33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

A GENERALISED MULTI-OBJECTIVE ISLAND MODEL FOR FLIGHT CONTROL SYSTEM CLEARANCES

Patrick Piprek[1], Pedro Miguel Dias[1] & David Schwalb[1]

[1]Airbus Defence and Space GmbH, Flight Dynamics MAN, Rechliner Straße, 85077 Manching

Abstract

The clearance process of fighter aircraft is a very challenging task due to the large amount of possible flight conditions, pilot commands, potential failure scenarios, and uncertainties that must be assessed to find the most critical cases. To automatize the selection from this parameter space and increase the confidence in finding the worst-case combinations, this paper proposes an island-model-based framework allowing for multi-objective optimisation. The proposed scheme employs a two-stage parallelisation capability, both parallelising the islands themselves as well as the model evaluation inside them. By this, it enables great scalability for computations on a high performance computing cluster, while it is also very flexible in terms of potentially used optimisation algorithms. Furthermore, it is very generic with respect to the evaluated model dynamics as well as the used objectives, applied constraints, and optimisation parameters. The proposed framework is demonstrated in a first step with test functions commonly used for global optimisation problems. Finally, an application example within a fighter aircraft clearance of a severe turbulence assessment using the described framework is shown.

Keywords: global optimisation, multi objective optimisation, generalised island model, fighter aircraft clearance, flight control system

1. Introduction

In modern fly-by-wire aircraft, the flight control system (FCS) plays a key role in the safety, performance, and reliability of the aircraft. The complexity and dependency on its correct behaviour further increases in modern fighters, which are often designed to be naturally unstable in order to enhance manoeuvrability and agility. Such aircraft solely rely on the FCS to artificially stabilize the aircraft. The robustness and correct functioning of the FCS across the applicable flight envelope is assessed in a clearance process. This assessment typically consists of the application of several sub-system tolerances, aircraft configurations, manoeuvrers, and time-variant failures, all applied to an aircraft with highly non-linear dynamics. As a direct consequence, the task at hand becomes complex and multi-dimensional. Up to these days, much of this assessment work is carried out manually, with the assessment engineers selecting predefined flight conditions based on previous experience and combinations of parameters in a so-called grid approach [1]. Here, optimisation-based schemes offer significant potential to reduce the manual workload and, specifically, the “engineering judgement” involved in deciding which parameter combinations are critical. Thus, they are chosen in this study as the method for the proposed generalised multi-objective FCS clearance framework.

In general, clearance of FCS using more modern approaches than a grid-based solution have been studied in [2], with optimisation methods being one option also used and applied as an extension for some time now [3]. Compared to traditional approaches, like the mentioned grid-based evaluation [1], the benefit is that they offer a mathematical foundation in the sense that they converge to the optimal points under algorithm-dependent conditions. Additionally, they remove the necessity of a discrete, grid-based specification towards a more real-world related continuous evaluation. Finally, the accessibility of high-performance computing (HPC) clusters gives the opportunity to exploit more

capabilities of optimisation algorithms, as the evaluation and convergence times can be reduced significantly by those to within a reasonable time frame due to their parallelisation capabilities.
As noted, there have already been approaches to clear flight control laws using e.g. gradient-based optimisation schemes [3, 4], which may also consider uncertainties [5]. Furthermore, study [6] introduced the optimisation of the angle of attack with different objective function models and compares the numerical result to the analytical optimum. Although these approaches generally offer fast convergence due to the gradient step, they require smooth enough dynamics and controllers as well as continuous optimisation parameters to be viable in general applications. Additionally, multi-objective problems are normally not straightforward to solve. Thus, they are not yet feasible in the highly non-linear and non-smooth multi-objective FCS clearances of fighter aircraft.
To overcome the drawbacks of gradient-based optimisation in terms of FCS clearance, studies have been conducted with a global optimisation-based perspective [7–10]. These schemes offer more promise towards dealing with non-smooth optimisation problems. In studies [7] and [8], differential evolution and genetic algorithms, in connection with local improvements by means of gradient-based optimisation, are proposed. The goal is to optimise a handling qualities objective of a delta-canard aircraft. Following, study [9] optimises time-varying pilot inputs for FCS clearances. Both global as well as local optimisation methods are checked. Finally, study [10] introduces the connection of global optimisation with game theory for the carefree handling clearance of a fighter aircraft. Additionally, it introduces parametrizations for the valid range of pilot inputs. Consequently, effort to use optimisation within, specifically, fighter aircraft clearance has frequently been carried out. However, most of the proposed schemes have been focused on single-objective optimisation, which is a drawback for actual clearance assessment as multiple requirements, and thus objectives, must be checked at once. Furthermore, the exploitation of the full capabilities of HPC clusters has also not yet been taken fully into account in connection with optimisation-based FCS clearance.
Consequently, this paper proposes a global multi-objective optimisation framework as a means for automatizing clearance assessments and enhancing the chances of finding relevant problematic regions of the flight envelope. The proposed algorithm is based on the concept of the generalised island model [11] and its implementation is centred around achieving maximum flexibility and scalability. This specifically applies to the range of potentially applied optimisation algorithms that can be incorporated. Furthermore, it is suitable for computationally demanding objective function evaluations with multiple objectives due to its two-stage parallelisation capabilities.
To show the developed algorithm, the paper is structured as follows: Section 2 introduces the theoretical background and the underlying algorithms of the proposed framework. Following, Section 3 shows the proposed multi-objective island model implementation. This proposal is verified in Section 4 by means of standard global optimisation test functions, while Section 5 shows the application in an actual fighter aircraft clearance assessment. A summary and an outlook are given in Section 6.

2. Theoretical Background

This section summarises the theory of the used algorithms connected within the proposed FCS clearance framework. It specifically introduces the considered optimisation problem formulation, details the specific requirements on objectives and constraints, and introduces the basic optimisation algorithms.

