A Knee Point based Evolutionary Multi-objective Optimization for Mission Planning Problems

Cristian Ramirez-Atencia  
Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Spain  
cristian.ramirez@inv.uam.es

Sanaz Mostaghim  
Faculty of Computer Science, University of Magdeburg, Germany  
sanaz.mostaghim@ovgu.de

David Camacho  
Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Spain  
david.camacho@uam.es

ABSTRACT
The current boom of Unmanned Aerial Vehicles (UAVs) is increasing the number of potential industrial and research applications. One of the most demanded topics in this area is related to the automated planning of a UAVs swarm, controlled by one or several Ground Control Stations (GCSs). In this context, there are several variables that influence the selection of the most appropriate plan, such as the makespan, the cost or the risk of the mission. This problem can be seen as a Multi-Objective Optimization Problem (MOP). On previous approaches, the problem was modelled as a Constraint Satisfaction Problem (CSP) and solved using a Multi-Objective Genetic Algorithm (MOGA), so a Pareto Optimal Frontier (POF) was obtained. The main problem with this approach is based on the large number of obtained solutions, which hinders the selection of the best solution. This paper presents a new algorithm that has been designed to obtain the most significant solutions in the POF. This approach is based on Knee Points applied to MOGA. The new algorithm has been proved in a real scenario with different number of optimization variables, the experimental results show a significant improvement of the algorithm performance.

CCS CONCEPTS
•Applied computing → Avionics; Multi-criterion optimization and decision-making; •Computing methodologies → Planning for deterministic actions; Motion path planning; •Theory of computation → Constraint and logic programming; •Evolutionary algorithms;

KEYWORDS
UAVs, Mission Planning, Constraint Satisfaction Problems, Evolutionary Multi-objective Optimization, Knee Point

1 INTRODUCTION
The fast technological improvements in Unmanned Aerial Vehicles (UAVs) capabilities has opened up new commercial applications for the industry. In the last few years, these vehicles have been used in many domains such as surveillance [7], flight training [11], payload visualization [13] or disaster and crisis management, since they avoid risking human lives while their manageability permits to reach areas of hard access. Some new scenarios have appeared where a swarm of UAVs is needed to operate in a coordinated way [12]. This swarm of vehicles needs to be controlled using a set of Ground Control Stations (GCSs), and requires new reliable mission planning systems, which should be able to handle a large number of constraints.

Mission Planning for a team of UAVs involves generating tactical goals, commanding structure, coordination, and timing. This problem implies the assignment of several tasks to the vehicles performing them, along with the assignments of vehicles to GCSs. In this context, there are several variables that influence the selection of the most appropriate plan, such as the makespan of the mission, the cost or the risk, i.e. it is a Multi-Objective Optimization Problem (MOP). This kind of problem can be solved using Multi-Objective Evolutionary Algorithms (MOEAs), obtaining the optimal set of non-dominated solutions or Pareto Optimal Frontier (POF).

One critical problem after obtaining the optimal solutions is the process of selecting one of the optimal solutions (decision making) by the mission operator. In some complex missions, the number of obtained non-dominated solutions could be very large, and the decision making becomes a hard process for the operator. If the operator provides some a priori information about the decision making, this could be used in the optimization process. Nevertheless, most of the times the operator does not provide this information, and it is necessary to consider other approaches for filtering the solutions.

In this paper, a new Evolutionary Multi-Objective Optimization (EMO) algorithm based on Knee Points has been developed to look for the most significant solutions in the POF. In this algorithm, the concept of domination is changed to cone-domination, where a larger portion or a cone region is considered when generating the solution frontier. This algorithm is applied to a real Mission Planning Problem with 7 optimization functions. Several experiments have been designed in order to examine the reduction of the number of solutions while maintaining the most significant ones.

The rest of the paper is structured as follows. Next section describes MOPs, the concept of Knee Points and the Cone-Domination. Section 3 describes the Mission Planning Problem (MPP). Section 4 presents the newly developed Knee-Point based Evolutionary Multiobjective Optimization algorithm, as well as the encoding and
fitness for the MPP. In Section 5 the new approach is tested with a real Mission Scenario and compared with a previous approach. Last section presents the final analysis and conclusions of this work.

