

Multi-objective analysis of computational models

Stephane Doncieux^{1,*}, Jean Liénard¹, Benoît Girard¹, Mohamed Hamdaoui², Joël Chaskalovic³,

1 Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222, Paris, France

2 Department of Applied Mathematics and Systems, Ecole Centrale Paris, Paris, France

3 Institut Jean le Rond d'Alembert, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7190, Paris, France

* **E-mail: stephane.doncieux@isir.upmc.fr**

abstract

Computational models are of increasing complexity and their behavior may in particular emerge from the interaction of different parts. Studying such models becomes then more and more difficult and there is a need for methods and tools supporting this process. Multi-objective evolutionary algorithms generate a set of trade-off solutions instead of a single optimal solution. The availability of a set of solutions that have the specificity to be optimal relative to carefully chosen objectives allows to perform data mining in order to better understand model features and regularities. We review the corresponding work, propose a unifying framework, and highlight its potential use. Typical questions that such a methodology allows to address are the following: what are the most critical parameters of the model? What are the relations between the parameters and the objectives? What are the typical behaviors of the model? Two examples are provided to illustrate the capabilities of the methodology. The features of a flapping-wing robot are thus evaluated to find out its speed-energy relation, together with the criticality of its parameters. A neurocomputational model of the Basal Ganglia brain nuclei is then considered and its most salient features according to this methodology are presented and discussed.

Nomenclature

\mathbf{X}	parameter to be optimized ($\mathbf{X} \in \mathbb{R}^m$)
$f(\mathbf{X})$	function to be optimized
\prec	dominance relation
\preceq	dominance or equality relation
χ	set of Pareto set approximations
Γ	set of independent runs of the optimizer
$\alpha_\chi(z)$	attainment function
Ψ_1	attainment surface corresponding to a value of 1 of the attainment function
Ψ_0	attainment surface corresponding to a value of 0 of the attainment function
$I_H^p(\chi_0)$	hypervolume of the set of points χ_0 relative to the reference point p
$\eta(\chi_0)$	nadir point
$\iota(\chi_0)$	ideal point
$\bar{\eta}$	conservative nadir point
<i>Flapping wing aircraft example</i>	
DI	wing dihedral
TW_i	internal twist
TW_e	external twist
a_{DI}	amplitude of the wing dihedral
p_{DI}	period of the wing dihedral
a_{TW_i}	amplitude of the internal twist
p_{TW_i}	phase of the internal twist

r_{TW_i}	reference of the internal twist
a_{TW_e}	amplitude of the external twist
p_{TW_e}	phase of the external twist
r_{TW_e}	reference of the external twist
τ	instantaneous torque
ω	instantaneous rotational speed
P	mechanical power
b	aircraft wingspan
S	wing area
c_m	mean chord
U	aircraft cruise velocity
ν	air kinematical viscosity
Re	Reynolds number
St	Strouhal number
k	reduced frequency
κ_1	reduced internal twist frequency
κ_2	reduced external twist frequency
<i>Basal ganglia model</i>	
BG	basal ganglia brain nuclei
CBG	Contracting Basal Ganglia model [1]
GPR	Gurney, Prescott and Redgrave basal ganglia model [2]
WTA	Winner-Takes-All
GPI_j	j -th channel of the basal ganglia output (Globus pallidus internal)
sc	selected channel
nsc	non selected channel
GPe	Globus Pallidus external
$D1$	Striatum neurons with D1 dopamine receptors
$D2$	Striatum neurons with D2 dopamine receptors
FSI	Fast Spiking Interneurons

Introduction

Computational models are now pervasive in most domains of science, including astrophysics, mechanics, neuroscience, geophysics, to name a few. They allow to compute the behavior of a system out of a particular context and of specific mathematical relations, thus allowing the simulation of the observed phenomenon [3]. The specificity of the scientific approach relying on such models is that the human researcher generally doesn't master the whole process ranging from the initial mathematical model to the behavior exhibited in a particular context, leading to a so called *epistemic opacity* [4]:

A process is epistemically opaque relative to a cognitive agent X at time t just in case X does not know at t all of the epistemically relevant elements of the process.

Such an opacity stems from the fact that scientific work in this context involves so complex computations, that no human or group of humans can handle it without the help of a computer, thus leading to an hybrid scenario in which humans and machines are both critical. Likewise, the required validation of a computational model involves fitting model outputs with experimental data. A trial and error process may then be required to find the most adapted parameters among those that are justified by the assumptions made while building the model. This intrusion of heuristics, although perfectly justified [5], also adds some opacity in the research process based on computational models.

A computational model allows to construct new knowledge on the basis of what is learned during the model construction process and of the manipulations it allows [6]. [7] proposes to consider the following steps in the process of knowledge acquisition from a model:

- *denotation* step: relations are built between the model and the target, e.g. model subparts are justified or discussed on the basis of current knowledge;
- *demonstration* step: investigations on the features of the model;
- *interpretation* step: conversion of the findings from the model to the target system.

On a complex model, the demonstration step may be particularly difficult because of the epistemic opacity. Features to study may naturally stem from the construction process, but some properties – as those emerging from the interaction of different parts, for instance – can hardly be induced if no evidence drives the exploration in this particular direction. In this perspective, the computational model is in itself a complex system to study in order to unravel its unforeseen features. Such an object could be studied in the same way as natural systems, i.e. by building a simpler model exhibiting the same behavior, but there is a specificity that can be exploited to propose an alternative procedure: the ability to compute the behavior of the model allows to perform a huge number of experiments with different contexts or with different model parameters. This particular feature can be exploited in a systematic way to generate experimental data that will help studying and then understanding such models.

We propose such a method to help analyzing computational models thanks to a procedure generating dedicated experimental data. Such data are not outputs of the model, but sets of parameter values of the model associated to selected behaviors. The method automatically finds both parameter sets and behaviors, thus opening the way to the analysis of these particular behaviors. The method relies on the idea that the model optimizes more than one function. Such functions may correspond to an evaluation of how the models fits to several different experimental data or of how the model fits to other validated analytical models of the studied phenomenon. Once such functions have been defined, the method consists in using multi-objective optimization algorithms in order to generate a dense approximation of the set of trade-off solutions. Looking for a dense approximation and not a sparse one opens the way to the study of regularities in both parameter and behavior spaces.

To reach this goal, we propose to use multi-objective evolutionary algorithms, that are efficient optimization algorithms in a multi-objective context and that also feature a high versatility, allowing to optimize any kind of model for any kind of objective function. The only restriction results from the inner working of evolutionary algorithms: as they rely on the evaluation of up to millions of different solutions, computational models will need to be fast enough – or available computational power needs to be high enough – to compute their behavior a huge number of times. The set of trade-off solutions will then be analyzed in order to gain new knowledge about the model, like the importance of a particular parameter, the different possible model behaviors and their features, etc. The proposed methodology relies on the following steps:

1. defining the relevant functions to be optimized (at least two);
2. finding a dense approximation of the set of trade-off solutions;
3. analyzing these points both in the parameter space and in the space of optimized functions values, i.e. the objective space.

After a review of related work, some basic notions required to understand the different aspects of the methodology are presented. The different steps of the method are then proposed together with a list of questions that the method might contribute to answer. Some tools to be used to extract knowledge from generated data relative to these questions are also reviewed. Two examples are then considered and studied with the proposed methodology:

- a flapping wing aircraft model;
- a model of the basal ganglia brain nuclei.

After a short introduction to the corresponding models, the methodology is applied to their study and the knowledge thus gained is highlighted.

Related work

Optimization methods in general and evolutionary algorithms in particular are frequently used in the engineering field to increase the profitability of an industrial process or the efficiency of a system. Besides this use, whose economical consequences has driven the optimization field for years, such methods can be used to support scientists' work.

The most typical use consists in tuning parameters of models relative to given performance functions. The performance in this case is generally the closeness to available experimental data. A second and more specific use consists in considering the case of a natural system which has been optimized in some way through its history. Using an optimization process and looking at the results can be used as a validation for hypotheses on the objectives subject to optimization, or on the search space. Such an approach is used in biology, for instance, with the hypothesis that animals or plants had to optimize their ability to propagate their genes because of natural selection. A last use consists in using optimization in a data mining context, in order to discover new knowledge from a huge set of experimental data or even, and this is what we propose here, to generate data to dig in.

Model tuning

Computational models aim at reproducing observed phenomena to validate hypotheses on the inner working of the involved systems and make predictions on the behavior to be observed in a different context. As the behavior of most models is highly dependent on some parameters whose values cannot always be deduced from direct observations, finding out the best parameter set to reproduce available experimental data is then a critical task for model designers. In this case, optimization is of particular interest with a cost function that typically measures the distance of model outputs to experimental data. Such use of optimization algorithms are frequent.

In computational neuroscience, where the models are of increasing complexity, it has been used, for instance, to tune single neuron models [8], neural fields [9, 10] or neural network models of specific functions [11–13].

In hydrological modeling, model parameter identification is a difficult task because of their intercorrelations that may result in compensating errors. Optimization algorithms have then been used to identify model parameters and evolutionary algorithms revealed to be competitive with other optimization algorithms [14, 15]. Considering that antagonists objectives actually have to be optimized, multi-objective approaches have been proposed [16, 17], see [18] for a review. Other physics related fields, like materials science, also consider model optimization with evolutionary algorithms [19–21], see [22] for a review.

What is "optimized" by a particular process?

Optimization tools can also be used to test hypotheses about a natural process that also performs some kind of optimization. In this case, a computational model is also optimized, but for a different purpose than mere model building. Several cost functions are defined and used to optimize the model. The one providing the closest behavior to the observed phenomena will be supposed to be the valid hypothesis.

Biologists use such an approach to study the criteria optimized by an animal. Starting with an idealized model that is generic enough, several cost functions are proposed and the resulting behaviors

are compared to experimental data to see if they fit. The conclusions of such work may be the confirmation of hypothesis on the objectives definition and on the search space. Such an approach has been used to study biomechanical adaptations in bone tissue [23] or migration behavior of birds [24], for instance. Likewise, hypotheses on the influence of wiring cost on neuron placement were investigated with the use of optimization algorithms [25].

Evolutionary algorithms and data mining

Scientists have to face two different challenges, while studying a phenomenon:

- extracting new knowledge from experimental data in order to validate an hypothesis (relevance of a model, for instance);
- and before that, generating relevant experimental data to analyze.

These two steps are of utmost importance in a scientist work. The first point consists in extracting an answer to a given question from a potentially huge set of data. This is the goal of the data mining field which has proposed algorithms to make this process more efficient, see *Data Mining* section below for an introduction. Evolutionary algorithms have been proposed to support this step. In this context, evolutionary algorithms are used to induce rules that can predict future data [26, 27], i.e. they are automatically building a predictive model. Other uses of evolutionary algorithms include clustering or data preparation [28].

The second point corresponds to the design of the experiment that subsequent analyses will rely on. This step can hardly be completely automatized, but optimization algorithms can be part of it. Multi-objective evolutionary algorithms revealed in different context to generate data from which new knowledge can be extracted. [18], for instance, used it to get some insight on model errors and uncertainties. [29] used multi-objective optimization in an engineering context to generate data whose analyses helped discovering design principles for some devices. [30], for instance, used such an approach to find design principles for electric brushless motors. The approach, called *INNOVIZATION* – which stands for INNOVation through optimIZATION – consists in first choosing some antagonistic objectives (two, at least), then using multi-objective optimization tools to get an approximation of the set of Pareto optimal solutions, and then look at the solutions to discover regularities specific to the considered system. Self-Organizing Maps (SOM) and Analysis of Variance (ANOVA) were also used to analyze designs resulting from a multi-objective approach and better understand the relations between design variables and objective functions for several aerodynamic design problems [31, 32]. The proposed framework unifies such approaches and extends its use to different aspects of computational model analysis.

Background

Multi-objective problems

Optimization problems can be formalized as follows:

Find the parameter $\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{Bmatrix} \in \mathbb{R}^m$ that maximizes (or minimizes) $f(\mathbf{X})$ under the

constraints:

$$\begin{aligned} g_j(\mathbf{X}) &\leq 0, & j = 1, 2, \dots, p \\ l_j(\mathbf{X}) &= 0, & j = 1, 2, \dots, q \end{aligned}$$

where $g_j(\mathbf{X})$ and $l_j(\mathbf{X})$ are functions that express respectively inequality and equality constraints.

