DSSO:

DISTRIBUTED SATELLITE SYSTEMS MODEL-BASED ARCHITECTING FOR EARTH OBSERVATION MISSIONS

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

David Llaveria Godoy

In partial fulfilment

of the requirements for the degree in

TELECOMUNICATIONS SYSTEMS ENGINEERING

Advisor: Eduard Alarcón, Carles Araguz

and Elisenda Bou

Barcelona, June 2017

Abstract

As many things in the technological world, satellites have been decreasing in size and cost and they have become part of our life. Many research centers and private companies are developing satellite constellations, large groups of satellites that will work together towards a common mission.

This project focuses on designing a framework that will help the mission designers to decide which constellation is the optimal one for their specific missions. The thesis implements an optimization framework that evaluates the performances and some qualitative attributes of the constellations. Moreover, a Matlab tool capable of choosing between millions of architectures has been implemented.

Furthermore, this framework has been used in a specific use-case study, the marine weather forecast. This study shows that the chosen architecture not only depends on the number of satellites and their orbital configuration but also on the instruments that are embarked on them, the requirements of the use-case and on the cost.

Resum

Com d'altres aspectes en el mon tecnològic, els satèl·lits han anat disminuint la mida i el cost i s'han convertit en part de les nostres vides. Molts centres de recerca i companyies privades desenvolupant constel·lacions de satèl·lits, grans grups de satèl·lits que treballen junts perseguint la mateixa missió.

Aquest projecte es centra en el disseny d'un framework que ajudarà als dissenyadors de missions espacials a decidir quina constel·lació es la òptima per a determinades missions. Aquesta tesi implementa un framework d'optimització que avalua el rendiment i alguns atributs qualitatius d'aquestes constel·lacions. A més, una eina de Matlab capaç d'escollir entre milions d'arquitectures ha sigut implementada.

Finalment, aquest framework s'ha utilitzat en l'estudi d'una missió especifica, com es previsió meteorològica marítima, per a provar-lo. Aquesta tesi mostra que l'arquitectura escollida no nomes depèn del nombre de satèl·lits i de la seva configuració orbital, si no que també dels instruments específics embarcats en ells, dels requeriments de la missió d'estudi i del seu cost.

Resumen

Como otros aspectos en el mundo tecnológico, los satélites han ido reduciendo su tamaño y su coste y se han convertido en parte de nuestras vidas. Muchos centros de investigación y compañías privadas están desarrollando las llamadas constelaciones de satélites, grandes grupos de satélites que trabajan juntos persiguiendo una misma misión.

Este este proyecto se centra en el diseño de un framework que ayudará a los diseñadores de misiones espaciales a decidir que constelación es la óptima para determinadas misiones. Esta tesis implementa un framework de optimización que evalúa el rendimiento y algunos atributos cualitativos de estas constelaciones. Además, una herramienta de Matlab capaz de escoger entre millones de arquitecturas ha sido implementada.

Finalmente, este framework se ha utilizado en el estudio de una misión específica, previsión meteorológica marítima, para probarlo. Esta tesis muestra que la arquitectura escogida no solo depende del número de satélites y su configuración orbital, sino que también de los instrumentos embarcados en ellos, de los requisitos de la misión y de su coste.

To Evelt for her continuous support and love and
for encourage me to pursue my goals.

Acknowledgements

First, I would like to thank to professor Eduard Alarcon, Carles Araguz and Elisenda Bou for giving me the opportunity to participate in a European project related to space and for let me work within their team as one member more. This project had been an amazing experience, and has taught me a lot of what is a real project and gave me a push to my space motivation.

I would also like to give my gratitude to professor Eduard Alarcon because he has initiated me in a real research project and he has helped me on everything. His enthusiasm in all what he does let me to really enjoy working with him.

Moreover, Carles Araguz deserves a special mention too, because he has been my colleague all the time, I really appreciated the opportunity to work with him, and he has contributed to my work with his own work and his experience on the field. I spend a lot of hours working hand in hand with him.

I would like to give my gratitude to all the members of the NanoSat Lab, from professor Adriano Camps to all the members that work on it for help me whenever I need and for be my co-workers during the project. I enjoyed the experience with them.

Revision history and approval record

Revision	Date	Purpose
0	01/06/2017	Document creation
1	08/06/2017	Document revision
2	12/06/2017	Document revision
3	19/06/2017	Document revision
4	20/06/2017	Document revision
5	24/06/2017	Document revision
6	29/06/2017	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
David Llaveria	dllaveriag@gmail.com
Eduard Alarcón	
Elisenda Bou	
Carles Araguz	

Written by:		Reviewed and approved by:	
Date	01/06/2017 – 24/06/2017	Date	29/06/2017
Name	David Llaveria	Name	Eduard Alarcón
Position	Project Author	Position	Project Supervisor

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Table of contents

List of Figures

List of Tables

1. Introduction

In the last decades, observing the earth has become a trend in our lives in many technological, agricultural and societal endeavours. Nowadays, satellite imagery and data is pervasive and used in many dissimilar contexts, such as weather forecast, oil spill monitoring or natural resource management [1]. Science has benefited from data captured in space for a myriad of applications: monitoring of terrain displacement [2], biomass estimation [3] and fire monitoring [4]. The requirements demanded for these applications, like the spatial resolution or the revisit time, have been increasing year by year. Revisit time, for instance, is one of the most demanding requirements. It is possible for some applications that one image of the same place on Earth must be refreshed every 3 hours, something that single satellite architectures are not capable of.

This requirement has led to the distributed satellite systems (DSS) concept. A DSS is a system of satellites that work together to fulfil a common goal. There are different approaches towards DSS, such as constellations, clusters, trains swarms, federated satellite systems (FSS) or fractionated satellite systems (fracsats). They are briefly described below.

Constellation: Constellations are DSS formed by groups of satellites orbiting independently, their number of units and their orbits are designed to achieve continuous global coverage. Some of the most famous constellations are the Global Positioning Systems (GPS) [5], The Global Navigation Satellite System (GLONASS) [6], the Disaster Monitoring Constellation (DMC) [7], a system with four satellites that provides high frequency imaging anywhere on the globe or the Iridium one [8], a satellite communications constellation with 66 satellites orbiting at the same time.

Cluster: On the other hand, the spacecraft of a clusters orbit in close formation. These satellites exchange data in order to maintain their configuration, which is often required by their observational requests. TerraSAR-X/TanDEM-X [9] and FASTRAC [10] are examples of this concept.

Train: Trains are coordinated groups of satellites that closely follow each other along the same orbital track. Examples of these are Afternoon train [11] and the Morning constellation [12].

Swarm: A satellite Swarm is a network of interconnected satellites that do not require or maintain a certain formation. Swarms are envisioned as a large group of satellites, in the order of hundreds. Despite this concept still being explored, the European Space Agency (ESA) demonstrated their feasibility with the project SWARM, in 2010 [13].

Fractionated spacecraft: A completely different, and novel, approach is the one of fractionated spacecraft, in which satellites are built from physically detached modules [14]. In a fractionated spacecraft, several modules would orbit in close formation and would wirelessly share their resources with the satellite infrastructure (e.g. ground link bandwidth, processing capabilities, or even power). The F6 project [15] was supposed to demonstrate the feasibility and cost-effectiveness of fractionated spacecraft but it was cancelled in 2013 due to the immaturity of all their required technologies.

Federated Satellite Systems: FSS essentially consist in satellite networks trading previously inefficiently allocated and unused resource commodities such as downlink bandwidth, storage, processing power and instrument time. FSS try to circumvent the underutilization of expensive space assets in already existing missions [16] [17].

In addition, remote sensing techniques have evolved and have been improving both the quality of the measurements and the cost of the Earth Observation systems and technologies. One of such advancements is the appearing of new satellite platforms and design concepts, namely, small satellites and miniaturized payloads. This allows the creation of bigger and cheaper constellations, such as Cyclone Global Navigation Satellite System (CYGNSS) [18], a constellation with 8 small satellites.

This thesis (DSSO) is aimed at studying these types of systems and to propose a high-level design methodology based on optimization that could aid future mission implementers in their endeavours.

1.1. Statement of purpose

The main purpose of this project is to provide a framework capable of selecting the optimal DSS in terms of performance and cost. This project has two parts, the first one, a mathematical optimization framework that defines the whole procedure used for the selection. This framework, in turn, is divided into three main sub-parts, namely:

- Generation of a set of architectures from a group of DSS archetypes, these architectures are different combinations of all the design variables and covers the full design space region studied.

- Computation of cost and assessment of the quality attributes of these architectures.

- Aggregation of their characteristic figures to derive a single score that can be used to compare the architectures and select the most optimal one.

If the first part is the mathematical framework, the second part is the analysis of results, and the study of the design space. Ultimately, this optimization framework and analysis can also be used to design future DSS architectures.

DSSO provides the procedure for the selection of the outcoming architecture and a Matlab software tool ready to generate and work with more than five million architectures with different platform size combinations and sensor combinations, with such amount of information to process, the software has to be efficient and computationally optimized in order to do not have an untreatable problem. Moreover, this project gives the final results and the best architectures of the studied ones.

1.2. Methods and procedures

Despite the fact that the framework hereby presented is implemented from scratch, its design leverages from previous studies and analysis of DSS capabilities carried out by the author during the Advanced Engineering Projects semester. In addition to this, several generic optimization approaches have been combined to be able to achieve the goals of this thesis. Some of these methods and general concepts can be read in [19], [20], [21] and [22]. Furthermore, some optimization examples can be read in [23] and [24] to which the reader is directed for further details.

1.3. Work plan

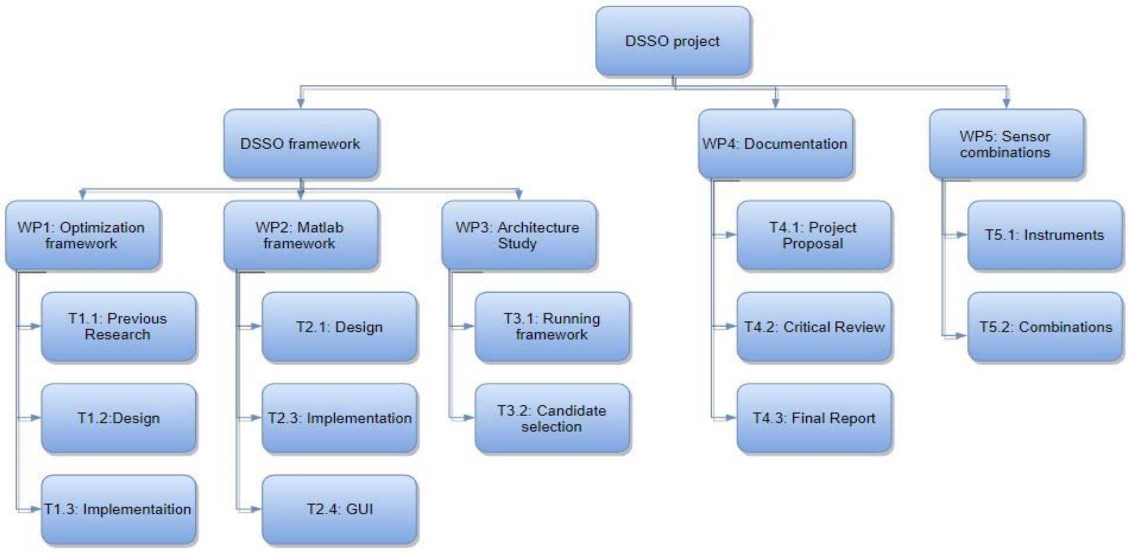


Figure 1. DSSO Work Packages

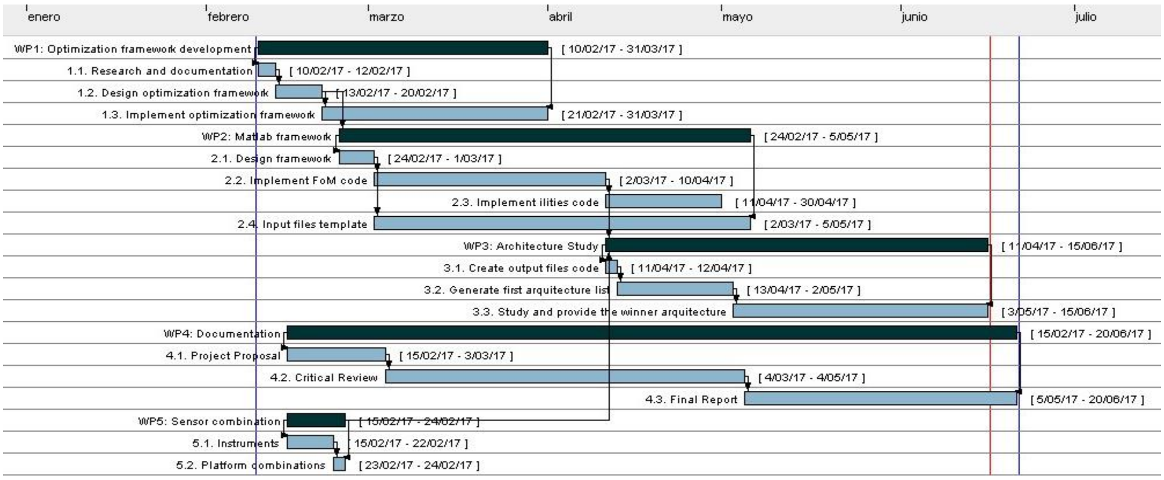


Figure 2. DSSO Gantt diagram.

For a detailed description of the work packages and milestones, see the appendix V.

1.4. <u>Changes from the initial work plan</u>

Given that this project has been developed in the frame of a H2020 Research project, the workplan has been constantly adjusted to the project's scheduled deliverables and planning modifications. Some of these modifications, also, affect the inner structure of this project and add new tasks that were not planned before.

One of these changes has led to a large number of architectures that gives some troubles with the hardware limitations. Therefore, the Matlab framework has been optimized, at first, SQL database was the chosen option, it works but the simulation last more than two hours, with that processing time, it was impossible to made minor changes in the process, therefore, the final Matlab framework simulation takes around ten minutes of processing.

The schedule has also suffered modifications for the distributed satellite system simulator part. The final requirements and definition for this part, have yet not been addressed by the involved partners in the research project and are, therefore, out of the scope of this thesis.

2. Background:

2.1. ONION project

This thesis has been carried out in the frame of the *Operative Network of Individual Observational Nodes* (ONION) project, funded by European Commission's Horizon 2020 research and innovation programme (under grant agreement No. 687490). The work presented in this report has been developed at the Nano-Satellite and Payload Laboratory of the Technical University of Catalonia, as one of the partners of the ONION consortium.

The main objective of ONION is to enable mission designers and implementers to decide which distributed satellite architectures to develop for competitive imaging from Space, and establish the requirements for communications support. The ONION concept is proposed to supplement, in an incremental way, some of the currently available European Earth Observation infrastructures, like Copernicus. Such complementary approaches are envisioned to contribute to maintaining European competitiveness in serving future scientific needs.

The ONION project unfolds into five objectives, namely:

- To review the emerging fractionated and federated observation system concepts.
- To identify potential benefits to be obtained considering observation needs in different Earth Observation domains.
- To identify key required technology challenges, to be faced in Horizon 2021-2027.
- To validate observation needs with the respective user communities to be fit for purpose in terms of scientific and commercial applications.
- To propose an overall strategy and technical guidelines to implement such concepts at Horizon 2021-2027.

The work presented in this report is encompassed within ONION's Task 3.4, devoted to select the candidate architecture which the consortium will design in subsequent Task 3.5. In order to do so, T3.4 (and hence, this thesis), leverages on a previous exploration of the architectural tradespace performed in Task 3.2 of the ONION project, and provides data to Task 3.3. This latter task is devoted to study the architectures identified in T3.4 with refined subsystem and instrument models and custom mission-analysis software tools. Both the exploration of the design space and the detailed analysis of architectures are out of the scope of this thesis.

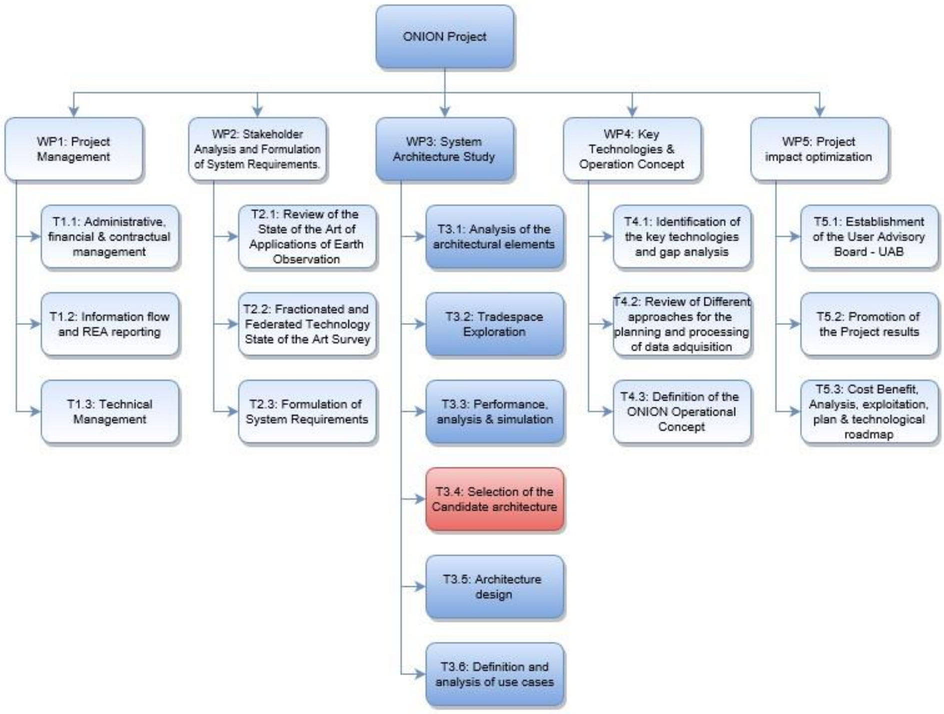


Figure 3. ONION Breakdown Structure.

Task 3.4 takes a set of constellation archetypes that has been generated in 3.2 task and simulated in 3.3 task and, as an output, returns the selected candidate. In order to satisfy the required activities planned for T3.4, the design of the architecture selection framework has been split in the following stages:

- *Sensor survey and architecture generation:* The stage is sub-divided in two parts. The first, Selection of instruments archetypes, uses the inputs coming from task 3.6 and chooses and combines the instruments that will be used for the specific use-case study. The output of this task is a set of sensor combinations that will be placed on the different satellites of the architectures. The second part of this stage combines the DSS archetypes from task 3.2 and the set of sensor combinations and generates a set of architectures (more than 5.5 million architectures for the use-case studied in this thesis.)
- *DSSO pre-selection*: The pre-selection stage is sub-divided in five parts. Four of them works separately and give the inputs to the last one. These parts use the outputs of the previous sections and of task 3.2 and 3.6 and, through the optimization framework, give the ranking of the architectures in order to selected some of them for the refined analysis of task 3.3[1].
- *DSSO Selection of optimal candidates*: In this stage, the re-simulated architectures coming from task 3.3 and uses again the optimization framework to select the optimal architecture that will be designed in task 3.5.
- *Candidate study*: The final candidate is studied through a sensitivity analysis in order to see how robust is the selection. Moreover, a study of the better launch strategy will be performed for the candidate.

In the next figure, it is shown the diagram of task and its interconnection with other tasks of ONION. Marked in red, the tasks fulfilled by the DSSO framework.

[1] Task 3.3 will perform in depth analysis of the pre-selected architectures that the ones did before. That is done in two iterations because simulate 5.5 million architectures will take several months of processing and it is not feasible. The refined analysis will be performed for 20-30 architectures.

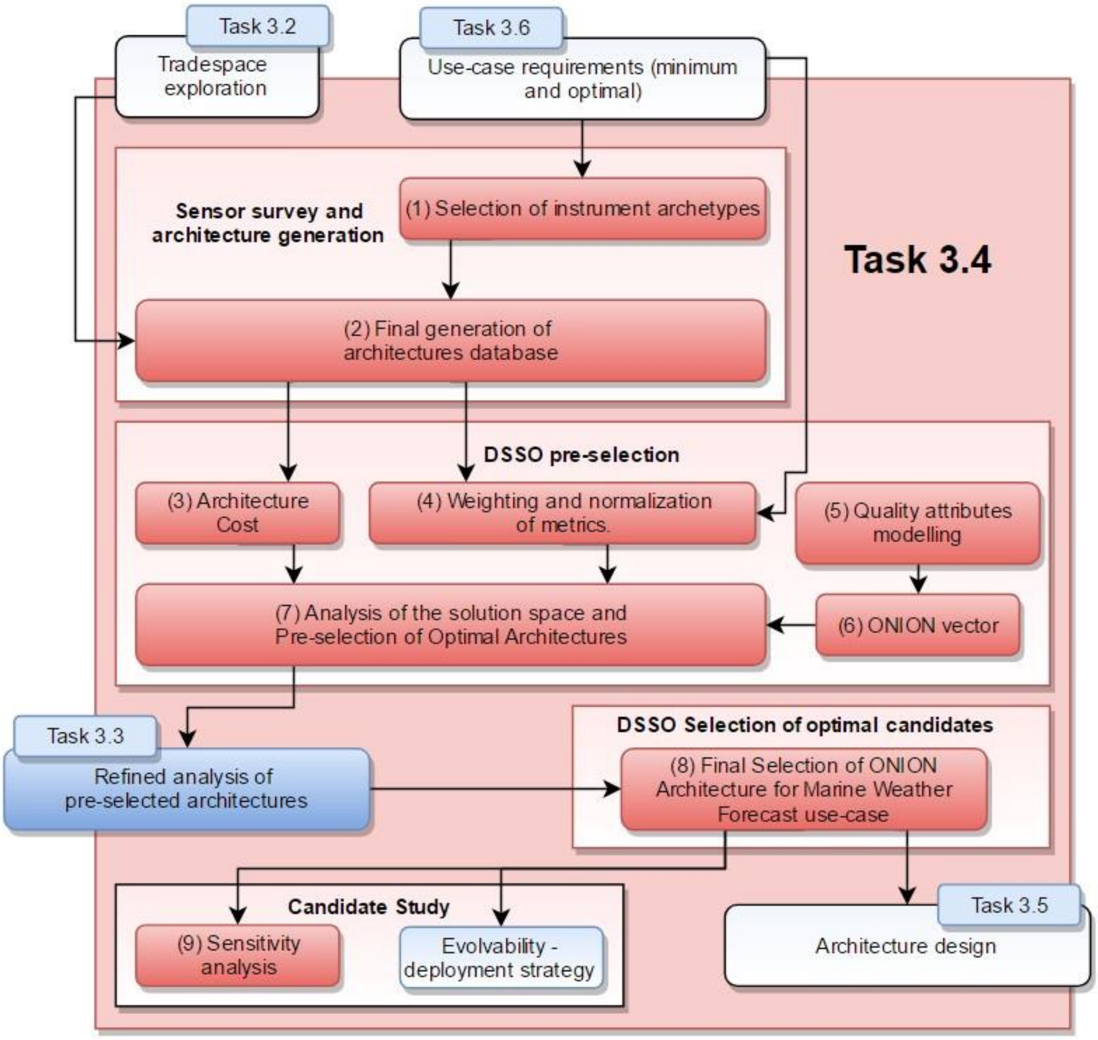


Figure 4. DSSO interface and information flow in the context of ONION WP3 tasks.

3. Optimization procedure:

DSSO has as its primary goal the selection of the most optimal architecture for a given use-case. For the test of the framework and as the first use-case studied by ONION, the Marine Weather Forecast use-case has been chosen. The optimization framework is based upon an aggregated figure of merit that encompasses:

- Performance requirements of the final application as defined by the use-case: revisit time, latency, spatial resolution and accuracy. They can be seen in table 13.
- Costs.
- Qualitative architectural attributes that are assessed numerically (e.g. robustness, versatility, maturity, practicality…)
- Strategic decisions enforced by the consortium partners (including, but not limited to, the weighting of performance metrics and qualitative attributes) plus other filters.

This procedure follows the scheme in the figure 4. The numbers near the title of some sections corresponds to the numbers in brackets in the figure 4.

3.1. Figure of Merit

3.1.1. Notation

The FoM is calculated for each of the architectures (ij) that come from one implementation (j) from one of the archetypes (i), a platform size distribution and a sensor combination. These architectures can perform a set of measures (k) from the use-case. For each measure, a specific figure of merit is computed (Γ_{ijk}), and subsequently these ones are aggregated to compute the final FoM (Γ_{ij}).

Variable	Description
$\boldsymbol{i}$	Archetype generated in the tradespace exploration.
$\boldsymbol{j}$	Architecture implementation that embarks a given sensor combination in its platforms.
$\boldsymbol{k}$	A measurement defined in the use-case.
$\boldsymbol{\Gamma_{ijk}}$	Measurement-specific figure of merit of an architecture.
$\boldsymbol{\Gamma_{ij}}$	The overall figure of merit of an architecture (cost and modifiers not included).
$\boldsymbol{N_K}$	Total number of measurements defined in use-case.

Table 1. Notation summary.

3.1.2. Figure of merit without cost.

The FoM of each architecture is computed as the root mean square of the specific figures of merit of each measurement of the use-case.

$$\boldsymbol{\Gamma_{ij} = \sqrt{\frac{1}{N_K} \cdot \sum_{k} \Gamma_{ijk}^2}}$$

The specific figure of merit of each measurement is calculated from an aggregation of normalized values and their respective weights.

$$\Gamma_{ijk} = \prod\nolimits_{m} q(m)^{\gamma_m}$$

Where m is the metric value, $q(m)$ is the normalized value and γ_m is the respective weight.

3.2. Normalization function (4)

To have the values of the inputs at the same rank of values, namely between 0 to 1, where 1 is the best score, it is needed a normalization process for each input parameter. So, considering the input parameter, as well as the required and the optimal values for each metric, a mix of an exponential score function and the Wymore's score function [21]. That is an exponential function but limited by the optimal and required values as the Wymore ones. And, also, with a minimum score different from zero in order to not have an aggregated metric of zero when the required parameters are not met, but the architecture can perform the measure. Zero will apply only for those architectures that are not suitable to do the measure.

Two cases are defined for these functions, the increasing and the decreasing, depending on the desired values. If the optimal value is larger than the required one, we have a increasing function, conversely, if we have an optimal value smaller than the required one, we have a decreasing function.

The decreasing function is:

$$q(m) = \begin{cases} Q + \dfrac{1 - \exp\left(-\dfrac{m_b - m}{\rho}\right)}{1 - \exp\left(-\dfrac{m_b - m_a}{\rho}\right)} \cdot (1 - Q) & \text{when } m_a \leq m \leq m_b \\ 1 & \text{when } m < m_a \\ Q & \text{otherwise} \end{cases}$$

The increasing function is:

$$q(m) = \begin{cases} Q + \dfrac{1 - \exp\left(-\dfrac{m - m_b}{\rho}\right)}{1 - \exp\left(-\dfrac{m_a - m_b}{\rho}\right)} \cdot (1 - Q) & \text{when } m_b \leq m \leq m_a \\ 1 & \text{when } m > m_a \\ Q & \text{otherwise} \end{cases}$$

Where Q is the chosen minimum score of the normalization. m_a is the worst accepted value, the required one, and m_b is the better value, the optimal one. Finally, the ρ coefficient adjusts the exponential response of $q(m)$ and is defined as a fraction of the normalization range.

$$\rho = \frac{m_b - m_a}{P}$$

Depending on the value of P the normalization function has different shapes, as shown in the next figure. For the study carried out in DSSO the Q value is set to 0.1 and the P value is set to 3 and the values of m_a and m_b are defined from the user requirements identified in [25].

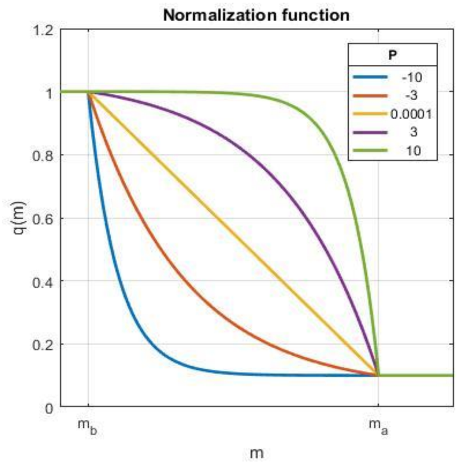


Figure 5. Metric normalization function for different P values.

3.3. Weights (4)

The next step of the optimization procedure is to weight the influence on the figure of merit. Weights are normalized with the next equation.

$$\gamma_m = \frac{w_m}{\sum w_i}$$

Where w_m is an integer value that represents the non-normalized weight of a given metric. That value represents the importance that those who are carrying out the study give to each of the metric and could be any positive number. For the study in DSSO homogenous weighting is used, the three metrics has γ_m equal to 0.33.

This equation is a generalization of the Rank Sum Weights equation, where Rank Sum Weights only allows to rank all the metrics in order, the used equation allows to give the same rank to more than one metric and has more granularity, for example one can give a scores of 100, 100 and 75 to three metrics, and for the Rank Sum Weights [24] the ranks must be 1, 2 and 3.

3.4. Cost normalization (3)

When an optimization procedure is carried out, the economic cost of each candidate is important in order to choose the very optimum one, so, it must be considered in the final ranking. In the study in DSSO the architectures have costs from 0.4 to 1000 million USD, which has a significant impact on the final ranking, therefore a compression of the dynamic range is needed. After different approaches, the cost is normalized with the same function as the rest of metrics, but with other parameters, and it is added to the FoM as a multiplicative parameter with a weight of 1.

$$\boldsymbol{\Gamma}_{\boldsymbol{ij}_{\boldsymbol{cost}}} = \boldsymbol{\Gamma}_{\boldsymbol{ij}} \ . \ q(Cost)$$

For the study in DSSO the normalization parameters applied to the cost are shown in next table.

Parameter	Value	Remarks
$\boldsymbol{P}$	0.0001	$q(x) \simeq -mx + n$
$\boldsymbol{m_a}$	Maximum cost in database.	
$\boldsymbol{m_b}$	Minimum cost in database.	

Table 2. Cost normalization parameters.

3.5. Ilities (5)

The final FoM includes a set of multiplicative modifiers that evaluates qualitatively some attributes of the architectures. These values are normalized values that are weighted and subsequently multiplied together to the FoM.

$$\boldsymbol{\Gamma'}_{\boldsymbol{ij}_{\boldsymbol{cost}}} = \boldsymbol{\Gamma}_{\boldsymbol{ij}_{cost}} \ . \ A$$

$$Where: \qquad A = \prod_n \alpha_n$$

Being α_n the weighted modifier value.

In that case the weight of the modifier is the base of an exponential function and the normalized value of the modifier quality is in the exponent. Therefore, the smaller the weight the higher the impact on the final metric.

$$\alpha_n(ij) = b_n^{1-a_{nij}} \qquad a_{ij}, b_n \in [0,1]$$

Where a_{nij} is the value of the modifier *n* for the architecture *ij*, and b_n is the weight for that modifier. In the next figure, it can be seen the response of the α function for different values of *b*, with respect to variable *a*.

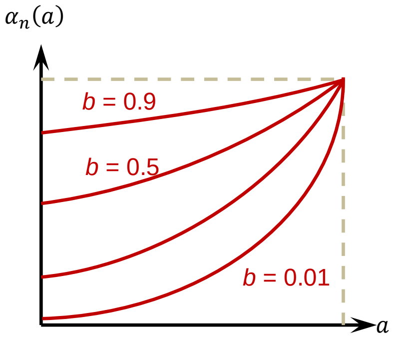


Figure 6. Modifier values for different weights.

It is recommended that the values of the weights for the modifiers should be within the range 0.5 to 1, being 1 a neutral modifier.

There are three classes of modifiers. Modifiers that are related to instrument characteristics, architectural modifiers and modifiers linked to the use-case description. The next table summarizes the meaning of the modifiers and their classes.