## 2 MULTI-OBJECTIVE OPTIMIZATION

Real-world decision problems often require the solutions to meet multiple performance criteria (or objectives) simultaneously. These objectives are often conflicting, wherein an improvement in one objective cannot be achieved without detriment to another objective. In this case, there is no single solution to a MOP that can be selected objectively; rather a set of solutions exists representing different performance trade-offs between criteria. A minimization MOP can be mathematically defined as follows:

\[
\begin{align*}
\min f(x) &= (f_1(x), f_2(x), ..., f_m(x))^T \\
\text{subject to } x &\in S \subseteq \mathbb{R}^n
\end{align*}
\]

where \( x = (x_1, x_2, ..., x_n)^T \) is a vector of \( n \) decision variables from the decision space \( S; f : S \to \Theta \subseteq \mathbb{R}^m \) consists of a set of \( m \) objective functions, and a mapping from \( n \)-dimensional decision space \( S \) to \( m \)-dimensional objective space \( \Theta \).

**Definition 2.1.** Given two decision vectors \( x, y \in S \), \( x \) is said to Pareto dominate \( y \), denoted by \( x < y \), if:

\[
\forall i \in \{1, 2, ..., m\} \quad f_i(x) \leq f_i(y) \\
\exists j \in \{1, 2, ..., m\} \quad f_j(x) < f_j(y)
\]

**Definition 2.2.** A decision vector \( x^* \in S \), is Pareto optimal if \( \nexists x \in S, x < x^* \).

**Definition 2.3.** The Pareto set \( PS \), is defined as:

\[ PS = \{x \in S| x \text{ is Pareto optimal}\} \]

**Definition 2.4.** The Pareto front \( PF \), is defined as:

\[ PF = \{f(x) \in \mathbb{R}^m| x \in PS\} \]

The goal of EMO algorithms is to find the non-dominated objective vectors which are very close or even on the PF (convergence), and also to generate a good distribution of these vectors over the PF (diversity). The most used algorithm over the last decade in this field has been the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [4], which uses non-dominated ranking for the convergence and crowding distance for the diversity of the solutions.

### 2.1 Knee Points and Cone Domination

A number of EMO algorithms have been proposed to search for non-dominated solutions around reference points that are usually assumed to be given by a Decision Maker (DM) based on his/her preferences [8]. However, setting the reference point needs a priori knowledge that the DM sometimes does not have. In order to obtain favourable solutions without a priori knowledge, "knee points" [5] can be used. If the Pareto front has a clear knee point, most DMs may prefer solutions around the knee point. This is because a small improvement of any objective from the knee point leads to a large deterioration of at least one of the other objectives. In Figure 1, an example Pareto front showing which points are knee points and which are not is presented.

In the last decade, a few works have been presented that deal with knee point based MOEAs. Branke et al. [3] modified the crowding distance criterion in NSGA-II using angle-based and utility-based measures for focusing on knee points. Schütze et al. [14] designed two update methods based on maximal convex bulges for focusing the search of the algorithm to knee points. Bechikh et al. [2] proposed an extension of the reference point NSGA-II that uses the normal boundary intersection method to emphasize knee-like points. In this work, an angle-based measure is presented for focusing the search of knee points. In this context, the domination criterion defined earlier is changed to cone-domination. For this, a weighted function of the objectives is defined as follows:

\[
\Omega_i(f(x)) = f_i(x) + \sum_{j=1,j\neq i}^m a_{ij}f_j(x), \quad i, 1, 2, ..., m
\]

where \( a_{ij} \) is the amount of gain in the \( j \)-th objective function for a loss of one unit in the \( i \)-th objective function. The above set of equations requires the matrix \( a \), which has one in its diagonal elements.