When $f(\mathbf{X})$ takes its values in \mathbb{R} , the problem is called a mono-objective problem. When $f(\mathbf{X})$ takes its values in \mathbb{R}^n with $n > 1$, the problem is called a multi-objective problem. A mono-objective problem generally has only a single optimal solution. Single objective optimizations result then generally in a single solution. In a multi-objective problem, when the objectives are antagonistic, multiple trade-off solutions do exist. Multi-objective optimization algorithms may then return either one particular trade-off or a set of different trade-offs. In this work, we will focus on this last case.

In a single objective context, comparing two solutions is straightforward: \mathbf{X}_1 is better than \mathbf{X}_2 if and only if $f(\mathbf{X}_1) > f(\mathbf{X}_2)$. In a multi-objective context, comparing two solutions is more difficult. A straightforward approach consists in computing a weighted sum of each component $f_i(\cdot)$ of $f(\cdot)$. With such an approach, \mathbf{X}_1 is better than \mathbf{X}_2 if and only if $\sum_{i=1}^n w_i f_i(\mathbf{X}_1) > \sum_{i=1}^n w_i f_i(\mathbf{X}_2)$. This approach consists in choosing *a priori* the relative importance of each objective. Another approach is possible in which no such choice is required. Such an approach requires to define a new dominance relation adapted to the multi-objective case. In this work, we will use the Pareto dominance relation defined as follows:

A solution \mathbf{X}_1 is said to dominate another solution \mathbf{X}_2 , if both conditions 1 and 2 are true:

1. the solution \mathbf{X}_1 is not worse than \mathbf{X}_2 with respect to all objectives;
2. the solution \mathbf{X}_1 is strictly better than \mathbf{X}_2 with respect to at least one objective.

In the following, $\mathbf{X}_2 \prec \mathbf{X}_1$ will indicate that \mathbf{X}_1 dominates \mathbf{X}_2 , $\mathbf{X}_2 \preceq \mathbf{X}_1$ will indicate that either \mathbf{X}_1 dominates \mathbf{X}_2 or that $\mathbf{X}_1 = \mathbf{X}_2$.

This dominance relation is not a strict but a partial ordering relation. The consequence is that, using this ordering relation, there are generally multiple optimal solutions. As we rely on the Pareto dominance relation, such solutions are called Pareto optimal solutions. The set of non-dominated solutions within the entire feasible search space is the called the globally Pareto-optimal set.

The existence of multiple optimal solutions is the core of the proposed approach: a multi-objective optimization problem will be defined, its Pareto-optimal set will be searched for and analyzed using data mining techniques to provide new insights on the considered computational model.

Multi-objective optimization algorithms

How to find the globally Pareto-optimal set? The models for which the proposed approach will be the most useful will be complex, non linear, discontinuous models, that may have a temporal aspect. Such models are the most difficult to understand and are then those for which the need for analysis tools is the most critical. This requirement implies to choose versatile optimization tools. Metaheuristics are then a good choice for their ability to deal with such problems [33]. They don't impose any mathematical constraint and are thus widely applicable, but with the drawback of a convergence to an *approximation* of the optimal solutions.

Multi-objective evolutionary algorithms

Among metaheuristics, evolutionary algorithms have shown to be particularly efficient in a multiobjective context [34]. This feature stems from a specificity of evolutionary algorithms: they optimize in parallel a set of solutions instead of a single one. It makes them easy to adapt to a multi-objective context as each point of the population may converge to a particular trade-off solution. Multiple algorithms have been defined – like NSGA-II [35] or ϵ -MOEA [36], for instance – and they proved to be efficient in multiple applications [37].

Evolutionary algorithms rely on an inspiration from Darwinian natural selection and feature a set of candidate solutions, called a population. Each solution is evaluated and its performance is used to apply a selection pressure: fittest individuals survive to the next generation and generate proportionally more

similar and new solutions than others. New solutions are copies of their ancestors with some random alterations, called mutations. They can also have several ancestors, inheriting some parts of each ancestor through a crossover operator. A typical evolutionary algorithm can be summarized as follows:

1. generate a random set of solutions
2. evaluate each of them
3. select the fittest solutions
4. generate new solutions
5. evaluate each of them
6. if the number of cycles is below a chosen threshold, go back to 3, otherwise stop the optimization.

See [38] for an introduction to evolutionary algorithms.

As evolutionary algorithms only require to evaluate the objective function $f(\cdot)$ at different points, these algorithms indeed impose no mathematical constraints. They typically don't need to know the derivative of the function, for instance. Their drawback is that they require a huge number of evaluations to converge (typically from 10^5 to 10^6 evaluations). The consequence for the proposed approach is that the computational models under study must be fast enough to allow a huge number of computer simulations.

Indicators

In the following, different indicators to be used within the proposed multi-objective analysis method are defined.

Let denote χ the set of approximations of the globally Pareto optimal set generated by a set of runs Γ . As evolutionary algorithms feature a stochastic behavior, each time an optimization is actually performed, a particular approximation of the globally Pareto-optimal set of points is also generated. χ_i will be the approximation of the Pareto optimal set generated by the i -th run.

The *attainment function* $\alpha_\chi(z)$ [39] aims at measuring the performance of an algorithm while taking into account its stochastic feature. $\alpha_\chi(z)$ can be defined as the probability of finding at least one solution which attains z (in our case that dominates z) out of the set of optimization results χ . It can be estimated via its empirical counterpart defined over r sets of optimization results as follows:

$$\alpha_r(z) = \frac{1}{r} \sum_{i=1}^r I\{\chi_i \supseteq z\}$$

where $I\{\cdot\}$ is the indicator function, equal to 1 if the assertion is true or 0 else. \supseteq represents the following relation:

$$\chi_i \supseteq z \equiv \exists x \in \chi_i \mid x \succeq z$$

An *attainment surface* [40] can then be defined as the hyper surface described by a particular value of the attainment function. Some particular attainment surfaces will be of particular interest in the following: the 1-attainment surface and the 0-attainment surface. The 1-attainment surface Ψ_1 represents the set of the "worst" points among the solutions, i.e. the set of the most dominated points. The 0-attainment surface Ψ_0 represents the set of the "best" points among the solutions, i.e. the set of non-dominated solutions over all the results.

In the following, as suggested in [41], for the definition of the hypervolume described below, we will simplify this definition and consider that χ is composed of only one set of points at a time, noted χ_0 . The definition of $\alpha_{\chi_0}(z)$ becomes (figure 1, left):

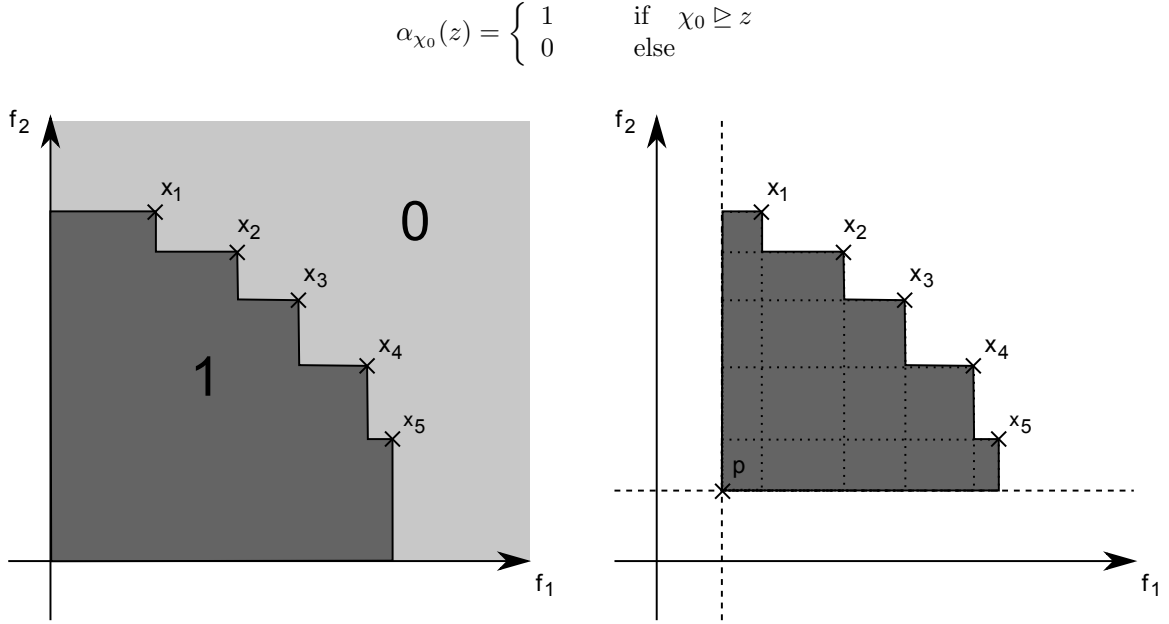


Figure 1. Left: values of the attainment function $\alpha_{\chi_0}(z)$ for $\chi_0 = \{x_1, x_2, \dots, x_5\}$. Right: the hypervolume $I_H^p(\chi_0)$ is the area of the surface colored in gray.

The hypervolume indicator of a particular non dominated set χ_0 , $I_H^p(\chi_0)$ [41, 42] can be defined, for a given reference point p as:

$$I_H^p(\chi_0) = \int_{\phi_p} \alpha_{\chi_0}(z) dz$$

with $\phi_p \subset \mathbb{R}^n$ the set of values above p , i.e. $\phi_p = \{y \in \mathbb{R}^n | \forall i, y_i \geq p_i\}$. This is a generalized definition that holds for any dominance relation [41]. In the case of the Pareto dominance relation, as we will use here, and in two dimensions, it actually corresponds to the area of the union of the rectangles defined by the points (p, x_i) with $x_i \in \chi_0$ (figure 1, right).

The *nadir* objective vector $\eta(\chi_0)$ of a set $\chi_0 \subset \mathbb{R}^n$ is defined as the point with the least value possible for each objective:

$$\eta(\chi_0) = \{\min_{y \in \chi_0}(y_i), i = 1, 2, \dots, n\}$$

Likewise, the *ideal* objective vector $\iota(\chi_0)$ of a set $\chi_0 \subset \mathbb{R}^m$ is defined as the point with the maximum value possible for each objective:

$$\iota(\chi_0) = \{\max_{y \in \chi_0}(y_i), i = 1, 2, \dots, m\}$$

Data mining

The ability to gather huge sets of data is facilitated by modern technologies. This has pushed forward the need to automate the process of extracting from such huge data sets consistent relationships between

variables, leading to the Data Mining field. Using a combination of machine learning, statistical analysis, modeling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results.

Two main families of data mining tools do exist: the first one, called the *supervised data mining*, consists in identifying the relations between a *target* variable y , (which can be a scalar, a vector or a tensor), and a set of m *predictors* (x_1, x_2, \dots, x_m) . The purpose of supervised data mining tools is to build out of available data a formal relation $y = F(x_1, x_2, \dots, x_m)$ between the selected predictors which are identified as the most discriminant to explain and to model y . The most famous techniques which belong to this family are: oriented webs, segmentation by decision trees and neural networks. The second family of data mining tools is called the *non supervised* one. Its purpose is to characterize, inside a given population, homogeneous subgroups called clusters without any *a priori* on a potential target variable. The famous techniques that one can find in this family are: non oriented webs, topology by Kohonen maps, two steps cluster, etc.

The process of data mining actually consists of three steps:

1. data processing.
2. data modeling.
3. deployment.

Step 1: data processing. This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records. Then, depending on the nature of the problem, this first stage of the process of data mining may involve a simple choice of predictors for a regression model or need elaborate exploratory analyses using a wide variety of graphical and statistical methods in order to identify the most relevant variables that can be taken into account in the data modeling step.

Step 2: data modeling. This stage involves considering various models to identify the best one based on their predictive performance. It might seem like a simple operation but it sometimes involves a very complex and elaborate process. There are a lot of techniques developed to achieve that goal as we explain in the beginning of this section. The selection of the future operational model consists in applying the different potential models to the same data set and then comparing their performance to choose the best.

Step 3: deployment. That final step involves the model selected as the best in the previous stage to apply it on new data in order to generate predictions or estimates of the expected target.

Numerous books review the theory and practice of data mining; the following references are general books on data mining, representing a variety of approaches and perspectives [43–47].