Var.	Description	Type
α_F	***Critical measurements:*** architectures that satisfy more measurements with high priority have better score.	Use-case
α_P	***Practicality:*** the need to process large amounts of data worsens the architecture score (e.g. constellation with 24 SAR is unfeasible from the data processing perspective).	Instrument
α_D	***Data relevance:*** based on sensing constraints (e.g. limited by cloud cover), and the relevance of their generated data with respect to a given variable. The more relevant the generated data is, the better score.	Instrument
α_V	***Versatility:*** architectures that can generally present better *sensor-agnostic* figure-of-merit have better score.	Architectural
α_M	***Maturity:*** maturity of the sensing technology. Architectures based on mature instruments have better score.	Instrument

Table 3. Summary of modifiers.

3.5.1. Critical Measurements

For each use-case, there are different measurement, but not all of them have the same importance. This modifier evaluates the number of the important measurement that the architecture could perform. For those architectures that can measure critical parameters, the critical measurements modifier is set to a certain value that represents the total number of critical parameters they measure. It is defined as:

$$a_F = \frac{\left(N_{F(total)} + 1\right) - N_{F(ij)}}{N_{F(total)}}$$

where $N_{F(total)}$ is the number of high priority measurements, and $N_{F(ij)}$ is the number of critical measurements satisfied by the architecture ij. $N_{F(total)}$ in the DSSO study for Marine Weather Forecast use-case is 4.

3.5.2. Practicality

The Practicality modifier gives a value to the amount of data generated by its sensors. For an architecture, the data rates of its sensors are accumulated to obtain the total amount of data that this architecture generates. This amount of data is normalized by the same score function than the metrics.

But in that case, for DSSO study, the m_b value is considered as the 25% fraction of the maximum data rate generated by the full set of architectures, the m_a value is the 90% fraction, the P value is set to -3 and the Q value is 0.

The data rate (D_{ij}) is calculated as the sum of the data rates of each node, where the data rate of one node is the sum of the data rates of each sensor on this node, as follows:

$$D_{ij} = \sum_{n \in N_{ij}} \sum_{s \in S_{ijn}} d_s \qquad a_P = \begin{cases} 1 \leftrightarrow D_{ij} < D_P \\ 0 \leftrightarrow D_{ij} \geq D_P \end{cases}$$

where S_{ijn} is the number of instruments in a specific node, N_{ij} is the number of nodes in the architecture ij and d_s the data rate of the instrument s. D_{ij} is the value that is normalized with the score function.

3.5.3. Maturity

Some instrument embarked on the architectures are new technologies and they are not as reliable as the more mature ones. This fact is evaluated by the maturity modifier. Instruments with low TLR are assigned with a value of 0 and the other ones have a value of 1. The maturity of a specific sensor combination (described on chapter 4) is computed as the mean of this values. Finally, the maturity of an architecture is the mean of the sensor combinations embarked on it.

$$a_{Mn} = \frac{1}{S_n} \sum_{s \in S_n} m_s$$

$$a_M = \frac{1}{Ns_{ij}} \sum_{n \in Ns_{ij}} a_{Mn}$$

where S_n is the number of instruments in the sensor combination n, Ns_{ij} is the number of different sensor combinations embarked on the architecture, m_s the maturity of the instrument s (0 or 1). a_{Mn} is the maturity of the sensor combination n.

3.5.4. Data Relevance

Data relevance measures the performance of an architecture measuring the parameters of the use-case. Each instrument archetype has a relevance factor for each parameter that it can measure. This factor depends on the relevance index provided in Observing Systems Capability Analysis and Review Tool (OSCAR) database [26] and on the actual operational limitations of the sensor for a given measurement.

$$R_{(\text{norm})} = \frac{6 - R}{5}$$

$$a_D = \frac{1}{N'_{Kij}} \cdot \sum_{k \in K'_{ij}} R_{k(\text{norm})}$$

where R is the relevance factor, N'_{kij} is the number of measurements satisfied by the architecture and k are these measurements.

R	Relevance for measuring a given variable	Performance with actual operational limitations
1	Primary	Not influenced.
2	Very high	Negligible change.
3	High	Slightly worsened.
4	Fair	Heavily worsened.

R	Relevance for measuring a given variable	Performance with actual operational limitations
5	Marginal	Almost non-operative.

Table 4. Data relevance factor definition.

3.5.5. Versatility

The versatility modifier evaluates the goodness of each of the archetypes from which each architecture is generated. That can be seen as a supra-figure of merit Γ_i calculated as the mean root square of the FoMs of all the architectures that share archetype and the platform size combinations normalized to the maximum number of possible combinations. Summarizing, versatility gives the overall performance of all the sensor combination that could be embarked on the constellation.

$$a_V = \Gamma_i \cdot \sqrt{\frac{C_i}{C_{max}}} = \sqrt{\frac{1}{C_{max}} \cdot \sum_{j=1}^{C_i} \Gamma_{ij}^2}$$

where C_i is the number of sensor combination of one archetype and C_{max} is the maximum numbers of sensor combinations from the database.

3.6. Final Figure of Merit (7)

Once all the parameters are calculated, encompassing the basic FoM, the normalized cost and all the modifiers, all of them are multiplicated in a final figure of merit. That final figure of merit is the one which will be used to rank all the architectures and to choose the desired candidates. The resulting final equation is, pivotal in this study, shown below:

$$\Gamma_{ij_{final}} = q(C_i) \cdot \prod_n b_n^{1-a_{nij}} \sqrt{\frac{1}{N_k} \cdot \sum_k \left(\prod_m q(m_{ijk})^{\gamma_m} \right)^2}$$

4. Methodology

4.1. Inputs

The DSSO study has been carried out for a specific use-case and a set of architectures resulting from a tradespace exploration coming from the previous tasks of the ONION project [25] [27]. Moreover, the use-case has been studied, also, in previous tasks of the general project, resulting in a set of requirements needed for the optimization.

For the use-case study, a relevance survey has been made, and the most relevant use-case has been chosen for the first study. Finally, the Marine Weather Forecast use-case is the selected one and the one that is used in the study carried on in this thesis. This use-case defines a set of seven measurements and four metrics some of which have been considered for the FoM calculation. From the last missions in the OSCAR database [28] there has been extracted the optimal and the required values of these metrics for each measure of the use-case. Also, each metric had given an importance value, that would correspond to the weights to calculate the FoM.

The measurements and the metrics are shown in the two tables below and the requirements can be seen in the appendix I.

Measurements
Ocean surface currents.
Wind speed over sea surface.
Significant wave height.
Dominant wave direction.
Sea Surface Temperature.
Atmospheric pressure over sea surface.
Sea-ice cover

Table 5. Measurements for the Marine Weather Forecast use-case.

Metrics	
Revisit time	Time between two consecutive measures done at the same point on the Earth surface.
Spatial Resolution	Size that each pixel on the image represents in the Earth.
Latency	Time between the request order of a user and the delivery of the data.
Accuracy[2]	This metric is related to the measured offset, radiometric and spatial resolutions and other noise-contributing factors (i.e. pointing accuracy of the platform)

Table 6. Metrics for the Marine Weather Forecast use-case.

Previous tasks in the ONION project, identified the set of critical design variables to generate ONION architectures and defined their ranges of possible values. These variables have been fully combined and accordingly generated the first set of archetypes. In the table below it is shown the values of these parameters.

[2] Accuracy has not been considered at the first coarse iteration.

Variable	Possible values
Nodes	4,5,6,8,10,12,15,20,24
Planes	2,3,4,5,6,8
Orbit Altitude	500,650,800
Walker Pattern [29] [30]	Delta, Star
ISL	100%, 75%, 50%, 25%, 0%

Table 7. Archetype generation. Decision variables.

As it can be seen, the total number of archetypes that can be generated with these parameters are $N_i = 9 \cdot 6 \cdot 3 \cdot 2 \cdot 5 = 1620$. Some of the combinations have no sense, like four nodes within 8 planes. So, the final number of archetypes for the 0study has been 1440. These archetypes have been delivered to our team after the corresponding simulations have been made. Furthermore, each of the archetypes were matched to its performance metrics. All of them were delivered to our team in a xml database file.

4.2. Platform distributions and sensors

Even though the sensors are considered to be inputs for the selection framework, they were not for DSSO project. Therefore, a preliminary study of which sensor would be useful for the studied use-case had to be carried out. This section corresponds to (1) in figure 4 scheme.

First, a list of sensors capable of performing the desired measures was selected from [27]. Then, they were matched to real missions that embark one or more of these sensors and, from them, the desired parameters of these instruments were extracted. Moreover, to reduce the large number of possible combinations of these sensor, the instruments that do not fulfil the requirements had been discarded. Finally, a list of 9 sensors is the one that had been used on the DSSO study.

Name	Reference mission
GNSS-R	DDMI (CYGNSS) [31]
GNSS-RO	BlackJack (GRACE) [32]
Optical Imager (Medium) VIS/NIR/TIR	AVHRR/3 (MetopC) [33]
Altimeter, Ka	Altika (SARAL) [34]
MWR W, Y (Small)	TEMPEST-D [35]
MWR K, Ka, W (Medium)	SSM/I [36]
MWR X, K, Ka, W (Heavy)	TMI (TRMM) [37]
SAR Altimeter	SIRAL (CryoSat-2) [38]
SAR-X	Severjanin-M [39]

Table 8. List of sensors for the Marine Weather Forecast use-case.

The spatial resolution for each one of the altitudes and measurements, the data relevance value for each measurement, the data rate, swath, maturity, mass and power used can be seen in the appendix I.

The instrument had to be embarked on the different nodes of the architectures. From the architecture definition three sizes of platforms have been defined, namely: Heavy, Medium, Small. Each one has a maximum payload mass capacity, 200 kg for the Heavy ones, 50 kg for the Medium and 3 kg for the

Small ones. Therefore, the next step was to combine the sensor in order to fit them in these sizes. In order to do that, the redundant combinations, for example, with two sensors capable to perform the same measurements, had been avoided. The final list of combinations has twelve of them, two for the small platforms, three for the medium ones and seven for the heavy platforms.

#	Platform	Instrument set
1	Small	GNSS-R
2	Small	MWR (small)
3	Medium	GNSS-R + Optical imager
4	Medium	GNSS-R + MWR (small) + RA
5	Medium	GNSS-R + MWR (medium)
6	Heavy	Optical imager + RA
7	Heavy	Optical imager + SAR-Altimeter (+MWR-nadir)
8	Heavy	SAR-X
9	Heavy	SAR-X + Optical imager
10	Heavy	Optical imager + GNSS-R + RA + MWR-heavy
11	Heavy	Optical imager + GNSS-R + SAR-Altimeter + MWR-heavy
12	Heavy	SAR-X + MWR-small + MWR-heavy

And lastly, the final list of architectures was generated. Initially, fifteen distributions of platforms sizes were defined. Some of the preliminary results showed that the fifteen distributions were not enough, therefore, four more were added to have more granularity. The next table shows the nineteen distributions.

Platform distributions	Heavy (%)	Medium (%)	Small (%)
1	1	0	0
2	0,75	0,25	0
3	0,5	0,5	0
4	0,25	0,75	0
5	0	1	0
6	0,75	0	0,25
7	0,5	0,25	0,25
8	0,25	0,5	0,25
9	0	0,75	0,25
10	0,5	0	0,5
11	0,25	0,25	0,5
12	0	0,5	0,5
13	0,25	0	0,75

Platform distributions	Heavy (%)	Medium (%)	Small (%)
14	0	0,25	0,75
15	0	0	1
16	0,09	0,09	0,82
17	0,2	0,2	0,6
18	0,06	0,17	0,77
19	0,15	0,15	0,7

Table 9. Preliminary platform size distributions.

With these distributions and the different sensor combinations, more than 600K architectures have been generated. The Matlab framework can work with them, but it was computationally and hence time demanding. By virtue of an optimization of the software the computation time was considerably reduced and the platform distribution approach had been changed to take advantage of the software. Therefore, instead of using a limited number of distributions, a full set of size combinations has been generated. For example, for an architecture with four nodes, fifteen combinations are possible, but for an architecture of 24 nodes, there are 325 combinations.

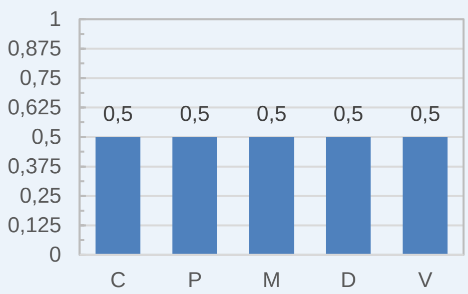
Moreover, with the sub-archetypes generated a new step was required. Now, the sensor combinations had been added to each of the sub-archetype. With all the combinatory the final set of architectures had been generated, with a total number of more than 5.5 million architectures to study in the optimization procedure. This procedure refers to (2) in the scheme of the figure 4.

4.3. ONION vector (6)

Finally, as the last input that the optimization framework needs, there are the weights of the ilities. We asked for it to the consortium of the ONION project and we called the set of weights as the recalled ONION vector. The vector had the five bases for the modifiers calculation ordered as follows: Critical Measurements (C), Practicality (P), Maturity (M), Data relevance (D) and Versatility (V).

$$\boldsymbol{ONION\ vector} = [\,\boldsymbol{C}, \boldsymbol{P}, \boldsymbol{M}, \boldsymbol{D}, \boldsymbol{V}\,]$$

As this procedure is brand new, the selection was not straightforward. In order to simplify the decision, some cases had been studied with their respective results and then, delivered to the consortium. A total number of four cases had been studied. An equalized case, a conservative one, another one called bold case and a last one created after the other three, from the comments and suggestions proposed by the consortium. The four cases can be seen below.

Case	Weights	Description
A		***Equalized*** All modifiers are given a lot of importance. This heavily constrains the design space and only yields architectures that are very practical, mature and versatile and the sensors of which produce relevant data and satisfy all the critical parameters of the use-case.

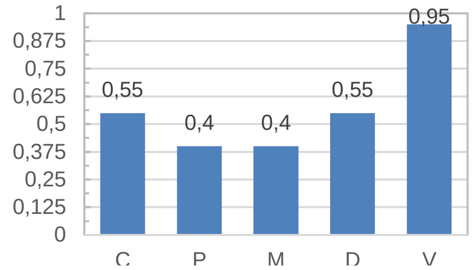
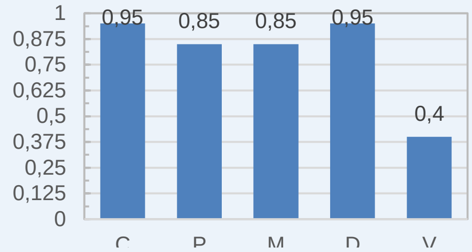
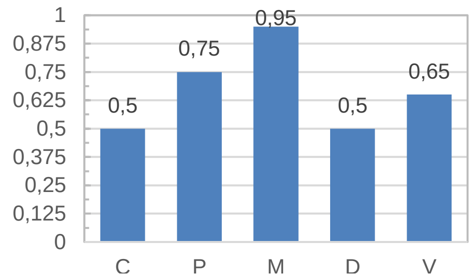
Case	Weights	Description
B		***Conservative*** It gives priority to architectures that are based on mature sensing technologies, and which are still practical (i.e. do not generate large volume of data). Architectural versatility is almost not considered.
C		***Bold*** It produces, conversely, architectures that are very versatile, while lowering the importance of maturity, data relevance and practicality.
D		***Consortium #1*** This case does not prioritize *maturity*. Likewise, both the *versatility* of architectures and their *practicality* has been kept relatively heavy aiming at selecting architectures that satisfy the pursuits and goals of the ONION project and promote innovative architectures. Finally, to prune sensor combinations that have low *data relevance* indices, its modifier has been applied also with a strong weight.

Table 10. Selection cases for the ONION vector.

After some deliberations, the final ONION vector was $[\ 0.5\ \ 0.75\ \ 0.95\ \ 0.5\ \ 0.65\ ]$

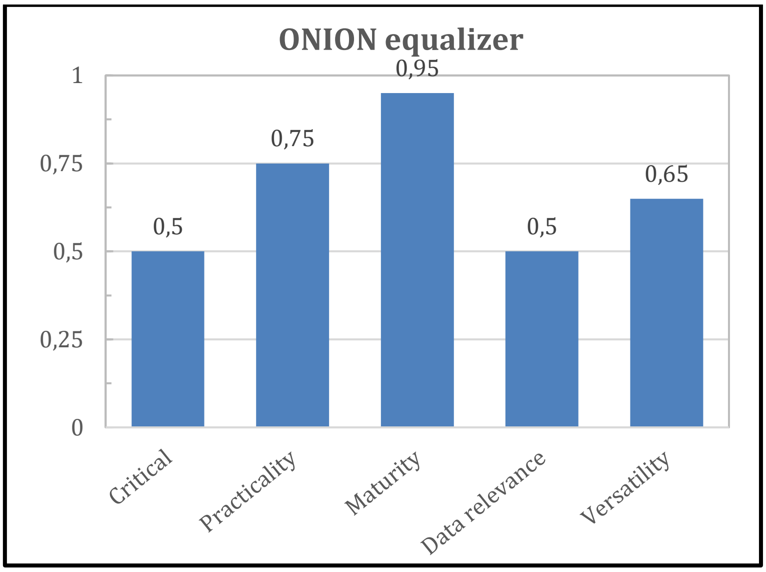


Figure 7. Definitive ONION vector.

The set of plots and results used to make the final decision from these four cases can be found in the appendix II.

5. Software Tool

5.1. Matlab optimization framework

Once the optimization framework has been defined and formulated, it is needed, for the DSSO study, a software tool capable of processing the total number of architectures. The chosen environment has been Matlab. Matlab has a very fast capability to perform calculations with large matrices if loops are avoided. As in this case, there is a large number of architectures, 5.5 million, so, the code had been oriented towards matrix calculations and loops had been avoided as much as possible.

5.1.1. Final Software architecture

The software is divided and structured into different parts as seen in the figure below:

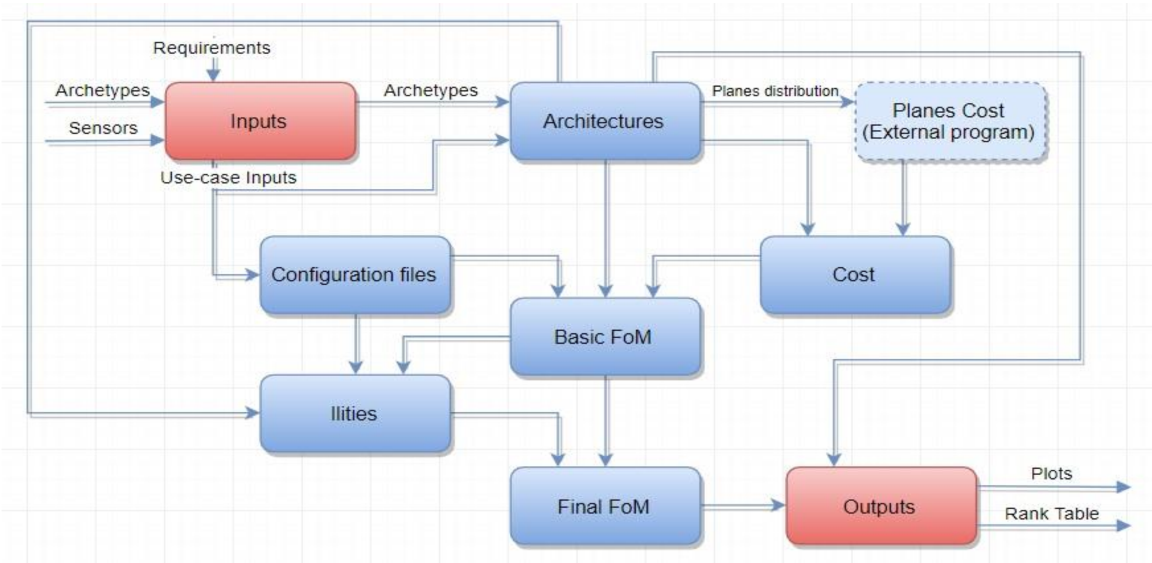


Figure 8. Matlab framework architecture.

Each one of these parts has different scripts, functions or Matlab variable files and all of them had to be run sequentially in order to have the final results the first time, but some of them are not needed to be executed once the variable files of their respective part had been generated.

- **Inputs:** This part includes the two necessary files for the optimization procedure. They are: the archetype xml with all the metrics and configurations of the archetypes and the excel that includes the instrument data, the sensor combinations and the requirements and weights for the metrics of the use-case. One of these files is needed for each use-case that would be studied. Once the input part is executed, almost all the configuration files needed for the rest of the software are generated, like the use case definition file or the requirements file. The xml file is parsed to a mat file also divides the percentage division of revisit time into a node by node division, interpolating the missing nodes data.
- **Architectures:** With the archetypes file and the sensor combinations mat files, this part of the software through a combinatorial process creates a table with the full list of architectures that will be studied and save it into a mat file. Furthermore, generates a database of the different orbital planes configuration, with the number of heavy, medium and small platforms, in order to calculate the launch cost of each plane.
- **Configuration files:** This part of the software represents the stored configuration mat files by the Inputs part.
- **Cost:** Using a cost model coming from ONION project and using the launch cost calculated with a small prolog program this part computes the total cost of all the architectures and add it to the mat file.

- **Basic FoM:** This is the part where most of the optimization procedure is carried out. As inputs for the Basic FoM part are the table of architectures, metrics and costs and the configuration files, such as the requirements one and the weights of the metrics. Finally, as outputs, this part generates a new table with some columns of the architectures table adding to them the FoM without cost and the FoM with cost, both without applying the ilities modifiers.
- **Ilities:** In this part of the framework all the values of all the ilities are calculated, that is the number a_n in the exponent of the modifier formula.
- **Final FoM:** Here the FoMs without modifiers and the ilities are aggregated with the respective equation. That generates the Final FoM with and without cost and adds them to the table coming from Basic FoM.
- **Outputs:** Finally, the software generates a set of predefined plots with the data of the last table and fulfil a table template with the 300 firsts architectures ranked by the Final FoM with cost.

5.1.2. Inputs for the framework

Two files are required to execute correctly the software, the first one, is the xml file with the configuration data of the archetypes and their respective metrics. This file must have a specific format, as follows:

```
<Architecture Platform_Heavy_Mid_Small_distr="1 0 0" ISL_nodes_percent="100" ConstellationID="1" ArchID="1">
  <Constellation ConstellationID="1" Planes="2" Pattern="Delta" Nodes="4" Altitude="800">
    <ONION_node swath_shape="0"
                swath="0"
                raan="0"
                norad="NA"
                incid_angle="37"
                inc="98.6"
                id="ONION_Generic_0_37"
                e="0"
                ap="0"
                agency="ONION EC"
                a="7170"
                M="0"/>
        (This is repeated for each node of the architecture)
  </Constellation>
  <Metrics revisit_90="14.0725"
           revisit_80="14.2306"
           revisit_75="14.3097"
           revisit_70="14.3492"
           revisit_60="14.4803"
           revisit_50="14.5458"
           revisit_40="14.5786"
           revisit_30="14.5949"
           revisit_25="14.6031"
           revisit_20="14.6072"
           revisit_100="13.7562"
           revisit_10="14.6093"
           TotalCost="281.1753"
           PayloadDeployed="800"
           MissionDataLatency="21.9097"
           LaunchCost="29.5067"
           DevelopmentCost="251.6687"
           Connectivity="0.59972"/>
</Architecture>
```

Table 11. XML architecture file format.

The second file is an Excel file with four sheets, one for each needed data, namely:

- *Instrument data:* All the data needed for the used instruments in the use case. The list of instruments, their mass, power used, data rate, maturity, swath and their metrics as the spatial resolution, also the data relevance is included.
- *Combinations:* All the created sensor combinations with the total mass and the total data rate.
- UseCase Requirements: This sheet includes all the measurement of the use-case with the optimum and required values for each metric.
- UseCase Weights: The last sheet has a specific weight to each metric for each measurement.

The tables of this excel file are shown in the appendix I.

5.1.3. Outputs of the framework

All the calculations are not intelligible if they are not properly shown to the user and, therefore, the Matlab framework generates automatically a set of plots and an excel file with the optimal architectures in terms of FoM. A total of 17 plots are generated, all of them but one has one version for the FoM without cost and another for the FoM with cost. The lists of them are shown in the next table.

Plot name	Description
Cost Vs FoM pareto	This plot shows a 2D plot with all the architectures, on the x axis is set the Cost variable and in the y axis is set the Final FoM without cost. Also, it shows the Pareto Frontier and the 30 better architectures are marked. There is only one plot of this type.
Surface	This plot shows a 3D surface with nodes and planes as variables and the FoM as third dimension. The surface follows the best architecture in each node-plane combination.
Contour	This plot is a 2D representation for the contour lines of the surface.
Global trends	This plot is a 4D scatter point, as in surface, of the FoM with respect of the nodes and planes, but the platform size distribution is added with a gradient of RGB colour of the points. Red for the heavy platforms, Green for the medium and blue for the Small. Only the first 1 million architectures are shown, due to hardware limitations.
Top 100 trends	This is a zoom of the Global trends plot for the 100 best architectures.
Top 10 trends	As before, is a zoom for the 10 best architectures.
Global Bars	This is a bar plot of the sorted FoM for the 1 million best architectures.
Top 100 Bars	This is a zoom for the top 100 architectures of the Global bars, but now with the gradient of colours as in Global trends.
Top 10 Bars	The same that Top 100 Bars but with only the 10 best architectures.

Table 12. Output plots of the Matlab framework.

An example of these plots can be found in the next chapter: Final Results.

Moreover, another function has been implement. This function generates a plot and saves it using the two variables passed as input parameters and has the option to use the FoM with or without cost. An example of the output of this function can be seen below.

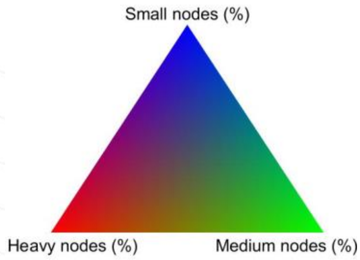


Figure 9. Platform distribution size colour gradient.

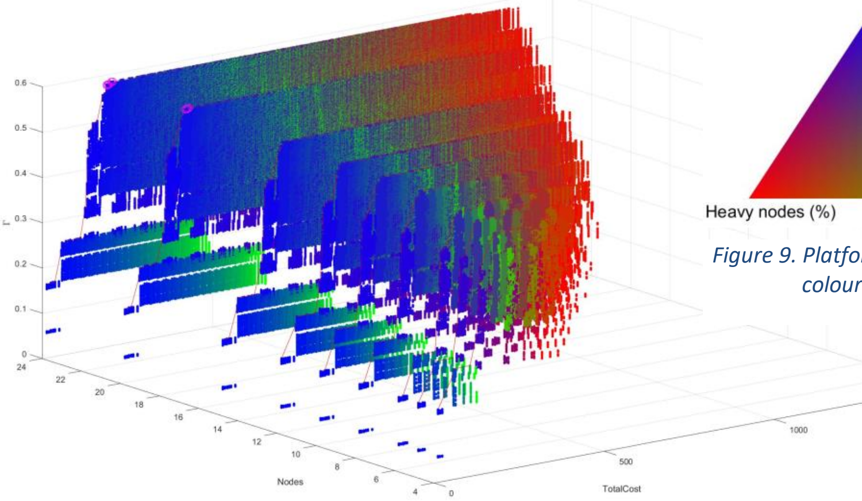


Figure 10. Partial plot resulting with nodes and total cost as variables.

For the Excel table, there is a template in the framework folders that is automatically fulfilled by the software. It shows the 300 best architectures, sort by Final FoM with cost. Moreover, it is added the ONION vector used to the study and the necessary parameters to design the chosen architecture.

5.1.4. Software optimization.

The initial iterations of DSSO study had less architectures than the final one. At first, Matlab can load the full file of architectures on memory and work with it, but in the second iteration this was not possible, because Matlab fills the hardware memory with this type of file very fast. Therefore, a first optimization was needed, SQL database was the chosen platform to load the file and work through an SQL-Matlab interface. The first executable version of the Matlab framework was with this approach, it could process more than 200,000 architectures but the running time was around one hour due to the connecting time between Matlab and SQL. It could be a valid tool, but it does not allow trying with many different ONION vectors, or many different parameters.

With the first final results, we observe that all the first architectures have some of the tradespace exploration limits platform size distributions, so new distribution were added to the process, this has given more than half a million architectures, and now the running time was almost two hours.

Another optimization iteration of the software, this time a long one and the definitive one, had been made. The SQL database were useful but the connecting time was too long, so SQL was discarded and the new approach uses Matlab variable files. They are difficult to read by the user but they are faster than working with SQL. Therefore, the code was rewritten almost from scratch but using some parts of the older one. After the optimization of the software, the running time for the half a million architectures was about three minutes. This gave us the opportunity to generate a larger database of architectures, all the platform size combinations with around 5.5 million architectures, that give a final running time of ten minutes.

As a final remark, the software had been prepared to be able to read any use-case, not only the one for what the DSSO study is working. That means that the use-case can have any number of measurements, instruments and sensor combinations, are the same for the orbital altitudes and the number of metrics. All had been automatized in order to read the input files, no matter how many rows will have the tables.

5.2. Sensitivity analysis (9)

A final step after the study of the architectures will be implemented. A sensitivity analysis for the final selection of architectures is a good tool to know how robust is the decision. The sensitivity analysis, a preliminary one, included in the DSSO project consists in a percentage variation of the metrics in order to see how affects it to the final ranking. The resulting plots shows the different metrics with their percentage variation and how the ranked changes for every 2.5% step.

In the ONION project, after a new iteration of finer simulations the sensitivity analysis will study the impact on the ranking that have some variations of the important variables. It only will be done to the best architectures because each variation of each iteration requires new very long simulations. This analysis is for the ONION project but does not fit on the DSSO project.

Below it can be seen a preliminary plot of this sensitivity analysis. The colors do not have any significance, they are only to have better visibility of the lines.

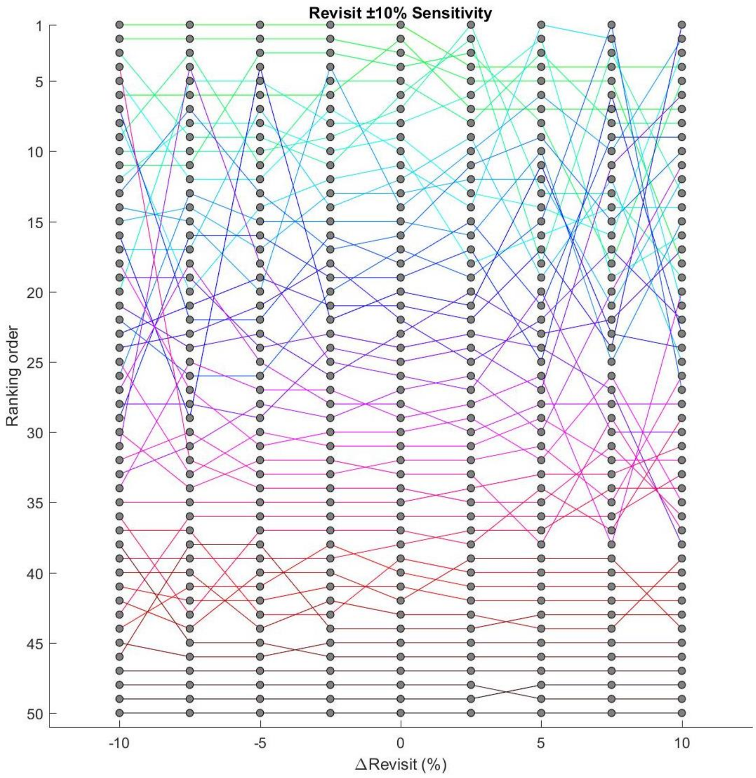


Figure 11. Sensitivity analysis preliminary plot.

6. Final Results

The previous chapters lead to this one, in which the results of the whole procedure will be shown. In this chapter, it can be found a small sample of some preliminary results and how they have affect the decisions made during the project and, finally, the definitive results.

