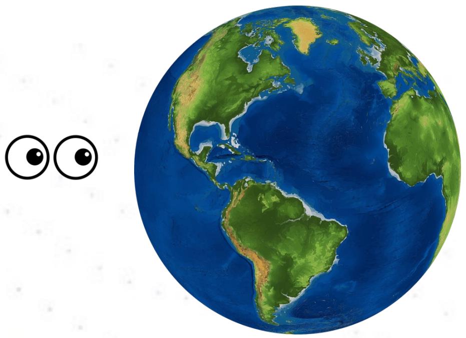
Constraint Model for the Satellite Image Mosaic Selection Problem

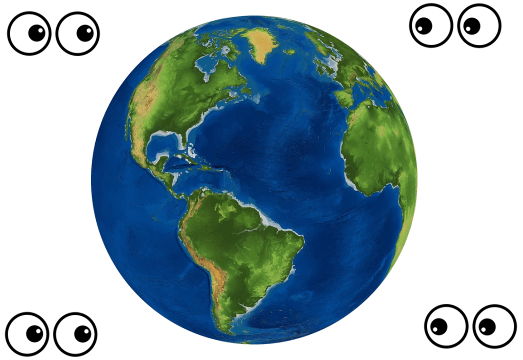
Manuel Combarro Simón, Pierre Talbot, Grégoire Danoy, Jedrzej Musial, Mohammed Alswaitti, Pascal Bouvry

Interdisciplinary Centre for Security, Reliability and Trust (SnT), Luxembourg
Poznan University of Technology, Poland

Increase in satellite imagery and applications



2014
192 EO satellites



2021
971 EO satellites

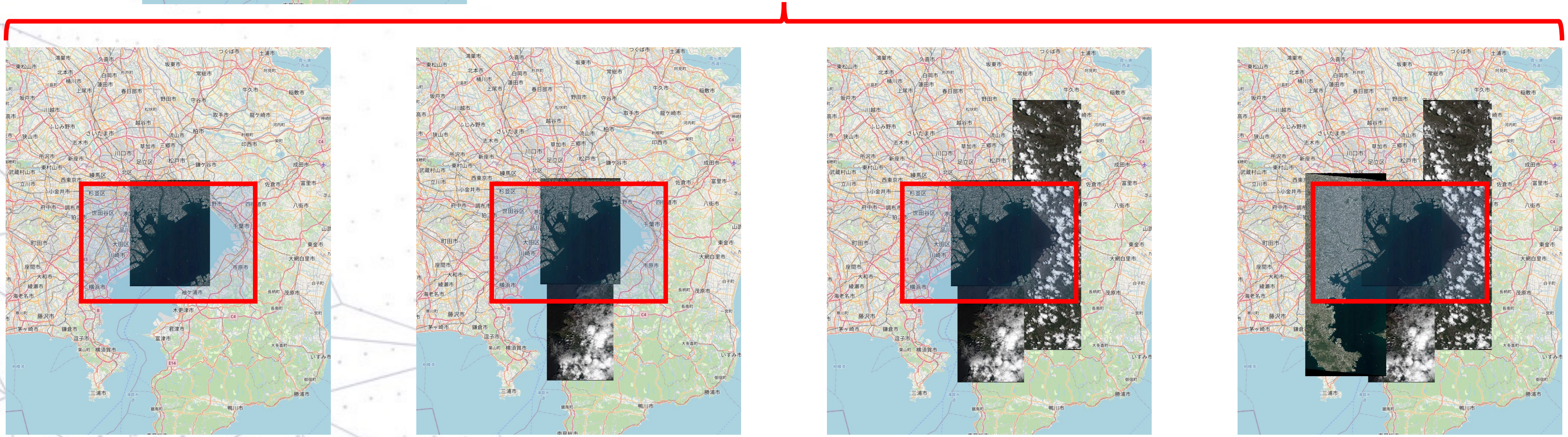
>100 TB of satellite imagery per **day**

Satellite images: Planet Labs, Inc.

