



University
of Glasgow

Energy-Aware Optimal SFC Embedding in Emulated MEC for IoT

Theviyanthan K., P. Harvey

TDIS '26, Edinburgh, Scotland · 27th April 2026



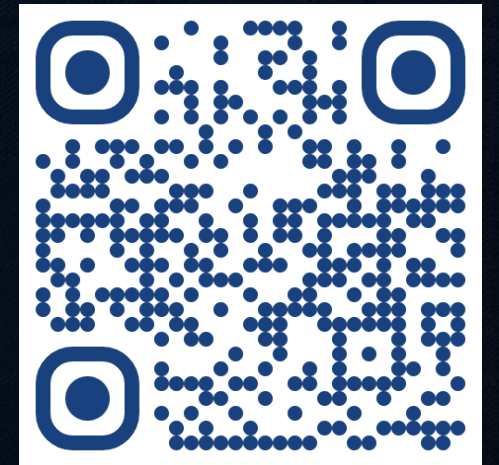
Theviyanthan (Thivi)
Krishnamohan

hello@thivi.dev

School of Computing Science,
University of Glasgow

Hello World! 🙌

I am a third-year Ph.D. student, researching
Genetic Algorithms for autonomous networks at
the University of Glasgow.



www.thivi.dev

Research Motivation

MEC for IoT



IoT data is processed close to the IoT devices at MEC, reducing latency and bandwidth demand in the cloud network.



MEC



IoT Devices



Energy Demand in MEC

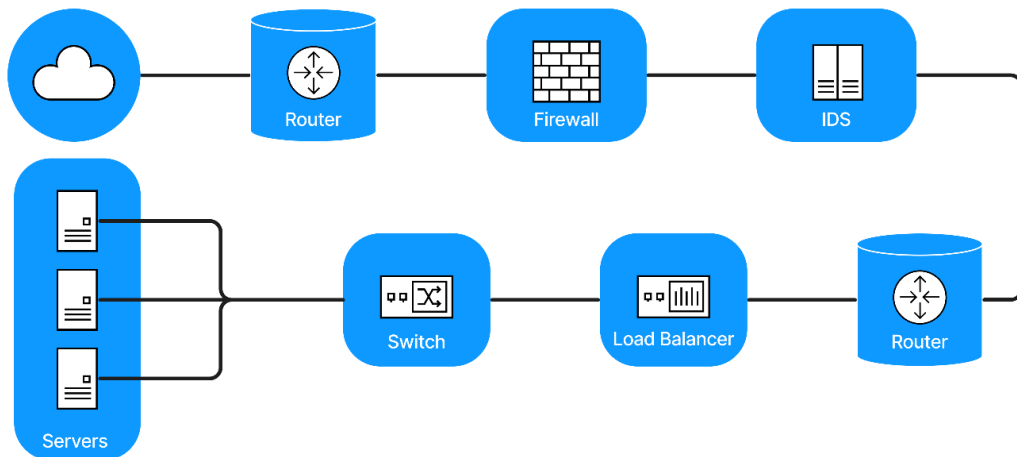
- Data processing at the edge may require more energy than in the cloud [1].
- The energy demand of MEC networks is rising due to
 - The increase in the number of servers deployed at the edge
 - The increase in the amount of data produced by IoT devices.
- This makes minimising the energy demand of MEC networks important.

[1] X. Sun and N. Ansari, "EdgeIoT: Mobile Edge Computing for the Internet of Things," in IEEE Communications Magazine, vol. 54, no. 12, pp. 22-29, December 2016, doi: 10.1109/MCOM.2016.1600492CM.

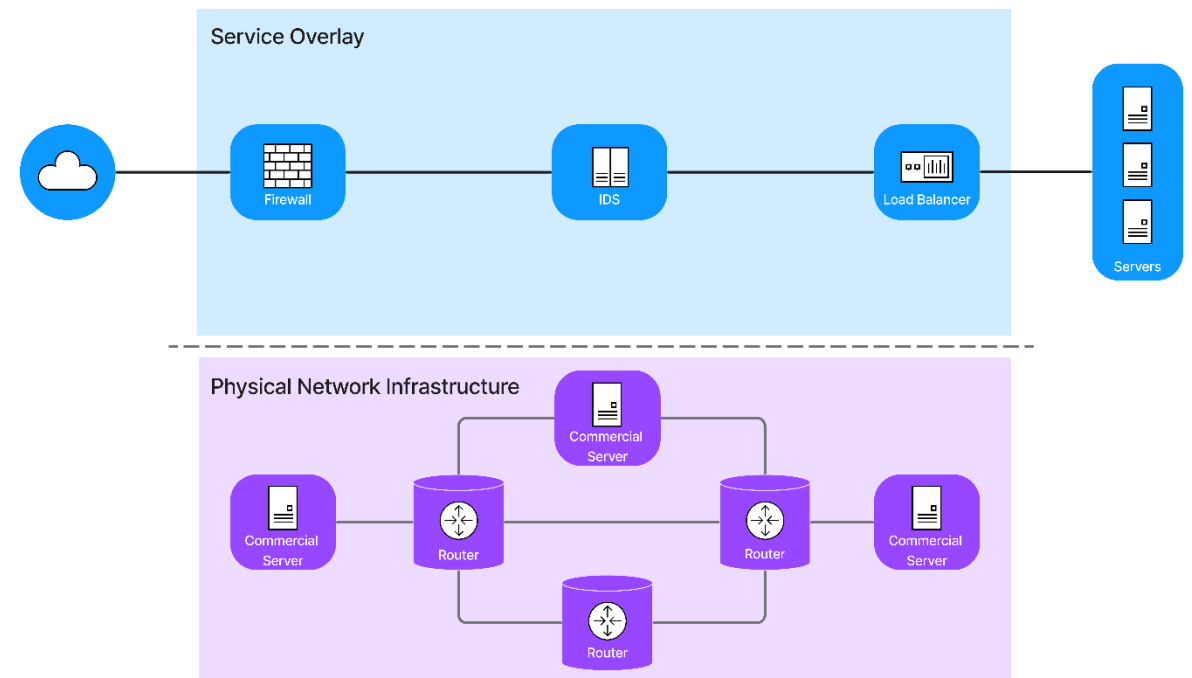
Service Function Chains (SFCs)

- SFCs combine Network Function Virtualisation and Software-Defined Networking to create a service overlay over the physical network.

