ing. Gregory De Ruyter
gregory.deruyter@kuleuven.be
Supervisor prof. Hans Hallez
Co-supervisors prof. Bart Vanrumste
prof. Mathias Verbeke

Fonds Wetenschappelijk Onderzoek Vlaanderen
Opening new horizons
Project number: 1SH9Y24N

KU LEUVEN BRUGGE · DistriNet · m-group · LEUVEN.AI · SCIENCE → AI

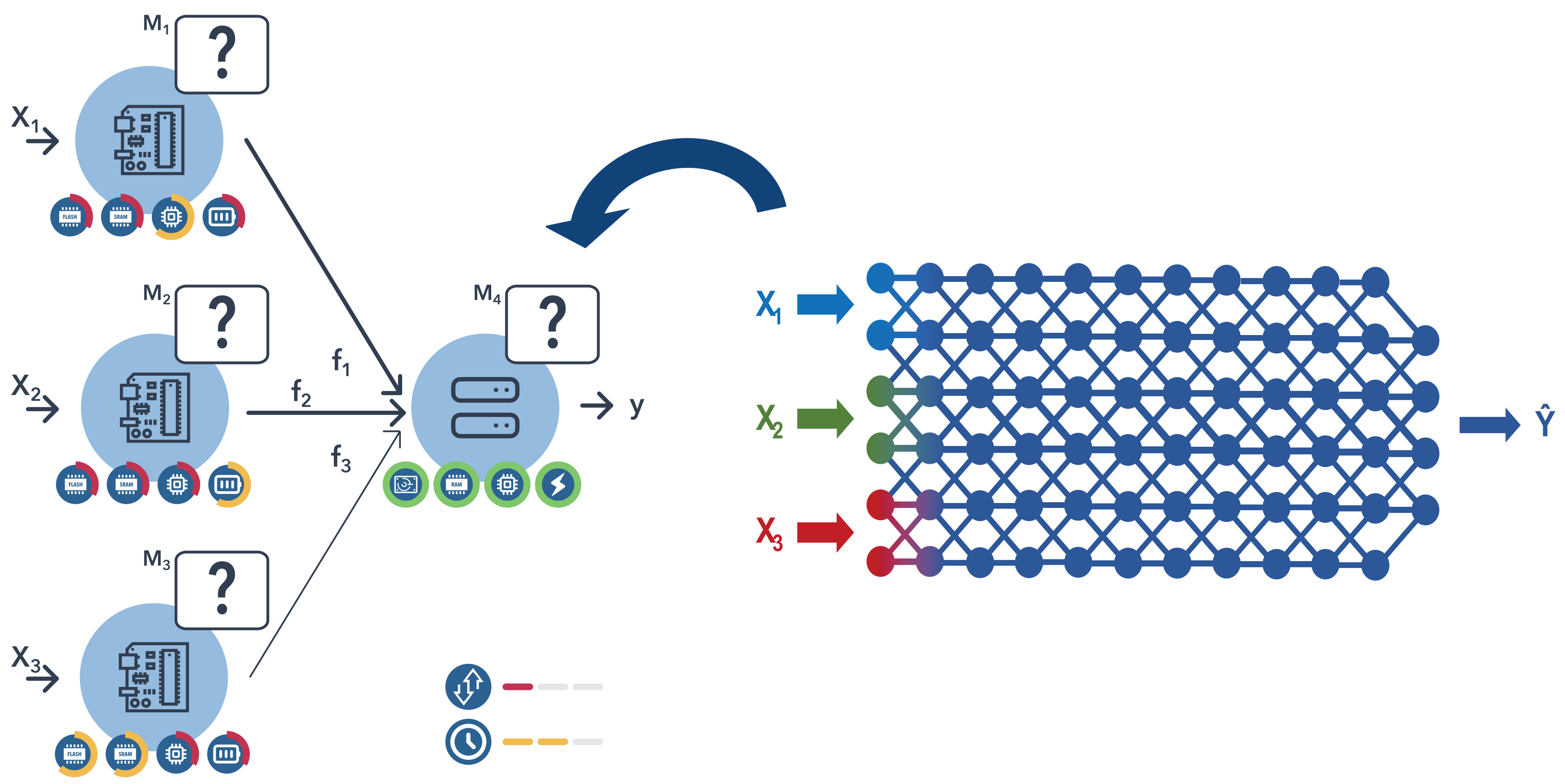
µDistriNAS: Multi-objective Neural Architecture Search for Distributed Neural Networks on Constrained Devices

Problem statement

Machine learning on the edge has several advantages over the traditional cloud-centric approach. **Edge computing** reduces the pressure on the network, decreases communication latency and takes away the possible privacy issues.

