

This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Using a Digital Twin as the Objective Function for Evolutionary Algorithm Applications in Large Scale Industrial Processes

Eklund, Miro; Sierla, Seppo A.; Niemistö, Hannu; Korvola, Timo; Savolainen, Jouni; Karhela, Tommi A.

Published in:
IEEE Access

DOI:
[10.1109/ACCESS.2023.3254896](https://doi.org/10.1109/ACCESS.2023.3254896)

Published: 09/03/2023

Document Version
Final published version

Document License
CC BY

[Link to publication](#)

Please cite the original version:

Eklund, M., Sierla, S. A., Niemistö, H., Korvola, T., Savolainen, J., & Karhela, T. A. (2023). Using a Digital Twin as the Objective Function for Evolutionary Algorithm Applications in Large Scale Industrial Processes. *IEEE Access*, 11(10005208), 24185-24202. <https://doi.org/10.1109/ACCESS.2023.3254896>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Received 15 February 2023, accepted 6 March 2023, date of publication 9 March 2023, date of current version 14 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3254896

RESEARCH ARTICLE

Using a Digital Twin as the Objective Function for Evolutionary Algorithm Applications in Large Scale Industrial Processes

MIRO EKLUND^{1,2}, SEPO A. SIERLA³, (Senior Member, IEEE), HANNU NIEMISTÖ²,
TIMO KORVOLA⁴, JOUNI SAVOLAINEN², AND TOMMI A. KARHELA^{2,3}

¹Department of Information Technology, Åbo Akademi University, 20500 Turku, Finland

²Semantum Ltd., 02150 Espoo, Finland

³Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland

⁴VTT Technical Research Centre of Finland, 02150 Espoo, Finland

Corresponding author: Miro Eklund (miro.eklund@semantum.fi)

This work was supported in part by Fortum Ltd., in part by Semantum Ltd., in part by the Business Finland under Grant 7879/31/2018, and in part by the Academy of Finland under Grant 348415.

ABSTRACT In this paper, we describe how the up-to-date state of a digital twin, and its corresponding simulation model, can be used as a fitness function of an evolutionary algorithm for optimizing a large-scale industrial process. An ICT architecture is presented for solving the computational challenges that arise when the fitness function evaluation takes considerable amount of time. Parallel computation of the fitness function in a cloud computing environment is proposed and the evolutionary algorithm is connected to the computational environment using the Function-as-a-Service approach. A case-study was conducted on the district heating network of Espoo, the second largest city in Finland. The study shows that the architecture is suited for optimizing the operating costs of the large district heating network, with over 800 km of water pipes and over 14 heat producers, reaching a cost-saving of an average of 2%, and up-to 4%, over the current industrial state-of-the-art method in use at the city of Espoo.

INDEX TERMS Cloud computing, evolutionary computation, digital twin, optimization, simulation.

I. INTRODUCTION

Evolutionary algorithms have emerged as an effective heuristic for optimization problems in the process industry, see [1], [2], [3], [4], [5], [6]. These algorithms generate solution candidates, and the optimality of the candidates is evaluated against a fitness or objective function. The algorithm continues to produce candidates with a higher fitness function value, resulting in a near optimal solution after the algorithm has been executed enough iterations. The careful design of the fitness function is crucial for ensuring the real-world relevance of the optimization and to manage computational complexity. In the literature, authors usually hand craft the fitness function as a formula. In applications to industrial processes, this approach works best at a limited scope, such as a control loop or subprocess. However, there is a lack of

applications in large-scale industrial processes in which the optimization task cannot be isolated to subprocesses.

As an example of such a process, district heating is considered. District heating is used for heating buildings in an energy efficient way [7]. In a district heating network, one or more power plants generates heat that is transported using hot water pipes to buildings throughout the network. Fuel costs can be minimized by adjusting the outgoing water temperature of those plants [8]. Slow thermohydraulic phenomenon occurs due to flow-delays in the hundreds of kilometres of pipes [9]. Optimizing for such a complex state is difficult, but a dynamic process simulator with a model of the district heating network can be used for this purpose [10]. A dynamic simulator captures automation systems and environmental conditions and can determine the time-dependent impacts of transients such as valve closures or weather changes [10]. When optimizing a district heating network by changing setpoints, faster changes will increase the complexity of the

The associate editor coordinating the review of this manuscript and approving it for publication was Atif Iqbal.

optimization problem without significant benefits to the operating plan of the network, due to the slow thermohydraulic phenomenon. Additionally, since the heat propagation in the piping network takes several hours, a long time window is needed to evaluate the goodness of a candidate solution, in order to avoid short-sighted optimization of immediate benefits at the expense of longer-term costs [9]. The length of the time window should be sufficient to capture heat propagation effects throughout the network and longer-term impact of charging or using the thermal energy storage. However, for reasons of computational complexity, a shorter time window is preferable.

A district heating simulation captures the network in an ideal condition. In practice, a large network continuously experiences minor outages of pipelines or plants. The piping has redundancies that permit the automation system to exploit alternative routes to ensure the smooth operation of the network despite these outages. Unfortunately, this undermines the real-world relevance of the simulation-based approach. In recent years, the digital twin has emerged as a technique for maintaining up-to-date virtual replicas of physical systems, to be used for various monitoring and optimization applications deployed in cyber-space, see [11], [12], [13], and [14]. In this paper, the term digital twin is defined as a model that represents a real physical system, which is updated 24/7 using real measurement data from the real physical system. We investigate the use of a digital twin of an industrial process to implement the fitness function of an evolutionary optimization algorithm. The following research questions are considered:

1. How can we use simulation and optimization algorithms to minimize the heating energy production cost of a district heating network with several production plants?
2. What kind of ICT architecture can support the use of a digital twin as the fitness function of an evolutionary algorithm that optimizes a large-scale industrial process?
3. What are the trade-offs between computational complexity and fidelity, and how can the computational challenges be mitigated with parallel computing and cloud computing?
4. How does the decision to use an evolutionary computation impact the architecture and computational challenges?
5. How does the proposed architecture and approach for handling the trade-offs perform in a case study? In this article, we investigated one specific type of evolutionary algorithm in a case study of the entire district heating system of Espoo, the second largest city in Finland.