A traditional network:



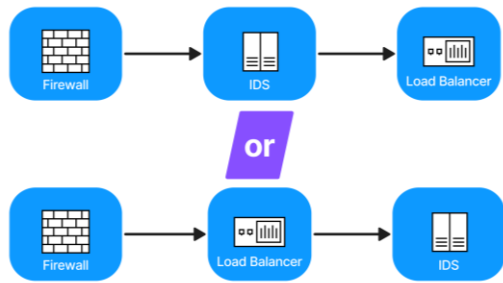
A Service Function Chain:



SFC Optimal Embedding

- Virtualisation enables SFCs to be programmatically embedded on the physical network.
- In turn, programmability enables optimal embedding of SFCs.
- We can develop optimisation approaches to embed SFCs such that the energy demand of MEC is minimised.

SFC Optimal Composition and Embedding (SFC-OCE)



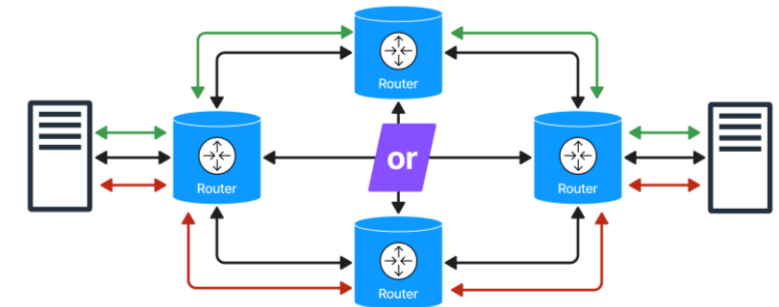
1. Chain composition

How should the Virtual Network Functions (VNFs) be ordered for optimal performance?



2. VNF embedding

Where should the VNFs be deployed for optimal performance?



3. Virtual Link embedding

How should the VNFs be linked for optimal performance?

- All three have to be optimised simultaneously to produce an optimal embedding, and this is an **NP-hard** optimisation problem [1].

[1] J. Gil Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," in IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 518-532, Sept. 2016, doi: 10.1109/TNSM.2016.2598420.

Research Gap

Sub-Optimal Embeddings

- We surveyed the literature from 2014 to 2025 to identify energy-aware SFC-OCE approaches.
- 17 studies optimised one or more of the SFC-OCE sub-problems.
- Only **one** optimised all three sub-problems.
- However, it used an exact algorithm.
 - Exact algorithms do not scale well when the number of SFCs to embed and the servers and switches in the MEC network increase.
- Most energy-aware optimisation approaches are designed to produce sub-optimal embeddings.

Low Fidelity Evaluation

- 13 studies performed numerical evaluation of their approach.
- 3 studies used simulation.
- 1 used a testbed.
- Numerical evaluation and simulation offer limited accuracy.

Our Approach

Genetic Algorithm

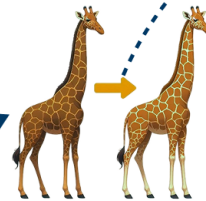
A population consists of solutions. Each solution is called an individual.



Population

Fitness of individuals is evaluated by a fitness function.

Individuals are subjected to random minor modifications.

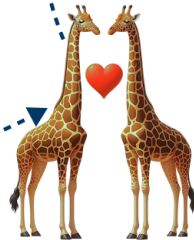


Mutation



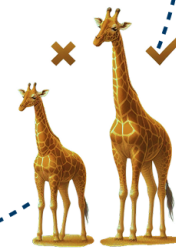
Evaluation

Two random individuals are recombined to produce new solutions.