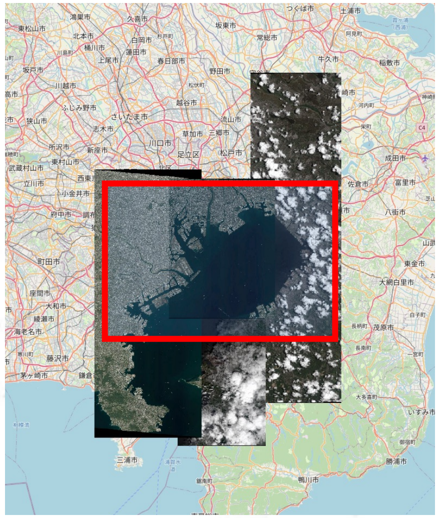
Satellite image mosaic

To cover large areas it is necessary

to merge several images together

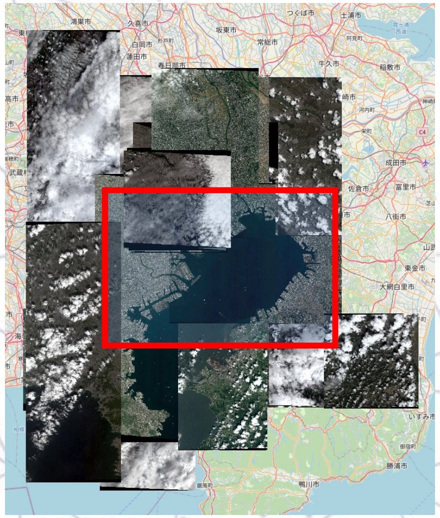


Satellite image mosaic



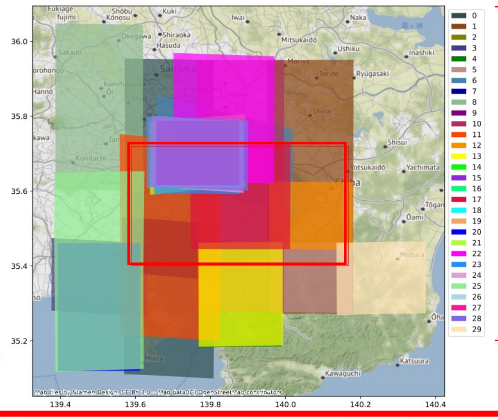
Related state of the art problems for mosaic generation:

- Geometric correction of the images
- Color harmonization
- Image stitching



Combinatorial problem of selecting the images to generate the mosaic

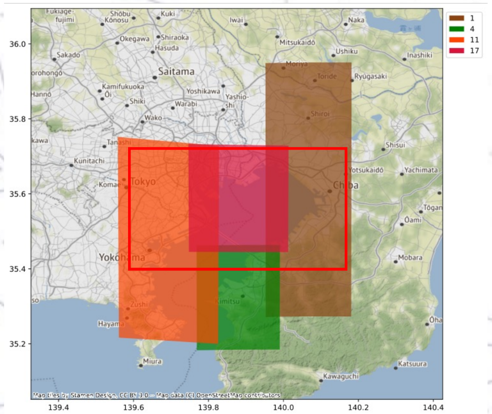
Satellite image mosaic



After a query with certain parameters:

N = 30 satellite images

Find the cover with the minimum number of images (NP-Hard)

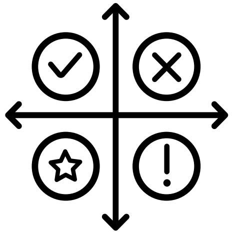


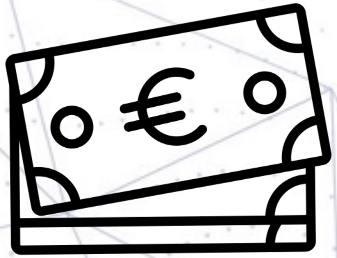
Build the mosaic

What it means “the best”?

