IEEE INTERNATIONAL WORKSHOP ON
Metrology for Living Environment

MetroLivEnv

IEEE Advancing Technology for Humanity

# Energy community management system based on real-time measurements and genetic algorithms

Massimiliano Proietti, Idea-Re Srl, Italy

Alberto Garinei, Idea-Re Srl, Italy

# Energy Community – what it is

Energy communities (ECs) are legal entities that allow citizens, small businesses and local authorities to produce, manage and consume their own energy.

ECs potential benefits:

- Increasing energy efficiency,
- Lowering electricity bills,
- Creating local job opportunities,
- Reducing carbon footprint,
- Social, economical, environmental sustainability,
- Green transition,
- Local energy independence.

# Energy Community – how it works

Components:

- Renewable Energy Systems (RES) e.g. solar, wind, hydro, geothermal…
- Energy Storage Systems (ESS) e.g. batteries
- Electric Vehicles Charging Points (EVCP)
- Controllable loads
- Uncontrollable loads

Energy producers (prosumers) and consumers of an EC share energy in order to minimize trades with external world.

Energy exchanges between EC users are virtual. Every user has her/his own energy provider

Economical benefits depend on the self-consuption of EC. Maximize local energy sharing to maximize self-consuption.

# Overview

## GOAL

- Study a real-time energy management system (EMS) for energy communities

## CHALLENGES

- ECs varying in size
- Several types of loads/energy generators/storage systems can be added to EC

## OUTPUT

- Mathematical model of EC self-consumption*
- EC simulator
- EC self-consumption optimizer (Genetic Algorithm)

*Self-consuption = min (energy generated by EC in an hour, energy used by EC in an hour) [ARERA]

# EMS - Model of Energy community

## Controllabe loads:

- Storage energy system (ESS)
- Electric Vehicle Charging Point (EVCP)

## Different types of EC users:

- simple consumer,
- consumer with storage and/or EVCP,
- prosumer (consumer that produces energy),
- prosumer with storage and/or EVCP.

# EMS - Model features and variables

## Variables (actions to be performed)

- Charging/discharging power of each storage at timestep t,
- Delivered power of each EVCP at timestep t.

## Features

- State of charge (SOC) of each storage at timestep t,
- Energy need of each user at timestep t,
- Production of each RES at timestep t,
- Minimum and maximum capacity of each storage,
- Maximum charging and discharging rate of each storage,
- Maximum deliverable power of each EVCP,
- Total energy delivered by each EVCP at timestep t.

# EMS – Optimization problem

Objective function that represents self-consuption

$$OBJ(t) = \left| RES_{TOT}^{t} - ECNET_{TOT}^{t} \right|$$

$$ECCONS_v^t + \sum_m P_{vm}^t \Delta t + \sum_l P_{EVCP,vl}^t$$

Renewable energy generated by the EC at timestep t

Algebraic sum at timestep t of EC loads, storage charging/discharging energy, EVCP delivered energy

Optimization problem for self-consuption maximization

$$\min \left| RES_{TOT}^{t} - ECNET_{TOT}^{t} \right|$$

# EMS – Genetic Algorithm

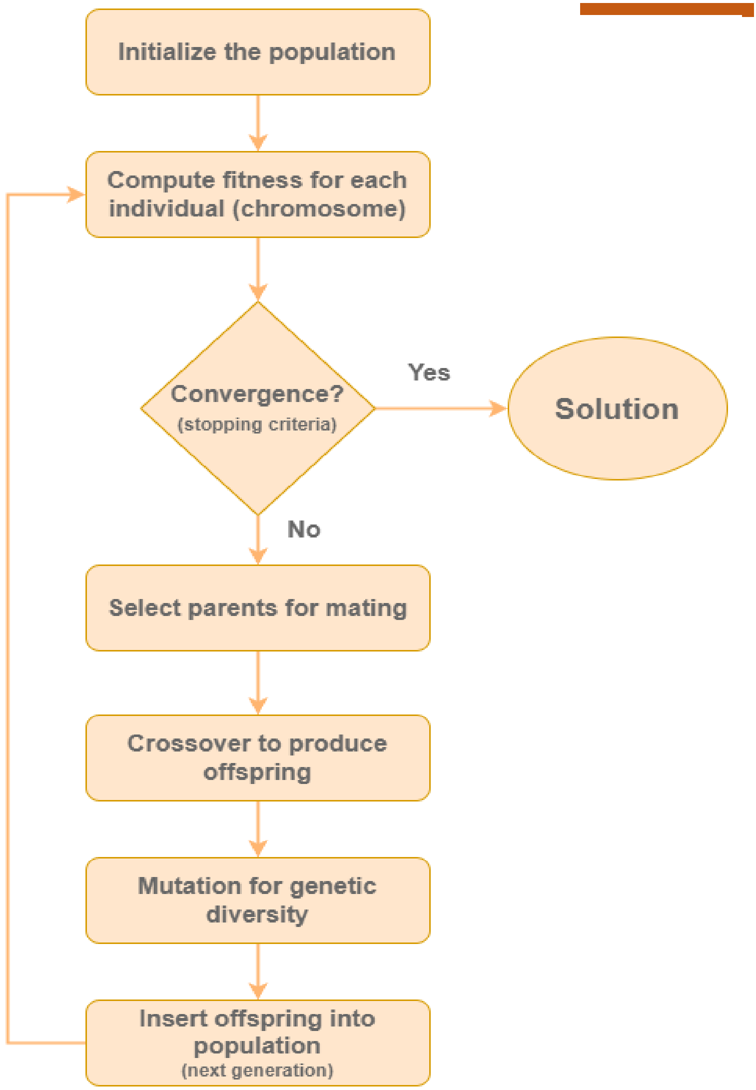
Genetic algorithm (GA) is a metaheuristic inspired by Darwin's natural selection