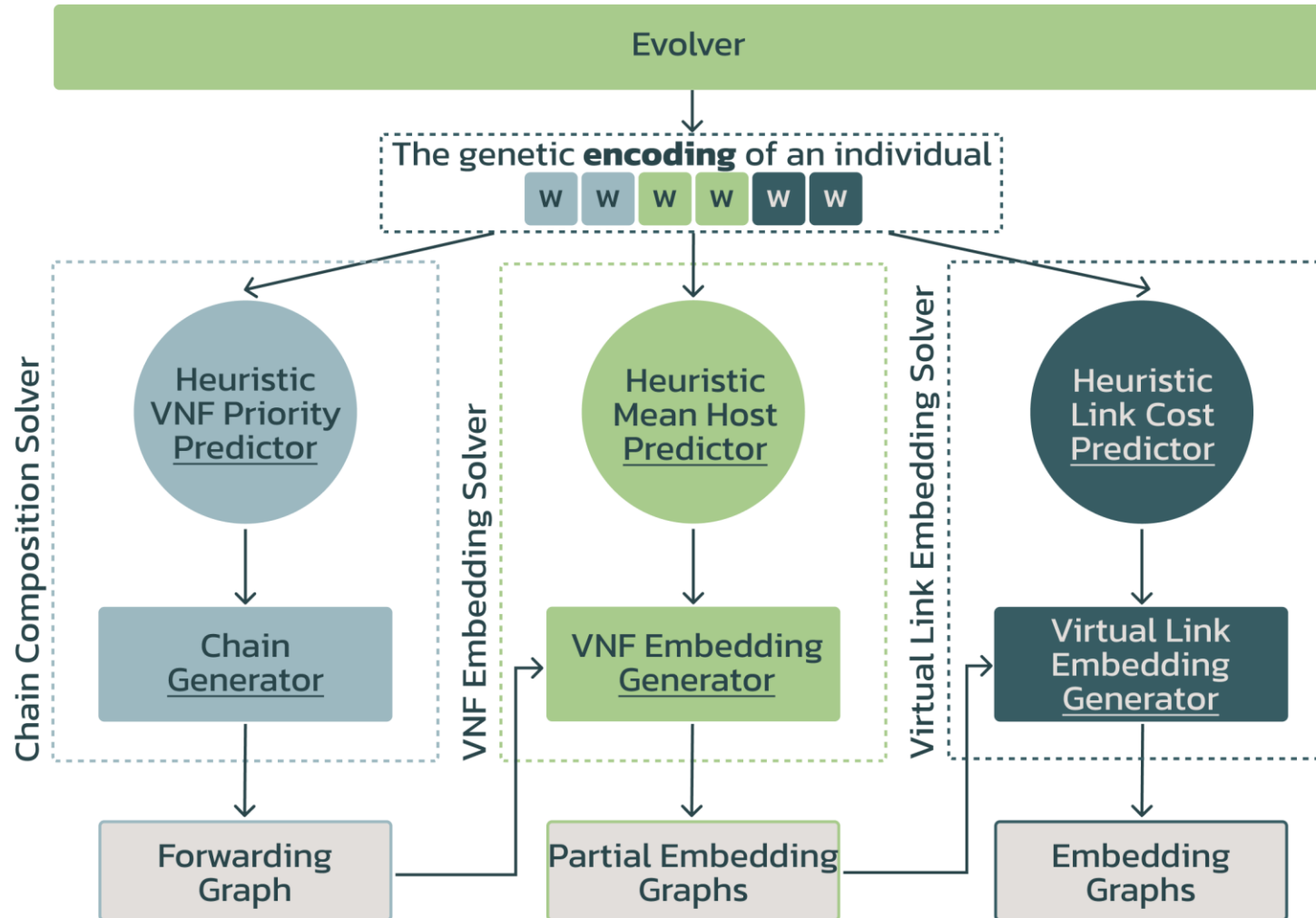
Crossover

The best individuals are selected for crossover.



Selection

Genetic Encoding using GENESIS



Optimisation Objectives

Max(Acceptance Ratio):

#Embedded SFCs/total SFC
Requests

Min(Energy Usage):

Total Consumed MEC
Energy

- Conflicting objectives => Pareto optimisation problem.
 - Embedding more network functions per server can increase its energy usage.
 - Embedding network functions across many servers can increase the overall energy usage of the MEC network as more servers are powered on.
- The optimisation approach has to strike a balance, minimising energy usage.

Power Model

- Based on a study on Google's data centre, we consider only the CPU usage of servers to compute the energy usage [1].
- Energy consumed by memory, storage and network equipment will be considered in future work.
- We use the linear CPU power model.

$$P(u) = p_{min} + (p_{max} - p_{min})u$$

[1] A. Radovanović et al., "Carbon-Aware Computing for Datacenters," in IEEE Transactions on Power Systems, vol. 38, no. 2, pp. 1270-1280, March 2023, doi: 10.1109/TPWRS.2022.3173250.