2.1. Optimisation Problem

In general, the following multi-objective minimisation problem, including constraints, is solved in an FCS clearance assessment:

$$
\begin{aligned}
\underset{\mathbf{z}}{\text{minimize}} \quad & \mathbf{J}(\mathbf{x};\mathbf{z}) , && \mathbf{J} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^o \\
\text{subject to} \quad & \mathbf{z}_{\text{lb}} \leq \mathbf{z} \leq \mathbf{z}_{\text{ub}} , && \mathbf{z} \in \mathbb{R}^n , \\
& \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x};\mathbf{z}) , && \mathbf{f} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m , \\
& \mathbf{c}_{\text{lb}} \leq \mathbf{c}(\mathbf{x};\mathbf{z}) \leq \mathbf{c}_{\text{ub}} , && \mathbf{c} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p , \\
& \boldsymbol{\psi}_{\text{eb}} = \boldsymbol{\psi}(\mathbf{x};\mathbf{z}) , && \boldsymbol{\psi} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^q .
\end{aligned}
\tag{1}
$$

Here, $\mathbf{z}$ is the set of optimisation parameters. The objective functions as well as equality and inequality constraints are given by the symbols $\mathbf{J}$, ψ, and $\mathbf{c}$, respectively. The state dynamics are defined by the function $\mathbf{f}$ for the states $\mathbf{x} \in \mathbb{R}^m$. Take into account that these are solved using a single shooting discretisation [12] in this study. Finally, the indices lb, ub, and eb denote the lower, upper, and equality bounds for the constraints, respectively.

The main practical caveat of the problem specification in (1) is the multi-objective definition. Such problems are generally difficult to solve, because they require Pareto optimality [13], which may be difficult to achieve. However, they are of high relevance in practical FCS clearance assessments because there is not only one objective, but multiple ones, based on the different requirements, that have to be analysed such as angle of attack limits and control surface saturations.

Consequently, the main contribution of this paper is the definition of a framework that allows for the optimisation of problems as specified in (1) in an efficient manner, specifically keeping in mind the large set of potential optimisation parameters that are encountered in FCS clearances. Furthermore, it is important to note that the objectives, evaluated as part of a FCS clearance, are generally discrete, meaning that they are for instance the maximum angle of attack encountered over the optimisation horizon, for which the time instance naturally changes for different scenarios. This makes the problem non-smooth by construction, in addition to the potentially anyway used set of discrete optimisation parameters, and consequently difficult to solve in nature.

2.2. Optimisation Algorithm

To solve the problem specified in (1), the use of genetic algorithms is proposed in this study. However, any optimisation algorithm suitable for the specific problem formulation to be solved can be applied in the proposed island model framework, which is detailed in Section 3. Consequently, the exact mathematical description of the genetic algorithm operators is out of the scope of this paper and the reader is referred to the literature. The following paragraphs will thus only give a general overview on some details of the algorithms required to understand the behaviour of the island model. In general, the proposed scheme was already successfully tested with the NSGA2 [14], NSGA3 [15], and omni-optimisation [16] genetic algorithms.

The basic flow of a genetic algorithm is visualised in Figure 1: In general, the idea is to define an initial population of the optimisation parameters, most often based on random sampling, and then evaluate the dynamics and constraints to get an initial overview on where valid critical regions may lie and sort the results accordingly. This is the non-dominated and crowding distance sorting step in the figure (which is more specific to the NSGA2 [14], but, in one way or the other, this idea is followed by all multi-objective genetic algorithms). Here, the individuals are sorted into different fronts from closest to farthest away from current Pareto front. Furthermore, solutions that are relatively isolated are preferred over solutions that are crowded together to ensure a well-spread approximation of the solution. Then, the genetic algorithm operators of “selection”, “crossover”, and “mutation” are applied to generate a new offspring population (which can be smaller in size than the initial population) that should improve on the solution, i.e. approximate the actual Pareto front better. This procedure is iterated until a stopping criterion, which is most often the number of generations, is reached. In general, the balance between exploitation and exploration of the search-space is essential for the performance of this kind of algorithms. Such balance is achieved mainly via crossover and mutation respectively, with the selection of the parents also playing a role to this.

2.3. Objective and Constraint Modelling

The following sections give an overview on the objective (Section 2.3.1) and constraint modelling (Sections 2.3.2) within the proposed optimisation framework that is required to efficiently solve the FCS clearance problem.

2.3.1. Objective Modelling

The objectives in this study are typically user-defined criteria that are referring to clearance requirements and are evaluated based on the analysis of time histories. As a result, different objectives can have different magnitudes, making it more difficult to compare them. Furthermore, while the optimisation algorithms are normally designed for minimisation problems, FCS clearances generally try to

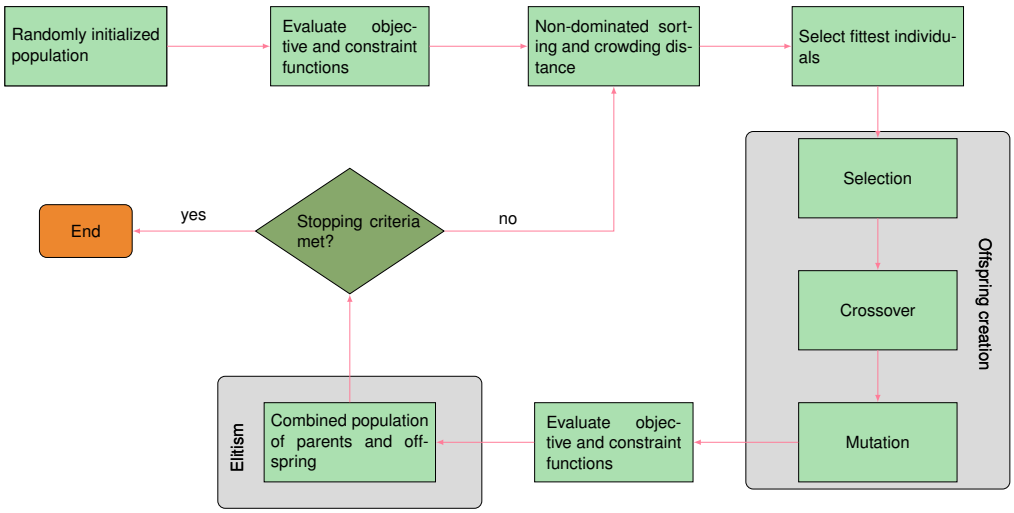

Figure 1 – Flowchart of a generic genetic algorithm applicable in the islands of the generalised island model.