II. RELATED WORK

A. DISTRICT HEATING NETWORKS

District heating networks are examples of large-scale industrial processes, which from an optimization task's point-of-view cannot be isolated to subprocesses. District heating is the established solution for heating buildings in Northern Europe, see [15], [16], and due to its energy efficiency benefits, see [7], it has recently been applied in a number of other countries, see [17], [18], and [19]. A power plant generates

heat, which is transmitted to end users in an urban district through a network of hot water pipelines. Heat-exchangers at the end user buildings extract the energy required for space heating and service water heating, and the cooled water returns to the power plant through separate return pipelines. The network can span the geographical area of a large city, and it may contain several heat-generating producers, some of which may be CHP (combined heat and power) plants, HOB (heat only boilers) plants, heat-pumps or waste heat facilities. The optimization target is to adjust the temperature of the outgoing water at each plant, so that the fuel costs are minimized, see [8]. The optimization is greatly complicated by the thermohydraulic delays in the network, since any change of temperature at a plant will take several hours to propagate to end users, see [20], depending on each user's distance from the plant. The demand for space and service water heating can be forecast, but the forecasts involve uncertainty. The price of fuel and electricity varies by the hour, and a price forecast is available to the optimizer. The district heating network permits broad variations in the temperature of the district heating water, so significant potential for fuel cost optimization is possible through proactively raising the temperature and using the hundreds of kilometres of district heating pipeline as a thermal energy storage.

In order to apply an evolutionary algorithm, the design of the fitness function becomes exceedingly difficult, due to slow thermohydraulic phenomenon and very large state space of the city-wide district heating network, see [9]. A simulator that captures this phenomenon at a sufficient level of detail can be used to implement the fitness function. As input, it receives a candidate solution that specifies the outgoing temperature at the power plants. The simulator includes the automation system of the network and the power plants, so the mass flow in the outgoing pipeline from the power plant will be automatically adjusted in simulation to meet the heating need at the end users. The automation system at the simulator will adjust the fuel burn accordingly, so the simulator will be able to output the fuel cost for the time interval to be optimized. This cost is the fitness value of the above-mentioned candidate solution. By adjusting the fidelity of the simulation model, a trade-off can be made between computational complexity and real-world relevance. In order to cope with the resulting computational complexity, parallel computing techniques can be used to exploit high performance computing resources in the cloud.

B. EVOLUTIONARY ALGORITHMS IN INDUSTRIAL PROCESSES

Evolutionary algorithms are optimization algorithms that are inspired by biology, where a population evolves through various operations from generation to generation in an attempt to find an optimal solution to the problem presented [21]. Operations that modify the population include mutations, recombination and selection. An "individual" in the population is referred to as a candidate solution and how well it solves the problem is evaluated using a fitness function [22].

Evolutionary algorithms are well-suited to solving various optimization problems, including black-box problems where the details of the problem are not known by the algorithm. Covariance matrix adaptation evolution strategy (CMA-ES) [23] is an optimization strategy for solving numeric optimization problems [24] and is classified as an evolutionary algorithm. In this sub-section we review successful applications of CMA-ES in industrial process optimization problems.

The scope of this paper does not cover general use-cases for evolutionary algorithms, nor the details of various evolutionary algorithms. For a comprehensive literary review on those subjects, see [25]. The problem dimension of the optimization problem presented in this paper is limited to 32 decision variables, see [26] for a comparison of variants of the CMA-ES algorithm, which work well with larger problem dimensions.

Evolutionary algorithms have been used successfully in various process industry optimization problems, see [27], [28], [29], and [30].

In this paper, we utilize CMA-ES for solving an optimization problem of the district heating network of Espoo, the second largest city of Finland. Generally speaking, in evolutionary algorithms, there is a concept of a population size. In CMA-ES, however, this is replaced by a normal distribution and each iteration \mathbf{Y} candidate solutions are sampled from this normal distribution. Then, the algorithm performs \mathbf{X} iterations, also known as generations, where each iteration attempts to find a better candidate solution to the problem than the previous iteration. Specific for CMA-ES, however, the candidate solutions are created from a multivariate normal distribution and it is this distribution that is modified each iteration. The distribution acts as the mutation operation in other evolutionary algorithms and the size and direction of the distribution vary from generation to generation.

CMA-ES have been used successfully in many process- and energy-systems. In [5], they applied CMA-ES to solve economic dispatch problems for conventional and wind-thermal power systems. In [6], they used CMA-ES to optimize the temperature of reactors, in a hydrocracking process. The latter also trained a model offline, then applied the trained model in an online process, much like [4] also did.

Some optimization problems have multiple objectives that must be optimized simultaneously. The scope of this paper is limited to exactly 1 optimization objective, but for more information on CMA-ES utilized on multi-objective problems, see [31], [32], and [33].

C. DIGITAL TWINS

A digital twin is a digital representation of a physical system, which is simulated in real-time alongside a real physical system, given the same input data as the real physical system [34]. Reference [35] defines three key aspects necessary to make a model a digital twin: The model itself, a data-set that “evolves” over time and a way to update or adjusting the model based on this data-set. According to [36], the term digital twin has been used in various contexts with different

definitions. Their definition of a digital twin is applicable for the digital twin of the Espoo district heating network discussed in this paper, and is the following:

“A virtual representation of a physical system (and its associated environment and processes) that is updated through the exchange of information between the physical and virtual systems.” [36].

Having a model of the physical system is useful in various stages of a project, whether it is a digital twin or not. Firstly, during the design-phase of the system, the model can guide the design process by showing weaknesses in the design or verify the correctness of the design [10], [37]. The model can also be used as part of training to introduce the physical system in a safe environment [38]. The model can further be connected to the physical system’s signals, making it a digital twin, which can provide virtual measurements of the physical system from places where no measurement sensors exists in the real-world [39]. Further, the digital twin can be used to find an optimal way of operating the physical system, e.g. to reduce costs or increase productivity [40].

Digital Twins, security considerations and the technologies surrounding them have been surveyed thoroughly by [41], [42] and [43]. References [44], [45], and [46] have investigate enabling technologies for digital twins, the latter also including a comprehensive summary of digital twins used in different fields.