6.1. Evolution of the results

During the different iterations and tests of the DSSO procedure, there have been generated a lot of plots, all of them cannot be placed here, for the larger set of them go to the appendix III. Below you can find some of the preliminary results, the most important ones that explains the changes made to the framework. At first some tests had been done, as explained in the ONION vector part, to determine which vector will be used for the rest of the DSSO study. These plots are placed in the appendix II about the ONION vector cases.

After that, with the ONION vector defined, two of the resulting plots are the next ones:

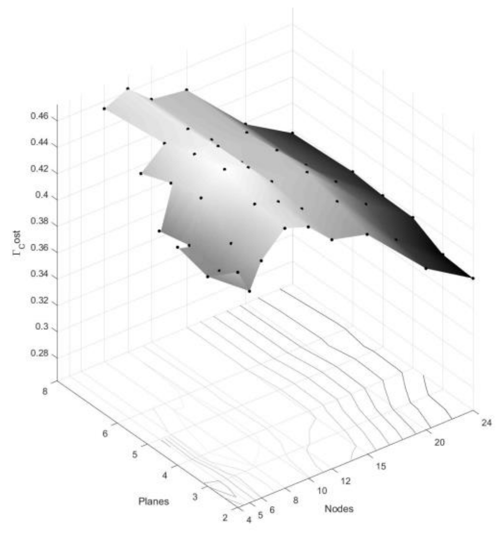


Figure 12. First iteration: surface with cost.

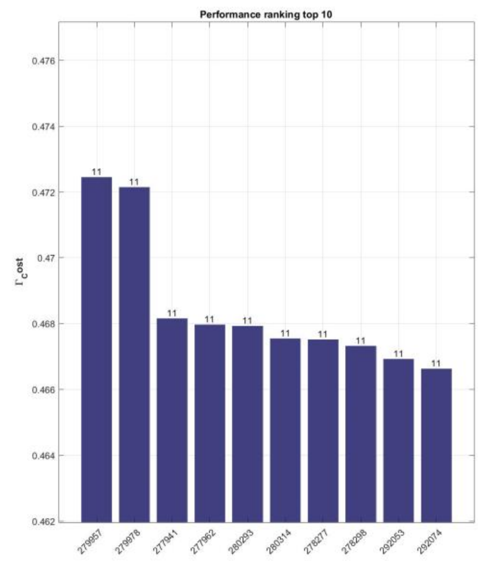


Figure 13. First iteration: Top 10 Bars with cost.

As it can be seen in the figures, there is one architecture with the higher FoM with 8 nodes and 4 planes, but if we see the top 10 bar plot, it corresponds to the eleventh distribution, 25% of the platforms are heavy, 25% are medium and the rest of them are small, so, we have 2 heavy platforms, 2 medium platforms and 6 small platforms. We realized that having two heavy platforms is the most relevant decision, because of the cost of this large platforms due to the SAR sensor embarked on them, it has the better resolution performance of all the sensors. Moreover, with two of these platforms always goes two medium ones, thanks to the optical sensor, that complements the optical one that goes on the heavy platforms. After fixing the heavy and the medium platforms and for a given number of planes, there was only one platform size distribution with small platforms, needed to complement the other platforms thanks to their small cost. At that moment, there were only the 15 firsts ones, so, the possible combinations were very limited.

Therefore, a second iteration was performed, this time with 19 distributions because it was considered that more small platforms could be placed on the architecture. The results were:

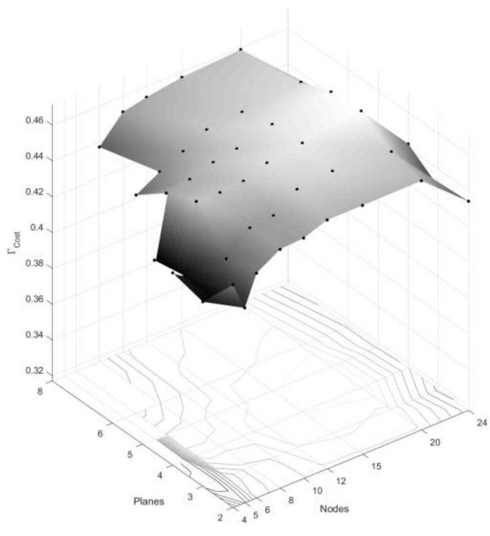


Figure 14. Second iteration: surface with cost.

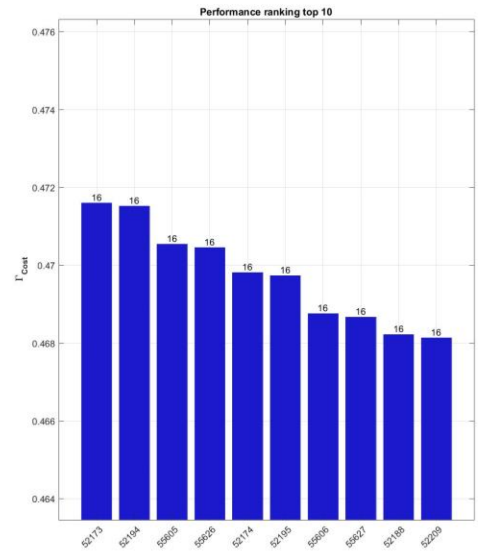


Figure 15. Second iteration: Top 10 Bars with cost.

In this iteration, some changes have result, as can be seen on the first figure, the new winner region was around 20 nodes and 4 or 5 planes. The distribution of all of them are the 16, one of the new ones, for the winner architectures with 20 nodes, that means 2 heavy platforms, 2 medium platforms and 16 small ones. As we can see, the previsions made were correct, but yet again, the optimal in terms of FoM architecture was on the limit of the design space. At that point, the hardware limits and the SQL connection time do not let us to study larger sets of architectures, so, we have two options, either to run the Matlab framework with a zoom on the design space around the optimal architectures or optimize the software, was selected the second option.

With the optimization done, the full set of distributions could be studied, so, 5.5 million architectures were processed and the results was:

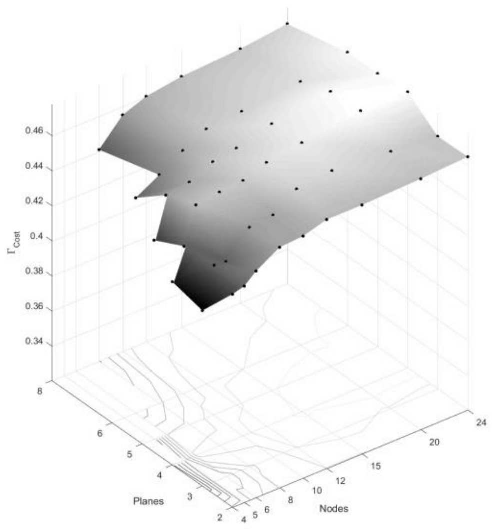


Figure 16. Third iteration: surface with cost.

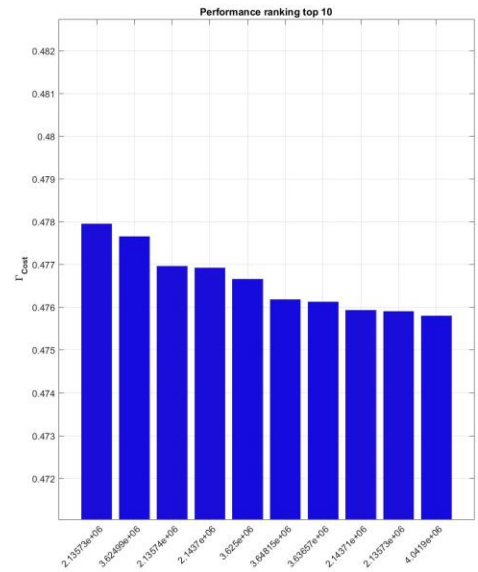


Figure 17. Third iteration: Top 10 Bars with cost.

This time, the surface plot shows that the winner architectures were the ones that have 20 or 24 nodes. With a distribution predominated by small platforms (blue colour), on the table it can be seen that these architectures have two heavy and one medium platforms, the rest of them were small. At that point, we cannot carry out tests with more nodes because we do not have inputs for architectures with more than 24 nodes. Furthermore, as the cost model used to calculate the cost of the small platforms was too coarse and it did not make much sense to add more small platforms because they only help with the latency and it was good enough, we decided the study would be enough to select the winner one. At this point, from the ONION consortium, we were told that there was an error on the revisit time calculation for only one node, so they send to us the new inputs and a final iteration was executed.

6.2. Final selection

After months of work, the final execution of the Matlab framework brought to us to a table with 300 architectures with slightly differences on the final FoM. But the next step of the ONION project cannot simulate such number of architectures because the next orbital simulation will be very accurate and they need a lot of time. Therefore, we chose 21 architectures from this list, the 10 first ones, as was agreed upon in the ONION project work plan, and eleven more, selecting the ones that have changes on some parameters, as number of nodes, planes, orbital altitude, ISL or sensor combinations different to the ten first. These additional architectures have been selected in order to have the study of different parameters since they are interesting for the ONION purpose and can be seen in the table 19 with the changes for what they have been selected in red.

Below you can find some of the plots for the last iteration, the rest are set in the appendix IV.

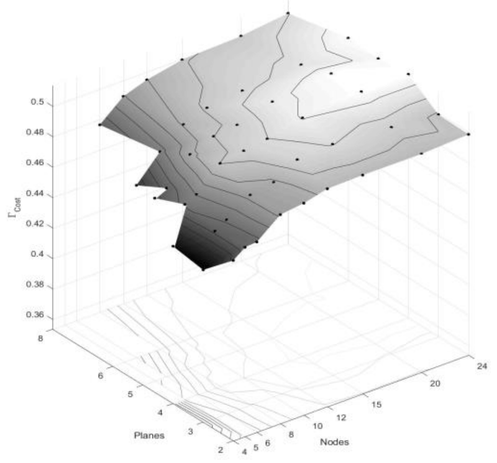


Figure 18. Final results: surface with cost.

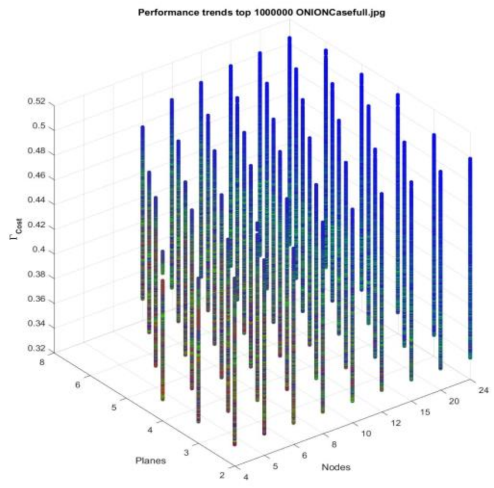


Figure 19. Final results: Global trends with cost.

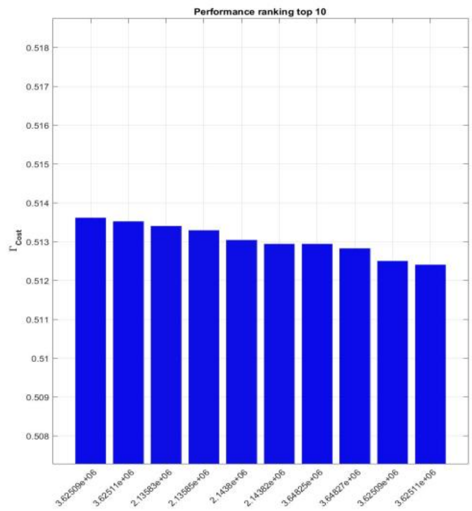


Figure 20. Final results: Top 10 bars with cost

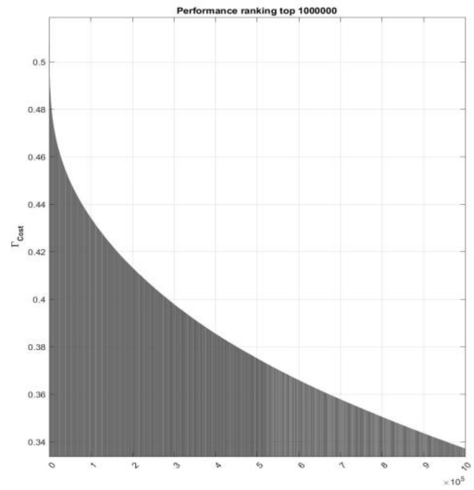


Figure 21. Final results: Global bars with cost.

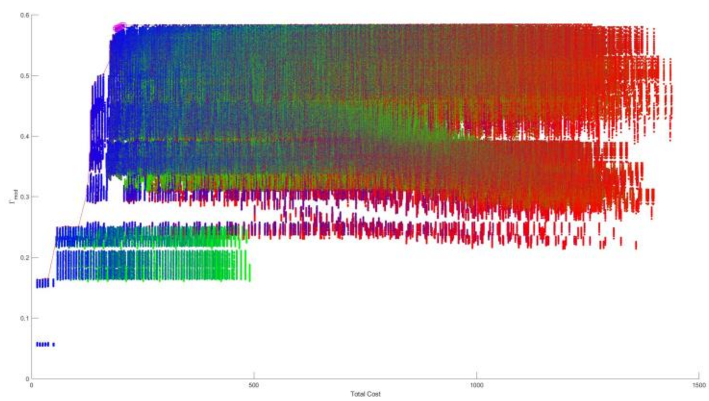


Figure 22. Final results: Cost Vs FoM with Pareto frontier.

These results show, as the third iteration, that the architectures with higher FoMs has 24 or 20 nodes and 4 or 5 planes, but this time, the platform size distribution, although it is dominated by small platforms as before, they only have one heavy platform and one medium platform. This change is due to the fact that the revisit times had an error on the previous iterations. Moreover, the sensor combinations that are present in most of the best architectures are the 8 for the heavy platforms, the 3 for the medium ones and, for the small, both combinations 1 and 2 are presents. And finally, the orbital altitude was at 500 km. Below it is shown a part of the selection table with the 21 selected architectures. Marked in red, the variables for which the architectures had been selected.

Ranking	Altitude	Walker pattern	Planes	Total	Heavy	Mid	Small	H. comb	M. comb	S. comb	ISL
1	500	Delta	4	24	1	1	22	8	3	1	100%
2	500	Delta	4	24	1	1	22	8	3	2	100%
3	500	Delta	4	20	1	1	18	8	3	1	100%
4	500	Delta	4	20	1	1	18	8	3	2	100%
5	500	Delta	5	20	1	1	18	8	3	1	100%
6	500	Delta	5	20	1	1	18	8	3	2	100%
7	500	Delta	6	24	1	1	22	8	3	1	100%
8	500	Delta	6	24	1	1	22	8	3	2	100%
9	500	Delta	4	24	1	1	22	9	3	1	100%
10	500	Delta	4	24	1	1	22	9	3	2	100%
17	500	Delta	4	24	1	1	22	8	3	1	**75%**
19	500	Delta	4	24	1	1	22	8	3	2	**75%**
20	500	Delta	4	24	1	1	22	9	**5**	1	100%
34	500	Delta	4	**15**	1	1	13	8	3	1	100%
36	500	Delta	4	**15**	1	1	13	8	3	2	100%
51	**650**	Delta	4	24	1	1	22	8	3	1	100%
52	**650**	Delta	4	24	1	1	22	8	3	2	100%
53	**800**	Delta	4	24	1	1	22	8	3	1	100%
54	**800**	Delta	4	24	1	1	22	8	3	2	100%
121	500	Delta	**3**	15	1	1	13	8	3	1	100%
132	500	Delta	4	**12**	1	1	10	8	3	1	100%
137	500	Delta	**8**	24	1	1	22	8	3	1	100%
139	500	Delta	4	**12**	1	1	10	8	3	2	100%

Table 13. Final results: Selection table.

7. Budget

In this chapter, the budget of this thesis is attached. At UPC a junior researcher receives 11,36€ per hour. The total amount of hours worked on the project has been 700 hours. Moreover, to develop the software framework has been used the Matlab tool and Microsoft Office package. The Matlab license chosen has been a student one, that is enough for the purpose of this thesis, and Microsoft Office license costs 7€ per months.

In the table below can be seen a detailed of the budget.

Name	Prize per unit	units	Total
Salaries	11.36 €/hour	700	7,952 €
Social Security	0.3*11.36=3.41 €/hour	700	2,386 €
Matlab Student license	69	1	69 €
Microsoft Office license	7€/months	5	35€
		Total budget	**10,442 €**

Table 14. Total Budget for DSSO project.

8. Conclusions and future development

DSSO project has accomplished three purposes: on the one hand, it has developed an optimization framework to design satellite constellations for a given Earth Observation application. Such framework was in pursuit of a wider optimization procedure than the ones that already exists. This objective has been accomplished with the qualitative modifiers, the ilities, added to the figure of merit in order to assess and quantify qualitative aspects of the architectures that go beyond their performance metrics.

The second purpose has been the design and implementation of a specific tool capable of carrying out the optimization calculations for a large set of architectures. This goal has been achieved with the implementation of a set of optimized Matlab scripts and processes capable of process, calculate and deliver to the user the figure of merit of more than 5.5 million architectures. The final execution times allows to run several times the tool to test different combinations of architectures or instruments, and is automatized to any use-case with different number of measurements or sensors.

Finally, a study of the instruments and combinations that would be used to the use-case Marine Weather Forecast has been completed, with a selection of some possible architectures that would carry out the measurements of this use-case with good performances.

8.1. Final results conclusions.

As seen in the chapter 6, the process to reach the last results has been hard and long with much time dedicated to analysing every one of the plots and the winner architectures to see how the decisions on the input variables take effect to the final ranking. The different iterations show that our insights had been accurate about how the selection works and it had concluded to a precise selection framework for the necessities of the ONION project, where DSSO is carried out.

Moreover, one of the most critical parts was the cost normalization because it had a significant impact upon the figure of merit. Some tests have been made during the process and the final results show that the chosen approach had been right. That can specially be seen in the figure 20, Cost Vs FoM with pareto frontier. In the next image, it is shown a zoom of the plot, just were the Pareto frontier has a change in its slope, from there the FoM without cost continues increasing but with a negligible pendent, we could think that the best architectures are those that are in this region. The 30 better architectures sorted by FoM with cost are selected with a pink circle and can be seen that all of them are just in this region, most of them forming the Pareto frontier.

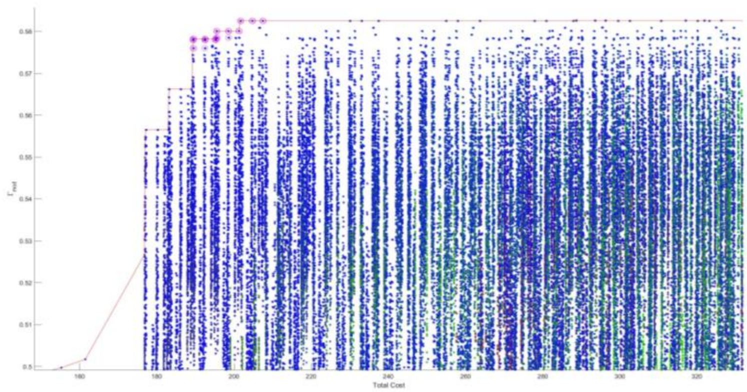


Figure 23. Pareto frontier zoom.

Summarizing, the results are ready to go to the next step of the ONION project, where the selection of architectures will be simulated in depth and, after that, they will be passed through the optimization framework to reorder them into a fine ranking, as it is scheduled inside the ONION project.

8.2. Future work

The optimization framework is ready to be applied to any use-case, but more ilities are planned to be incorporated to the study. These modifiers are:

- Robustness, that evaluates the capacity of an architectures to resists some failures, as one missing node or communications errors.
- Connectedness, that shows how an architecture is interconnect between its nodes.
- Evolvability, that measures how the performance of one architectures grows as nodes and planes are added to it.
- Reachability, that evaluates how well an architecture can be deployed by parts.

Some of these ilities are, just now in progress, like evolvability, but they still need research on them.

The software framework is ready to add any of these modifiers, their scripts only need to follow the same scheme than the one already presented and they must be placed where the other ilities scripts are. The software automatically searches and execute all the scripts placed on that folder. For now, the Matlab framework can run 5.5 million architectures but a system with 8 GB is pushed to its limits, so, if the software is needed for a larger set of architectures it would need another optimization or better computers.

Finally, from the work done in DSSO project there are planned some research papers that will be written in the next months.

Bibliography

[1] Surrey Satellite Technology Ltd, "Applications of Earth Observation," Guildford.

[2] R. Iglesias, A. Aguasca, X. Fabregas, J. j. Mallorqui, D. Monells, C. López-Martínez and L. Pipia, *Ground-based Polarimetric SAR Interferometry for the Monitoring of Terrain Displacement Phenomena,* Barcelona, 2015.

[3] A. M. Sousa, A. C. Gonçalves, P. Mesquita and J. R. Marques de Silva, *Biomass estimation with high resolution satellite images: a case study of Quercus rotundifolia.,* Évora, 2015.

[4] NASA, "NASA: Earth Observatory," [Online]. Available: https://earthobservatory.nasa.gov/Features/GlobalFire/fire_5.php.

[5] C. G. Green, P. Massatt y N. Rhodus, «The GPS 21 primary satellite constellation,» *Navigation,* vol. 36, march 1989.

[6] R. B. Langley, "GLONASS: review and update.," *GPS world 8.7,* 1997.

[7] P. Stephens, J. Cooksley and A. da silva Curiel, "Launch of the international Disaster Monitoring Constellation; the development of a novel international partnership in space," *Recent Advances in Space Technologies, 2003. RAST '03. International Conference on. Proceedings of ,* 2003.

[8] K. Maine, C. Devieux and P. Swan, "Overview of IRIDIUM satellite network," *Proceedings of the Microelectronics Communications Technology Producing Quality Products and Portable Power Emerging Technologies conference,* 1995.

[9] M. Zink, G. Krieger, h. Fiedler, i. Hajnsek and A. Moreira, "The TanDEM-X mission concept.," *Proceedings of the 7th European Conference on Synthetic Aperture Radar.,* 2008.

[10] S. Muñoz, R. W. Hornbuckle and E. Glenn Lightsey, "FASTRAC Early Flight Results.," *Journal of Small Satellites,* 2012.

[11] M. Schoeberl, "The afternoon constellation: a formation of Earth observing systems for the atmosphere and hydrosphere.," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium.,* vol. 1, 2002.

[12] M. D. Graziano, "Overview of distributed missions.," *Distributed Space Missions for Earth,* vol. 31, 2012.

[13] E. Friis-Christensen, H. Lühr and G. Hulot, "Swarm: A constellation to study the Earth's magnetic field.," *Earth, planets and space,* 2006.

[14] J. Guo, D. Maessen and E. Gill, "Fractionated Spacecraft: The new sprout in distributed space systems.," *Proceedings of the 60th International Astronautical Congress.,* 16 October 2009.

[15] O. Brown, P. Eremenko and B. Allen Hamilton, "Fractionated Space Architectures: A Vision for Responsive Space.," *4th Responsive Space Conference.,* 27 april 2006.

[16] A. Golkar, "Federated Satellite Systems (FSS): A Vision Towards an Innovation in Space Systems Design," *IAA Symposium on Small Satellites for Earth Observation.,* April 2013.

[17] A. Golkar and I. Lluch i Cruz, "The Federated Satellite Systems paradigm: Conceptand business case evaluation.," *Acta Astronautica,* vol. 111, 9 February 2015.

[18] C. S. Ruf, S. Gleason and Z. Jelenak, "The CYGNSS nanosatellite constellation hurricane mission," *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International ,* 17 January 2012.

[19] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge: Cambridge University Press, 2004.

[20] E. Crawley, B. Cameron and D. Selva, System architecture : strategy and product development for complex systems, Boston: Pearson, 2016.

[21] J. Daniels, W. P. Werner y A. T. Bahill, «Quantitative methods for tradeoff analyses,» *Systems Engineering,* vol. 4, nº 3, pp. 190-212, 2001.

[22] G. Kalton and I. Flores-Cervantes, *Weighting Methods,* Rockville, 1998.

[23] R. Marler and J. Arora, *Survey of multi-objective optimization methods for engineering.,* Iowa City, 2004.

[24] W. G. Stillwell, D. A. Seaver and W. Edwards, *A Comparison of Weight Approximation Techniques in Multiattribute Utility Decision Making,* 1981.

[25] ONION, «Stakeholders needs - Deliverable nº WP2 - D2.1».

[26] World Meteorological Organization (WMO), «Observing Systems Capability Analysis and Review Tool,» 2011-2017. [En línea]. Available: https://www.wmo-sat.info/oscar/.

[27] ONION, «Architectural elements analysis - Deliverable nº WP3 - D3.1».

[28] World Meteorological Organization, "Observing Systems Capability Analysis and Review Tool. Requirements," [Online]. Available: https://www.wmo-sat.info/oscar/requirements.

[29] J. Walker, "Continuous whole-Earth coverage by circular-orbit satellites pattern.," *Royal Aircraft Establishment Technical Report ,* 1977.

[30] J. Walker, "Circular orbit patterns providing continous whole earth coverage.," *Royal aircraft establishment,* 1970.

[31] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: DDMI,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/921.

[32] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: Blackjack (GRACE),» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/70.

[33] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: AVHRR/3,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/62.

[34] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: AltiKa,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/21.

[35] S. C. Reising y e. al, «Temporal Experiment for Storms and Tropical Systems Technology Demonstration (TEMPEST-D),» de *NASA Earth Science Technology Forum (ESTF 2015)*, Pasadena, CA, 2015.

[36] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: SSM/I,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/533.

[37] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: TMI,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/569.

[38] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: SIRAL,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/509.

[39] World Meteorological Organization, «Observing Systems Capability Analysis and Review Tool, Instrument: Severjanin-M,» [En línea]. Available: https://www.wmo-sat.info/oscar/instruments/view/502.

[40] H. Yang and W. Zhang, *A study of the weighting method for a certain type of multicriteria optimization problem.,* Xi'an, Shaanxi, 2001.

[41] S. Pierotti, *ONION Project Proposal: "A Fractionated Approach to Competitive Imaging from Space",* European Commission, 2015.

Appendices

I. Inputs

This section contains all the tables with the used parameters during the DSSO study. The first four tables are for the instruments parameters. In the spatial resolution table can be seen if the parameter meets the requirements, in green, if it is better than the optimal one, in a brighter green and if the requirements are not meet, in red colour. A 0 means that the sensor is not capable of perform the measure. For the Data relevance table, it is shown the value that affects the modifier with the same name.

The next two tables show the values of the requirements, optimal values and weights of seven use-case measures. Moreover, can be seen which measures are critical measurements for the appropriate modifier.

- **Instrument parameters:**

Name	Reference	Mass (kg)	Power (W)	Data rate (kbps)	Mature (y/n)
GNSS-R	CYGNSS, DDMI	2	12	200	n
Optical Imager (Medium) VIS/NIR/TIR	MetopC, AVHRR/3	31	27	515	y
Radar Altimeter, Ka	Altika, SARAL	40	85	43	y
MWR W, Y (Small)	TEMPEST-D	3	8	20	n
MWR K, Ka, W (Medium)	SSM/I	48,5	45	5	y
MWR X, K, Ka, W (Heavy)	TRMM, TMI	65	50	8,8	y
MWR K, Ka (for correction, nadir-looking)	SENTINEL-3, MWR	26,5	34	5	y
SAR Altimeter, Ku, C	SENTINEL-3, SRAL	70	149	12000	y
SAR-X	Severjanin-M	150	1000	1000	y

Table 15. Instrument list with mass, power, data rate and maturity.

Name	Nadir look angle (degrees)	Incidence angle (degrees)	Aperture (m)	Swath (km) @ 500 km	Swath (km) @ 650 km	Swath (km) @ 800 km
GNSS-R	35	-	0.3x0.3x0.05	714	935	1158
Optical Imager (Medium) VIS/NIR/TIR	55,4	-	0,2032	1600	2160	2800
Radar Altimeter, Ka	0	-	0.42x0.16	6,5	8	10
MWR W, Y (Small)	45		0,1	1000	1370	1700
MWR K, Ka, W (Medium)	-	53,1	1.83x1.98	900	1100	1300
MWR X, K, Ka, W (Heavy)	-	52,8	2,2	1000	1300	1500
MWR K, Ka (for correction, nadir-looking)	0	0	-	-	-	-
SAR Altimeter, Ku, C	0	0	1,2	12,3	15,9	19,6
SAR-X	-	25-48	13.4x0.25	284	355	421

Table 16. Instrument list with swath at different orbital heights.

Spatial Resolution

Name	Ocean Surface Currents	Windspeed over sea surface	Significant wave height	Dominant wave direction	Sea surface temperature	Atmospheric pressure	Sea ice cover
Altitude (km)	500						
GNSS-R	0	24,5	24,5	0	0	0	1,5
Optical Imager (Medium) VIS/NIR/TIR	0	0	0	0	***0,6***	***0,6***	0,6
Radar Altimeter, Ka	6,25	6,25	6,25	0	0	0	0
MWR W, Y (Small)	0	0	0	0	0	7,5	0
MWR K, Ka, W (Medium)	0	5,2	0	0	0	0	5
MWR X, K, Ka, W (Heavy)	0	4,5	0	0	18,8	0	10,3
MWR K, Ka (for correction, nadir-looking)	0	0	0	0	0	0	0
SAR Altimeter, Ku, C	***0,18***	***0,18***	***0,18***	***0,18***	0	0	0,18
SAR-X	1	1	1	1	0	0	1
Altitude (km)	650						
GNSS-R	0	31,9	31,9	0	0	0	2,00
Optical Imager (Medium) VIS/NIR/TIR	0	0	0	0	***0,8***	***0,8***	0,8
Radar Altimeter, Ka	8,12	8,12	8,12	0	0	0	0
MWR W, Y (Small)	0	0	0	0	0	10	0
MWR K, Ka, W (Medium)	0	7	0	0	0	0	7,2
MWR X, K, Ka, W (Heavy)	0	6,0	0	0	24,4	0	6
MWR K, Ka (for correction, nadir-looking)	0	0	0	0	0	0	0
SAR Altimeter, Ku, C	***0,24***	***0,24***	***0,24***	***0,24***	0	0	0,18
SAR-X	1	1	1	1	0	0	1
Altitude (km)	800						
GNSS-R	0	39,2	39,2	0	0	0	2,5
Optical Imager (Medium) VIS/NIR/TIR	0	0	0	0	***0,96***	***0,96***	0,96
Radar Altimeter, Ka	10	10,0	10	0	0	0	0
MWR W, Y (Small)	0	0	0	0	0	12	0
MWR K, Ka, W (Medium)	0	8,5	0	0	0	0	8
MWR X, K, Ka, W (Heavy)	0	7,3	0	0	30,0	0	7,3
MWR K, Ka (for correction, nadir-looking)	0	0	0	0	0	0	0
SAR Altimeter, Ku, C	***0,3***	***0,3***	***0,3***	***0,3***	0	0	0,3
SAR-X	1	1	1	1	0	0	1
Spatial Res. Requirements (km) m_b:	1	1	1	1	1	1	0,01
Spatial Res. Requirements (km) m_a:	25	10	25	15	20	25	12

Table 17. Spatial resolution for three different orbital altitudes by measurements.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Data Relevance

Name	Ocean Surface Currents	Windspeed over sea surface	Significant wave height	Dominant wave direction	Sea surface temperature	Atmospheric pressure	Sea ice cover
Altitude (km)	-						
GNSS-R	-	2	2	-	-	-	3
Optical Imager (Medium) VIS/NIR/TIR	-	-	-	-	5	4	5
Radar Altimeter, Ka	3	5	4	-	-	-	-
MWR W, Y (Small)	-	-	-	-	-	5	-
MWR K, Ka, W (Medium)	-	4	-	-	-	-	3
MWR X, K, Ka, W (Heavy)	-	3	-	-	5	-	2
MWR K, Ka (for correction, nadir-looking)	-	-	-	-	-	-	-
SAR Altimeter, Ku, C	3	5	4	5	-	-	3
SAR-X	2	2	2	2	-	-	1

Table 18. Data relevance values for each instrument by measurements.