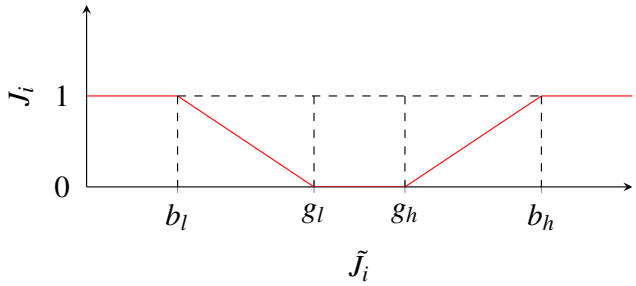

Figure 2 – Criteria transformation using bad-good values based on fuzzy logic for i-th objective.

solve maximisation problems (such as finding the worst, i.e. maximum, angle of attack). To account for both issues, a transformation based on fuzzy logic is performed [17]. This is depicted in Figure 2. The transformation can be shaped by the user using so-called bad-good value combinations. The optimal range of values is defined between good-low, g_l, and good-high, g_h, which are equally mapped to 0, thus being the optimal value the optimisation is trying to reach. Between bad-low, b_l, and good-low as well as good-high and bad-high, b_h, the values are mapped to an interpolated value between 0 and 1 and they provide an indication of how close the current solution is to the optimal one. Below bad-low and above bad-high, the values are mapped to 1 which is the worst possible value in the optimisation. Consequently, the mathematical definition of the mapping for the i-th objective is given by:

$$J_i = \max\left\{0, \max\left[\min\left(1, \frac{\tilde{J}_i - g_l}{b_l - g_l}\right), \min\left(1, \frac{\tilde{J}_i - g_h}{b_h - g_h}\right)\right]\right\} \tag{2}$$

Here, $\tilde{J}_i$ is the original, non-normalized objective value.
It is important to stress that the definition in Figure 2 also serves a further purpose in the context of FCS clearances: Here, it is generally not of interest to find the isolated extremal point (which fulfils all optimality conditions), but all parameter combinations that lead to unsafe operation of the aircraft. Thus, the mapping between good-low and good-high is generally an interval of all results that are considered unsafe operation.

2.3.2. Constraints

Constraints are handled in this implementation by applying a constrained-domination criterion [14]. This is an extension of the usual domination criterion. A solution i is said to constrained-dominate a solution j, if any of the following conditions is true:

1. Solution i is feasible and solution j is not
2. Solutions i and j are both infeasible, but solution i has a smaller overall constraint violation cv
3. Solutions i and j are feasible and solution i dominates solution j (standard domination criterion)

By convention, a negative constraint violation cv means that the solution is violating and it is therefore infeasible. If it is feasible, then it is always zero. The calculation of the constraint violation depends on the closest boundary that it is violating and on the nature of the constraint (continuous or discrete).

Continuous constraints Continuous constraints define continuous feasible regions defined by min and max values. Figure 3 illustrates the four distinct, reasonable feasible regions that can be defined.

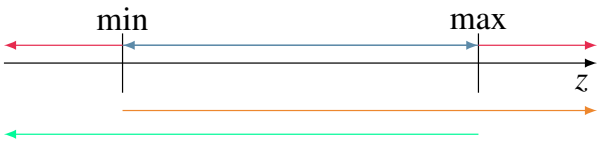

Figure 3 – Distinct feasible regions visualisation including allowed minimum, $\min$, and maximum, $\max$, values for continuous constraints.

Taking the region from a minimum value, $\min$, to positive infinity (orange arrow to the right) as an example, the constraint violation for the scalar variable z is calculated as follows:

$$cv = \begin{cases} 0, & \text{if } z \geq \min \\ \frac{z-\min}{|\min|}, & \text{else} \end{cases} \tag{3}$$

A similar logic is applied to the other regions, always taking the closest boundary.

Discrete constraints As opposed to continuous constraints, discrete constraints define a scattered feasible region that may always be mapped to integer values. Instead of min and max values, a set of values is defined. A threshold, ε, may be used to check if a solution is feasible or not to avoid potential issues with numerical precision.
Due to its discrete nature, the calculation of the constraint violation, cv, as specified before is not possible. Therefore, a binary cv is assumed, i.e., $cv = -1$ if violating and $cv = 0$ otherwise.

2.4. Hypervolume Performance Metric

There are multiple techniques for measuring the performance of multi-objective optimisation algorithms available in the literature [18]. Here, the focus is on the hypervolume (HV) metric, because it is one of the most used metrics due to its superior properties in measuring the performance of even different algorithms and making them comparable. This is a core requirement of the proposed scheme due to the fact that it should be independent of the actually used optimisation algorithm and whether or not the islands apply the same or different algorithms.
In general, the HV indicator is additionally popular, because it shows the closeness of the solutions to the optimal set and, to some extent, the spread of the solutions across objective space by means of a scalar measure. It measures the size of the portion of objective space that is dominated by a set of solutions collectively. One known shortcoming of this metric is that it is computationally heavy to calculate. However, faster approaches have been proposed which are also scalable to n-dimensions. This study uses one of these, the so-called “HV by slicing objectives” (HSO) algorithm [13, 19, 20]. It is based on the idea of processing one objective at a time, which significantly improves the computational speed. Here only a short overview of the HSO algorithm is given. For details regarding different steps in the algorithm the reader is referred to [13, 19, 20].
Before starting the algorithm, a key step is to set an appropriate reference point $\mathbf{r}$. This is specifically crucial if the evolution of the HV during the optimisation is to be used or interpreted in any way. In

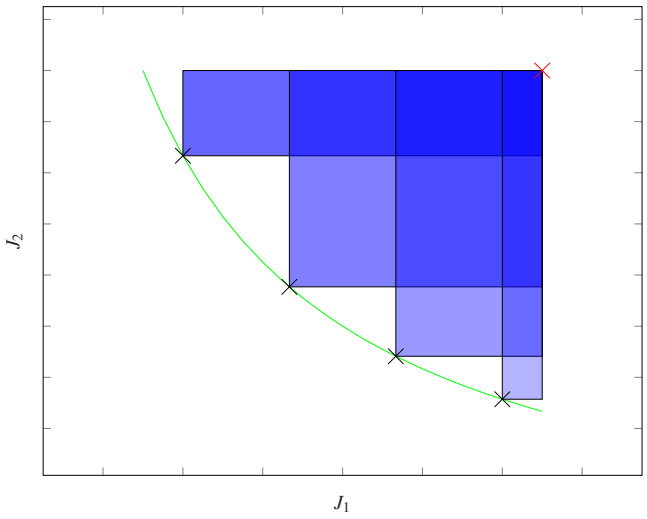

Figure 4 – General idea of hypervolume calculation for two-dimensional Pareto frontier.