Why GA?

- GA can generate high-quality sub-optimal solutions in high constrained problems
- GA has the ability to search in the whole space of solutions

GA requires:

- Genetic representation of solutions domain (gene, chromosome, population)
- Fitness function to evaluate solutions
- Operators to represent selection, crossover (recombination), mutation



# EMS – GA implementation

Gene: decision variable. Gene space is discrete, discretization step as hyperparameter.

Chromosome: string containing all the decision variables.

Penalty functions to handle constrains:

$$min\ f(x) \quad s.t.\ \ c_a(x) \leq 0, \qquad a = 1, \dots, p$$

$$minF(x),$$

$$F(x) = f(x) + \sum_{a=1}^{p} \alpha_a max[0, c_a(x)]^{\beta_a}$$

$\alpha_a, \beta_a$ are tunable hyperparameters

Mapping from objective function to GA fitness:

$$Fitness \ = \frac{1}{OBJ^2 + \ \epsilon}$$

# EMS – Optimization workflow

EC simulator produces forcasted loads and productions for entire day.

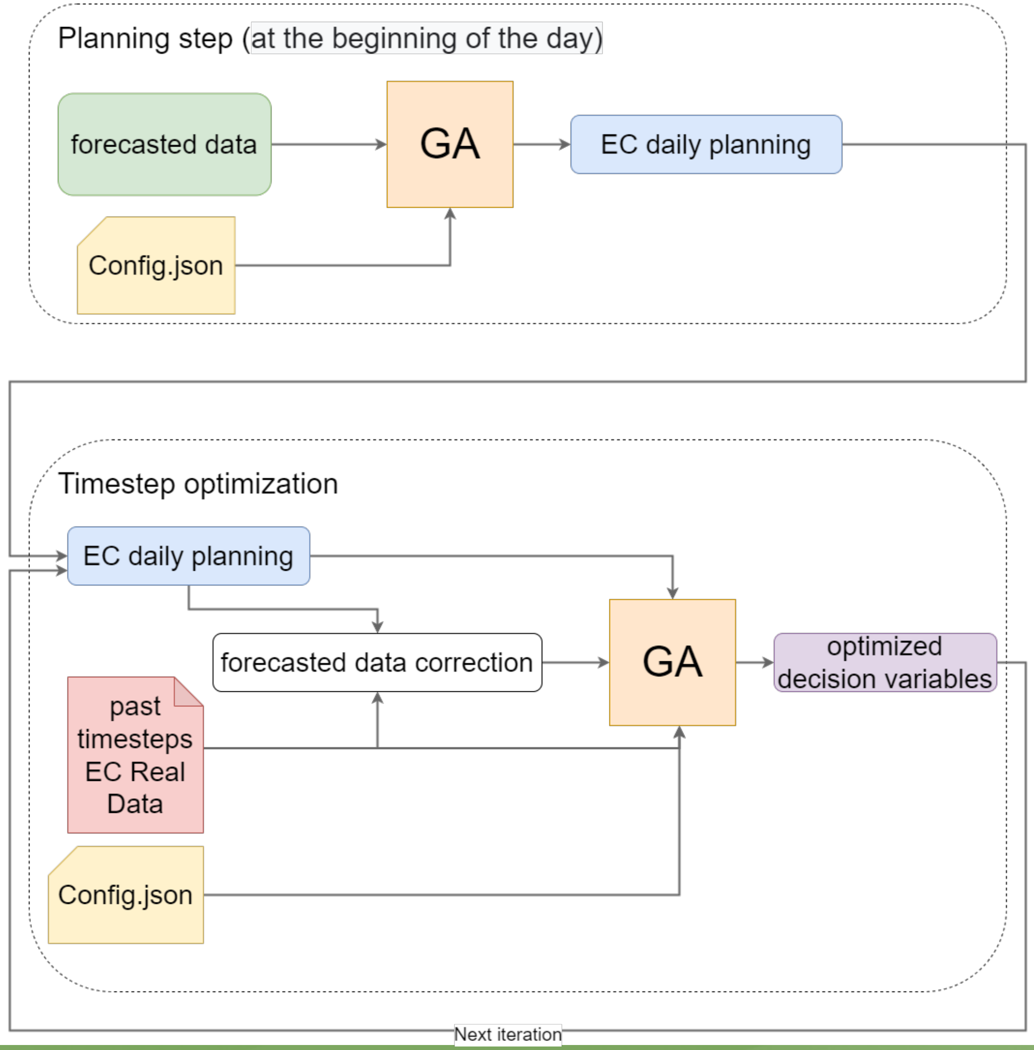
Config.json file contains information about storages and daily availability of EVCP.