The distributed edge environment and constraints of individual devices, however, make the design and deployment of **machine learning** models, especially deep learning models, more **challenging**.

Our framework, µDistriNAS, addresses this challenge by automating the neural architecture design process while considering the constraints of the distributed edge devices and leveraging the collective computational resources available.



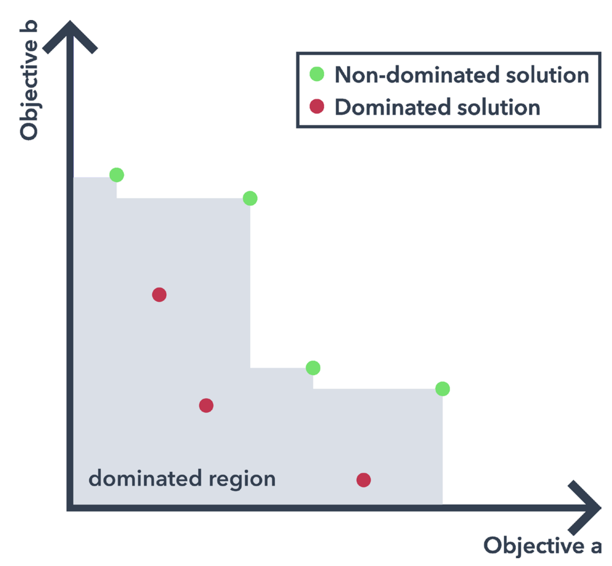
Domination criterion

A solution $N^{(1)}$ dominates another solution $N^{(2)}$ for objectives $f_1, \ldots, f_n$ if

$$\forall i \in 1,\ldots,n\colon f_i(N^{(1)}) \leq f_i(N^{(2)})$$

and

$$\exists j \in 1,\ldots,n\colon f_j(N^{(1)}) < f_j(N^{(2)})$$



Example:

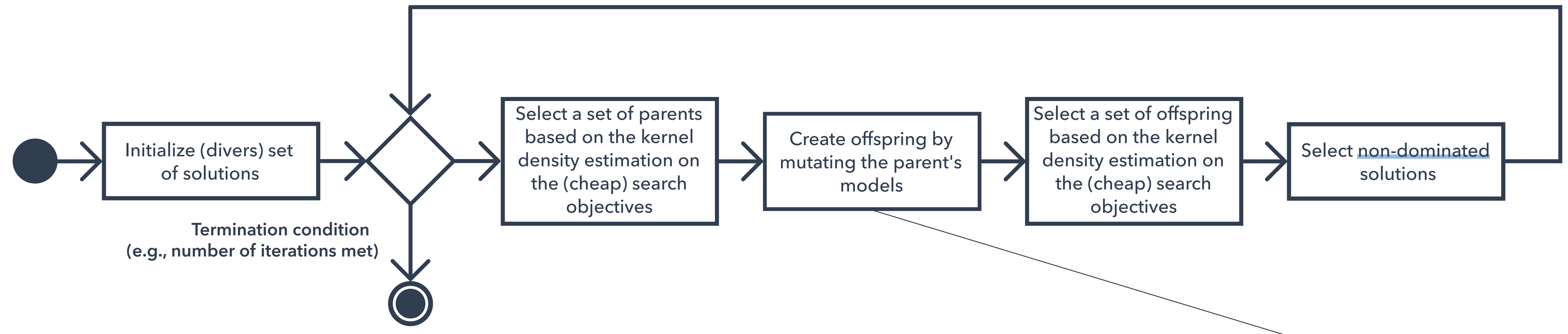
Given solutions $N^{(1)}$, $N^{(2)}$ & $N^{(3)}$ and objectives f_1, f_2 & f_3