Reference [47] introduces a classification system for digital twins. According to their definition, the Espoo district heating network digital twin introduced in this paper falls somewhere between the subcategories digital shadow and a digital twin. The operators of the district heating network can choose between various levels of automation, from manual to automatic. When operating in manual mode, the model described in this paper falls into the digital shadow category. In the automatic mode, however, instructions are sent directly to the district heating network from the CMA-ES optimization’s results that have been calculated based on the digital twin model. This virtual-to-physical and physical-to-virtual connections are typical characteristics of digital twins, according to [48].

D. CLOUD-COMPUTING ASPECTS OF DIGITAL TWINS

A systematic review of industrial digital twins has been done by [49], with a special focus on the cloud-based technologies enabling digital twins. A comprehensive reference model for a cloud-based digital twin architecture has been proposed by [50] and the related works therein. In their architecture, physical assets have a one-to-one virtual counterpart and when the physical asset undergoes a state change the virtual counterpart is updated to reflect this new state.

In some cases, monitoring the physical system is the goal of a digital twin, like the bridge-health monitoring cases discussed by [51] or the monitoring of robotics devices discussed by [52]. Sometimes a digital twin is used to optimize the operation of a physical system, as has been explored by [53]. In all

of the cases listed here, signals from sensors in the physical system are sent to a digital twin, which was deployed as a cloud-service. One of the enabling technologies for sending sensor data is MQTT (Message Queuing Telemetry Transport protocol) as described by [54].

The closest state-of-the-art to this paper involves a cloud-based optimization digital twin approach, and is introduced by [55]. In their approach, a neural network model was trained on time-series data from an ironmaking process factory. A genetic algorithm used this neural network model and attempted to optimize multiple variables in the ironmaking process. The cloud-based optimization service was used in real-time by the ironmaking process factory to optimize the current operational parameters affecting the ironmaking process. These aspects mimic the district heating case discussed in this paper, i.e. an evolutionary algorithm is executed on a cloud-based digital twin, to optimize the operations of the real physical system.

III. ARCHITECTURE AND METHODOLOGY

A. FUNCTION-AS-A-SERVICE

FaaS (Function-as-a-service) enables execution of parallel computing, without the function caller knowing anything about the servers on which the computations are performed. Essentially, FaaS is an ICT architecture that enables functions to be computed in parallel, where functions are hosted on a cloud-computing platform and exposed to clients as HTTP endpoints. Clients call the HTTP endpoint with a certain payload, e.g. json and receive a reply payload, e.g. json. The FaaS ICT architecture ensures that the call made by the client is scheduled to an existing or a new computational instance in the cloud-computing platform, typically a container. For the purpose of this paper, and from an evolutionary algorithm's point-of-view, a fitness function can be evaluated as a remotely computed function in a FaaS architecture.

FaaS comes in many forms, from pay-as-you-go solutions such as Google Cloud Functions,¹ Azure Functions,² and AWS Lambda³ to self-hosted options such as Knative,⁴ Nuclio⁵ or OpenFaaS.⁶ For a technology review of these and other FaaS platforms and a classification framework for them, see [56]. For a parallelism benchmark for the biggest pay-as-you-go FaaS providers, see [57].

A typical function in a FaaS platform is either a piece of code written in one of the languages supported by the FaaS provider [58], or a custom container image created by the user. In pay-as-you-go offerings, there are limitations placed on things like the operating system available, size of container images and size of function payloads. For many, the limitations are not a problem. However, with a self-hosted

solution such as OpenFaaS, the limitations can be avoided altogether.

Reference [59] identify “warm” containers as being an important pattern for FaaS application developers. A “warm” start refers to a container which is already running at the time a function-call is scheduled to it, whereas a “cold” start requires the container to be started. Various mitigation strategies for the “cold” start have been suggested, see [60] and [61].

Essentially, a “cold” start FaaS container architecture is acceptable when the number of function calls is low, infrequent and there is no time-constraints for how long it takes to receive the results from the function. This results in lower costs when the containers are in shut-down state, in exchange for slower function replies due to the time it takes to start the containers again. In contrast, a “warm” start architecture should be used when the function is frequently called and it is important to minimize the time it takes to receive the results from the function. From an optimization algorithm's point-of-view, such as discussed in this paper, a “warm” start architecture is crucial to ensure as-fast-as-possible function calls. [59]

B. CMA-ES ALGORITHM IMPLEMENTATION

CMA-ES is used as the evolutionary algorithm for generating the setpoints to the district heating network. Unlike the genetic algorithm, which generates a new population at each iteration, CMA-ES generates a normal distribution, from which a candidate solution is sampled [24]. In the Espoo district heating network case, the candidate solution contains functions of time that specify the supply temperature setpoints at the heat producers and the setpoints for usage of a thermal energy storage tank, the same two variables chosen by [8]. The candidate solution also contains the dispatch list, which is the optimized order in which the heat producers are allowed to be started in the network, based on forecast cost, capacity, and limitations. However, the final decision for how to operate the network is the responsibility of TLA (top-level automation), an automation software system. The TLA exists in the real network and the digital twin model, and requires inputs such as heat producer's supply temperature, and the dispatch list. The TLA has been developed completely separately from the work described in this paper.

The temperature setpoint is interpolated from hourly time series selected by CMA-ES. For heat accumulator, the setpoint is piece-wise constant value selected by CMA-ES for each hour, due to the slow thermohydraulic phenomenon intrinsic to a district heating network, described in section I. The phenomenon also involves a trade-off between computational complexity and accuracy, and for our case study a 16-hour time-window was chosen. Thus, the sampling of the CMA-ES candidate solution should give 16 values for the supply temperature setpoint and another 16 values for the thermal (heat) storage setpoint. Therefore, our formulation of the CMA-ES algorithm results in a 32-dimensional normal distribution, and sampling it results in these 16+16

¹<https://cloud.google.com/functions>

²<https://azure.microsoft.com/en-us/services/functions/>

³<https://aws.amazon.com/lambda/>

⁴<https://knative.dev/docs/>

⁵<https://github.com/nuclio/nuclio>

⁶<https://www.openfaas.com/>

TABLE 1. Pycma library parameters.