GA performs an optimization over the entire day (EC daily planning).

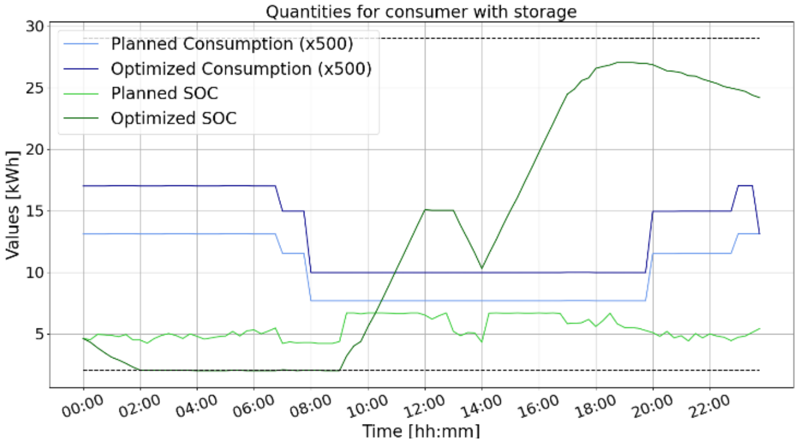
Timestep optimization is performed every 15 minutes.

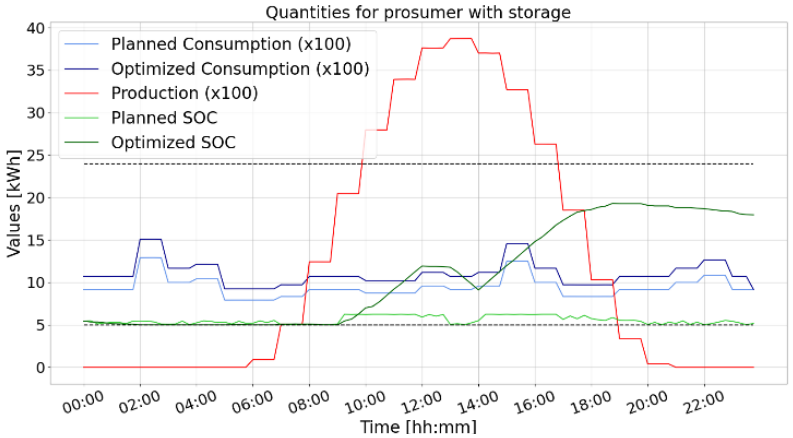
Before GA running, a correction of forecasted data is performed using real data from EC.