Multi-objective analysis

Multi-objective analysis relies on optimization tools, but its aim is not to find an optimal set of parameters. The method aims at helping the study of computational models by experts. Lots of computational models are an assembly of different parts whose behavior may be clearly known in separation, but whose interactions might be difficult to grasp. Multi-objective analysis method aims first at generating a set of specific solutions: Pareto optimal solutions, and then study their features. The goal of the method is then to find features of a model and identify points or regions of the parameter space that require further studies. In this context, the optimization tools are used not to increase the efficiency of a particular parameterized model, but to extract some knowledge on the considered model through the analysis of specific behaviors. The automatic selection of such behaviors is a key feature of the proposed approach.

For a given parameterized model, the multi-objective analysis method aims at answering questions like:

- What are the objective values within reach?

- What relations are there between the parameters and the objectives?
- How critical is a particular parameter? Which parameter(s) make a solution Pareto optimal?
- Are there a continuum or a set of qualitatively distinct solutions? In this case, where are the transitions and what are the features of the families of solutions?
- What characterizes a particular behavior relative to the others?

Furthermore, such a method can also help comparing different models. Actually, when models can be evaluated on different criteria, one can be better than the other for some trade-offs, while the other may perform better in other parts of the Pareto optimal set. The comparison of Pareto optimal sets provides then a more rigorous view on the relative performance of the models.

The method consists in performing the following steps:

1. Selection of the search space;
2. Definition of objective functions;
3. Computation of the Pareto front (or at least an approximation of it);
4. Analysis of the results.

The first three steps are classical to anyone familiar with optimization tools. The last step is the most important one of this methodology. We will list tools that can support it and highlight some of the questions they can answer. The examples provided below just aim at highlighting what a simple analysis can reveal on a computational model. A complete analysis of each of these examples goes far beyond the scope of this article. Following examples won't then exploit all of the analysis tools mentioned below.

Selecting the search space

Choosing the search space is generally the most simple step. The model to be studied is known beforehand and generally described by a vector \mathbf{X} of m parameters: x_1, x_2, \dots, x_m . The search space is then \mathbb{R}^m if unbounded or a subset of it.

Defining objective functions

Objective functions, that are also called cost functions or fitness functions, are the functions to be optimized. The most critical point at this step is to define at least two antagonistic objective functions. If the two functions are linearly dependent, a single solution can be optimal for both objectives: the set of Pareto optimal solutions will then be reduced to a single solution, as for a mono-objective optimization, thus not allowing to perform the proposed analysis.

Actually, any function numerically describing model features is worth considering for this step. This choice is clearly of critical importance for the analysis to be the most useful, as all the performed analysis will rely on these particular objective functions. The objective values within reach clearly depend on these functions, so do the relations between objectives and parameters and the other considered questions (how critical is a parameter? are there singularities? etc). Clearly, this step requires a deep expertise and must be carefully defined and justified for the results to be of any use. Typical functions to optimize are, for instance, accuracy of results and speed of response to both maximize or energy to minimize and accuracy to maximize, etc.

Computing the set of Pareto optimal solutions

All further studies rely on the availability of the set of Pareto optimal solutions. In most cases, only an approximation of this set is within reach and the "true" set of Pareto optimal solutions is not known. It is then crucial to do whatever possible to guarantee the closeness of the approximated set to the real one. Furthermore, to get the best from the last step of the method, i.e. the analysis of the results, it is important to get a *dense* estimation of the Pareto optimal set in order to be confident in the fact that the analysis does not miss important features because of a too coarse sampling. It also opens the way to the study of regularities.

Multi-objective evolutionary algorithms (MOEA) are good tools to consider in this search as they can provide a dense estimation of the Pareto optimal set in a single optimization run [34]. NSGA-II, for instance, is an efficient and yet simple algorithm that is now widely used in this context [35]. Evolutionary algorithms are stochastic optimization tools. Their estimation of the Pareto optimal set may then vary from one run to the other. Few theoretical work is focused on the study of MOEA convergence [48, 49]. Most work in this field is based on empirical studies as theoretical results have often revealed to be of no practical use yet, as stated in [48]:

there is some indication that many multiobjective selection concepts used in practice may fail theoretically, it is possible that even such concepts without an almost sure convergence may work better in practice than those where (global) convergence can be proven.

For the INNOVIZATION approach, [29] suggest to first use a multi-objective evolutionary algorithm, e.g. NSGA-II. The *nadir* and ideal points are evaluated and a local search is used to get a better optimized front [34, 50]. Last, the normal constraint method (NCM) [51] is used, starting at a few locations to check the accuracy of the optimized front.

Considering that we only have access to approximations of the set of Pareto-optimal solutions and that they are generated by a stochastic process, we suggest another approach, aimed at exploiting at best all the available data:

1. repeat the optimization several times;
2. evaluate the disparity between the generated sets;
3. compute Ψ_0 , the set of non-dominated solutions over all results;
4. for each point of Ψ_0 , evaluate how close or how far it is to the closest points of other Pareto sets.

Ideally, several different algorithms can also be used for the first step. Its goal is to increase the confidence in the results with regards to the stochasticity of the process used to generate them. The number of runs to actually perform should be high enough to ensure the statistical significance of the results, but the required computing power will limit it in practice.

The second step aims at evaluating the disparity between the runs. A too wide variability in the results is the symptom of a problem – ill parameterized evolutionary algorithm, for instance : population size too small or too few generations – thus suggesting that the generated solutions might be still improved. It suggests then to go back to the first step and perform other optimizations with new algorithms parameter values (or test other algorithms). A low variability increases the confidence in the fact that the runs have converged to a locally optimal set of solutions¹.

The third step provides the best approximation of the set of Pareto optimal solutions relative to available data. Analysis of the shape and patterns of Pareto optimal solutions must then be performed directly on this set rather than on the result of a particular run.

The last step aims at more precisely evaluating the disparity within the front, whereas the second step studied the global variability. A huge variability for a particular part of the front is a symptom that

¹It doesn't tell anything regarding the distance towards the global Pareto-optimal set.

should be known by the expert performing the analysis. It is representative of parts of the search space that were more difficult to explore for whatever reasons (model instability, for instance) and it may be worth to further analyze these particular points to understand where the variability comes from, or it may also be worth knowing it in order to focus the analysis on points for which the convergence is less questionable.

The first and third steps are straightforward. The fourth needs a dedicated evaluation algorithm that will be described below. The second requires to evaluate a disparity between approximations of the set of Pareto-optimal solutions. In two or three dimensions, this can be evaluated visually by simply plotting the different fronts and looking at their differences. Making a quantitative comparison is not as simple as comparing results of single objective optimization algorithms. As it is required to compare multi-objective optimization algorithms, it has been the subject of intensive work over the last decade, see [52] for a review. Comparison of generated sets of solutions can be performed on the basis of dominance ranking or performance indicators.

Dominance ranking consists in evaluating if a particular set of solutions dominates another one, i.e. if the solutions of the considered set dominate each of the solutions of the other one. This method is aimed at ranking optimization algorithms, not at evaluating the disparity of the generated solutions and thus won't be used here.

Multiple performance indicators have been proposed to describe with a single scalar the performance of a multi-objective algorithm [53]. They are mainly aimed at comparing algorithms, but also at guiding the search or at defining a stopping criterion. To evaluate the disparity between the different runs, we suggest to use the *hypervolume indicator*, that has the drawback of an exponential computational cost relative to the number of objectives, but with the advantage of being strictly monotonic (see section *Indicators* for its definition).

To evaluate the disparity between runs, we propose to compute the following values:

- For each of the performed run $r \in \Gamma$, identify the nadir objective vector η^r
- Discard the outliers: consider the conservative nadir point $\bar{\eta}$ defined as follows:

$$\bar{\eta} = \{\max_{r \in \Gamma}(\eta_i^r), i = 1, 2, \dots, m\}$$

The disparity between the generated sets is then evaluated as follows:

1. compute Ψ_0 and Ψ_1 ;
2. compute the difference of the hypervolumes of Ψ_0 and Ψ_1 relative to the conservative nadir point: $I_H^{\bar{\eta}}(\Psi_0)$ and $I_H^{\bar{\eta}}(\Psi_1)$ to evaluate the variability of the runs.

For the evaluation of the disparity within Ψ_0 , we suggest to compute, for each point of Ψ_0 , the distance to the closest points of each of the different Pareto sets Ψ_r and keep the maximum value. This value may either be used visually or, with a distance threshold, used to define a conservative set $\Psi_{cons} \subset \Psi_0$ defined as the set of points $p \in \Psi_0$ for which the distance is below the chosen threshold.

To sum up this part, the evaluation of the Pareto front we suggest is the following:

1. repeat the optimization several times, if possible with several different algorithms
2. evaluate the disparity between the generated sets:
 - (a) compute Ψ_0 and Ψ_1
 - (b) compute their respective hypervolumes relative to $\bar{\eta}$
 - (c) if the relative difference is above a given threshold, go back to step 1 with different parameter values or with other optimization algorithms
3. for each point of Ψ_0 , evaluate the maximum distance towards each Ψ_r .

Analysis of generated data

What is available after the former step is an approximation of the Pareto-optimal set of solutions. Each point associates a particular set of parameter values \mathbf{X} together with its corresponding objective values $f(\mathbf{X})$. Analyses proposed in the following aim at extracting knowledge from it. For each analysis, some typical questions the analysis might contribute to answer are listed.

Analysis of the Pareto set

Typical questions:

- What are the objective values within reach?
- What are the relations between objectives?
- Is the model ill-posed?

A first analysis focuses on the set of points in the objective space, i.e. in the space of $f(\mathbf{X})$ values. The lower and higher values for each objective as well as the shape of the set may be of interest for the information they provide on the computational model and on the set of contexts and parameters opened to the optimization. The shape of the front may highlight objective relations and an irregular shape may even be the symptom of an ill-posed model [18].

Analyzing the Pareto set implies to draw at least a part of it. In two or three dimensions, as for the examples presented in the following, it can be directly plotted, but for higher dimensions, dedicated visualization tools have been developed, like the scatter plot matrix, the value path or the star coordinates methods [34]. Subsequent analysis may rely on a direct observation of the relations between objectives, or on regression tools to identify it more formally.

Analysis of the parameters

Typical question:

- What are the relations between model parameters and objectives?

Each point of the Pareto set corresponds to a particular set of model parameters. Looking at how these parameters change may highlight important features of the model, like parameter uncertainties [17].

In the examples provided below, a particular parameter will be plotted against one or two objectives, thus not requiring complex visualization tools. Other analysis might need a global picture of the parameter set relation to objectives. Dedicated tools to project it in a two or three dimensional space are then required, like Kohonen maps [54], for instance.

Analysis of solutions

Typical questions:

- What are the typical behaviors of the model?
- What are the features of a particular solution or of a cluster of solutions?

- What are the most critical parameters?

As a Pareto set may contain a lot of different points and before undertaking deeper studies, it may be important to build clusters of solutions in order to identify the most salient behaviors. In some experiments, as in the second example presented below, the definition of such clusters may directly stem from expert knowledge. In other cases, automatic clustering methods can be used to detect groups of similar solutions on the Pareto front. The most commonly used algorithms to accomplish this task are k-means [55], fuzzy c-means [55], principal component analysis, hierarchical clustering [56] and Kohonen maps [57, 58].

Once clusters of solutions have been identified, the proposed analysis consists in looking at which parameters better describe a cluster relative to other clusters. Understanding what makes a solution Pareto-optimal or what is specific of a particular cluster of solutions may highlight the relative importance of each parameter. To drive this analysis, classification algorithms can automatically extract the most descriptive cluster features [59]. The results of such classifications is a decision tree. Starting from the whole set of considered points, the tree aims at subdividing it according to variable ranges. Each internal node is associated to a variable and the branches coming out of it are labeled with ranges of values. Each path from the root node to a particular node is then descriptive of a set of points and the goal of decision tree algorithms is to find the description sets that discriminate best between two given set of points. When the study focuses on a particular cluster, it will highlight the main features of this cluster relative to other clusters (may it be from the Pareto set or from the set of explored points).

Further analyses may focus on a specific solution. While expert knowledge can be used to highlight a particular point, simple selection rules have also been proposed, like the one of [60], that ranks Pareto optimal solutions by using a graphical method based on L^p , $p = 1, 2, \infty$ norms of the normalized vector of objectives. The chosen solutions are those that minimize the distance, measured in an L^p -norm, to the ideal point. The behavior of the chosen point can then be studied while relying on expert knowledge.