general, setting the reference point requires previous knowledge of the objective space boundaries. For the problems defined in this study, this is straightforward due to the fuzzy logic defined in Section 2.3.1, which is applied to the objective values: This would consequently makes the choice for the ideal reference point $\mathbf{r} = (1, 1, \ldots, 1)$. However, to avoid the possibility of getting a HV that is equal to zero, which may need special treatment in the algorithms, and could happen if the fuzzy logic breakpoint values are specified unreasonable for the problem, the reference point is set to $\mathbf{r} = (2, 2, \ldots, 2)$. By this the HV is always greater than zero by construction, but still independent of the actual problem formulation, which makes it generic in its application.
The HSO algorithm can be summarised as follows: In the first step of the actual algorithm, the non-dominated Pareto optimal solutions are sorted by their first objective values, which are then used to cut slices through the HV, where each slice itself is an $n-1$ HV in the remaining objectives. As a next step, all these HV of the different slices are calculated and multiplied by their depth values. The depth value is the depth of the slice in the first objective. All these n-objective values are then added to obtain the total HV. Figure 4 shows a small two-dimensional example, which would be a slice of a three-dimensional space, illustrating the calculation of the HV based on the current points (black crosses) on the (estimated) Pareto frontier (green line) with respect to the reference point (red cross). In this case rectangles are used to calculate an estimation of the current HV.

3. Multi-objective Island Model Framework for FCS Clearances

The following paragraphs introduce the proposed generalised, multi-objective island model framework that can be used for FCS clearances. It is a connection of the principles introduced in Section 2, combining state-of-the-art optimisation algorithms with large-scale parallelised evaluations on a HPC cluster. The framework is implemented in Java™, but also provides interfaces to SQLite™ databases and the command line to execute e.g., Fortran or Matlab® code.
The proposed algorithm has a two-stage parallelisation scheme in which not only each island is running in parallel its individual algorithm, but also the evaluation of the objective/constraint functions is performed in parallel as illustrated in Figure 5: Here, the outermost parallelisation is provided by OpenMPI (cyan box), a message passing interface for HPC [21]. In this scope, OpenMPI is used to schedule the communication between the processes and execute the code. The actual distribution of the processes and allocating of the cores is handled by SLURM [22]. It should be noted that SLURM may also distribute the processes onto different nodes of the HPC cluster (light-blue boxes).
Inside the actual implementation of the proposed framework, one core will take the role of a “master” (red square), whose purpose is basically to organize the program-internal communication (dashed blue lines) and execution of the code. This master will distinguish the available processes, allocated by OpenMPI, based on their defined rank onto the number of specified islands (grey boxes). In general, it will distribute the processes equally onto the islands, assuming an equal workload. How-

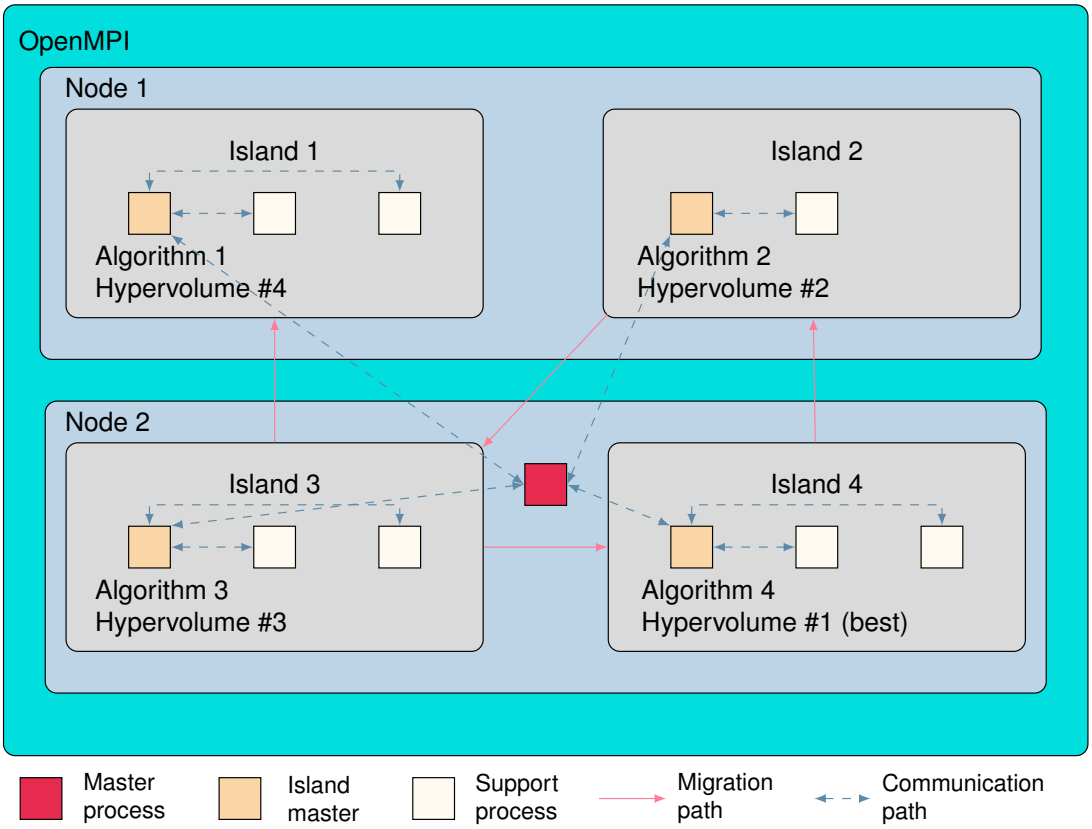

Figure 5 – High-level illustration of the proposed island model implementation including different parallelisation layers.

ever, different distribution schemes are possible, as e.g. indicated by “Island 2” in Figure 5, which has fewer processes. This non-equal distribution may specifically be reasonable if algorithms profit differently from parallelisation (e.g. gradient-based optimisers may generally benefit less from parallelisation than genetic algorithms). If there is no communication required/expected at this point, because the islands are busy with the execution of the optimisation, the master process may handle some clean-up and data transfer tasks.