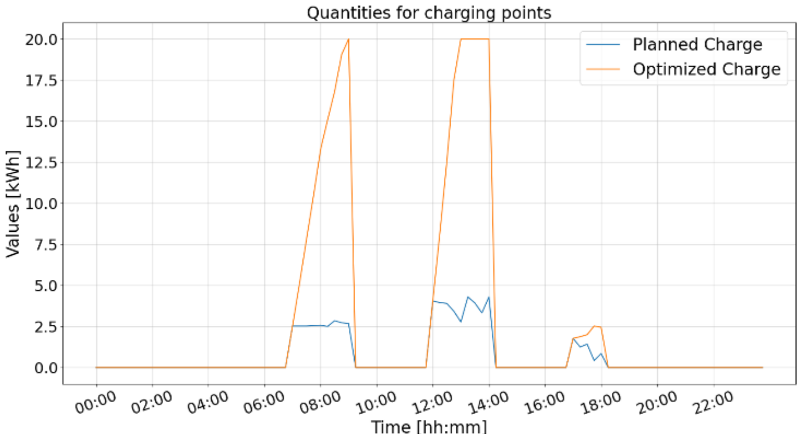
Optimized decision variables are used for optimization of next timestep.



# Results – EC case study



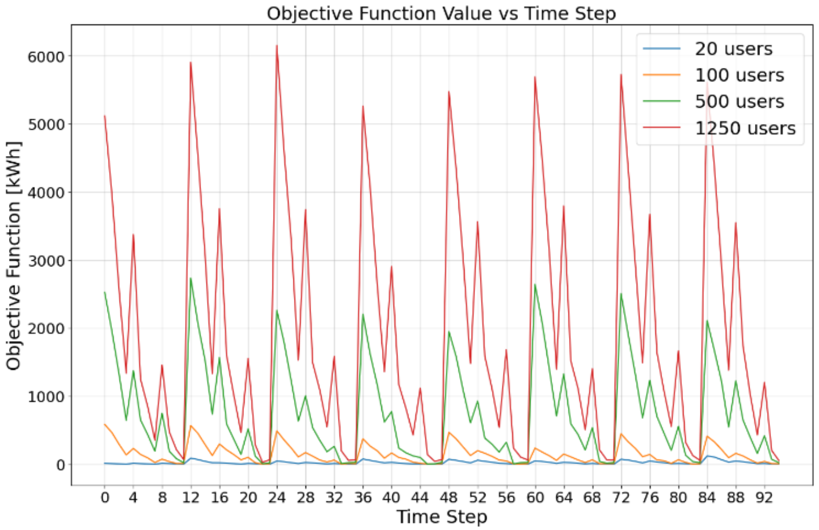

## EC features:

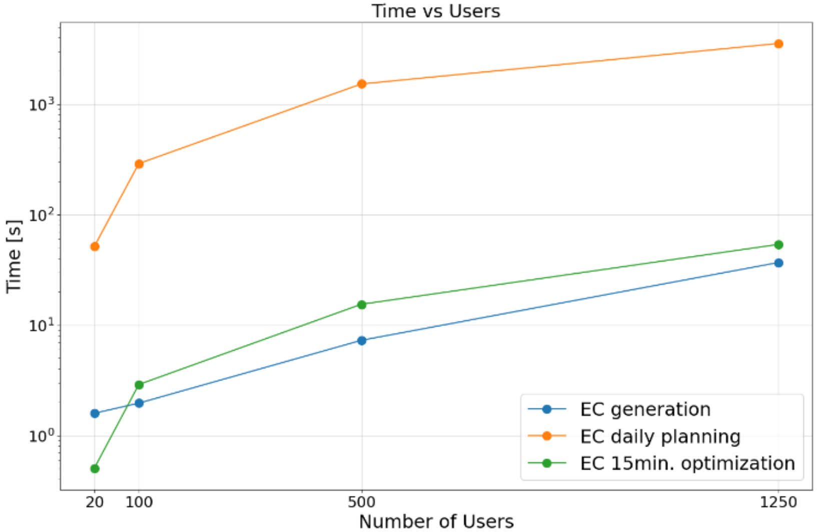
- 3 consumers without storage,
- 2 prosumers without storage,
- 2 consumers with storage,
- 2 EVCP (3 reservation slots).

## EVCP logic:

- If user requests a charge level, then the optimizer works to reach it (first two slots).
- If user doesn’t request a charge level, then the optimizer provide energy based on EC production availability (last slot).

# Results – Optimization algorithm scalability




Non-normalized objective function of Simulated ECs:

- 20 users
- 100 users
- 500 users
- 1250 users

Similar GA behaviour across the size of ECs.