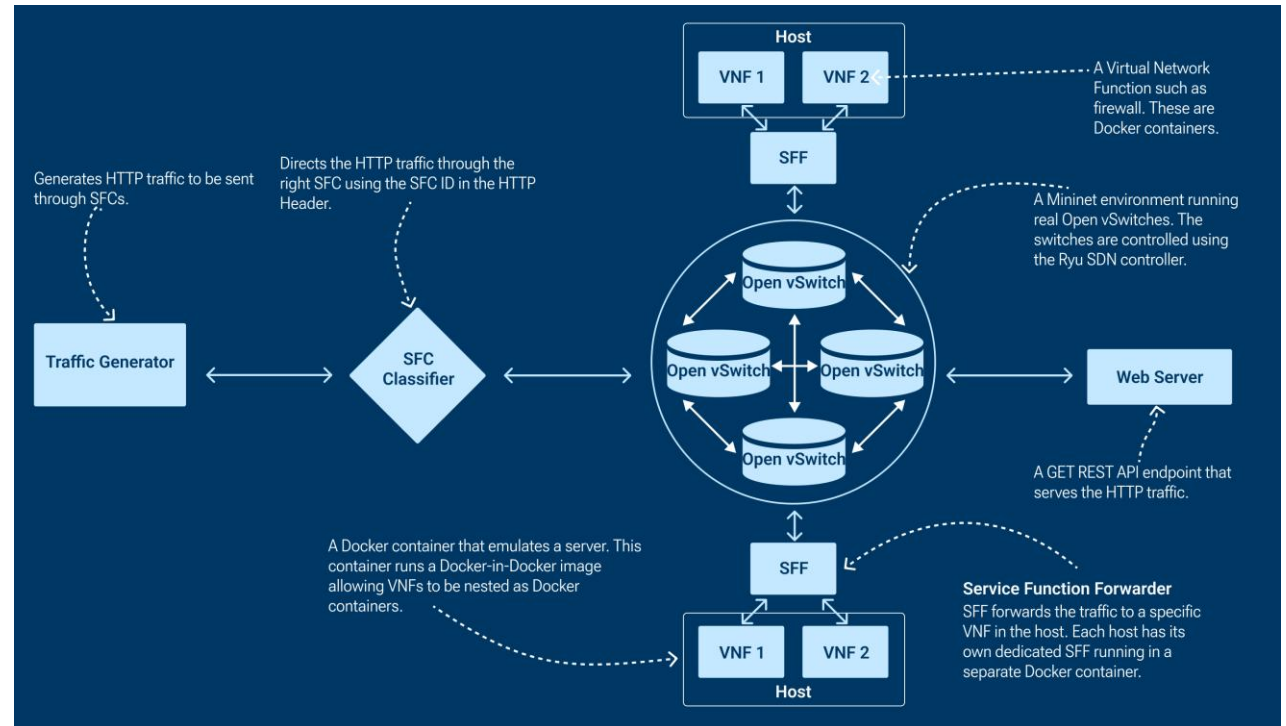
Fitness Evaluation

Online Evaluation	Offline Evaluation
Fitness is experimentally measured	Fitness is approximated by a model
✓ Evaluation accuracy is high	✗ Evaluation accuracy is low
✗ Slower evaluation	✓ Faster evaluation

- We use a **hybrid evaluation** approach, combining the speed of offline evaluation with the accuracy of online evaluation.

Online Evaluation

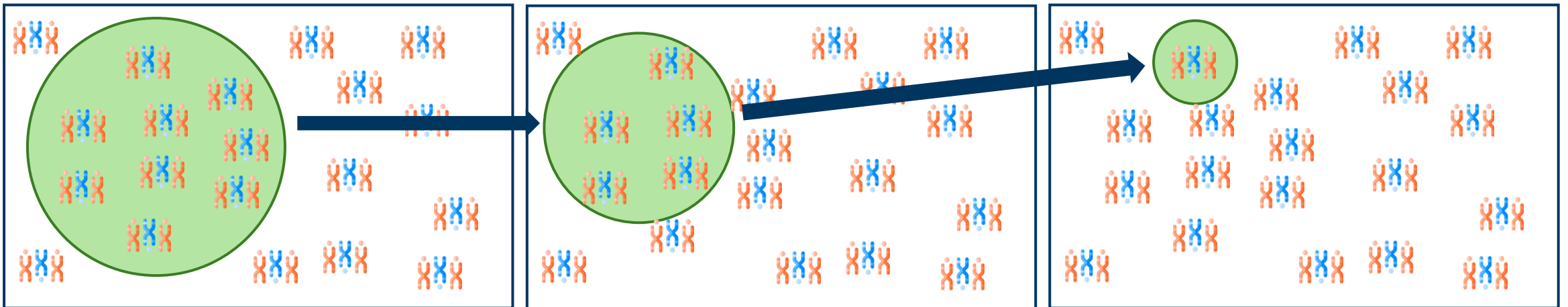
- We use **OpenRASE** [1], an SFC embedding emulator that we developed for online evaluation.
- OpenRASE uses Mininet and Docker containers to emulate a network environment.



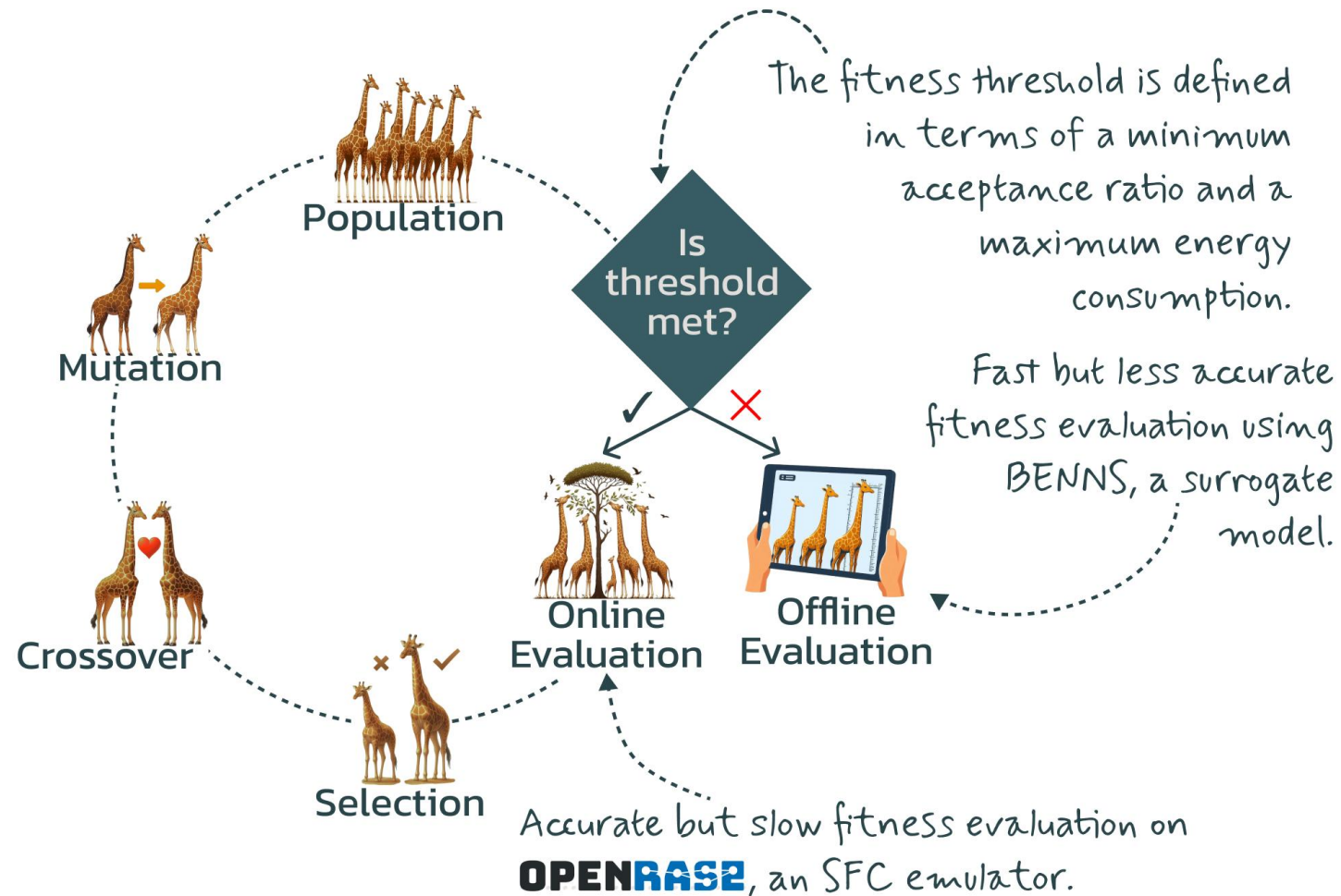
[1] T. Krishnamohan and P. Harvey, "OpenRASE: Service Function Chain Emulation," 2025 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2025, pp. 1-6, doi: 10.23919/SoftCOM66362.2025.11197454.