Now, inside the islands, one process is scheduled to be the “island master” (orange square), which handles the supporting processes (white squares) by providing them the tasks they should execute, while it is also the only process that communicates the island’s status with the master. This is for instance required for scheduling the migration between the islands. However, the island master, together with the supporting cores, will mainly deal with the non-linear simulation, analysis of the time histories, and evaluating the steps in the selected optimisation algorithm.

Such an implementation allows for great scalability and leverages the computational power of modern HPC clusters with multiple computing nodes. Moreover, this parallelisation scheme is well-suited for optimisation problems which have demanding objective and constraint function evaluations. This is typically the case for FCS clearances, in which every function evaluation translates into a non-linear simulation of several seconds with additional analysis of the corresponding time histories.

Furthermore, the framework in Figure 5 does not make any assumptions on the algorithm each island is running. This means that each island can run the same algorithm with different settings or, in an extreme case, even completely different classes of optimisation algorithms. Thus, despite the common choice for this kind of optimisation framework being population-based algorithms, the proposed island model also allows for different classes such as gradient-based optimisation or game theory, which all work together towards achieving the same goals. This full flexibility of mixing different algorithms can consequently be viewed as a hybrid metaheuristic approach in which several algorithms can be combined. The use of these hybrid approaches has often proven to be beneficial in solving

optimisation problems because the benefits of the global search can e.g. be directly combined with an improved local convergence [7, 8].

A relevant feature in the proposed approach is the migration operator: In general, migration is the exchange of data between the different islands about their best solutions, thus helping to improve the overall performance characteristics by sharing information. Most often, the choice of the migration topology is constrained by the distributed computation setup and is carried out by deterministic or fixed migration paths, e.g. in a one-way ring [23]. In this paper however, the migration topology is dynamic and the destination of migration is chosen based on the performance of each island. To retrieve this information, the fundamental concept is to combine the migration step with typical performance metrics used to assess the quality of the Pareto front in multi-objective optimisation. Because of its properties, the most commonly used for such purposes is the HV metric introduced in Section 2.4, which translates the quality of the Pareto front into a scalar value. It is therefore also applied in this study. Despite being often regarded as a costly operation, faster algorithms have been proposed to increase its performance [19]. Furthermore, the calculation of the metric is only required when the migration takes place and therefore, its impact on the overall performance is considerably lower than e.g. the non-linear simulations. It should be noted here that by design each island can only receive individuals from a single island with a better HV value, except for the best island ("Island 4" in Figure 5), which can receive individuals from any island. To establish these island pairings, the master process must be aware of the HV values of all islands. As a consequence, this requires the island model to be synchronized at this stage. Potential migration paths are depicted by directional red arrows in Figure 5. It should be noted that the number of exchanged solutions is a user-defined property of the framework. Additionally, it is mentioned here that the master process will communicate to each island with which island it should exchange data, while the island masters take over the task of the actual exchange. This avoids the communication of the data to the master, which would then only act as a relay reducing the performance.

It is further worth pointing out that the island model in its essence is not an optimisation algorithm but rather an optimisation framework. Thus, the claim of being a multi-objective optimisation algorithm is driven from the basic algorithms executed by each island. Generally, the migration operator implemented copes with single- and multi-objective optimisation problems. This is due to the fact that the HV calculation in the boundary case of a single-objective essentially becomes a distance and an area in the case of two objectives. Therefore, the performance of the island can always be assessed and the migration logic can also be carried out making the proposed framework applicable to general optimisation problem formulations.

Summarising, this section introduced the proposed generalised island model for multi-objective optimisation. Although it is applied in the scope of FCS clearances in this paper, the methodology is very generic and can be applied to any sort of optimisation problem formulation. Specifically, the possibility to use different classes of optimisation algorithms within the islands, the generic migration operator, and the two-stage parallelisation are superior properties of the proposed scheme.

4. Test Function Verification

In this section, the proposed framework introduced in Section 3 is verified using a standard test function for global optimisation. For this purpose, the unconstrained, two-objective ZDT4 test function is used, which is defined as follows [24]:

$$\begin{aligned} J_1(\mathbf{z}) &= z_1 \\ J_2(\mathbf{z}) &= h(\mathbf{z}) \cdot \left[1 - \sqrt{\frac{z_1}{h(\mathbf{z})}}\right] \end{aligned} \tag{4}$$

The auxiliary function, $h(\mathbf{z})$, is given as follows:

$$h(\mathbf{z}) = 1 + 10 \cdot (n-1) + \sum_{i=2}^{n} \left[z_i^2 - 10 \cdot \cos\left(4 \cdot \pi \cdot z_i\right)\right] \tag{5}$$

Here, n is the number of optimisation variables.

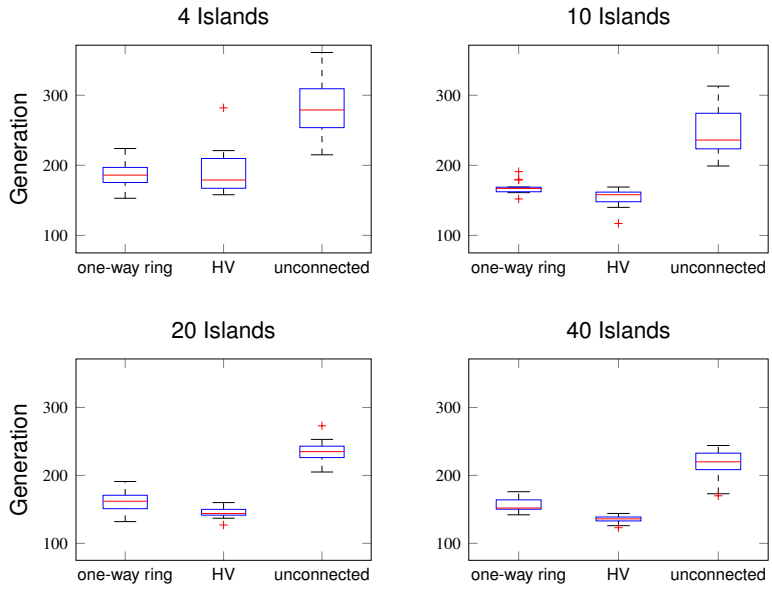

Figure 6 – Visualisation of convergence to optimal hypervolume for ZDT4 test function with different number of islands and migration operator.