Combinations

Name	1	2	3	4	5	6	7	8	9	10	11	12
GNSS-R	y	n	y	y	y	n	n	n	n	y	y	n
Optical Imager (Medium) VIS/NIR/TIR	n	n	y	n	n	y	y	n	y	y	y	n
Radar Altimeter, Ka	n	n	n	y	n	y	n	n	n	y	n	n
MWR W, Y (Small)	n	y	n	y	n	n	n	n	n	n	n	y
MWR K, Ka, W (Medium)	n	n	n	n	y	n	n	n	n	n	n	n
MWR X, K, Ka, W (Heavy)	n	n	n	n	n	n	n	n	n	y	y	y
MWR K, Ka	n	n	n	n	n	n	y	n	n	n	n	n
SAR Altimeter, Ku, C	n	n	n	n	n	n	y	n	n	n	y	n
SAR-X	n	n	n	n	n	n	n	y	y	n	n	y
Mass (kg)	2	3	33	45	51	71	128	150	181	138	168	218
Aggregated data rate (kbps)	200	20	715	263	205	558	12520	1000	1515	767	12724	1029

Table 19. Sensor Combinations.

- **Use-case measurement specifications**

Use-Case Parameter	Critical	Revisit time (h)		Latency (min		Spatial resolution (km)	
-	-	m_b	m_a	m_b	m_a	m_b	m_a
Ocean Surface currents	Y	24	6	60	6	25	1
Wind speed over sea surf (hor.)	Y	24	3	60	6	10	1
Signigicant wave height	Y	12	3	60	10	25	1
Dominant wave direction	y	12	3	60	6	15	1
Sea surface temperature	n	24	3	60	5	20	1
Atmospheric pressure	n	24	3	60	5	25	1
Sea-ice cover	n	24	3	60	10	12	0.01

Table 20. Use-case measurement requirements.

Metric	Ocean Surface Currents	Windspeed over sea surface	Significant wave height	Dominant wave direction	Sea surface temperature	Atmospheric pressure	Sea ice cover
Revisit time (h)	100	100	100	100	100	100	100
Spatial resolution (km)	100	100	100	100	100	100	100
Latency (min)	100	100	100	100	100	100	100

Table 21. Metric non-normalized weights.

II. Selection cases for ONION vector.

The following tables provide all the plots for the four cases analyzed in order to study the impact of the modifiers on the FoM before set the final ONION vector. First can be seen the effect of each one of the modifiers separately and then the four cases itself. The study has been made with the first 15 platform size distributions.

The following plots are provided:

Plot	Title	Remarks
a, d	Interpolated maximum FoM.	Black dots indicate maximum FoM for each point.
b, c	Design space	Only best 100,000 architectures.
e, j	Iso-FoM.	Contours with step of 10%.
f, i	Short-listed (10) architectures.	Transparent circles show location of columns.
g, h	Extended-range (100) set of architectures.	Idem.
k, m	Best 10 architectures.	Platform distribution option (1-15) shown over bars.
l, n	Best 100 architectures.	-
o, p	Best 100,000 architectures.	Colour information removed.

Table 22. Plots provided in appendix II.

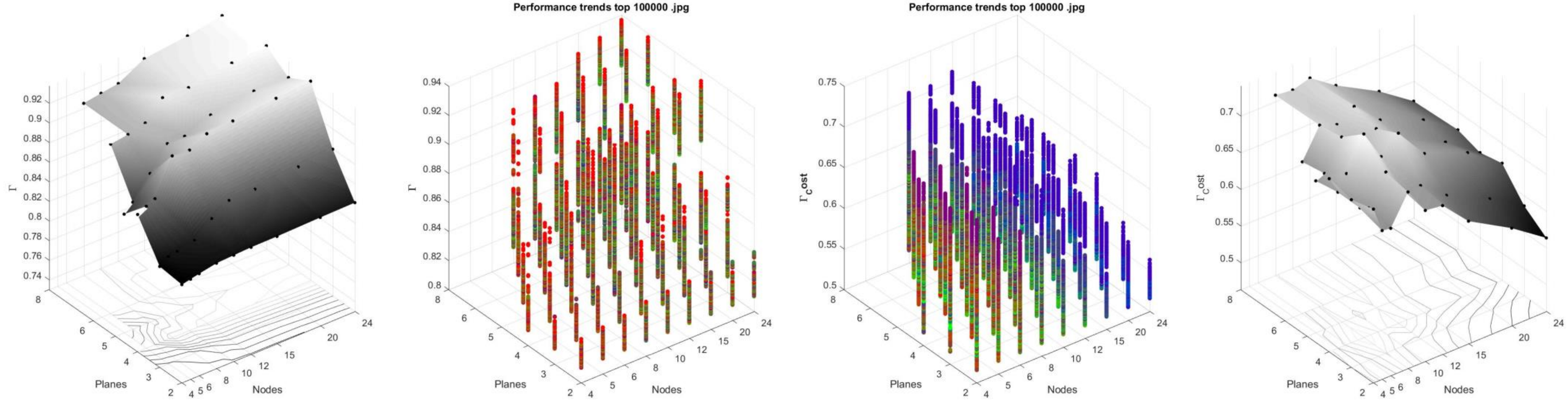
Platform distribution (in %)			
#	*Heavy*	*Mid*	*Small*
1	100	0	0
2	75	25	0
3	50	50	0
4	25	75	0
5	0	100	0

Platform distribution (in %)			
#	*Heavy*	*Mid*	*Small*
6	75	0	25
7	50	25	25
8	25	50	25
9	0	75	25
10	50	0	50

Platform distribution (in %)			
#	*Heavy*	*Mid*	*Small*
11	25	25	50
12	0	50	50
13	25	0	75
14	0	25	75
15	0	0	100

Table 23. Colour pattern for the 15 platform size distributions.

Architecture performances unmodified

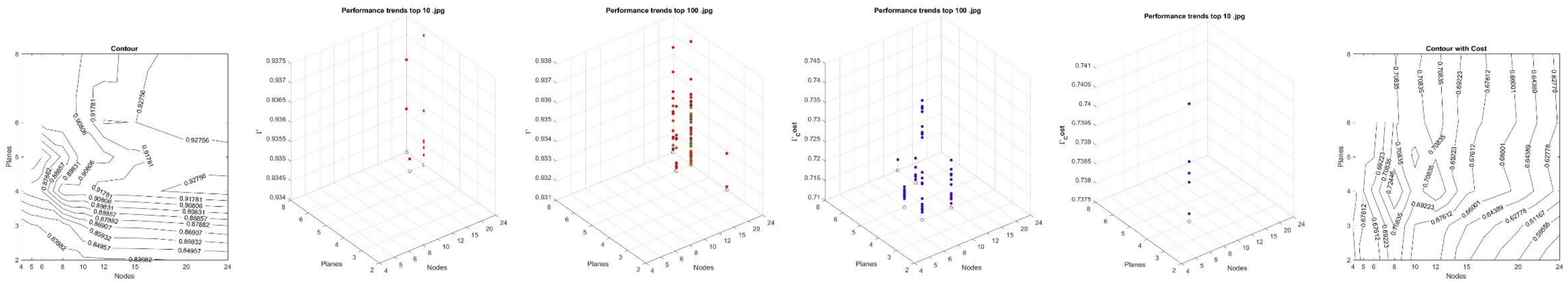


a. Interpolated maximum FoM without cost

b. Design space without cost

c. Design space with cost

d. Interpolated maximum FoM with cost



e. Iso-FoM without cost

f. Short-listed (10) architectures without cost

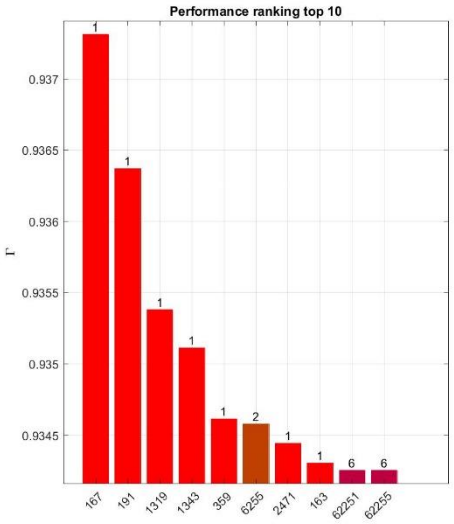
g. Extended-range (100) list of architectures without cost

h. Extended-range (100) list of architectures without cost

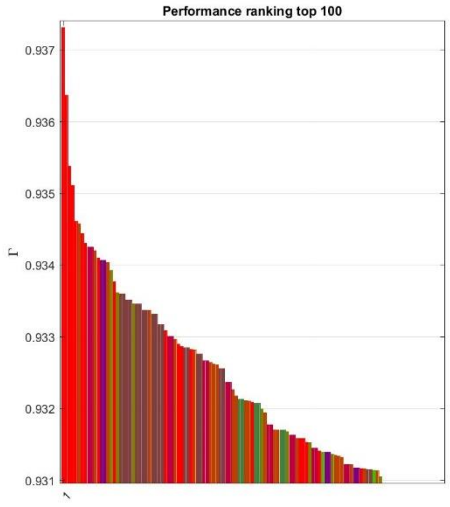
i. Short-listed (10) architectures with cost

j. Iso-FoM with cost

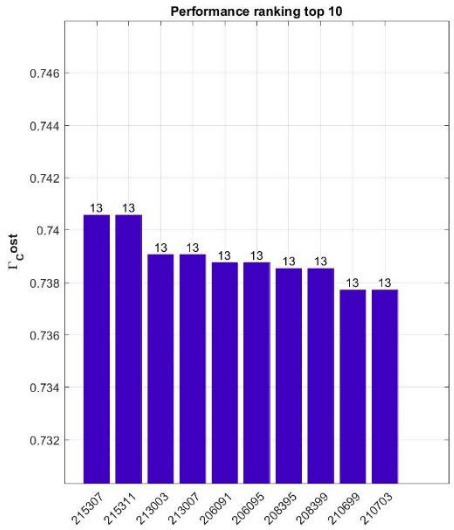
Architecture performances unmodified



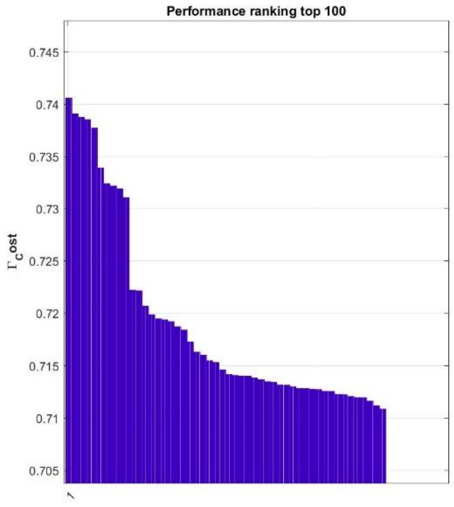
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)



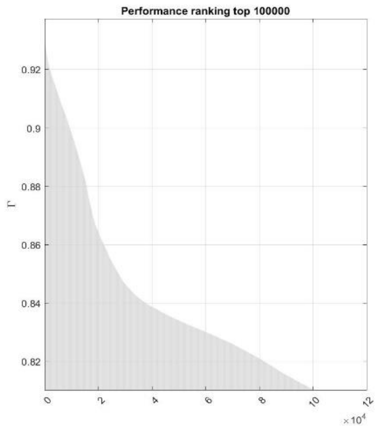
m. Best 10 architectures (with cost)



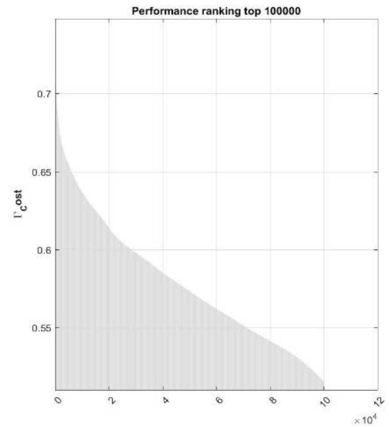
n. Best 10 architectures (with cost)

ONION vector

$\boldsymbol{b_F}$	1
$\boldsymbol{b_P}$	1
$\boldsymbol{b_M}$	1
$\boldsymbol{b_D}$	1
$\boldsymbol{b_V}$	1



o. Best 100,000 architectures (without cost)



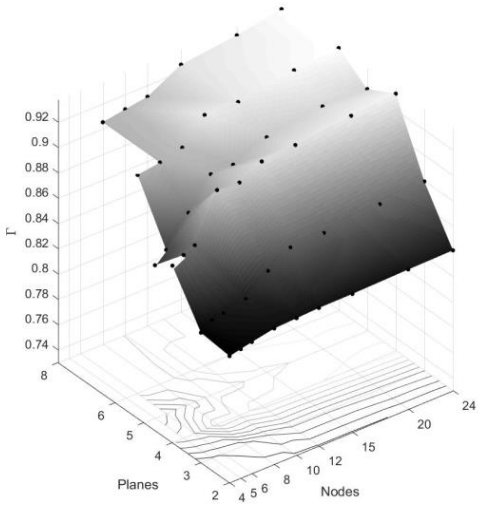
p. Best 100,000 architectures (with cost)

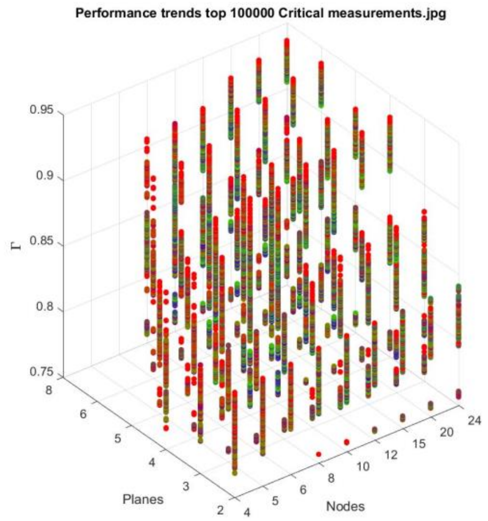
Table 24. Selection cases: Unmodified FoM plots.

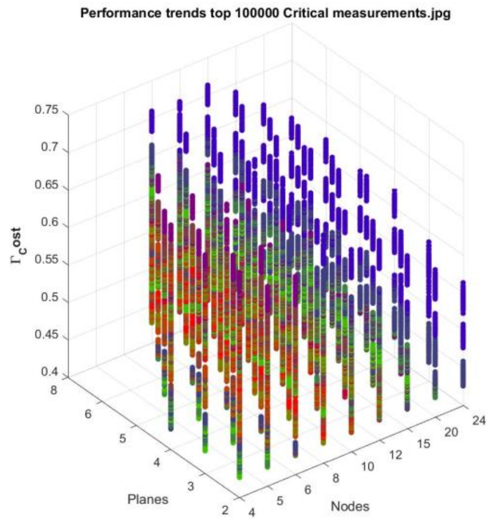
Parameter values

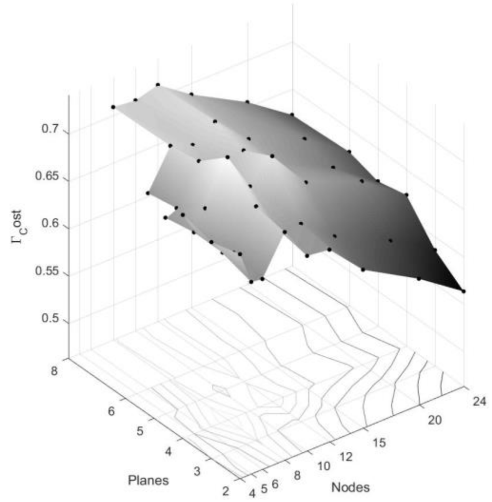
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Critical measurements







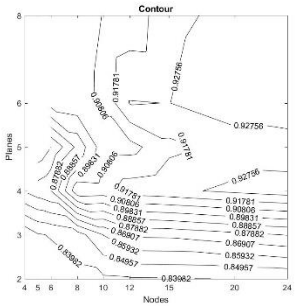


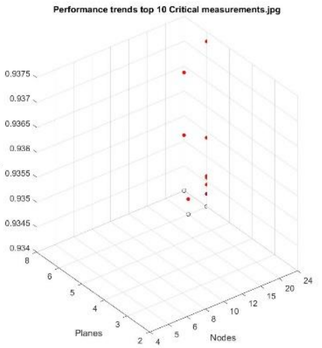
a. Interpolated maximum FoM without cost

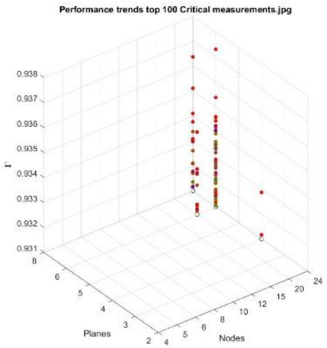
b. Design space without cost

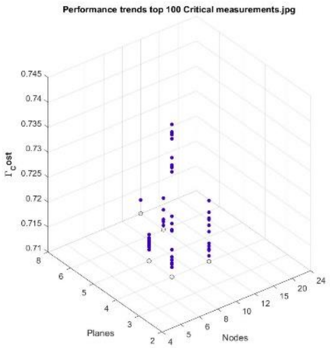
c. Design space with cost

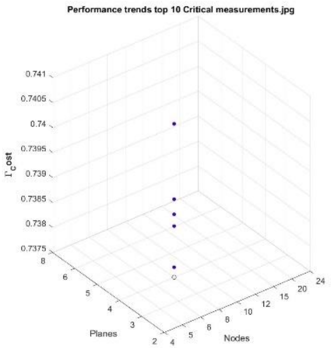
d. Interpolated maximum FoM with cost

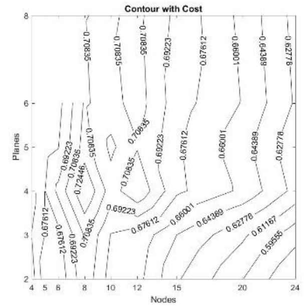












e. Iso-FoM without cost

f. Short-listed (10) architectures without cost

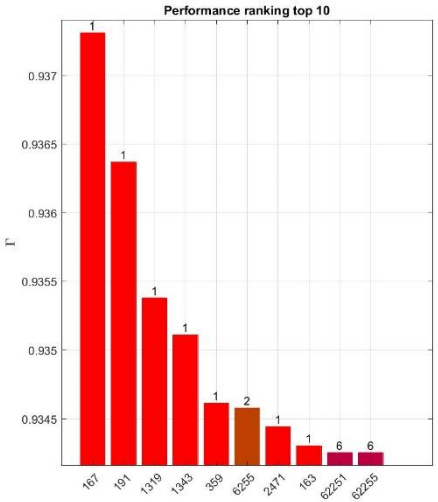
g. Extended-range (100) list of architectures without cost

h. Extended-range (100) list of architectures without cost

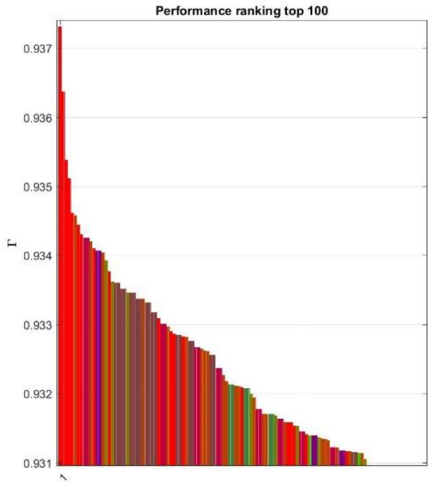
i. Short-listed (10) architectures with cost

j. Iso-FoM with cost

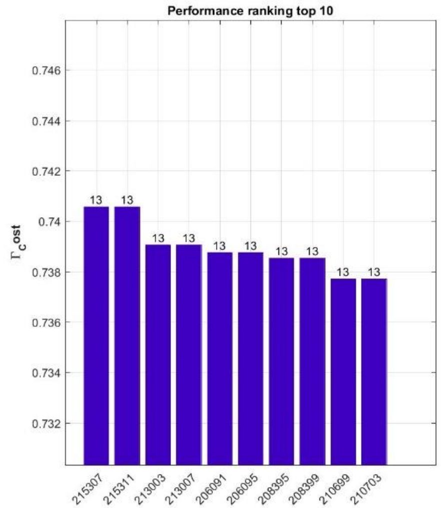
Critical measurements



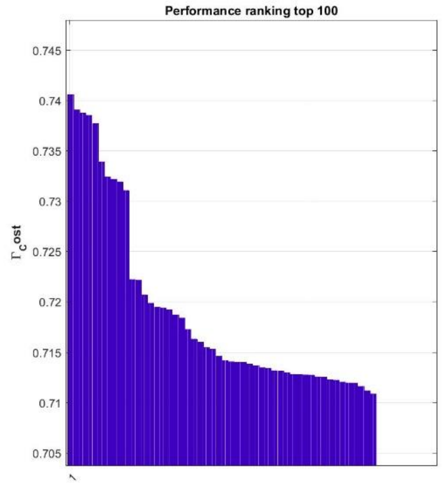
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)



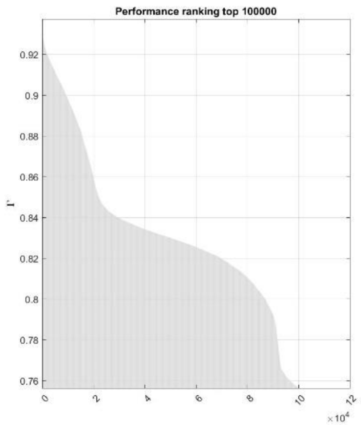
m. Best 10 architectures (with cost)



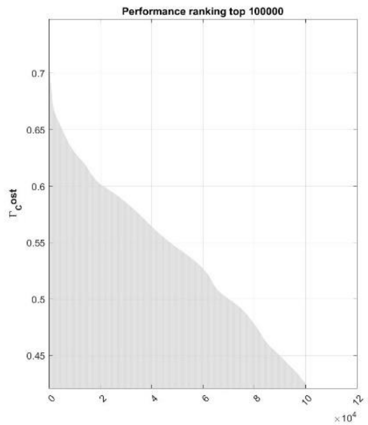
n. Best 10 architectures (with cost)

ONION vector

$\boldsymbol{b_F}$	0.1
$\boldsymbol{b_P}$	1
$\boldsymbol{b_M}$	1
$\boldsymbol{b_D}$	1
$\boldsymbol{b_V}$	1



o. Best 100,000 architectures (without cost)



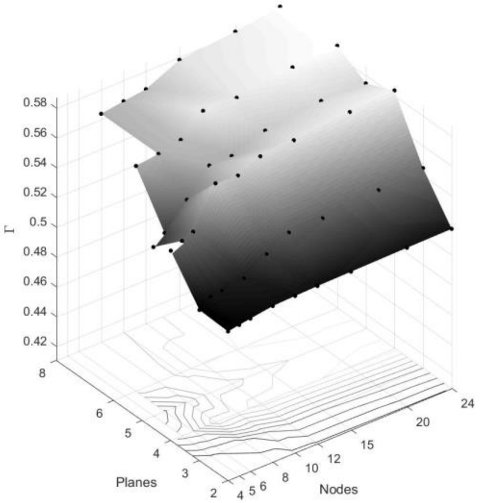
p. Best 100,000 architectures (with cost)

Parameter values

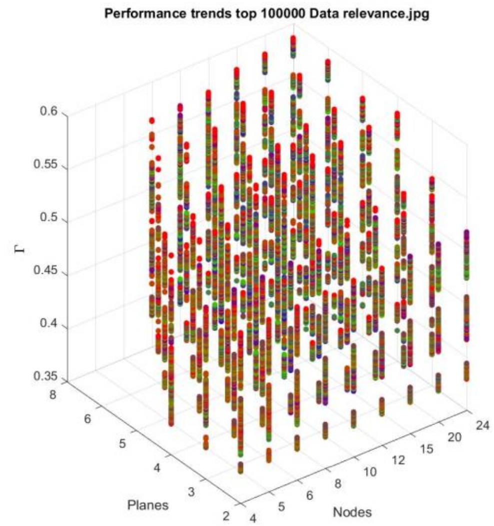
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 25. Selection cases: Critical measurements modifier plots.

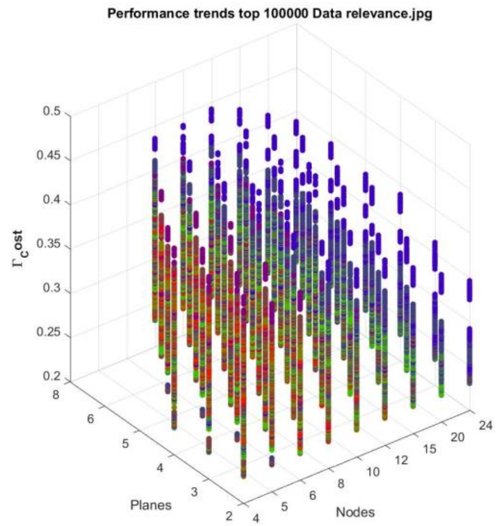
Data relevance



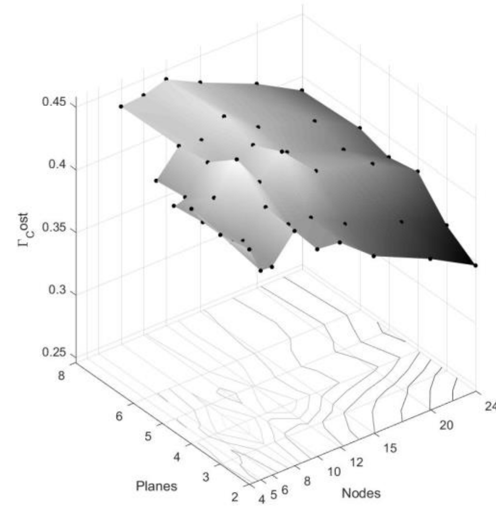
a. Interpolated maximum FoM without cost



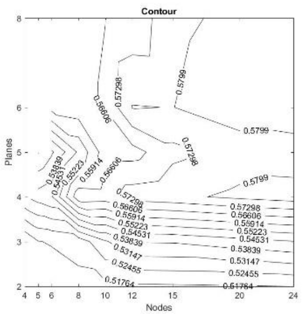
b. Design space without cost



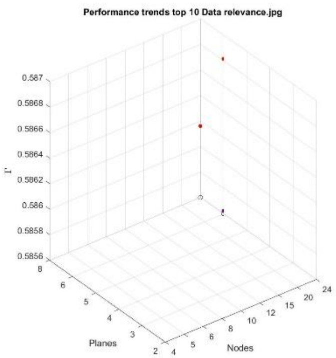
c. Design space with cost



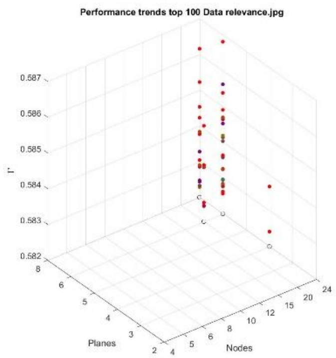
d. Interpolated maximum FoM with cost



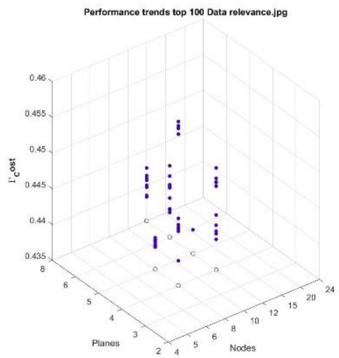
e. Iso-FoM without cost



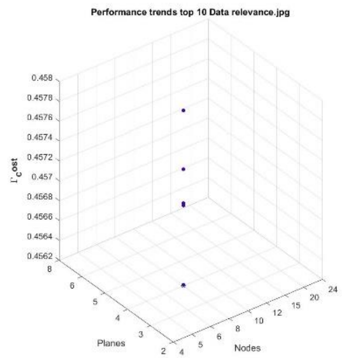
f. Short-listed (10) architectures without cost



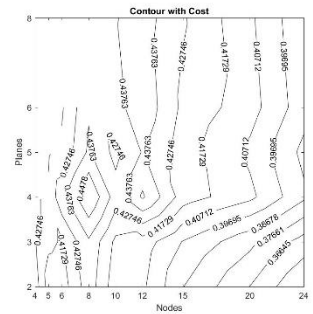
g. Extended-range (100) list of architectures without cost



h. Short-listed (10) architectures with cost

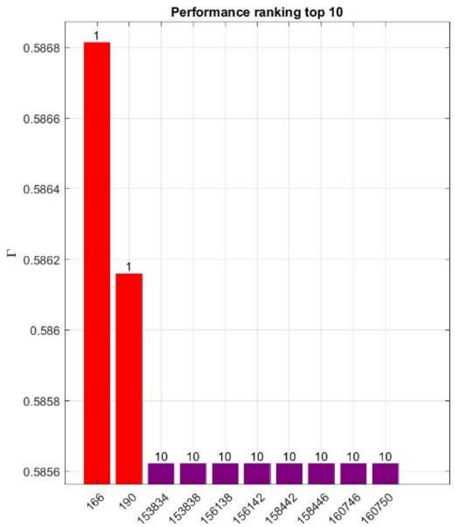


i. Extended-range (100) list of architectures without cost

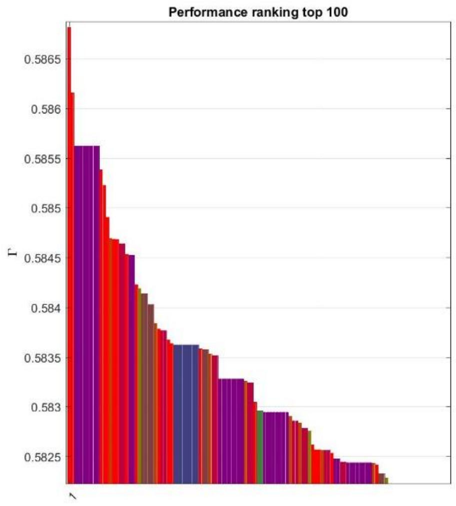


j. Iso-FoM with cost

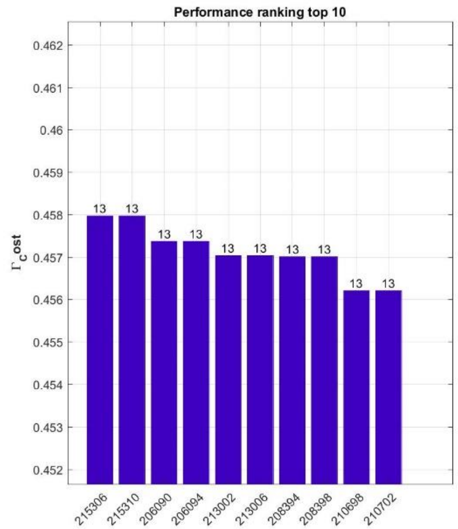
Data relevance



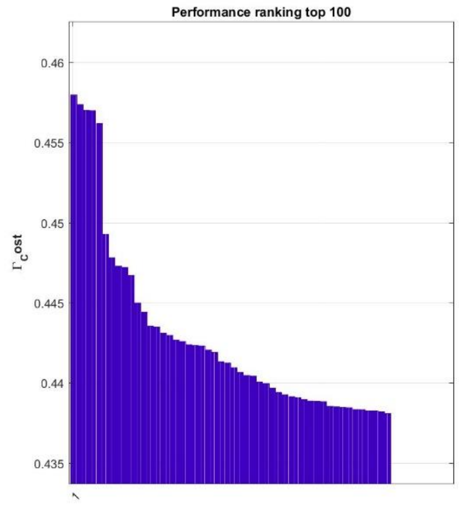
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)



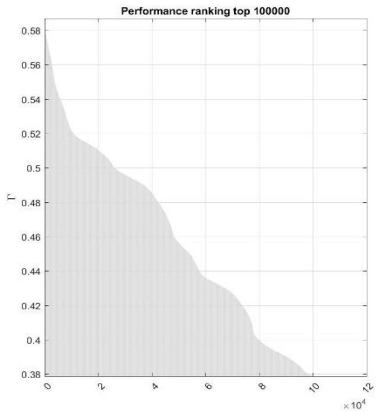
m. Best 10 architectures (with cost)



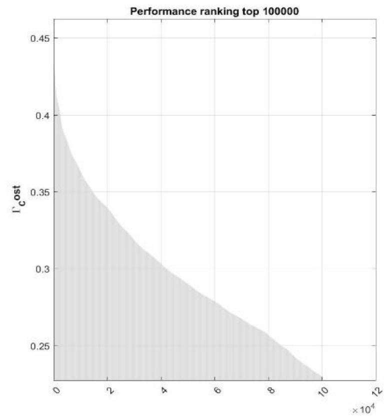
n. Best 10 architectures (with cost)

ONION vector

b_F	1
b_P	1
b_M	1
b_D	0.1
b_V	1



o. Best 100,000 architectures (without cost)



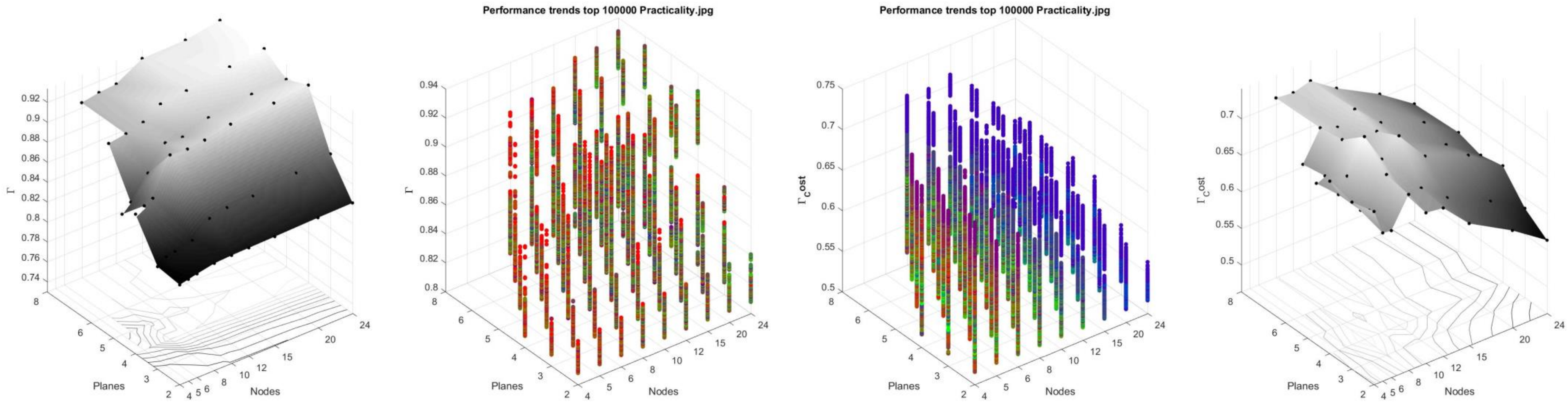
p. Best 100,000 architectures (with cost)

Parameter values

P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 26. Selection cases: Data relevance modifier plots.