| Parameter | Value | Description |
|---------------|------------|---|
| popsize | 36 | The population size, i.e. number of samples taken from the normal distribution. This represents the number of candidate solutions tried every iteration. |
| $\mathbf{x0}$ | 32 · [0.0] | The candidate solution phenotype array and its dimensions, contains 16 + 16 values. |
| σ_0 | Dynamic | The initial standard deviation (step-length), set to be the same as the last σ_0 of the previous optimization-cycle, if no previous σ_0 exists, then 1.0 is used. |

setpoint values. The optimality of the candidate solution is evaluated against a fitness function. The algorithm continues to produce candidates with a higher fitness function, hopefully resulting in a near optimal solution after the algorithm has been executed for a sufficient number of iterations. The careful design of the fitness function is crucial for ensuring the real-world relevance of the optimization and to manage computational complexity. In the literature of evolutionary algorithm applications to industrial processes (see section II-B), authors usually hand craft the fitness function as a formula. This approach works best at a limited scope, such as a control loop or subprocess. However, the optimization task often cannot be isolated to a subprocess in large-scale industrial processes.

As mentioned before, in our case study, the CMA-ES created candidate solutions for how the district heating network should be operated. A good candidate solution operates the plant as cheaply as possible, while keeping the heat at sufficient levels for all consumers. To test how well they performed, an equation was not used: instead, a dynamic process simulation was used. The inputs to the simulation were the candidate solution's setpoints, in addition to a dispatch list and forecast environment data. The simulation time was 28 (16+12) hours, simulated faster-than-real-time. The results of the simulation gave the fuel consumption for each heat-producing plant in the network for the 28 hours. Using fuel-costs, the total cost for operating the network could be calculated, which was the final fitness value for the candidate solution.

The CMA-ES algorithm [62] used in this project was implemented by the pycma library [63]. The CMA-ES algorithm was given 1 hour real-time to solve the optimization problem and the best candidate found at that point was chosen as the best solution. The pycma library parameters used and the description of them can be seen in Table 1. As can be seen, the population size was 36, the phenotype dimensions was 16+16 and the σ_0 depends on the previous optimization hour's results.

Smooth operation is preferred in a district heating network. This means that the operating plan suggested by the CMA-ES algorithm should not contain large increases or large decreases in temperature. This is true for both the supply temperature and the thermal storage tank setpoints. Pycma

TABLE 2. Parameters for the initial $\mathbf{C}(0)$ covariance matrix creation.

| Parameter | Value | Description |
|------------|----------------|---|
| θ | $\frac{1}{12}$ | The correlation between values in the generated distribution, a small θ means larger correlation. |
| s_{Tsp} | 1.0 | The supply temperature setpoint scaling, smaller modifications are allowed in the Ornstein-Uhlenbeck than for the storage tank setpoints. |
| s_{Qacc} | 3.0 | The heat storage tank setpoint scaling, three times larger modifications are allowed in the Ornstein-Uhlenbeck than for the supply temperature setpoints, allowing a larger area to be explored faster. |

library creates, in the beginning of the optimization, candidate solutions \mathbf{x} with zero mean and identity covariance matrix. We transform these to $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$. \mathbf{A} is chosen so that the initial covariance of \mathbf{y} is

$$\mathbf{C}(0) = \begin{pmatrix} s_{Tsp}^2 \mathbf{E}^\theta & 0 \\ 0 & s_{Qacc}^2 \mathbf{E}^\theta \end{pmatrix}, \quad (1)$$

where $\mathbf{E}_{ij}^\theta = e^{-\theta|i-j|}$ and \mathbf{b} is the initial guess. \mathbf{y} is the transformed candidate solution with proper smoothed temperature setpoints that are acceptable to the district heating network operation. \mathbf{b} is the initial guess for the solution (step 2b in Fig. 2), which is derived from the best candidate solution of the previous optimization run. The parameter values for θ , s_{Tsp} and s_{Qacc} and their descriptions can be seen in Table 2. Thus, supply temperature and thermal storage tank setpoints are initially independent of each other, and each has the covariance of the Ornstein-Uhlenbeck process ($e^{-\theta|i-j|}$) [64]. This is a well-known mathematical approach for achieving desired covariance between variables [65].

The transformation of the candidate solutions \mathbf{x} to smoothed candidate solutions \mathbf{y} is done in every iteration of the CMA-ES algorithm, using the same matrix \mathbf{A} and vector \mathbf{b} created before the first iteration.

C. CONCEPTUAL ARCHITECTURE

In our case-study, a dynamic process simulation model that contains a model of the district heating network of Espoo is used both as the digital twin and the optimization fitness function evaluator. As illustrated in step 1a of Fig. 2, the digital twin instance is continuously updated against sensor measurements from the physical process. Fig. 1 shows the flow of data from sensors to the digital twin. Sensors in the district heating network send signals to a shared data platform with timestamps, identifiers, and values, whenever a value changes. These values are further sent to an online simulation manager whenever the values changes. This online simulation manager ensures the tracking instance is running in real-time by sending values and simulation instructions periodically to the dynamic process simulator.

The conceptual architecture seen in Fig. 2 contains various steps and stages, labeled with a number for the step and a

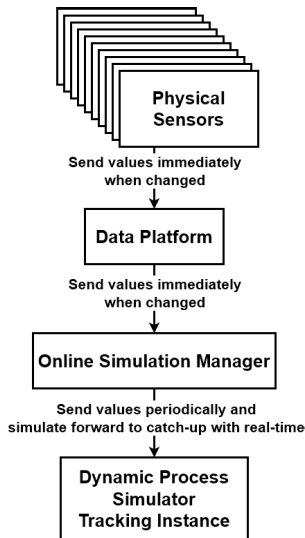


FIGURE 1. Data-flow from sensors in the real Espoo district heating network to the digital twin tracking instance.

letter for the stage, i.e. **1a** means step **1** stage **a**. The lists below describe each stage and step in detail.

Stages in Fig. 2:

- Stage **a**. Background stage, performed independently from the CMA-ES optimization algorithm.
- Stage **b**. Preparation stage performed once per hour, at the beginning of the optimization.
- Stage **c**. Main optimization stage, performed multiple times on multiple optimization simulation instances.
- Stage **d**. Clean-up stage performed once per hour, at the end of the optimization. At the end of this stage, the optimization algorithm has provided a suggested plan for how to operate the district heating network most cost-efficiently. The plan consists of the setpoints for the thermal storage tank and the heat producers' supply temperature for the next 16 hours. The thermal storage tank setpoint corresponds to a charge or discharge operation, i.e. the storage tank is either filled or emptied a certain amount.