Multi-objective problem:

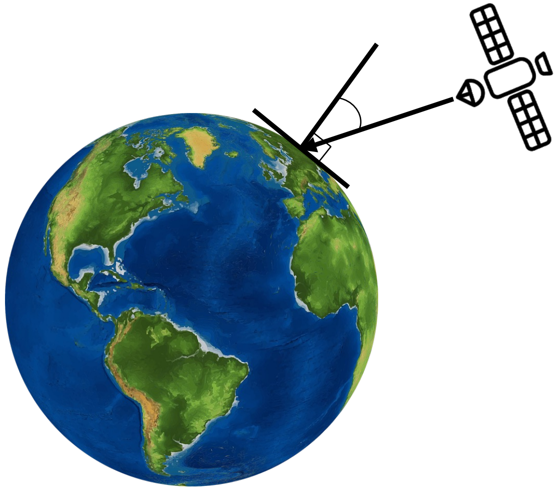
- Cost
- Cloud coverage
- Resolution
- Incidence angle





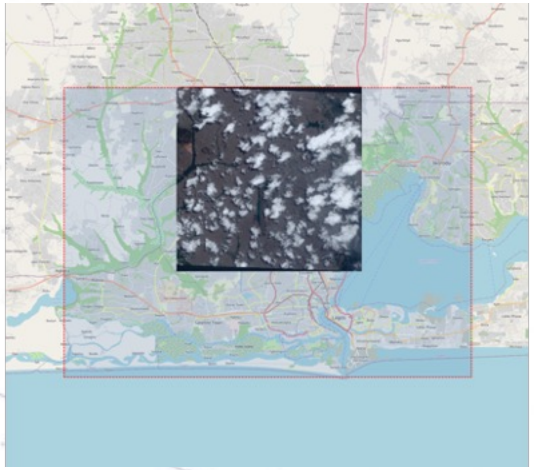
Resolution = r

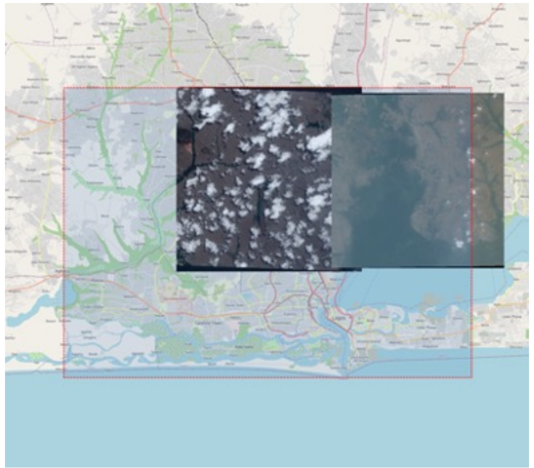
$r \times r$ cm^2



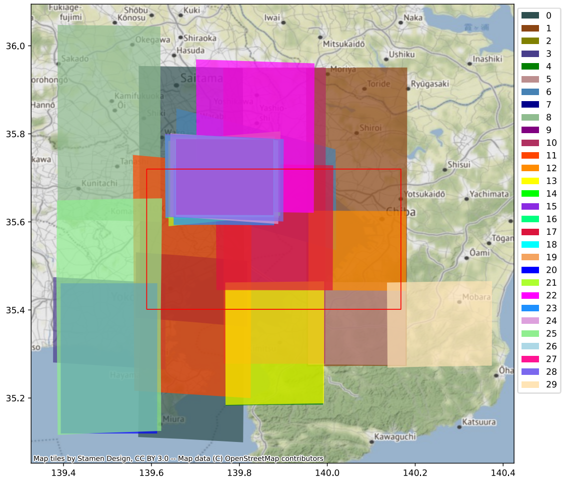
Cloud coverage

Cloud coverage does not have the same value throughout the image, like resolution or incidence angle. It is possible to reduce the cloud coverage by selecting images that overlap the cloudy areas of other images



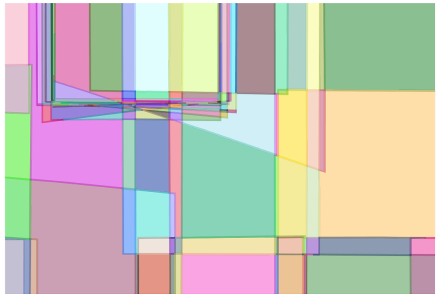


Preprocessing



Remove the area of images outside AOI

Find all intersections (parts) using the GEOS library[1]



The cover constraint and cost can be modeled as the classical ***weighted set cover*** problem