The first conducted test compares the convergence towards the optimal HV, given in [24], of the test function for a different number of islands and migration operators (population size: 80 ; maximum number of generations: 500; migration of 10 solutions every 10th generation). The results are displayed as box plots (25th and 75th percentiles based on fifteen optimisations per island and migration setups; outliers symbolized by red crosses) in Figure 6: Here, the results for four, ten, twenty, and forty islands (each island here only has one process as the objective evaluation is not complex-enough to require a split onto different processes), half running an NSGA2 [14] and the other half running an omni-optimisation [16] algorithm, are compared for three different migration operators. The first one is the "one-way ring" [23], a standard way of migration in which the islands communicate in a ring with a fixed partner, the second is the HV method introduced in Section 2.4, while the third one is an unconnected example for reference, i.e. no migration between the islands occurs. The results show that the HV-based migration is overall converging with the fewest number of generations to the optimal solution. For a small number of islands, the one-way ring migration performs very similar to the HV-based migration due to the limited number of possible partners for the migration, which makes it easier for an island to be positively affected by another higher-performance island. However, for more islands the HV-based migration improves the time to convergence by means of fewer generations and also reduces the percentile spread, thus increasing the confidence in the obtained results from different runs. Finally, it can be seen that the migration is in general improving convergence speed significantly as the unconnected setup performs worst for all cases. Thus, Figure 6 shows that the proposed framework including the HV-based migration specifically improves the convergence for a large number of islands, which may normally be used in complex problems such as fighter aircraft clearances.

The next results, displayed in Figures 7 and 8, show the convergence of the current population (blue circles) for one of the evaluated cases with four islands towards the analytical Pareto front (solid black line). This is displayed for different generations (numbers 10, 140, 170, and 390) that specifically also contain migration. The effect of this step is displayed by visualising the removed (red triangles) and added (green squares) points to the front. Here, Figure 7 shows the behaviour for the first two islands running an NSGA2 algorithm, while the omni-optimisation algorithm is dealt with in Figure 8. Generally, it can be stated that both algorithms converged to the analytic Pareto front after roughly 170 generations matching the results of Figure 6.

Regarding the effect of migration, Figure 8 is particularly interesting: Here, it can be seen that in generation 140 of the "Island 3" column, two very bad solution on the y-axis are removed by the migration and substituted by solutions very close to the front from another island. This proves the

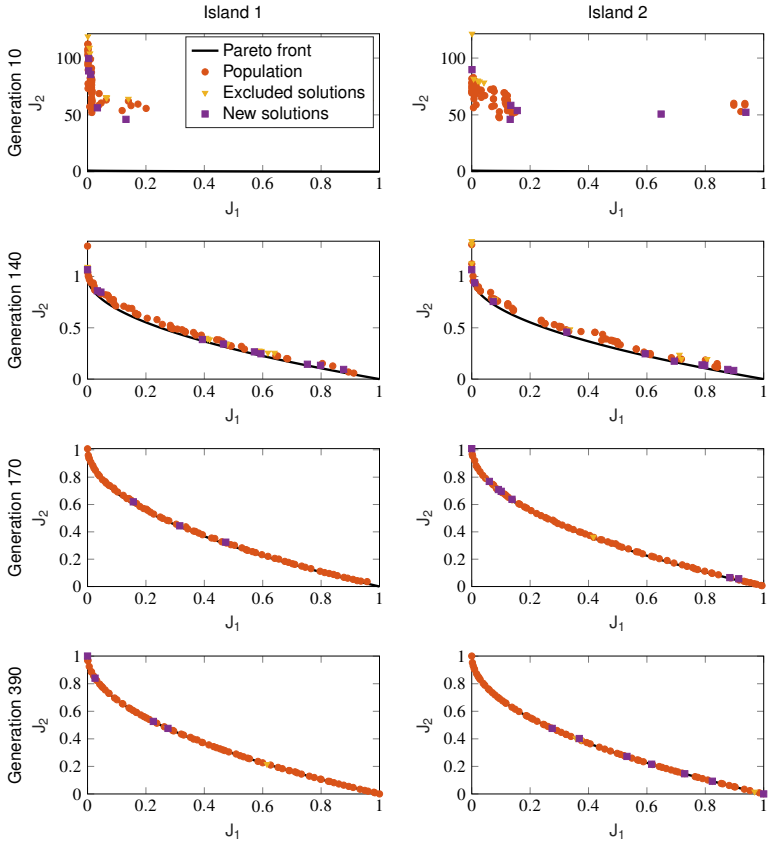

Figure 7 – Visualisation of convergence to Pareto front for ZDT4 test function displaying different generations and effect of migration for two islands running NSGA2.

claim of the migration being a strength of the algorithm to remove solutions that are of reduced practical interest as they are not part of the most critical solutions.

Concluding the test function section, Table 1 shows an overview of the average convergence and execution times, calculated based on the fifteen executed setups, as well as the encountered minimum and maximum times for the different number of islands with HV-based migration. Additionally, the standard deviations are displayed. It should be noted that the measured time not only includes the actual algorithm evaluation but e.g. also writing to files for means of tracking the progress. Furthermore, the mentioned synchronisation of all islands during the migration is considered. As it could be expected, convergence and execution times for four islands are lowest. However, the results for ten islands also a very similar behaviour to the one with four islands although the spread between the minimum and maximum value seems to become larger. The slowest execution is obtained with forty islands: However, the time is not even doubled while there are ten times more islands evaluated. Thus, the increase is within a very reasonable margin specifically considering that the results quality, as seen in Figure 6, could be improved drastically using more islands. It should also be considered that there is more time for synchronization required in the forty island case, which explains the increased measured times, and is specifically seen within the standard deviations as well. Furthermore, it can be seen that, using a suitable convergence condition e.g. for the HV, the actual execution time of the algorithms can be reduced by roughly 50% in all cases, which is an additional benefit of the HV-based migration operator that provides this information regularly.