Periodic pattern due to high weight of EVCP penalty function

Constraints violation (high value of Obj) due to high numbers of EVCP

Time needed:

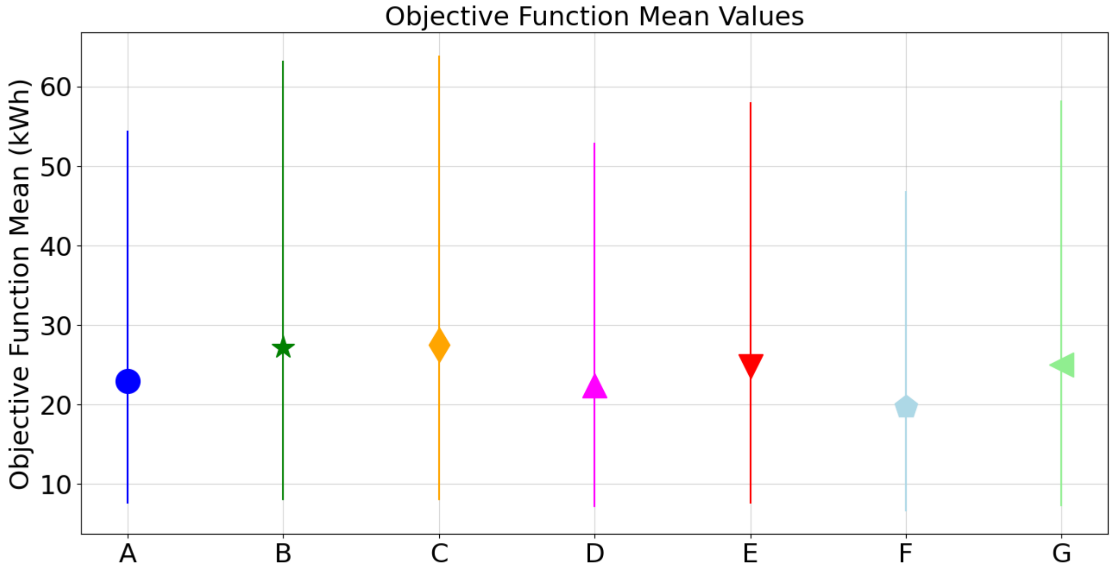
- to generate the EC, to make the daily planning, to perform the EC 15 minutes optimization

Computer specifications:

- Laptop AMD Ryzen 5 5600U with Radeon Graphics 2.30 GHz, RAM 16 GB, Windows 11 Pro OS

# Results – GA hyperparameters grid search

| | N° Gen. | N° Par. | Sol.Pop. | Step | Crossover | Mut. % | Plan [s] | Opt. 15' [s] |
|---|---|---|---|---|---|---|---|---|
| A | 100 | 5 | 10 | 0,01 | 2 points | 10 | 51,47 | 0,505 |
| B | **1000** | 5 | 10 | 0,01 | 2 points | 10 | 422,37 | 3,880 |
| C | 100 | **10** | 10 | 0,01 | 2 points | 10 | 54,61 | 0,620 |
| D | 100 | 5 | **100** | 0,01 | 2 points | 10 | 938,55 | 9,510 |
| E | 100 | 5 | 10 | **0,001** | 2 points | 10 | 68,82 | 0,772 |
| F | 100 | 5 | 10 | 0,01 | **1 points** | 10 | 52,56 | 0,573 |
| G | 100 | 5 | 10 | 0,01 | 2 points | **30** | 77,51 | 0,807 |




GA hyperparameters impact on operational time

EC features:

- 10 storages, 14 EVCP.

Metrics:

- daily mean of objective function for grid search evaluation.
- A value of the daily mean of objective function closer to zero can indicate more efficient optimization

# Future directions

## Drawbacks of GA

- black box optimization and cannot learn from data.
- same computational resources needed every timestep.
- Not viable in scaling number and size of managed ECs.

## Future directions: Reinforcement learning (RL)

- RL agent tries to learn good actions to do (policy) by maximizing the reward from the environment.
- Deep RL overcomes issues of traditional RL like high dimensionality and sparsity of action/state spaces.
- Model-free RL algorithms can be used when model dynamics (transition probabilities) is challenging.
- Deep RL can use demonstrations from human or artificial expert to speed-up the learning process.

# Conclusions

Energy Management System for Energy Communites that optimizes self-consuption

Developed Mathematical model allows to easily scale size and device types

Genetic Algorithm to find sub-optimal solutions because of its ability to search in the whole solutions space

Reinforcement Learning to enhance EMS performance in scaling scenarios

This paper is an output of the project “AIREC- Artificial Intelligence to improve Renewable Energy Communities” carried out by the partnership between Sistematica S.p.A. and Free Soft & Tech S.r.l.

# Thank you!