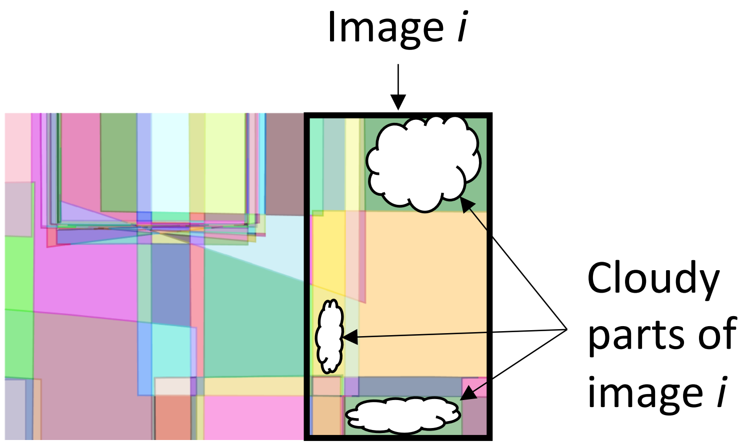
Universe = Union of intersections (parts)
Images -> Sets with parts and weight = cost

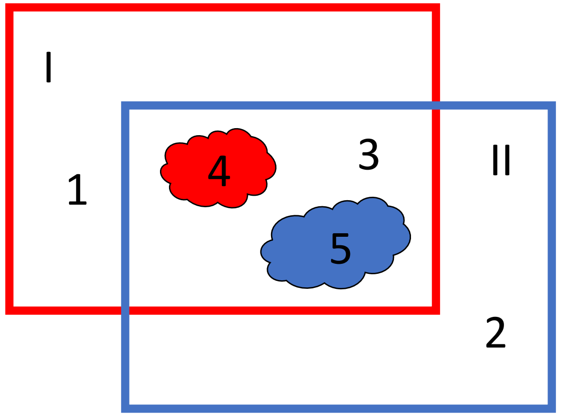
[1]GEOS coordinate transformation software library. Open Source Geospatial Foundation, 2021. URL: https://libgeos.org/.

Preprocessing – cloud detection

We use real high-resolution satellite image metadata and don’t perform cloud detection

- We assume that the portion of the satellite image inside the AOI has the same cloud percentange than the whole image.
- Randomly set some parts of the images as cloudy (totally covered by a cloud), until the cloud percetange is achieved.
- In real cases it could be that:
  - Most of the clouds in the image are outside of the AOI. This reduces the number of cloudy parts in the universe.
  - Clouds don’t fully cover the part. In this cases at least two parts are created one that represents the cloudy area and another one that represents the clear area.





In real case part 3 is divided in three parts

$$I = \{1,\underbrace{3,\bar{4},5}\}$$

$$II = \{2,\underbrace{3,4,\bar{5}}\}$$

$$I \cup II = \{1,2,3,4,5\}$$

CP Model

Cover constraint

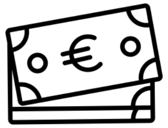
- Select a subset T of images P_i whose union is equal to the Universe (all the parts)

$$\bigcup_{i \in T} P_i = U$$

Optimize the following objectives:

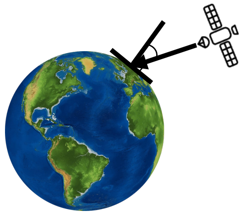
1. Sum of the cost of the images in T. (Equivalent to weighted set covering)

$$min \sum_{i \in T} W_i$$



2. Minimize maximal incidence angle (F) of the images

$$min\{ max\{ F_i \mid i \in T\}\}$$



CP Model

Optimize

3. Minimize the sum of the part's resolution. The resolution of a part *k* is equal to the min resolution of the images that contain it and are in the cover T.

$$min \sum_{k \in U} min \{ R_i \mid i \in T, k \in P_i \}$$

4. Minimize the cloudy area in cover T. For each part *k,* exists a set D_k with all the images that contain a non-cloudy view of *k*. A part *k* is non-cloudy in cover T if at least 1 image from D_k is in T.

$$V_k \Leftrightarrow \bigwedge_{i \in D_k} i \notin T$$

$$min \sum_{k \in U} V_k * A_k$$

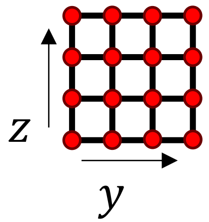
Exact Pareto front

CP approach, based on Gavanelli's work[1]