Steps in Fig. 2:

- 1a.** The digital twin simulation model is kept up-to-date with sensor values from the physical process.
- 2b.** Each hour, the CMA-ES algorithm keeps the previous hour's distribution's mean and the step-length (σ_0), but not the covariance matrix, and uses these as the initial guess for the next hour's optimization. If no previous best candidate solution exists, a default value is used instead (this occurs only once, when deploying the architecture for the first time).
- 3b.** The state of the digital twin is captured as an initial condition to initialize the separate optimization (fitness function) instances of the simulator.
- 4b.** The optimization simulation model requires as inputs the outside air temperature and the heat consumption at

end-user locations. Since the future is being simulated by the CMA-ES optimizer, real measurements cannot be used for these values. Instead, they are supplied by a forecast-generator for the duration of the optimization (16+12 hours). The forecast-generator had already been implemented before this project began and it utilized historical data and machine-learning for predicting consumer behaviour.

- 5c** A candidate solution is a 32-dimensional matrix with 16 CHP supply temperature setpoints and 16 thermal storage temperature setpoints, representing the temperature setpoints for the next 16 hours, one value per hour for both types.
- 6c** Using the 16 hours of setpoints generated by sampling the CMA-ES candidate solution, the fitness function instance is run for 16 hours. After that, to account for thermohydraulic delays in the system, it is run for a further 12 hours without decisions or optimizations.
- 7c** The result of running the fitness function instance for 16+12 hours is a fuel consumption, for all heat producers in the network, for each hour. The fitness function also includes penalty information if acceptable service for all consumers has not been reached. Fuel consumption is converted to operating costs for each producer, by multiplying the fuel cost for that hour with the consumption, efficiency of the producer and taxation of the specific fuel. The final result is the total operating cost for the 28-hour period, plus penalties, and this is the fitness value for the CMA-ES candidate solution.
- 8d** After 1 hour, the CMA-ES algorithm is stopped and the best candidate solution found is suggested as the operating plan of the district heating network.
- 9d** The system is currently deployed in a semi-automatic mode, in which all setpoints used during the optimization are sent to the operators of the district heating network for evaluation and they can choose to utilize those setpoints in the real district heating network if they accept them.

Since the training begins anew once per hour, there is only one hour time to complete the CMA-ES and to find the best solution. Due to computational limitations, it is not possible to run as many iterations of the CMA-ES as would be desirable to achieve a near optimal solution. To mitigate this problem, N fitness function instances of the simulator are run in parallel. In Fig. 2, this has been illustrated for $N=4$. The 32-dimensional distribution that is maintained by CMA-ES is sampled N times, and each sample is sent to a dedicated fitness function instance of the simulator. Each of these instances run in parallel and return a separate fitness value. The interaction between CMA-ES and the simulator is illustrated with dashed lines. To avoid clutter, these lines are drawn only between one candidate solution and one instance, but the same interactions are performed for all N instances. In addition, a medium fidelity model of the district heating network is used, where the number of pipes and consumers

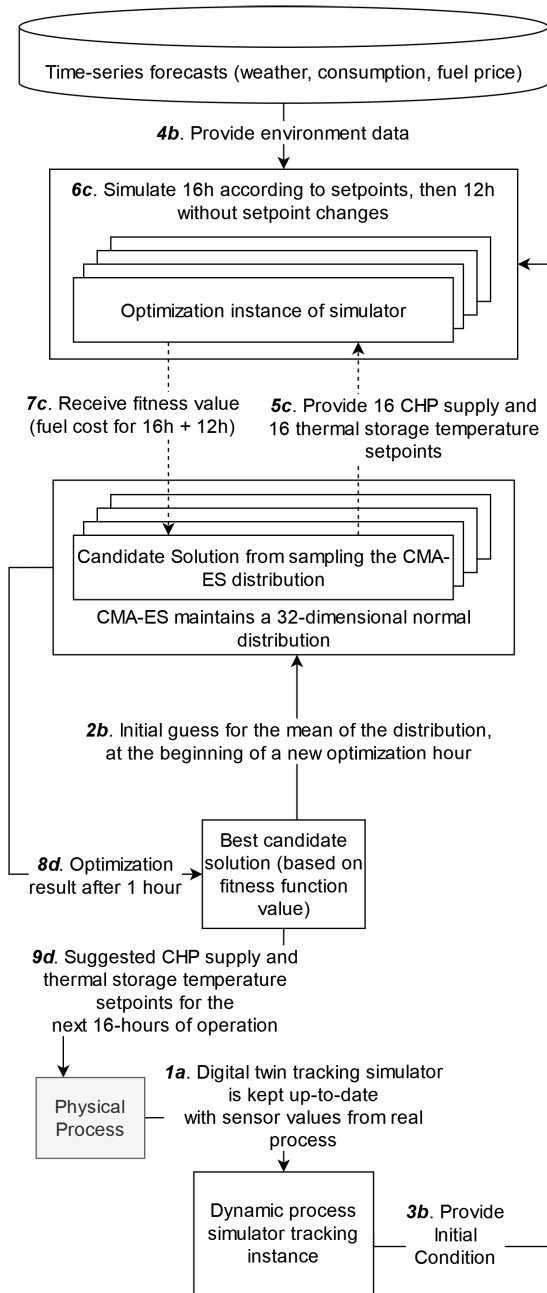


FIGURE 2. Conceptual architecture.

are reduced by combining them into fewer larger pipes and consumers.