Summarising, the viability of the framework proposed in Section 3 was proven in this section by means of analysing the ZDT4 test function. Results for a different number of islands, migration operators, and optimisation algorithms were shown, displaying the capabilities of the framework specifically with regard to scalability and flexibility. Following this verification by means of a test function, the next section deals with an actual clearance assessment for a fighter aircraft.

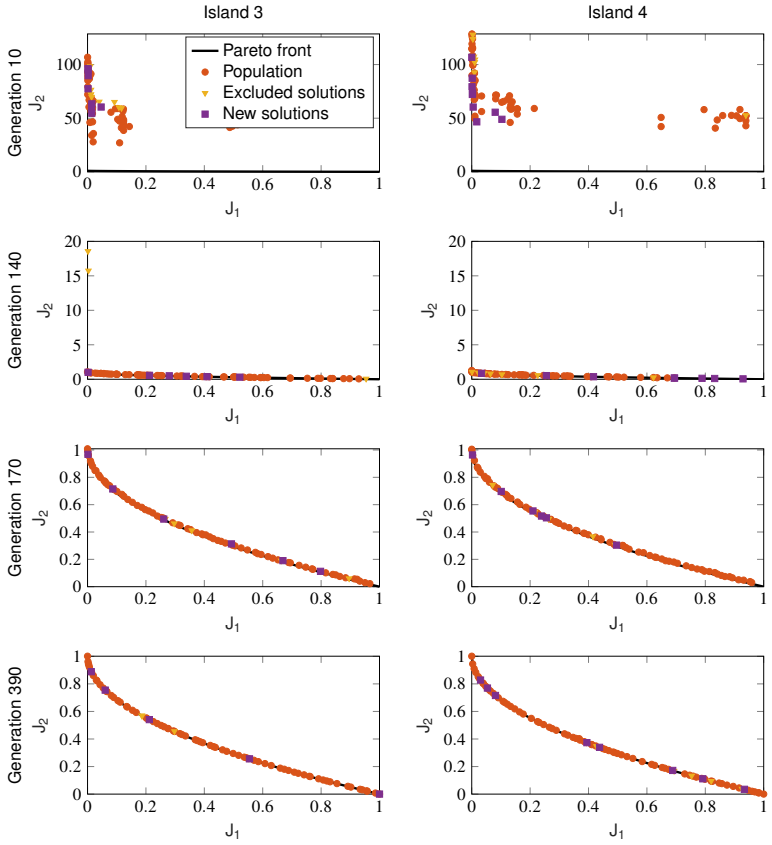

Figure 8 – Visualisation of convergence to Pareto front for ZDT4 test function displaying different generations and effect of migration for two islands running omni-optimisation.

Number of Islands	Convergence Time [s]				Execution Time [s]			
	Average	Minimum	Maximum	Standard deviation	Average	Minimum	Maximum	Standard deviation
4	6.86	5.23	8.86	1.09	11.36	10.30	14.30	1.15
10	6.80	4.57	10.57	1.39	12.98	11.48	16.97	1.72
20	7.52	5.35	12.66	1.73	14.63	12.84	20.87	2.62
40	10.89	7.03	15.74	2.39	21.34	17.38	26.99	3.97

Table 1 – Convergence and execution time statistics of ZDT4 test function evaluation for different number of islands with HV-based migration.

5. Application in Clearance Assessment

This section shows an application example of the proposed framework from Section 3 in an actual fighter aircraft clearance task: This task is the assessment of severe turbulence within an otherwise failure-free aircraft. This configuration should ideally be carefree, i.e. the pilot shall be able to input any (for the flight condition reasonable) input without destabilizing the aircraft [1]. This is modelled by different clinical pilot inputs that cover a broad range of actually used commands by pilots during operational flight [1]. The choice of the worst case manoeuvres in connection with the throttle position is an optimisation parameter. Further parameters are the envelope point, the air brake position, the tolerance on measurement accuracy of the air data system, the configuration (i.e. external stores), the mass estimation error, and the so-called “fuel sloshing”, which describes the movement of fuel in the tank due to manoeuvring. For the dynamics, a full, non-linear rigid-body simulation of the fighter aircraft is used.

Within the framework, three islands and a total of 100 processes are used. This means that each island has 33 processes associated to it. In this example, all islands run using an NSGA2 algorithm [14] with an initial population size for each island of 200 and subsequent populations having the size of 150. The number of generations per island is limited to 150 and the migration size is 10 every 10 gener-

Figure 9 – Initial objective values plotted inside flight envelope for the three islands based on random sampling in zero-th generation.

ations. This results in a total of $45,300$ non-linear simulations including time history analysis, which completed after roughly 1.5 hours on the HPC cluster (assuming a serial evaluation taking around $1\,\mathrm{s}$ per simulation, the evaluation would have taken roughly 12.5 hours). Finally, a two-objective optimisation problem is looked at with one objective being the normalized overshoot above the design FCS angle of attack limit for this flight condition, J_{AoA}, and the second one being the normalized maximum absolute angle of sideslip, J_{AoS}. Both of these variables are common initial assessment quantities to find critical situations in an FCS clearance. Flight envelope plots (altitude over Mach number) for different generations of the algorithms are discussed in the following: Figure 9 shows the flight envelopes for the three islands and two objectives for the initial, i.e. the randomly-sampled, populations. Here, one benefit of the island model is already clear in the sense that the different islands inherently deal with the non-equal spread introduced by the sampling. For instance, “Island 2” has comparably large “white areas” in the medium Mach-medium altitude domain, i.e. regions where no initial points are calculated. This is covered by the other islands fairly well, thus, removing the requirement to have sophisticated algorithms that produce an initial population because eventually critical solutions, if they are located in such regions, are exchanged via the migration.

Following, Figure 10 shows an intermediate solution after twenty generations (and two migrations). At this stage, the angle of attack objective is already similarly developed for all islands, while differences in the results are particularly seen in the angle of sideslip objective. However, the migration already seemed to have done a good job, together with the normal improvement of the algorithm, to distribute information as e.g. “Island 2” now has points in the originally empty region, which turned out to be comparably critical.