**Definition 2.5.** A solution \( x \) is said to cone-dominate a solution \( y \), denoted by \( x <c^* y \), if:

\[
\forall i \in \{1, 2, ..., m\} \quad \Omega_i(f(x)) \leq \Omega_i(f(y)) \\
\exists j \in \{1, 2, ..., m\} \quad \Omega_j(f(x)) < \Omega_j(f(y))
\]

Let us illustrate the concept for two \( (m = 2) \) objective functions. The two weighted functions are as follow:

\[
\begin{align*}
\Omega_1(f_1, f_2) &= f_1 + a_{12}f_2 \\
\Omega_2(f_1, f_2) &= a_{21}f_1 + f_2
\end{align*}
\]

The above equations can also be written as

\[
\Omega = \begin{bmatrix} 1 & a_{12} \\ a_{21} & 1 \end{bmatrix} f, \quad \text{or}, \quad \Omega = af
\]

Figure 2 shows the contour lines corresponding to the above two linear functions passing through a solution \( A \) in the objective space. All solutions in the hatched region are dominated by \( A \).
A Knee Point based EMO for Mission Planning Problems

The Multi-UAV Cooperative Mission Planning Problem (MCMPP) consists of the assignment of a number $p$ of tasks, $T = \{t_0, t_1, \ldots, t_p\}$, to a team of $q$ UAVs, $U = \{u_0, u_1, \ldots, u_q\}$, at a specific time interval in a specific geographic zone. There are different kinds of tasks (e.g. photographing or escorting a target, monitoring a zone, etc.). Some of them can be performed by several UAVs (Multi-UAV), reducing the time needed to perform the task (e.g. mapping an area, or Step & Stare). Tasks can be performed using the sensors available by the UAVs in the mission (e.g. Electro-optical or Infrar-red (EO/IR) cameras, Synthetic Aperture Radars (SARs), etc.).

Additionally, the vehicles performing the mission have some features that must be taken into account in order to check if a mission plan is correct: its initial position, its initial fuel, its autonomy or maximum flight time, its range or maximum flight distance, its cost per hour of usage, its available sensors, and one or more flight profiles. A vehicle’s flight profile specifies at each moment its speed, its fuel consumption ratio and its altitude.

Besides, there exist a number $r$ of GCSs, $G = \{g_0, g_1, \ldots, g_r\}$, which must be assigned the UAVs that they control. These GCSs have also some features to be taken into account, such as the number of UAVs they can control or their communication ranges.

Figure 4 presents a Mission Scenario with 7 tasks, 5 UAVs and 3 GCSs. As shown in this figure, there could also exist some No Flight Zones (NFZs). These zones must be avoided in the trajectories of the UAVs during the mission.
In the planning process, the paths for vehicles between tasks, for departure and return are computed. In order to evaluate the durations, fuel consumption and other variables, it is necessary to know which of the UAV’s flight profiles will be used for each path, providing the fuel consumption ratio, speed and altitude as previously mentioned. For this reason, in these cases, the flight profiles used must also be assigned to solve the mission.

A common way to model this kind of problems is using Constraint Satisfaction Problems (CSPs) [9], since we need to find the correct schedule of resource-task assignments satisfying some constraints. A CSP defines a set of variables $X = \{x_1, \ldots, x_v\}$, each one with a specific domain $D_i$, and a set of constraints $C = \{C_1, \ldots, C_u\}$ restricting the values that variables can simultaneously take. A solution to a CSP is an assignment of values $d_i \in D_i$ to $x_i$, $1 \leq i \leq v$ that satisfies all the constraints.

Given the assumptions explained above, we can consider several sets of variables for the MCMPP:

- **Assignments of tasks to UAVs.** As some tasks could be Multi-UAV, these variables are represented as a binary array of size $p \times q$.
- **Orders,** which define the order in which each UAV performs the tasks assigned to it.
- **Assignments of UAVs to GCSs,** giving the control of each UAV to a concrete GCS during the mission.
- **Path Flight Profiles,** setting the flight profile that the vehicle must take for the path performance.
- **Return Flight Profiles,** similar to the previous set of variables but for the return path of each UAV.
- **Sensor used in the task performance.** These variables set the sensor of the vehicle that will be used during the task performance. It will be necessary to consider these variables just in the case that the vehicle performing the task has several sensors that could perform that task.
Finally, there are several constraints related with the different complexity issues explained:

- **Sensor constraints**: they check if a UAV has the sensors needed to perform its assigned tasks.
- **Order constraints**: they assure that the values of the order variables are less than the number of tasks assigned to the UAV performing that task.
- **GCS constraints**: they assure that the GCSs assignments are correct. UAVs are assigned to GCSs able to control them, and they are located within the GCS coverage area.
- **Temporal constraints**: they assure the consistency of all the times and durations involved in the mission planning.
- **Dependency Constraints**: these constraints are related to vehicle and time dependencies between tasks. **Vehicle dependencies** consider if two tasks must be assigned to the same UAV or different UAVs. Time dependencies consider the order and times of performance for two concrete tasks, using the Allen’s Interval Algebra[1].
- **Autonomy constraints**: they assure that the total flight time for each vehicle is less than its vehicle autonomy time.
- **Distance constraints**: they assure that the distance traversed by each vehicle is less than its range.
- **Fuel constraints**: they assure that the fuel consumed by each vehicle is less than its initial fuel.

4 KNEE-POINT BASED EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION FOR MPPS

To deal with the large amount of constraints and the large search space of the problem, there are several approaches that deal with MOEAs in highly complex problems [15] and others that consider hybridizing Multi-Objective Genetic Algorithms (MOGAs) and CSPs [6]. In a previous work, a hybrid approach based on MOGA and CSP for the MCMPP was proposed [10]. In our work, an extension of this approach has been developed in order to reduce the number of solutions obtained and look for Knee-points instead of non-dominated solutions. To do this, the cone-domination concept explained in section 2.1 will be used to replace the standard Pareto-domination.

This approach is based on an extension of NSGA-II[4], where the CSP is considered inside the fitness function, checking that the solutions fulfill all the constraints. When a solution does not fulfill some of the constraints, the number of constraints checked until fail in the propagation phase of the CSP is used as a weighted penalty function in order to reproduce the solutions fulfilling the highest number of constraints.

On the other hand, we change the non-dominated ranking used by NSGA-II with the non-cone-domination ranking with a specific angle $\phi$, which will be passed as a parameter of the algorithm. In the following subsections, a brief description of the encoding and the fitness function is presented and the new algorithm is described.

4.1 Encoding

The encoding of this approach consists of six different alleles representing the CSP variables described in the previous section:

1. **UAVs assigned to each task**:.

   - **Task orders**: This variable is represented as a permutation of numbers indicating the orders of the tasks.
   - **GCSs controlling each UAV**.
   - **Flight profiles used for each UAV to each assigned task**.
   - **Sensors used for the task performance** by each UAV.
   - **Flight Profiles used by each UAV to return** to the base.

If the $T_i$ task is Multi-UAV, then the corresponding cell of each allele contains a vector representing the different UAVs, flight profiles or sensors assigned to this task.

In the reproduction phase of the algorithm, we apply a 2-point crossover and an uniform mutation to the first to the sixth alleles; the Partially-Matched Crossover (PMX) and an Insert Mutation are applied to the second allele.

4.2 Fitness function

The fitness function examines that all constraints are fulfilled for a given solution. If not, it stores the number of constraints fulfilled by the solution. If all constraints are fulfilled, then the fitness works as a multi-objective function minimizing the objectives of the problem. These objectives are, in order of importance:

1. **The total cost** of the mission.
2. **The end time of the mission** or **makespan**.
3. **The risk** of the mission, which is computed as an average percentage indicating how risky the mission is (e.g. UAVs finishing the mission with low fuel, UAVs flying near to the ground or UAVs flying close between them).
4. **The number of UAVs** used in the mission.
5. **The total fuel consumption**.
6. **The total flight time**.
7. **The total distance traversed**.

4.3 Algorithm

The Knee-Point based NSGA-II (KPNSGA-II) for Mission Planning Problems is presented in Algorithm 1. In this new approach, firstly the initial population is randomly generated (Line 1) and the convergence factors initialized (Lines 2-4). On the other hand, the maximum and minimum values for each objective is initialized to a vector of $m$ zeros (Line 5) and to the vector $\mathbf{M}$ of maximum objective values (Line 6), respectively. These values will be updated with each solution evaluated and then used to normalize the objectives.