The setpoints of the district heating model are updated once per hour, so the training of the CMA-ES algorithm is performed once per hour, as specified in the flowchart in Fig. 3. The algorithm has two nested loops: one iteration of the outer loop is one iteration of the CMA-ES algorithm, and the inner loop exploits the parallel computation capabilities of the architecture. Specifically, the inner loop initiates N concurrent fitness function instances of the simulator (see Fig. 2) to evaluate the fitness of N candidate solutions sampled from the distribution maintained by the CMA-ES algorithm. The

TABLE 3. Symbols used in flowcharts.

| Symbol | Description |
|----------------|--|
| N | The number of candidate solutions in one iteration of CMA-ES |
| IC_{twin} | Initial condition for a simulator based on the current state of the digital twin simulator |
| t | Running time of the algorithm |
| Sim_{ref} | A reference simulator used to estimate the state of the system 1h into the future |
| IC_{ref} | Initial condition of Sim_{ref} |
| $distribution$ | 32-dimensional normal distribution maintained by CMA-ES |
| $candidate_i$ | The i^{th} candidate solution sampled from the 32-dimensional distribution, containing 16 supply temperature setpoints and 16 thermal storage tank setpoints |
| $fitness_i$ | The fitness value of $candidate_i$ |
| Sim_i | The fitness function simulator for obtaining $fitness_i$ for $candidate_i$ |
| $sp_{CHP}[]$ | 16 element array of setpoints for the supply temperature of the CHP, HOB, heat-pump or waste heat producers |
| $sp_{tank}[]$ | 16 element array of setpoints for the temperature of the thermal storage tank |
| $fuel[]$ | 28 element array of hourly fuel consumption of all the CHP, HOB, heat-pump or waste heat producers, as determined by Sim_i |
| $price[]$ | 28 element array of hourly fuel prices |

inner loop of Fig. 3 references another flowchart in Fig. 4 that specifies how the fitness of one candidate solution is evaluated. A live deployment of the algorithm requires that the CMA-ES algorithm is able to output the setpoints to the district heating network after 1-hour of execution. Thus, the outer loop in Fig. 3 is iterated until the 1-hour execution time limit is reached. In other words, the number of iterations of the CMA-ES algorithm is limited by the computational capacity. Thus, the procedure will generate setpoints to run the system starting one hour into the future. For this purpose, the fitness function instances of the simulator are initialized with the expected state of the system one hour into the future. This state is obtained by initializing a reference simulator with the initial condition of the digital twin and running it one hour into the future. The reference simulator and fitness function simulators are all running faster than real time to estimate the future state of the district heating network, so they use forecasts of weather and consumption as the environmental data (see Fig. 2). All italicized symbols in Fig. 3 and Fig. 4 are defined in Table. 3.

IV. DETAILED DESIGN AND IMPLEMENTATION

In our district heating network case, there was a time-constraint to find an optimal solution to suggested to the district heating network operators. As such, an algorithm that could provide good candidate solutions given only a few iterations was needed. From the related work described above, CMA-ES has been proven to be a good choice when the number of iterations is limited. A complex dynamic simulation model of the entire district heating network of Espoo is used to solve the CMA-ES candidate solutions

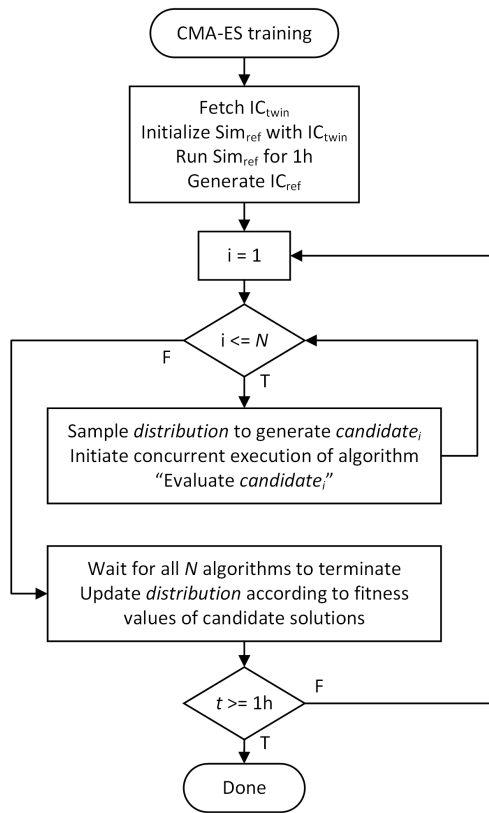


FIGURE 3. One iteration of the CMA-ES algorithm.

where a global optimum is impossible to calculate numerically. The model is more complex than the PID controller model, data-driven hydrocracking model or thermodynamic chemical/phase models introduced in the related works above.

In the previously mentioned works, a single computational program was enough to calculate the results of all candidate solutions. However, in our approach, we combined CMA-ES and serverless concepts such as Function-as-a-Service to utilize distributed computational capacity when solving the CMA-ES candidate solutions. A distributed and thus parallel computational approach was needed due to the complexity and simulation time of the model.

The components needed for the candidate ICT architecture presented in this paper are as follows: A simulation tool, an evolutionary algorithm implementation, a function-as-a-service framework where multiple instances of the simulation tool is deployed. For our case, the simulator used is Apros,⁷ a dynamic process simulator, see [10] for a scientific introduction to Apros and [39] for examples of simulation-based digital twin approaches using Apros. The evolutionary algorithm, CMA-ES, was implemented in Python, based on the package pycma.⁸ The function-as-a-service framework used was OpenFaaS. See Fig. 5 for an overview of the architecture

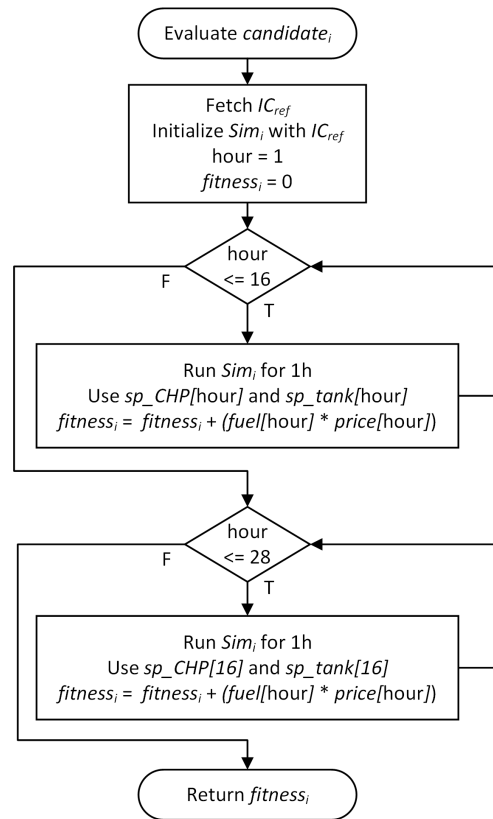


FIGURE 4. Concurrent evaluation of the fitness of one candidate solution.