Practicality

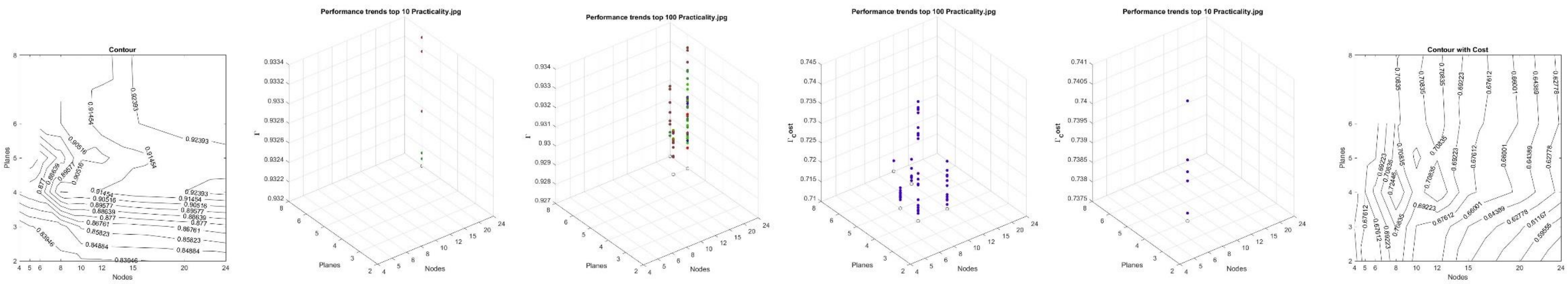


a. Interpolated maximum FoM without cost

b. Design space without cost

c. Design space with cost

d. Interpolated maximum FoM with cost



e. Iso-FoM without cost

f. Short-listed (10) architectures without cost

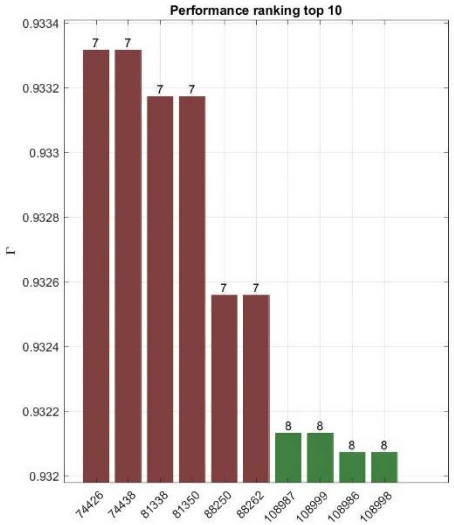
g. Extended-range (100) list of architectures without cost

h. Extended-range (100) list of architectures without cost

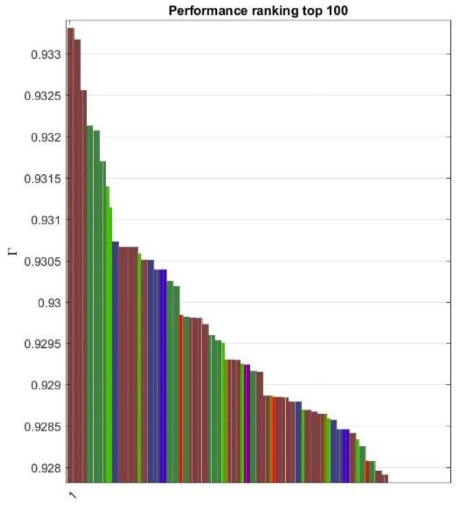
i. Short-listed (10) architectures with cost

j. Iso-FoM with cost

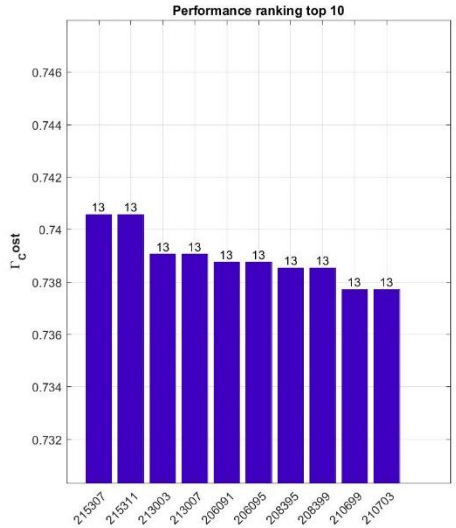
Practicality



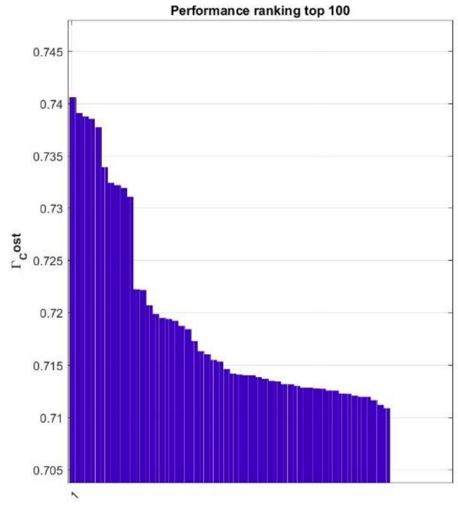
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

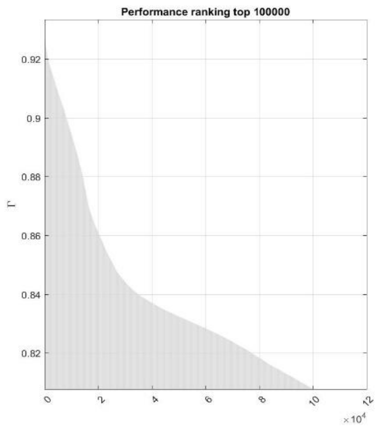


m. Best 10 architectures (with cost)

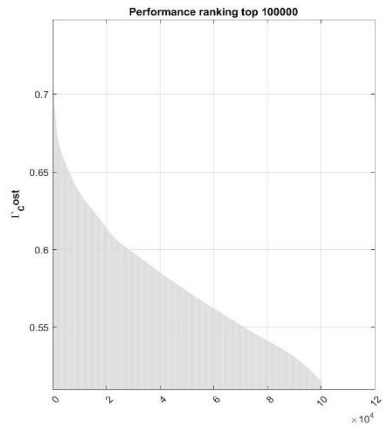


n. Best 10 architectures (with cost)

ONION vector	
b_F	1
b_P	0.1
b_M	1
b_D	1
b_V	1



o. Best 100,000 architectures (without cost)

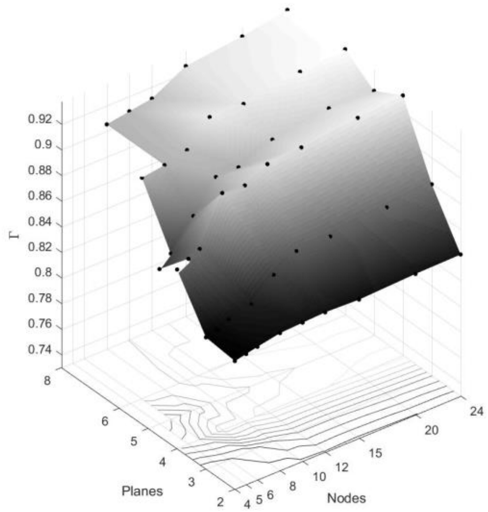


p. Best 100,000 architectures (with cost)

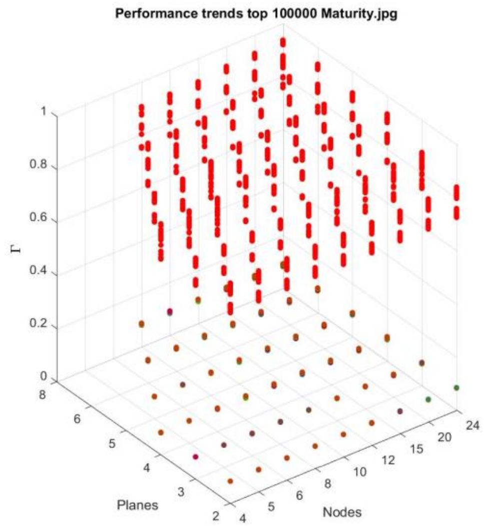
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 27. Selection cases: Practicality modifier plots.

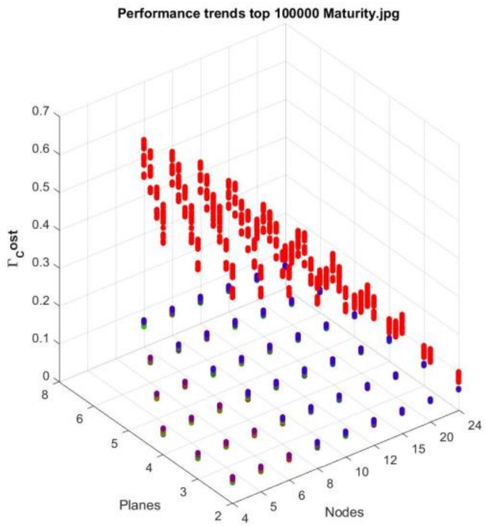
Maturity



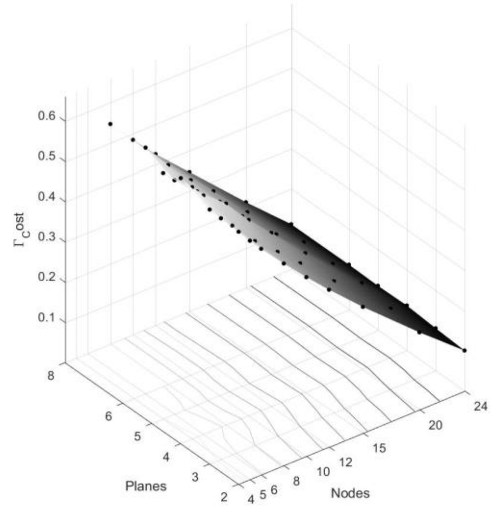
a. Interpolated maximum FoM without cost



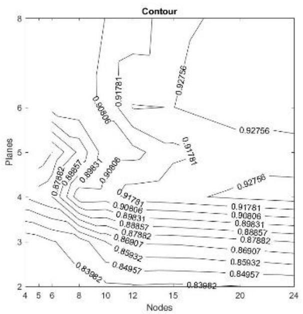
b. Design space without cost



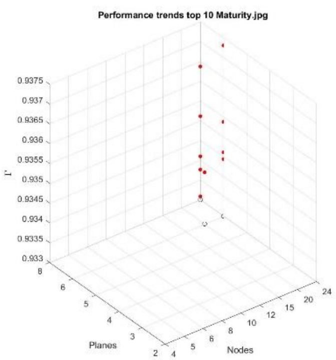
c. Design space with cost



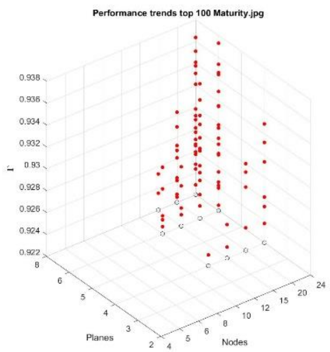
d. Interpolated maximum FoM with cost



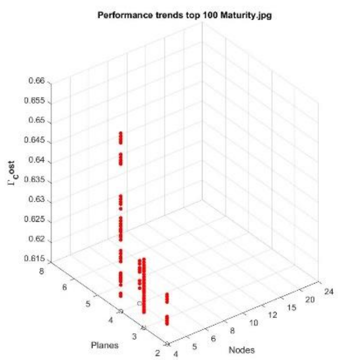
e. Iso-FoM without cost



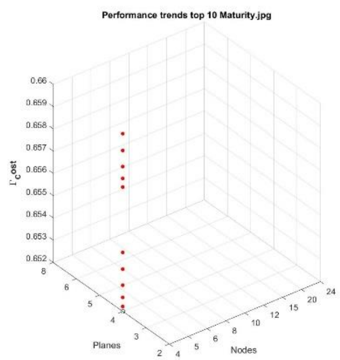
f. Short-listed (10) architectures without cost



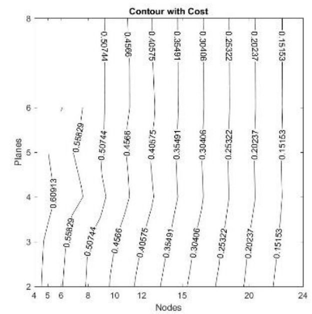
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost

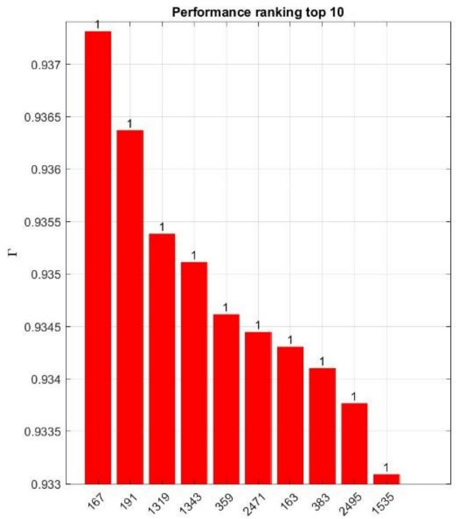


i. Short-listed (10) architectures with cost

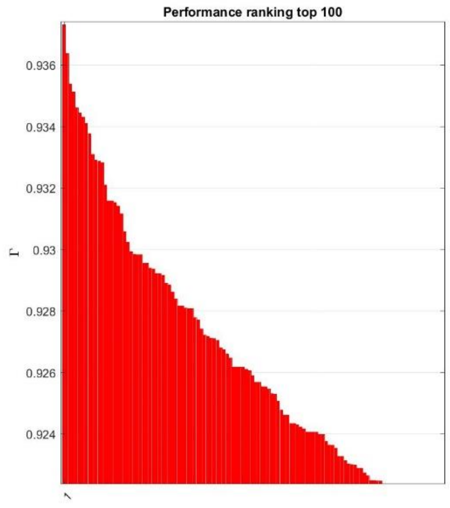


j. Iso-FoM with cost

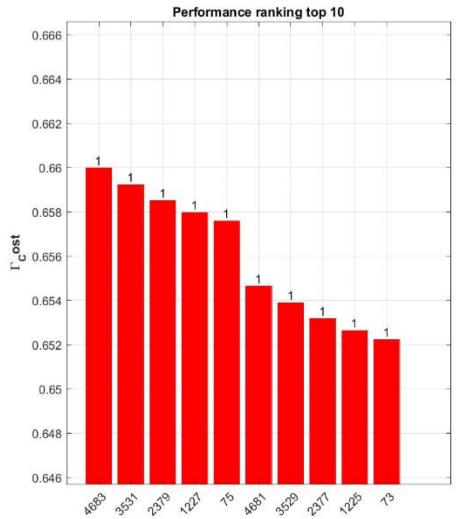
Maturity



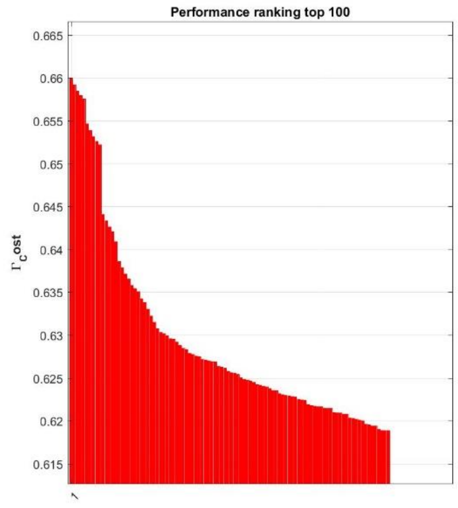
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

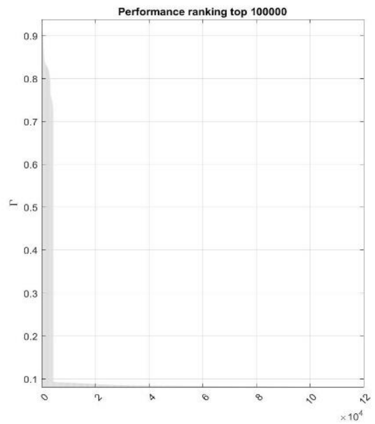


m. Best 10 architectures (with cost)

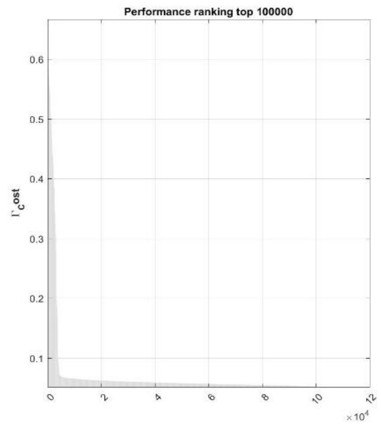


n. Best 10 architectures (with cost)

ONION vector	
b_F	1
b_P	1
b_M	0.1
b_D	1
b_V	1



o. Best 100,000 architectures (without cost)

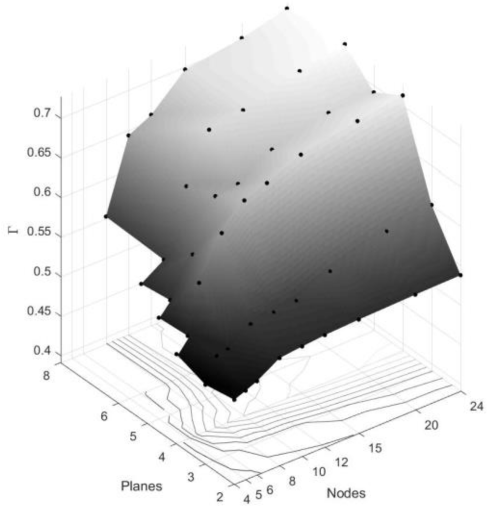


p. Best 100,000 architectures (with cost)

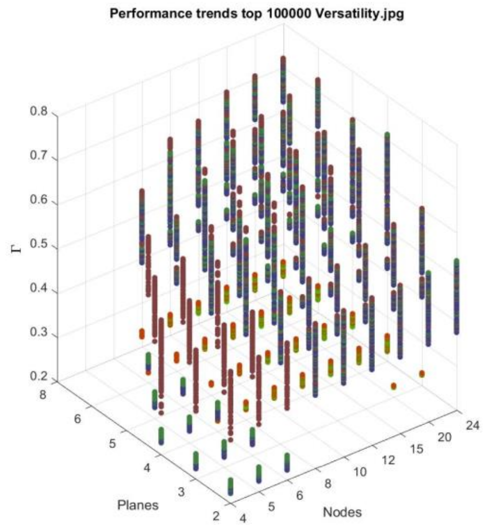
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 28. Selection cases: Maturity modifier plots.

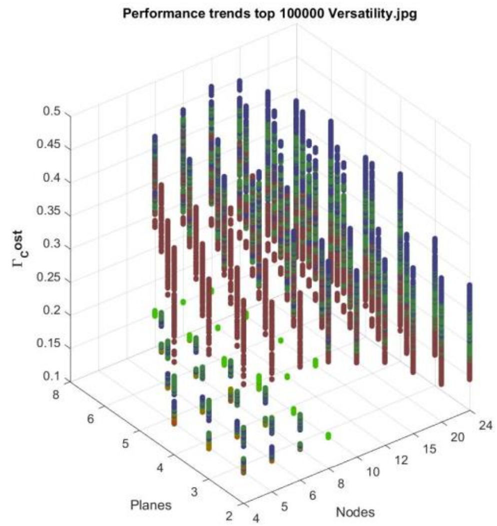
Versatility



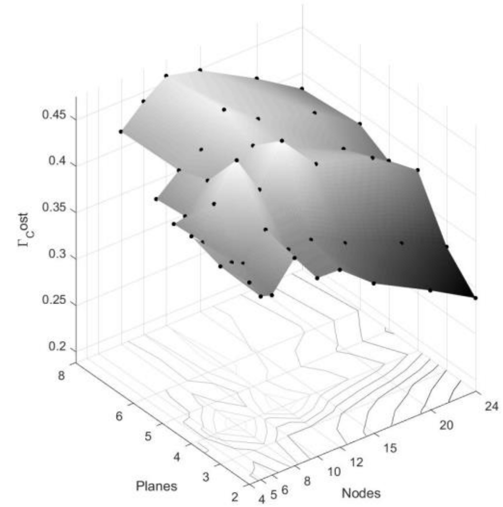
a. Interpolated maximum FoM without cost



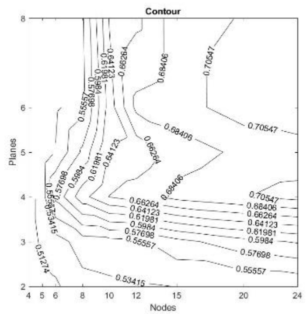
b. Design space without cost



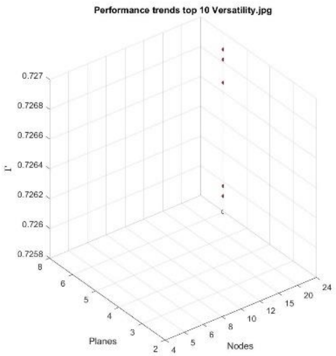
c. Design space with cost



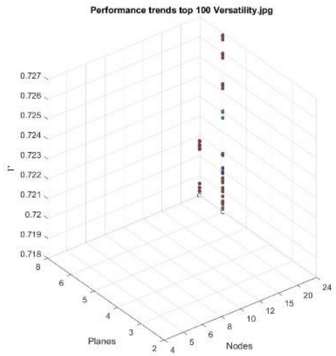
d. Interpolated maximum FoM with cost



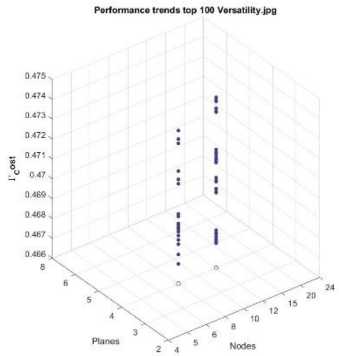
e. Iso-FoM without cost



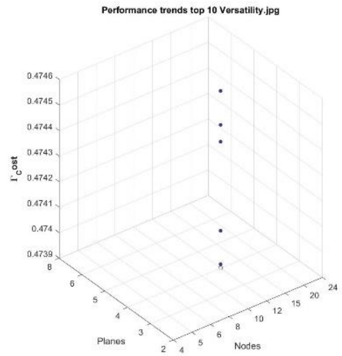
f. Short-listed (10) architectures without cost



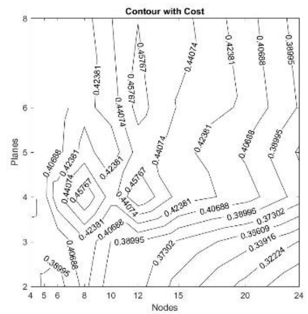
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost

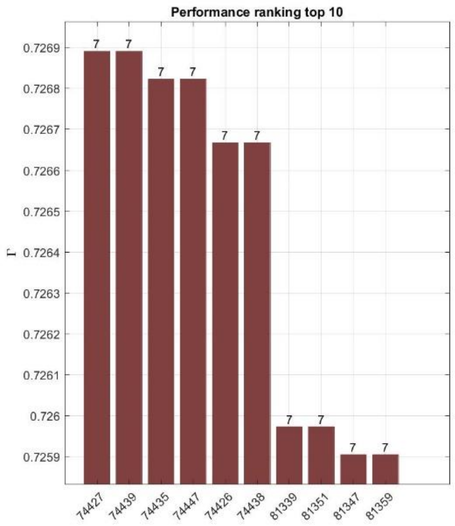


i. Short-listed (10) architectures with cost

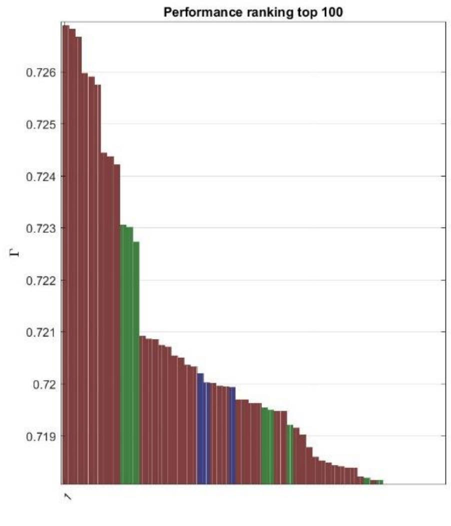


j. Iso-FoM with cost

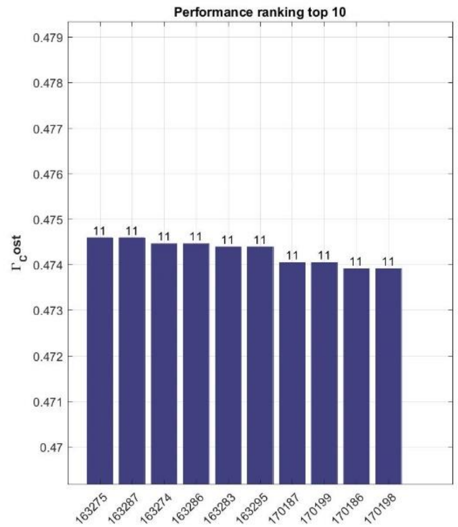
Versatility



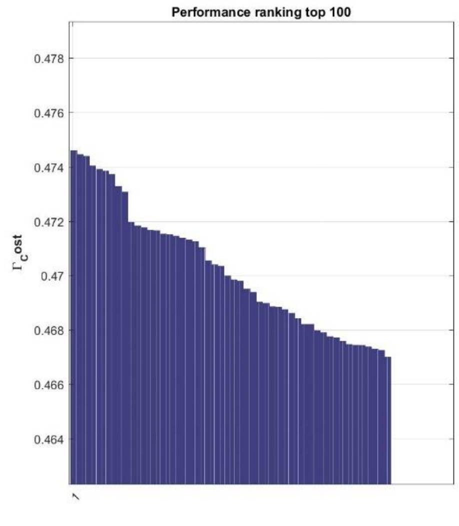
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

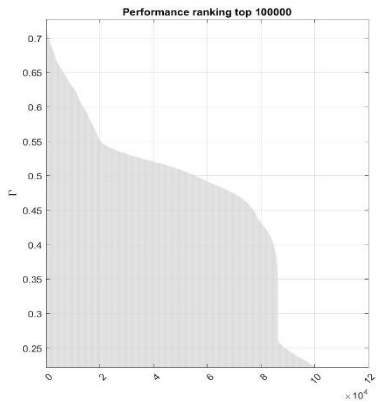


m. Best 10 architectures (with cost)

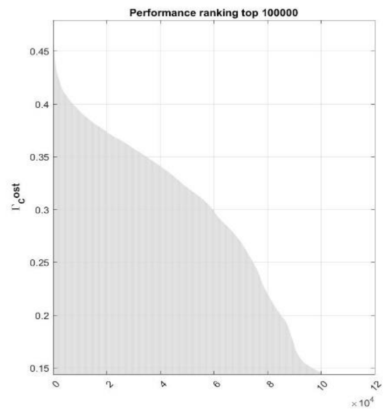


n. Best 10 architectures (with cost)

ONION vector	
b_F	1
b_P	1
b_M	1
b_D	1
b_V	0.1



o. Best 100,000 architectures (without cost)

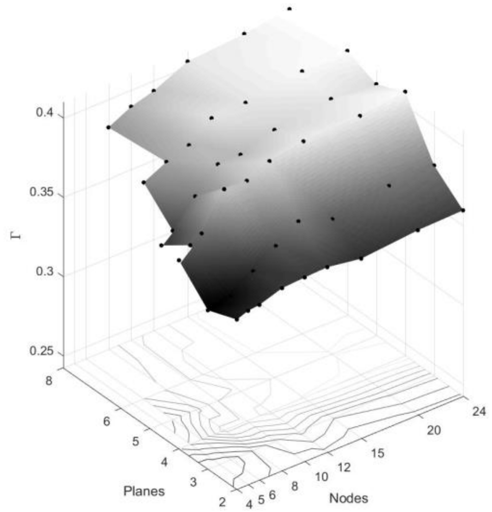


p. Best 100,000 architectures (with cost)

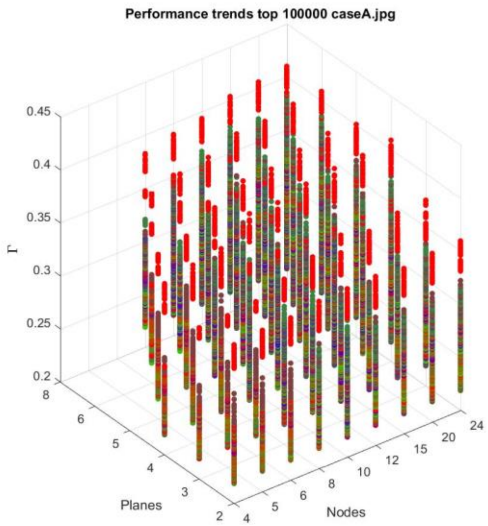
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 29. Selection cases: Versatility modifier plots.