The evaluation of the individuals is performed by the fitness function (Lines 8-17), where after checking the solution with the CSP model (Line 9), if the solution is valid, then the Multi-Objective values are computed and stored inside the fitness (Line 11). In addition, as has been explained before, the maximum and minimum objective values are updated using this solution (Lines 12-13). If the solution is not valid, then the number of fulfilled constraints is stored inside the fitness (Line 16).

Following the NSGA-II approach, the new population is created using the BuildArchive function (Line 18). In this function, an array of vectors containing the solutions grouped by their level of non-dominance is created. In this array, the first vector will contain the non-dominated solutions of the population; the second one, the non-dominated solutions among the rest of the population without the solutions of the first vector; the third one, the non-dominated
solutions among the population without the solutions of the first and second vectors, and so on.

Here we consider the non-cone-dominated solutions instead of the dominated ones, as was explained in section 2.1. This new approach of cone-domination is shown in Algorithm 2. In this function, as can be seen, a \( \phi \) value indicating the angle of the cone domination is needed. This function first checks for non-valid solutions, giving the dominance to the one fulfilling more constraints. Then, if both solutions are valid, it normalizes the objective vectors with the maximum and minimum values. Finally, it computes the cone domination function, following equations 5 and 14 for each objective; and checks if the second solution is cone-dominated by the first.

Algorithm 2: ConeDominate\((A, B, \phi, maxP, minP)\)

Input: Solutions A and B to check for cone-domination. The angle \( \phi \) (in degrees) for every face of the cone. Vectors \( maxP \) and \( minP \) containing the actual maximum and minimum values for all \( m \) objectives.

Output: TRUE if A dominates B, FALSE otherwise.

1. if \( A.f\text{.fit.validConst} > B.f\text{.fit.validConst} \) then
   2. \(
   \text{return TRUE}
   \)
3. else if \( A.f\text{.fit.validConst} < B.f\text{.fit.validConst} \) then
   4. \(
   \text{return FALSE}
   \)
5. \( x \leftarrow A.f\text{.objectives} - minP \)
6. \( y \leftarrow B.f\text{.objectives} - minP \)
7. \( \text{dominates} \leftarrow \text{FALSE} \)
8. for \( i \leftarrow 1 \) to \( m \) do
   9. \( \text{cone1} \leftarrow x[i] \)
10. \( \text{cone2} \leftarrow y[i] \)
11. for \( j \leftarrow 1 \) to \( m \) do
12. \( \text{cone1} \leftarrow \text{cone1} + \tan\left(\frac{\phi - 90}{2}\right) \cdot x[j] \)
13. \( \text{cone2} \leftarrow \text{cone2} + \tan\left(\frac{\phi - 90}{2}\right) \cdot y[j] \)
14. if \( \text{cone1} < \text{cone2} \) then
15. \( \text{dominates} \leftarrow \text{TRUE} \)
16. else
17. \( \text{return FALSE} \)
18. \( \text{return dominates} \)

After the array of vectors containing the ranked solutions is created, similarly to NSGA-II, a sparsity value based on the crowding distance is given to each solution at every vector. Then, a tournament selection (Line 25) is used to select the individuals for the genetic operators. The crossover operator (Line 26) consists of an extension of the 2-point crossover and the PMX independently applied to each allele of the chromosome. The mutation operator (lines 27-28) is also an extension of the uniform and the insert mutation applied to each allele (as explained in section 4.1). Finally, the stopping criteria compares the non-dominated solutions obtained so far at each generation with the solutions from the previous generation (Lines 19-23). If the obtained solutions remain unchanged for a number of generations, then the algorithm stops and returns these solutions as an approximation of the POF.

5 EXPERIMENTS

In the experiments, we test the newly implemented KPNNSGA-II. KPNNSGA-II needs an angle as input, which is used to extend the angle of cone-domination (see Figure 3) used in the algorithm (the angle in the standard Pareto approach is 90 degrees). In these experiments, we compare the NSGA-II approach with KPNNSGA-II
using 120, 135 and 150 degree angles (which we called KPNSGA-II-120, KPNSGA-II-135 and KPNSGA-II-150, respectively).