Example from Aerodynamics: study of a flapping wing aircraft

The first example concerns the study of a flapping wing aircraft. Although the underlying physics is locally well understood, the impact of wing beats on aerodynamical forces and the question of what wing beats to generate to get a particular behavior of the aircraft is still a subject of intense research. The multi-objective analysis is used here to generate a set of wing beats to study. As few experimental data are available, the results will be related to what biologists did observe on birds of similar size. Biologists have identified relations between significant parameters like wing area, cruising speed, wing span, flapping wing frequency, wing load or mass [61]. Wing kinematics have also been studied for several species [62–64]. The goal of the multi-objective analysis is to identify some specific wing beats to be tested in an experimental device. The choice of the wing beats is driven by the need to enhance the model or more generally the understanding of the underlying physical phenomena.

Description of the model

Aerodynamical forces created by wing movements are computed with a semi-empiric, quasi-steady-aerodynamics model. Each wing is decomposed in three panels. For each panel, the local incident airspeed is evaluated. The computation of aerodynamic forces has been extended over classical lift computational models to better evaluate the forces at high incidence angles, see [65, 66] for a detailed description. The panels are considered as non deformable solids, connected to their neighbors via joints, as shown on figure 2. The integration of the forces and the movements of the parts are computed using ODE software library².

²<http://ode.org>

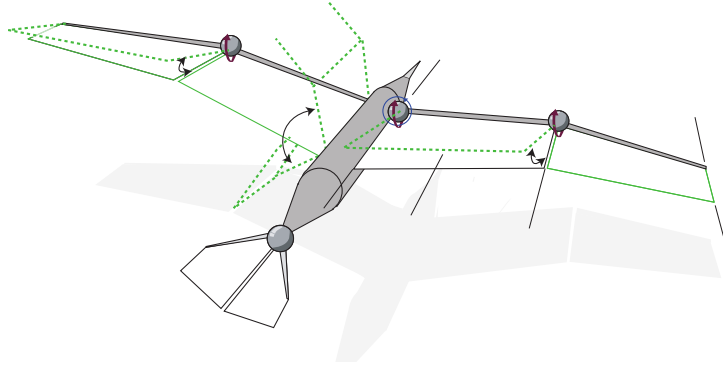


Figure 2. Degrees-of-freedom of the simulated flapping wing aircraft.

Wing kinematics are described by the following equations:

$$DI = a_{DI} \sin(2\pi t/p_{DI}) \quad (1)$$

$$TW_i = r_{TW_i} + a_{TW_i} \sin(2\pi(t/p_{DI} + p_{TW_i})) \quad (2)$$

$$TW_e = r_{TW_e} + a_{TW_e} \sin(2\pi(t/p_{DI} + p_{TW_e})) \quad (3)$$

where DI is the wing dihedral, TW_i the internal twist and TW_e the external twist. Wing kinematics are then described by eight parameters subject to the evolutionary optimization. The chosen ranges for each value are the following:

- amplitude of the dihedral: a_{DI} in $[0; 45^\circ]$
- period of the dihedral (and of all other degrees of freedom): p_{DI} in $[0.2; 1s]$
- reference of the internal twist: r_{TW_i} in $[-22.5; 22.5^\circ]$
- amplitude of the internal twist: a_{TW_i} in $[0; 45^\circ]$
- phase of the internal twist: p_{TW_i} in $[0; 1]$
- reference of the external twist: r_{TW_e} in $[-22.5; 22.5^\circ]$
- amplitude of the external twist: a_{TW_e} in $[0; 45^\circ]$
- phase of the external twist: p_{TW_e} in $[0; 1]$

Objectives to optimize

The energy-speed relation is characteristic of a particular shape and of a particular wing kinematic. We will then choose to optimize both speed and energy to empirically evaluate this relation. As we are interested in low as well as high speeds, we will perform two sets of experiments, one in which the speed is minimized and one in which the speed is maximized.

The objectives to maximize are the following:

- experiment 1:
 - $+1 \times$ average aircraft speed

- $-1\times$ average mechanical power
- experiment 2:
 - $-1\times$ average aircraft speed
 - $-1\times$ average mechanical power

Following [65], the instantaneous mechanical power is computed as the scalar product between the instantaneous torque (τ) and the instantaneous rotational speed (ω) for a joint: $P = \sum_i |\tau_i \cdot \omega_i|$. The mechanical power objective only takes into account the shoulder joints which are the main contributors to energy consumption. It should be noted that the power is always considered as positive. The mechanical power is then over estimated, as the torques required to accelerate or to slow down are considered as equivalent.

NSGA-II is used to perform the search, with a population size of 500 and during 1000 generations. Evolved parameters are represented as vectors of real values with a polynomial mutation and a sbx crossover, as described in [34], p124.

Results

Four different runs have been performed for each of the two setups.

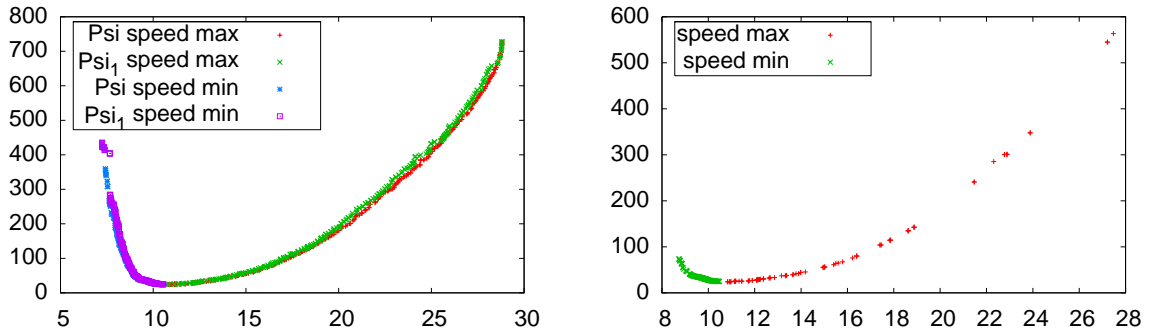


Figure 3. Left: Ψ_0 and Ψ_1 for the experiments on the flapping wing aircraft. "speed min" corresponds to the speed minimization experiments and "speed max" correspond to the speed maximization experiment. X-axis is the speed in $m.s^{-1}$, Y-axis is the energy in W . Right: Filtered version of Ψ_0 for the experiments on the flapping wing aircraft. Plotted points are those whose distance towards the Pareto fronts of other runs is lower than a chosen threshold ϵ (here $\epsilon = 1$)

Ψ_0 and Ψ_1 have been computed and plotted on figure 3, left. The conservative hypervolumes are respectively $I_p^H(\Psi_0) = 414.9$ and $I_p^H(\Psi_1) = 393.4$ for the speed minimization experiment and $I_p^H(\Psi_0) = 6267.4$ and $I_p^H(\Psi_1) = 6117.3$ for the speed maximization. The difference is then 5.2% of $I_p^H(\Psi_0)$ for speed minimization and 2.4% of $I_p^H(\Psi_0)$ for speed maximization. The disparity is then low, but not negligible, especially for the speed minimization case. The plot of the filtered version of Ψ_0 (figure 3, right), highlights that the differences are concentrated on the lowest speeds and, in a smaller extent, also on the highest speeds. This disparity reveals that the search was more difficult for these particular speeds.

In the following, we consider the points from Ψ_0 and study their features from an aerodynamical point of view. The ornithopter is characterized by a wingspan b of 1.93 m (both wings), a wing area S of 0.407 m^2 , a mean chord c_m of 0.2 and a variable cruise velocity U . The air kinematical viscosity ν is about

$15 \cdot 10^{-6}$. To analyze the Pareto optimal solutions, the following aerodynamical dimensionless numbers are defined:

- Reynolds number: $Re = \frac{Uc}{\nu}$
- Strouhal number: $St = \frac{\sin(\frac{\pi \alpha_{Dl}}{180})b}{U p_{DI}}$
- Reduced frequency: $k = \frac{\pi c}{U p_{DI}}$
- Reduced internal twist frequency: $\kappa_1 = 2|\frac{\pi \alpha_{TWi}}{180}|k$
- Reduced external twist frequency: $\kappa_2 = 2|\frac{\pi \alpha_{TWe}}{180}|k$

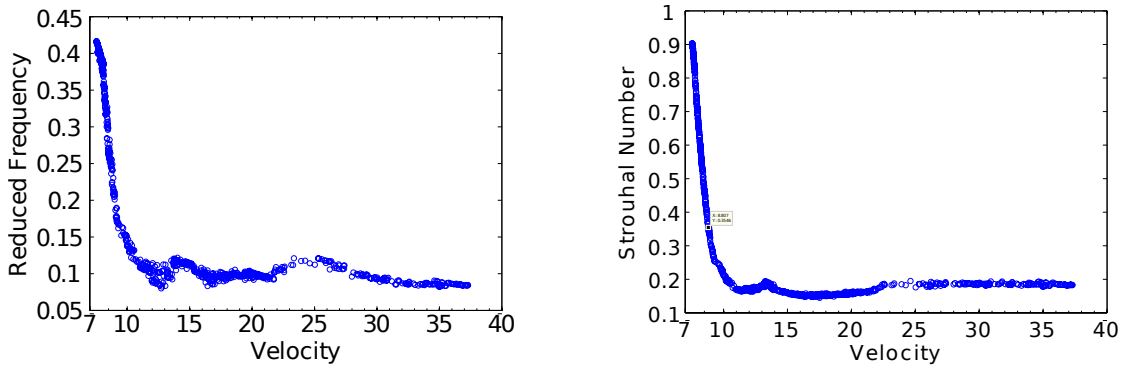


Figure 4. Left: Reduced frequency versus the cruise Velocity. Right: Strouhal number versus the cruise Velocity

If we look at the reduced frequency (see Figure 4, left), we can see that for velocities inferior to $8.7 \text{ m}\cdot\text{s}^{-1}$ the reduced frequency varies between 0.25 and 0.42. These values are not standard for birds for which the reduced frequency oscillates between 0.1 and 0.25 [67]. Thus, unusual high frequencies are experienced by our ornithopter at low speeds. These high frequencies may explain the optimization problems for low speeds, as they may result in an instability of the physics simulation. At higher speeds the reported reduced frequencies agree with the usual values encountered for birds.

Moreover, the Strouhal number (see Figure 4, right) oscillates between 0.35 and 0.9 when the cruise velocity varies from 7 to $8.8 \text{ m}\cdot\text{s}^{-1}$, which is not common for birds ($St \sim 0.15\text{-}0.35$) [68]. This indicates that unusual high dihedral velocities are experienced by our ornithopter at low speeds. For higher velocities the Strouhal numbers agree with the usual values reported for birds.

In addition to that if we look at the reduced internal and external frequencies (see Figures 5 left and right), we notice that they take high values at low speeds, the external reduced frequency κ_2 being slightly higher than the internal one κ_1 . This means that our ornithopter experiences high twist velocities at low speeds.

Furthermore, if we take a look at r_{TWi} and r_{TWe} (see Figure 6), we can see that the wing is highly twisted at low speeds which is characteristic of the flight of birds at these speeds [62]. At higher speeds the wing's twist decreases as expected [62].

To sum up the previous statements, we can say that low speeds are characterized by both high twist and dihedral velocities that can be attributed to the generation of the necessary lift at low speeds [69] while the high frequencies experienced are directly related to the generation of the power for flight [63].

