H}_A^{+,\star}\, \mathbf{\Omega_v}, \quad \mathbf{H}_{B\mathbf{v}} = \mathbf{\Omega}_{\mathbf{v}}^t\, \mathbf{H}_B^{\star}\, \mathbf{\Omega_v} & (p \times p) \end{cases} \tag{195}$$

are calculated once for all, prior to the Nash games.

The following vector

$$\phi_{AB} = (1-\varepsilon)\phi_A + \varepsilon \phi_B \qquad (p \times 1) \tag{196}$$

and matrices

$$\begin{cases} \mathbf{Q_v} = \mathbf{H}_{AB\mathbf{v}} = (1-\varepsilon)\, \mathbf{H}_{A\mathbf{v}} + \varepsilon\, \mathbf{H}_{B\mathbf{v}} & (p \times p) \\ \mathbf{H}_{AB\mathbf{u}} = (1-\varepsilon)\, \mathbf{H}_{A\mathbf{u}} + \varepsilon\, \mathbf{H}_{B\mathbf{u}} & (p \times n_1) \end{cases} \tag{197}$$

are calculated once after each update of ε ("outer loop"). Additionally the matrix $\mathbf{Q_v}$ is factorized by a call to the DPOTRF procedure of the Lapack Library,

Lastly, after each update of vector $\mathbf{u}_1$ ("inner loop"), the following vector is calculated

$$\phi_{AB\mathbf{u}} = \phi_{AB} + \mathbf{H}_{AB\mathbf{u}}\, \mathbf{u}_1 \tag{198}$$

and the system (192) is solved by backward substitution by call to the DPOTRS procedure. The solution is denoted $\mathbf{v}_1$.

Condition ensuring a positive-definite matrix $\mathbf{Q_v}$. Let

$$c = \frac{\varepsilon}{1-\varepsilon}, \quad \text{and } \mathbf{q_v} = \frac{\mathbf{Q_v}}{1-\varepsilon} = \mathbf{H}_{A\mathbf{v}} + c\,\mathbf{H}_{B\mathbf{v}}. \tag{199}$$

Let us first examine the quadratic form $q_{A\mathbf{v}}(\mathbf{x})$ associated with the matrix $\mathbf{H}_{A\mathbf{v}}$:

$$q_{A\mathbf{v}}(\mathbf{x}) = \mathbf{x}^t\,\mathbf{H}_{A\mathbf{v}}\,\mathbf{x} = \mathbf{y}^t\,\mathbf{H}_A^{+,\star}\,\mathbf{y} \quad (\mathbf{x} \in \mathsf{R}^p) \quad (\mathbf{y} = \mathbf{\Omega_v}\mathbf{x} \in \mathsf{R}^n). \tag{200}$$

Evidently, since the matrix $\mathbf{H}_A^{+,\star}$ is real-symmetric positive-definite, the quadratic form is non-negative. Additionally, the equation $q_{A\mathbf{v}}(\mathbf{x}) = 0$ implies that $\mathbf{y} = 0 \in \mathsf{R}^n$, and since the column vectors of the matrix $\mathbf{\Omega_v}$, as a subset of the eigenvectors of a real-symmetric matrix, are linearly independent, in fact orthogonal, this requires that $\mathbf{x} = 0 \in \mathsf{R}^p$. Hence, the matrix $\mathbf{H}_{A\mathbf{v}}$ is positive-definite. This legitimates the following factorization

$$\mathbf{q_v} = \mathbf{H}_{A\mathbf{v}}^{\frac{1}{2}}\,[\mathbf{I}_p + c\mathbf{H}_{BA}]\,\mathbf{H}_{A\mathbf{v}}^{\frac{1}{2}} \tag{201}$$

where:

$$\mathbf{H}_{BA} = \mathbf{H}_{A\mathbf{v}}^{-\frac{1}{2}}\,\mathbf{H}_{B\mathbf{v}}\,\mathbf{H}_{A\mathbf{v}}^{-\frac{1}{2}} \tag{202}$$

It is clear from (201) that the matrix $\mathbf{q_v}$ is positive-definite iff the matrix $\mathbf{I}_p + c\mathbf{H}_{BA}$ itself is so. Thus let λ_{BA} be the (algebraically) smallest eigenvalue of the matrix $\mathbf{H}_{BA}$. The condition writes

$$1 + c\lambda_{BA} > 0. \tag{203}$$

Hence, if

$$\lambda_{BA} \geq 0, \tag{204}$$

no limitation should be imposed on ε besides $\varepsilon < 1$. Otherwise, if

$$\lambda_{BA} < 0, \tag{205}$$

the condition is:

$$\varepsilon < \varepsilon_{\max} = \frac{1}{1-\lambda_{BA}}. \tag{206}$$

In summary, we are led to compute the eigenvalue λ_{BA} and test. The calculation of the eigenvalue itself is performed by solving the generalized eigenvalue-problem $\mathbf{H}_{B\mathbf{v}}\mathbf{x} = \lambda\mathbf{H}_{A\mathbf{v}}\mathbf{x}$ which admits the same set of eigenvalues. This is done a priori, yielding the limit $\varepsilon_{\max}$ on ε.

C.3 Global algorithm

See flowchart in Figure 24 next.