Offline Evaluation

- We use **BENNS**, a surrogate model, for offline evaluation.
- The goal of BENNS is to guide the GA toward the optimal region of the search space *as quickly as possible*.



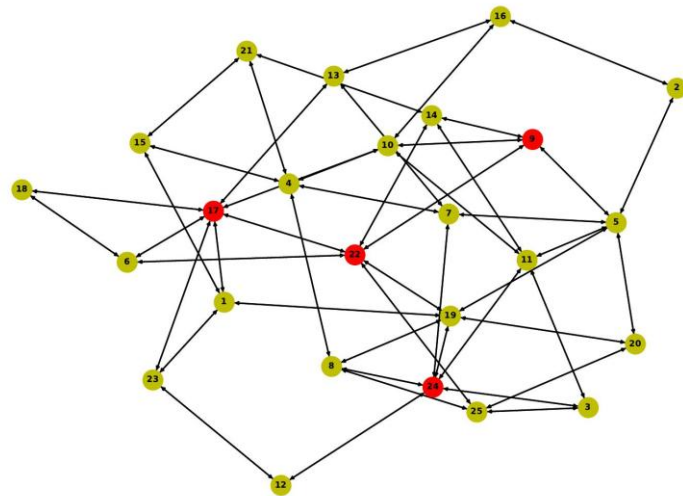
Hybrid Evolution



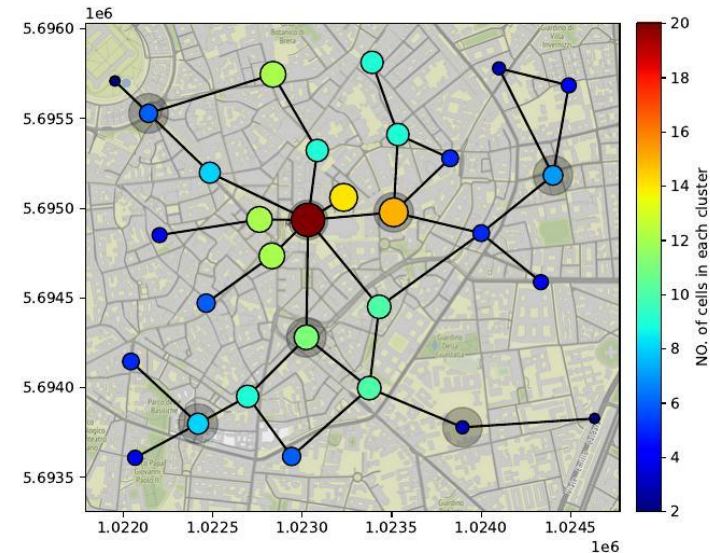
Evaluation

Experiment Design

- We ran two experiments using two MEC topologies.
 - A randomly generated 25N50E MEC topology
 - Milan City Centre MEC topology



25N50E



Milan City Centre

Source: <https://github.com/bnxng/Topo4MEC/tree/main>

Traffic Trace

- We used a real traffic trace from an IoT testbed.
- We scaled down an hour of the 24-hour traffic trace to 1 minute.
- So, an experiment took a minimum of 24 minutes.