However, the reported values of Strouhal number and reduced frequencies for our ornithopter are quite high at these speeds. At this stage, to explain these values different hypotheses can be considered:

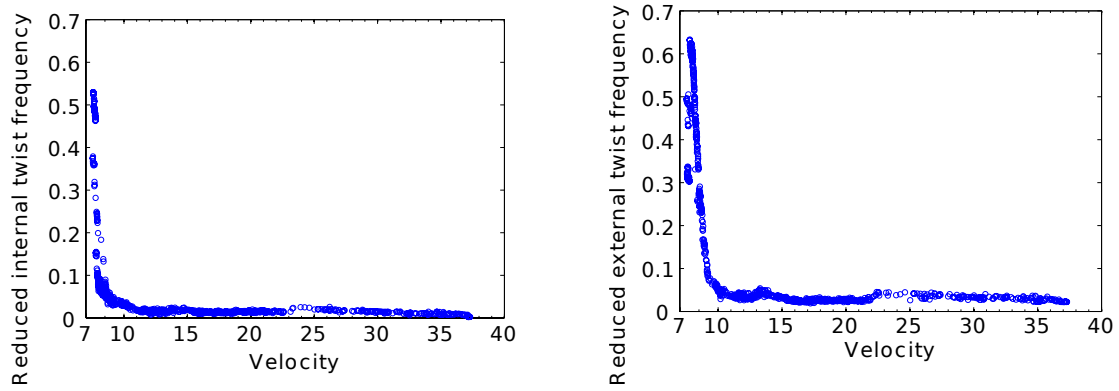


Figure 5. Left: Reduced internal frequency versus the cruise Velocity. Right: Reduced external frequency versus the cruise Velocity

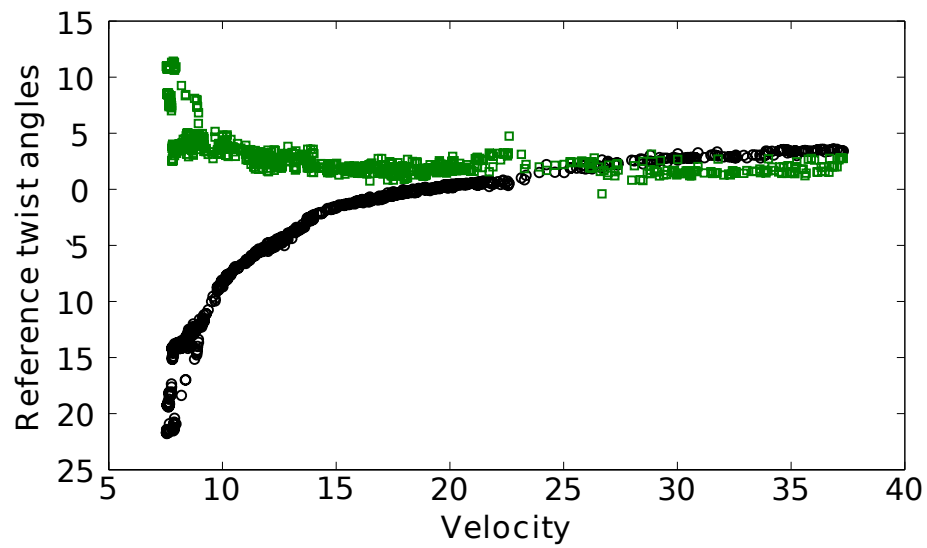


Figure 6. Reference twist of internal (\circ) and external (\square) panels versus the cruise Velocity

- the accuracy of the aerodynamical model [65,66] at low speeds is low;
- the convergence of the optimization process at low speeds is too poor;
- the ornithopter's configuration is not compatible with low speeds.

Concerning the aerodynamical model, if we compute the Reynolds number on the Pareto front we can see that it varies between 10^5 and $5 \cdot 10^5$ while the velocity fluctuates between 7.5 and 37.3. At these Reynolds numbers, commonly experienced by birds similar to our ornithopter [70], transitional flows may occur [67] especially at low speeds. Thus, one has to make sure that the aerodynamical model can handle with reasonable accuracy low Reynolds number effects by comparing the model with experimental results or CFD³ simulations. The kinematics associated to points in the low speed part of the Pareto front can then be used on a real experimental device in a wind tunnel or with CFD simulations to test the accuracy of the model and more precisely focus a study on the phenomena that the proposed model might neglect at this speed.

The most interesting solutions from an energetic consumption point of view are those which minimize the energy. A characterization of these solutions versus the other solutions by a binary decision tree [57,71] allows to identify the kinematic parameters that are important to define these interesting solutions.

A neighbourhood of the solution of minimal energy was defined by taking all the solutions such that the relative variation on the energy is less than 5 per cent. Then, a description of this neighbourhood relative to other Pareto optimal solutions was sought using Classification and Regression Tree algorithm (CART). It highlighted one significant rule:

$$0.48 \leq p_{DI} \text{ and } r_{TW_i} \leq -5.85 \text{ and } 8.5 \leq a_{TW_e} \quad (4)$$

The obtained decision tree is 93 % accurate and ranks the kinematical variables appearing in (4) by order of importance. Thereby, p_{DI} is more important than r_{TW_i} which is more important than a_{TW_e} to define the minimal energy solutions. This result is qualitatively acceptable from an aerodynamical point of view. Indeed the flapping period, the internal twist reference and the internal twist amplitude play an important role in the aerodynamics of our ornithopter [62,71]. Anyway, as there is no previous work on the subject, the only way to validate it is to launch the same experiments in a windtunnel and compare the obtained minimal energy solutions.

Furthermore, the precedent result gives us a compact vision of the kinematical model for the minimum energy solutions. For these solutions, the decision tree suggests that the important parameters for the model are the flapping period, the internal twist reference and the external twist amplitude. A possible way to check this result *a posteriori* is to build a reduced model of the kinematics with just the three preceding kinematic parameters, recompute a Pareto front in the conditions stated before and compare the minimal energy solutions obtained.

Summary

We have optimized the parameters of the wing kinematics of a simulated flapping wing aircraft. To empirically generate its speed-energy relation, we used both speed and energy as objectives to optimize. Key steps and main results are listed below:

- At first, we evaluated the convergence of the optimization runs and highlighted a convergence problem in the low speed part;
- we then performed an aerodynamical study of the Pareto optimal solutions. For speeds higher than $\sim 9 \text{ m.s}^{-1}$, these solutions are characterized by realistic values of Strouhal number, reduced frequency and Reynolds number which comforts us in the validity of these solutions and thus of the model at these speeds;

³Computational Fluid Dynamics

- low speeds Pareto optimal solutions are characterized by highly twisted wings, high frequencies, high dihedral and twist velocities inherent to the flight at these speeds but the order of magnitude of the reported values is not realistic which shows that further investigations are needed. Three potential hypotheses may explain the high values encountered at low speeds: the inaccuracy of the aerodynamical model, the failure of the optimization process or the incompatibility of the ornithopter’s geometry and/or kinematics with these speeds;
- the minimum energy solutions are characterized by three kinematical parameters: the flapping period, the internal twist reference and the external twist amplitude. This result gives us a compact vision or reduced order model of the kinematical model for these solutions.

Example from Neuroscience: study of a Basal Ganglia model

The Basal Ganglia (BG) is a set of interconnected brain nuclei. Located under the cortex, the BG is present in all the vertebrates, its structure remaining largely identical throughout the species. In order to gain a better understanding on the mechanisms governing it, various neuro-computational models have been proposed (for a review, see [72]). Among them are the recent CBG model [1] and the more classical GPR model [2]; the parameters of both of them have been evolved with MOEA in a previous work [73]. We summarize here the results obtained and we push further the multi-objective analysis. This analysis will focus in particular on the relative importance of connections between nuclei for each model, and does provide a way to compare the two models.

Description of the models

The considered models of the BG are made of rate-coding artificial neurons (either leaky-integrators in the GPR or locally-projected dynamical systems in the CBG), representing the average activity of a population of real neurons. These models are basically described by a first order differential equation combined with a transfer function or a projection operator. They are represented by circles in figure 7. These neurons are grouped in nuclei which have been anatomically identified (boxes in figure 7). The main components of these models are the definition of the graph of connections between the nuclei, the weights of these connections and some internal properties for each nucleus. The architectures of the CBG and GPR models are respectively shown in the right and left panels of figure 7. The slight differences between them consist in the adjonction of connections in the CBG that were unknown or not included at the time when the GPR model was elaborated. The strengths of the connections and the internal nucleus properties account for 25 parameters for the CBG, and 20 for the GPR. Each connection parameter is to be evolved in the range $[0.05, 1]$, as setting a connection weight to zero would be equivalent to delete this connection. For more details on the whereabouts of each models, we refer to the original articles [1,2] or to our previous work [73].

The biological knowledge used to constrain the weights of the connections and the internal properties of the nuclei is rather weak, hence the modeler has to hand-tune almost every parameter to obtain a satisfying model. The MOEA were used to evolve the parameters while sticking to the architectures of the CBG and GPR models, in an attempt to shed light on the role of the various parameters and to compare the two architectures.

Objectives to optimize

According to the mainstream hypothesis, the BG operates a generic selection system [74]. The corresponding computational setup postulates the existence of a certain number of *channels* corresponding to possible alternatives; these are selected by disinhibition: the outputs of the channels are inhibitory and active by default, thus inhibiting their targets, when a channel is selected, its output is decreased, and

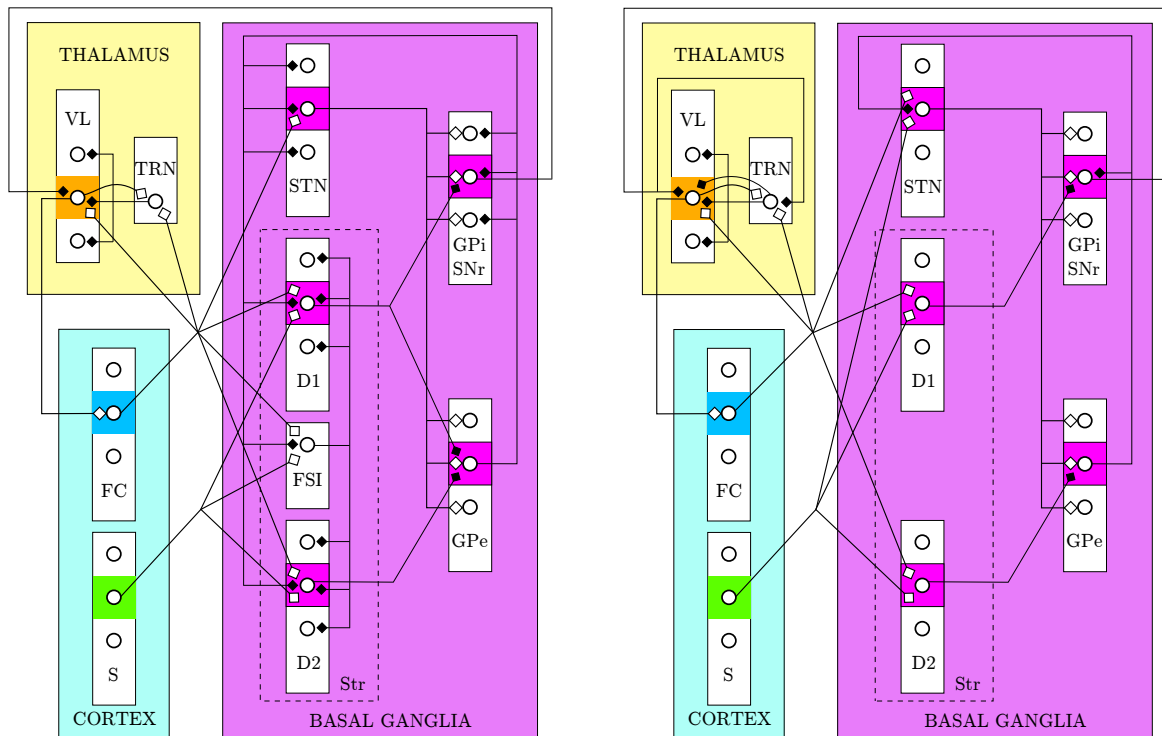


Figure 7. Schemes of the CBG (left) and GPR (right) architectures, representing nuclei from the Cortex, Thalamus and Basal Ganglia with labeled boxes. The inputs are the Saliences (S) in the cortex, and the outputs are the activity of the GPi/SNr neurons. Three channels in competition are represented in most nuclei. The outgoing connections of the shaded channel are the only one shown. White endings are excitatory, black are inhibitory.

thus its inhibitory effect canceled. The computational hypothesis is then formulated as a winner-takes-all among competitive channels, assigning the minimum activity to the channel with the highest input, while the others are activated as much as possible. This simple model of BG behavior will be used as a basis of the objective functions definition.

To ensure that the evolved models behave as a winner-takes-all, $N = 500$ random input vectors have been submitted to each individual, drawn in $[0, 1]$ from a uniform distribution. Two objectives were to be minimized. First, the channel corresponding to the largest input (the selected channel "sc") was to have minimal activity. Hence, the first fitness function is :

$$f_1 = \frac{\sum_N GPi_{sc}}{N}$$

But this is not sufficient to obtain a WTA algorithm, as this could lead to the complete disinhibition of every channel. Therefore, a second objective was defined as the mean of all the other channels, the *not selected* channels "nsc", of cardinal "#(nsc)" :

$$f_2 = 1 - \frac{\sum_N \left(\frac{\sum GPi_{nsc}}{\#(nsc)} \right)}{N}$$

Any set of parameters minimizing these two objectives implements a WTA algorithm. Regardless of the inputs, a model with the fitnesses ($f_1 = 1, f_2 = 0$) always selects all the channels, and a model with ($f_1 = 0, f_2 = 1$) never selects any channel. All the other values for (f_1, f_2) represent different approximations of a WTA.