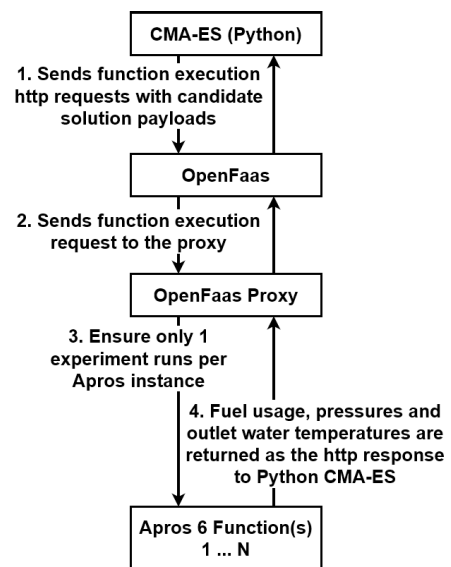


FIGURE 5. OpenFaaS architecture for utilizing parallel Apros 6 instances as fitness functions in an evolutionary algorithm (CMA-ES).

and section IV-A for an overview of the district heating network Apros model.

The Python code was written in such a way that all candidate solutions were executed in parallel. Each of the candidate solutions perform an http request to our OpenFaaS instance.

⁷<http://www.apros.fi/en/>

⁸<https://github.com/CMA-ES/pycma>

The payload of the http request contains the instructions to the Simulator (Apros 6) to simulate the model in a particular way, representing the candidate solution. The functions return the fuel consumed during the simulation, as well as the consumer's pressures and outlet water temperatures. The predicted prices for fuel is fetched from a forecasting service and used by the CMA-ES Python algorithm to calculate the actual fitness value for a candidate solution, based on the fuel consumed by the candidate solution.

A. THE SIMULATION MODEL

The underlying simulation model was implemented in the Apros[®] simulation software [66]. The model can be divided into two main parts, a thermohydraulic model and an automation model. The thermohydraulic model is used to simulate the flow rates, temperatures and pressures throughout the network and is based on the conservation equations for mass, energy and momentum. The equations are partial differential equations with one spatial coordinate and are discretized with respect to this coordinate with a so-called staggered grid scheme in which two grids are used, one for solving pressures and temperatures and another for flows [67]. This results in a set of ordinary differential equations, which in turn are integrated over time with the implicit Euler method. This results then, during every time step, in a set of algebraic equations which are solved. In addition, inside every time step the material properties of the fluid, e.g. density and viscosity, are estimated, e.g. using lookup tables.

In order to derive correct coefficients to the underlying equations, physical dimensions of the network are used. Namely, pipeline lengths, diameters, elevations and pressure loss parameters are used. Furthermore, equipment characteristics such as pump curves are entered to the model and taken into account via source terms in the equations. Finally, pipe and insulation materials are taken into account when calculating the temperatures and heat losses to the environment.

The automation model consists of function blocks which are similar to those found in all real-life automation system. Such blocks included adders, multipliers, limiters, integrators, (PID) controllers etc., which are connected to each other using analog or binary signals. The resulting automation model is solved in a sequential modular manner every time step. In this solution method boundary-type modules (set points) are selected as a starting points and then the solution proceeds module by module through the network, following the signal connections. Loops arising from the automation model structure are cut either at user defined locations or arbitrarily.

The two models are connected, similarly as in real-life: measurement modules pick desired values from the thermohydraulic model for the automation model and so-called actuator modules transfer automation's control signals to the thermohydraulic model (e.g. valve positions or pump rotation speeds). Furthermore, in the tracking instance, the online simulation manager sends measurement data to modules of

the automation model which then processes and transfers them to the thermohydraulic model.

In order to facilitate efficient modelling, the software was extended to better suit district heating simulations using so-called user components or in other words hierarchical models. With user components certain parts of the model which are replicated numerous times can be packaged into one reusable component, a prime example being a district heating pipe which includes two pipelines (one going towards the customers and one returning) as well as heat loss calculations.

Two versions of this model was developed, a high-fidelity and a medium fidelity versions. For the purpose of the CMA-ES optimization, in order to limit the computational requirements, the medium fidelity model was used. The medium fidelity model was validated in two stages. Firstly, an expert evaluation was used to ascertain that the results are reasonable and an optimization can be built upon them. Secondly, comparisons were made with real-life measurements over five one-week validation periods. For each validation period, over 100 measurements were utilized. Three validation figures were chosen to be presented in this paper. They represent the district heating network operating at an average outside air temperatures of Espoo. Fig. 6 is important, because the heat reaching the consumers must be accurately modelled. Fig. 7 is an example of the behaviour of the producers in the model. Fig. 8 is important to model correctly, since the optimization algorithm decides how it should be operated. Due to confidentiality, values have been scaled with (2), where MAX was the maximum value of the reference and simulated values. Error tolerances in the validations is defined as 5% below or above the reference data. All measurements for the medium fidelity model should preferably always stay within the error tolerance region, but case-by-case judgement is applied for measurements that go outside this region, as seen in Fig. 7. In our case study, the result of all of the case-by-case judgements by industrial end users was positive in the sense that the model was approved for industrial application of the optimization problem presented in this paper. As is normal, during the validation discrepancies were detected. Their severities were evaluated and most important ones were corrected. This iteration was continued until the end customers and the development team were satisfied with the model.