Dynamic Network Environment

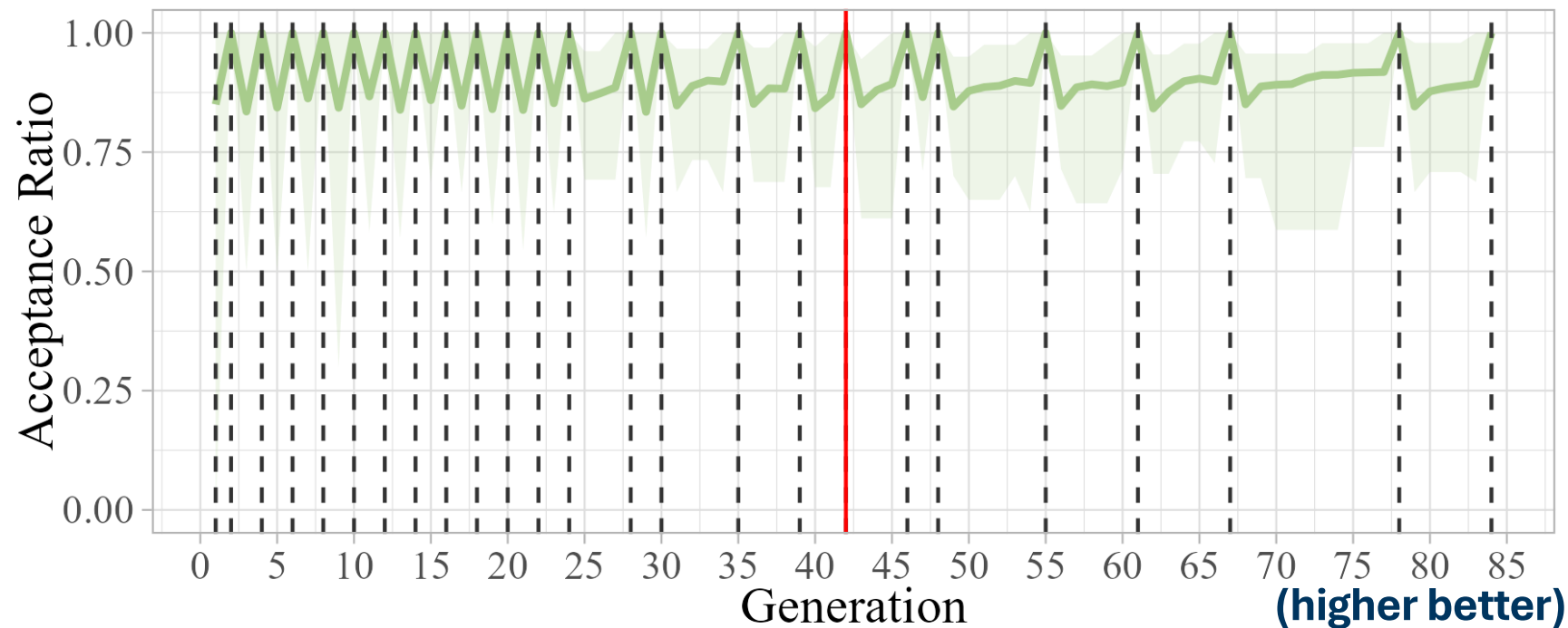
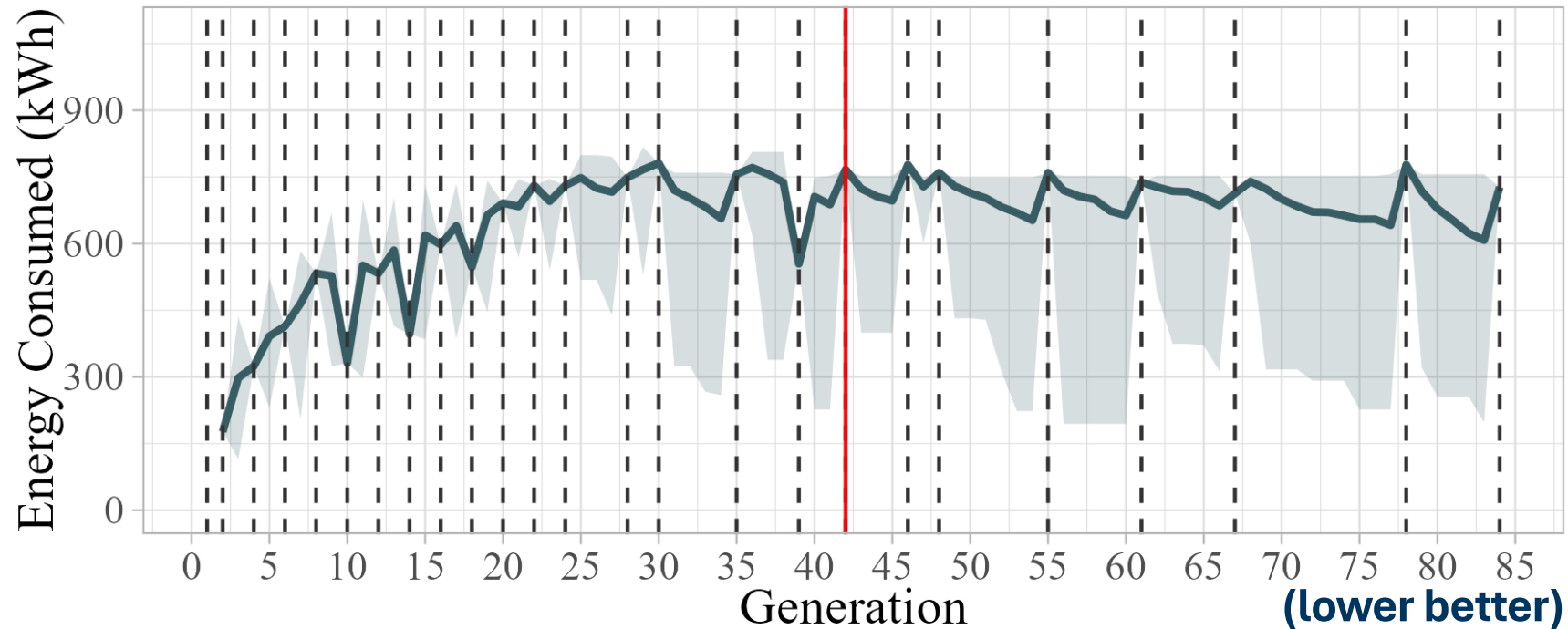
- Every 1 minute
 - 2 SFC Requests arrived
 - Traffic pattern changed
- From the 18th minute onwards, every 1 minute, a host crashed.