Results

Ten runs have been first launched with a CBG architecture and their results compared and analyzed in details; ten runs with a GPR architecture were then computed for comparison purposes (see the comparison between the two architectures below). Ψ_0 and Ψ_1 have been computed and plotted on figure 8, left, for the CBG runs. Their conservative hypervolumes are respectively $I_p^H(\Psi_0) = 0.593$ and $I_p^H(\Psi_1) = 0.588$. The difference between the two is very low (0.7% of $I_p^H(\Psi_0)$). The confidence in the convergence of the runs is then high as the different runs have generated noticeably similar results. This is confirmed by the plot of figure 8 (right) that shows a dense representation of the Pareto front with no particular gap, contrary to the previous experiment.

Interpretations of the entire set of solutions

Some connection weights are minimized or maximized for the whole Pareto front. This result can be interpreted as reflecting the relative contribution of the connections for the objectives. Hence, connections that are maximized at the upper bound of the search space, are of utmost importance for the selection. They reflect the parts of the circuit that are essential to achieve some kind of selection.

On the contrary, some connection parameters are minimized. Excluding the hypothesis that the mathematical formalism or the parameters search space are not adequate in modeling the reality, this means that either (1) these connections are indeed very weak, or even non-existent; or (2) these connections are useful for another purpose that is not expressed by the objectives.

Other parameters have been found to be set at random, this can be quantified with a very low measure of auto-correlation (see figure 9 for an example of a rather precisely set versus a randomly set parameter). The interpretation for these connections is that they are indifferent to the objective functions, neither contributing nor being against the supposed function.

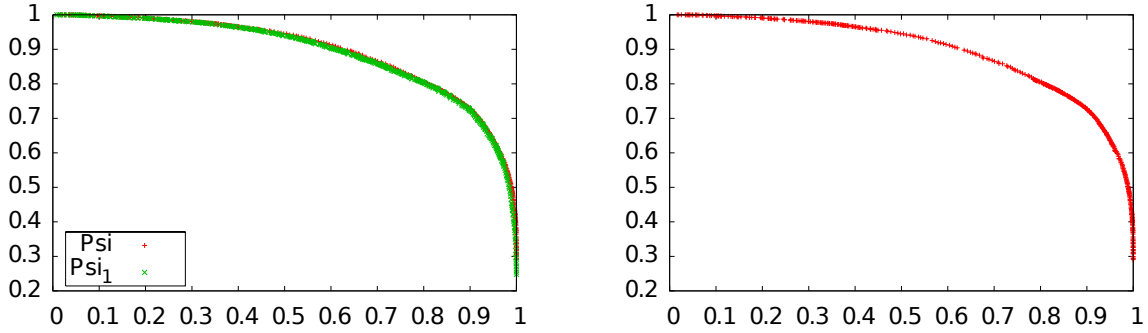


Figure 8. For both graphs, X and Y axis are the f_1 and f_2 fitnesses.

Left: Ψ_0 and Ψ_1 for the experiments on the basal ganglia model.

Right: Filtered version of Ψ_0 for the experiments on the basal ganglia model. Plotted points are those whose distance towards the Pareto fronts of other runs is lower than a chosen threshold ϵ (here $\epsilon = 0.01$).

We provide here only the guidelines for interpretations of parameters at the bounds or randomly set, as the biological implications for the specific parameters of the Basal Ganglia evolutions have been discussed in details elsewhere [73].

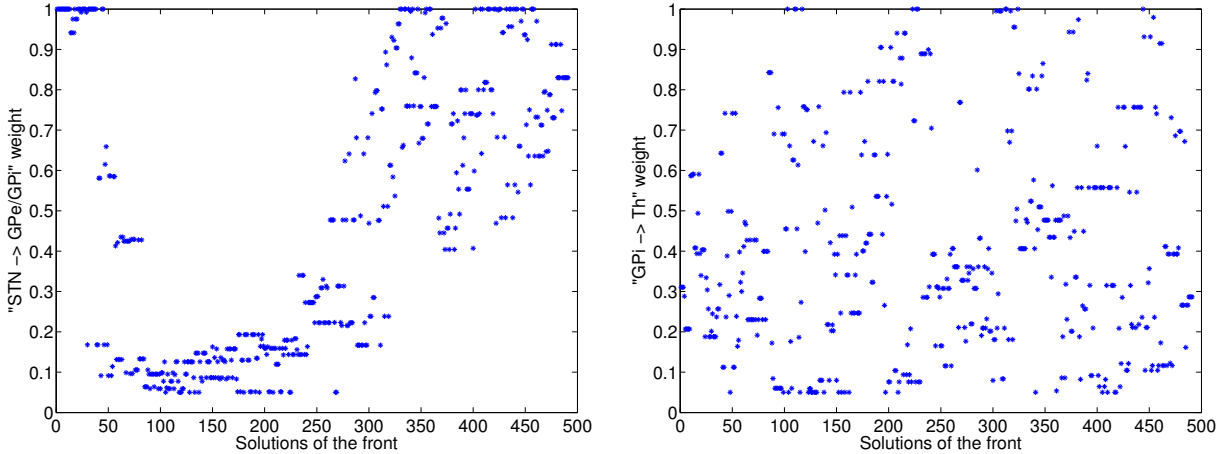


Figure 9. Plot of the STN \rightarrow GPe/GPi (left) and the GPi \rightarrow VL (right) connection weight parameters for each solution of the Pareto front. The X-Axis lists the solutions along the front, beginning with those who rank best for the first objective and ending with those who rank worst the first objective. While the STN \rightarrow GPe/GPi appears to be set rather precisely by the evolution, the GPi \rightarrow VL is apparently set at random.

Choice and analysis of a particular set of solutions

The whole set of solutions implements different kinds of selection, but some of these trade-offs are probably not interesting solutions. This is especially true for the solutions located at the endings of the pareto front, which tend either to never select any channel or to always select all the channels. Hence there is a need to choose a subset of solutions on the basis of expert knowledge. The comparative analysis of these

solutions relative to the whole set of solutions aims to bring a characterization of the most interesting models.

We began by evaluating the compatibility of the solutions with known biological data. To do this, we exploited two known facts. First, in the absence of input, the GPi, which is the output nucleus of the Basal Ganglia, is active. Its firing rate at rest, called the *base level*, should then correspond to a rather high value. Second, in a competition situation, the unselected channels should have an output higher than the base level, but the selected channel should have a lower output. By plotting the base level along the output of the selected channel as well as the mean output of the other channels, we have been able to identify a biologically plausible zone of the front (figure 10).

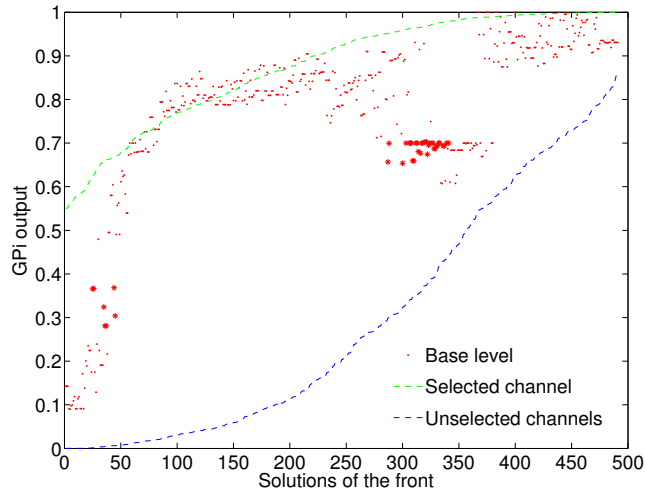


Figure 10. Plot of the output levels of each solution. The blue line corresponds to the mean output of the selected channel, and should be minimized according to objective 1. The green line corresponds to the mean outputs of the other channels, and should be maximized according to objective 2. The red dots correspond to the base level, i.e. the output in the absence of input. The dots corresponding to the most biologically plausible solutions (see text) are emphasized.

Another way to evaluate the quality of the selection is with an engineering approach. We elaborated a measure of the ability to discriminate between the two best channels in input, by looking at how different their outputs are. Indeed, for many of the obtained solutions, when the channel with the highest input and its closest competitor are very close, both of them are selected simultaneously. To evaluate this, we scored the percentage of random inputs that led to the selection of two channels, i.e. that led to the case where the two best channels in outputs are closer than an arbitrary value of 0.01. The results are shown on figure 11, they correspond roughly to the solutions we obtained with the previous criterion (figure 10). This shows that the solutions operating a biologically plausible selection are also amongst the most efficient ones.

By considering the subset of solutions efficient for both criterions, we can now analyze what makes them different from the other solutions, with the help of decision tree algorithms. To prepare this analysis, we began by normalizing the numbers of the solutions for each state by replicating the plausible solutions seven times, leading to a population comprising roughly as much plausible solutions (497) as non-plausible ones (501). After this, we applied a standard Classification And Regression Tree (CART) algorithm to separate the plausible "P" solutions from the non-plausible "NP" solutions (figure 12).

This decision tree shows that to ensure a biologically plausible selection, the $GPe \rightarrow D1$ connection has to be superior than 0.63. This is relevant to a biological interpretation, because the $GPe \rightarrow D1$ connection is a subject of debate. Indeed, the connection from GPe to Striatum has been acknowledged for a long

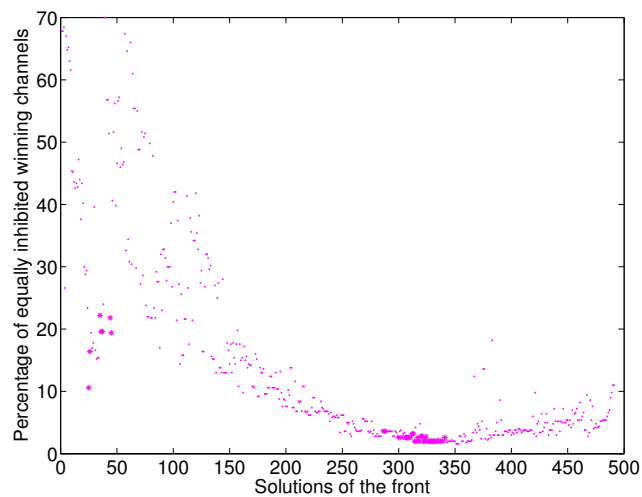


Figure 11. Plot of percentage of dual selection of the two best channels in input. The solutions labeled previously as "biologically plausible" are emphasized.

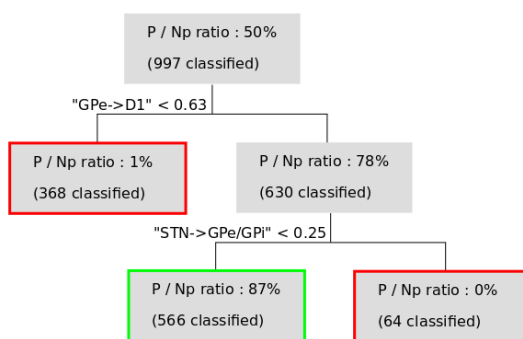


Figure 12. Binary decision tree representing the best parameters to discriminate between biologically plausible solutions ("P" solutions) and non-biologically plausible ones ("Np"). See text for methodology.

time in various species [75,76]. Anatomically, [77] showed that some GPe neurons targeted preferentially Fast Spiking Interneurons (FSI), with approximately half of their axonal terminaisons ending on FSI and the remaining terminaisons ending on other neurons of the Striatum. Electrophysiologically, [78] showed recently an inverse correlation in vivo between the GPe neurons firing rate and FSI firing rate, which is coherent with the idea that the GPe target them with an inhibitory influence. While these data suggest a strong influence of GPe neurons on FSI, they do not mean that their influence on the D1 and D2 neurons is weak or non-existent, and this connection has indeed been exploited in computational modeling work [1]. The present result suggests that this connection should not only exist, but it should also be potent.