- Run satisfaction constraint solver iteratively
- Add new constraint representing the Pareto front. Suppose maximization, ex:
    - Solution sat -> $x = 10$ and $y = 5$
    - Add constraint ($x > 10 \vee y > 5$)
- Check if the new solution belongs to the front
- Stop when unsatisfiable (optimal Pareto front)

MILP approach, SAUGMENCON[2] (faster implementation of ϵ-constraint idea, skip unnecessary steps)

- Run optimization of one fixed objective and set the others as constraints. Suppose maximization, ex:
    - $\mathrm{Opt}(x)$ s.t. $y \geq 5, z \geq 10; y_{min} = 5, z_{min} = 10$
    - Add constraint $y \geq 5 + 1$
    - Repeat until unfeasible or $y \geq y_{max}$, then
    - Set $z \geq z_{previous\ value}$+1 and $y \geq y_{min}$. Ex:
        - $\mathrm{Opt}(x)$ s.t. $y \geq 5, z \geq 10 + 1$
- Stop when all possible values are explored, from y_{min}toy_{max}, and from nadir $z_{min}$to z_{max}.
- All the optimization results are points of the exact Pareto front

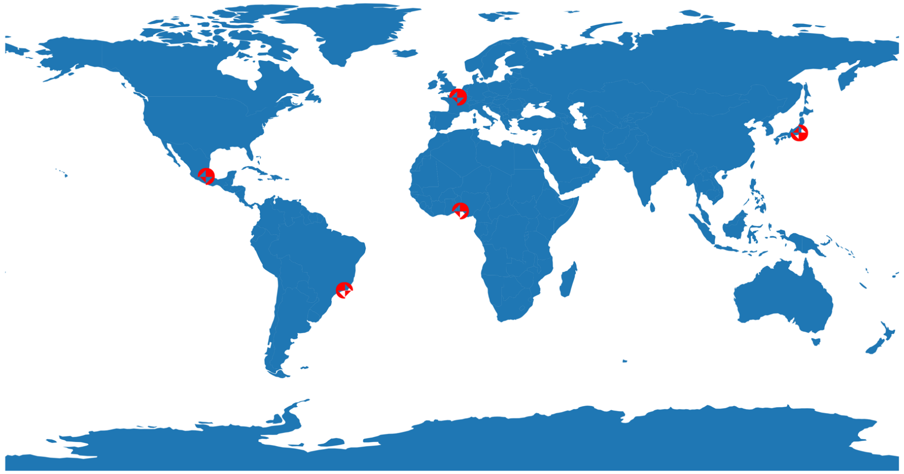


1. Marco Gavanelli. An algorithm for multi-criteria optimization in CSPs. In ECAI 2002: 15th European Conference on Artificial Intelligence, July 21-26, 2002, Lyon France: Including Prestigious Applications of Intelligent Systems (PAIS 2002): Proceedings, volume 77, page136. IOS Press, 2002
2. Weihua Zhang and Marc Reimann. A simple augmented ε-constraint method for multi-objective mathematical integer programming problems. European Journal of Operational Research,234(1):15–24, apr 2014. doi:10.1016/j.ejor.2013.09.001

Experimental setup

5 AOIs with 30, 50, 100, 150*, 200* images

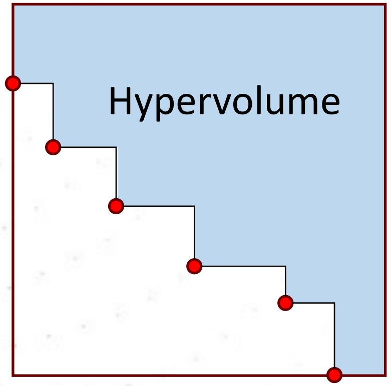
- Mexico City, Mexico
- Rio de Janeiro, Brazil
- Paris, France
- Lagos, Nigeria (max 145 images)
- Tokyo, Japan