Fitness Threshold

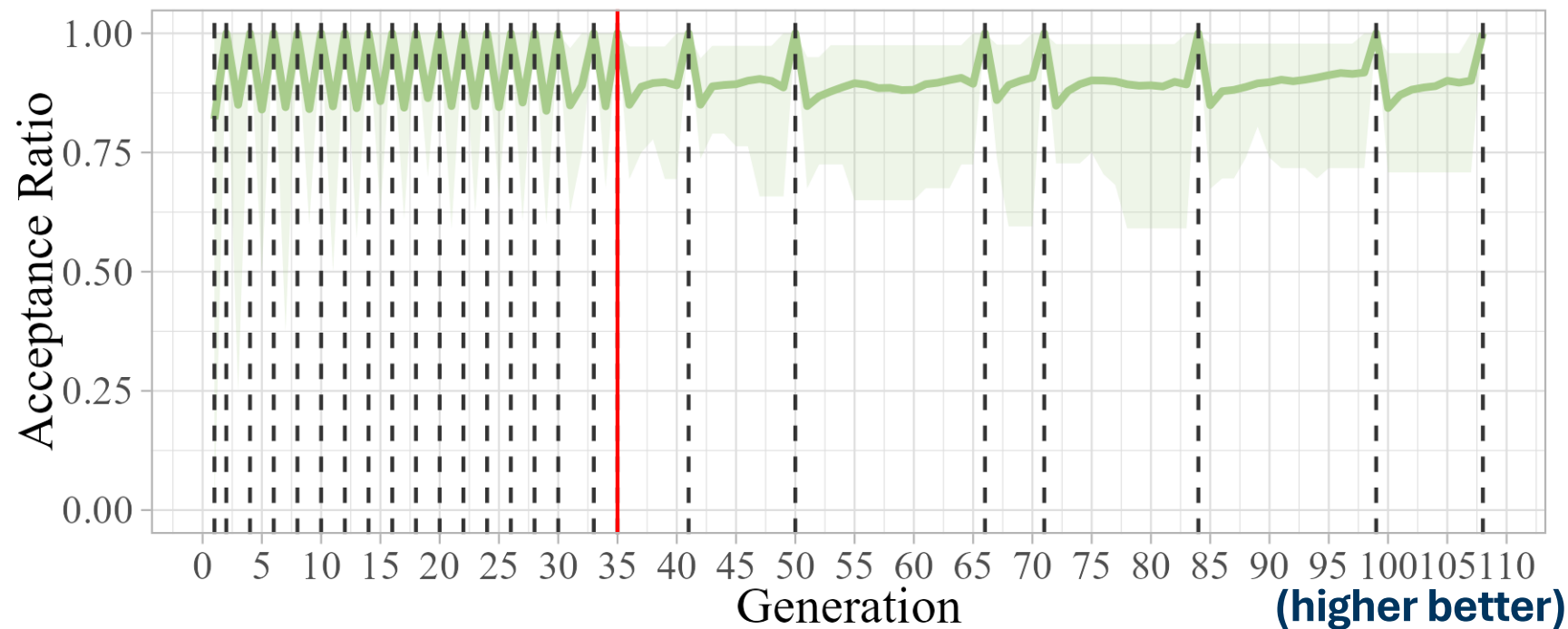
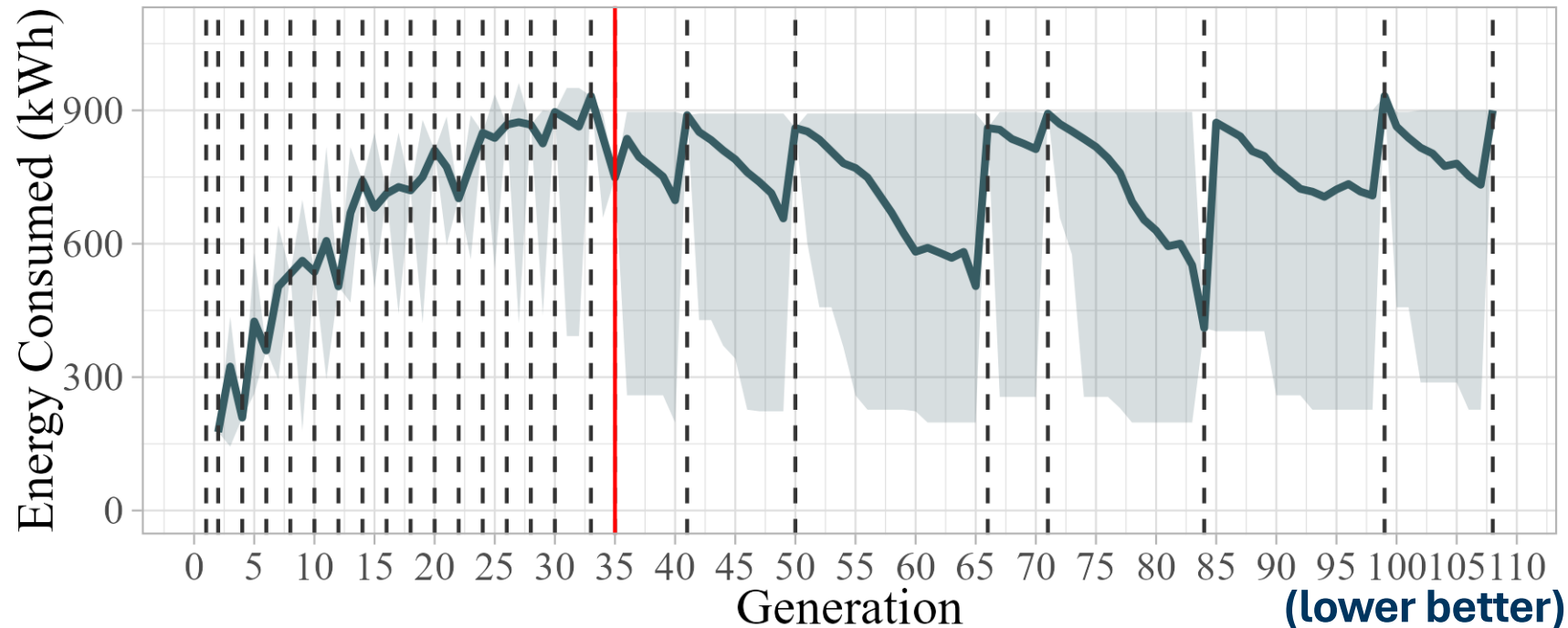