The percentage of solutions classified by the other branches of the tree is far less important, hence we have to be cautious in the interpretation of these. Furthermore, this would require a more detailed biological analysis, going out of the scope of this paper, so we choose not to continue further this analysis.

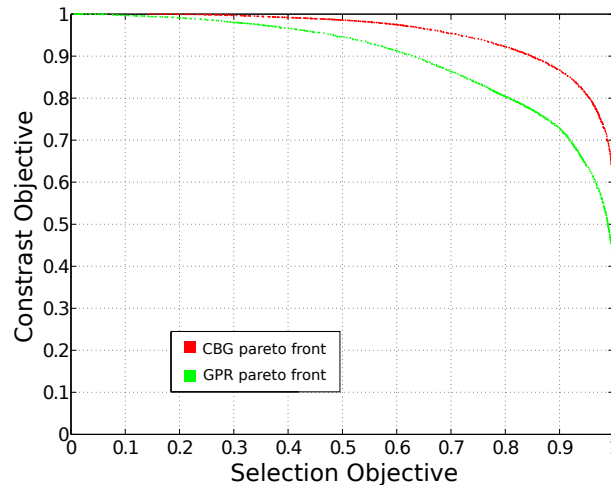


Figure 13. The fitnesses of the best fronts of the CBG (red) and GPR (green) fronts permit to compare the two architectures.

Comparison between different architectures

The CBG model has been built upon another model, called the "GPR" model [2]. The main difference between the two models is that the CBG takes into account biological data that were unknown at the time of the elaboration of the GPR. This results in a slightly different connection scheme (figure 7).

To evaluate whether the CBG or GPR architecture is more suitable to operate a selection process, the evolution of the GPR architecture has been done with a setup similar to the one done for the CBG. The parameters, which are fewer in the GPR (20 in place of 25), have been evolved under the same constraints. All the runs have converged toward the same approximation of the Pareto front, as they did for the CBG. Results showed that the solutions of the CBG strictly dominated the solutions of the GPR for the whole front (figure 13). For any solution from the GPR front, there is a solution from the CBG front that has a better score for both objectives, resulting in a better selection as we defined it. This hence leads to the conclusion that the changes in connectivity, as well as the additional degree of freedom of the CBG, globally enhanced the selection functionality. This is particularly interesting as, to the extent of our knowledge, there is no other way to assess whether one architecture or another is more suitable to exhibit a selection process.

Summary

We evolved the "CBG" architecture of the Basal Ganglia, to make it operate a selection amongst its entries. To this aim, we defined two antagonist objectives scoring the quality of the selection done on a given set of inputs. We list below the key steps and results of the multi-objective analysis of the evolution:

- Before interpreting the results of the evolution, we made sure that they converged to a good approximation of the Pareto-optimal solutions.
- We analysed the whole set of optimal solutions. Some parameters are either maximized, minimized, or set of at random. We provided guidelines for interpreting the functional signification of these parameters.
- We then analyzed the specific set of interesting solutions, after having characterized it by two criterions. By using a binary regression tree, we are able to make a prediction concerning a specific connection, the GPe \rightarrow D1 connection. This connection has a key role in the selection operated, and should be set at a rather high value. This result is particularly interesting as, even if the existence of this connection has been known for a long time, it has rarely been used in modeling works.
- Finally, we did optimize parameters of another architecture which is named "GPR", with the same setup as for the CBG. This showed that the solutions of the CBG dominated the solutions of the GPR, resulting in a better selection ability. This comparison highlights the relevance of the additional connections of the CBG, and provides an unique mean to benchmark two different architectures.

Discussion

The proposed framework listed only the most frequently used data mining algorithms. Other tools can be used depending on the need. Likewise, the examples are provided to illustrate the potential of the approach. Deeper analyses would require to go deeper in the respective fields of these models, what goes beyond the scope of this article. From a methodological point of view, a good practice would be to use this framework to first check in a systematic way a set of well established knowledge from the literature. Further work should then focus on the discrepancies that highlight model errors or incompleteness. A complete agreement to the literature would be a first validation of the model. In a second step, hypotheses currently made can be tested this way by checking their compatibility with the generated data. Although it does not represent a validation as strong as direct observations of the modeled phenomenon, it might help when such observations are not possible or when their cost is so high that a lot of evidence is required to justify such experiments. Although it is somewhat reassuring to follow a systematic path guided by a well defined goal, such analyses is probably also very valuable when little is known in the literature. It allows to propose new hypotheses and then also new experiments to be done on the modeled phenomenon to validate them.

We have proposed different ways to check whether the set of optimization runs did converge or not. The goal is to reduce the impact of the stochastic aspect of evolutionary algorithms. A too wide variability indicates that the run did not converge, suggesting that other runs with different algorithm parameters should be performed. Likewise, a finer analysis of the disparity between the runs may reveal area of the objective space for which the optimization was more difficult. Although it is not possible to give a simple and systematic interpretation of this symptom, one of the possibilities is model instabilities and identifying such area of instability may be of interest for the expert performing the analysis. One may wonder about the consequence of the lack of convergence proof on the potential interest of the method. Despite all efforts put into the verifications mentioned before, the convergence to the global optimum will never be sure. Is it a problem for the proposed methodology? Actually, the goal of the multi-objective

analysis is to find a set of "interesting" solutions to study. No matter how such points are generated, they are nothing more than specific parameter sets of the model. It means that, even if they are not optimal, they are representative of some model behaviors and are then worth studying. The only risk associated to the non convergence towards the global Pareto-optimal set, is that there might be other behaviors that the analysis will miss. The consequence is that the conclusions of the analysis should rely on the observed behaviors only. Without other hypothesis on model compatibility with optimization, nothing should be deduced about the capacity of the model to generate a non observed behavior, as this absence may stem from a premature convergence towards a local optimum and not from a particular failure of the model.

Likewise, the proposed comparison between models evaluates at the same time the capacity for the model to optimize the given objectives, but also its compatibility with the optimization algorithm. We might think about a model which is particularly difficult to optimize, for instance if its performance, as defined by the objectives, decreases very fast as its parameters get away from optimal values. Optimizing such a model corresponds to finding a needle in a haystack and in this case, optimization will surely converge towards a local and easier to find optimum. It is then possible that a comparison performed as described above will conclude that such a model is less efficient than another one, which is easier to optimize and converges towards an optimum that is above the local optimum of the first model, but maybe not above its global optimum. To be rigorous then, while performing such a comparison, it is necessary to make the hypothesis that both models are compatible with the optimization, i.e. that their global optimum is not on a sharp peak of performance. Although this is difficult to check in general, in practice, such an hypothesis can be tested on well known parameter sets used to study the model. If small changes in these sets results in a performance that is very low (for instance of the same order of magnitude than that of randomly generated sets of parameters), then it is clear that such an hypothesis doesn't hold. In the other case, the hypothesis is unfortunately not validated but can be considered, by default, as reasonable.

Finally, the motivation of this work was to propose an approach to tackle the epistemic opacity inherent to at least some computational models. At this point, we may wonder if we succeeded in making such models less opaque. We have proposed the use of multi-objective optimization algorithms that rely on heuristics, in particular on an inspiration from natural selection and with no strong mathematical grounding. In a second step, we have presented data mining tools to extract from the generated data new knowledge. Doesn't it contribute to make the conclusions weak because of an opaque process? Even if the generation of the data relies on the evolutionary optimization, at the end, what are considered are just some sets of parameters of the model. The method used to generate them is of little impact on the conclusions, as long as no definite conclusions are drawn on the absence of observation of some model behaviors. The evolutionary optimization can then be considered as black box heuristics to discover set of parameters that are worth studying. In this respect, they are not less justified than a manual exploration, as an expert may be driven by biases that are no more justified. Actually, the exploration made by the evolutionary algorithm has the advantage of not being biased by any a priori knowledge on the model. On the other side, an expert, as it can reasonably observe only a small number of model behaviors, will be strongly biased towards parameter values that seem intuitively interesting to him and this intuition may be misleading. The second step, i.e. the analysis step, is not different from current scientific approach were a scientist has to dig in huge amount of data to extract the knowledge he is looking for. In that perspective and from an epistemological point of view, nothing new is proposed here. The opacity of this part of the knowledge building process, if any, is then similar to that of any other experimental science. The epistemic opacity has then not increased. Was it reduced? The analysis highlights potentially unforeseen features of the model, thus leading to a better understanding of its inner working and as a consequence to a reduction of the underlying epistemic opacity, at least for the aspects on which the analysis was focused.

Conclusion

We have proposed a framework for studying computational models relying on the use of multi-objective optimization algorithms. The main principle of the methodology consists in exploiting the ability to simulate the behavior of such models a huge number of times to generate model parameter sets that are worth studying. The choice of these data is driven by the optimization of multiple conflicting objectives at the same time, resulting in a set of optimal solutions rather than a single optimal solution. Multi-objective evolutionary algorithms were used for their versatility and for their ability to generate a dense approximation of the set of the best trade-off solutions. Analysis to perform were suggested together with typical questions they may contribute to answer. Two different examples were used to illustrate the potential of the approach: the study of a simulated flapping wing aircraft and the study of basal ganglia brain nuclei models. On the flapping wing aircraft model, the method has highlighted results compatible with the literature for medium to high speeds and non realistic results for low speeds, suggesting to focus further work on this particular point. Likewise, three kinematic parameters revealed to be critical for the minimum energy solutions, opening the way to a reduced model design. On the basal ganglia brain nuclei, the method revealed that the most biologically plausible solutions are also those that are the closest to the functionality attributed to this brain structure. The method also highlighted the importance of a particular connection, whose importance is debated in the neuroscience community and thus suggesting important new hypothesis on its potential use. Finally, a comparison of two brain models using the multi-objective method was performed and the model including the most recent neuroanatomical knowledge dominated the other, thus confirming the importance of the new modeled structures. For the two models, all these results were directly deduced after an analysis of the data generated by the multi-objective evolutionary algorithm.

Acknowledgements

This project was partially funded by the ANR EvoNeuro project, ANR-09-EMER-005-01. Hamdaoui M. has been supported by the Systematic cluster through the CSDL project (DGT 117407) (<http://www.systematic-paris-region.org/fr/projets/csdl>).

References

1. Girard B, Tabareau N, Pham Q, Berthoz A, Slotine J (2008) Where neuroscience and dynamic system theory meet autonomous robotics: a contracting basal ganglia model for action selection. *Neural Networks* 21: 628–641.
2. Gurney K, Prescott T, Redgrave P (2001) A computational model of action selection in the basal ganglia. II. Analysis and simulation of behaviour. *Biological cybernetics* 84: 411–423.
3. Humphreys P (2002) Computational Models. *Philosophy of Science* 69.
4. Humphreys P (2009) The philosophical novelty of computer simulation methods. *Synthese* 169: 615–626.
5. Humphreys P (1995) Computational science and scientific method. *Minds and Machines* 5: 499–512.
6. Morgan MS (1999) Learning from models. *Models as mediators: perspective on natural and social science* : 347—388.
7. Hughes RIG (1997) Models and Representation. *Philosophy of science* 64: 325—336.
8. Van Geit W, De Schutter E, Achard P (2008) Automated neuron model optimization techniques: a review. *Biological cybernetics* 99: 241–51.
9. Igel C, Erlhagen W, Jancke D (2001) Optimization of Neural Field Models. *Neurocomputing* 36: 225–233.