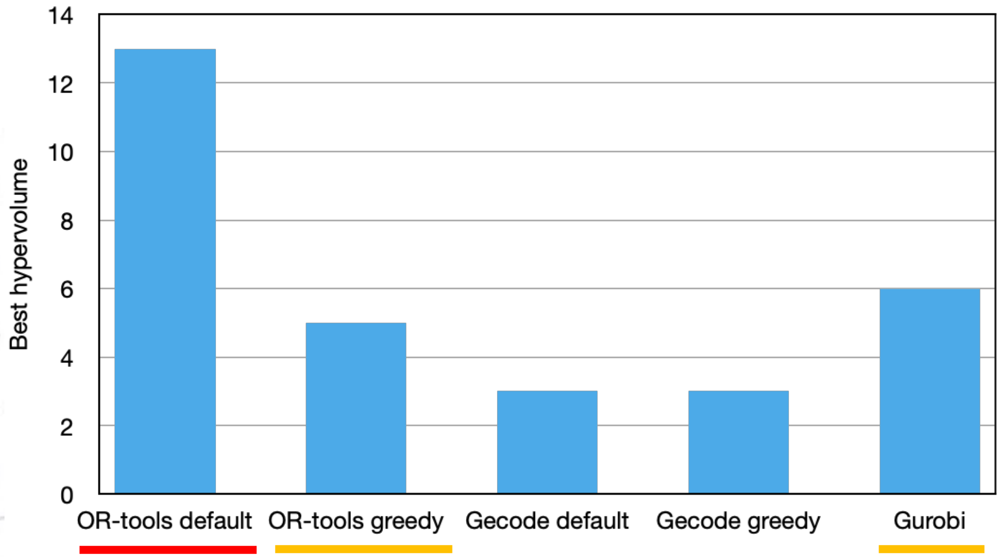
Running time: 1h

Solvers:

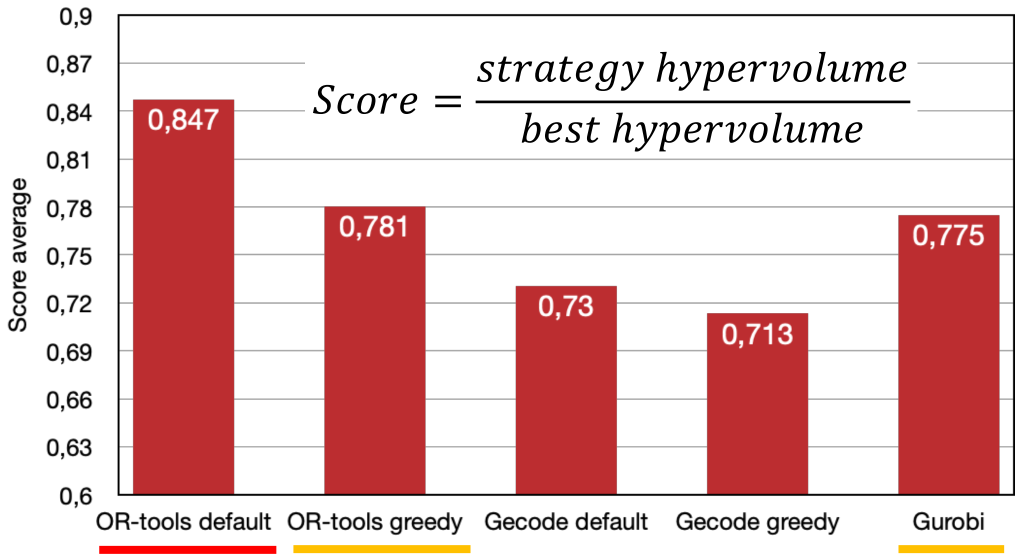
- CP
  - OR-tools
  - Gecode
- MILP
  - Gurobi



Experimental results



Number of times each approach had the best hypervolume



Average score for each strategy

Conclusions

- CP model, solved with OR-tools default search, obtained the best results.
- The proposed search strategy could not outperform the default search strategies of the constraint solvers, but for Gecode it produced similar results.
- The CP model produced better Pareto fronts than the MILP model (in the giving running time). This could be related to the method used to generate the Pareto front.
- For future research it will be interesting to:
  - Compare different approaches to generate the exact Pareto front for the CP and MILP models, based on the metric anytime behavior for the hypervolume.
  - Propose heuristics to tackle bigger instances and evaluate its performance against the proposed CP and MILP models.

Parallel Computing and Optimisation Group

Contact:

Manuel Combarro Simón
Doctoral researcher
manuel.combarrosimon@uni.lu

Pierre Talbot
Research associate
pierre.talbot@uni.lu

Connect with us

@SnT_uni_lu

SnT, Interdisciplinary Centre for Security, Reliability and Trust