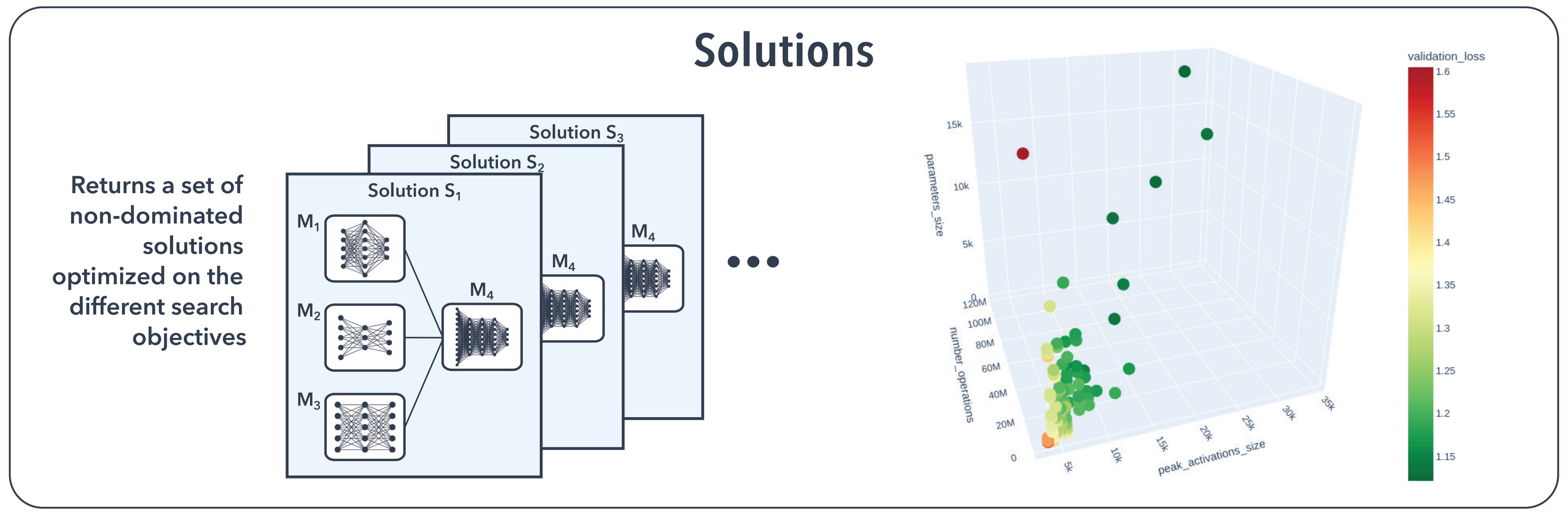
$f_1(N^{(1)}) = 1$ $\quad f_1(N^{(2)}) = 2$ $\quad f_1(N^{(3)}) = 3$ → $N^{(1)}$ is dominated by $N^{(2)}$

$f_2(N^{(1)}) = 4$ $\quad f_2(N^{(2)}) = 4$ $\quad f_2(N^{(3)}) = 1$ → $N^{(2)}$ and $N^{(3)}$ are both non-dominant

Methodology

µDistriNAS uses evolutionary search to explore the search space and keep a population of possible neural network architectures. In addition to the model performance, it uses **multiple search objectives** that are related to the constraints of the distributed edge environment to guide the search. Rather than combining these different objectives into one search objective, a diverse set of non-dominated solutions is maintained that ideally approximates the Pareto front.





Search Space

Convolutional block

time series	images
conv1d	conv2d
batch norm	batch norm
relu	relu
pool1d	pool2d

Mutations

- Insert conv block
- Remove conv block
- Enable / disable batch normalization
- Alter number of filters
- Alter kernel size
- Alter pooling factor of input or aggregation branch

Search objectives

Multiple search objectives are utilized to optimize with respect to the resource constraints for the individual edge devices, as well as the constraints from the edge network and the application itself.

Constraints

Per-device constraints
- FLASH memory
- SRAM memory
- Energy consumption

Network constraints
- Limited bandwidth

Application constraints
- Minimal model performance
- Minimal application latency
  = inference + communication latency

Search objectives

Per-device objectives (evaluated for each model partition):
- FLASH usage → model size
- SRAM usage → peak activations size
- Energy consumption → number of multiply-accumulate (MAC) operations
- Inference latency → number of multiply-accumulate (MAC) operations

General objectives (evaluated for the entire model):
- Communication latency → number of hops
- Bandwidth usage → size of exchanged deep features
- Model performance → validation loss on prediction task

Future work

Experimentation and ablation study

Further experimentation on multivariate time series and image datasets from literature

Exploration of other neural architectural elements

Include architectural elements from state-of-te-art and mobile neural nets (such as skip connections, different convolution types, ...) in the search space to achieve a higher model performance and a lower footprint.

Integration of extisting models

Explore how to integrate pre-trained models to optimize the execution time of the NAS.