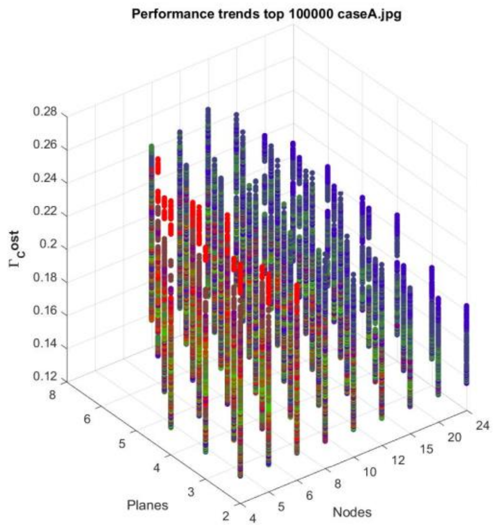
A – Equalized weights



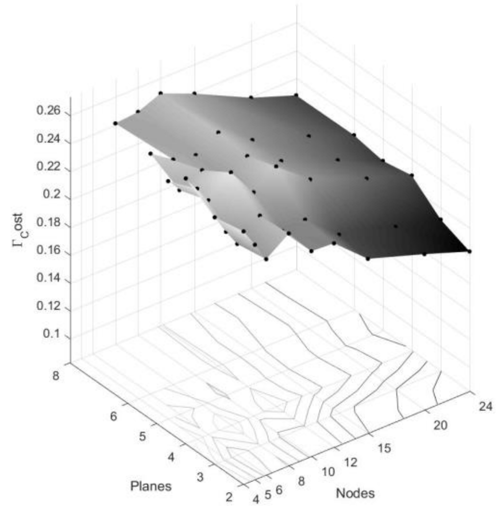
a. Interpolated maximum FoM without cost



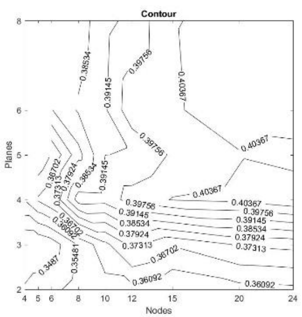
b. Design space without cost



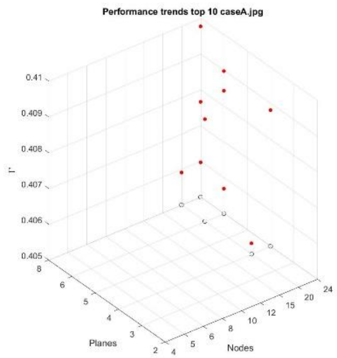
c. Design space with cost



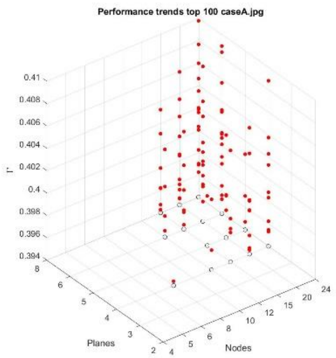
d. Interpolated maximum FoM with cost



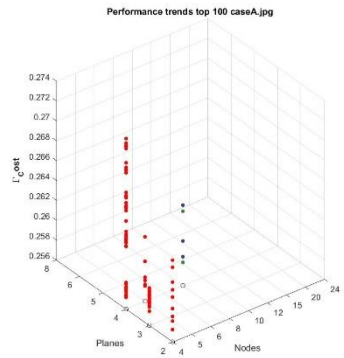
e. Iso-FoM without cost



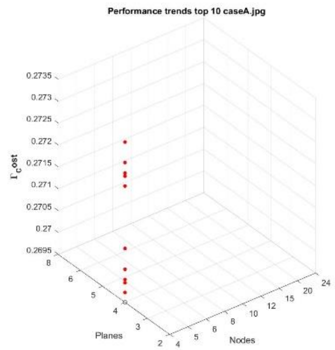
f. Short-listed (10) architectures without cost



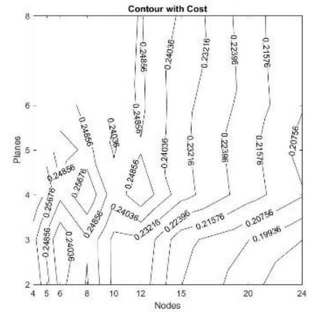
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost



i. Short-listed (10) architectures with cost

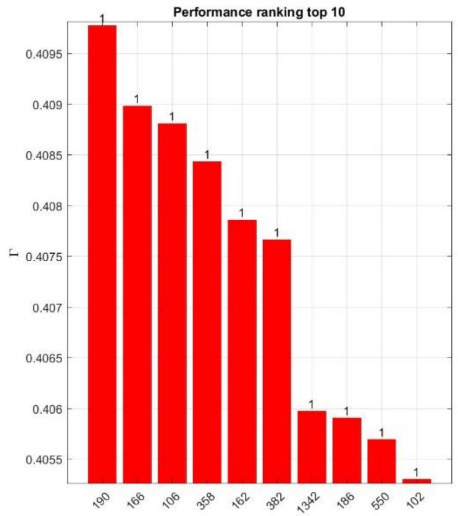


j. Iso-FoM with cost

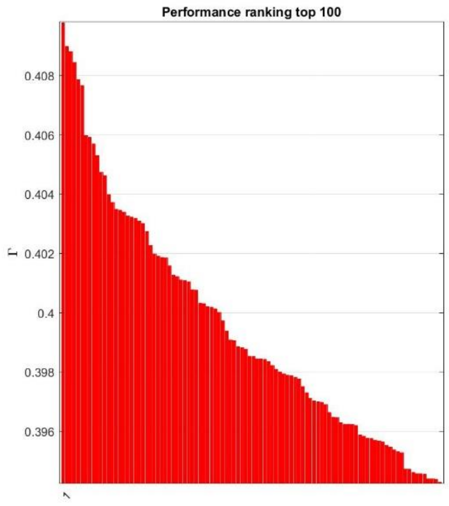
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

telecom BCN

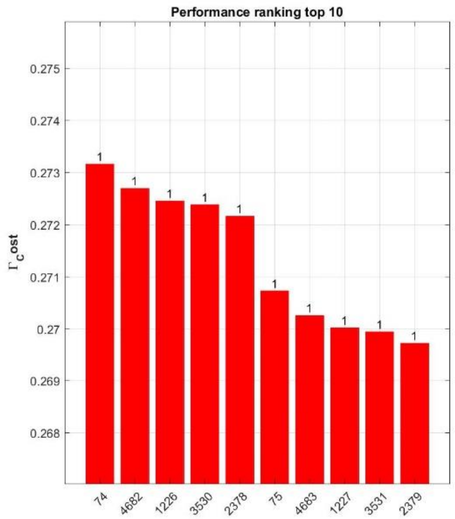
A – Equalized weights



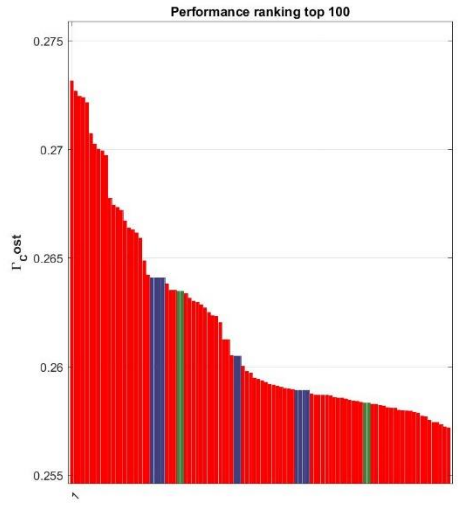
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)



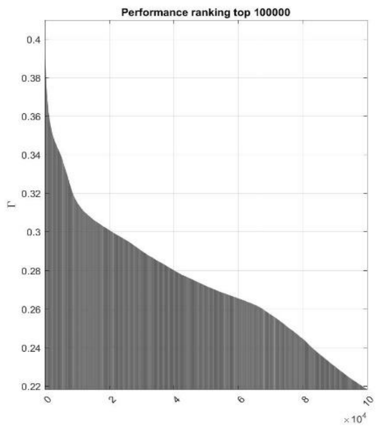
m. Best 10 architectures (with cost)



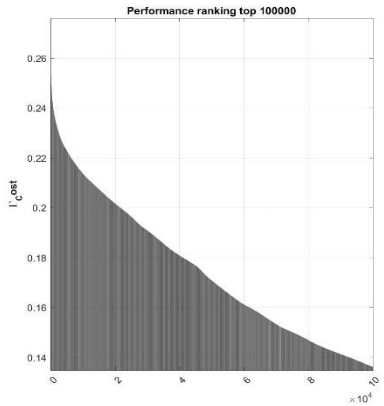
n. Best 10 architectures (with cost)

ONION vector

b_F	0.5
b_P	0.5
b_M	0.5
b_D	0.5
b_V	0.5



o. Best 100,000 architectures (without cost)



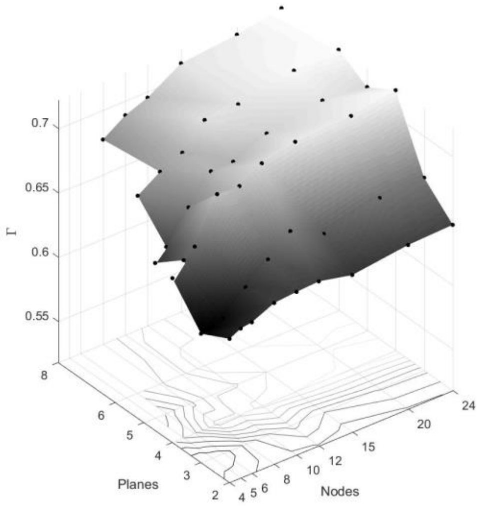
p. Best 100,000 architectures (with cost)

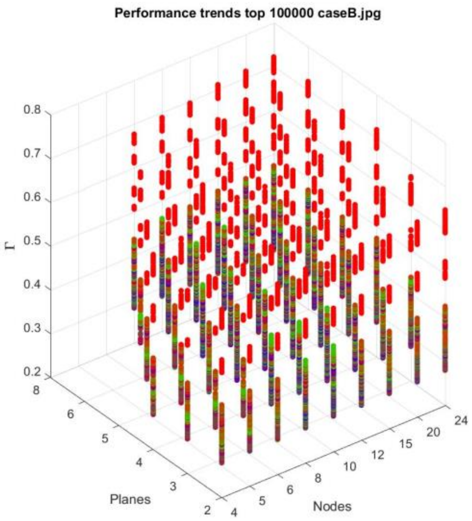
Parameter values

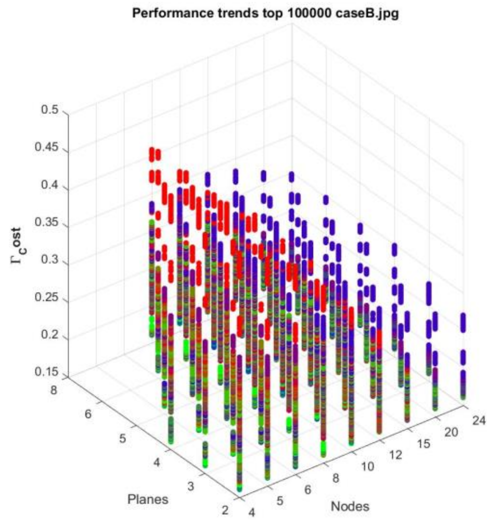
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

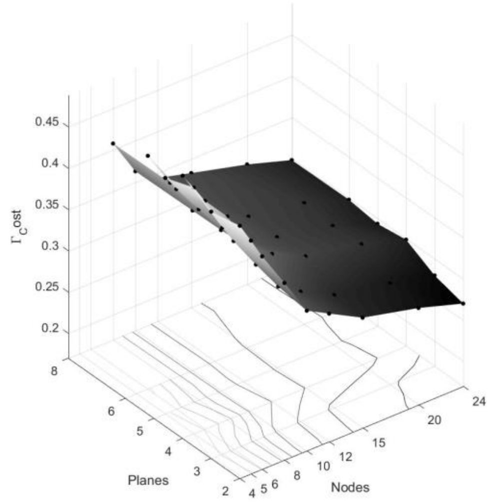
Table 30. Selection cases: Equalized case plots.

B – Conservative design







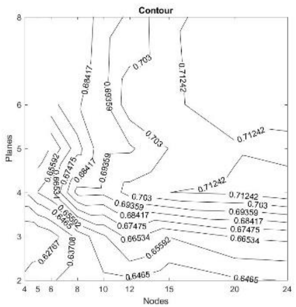


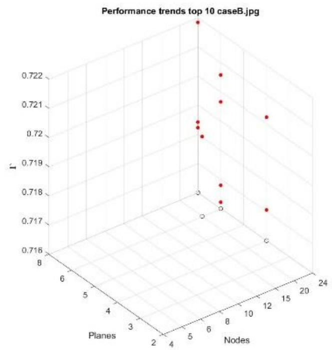
a. Interpolated maximum FoM without cost

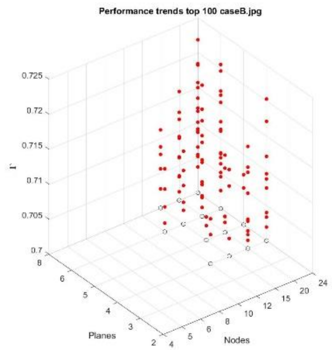
b. Design space without cost

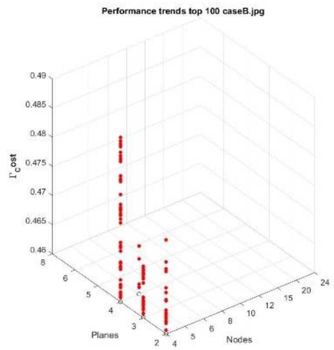
c. Design space with cost

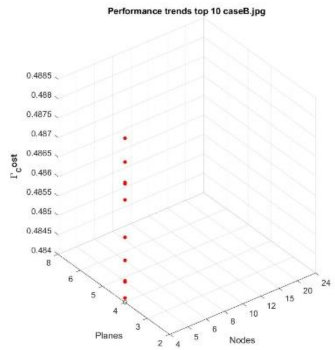
d. Interpolated maximum FoM with cost

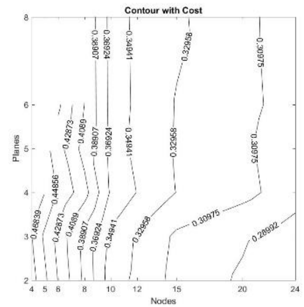












e. Iso-FoM without cost

f. Short-listed (10) architectures without cost

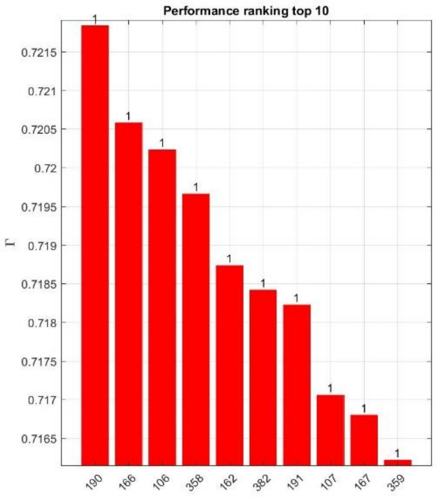
g. Extended-range (100) list of architectures without cost

h. Extended-range (100) list of architectures without cost

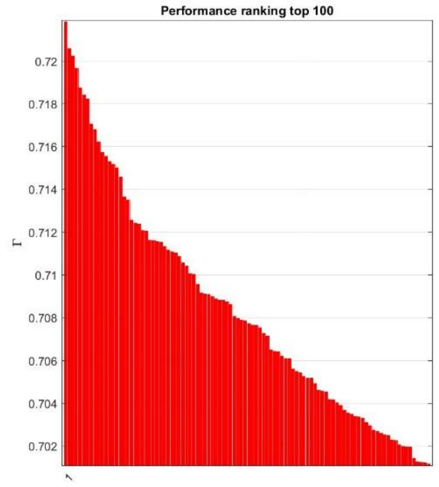
i. Short-listed (10) architectures with cost

j. Iso-FoM with cost

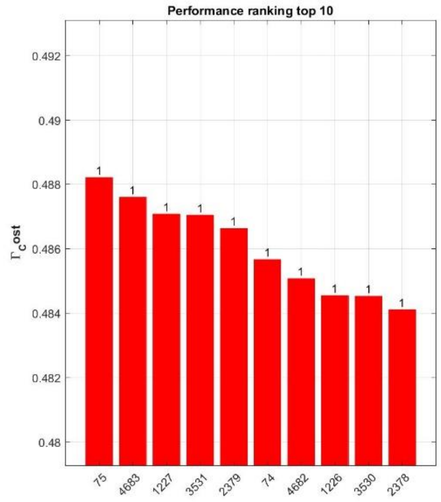
B – Conservative design



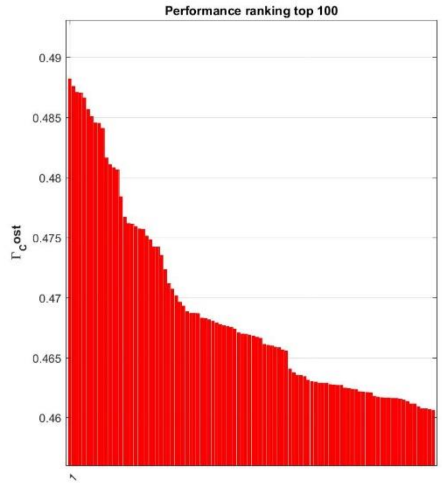
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

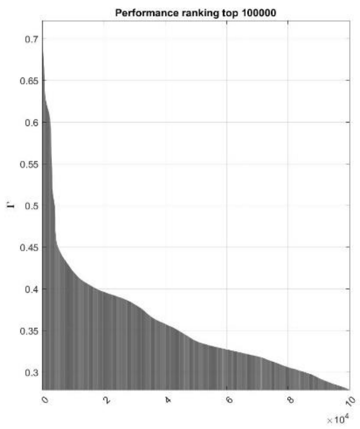


m. Best 10 architectures (with cost)

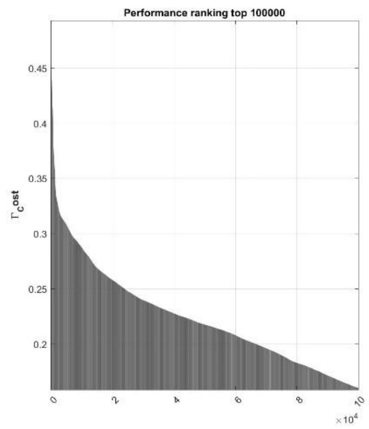


n. Best 10 architectures (with cost)

ONION vector	
$\boldsymbol{b_F}$	0.55
$\boldsymbol{b_P}$	0.4
$\boldsymbol{b_M}$	0.4
$\boldsymbol{b_D}$	0.55
$\boldsymbol{b_V}$	0.95



o. Best 100,000 architectures (without cost)

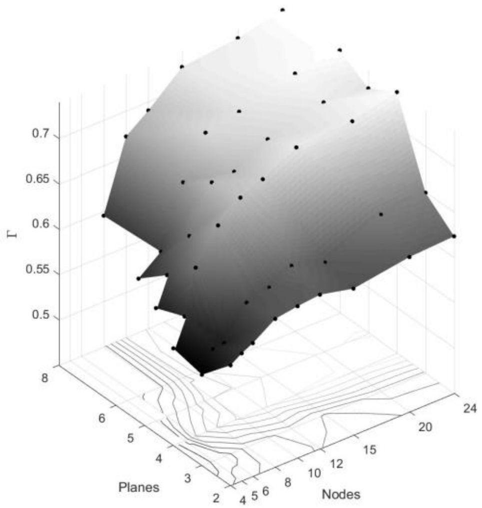


p. Best 100,000 architectures (with cost)

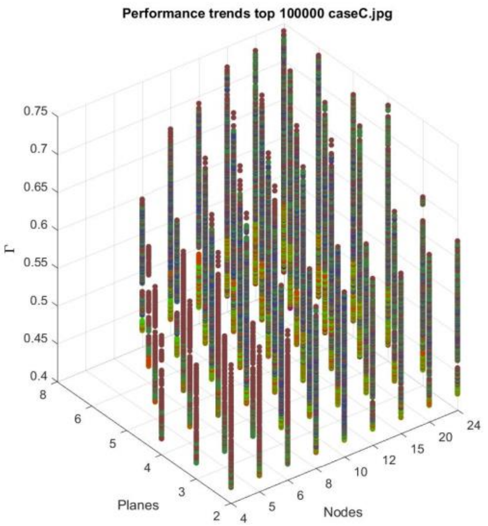
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 31. Selection cases: Conservative case plots.

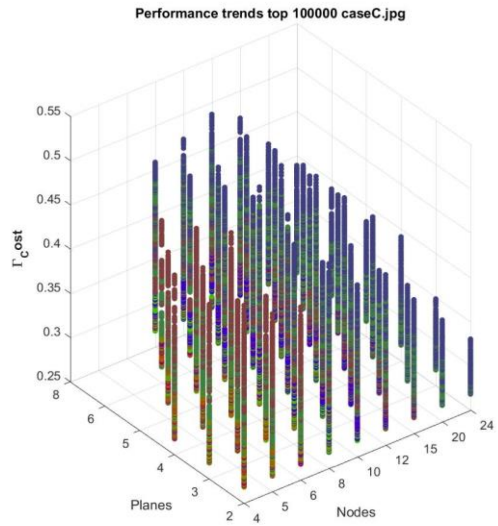
C – Bold designs



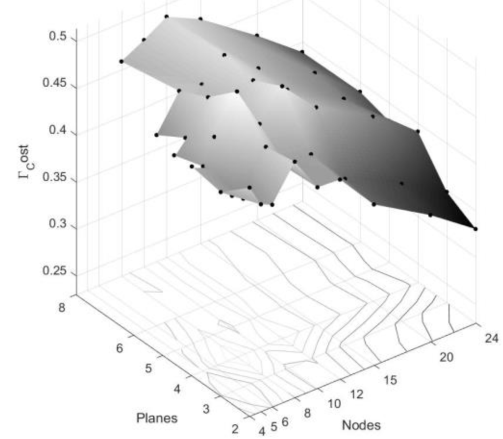
a. Interpolated maximum FoM without cost



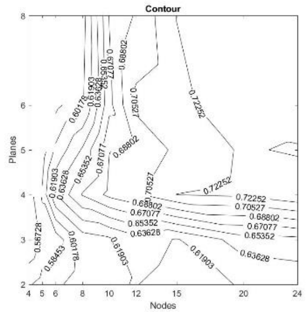
b. Design space without cost



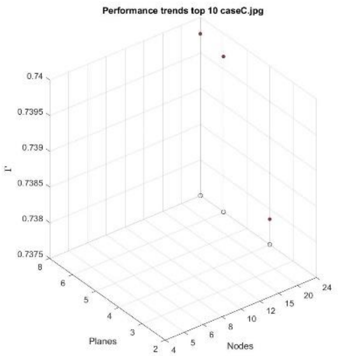
c. Design space with cost



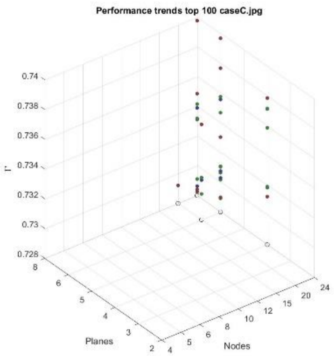
d. Interpolated maximum FoM with cost



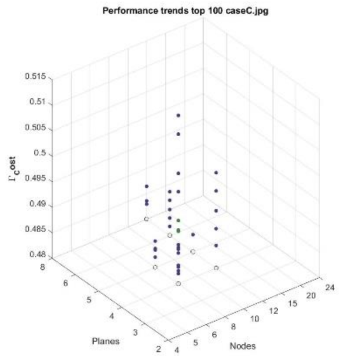
e. Iso-FoM without cost



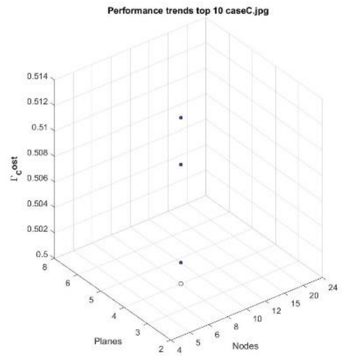
f. Short-listed (10) architectures without cost



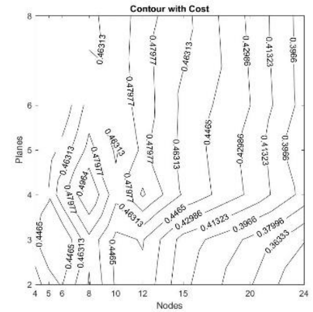
g. Extended-range (100) list of architectures without cost



h. Short-listed (10) architectures with cost

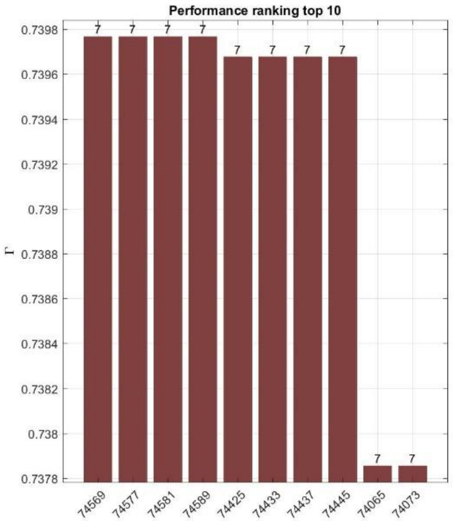


i. Extended-range (100) list of architectures without cost

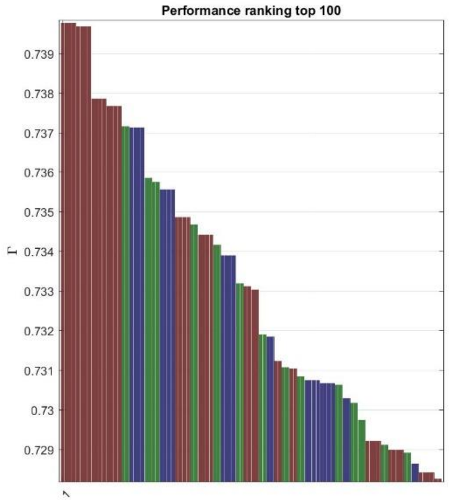


j. Iso-FoM with cost

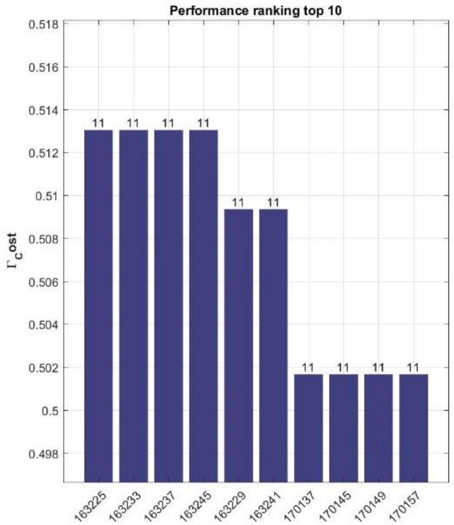
C – Bold designs



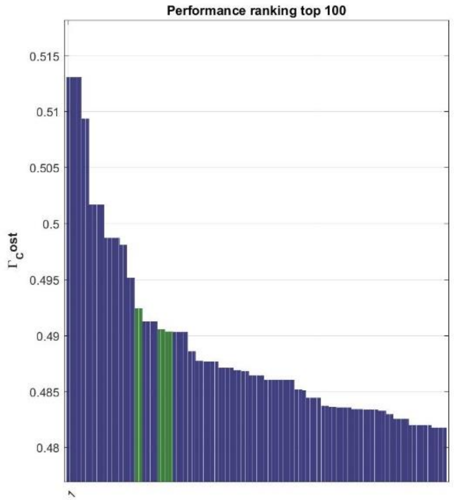
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

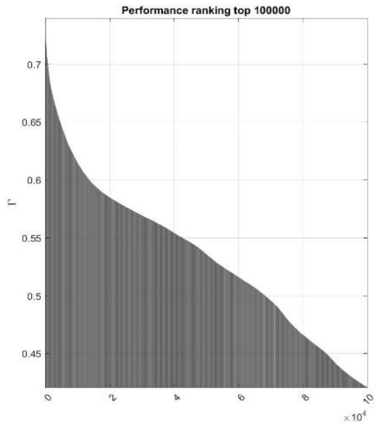


m. Best 10 architectures (with cost)

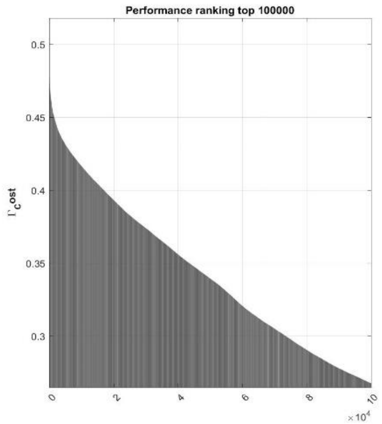


n. Best 10 architectures (with cost)

ONION vector	
b_F	0.95
b_P	0.85
b_M	0.85
b_D	0.95
b_V	0.4



o. Best 100,000 architectures (without cost)

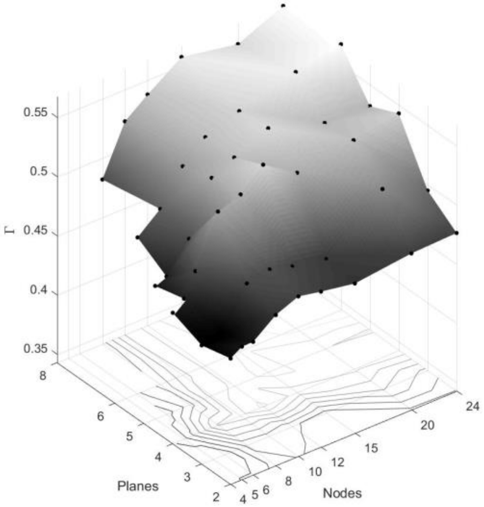


p. Best 100,000 architectures (with cost)

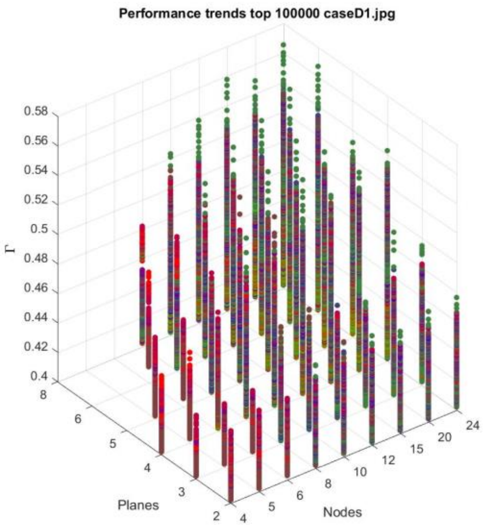
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 32. Selection cases: Bold case plots.