Results

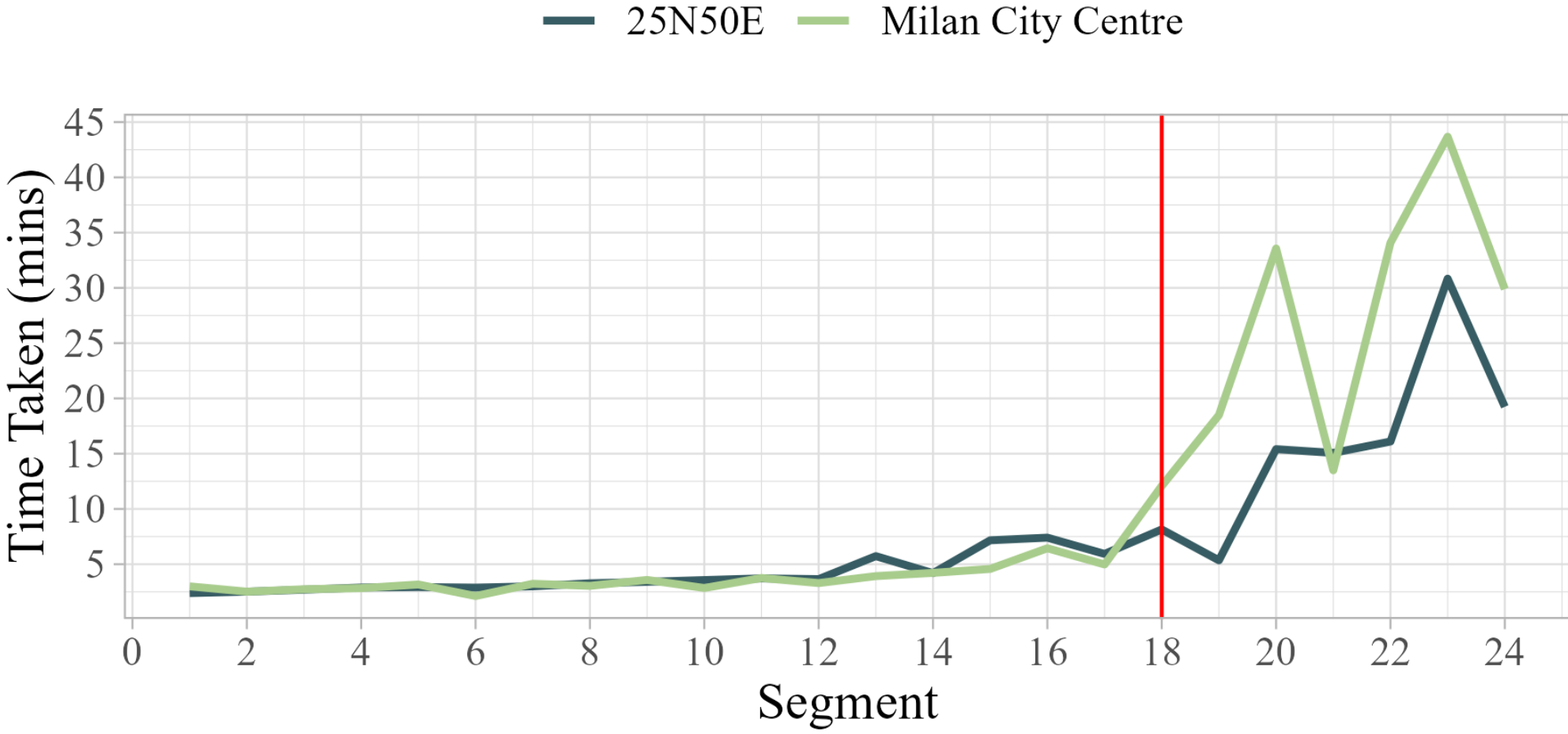
25N50E Experiment



Milan City Centre Experiment



Time Taken to Converge

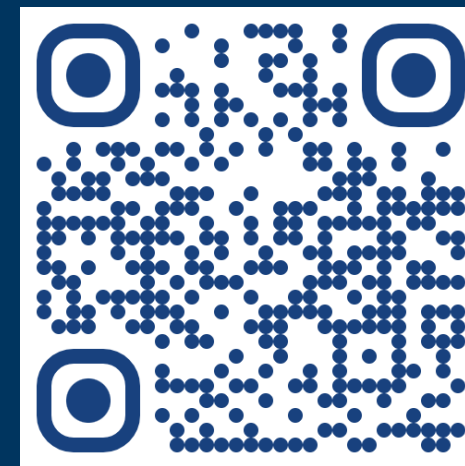


Conclusion

Conclusion

- Our hybrid evolution approach using GENESIS can embed SFCs in dynamic network environments, minimising energy consumption and maximising acceptance ratio.
- The approach converges on an optimal solution within a median of 3.84 minutes.
- It minimises energy consumption to a median of 730.8 kWh.

Questions



www.thivi.dev

Thank You