A real mission with 7 objectives, one of them Multi-UAV, 5 UAVs, 3 GCSs and 3 NFZs (see Figure 4) is used in the experiments.

In the experiments, we examine the optimization problem with different number of objectives (we increase the number of objectives for each test). The test with two objectives optimizes cost and makespan; the test with three objectives optimizes cost, makespan and risk; the test with four objectives optimizes cost, makespan, risk and number of UAVs; etc. Each experiment has been executed 10 times, and the mean and standard deviation are recorded. The population of the algorithm has been set to 200, the maximum number of generations to 300, the stopping criteria to 10 and the mutation probability to 5%.

In order to compare the results, we compute the hypervolume with the normalized objectives for each solution set. The maximum point (1, 1, 1, ... ) is set to be the reference point. The results are shown in Table 1. Table 2 shows the number of solutions obtained for each approach.

Table 1: Hypervolume of the solutions obtained by NSGA-II and KPNSGA-II with 120, 135 and 150 cone angles.

<table>
<thead>
<tr>
<th>No. Objectives</th>
<th>NSGA-II</th>
<th>KPNSGA-II-120</th>
<th>KPNSGA-II-135</th>
<th>KPNSGA-II-150</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.644±0.018</td>
<td>0.616±0.032</td>
<td>0.624±0.026</td>
<td>0.558±0.029</td>
</tr>
<tr>
<td>3</td>
<td>0.644±0.008</td>
<td>0.738±0.021</td>
<td>0.751±0.016</td>
<td>0.534±0.041</td>
</tr>
<tr>
<td>4</td>
<td>1.025±0.035</td>
<td>0.633±0.026</td>
<td>0.623±0.015</td>
<td>0.578±0.073</td>
</tr>
<tr>
<td>5</td>
<td>1.265±0.010</td>
<td>1.024±0.055</td>
<td>1.018±0.054</td>
<td>0.990±0.060</td>
</tr>
<tr>
<td>6</td>
<td>1.125±0.026</td>
<td>0.698±0.035</td>
<td>0.659±0.021</td>
<td>0.848±0.816</td>
</tr>
<tr>
<td>7</td>
<td>1.032±0.053</td>
<td>0.697±0.095</td>
<td>0.580±0.057</td>
<td>0.554±0.032</td>
</tr>
</tbody>
</table>

As can be observed, the results obtained with the new KPNSGA-II algorithm highly reduce the number of solutions while the loss of hypervolume is pretty low. An example representation of the hypervolume obtained with the 2-objective approach is presented in Figure 5. Here, it is notable that the points in red (solutions of the NSGA-II approach) in the top centre of the figure reduce slightly the cost while increasing much more the makespan. These solutions are omitted by KPNSGA-II due to their low significance.

Figure 6 shows a scatter-plot for the solutions obtained with the 3-objectives problem. Here, it is also notable that some solutions obtained by NSGA-II in the upper part of the region are omitted with KPNSGA-II.

If we look at the results obtained from 4-objectives onwards, it can be observed that the loss of hypervolume is higher. This is because of the forth objective (the number of UAVs) which is an integer variable in the range [1, 5]. Therefore, a change in its value has a higher impact on the hypervolume. Nevertheless, as can be seen, as the number of objectives increases, the hypervolumes for KPNSGA-II gets better, because the influence of the number of UAVs gets smaller.

In addition, Table 3 shows the number of required generations to converge. We can observe that the KPNSGA-II algorithm converges faster than the others. Concretely, the higher the cone angle, the quicker the convergence, specially with many objectives, where the NSGA-II approach was not even able to converge within the maximum number of generations. Additionally, it is notable that the required number of generations to converge when increasing the number of objectives remains similar for KPNSGA-II, which reveals that this approach is time efficient.