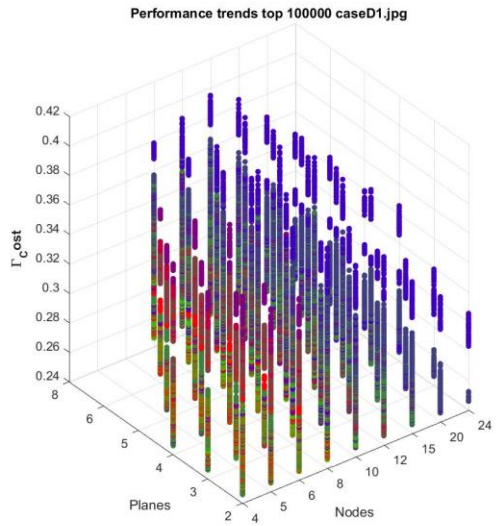
D – Consortium #1



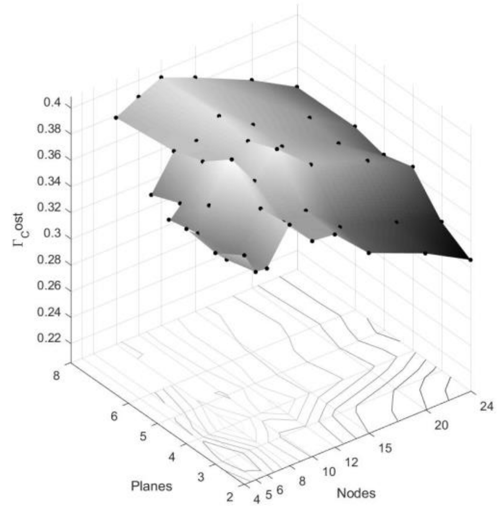
a. Interpolated maximum FoM without cost



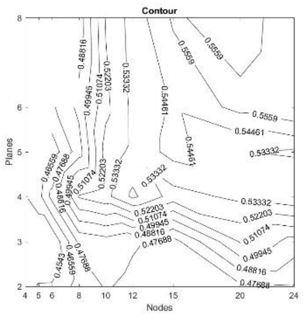
b. Design space without cost



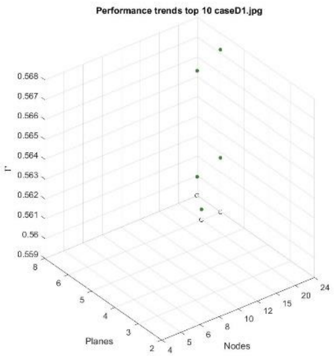
c. Design space with cost



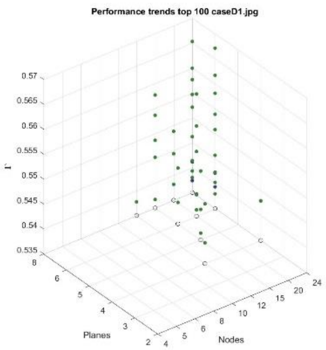
d. Interpolated maximum FoM with cost



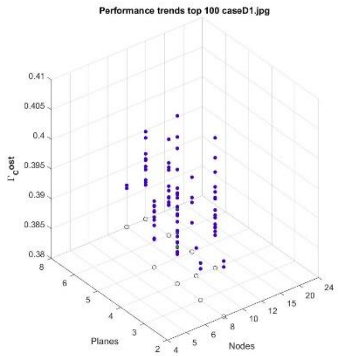
e. Iso-FoM without cost



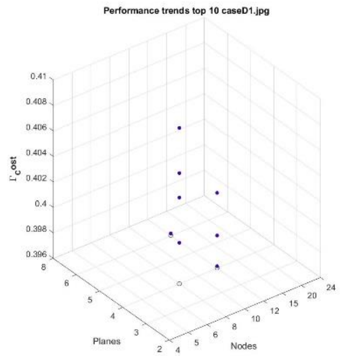
f. Short-listed (10) architectures without cost



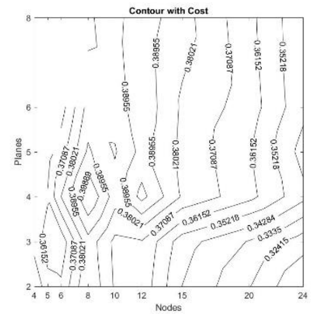
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost

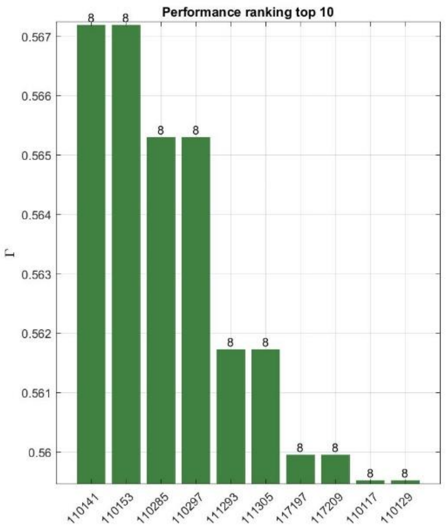


i. Short-listed (10) architectures with cost

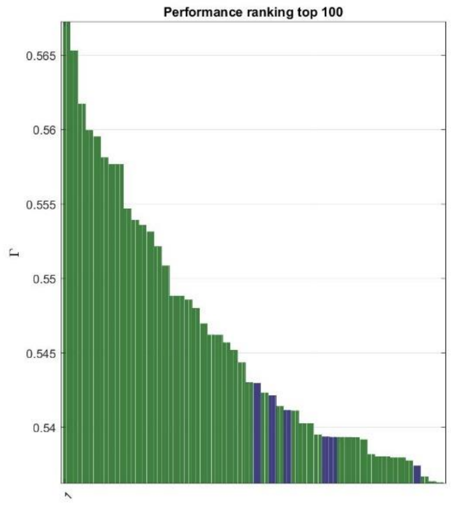


j. Iso-FoM with cost

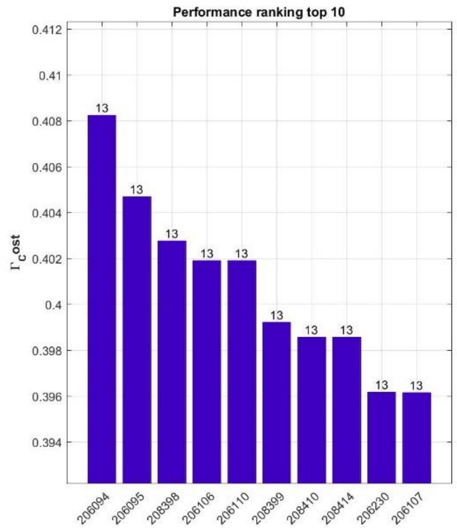
D – Consortium #1



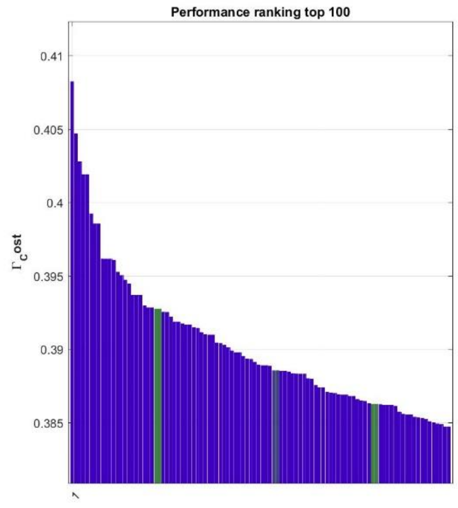
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

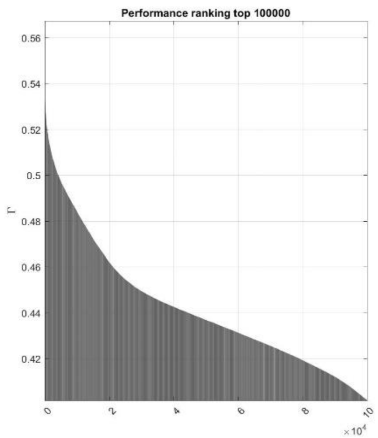


m. Best 10 architectures (with cost)

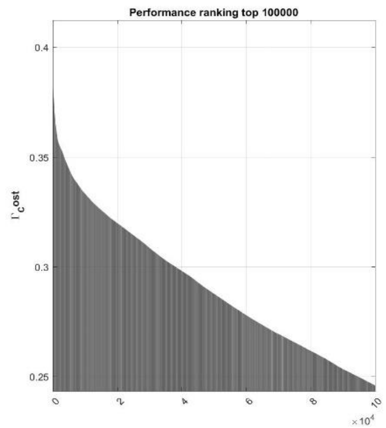


n. Best 10 architectures (with cost)

ONION vector	
$\boldsymbol{b_F}$	0.5
$\boldsymbol{b_P}$	0.75
$\boldsymbol{b_M}$	0.95
$\boldsymbol{b_D}$	0.5
$\boldsymbol{b_V}$	0.65



o. Best 100,000 architectures (without cost)



p. Best 100,000 architectures (with cost)

Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 33. Selection cases: Consortium proposed case plots.

III. Preliminary plots.

In this appendix can be seen the preliminary results that has been studied in order to improve both the optimization framework and its inputs. The following tables provide all the plots for the three cases analysed during the preliminary results chapter. The first one has 15 platform size distributions, the second one has 19 distributions and the last one has the results of the full combination of platforms sizes.

The following plots are provided:

Plot	Title	Remarks
a, d	Interpolated maximum FoM.	Black dots indicate maximum FoM for each point.
b, c	Design space	Only best 100,000 architectures.
e, j	Iso-FoM.	Contours with step of 10%.
f, i	Short-listed (10) architectures.	Transparent circles show location of columns.
g, h	Extended-range (100) set of architectures.	Idem.
k, m	Best 10 architectures.	Platform distribution option (1-15) shown over bars. Except for the last set of plot.
l, n	Best 100 architectures.	-
o, p	Best 100,000 architectures.	Colour information removed.

Table 34. Plots provided in appendix III.

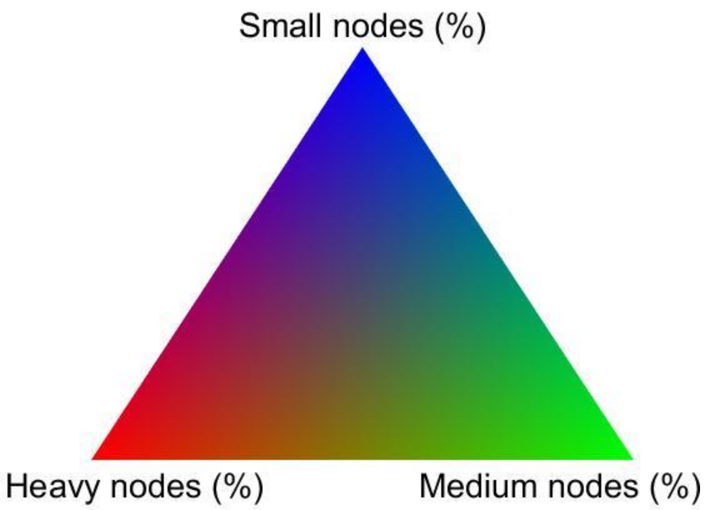
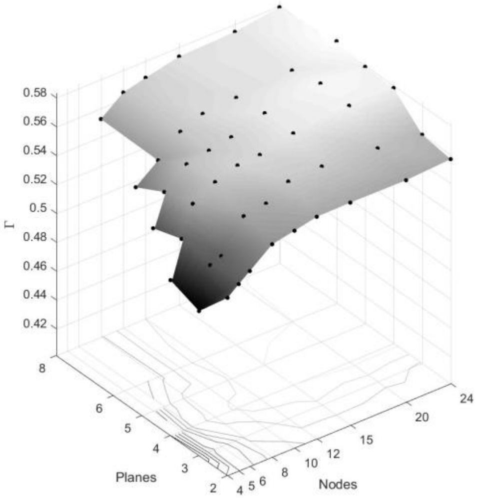
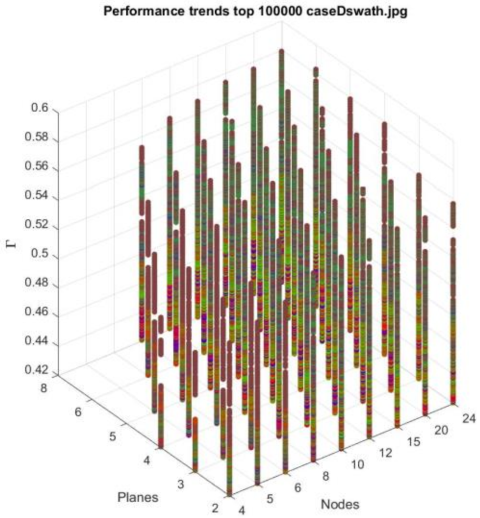


Figure 24. Platform distribution size colour gradient.

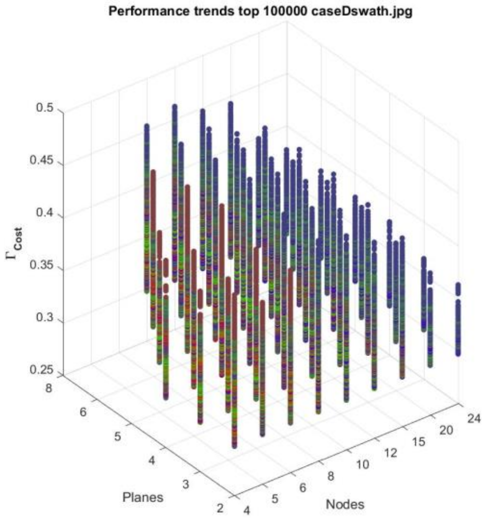
Preliminary Resutls: Frist iteration – 15 distributions



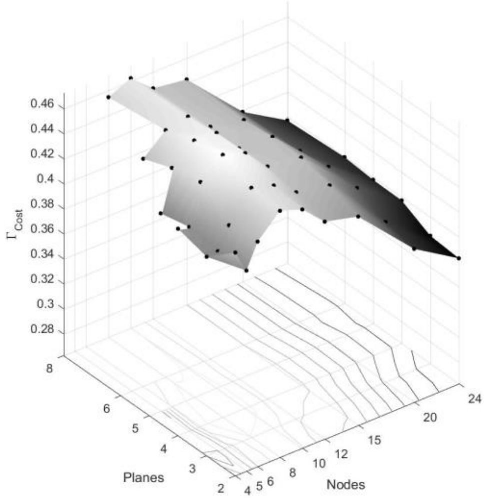
a. Interpolated maximum FoM without cost



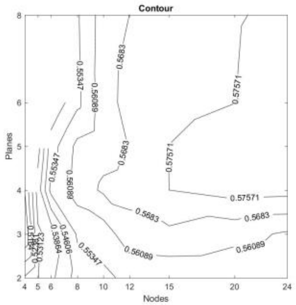
b. Design space without cost



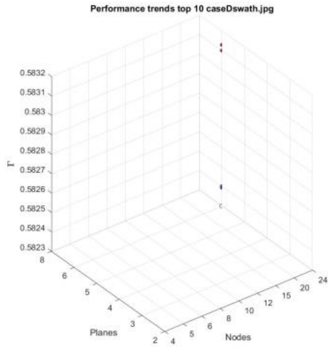
c. Design space with cost



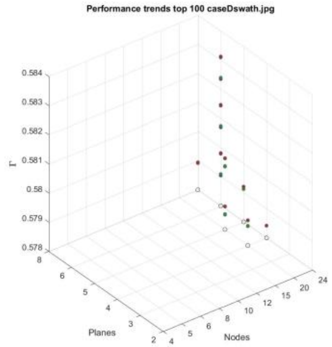
d. Interpolated maximum FoM with cost



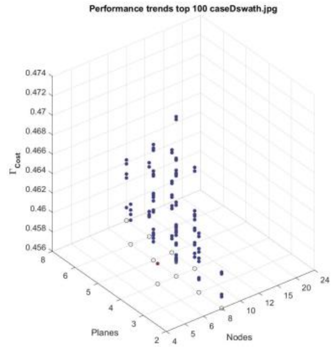
e. Iso-FoM without cost



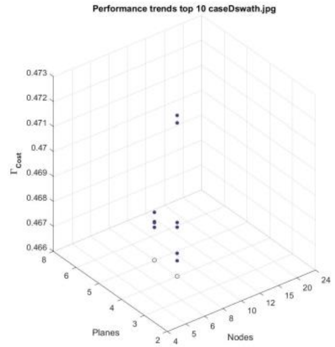
f. Short-listed (10) architectures without cost



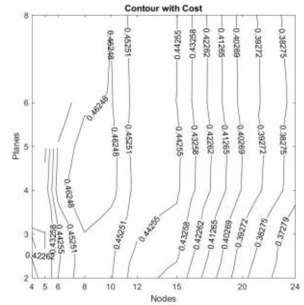
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost

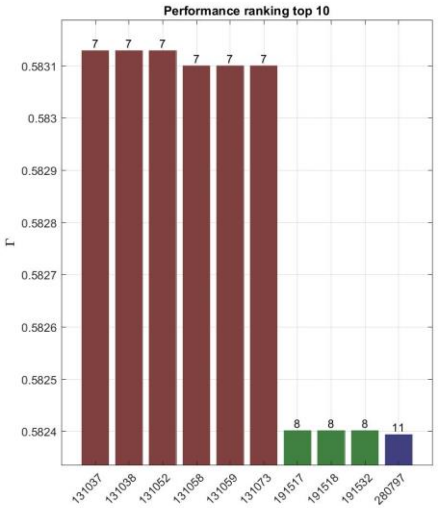


i. Short-listed (10) architectures with cost

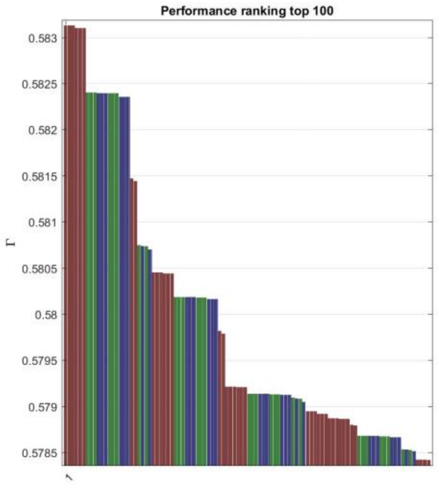


j. Iso-FoM with cost

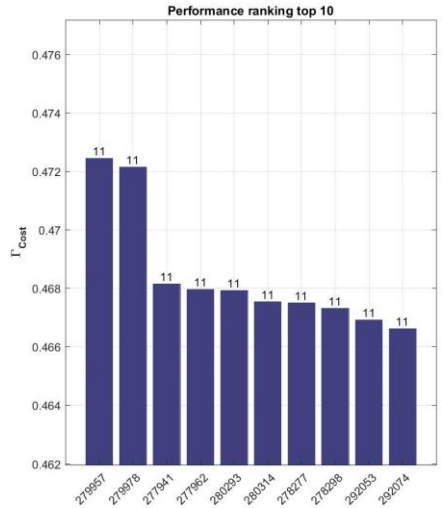
Preliminary Resutls: Frist iteration – 15 distributions



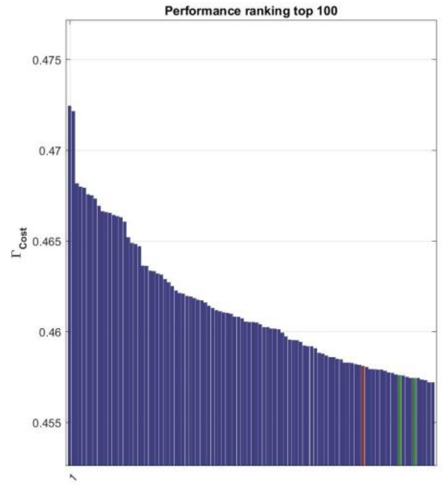
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

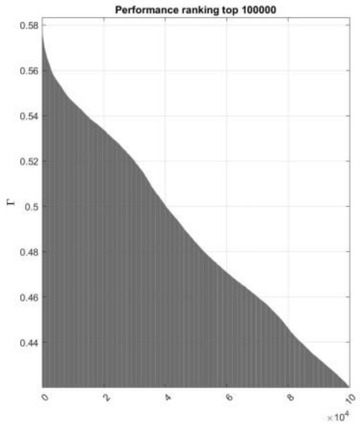


m. Best 10 architectures (with cost)

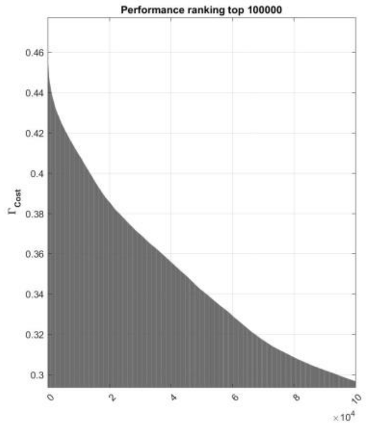


n. Best 10 architectures (with cost)

ONION vector	
$\boldsymbol{b_F}$	0.5
$\boldsymbol{b_P}$	0.75
$\boldsymbol{b_M}$	0.95
$\boldsymbol{b_D}$	0.5
$\boldsymbol{b_V}$	0.65



o. Best 100,000 architectures (without cost)



p. Best 100,000 architectures (with cost)

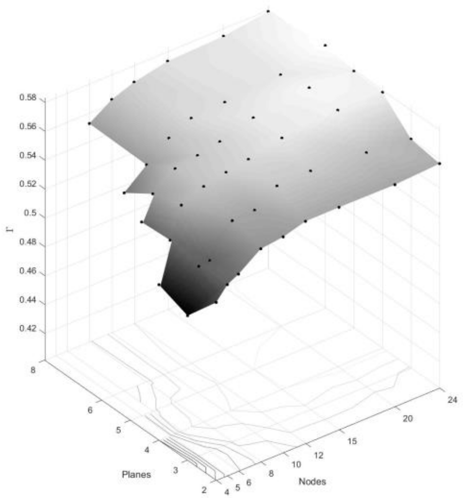
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 35. Preliminary results: First iteration - 15 platform size distributions.

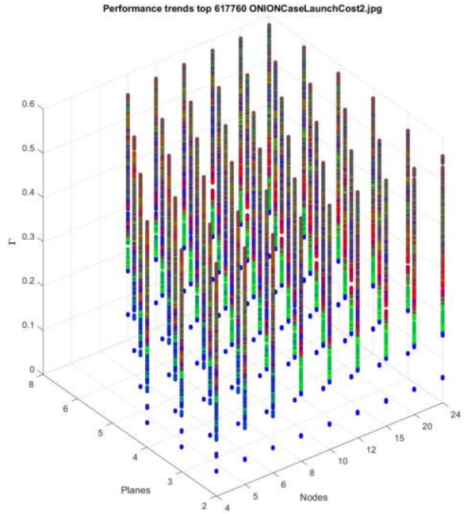
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

telecom BCN

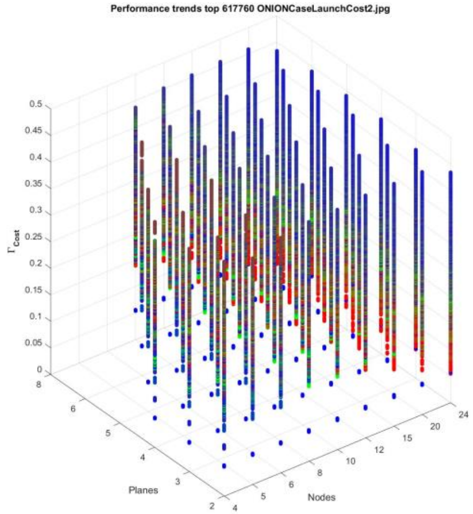
Preliminary Resutls: Second iteration – 19 distributions



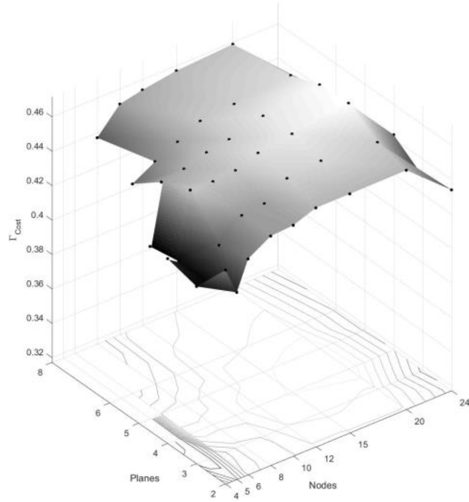
a. Interpolated maximum FoM without cost



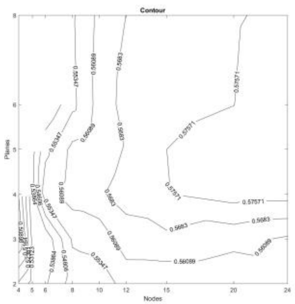
b. Design space without cost



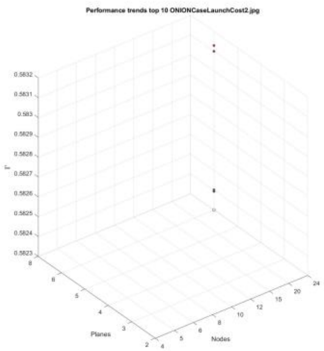
c. Design space with cost



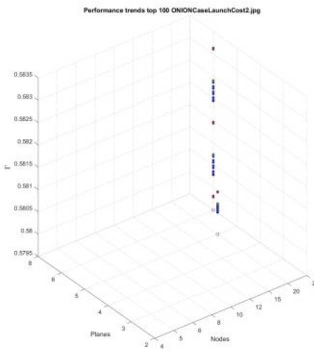
d. Interpolated maximum FoM with cost



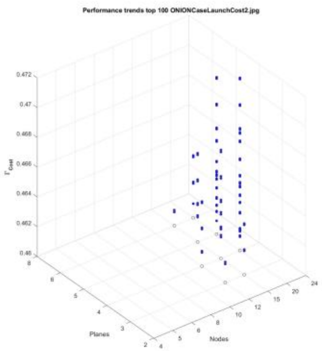
e. Iso-FoM without cost



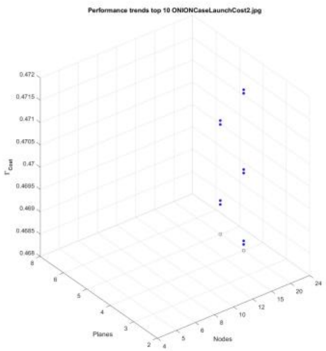
f. Short-listed (10) architectures without cost



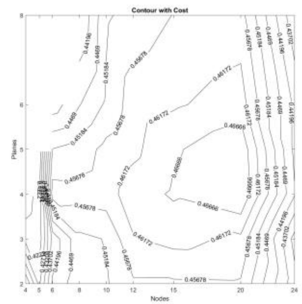
g. Extended-range (100) list of architectures without cost



h. Short-listed (10) architectures with cost

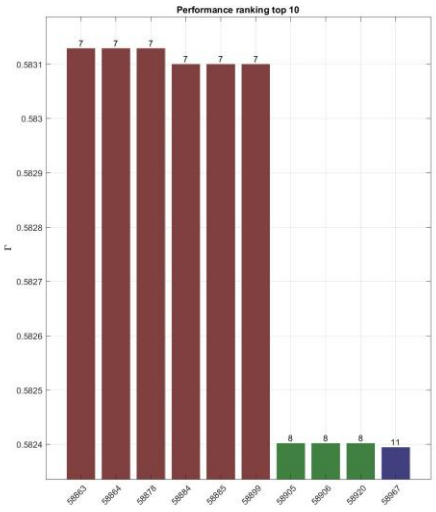


i. Extended-range (100) list of architectures without cost

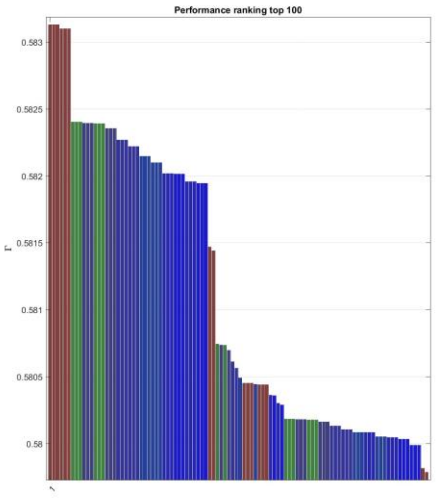


j. Iso-FoM with cost

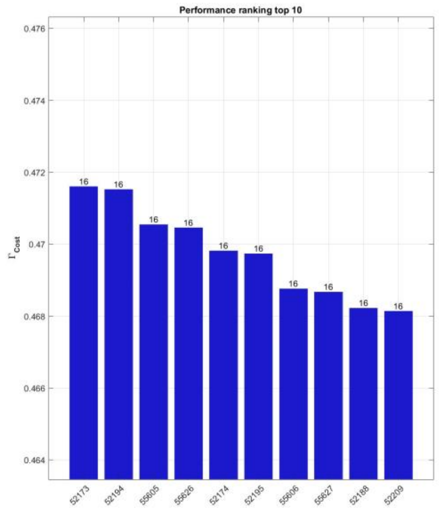
Preliminary Resutls: Second iteration – 19 distributions



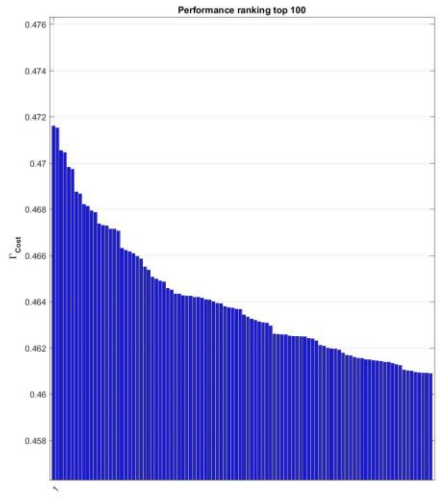
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

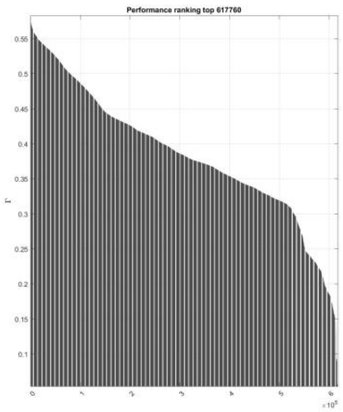


m. Best 10 architectures (with cost)

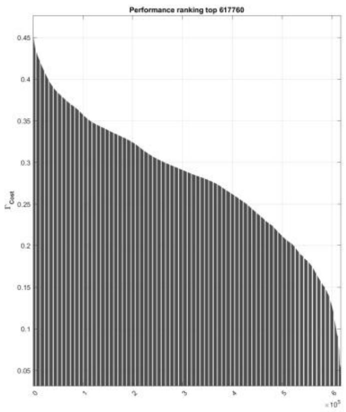


n. Best 10 architectures (with cost)

ONION vector	
$\boldsymbol{b_F}$	0.5
$\boldsymbol{b_P}$	0.75
$\boldsymbol{b_M}$	0.95
$\boldsymbol{b_D}$	0.5
$\boldsymbol{b_V}$	0.65



o. Best 100,000 architectures (without cost)

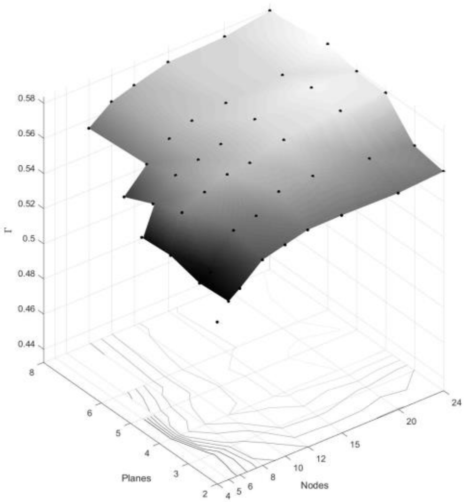


p. Best 100,000 architectures (with cost)

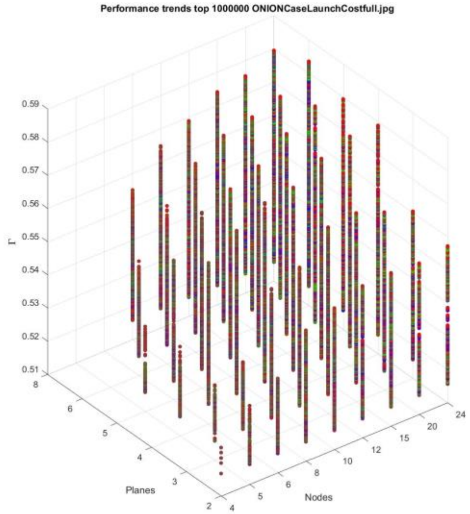
Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 36. Preliminary results: second iteration - 19 platform size distributions.