10. Quinton JC (2010) Exploring and Optimizing Dynamic Neural Fields Parameters Using Genetic Algorithms. In: IEEE World Congress on Computational Intelligence 2010 - WCCI 2010. Barcelona, Spain. URL <http://hal.archives-ouvertes.fr/inria-00488914/fr/>.
11. Cangelosi A, Parisi D (1997) A Neural Network Model of Caenorhabditis Elegans: The Circuit of Touch Sensitivity. *Neural Processing Letters* 6: 91–98–98.
12. Degris T, Lacheze L, Boucheny C, Arleo A (2004) A spiking neuron model of head-direction cells for robot orientation. In: Proceedings of the Eighth Int. Conf. on the Simulation of Adaptive Behavior, from Animals to Animats. pp. 255–263. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.6%48&rep=rep1&type=pdf>.
13. Wang Y, Li S, Chen Q, Hu W (2007) Biology inspired robot behavior selection mechanism: using genetic algorithm. In: LSMS 2007. LNCS, vol. 4688. Springer, pp. 777–786. URL <http://portal.acm.org/citation.cfm?id=1778240.1778328>.
14. Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research* 28: 1015–1031.
15. Wang QJ (1991) The Genetic Algorithm and Its Application to Calibrating Conceptual Rainfall-Runoff Models. *Water Resources Research* 27: 2467–2471.
16. Ogou P, Vijai H, Sorooshian S (1998) Multi-objective global optimization for hydrologic models. *Journal of Hydrology* 204: 83–97.
17. Gupta HV, Sorooshian S, Yapo PO (1998) Toward improved calibration of hydrologic models : Multiple and noncommensurable measures of information. *Water Resources* 34: 751–763.
18. Efstratiadis A, Koutsoyiannis D (2010) One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal* 55: 58–78.
19. Chaparro B, Thuillier S, Menezes L, Manach P, Fernandes J (2008) Material parameters identification: Gradient-based, genetic and hybrid optimization algorithms. *Computational Materials Science* 44: 339–346.
20. Zhu F, Jin X, Guan F, Zhang L, Mao H, et al. (2010) Identifying the properties of ultra-soft materials using a new methodology of combined specimen-specific finite element model and optimization techniques. *Materials & Design* 31: 4704–4712.
21. Aguir H, BelHadjsalah H, Hambli R (2011) Parameter identification of an elasto-plastic behaviour using artificial neural networks–genetic algorithm method. *Materials & Design* 32: 48–53.
22. Muñoz Rojas Pa, Cardoso EL, Vaz M (2010) Parameter Identification of Damage Models Using Genetic Algorithms. *Experimental Mechanics* 50: 627–634.
23. de Margerie E, Tafforeau P, Rakotomanana L (2006) In silico evolution of functional morphology: A test on bone tissue biomechanics. *Journal of the Royal Society, Interface / the Royal Society* 3: 679–87.
24. Vrugt J, Van Belle J, Bouten W (2007) Pareto Front Analysis of Flight Time and Energy-Use In Long-Distance Bird Migration. *Journal of Avian Biology* 38: 432–442.
25. Chen BL, Hall DH, Chklovskii DB (2006) Wiring optimization can relate neuronal structure and function. *Proceedings of the National Academy of Sciences of the United States of America* 103: 4723–4728.
26. Bongard J, Lipson H (2007) Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 104: 9943–9948.
27. Schmidt M, Lipson H (2009) Distilling free-form natural laws from experimental data. *Science (New York, NY)* 324: 81–5.
28. Freitas AA (2002) *Data Mining and Knowledge Discovery With Evolutionary Algorithms*. Springer-Verlag Berlin and Heidelberg GmbH & Co. K. URL <http://www.amazon.fr/Mining-Knowledge-Discovery-Evolutionary-Algorithms/dp/3540433317>.
29. Deb K, Srinivasan A (2006) Innovization: Innovating design principles through optimization. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation. Seattle, Washington, USA: ACM, pp. 1629–1636. URL <http://portal.acm.org/citation.cfm?id=1143997.1144266>.

30. Deb K, Sindhya K (2007) Deciphering Innovative Principles for Designing Electric Brushless DC Permanent Magnet Motors. KanGAL Report No2007011 .
31. Jeong S, Chiba K, Obayashi S (2005) Data Mining for Aerodynamic Design Space. *Journal Of Aerospace Computing, Information, and Communication* 2: 452–469.
32. Obayashi S, Chiba K (2007) Data Mining for Multidisciplinary Design Space of Regional-Jet Wing. *Journal of Aerospace Computing, Information, and Communication* 4: 1019–1036.
33. Michalewicz Z, Fogel DB (1999) *How to Solve It: Modern Heuristics*. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 467 pp. URL <http://www.amazon.fr/How-Solve-Heuristics-Z-Michalewicz/dp/35%40660615>.
34. Deb K (2001) *Multi-objective optimization using evolutionary algorithms*. Wiley.
35. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6: 182–197.
36. Deb K, Mohan M, Mishra S (2003) A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. KanGAL report .
37. Coello CA, Lamont GB, (plus) (2004) *Applications Of Multi-Objective Evolutionary Algorithms*. World Scientific Publishing Co Pte Ltd, 792 pp. URL <http://www.amazon.fr/Applications-Multi-Objective-Evolutionary-Algorithms-Carlos/dp/9812561064>.
38. Eiben AE, Smith JE (2008) *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer.
39. Grunert Da Fonseca V, Fonseca CM, Hall AO (2001) Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. *Lecture Notes in Computer Science/Evolutionary Multi-Criterion Optimization 1993*: 213–225.
40. Fonseca CM, Fleming PJ (1996) On the performance assessment and comparison of stochastic multiobjective optimizers. In: Heidelberg SB, editor, *Parallel Problem Solving from Nature — PPSN IV/Lecture Notes in Computer Science*. volume 1141, pp. 584—593. doi:10.1007/3-540-61723-X. URL <http://www.springerlink.com/index/f936513164j1574u.pdf>.
41. Zitzler E, Brockhoff D, Thiele L (2007) The Hypervolume Indicator Revisited : On the Design of Pareto-compliant Indicators Via Weighted Integration. *Lecture Notes in Computer Science/ Evolutionary Multi-Criterion Optimization 4403*: 862–876.
42. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Grunert Da Fonseca V (2003) Performance assessment of multiobjective optimizers: An analysis and review. *IEEE transactions on evolutionary computation* 7: 117—132.
43. Han J, Kamber M (2006) *Data mining: concepts and techniques*. Morgan Kaufmann. URL <http://books.google.com/books?hl=en&lr=&id=AfLOt-Yz%0rEC&oi=fnd&pg=PP1&dq=Data+Mining:+Concepts+and+Techniques&ots=Uu-VtMery6&sig=8aNWnn87f4kE8n1v292Ar8UnvNM>.
44. Hastie T, Tibshirani R, Friedman J (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag New York Inc. URL <http://www.amazon.fr/Elements-Statistical-Learning-Inference-%Prediction/dp/0387952845>.
45. Berry M, Linoff G (1999) *Mastering data mining: The art and science of customer relationship management*. John Wiley & Sons, Inc. New York, NY, USA. URL <http://portal.acm.org/citation.cfm?id=555358>.
46. Edelstein HA (1999) *Introduction to Data Mining and Knowledge Discovery*. Two Crows Corp. URL <http://www.amazon.fr/Introduction-Data-Mining-Knowledge-Disco%very/dp/1892095009>.
47. Weiss S, Indurkha N (1998) *Predictive data mining: a practical guide*. Morgan Kaufmann. URL <http://books.google.com/books?hl=en&lr=&id=xzVD8C2Y%pnQC&oi=fnd&pg=PR11&dq=Predictive+Data+Mining:+A+Practical+Guid%e&ots=ILb8myNeQK&sig=90dWHONhiSxFSc8OTE8GLKvcY5w>.
48. Hanne T (1999) On the convergence of multiobjective evolutionary algorithms. *European Journal of Operational Research* 117: 553–564.

49. Rudolph G, Agapie A (2000) Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC2000). pp. 1010—1016.
50. Benson HP (1978) Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Applications* 26: 569–580.
51. Messac A, Mattson CA (2004) Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. In: 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. Palm Springs, CA.
52. Zitzler E, Knowles J, Thiele L (2008) Quality Assessment of Pareto Set Approximations. In: Multiobjective Optimization: Interactive and Evolutionary Approaches, Berlin, Heidelberg: Springer, volume 5252 of *Lecture Notes in Computer Science*. pp. 373–404. doi:10.1007/978-3-540-88908-3. URL <http://www.springerlink.com/content/y06rr7220126712u>.
53. Knowles J, Corne D (2002) On metrics for comparing non-dominated sets. In: Proc. of 2002 Congress on on Evolutionary Computation. pp. 711—716. URL <http://doi.ieeecomputersociety.org/10.1109/CEC.2002.1007013>.
54. Kohonen T (2001) Self-Organizing Maps, volume 1 of *Springer series in information sciences, ISSN 0720-678X ; 30*. Berlin, Germany: Springer, 3rd edition, 501 p. pp. ISBN : 3-540-67921-9.
55. Zio E, Bazzo R (2010) Multiobjective optimization of the inspection intervals of a nuclear safety system: A clustering-based framework for reducing the pareto front. *Annals of Nuclear Energy* 37: 798 - 812.
56. Goel T, Vaidyanathan R, Haftka RT, Shyy W, Queipo NV, et al. (2007) Response surface approximation of pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering* 196: 879 - 893.
57. Hamdaoui M, Chaskalovic J, Doncieux S, Sagaut P (2010) Using multiobjective evolutionary algorithms and data-mining methods to optimize ornithopters' kinematics. *Journal of Aircraft* 47: 1504-1516.
58. Jeong S, Chiba K, Obayashi S (2005) Data mining for aerodynamic design space. *Journal of aerospace computing, information, and communication* 2: 452-469.
59. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and Regression Trees. Chapman & Hall/CRC. URL <http://www.amazon.fr/Classification-Regression-Trees-Leo-Breiman/dp/0412048418>.
60. Blasco X, Herrero J, Sanchis J, Martnez M (2008) A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences* 178: 3908 - 3924.
61. Tennekes H (1996) The simple science of flight (from insects to jumbo jets). MIT Press.
62. Tobalske B, Dial K (1996) Flight kinematics of black-billed magpies and pigeons over a wide range of speeds. *J Exp Biol* 199: 263-280.
63. Park K, Rosen M, Hedenstrom A (2001) Flight kinematics of the barn swallow (*hirundo rustica*) over a wide range of speeds in a wind tunnel. *J Exp Biol* 204: 2741-2750.
64. Hedrick TL, Tobalske BW, Biewener AA (2002) Estimates of circulation and gait change based on a three-dimensional kinematic analysis of flight in cockatiels (*Nymphicus hollandicus*) and ringed turtle doves (*Streptopelia risoria*). *Journal of Experimental Biology* 205: 1389–1409.
65. de Margerie E, Mouret JB, Doncieux S, Meyer JA (2007) Artificial evolution of the morphology and kinematics in a flapping-wing mini UAV. *Bioinspir Biomim* 2: 65–82.
66. Druot T (2004). Technical report on the implementation and validation of a flight mechanics simulator for flapping articulated wings.
67. Shyy W, Lian Y, Tang J, Viieru D, Liu H (2007) Aerodynamics of low Reynolds number flyers. Number 22 in Cambridge Aerospace Series. Cambridge University Press. ISBN-13: 9780521882781.
68. Taylor G, Nudds R, Thomas A (2003) Flying and swimming animals cruise at a Strouhal number tuned for high power efficiency. *Nature* 425: 707–711.
69. Walker JA (2002) Rotational lift: something different or more of the same? *J Exp Biol* 205: 3783-3792.

70. Spedding G, Hedenstrom A, McArthur J, Rosen M (2008) The implications of low-speed fixed-wing aerofoil measurements on the analysis and performance of flapping bird wings. *Journal of Experimental Biology* 211: 215.
71. Hamdaoui M (2009) Optimisation multicritères de l'efficacité propulsive de mini-drônes biomimétiques à ailes battantes par algorithmes évolutionnaires. Ph.D. thesis, Université Pierre et Marie Curie.
72. Cohen M, Frank M (2009) Neurocomputational models of basal ganglia function in learning, memory and choice. *Behavioural brain research* 199: 141–156.
73. Liénard J, Guillot A, Girard B (2010) Multi-objective Evolutionary Algorithms to Investigate Neurocomputational Issues: The Case Study of Basal Ganglia Models. *From Animals to Animats 11* : 597–606.
74. Redgrave P, Prescott T, Gurney K (1999) The basal ganglia: a vertebrate solution to the selection problem? *NEUROSCIENCE-OXFORD*- 89: 1009–1024.
75. Staines W, Atmadja S, Fibiger H (1981) Demonstration of a pallidostriatal pathway by retrograde transport of HRP-labeled lectin. *Brain Research* 206: 446–450.
76. Beckstead R (1983) A pallidostriatal projection in the cat and monkey. *Brain research bulletin* 11: 629–632.
77. Bevan M, Booth P, Eaton S, Bolam J (1998) Selective innervation of neostriatal interneurons by a subclass of neuron in the globus pallidus of the rat. *Journal of Neuroscience* 18: 9438.
78. Gage G, Stoetzner C, Wiltschko A, Berke J (2010) Selective Activation of Striatal Fast-Spiking Interneurons during Choice Execution. *Neuron* 67: 466–479.