Comparing the results obtained with the different angles, we can conclude that the approach with 150 degrees returns very few
solutions, which is only good if the operator does not want to make a hard decision, but let the algorithm decide. Nevertheless, as most of the times the operator wants to see a reasonable number of solutions before making a decision, the approaches with 120 and 135 degrees can deliver more solutions. If some additional decision making process is performed after the algorithm, such as ranking or filtering the solutions, then the approach with KPNSGA-II-120 will be better because of the higher number of solutions with lower hypervolume loss. Nevertheless, if no decision making process is made, KPNSGA-II-135 could be better for the operator in order to reduce his decision making time.

6 CONCLUSIONS

In this work, we have presented an extension of the NSGA-II algorithm based on Knee Points in order to focus the search of the algorithm in significant solutions. For this, we have used the concept of cone-domination, which substitute the domination concept in the algorithm. In addition, we have presented and described the Multi-UAV Mission Planning Problem, a complex problem which is used to test this algorithm. This problem was modelled as a Constraint Satisfaction Problem (CSP), and then solved using the developed algorithm, where the encoding of the algorithm is the same as the variables of the CSP. For this problem, 7 optimization objectives were considered, and treated in the experimental phase increasingly, performing experiments from just 2 objectives to experiments with all the 7 objectives. In the experimental phase, three different angles (120, 135 and 150) were used for the cone-domination. The results showed that the KPNSGA-II approach returns a small number of solutions which maintain most of the hypervolume compared to NSGA-II. Concretely, it was shown that our new approach maintains a similar number of solutions as the number of objectives increase, while the hypervolume loss keeps acceptable values in comparison.

On the other hand, the results showed that the number of generations needed to converge is also improved for the new algorithm, outperforming the results obtained with NSGA-II for a high number of objectives. In this case, NSGA-II could not find the complete POF, while KPNSGA-II could converge.

In future works, we will test this algorithm with a benchmark of Mission Planning Problems of different complexity. In addition, to improve the DM process for the operator, we will also develop ranking methods for the solutions generated by KPNSGA-II.

Table 3: Number of generations needed to converge for NSGA-II and KPNSGA-II with 120, 135 and 150 cone angles.

<table>
<thead>
<tr>
<th>No. Objectives</th>
<th>NSGA-II</th>
<th>KPNSGA-II-120</th>
<th>KPNSGA-II-135</th>
<th>KPNSGA-II-150</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>85.±2.662</td>
<td>56.±2.167</td>
<td>51.±6.114</td>
<td>53.±1.845</td>
</tr>
<tr>
<td>3</td>
<td>139.±4.37</td>
<td>76.±1.151</td>
<td>64.±4.189</td>
<td>61.±7.111</td>
</tr>
<tr>
<td>4</td>
<td>276.±2.76</td>
<td>78.±2.124</td>
<td>63.±3.203</td>
<td>53.±2.532</td>
</tr>
<tr>
<td>5</td>
<td>300.±0.00</td>
<td>58.±1.08</td>
<td>64.±1.144</td>
<td>55.±7.166</td>
</tr>
<tr>
<td>6</td>
<td>300.±0.00</td>
<td>64.±2.44</td>
<td>53.±9.14</td>
<td>55.±3.102</td>
</tr>
<tr>
<td>7</td>
<td>300.±0.00</td>
<td>71.±2.19</td>
<td>66.±3.87</td>
<td>55.±9.158</td>
</tr>
</tbody>
</table>

Cristian Ramirez-Atencia, Sanaz Mostaghim, and David Camacho

ACKNOWLEDGMENTS

The authors would like to thank the support obtained from Airbus Defence & Space, specially from SAVIER Open Innovation project members: José Insensier, Gemma Blasco and César Castro.

The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. This work is supported by Airbus Defence & Space under project SAVIER (FUAM-076914 and FUAM-076915); Spanish Ministry of Economy and Competitiveness under project EpHEMECH (TIN2014-56494-C4-4-P) and Comunidad Autónoma de Madrid under project CIDERCINE (CAM grant S2013/ICE-3959), both under the European Regional Development Fund FEDER, and European Union’s Justice Program (2014-2020) under RiskTrack project (723180) – JUST-2015-JCOO-AG-JUST-2015-JCOO-AG-1. The contents of this publication are the sole responsibility of their authors and can in no way be taken to reflect the views of the European Commission.

REFERENCES