Concluding, Figure 11 shows the final solution envelopes after reaching the maximum number of generations. It can be seen that all algorithms found similar regions of critical objective values. Still, there are some differences in outliers or, presumably, less critical regions. As mentioned, this is a further benefit of the proposed framework because, except for the migration exchange, the algorithms evolve independently thus giving a more detailed overview on the actual result.

Summarising, this section has proven that the proposed island model framework is suitable for the application in highly-complex optimisation tasks, which was shown here for a typical fighter aircraft clearance assessment. Specifically, the parallelisation capabilities as well as the migration make the framework suitable for complex problems, where the actual solution is not known from experience, which normally makes it difficult to design a suitable solution scheme in another way.

Figure 10 – Objective values after two migrations plotted inside flight envelope for the three islands based on NSGA2 algorithm evolution.

Figure 11 – Final objective values after reaching maximum number of generations plotted inside flight envelope for the three islands based on NSGA2 algorithm evolution.

6. Summary and Outlook

This paper presents a framework for clearance of flight control systems using multi-objective optimisation algorithms within an island model approach. The method is specifically tailored to efficiently solve clearance tasks with large parameter space, occurring for instance for fighter aircraft, by employing a two-stage parallelisation scheme, which parallelises the islands and the dynamic model/constraint evaluation independently. By this, the proposed scheme specifically thrives on high-performance computing clusters. Furthermore, this framework gives great flexibility by allowing the use of different classes of optimisation algorithms in each island, while maintaining a formulation that is also independent of the actual optimisation task, by e.g. implementing a generic migration operator based on a hypervolume. Summarising, this enables the use in a wide range of problem formulations without the necessity to adapt the general framework.
Future developments may deal with the incorporation of other basic optimisation algorithms such as gradient-based optimisation schemes including sensitivities. Furthermore, more detailed models, e.g. in terms of pilot inputs, may be applied and statistical evaluation methods should be incorporated.

7. Contact Author Email Address

patrick.piprek@airbus.com

8. Copyright Statement

References

[1] R. Stich, *Clearance of Flight Control Laws for Carefree Handling of Advanced Fighter Aircraft*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[2] C. Fielding, A. Varga, S. Bennani, and M. Selier, eds., *Advanced techniques for clearance of flight control laws*. Lecture Notes in Control and Information Sciences, New York, NY: Springer, Sept. 2006.

[3] A. Varga, A. Hansson, and G. Puyou, *Optimization Based Clearance of Flight Control Laws: A Civil Aircraft Application*. Berlin: Springer Berlin Heidelberg, 01 2011.

[4] J. W. M. Diepolder, *Optimal Control Based Clearance of Flight Control Laws*. Dissertation, Technische Universität München, München, 2021.

[5] J. Diepolder, P. Piprek, B. Grüter, T. Akman, and F. Holzapfel, "Aircraft safety analysis using generalized polynomial chaos," in *Air Traffic Management and Systems III* (E. N. R. Institute, ed.), (Singapore), pp. 67–81, Springer Singapore, 2019.

[6] A. A. Herrmann and J. Z. Ben-Asher, "Flight control law clearance using optimal control theory," *Journal of Aircraft*, vol. 53, no. 2, pp. 515–529, 2016.

[7] P. P. Menon, J. Kim, D. G. Bates, and I. Postlethwaite, "Clearance of nonlinear flight control laws using hybrid evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 689–699, 2006.

[8] P. Menon, D. Bates, and I. Postlethwaite, "A deterministic hybrid optimisation algorithm for nonlinear flight control systems analysis," in *2006 American Control Conference*, pp. 6 pp.–, 2006.

[9] D. Skoogh, P. Eliasson, F. Berefelt, R. Amiree, D. Tourde, and L. Forssell, "Clearance of flight control laws for time varying pilot input signals," *IFAC Proceedings Volumes*, vol. 42, no. 6, pp. 343–348, 2009. 6th IFAC Symposium on Robust Control Design.

[10] R. Rodríguez Robles, M. Boullosa, and F. Nieto, "Flight control laws carefree handling clearance of a highly manoeuvrable aircraft using multi-strategy adaptive global optimization," in *EUCASS (European Conference for Aero-Space Sciences)*, 07 2017.

[11] D. Izzo, M. Ruciński, and F. Biscani, *The Generalized Island Model*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[12] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. Advances in design and control, Philadelphia, Pa.: Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), 2nd ed. ed., 2010.

[13] J. D. Knowles, *Local-search and hybrid evolutionary algorithms for Pareto optimization.* Doctoral thesis, University of Reading, Reading, 2002.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[15] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[16] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *Evolutionary Multi-Criterion Optimization* (C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, eds.), (Berlin, Heidelberg), pp. 47–61, Springer Berlin Heidelberg, 2005.

[17] H.-D. Joos, "A multiobjective optimisation-based software environment for control systems design," in *Proceedings. IEEE International Symposium on Computer Aided Control System Design*, pp. 7–14, 2002.

[18] N. Riquelme, C. Von Lücken, and B. Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American Computing Conference (CLEI)*, pp. 1–11, 2015.

[19] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[20] C. Fonseca, L. Paquete, and M. Lopez-Ibanez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *2006 IEEE International Conference on Evolutionary Computation*, pp. 1157–1163, 2006.

[21] R. L. Graham, T. S. Woodall, and J. M. Squyres, "Open MPI: A flexible high performance MPI," in *Parallel Processing and Applied Mathematics*, pp. 228–239, Springer Berlin Heidelberg, 2006.

[22] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing* (D. Feitelson, L. Rudolph, and U. Schwiegelshohn, eds.), (Berlin, Heidelberg), pp. 44–60, Springer Berlin Heidelberg, 2003.

[23] M. Rucinski, D. Izzo, and F. Biscani, "On the impact of the migration topology on the island model," *Parallel Comput.*, vol. 36, pp. 555–571, 2010.

[24] Zitzler, Eckart, Deb, Kalyanmoy, and Thiele, Lothar, "Comparison of multiobjective evolutionary algorithms: empirical results," *Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology Zürich (ETH)*, 1999.