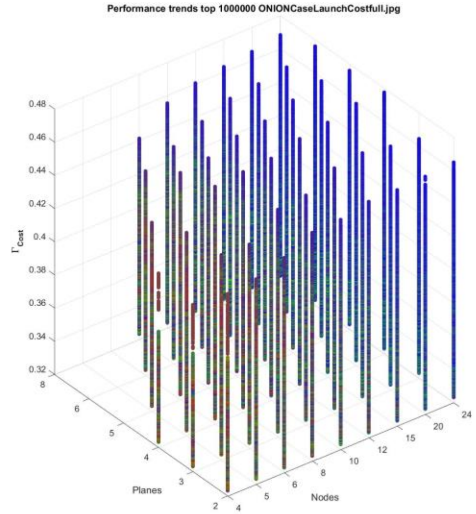
Preliminary Resutls: Third iteration – Full distributions



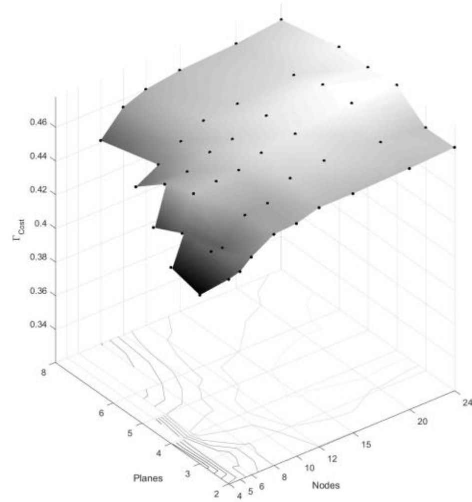
a. Interpolated maximum FoM without cost



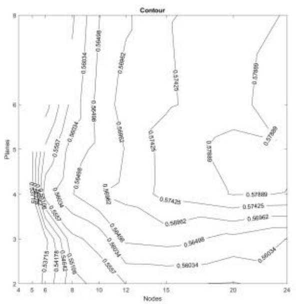
b. Design space without cost



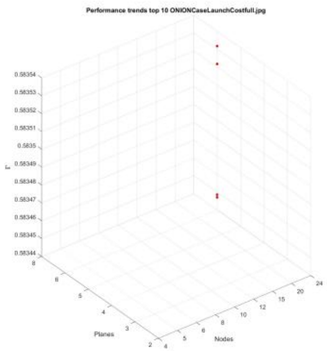
c. Design space with cost



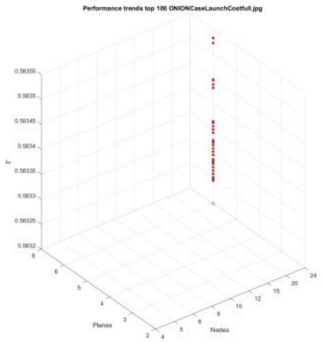
d. Interpolated maximum FoM with cost



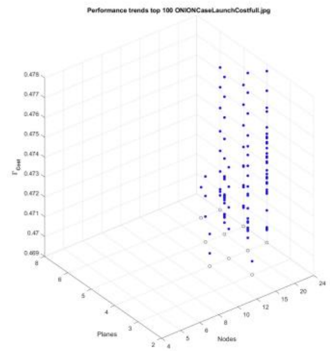
e. Iso-FoM without cost



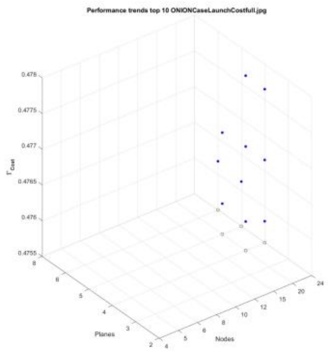
f. Short-listed (10) architectures without cost



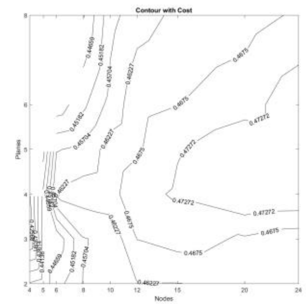
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost

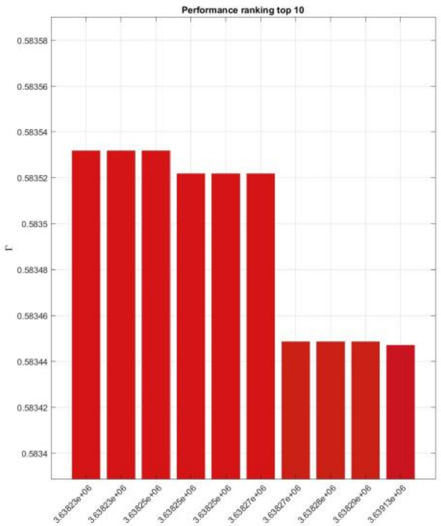


i. Short-listed (10) architectures with cost

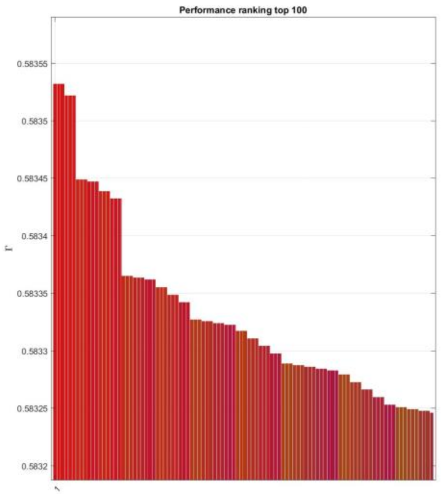


j. Iso-FoM with cost

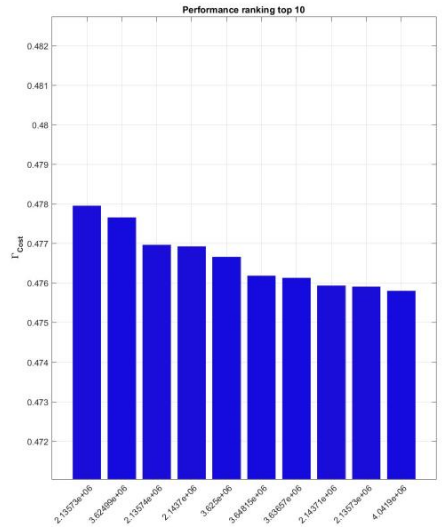
Preliminary Resutls: Third iteration – Full distributions



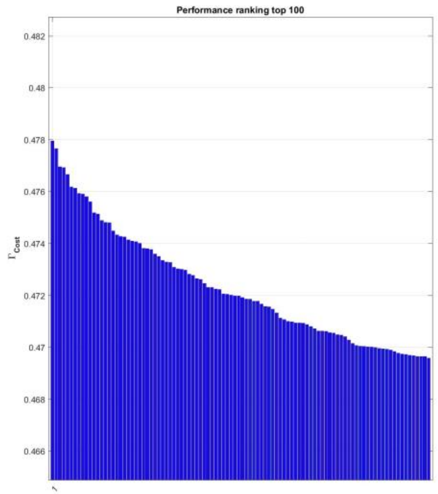
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)

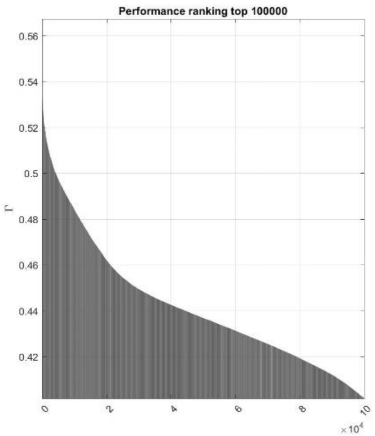


m. Best 10 architectures (with cost)

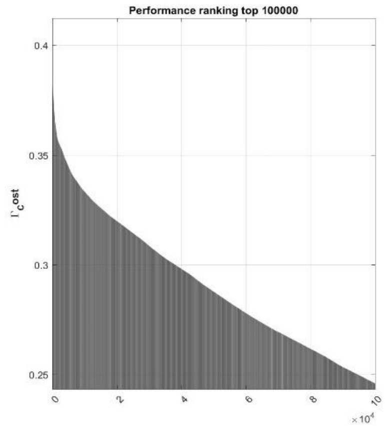


n. Best 10 architectures (with cost)

ONION vector	
b_F	0.5
b_P	0.75
b_M	0.95
b_D	0.5
b_V	0.65



o. Best 100,000 architectures (without cost)



p. Best 100,000 architectures (with cost)

Parameter values	
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Table 37. Preliminary results: third iteration – Full distributions.

IV. Final results.

This chapter contains the plots and the table of the final results for the MWF use-case study. Below can be found the same plots as in the other appendices but also the pareto frontier one and five new plots, the first four shows the influence on the final figure of merit of the number of heavy, medium or small nodes and, also the planes. The last plot shows the influence of the number of planes on the total cost of the architecture.

The following plots are provided:

Plot	Title	Remarks
a, d	Interpolated maximum FoM.	Black dots indicate maximum FoM for each point.
b, c	Design space	Only best 100,000 architectures.
e, j	Iso-FoM.	Contours with step of 10%.
f, i	Short-listed (10) architectures.	Transparent circles show location of columns.
g, h	Extended-range (100) set of architectures.	Idem.
k, m	Best 10 architectures.	-
l, n	Best 100 architectures.	-
o, p	Best 100,000 architectures.	Colour information removed.
q	Pareto Frontier	Full set of architectures (5.5 million), marked in pink the 30 first architectures.
r	Heavy nodes influence on FoM.	Marked in pink the 30 first architectures.
s	Medium nodes influence on FoM.	Marked in pink the 30 first architectures.
t	Small nodes influence on FoM.	Marked in pink the 30 first architectures.
u	Planes influence on FoM	Marked in pink the 30 first architectures.
v	Planes influence on Cost	Marked in pink the 30 first architectures.

Table 38. Plots provided in appendix IV.

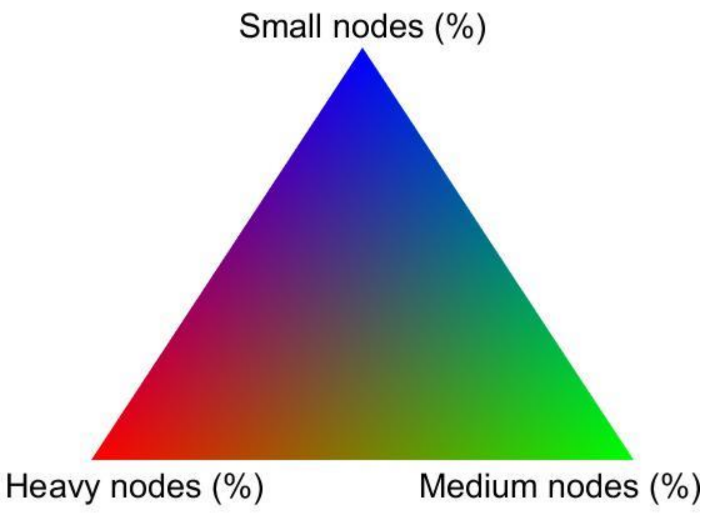
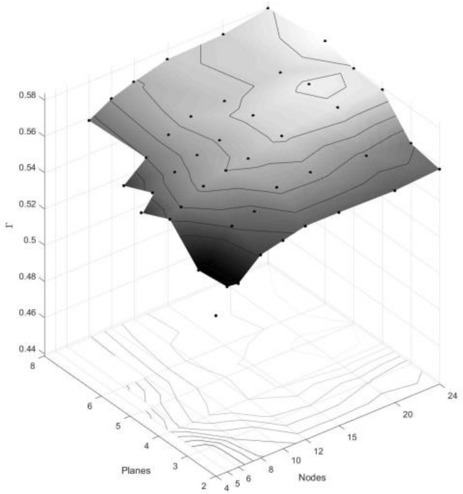
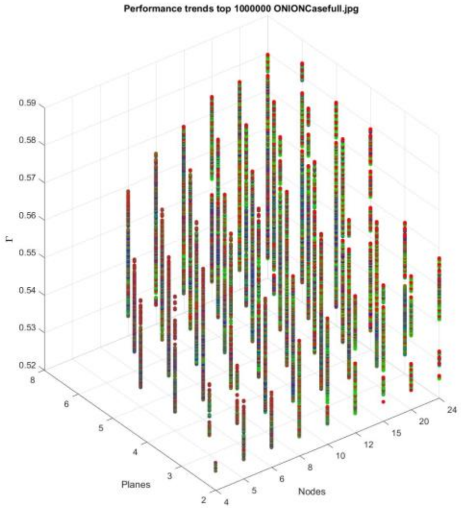


Figure 25. Platform distribution size colour gradient.

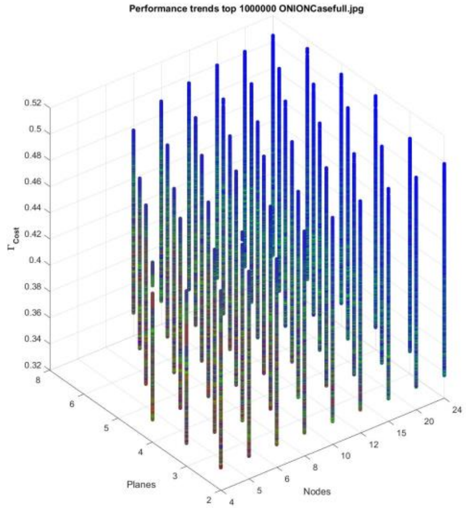
Final Resutls



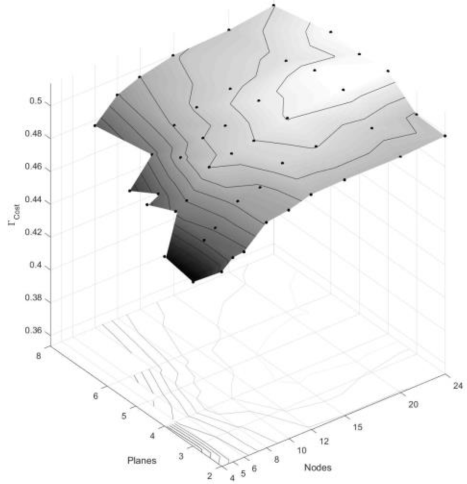
a. Interpolated maximum FoM without cost



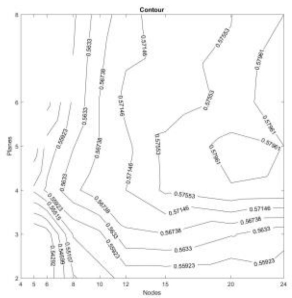
b. Design space without cost



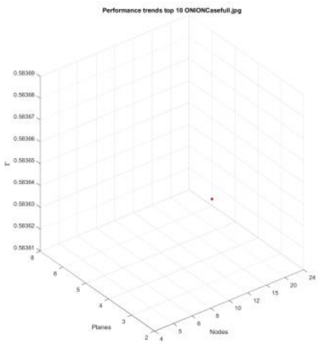
c. Design space with cost



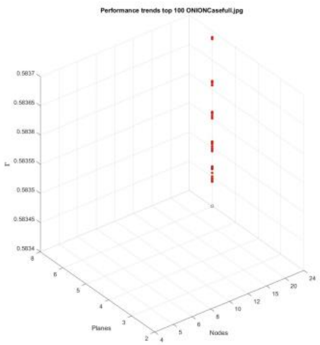
d. Interpolated maximum FoM with cost



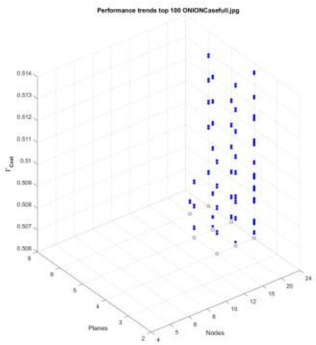
e. Iso-FoM without cost



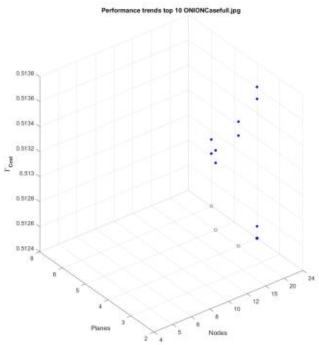
f. Short-listed (10) architectures without cost



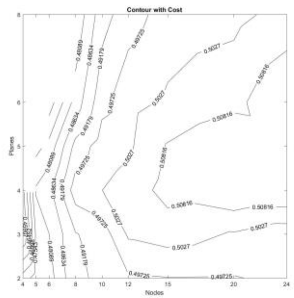
g. Extended-range (100) list of architectures without cost



h. Extended-range (100) list of architectures without cost

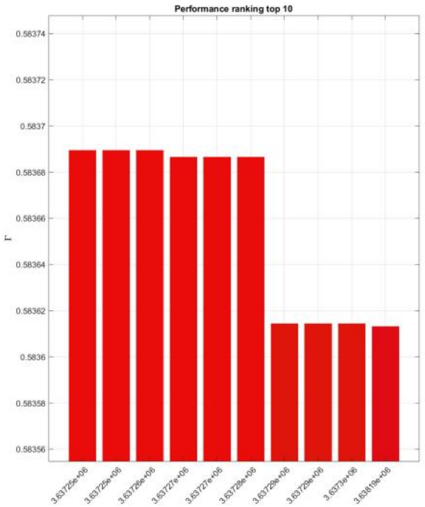


i. Short-listed (10) architectures with cost

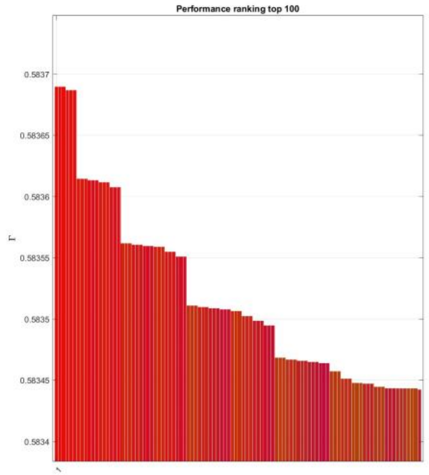


j. Iso-FoM with cost

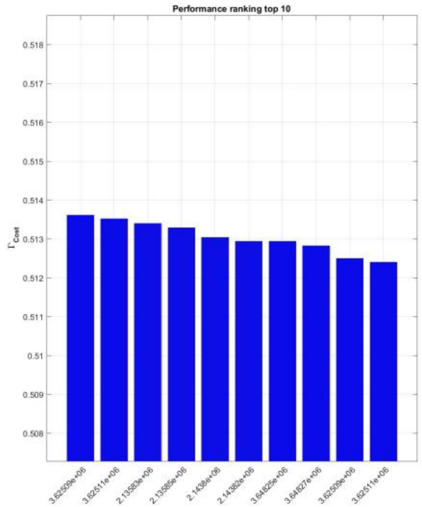
Final Resutls



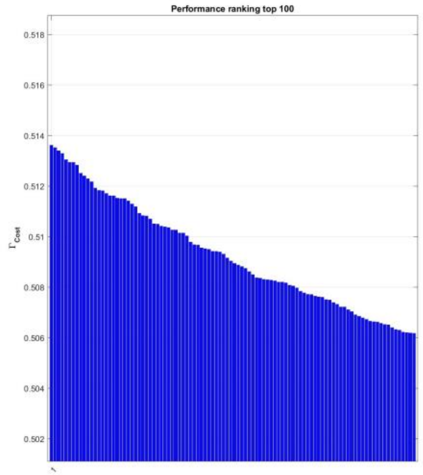
k. Best 10 architectures (without cost)



l. Best 100 architectures (without cost)



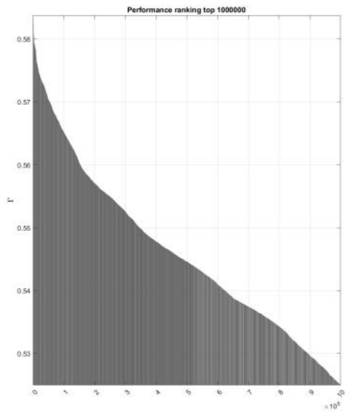
m. Best 10 architectures (with cost)



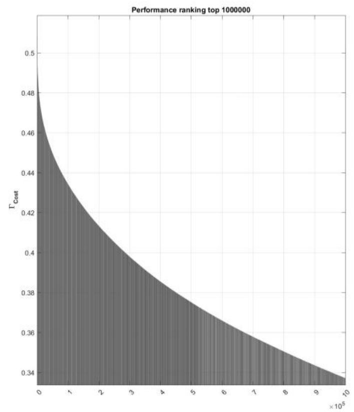
n. Best 10 architectures (with cost)

ONION vector

$\boldsymbol{b_F}$	0.5
$\boldsymbol{b_P}$	0.75
$\boldsymbol{b_M}$	0.95
$\boldsymbol{b_D}$	0.5
$\boldsymbol{b_V}$	0.65



o. Best 100,000 architectures (without cost)

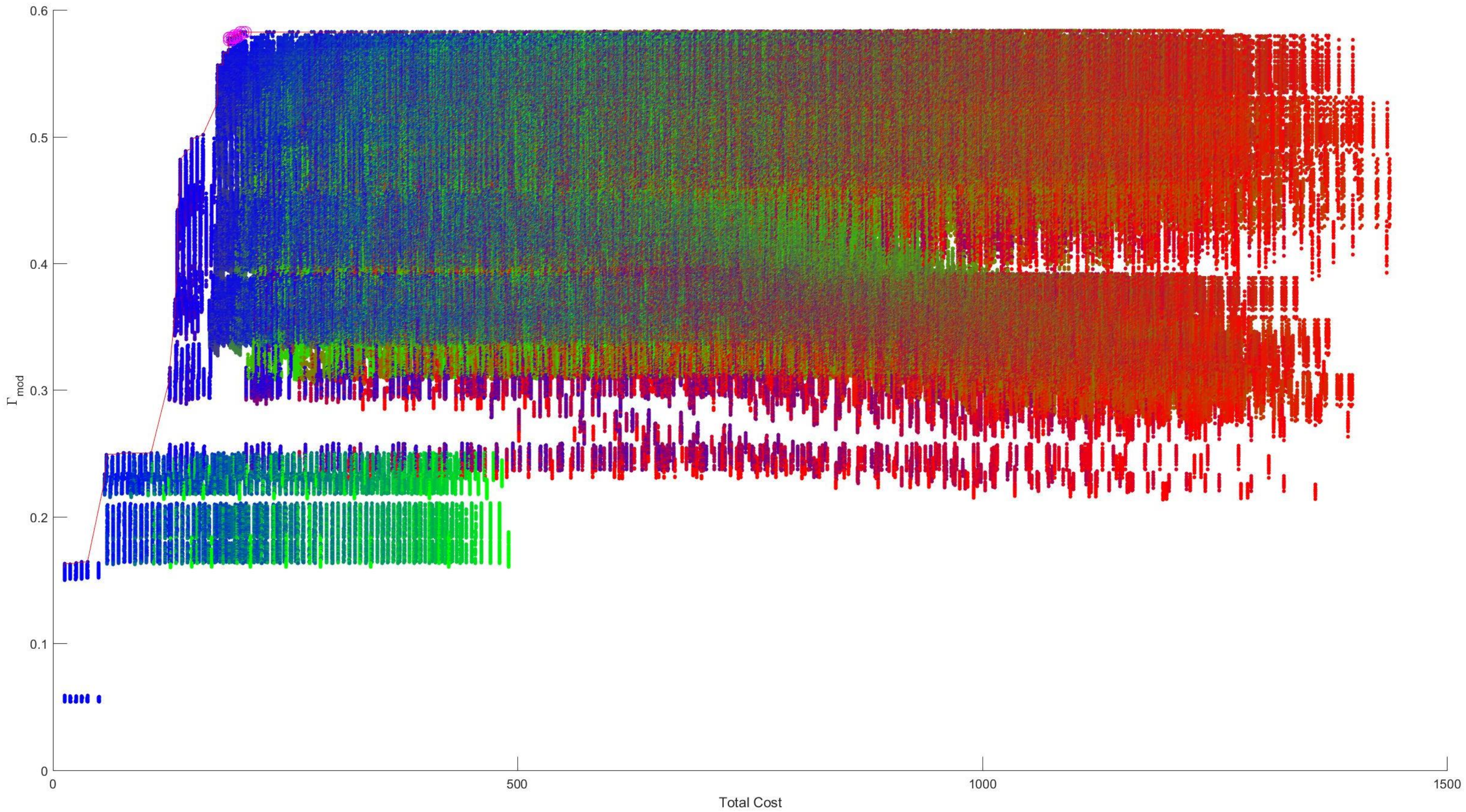


p. Best 100,000 architectures (with cost)

Parameter values

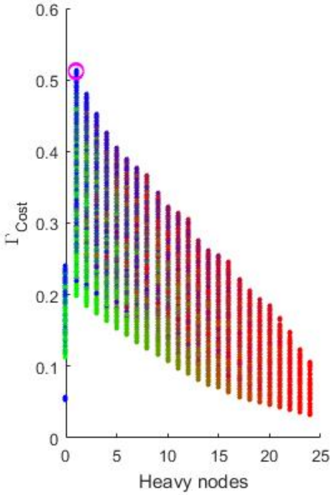
P	3
Q	0.1
$\gamma_{spatial}$	0.33
$\gamma_{revisit}$	0.33
$\gamma_{latency}$	0.33

Final Resutls

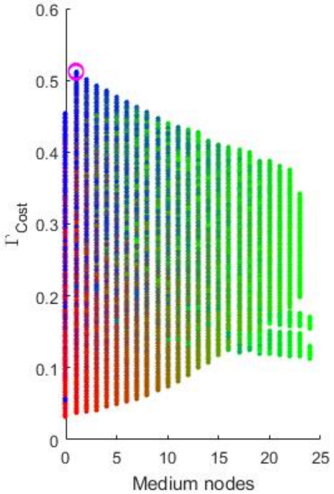


q. Pareto Frontier

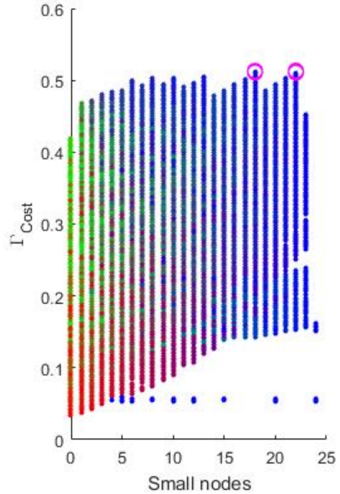
Final Resutls



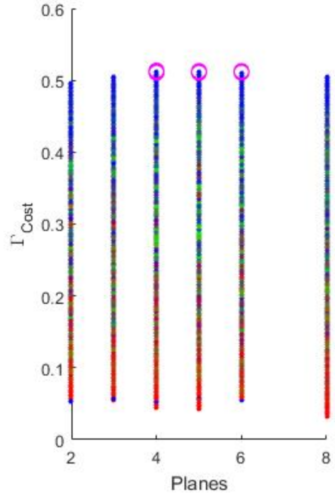
r. Heavy nodes influence on FoM.



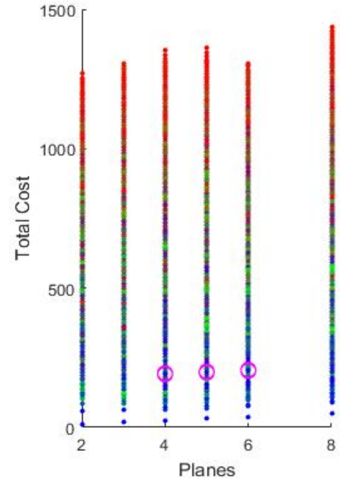
s. Medium nodes influence on FoM



t. Small nodes influence on FoM



u. Planes influence on FoM



v. Planes influence on Cost

Table 39. Final results.

V. Work Packages and Milestones.

In this appendix can be found the detailed work packages and the milestones.

Work Packages:

Project: Optimization framework development	**WP ref: 1**	
Major constituent: Research and design of the framework	Sheet 1 of 5	
Short description: Develop the optimization environment to return a normalized aggregated metric that allow us to carry out the optimization study.	Planned start date: 10/02/2017 Planned end date: 01/05/2017	
Internal task T1: Research of previous work on optimization field and documentation of ONION project. **Internal task T2:** Design the main methodology for the optimization framework. **Internal task T3:** Develop the final optimization framework.	Deliverables: (see milestones below)	Dates: (see milestones below)

Table 40. WP 1: Optimization framework development.

Project: Matlab framework	**WP ref: 2**	
Major constituent: Matlab Software	Sheet 2 of 5	
Short description: Design and develop the framework capable of running the previous algorithm.	Planned start date: 24/02/2017 Planned end date: 10/05/2017	
Internal task T1: Design an automatic framework taking into account future changes on the use case or new use cases. **Internal task T2:** Develop the code to compute the FoM without ilities. **Internal task T3:** Develop the code to compute the modifiers and the final FoM. **Internal task T4:** Create the input files template.	Deliverables: (see milestones below)	Dates: (see milestones below)

Table 41. WP 2: Matlab framework.

Project: Architecture Study	**WP ref: 3**	
Major constituent: Data processing	Sheet 3 of 5	
Short description: Make a study of the data generated by the Matlab framework in order to provide the winner architectures.	Planned start date: 24/02/2017 Planned end date: 19/05/2017	
Internal task T1: Prepare the code to create the desired output files. **Internal task T2:** Run the Matlab framework with all the input data and generate the first candidate selection. **Internal task T3:** Process and study the data and provide the winner architectures.	Deliverables: (see milestones below)	Dates: (see milestones below)

Table 42. WP 3: Architecture Study.

Project: Documentation	**WP ref: 4**	
Major constituent: Documentation	Sheet 4 of 5	
Short description: TFG Final report writing.	Planned start date: 15/02/2017 Planned end date: 20/06/2017	
Internal task T1: Project Proposal. **Internal task T2:** Critical Review report. **Internal task T3:** Final report.	Deliverables: (see milestones below)	Dates: (see milestones below)

Table 43. WP 4: Documentation.

Project: Sensor combinations	**WP ref: 5**	
Major constituent: Survey and analysis.	Sheet 4 of 5	
Short description: Survey of the instruments that will be used on the architecture generation and	Planned start date: 15/02/2017 Planned end date: 24/02/2017	
Internal task T1: Instruments **Internal task T2:** Platform combinations	Deliverables: (see milestones below)	Dates: (see milestones below)

Table 44. WP5: Sensor combinations.

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
1	1	Initial Research	-	
1	2	Design optimization framework	Previous Formulation PDF	20/02/2017
1	3	Implement optimization framework	Final Formulation PDF	31/03/2017
1	4	Ilities model	Ilities Formulation PDF	15/04/2017
2	1	Design framework	-	05/03/2017
2	3	Implement framework	Plots and figures PDF	30/04/2017
2	4	Input files	Input files template	05/05/2017
3	2	Candidate preselection	First selection of architectures	27/04/2017
3	2	Definitive Candidate selection	List of 30 architectures selected PDF	10/05/2017
3	2	Final Candidate selection	Study and results PDF	15/06/2017
4	1	Project Proposal	Project Proposal PDF	03/03/2017
4	2	Critical Review Report	Critical Report PDF	07/05/2017
4	3	Final Report	Final Report PDF	20/06/2017

Table 45. Milestones.

Glossary

TFG: Degree Thesis.

ONION: Operational Network of Individual Observation Nodes (European project).

OASF: ONION Architecture Selection Framework.

FSS: Federated Satellite Systems.

FoM: Figure of Merit.

TLR: Technology Readiness Level.

DMC: Disaster Monitoring Constellation.

MWF: Marine Weather Forecast.

DSS: Distributed Satellite Systems.