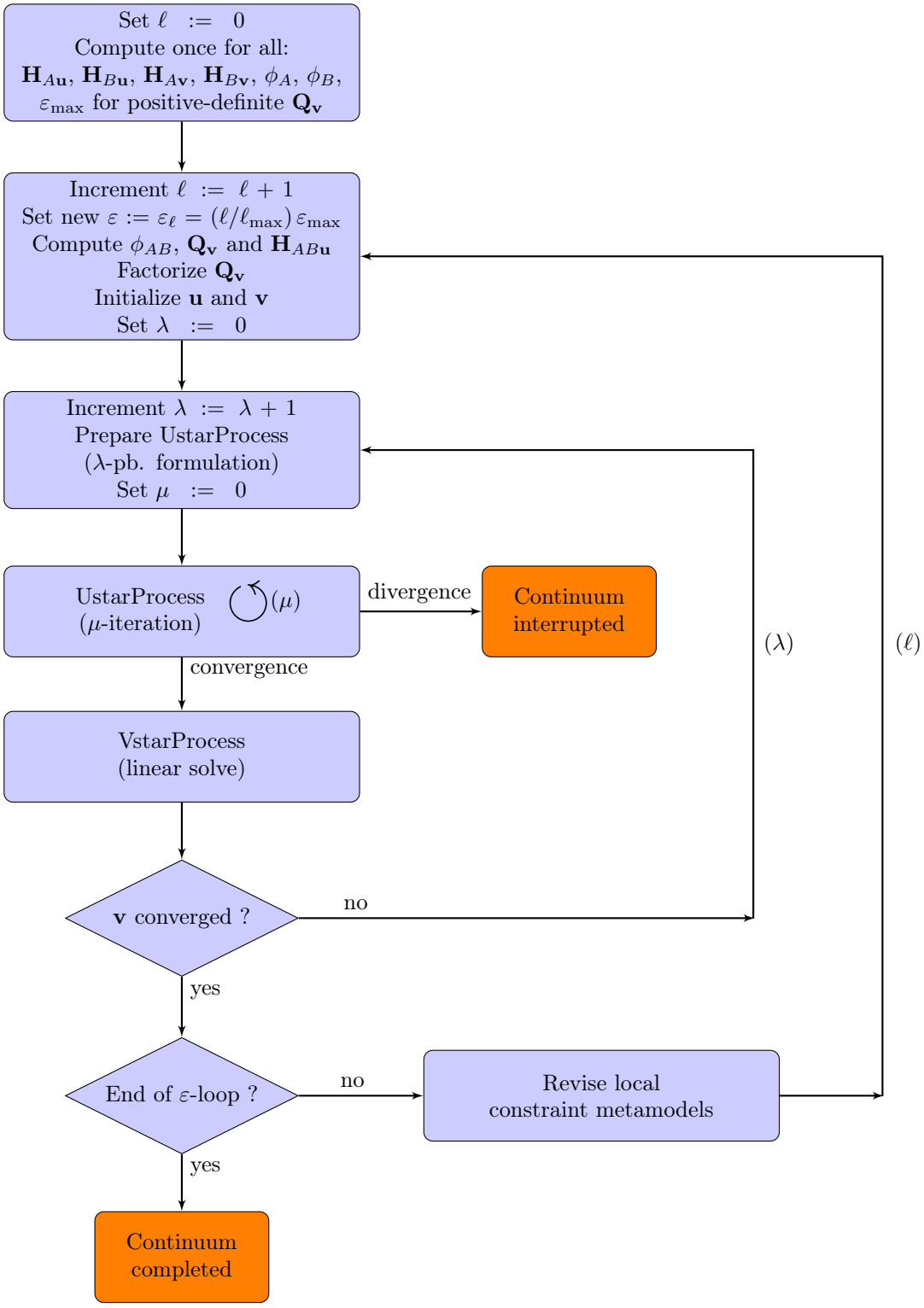


Figure 24: Flowchart of the (ℓ, λ, μ) iterative algorithm for the determination of the continuum of Nash equilibria. The continuation parameter ε in incremented in the interval $[0, \varepsilon_{\max}]$ over which the VstarProcess corresponds to the solution of a convex minimization problem. Most frequently the iteration terminates when the minimization problem in the UstarProcess ceases to be convex ("Continuum interrupted"), and exceptionally when the incrementation of ε can be continued up to $\varepsilon = 1$ ("Continuum completed").

Contents

1 Introduction **4**

2 Standard algorithm **4**

3 Metamodel-assisted variant **10**
3.1 Game preparation 10
3.2 Step-by-step determination of the continuum of Nash equilibria 14

4 Analytical testcases solved using the Nash-MGDA Platform **15**
4.1 Test-case TC4 15
4.2 Variations about the classical Fonseca-Fleming test-case 25

5 A test-case in structural analysis: design of a sandwich panel **30**

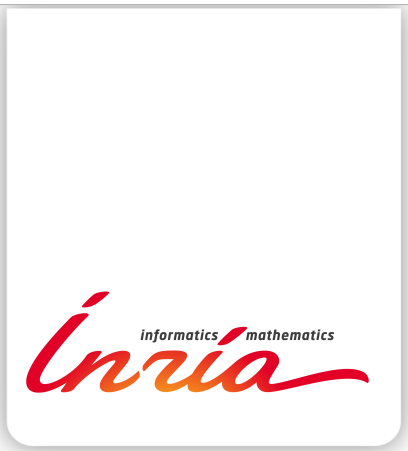
6 A test-case in flight mechanics **37**

7 Conclusions **45**

A Geometrical database **47**

B Metamodels **50**
B.1 Quadratic metamodels for the nonlinear constraints 50
B.2 Quadratic metamodel for the prime steering function 51
B.3 Quadratic metamodel for the secondary steering function 52

C Optimization of u and v in the Nash game **53**
C.1 Constrained optimization of $\mathbf{u} \in \mathsf{R}^{n-p}$ for fixed $\mathbf{v} = \mathbf{v}_0 \in \mathsf{R}^p$ 53
C.2 Unconstrained optimization of $\mathbf{v} \in \mathsf{R}^p$ for fixed $\mathbf{u} = \mathbf{u}_1 \in \mathsf{R}^{n-p}$ 55
C.3 Global algorithm 57



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399