A digital twin model can be modified and adapted to the tracking data in order to better fit the desired measurements, see [39]. However, in our case, the Apros model is not adapted in this way. Instead, KPIs (key-performance indicators) track how well the digital twin model is calculating the same outputs which exists in the real system. Optimized operating plans suggested by the CMA-ES algorithm are not used when the tracking digital twin's KPIs are bad. The reason for no adaptation is to speed up the initialization-step of the CMA-ES algorithm, where all parallel optimization Apros instances are reset into a state that matches the tracking digital twin. Exporting the Apros 6 model and importing it is a

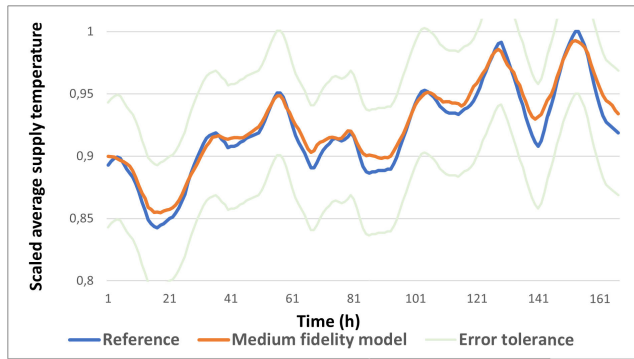


FIGURE 6. Model validation comparison of consumer average supply temperatures.

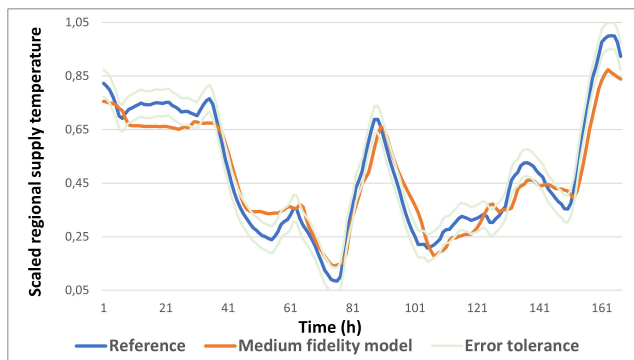


FIGURE 7. Model validation comparison of a representative consumer's regional supply temperature.

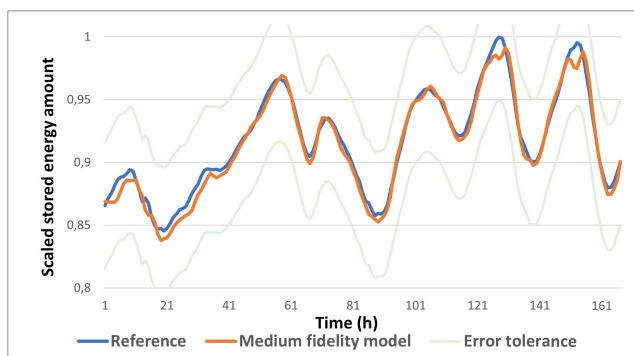


FIGURE 8. Model validation comparison of energy stored in the heat storage tank.

time-consuming operation for a model of any decent complexity, which is why an orders-of-magnitude faster initial condition is used. However, an initial condition does not capture adaptations made on a model.

B. PARALLEL CMA-ES ARCHITECTURE

Due to the relatively slow simulation speeds of the dynamic process simulator, multiple instances of the simulator are needed to calculate the fitness function values of all the candidate solutions. In our case study, the CMA-ES algorithm's population size was 36 and therefore to maximize

the simulation speed, 36 Apros 6 function instances were deployed, ready to compute the fitness function values of the candidate solutions generated by CMA-ES, one Apros 6 instance for each candidate solution. A headless version (i.e. non-desktop version, without user-interface) of Apros 6 was packaged as a Windows Docker container, using Windows containers.⁹ These headless Apros 6 instances implemented the function interface and registered as functions in the OpenFaaS instance.

For the purpose of the architecture discussed in this paper, FaaS means that the Python CMA-ES algorithm contacts the OpenFaaS instances using an HTTP request whenever it wants a candidate solution's fitness function value evaluated. The OpenFaaS instance then distributes the request to one of the available Apros 6 instances. In other words, a fitness function evaluation in the CMA-ES algorithm is, in our architecture, a single HTTP request with a payload as the input, and the http response content is the fitness function results. The fitness function itself is a simulation in Apros 6, using a specific model, initial condition, environment data and candidate solution setpoints.

The deployed function often comes with limitations in ready-made solutions such as AWS Lambda. For a complex computational function such as Apros, a custom solution was required. A Windows docker container running of-watchdog¹⁰ as a reverse proxy for receiving HTTP requests was implemented. For Apros 6, a plugin was developed that starts a local service that the of-watchdog is connected to. Of-watchdog serves as the endpoint of the docker container and ensures that the container used is compatible with the OpenFaaS framework. The of-watchdog is a reverse proxy that sits between the OpenFaaS server and the Apros function implementation. See Fig. 9 for a visualization of how http requests are handled by the of-watchdog and Apros inside the container.

To mitigate the cold-start problems described in section III-A, the amount of Apros containers available to respond to the FaaS jobs was kept static. Further, all containers were kept running at all times, ensuring "warm"-start for all the function calls.

C. NOISY NEIGHBOUR PROBLEM

An important aspect in cloud computing is the noisy neighbour problem, described in greater detail by [68] and [69]. The servers typically used in cloud computing are virtual machines, hosted on a physical server with multiple other virtual machines. The person paying for the virtual machine does not control the physical server or the other virtual machines. This means that it is possible to pay for a virtual machine that is running on a heavily utilized physical server. This is noticed in the performance of the virtual machine, which can be significantly slower than another equal virtual machine running on another physical server. In the CMA-ES optimization

⁹<https://docs.microsoft.com/en-us/virtualization/windowscontainers/>

¹⁰<https://github.com/openfaas/of-watchdog>

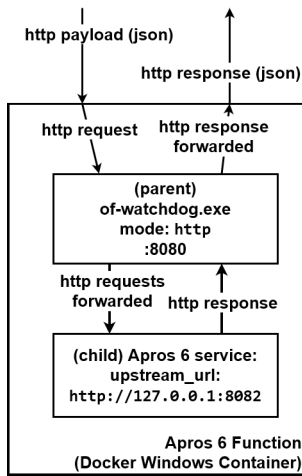


FIGURE 9. Apros 6 function container with OpenFaaS function implementation, using of-watchdog proxy. HTTP requests are received by the container’s of-watchdog instance and forwarded directly to the Apros 6 service.

scenario, each simulation result must finish in order for the CMA-ES algorithm to create and start the next generation. This means the CMA-ES algorithm’s performance is equal to the slowest virtual machine that is running the Apros optimization jobs. The noisy neighbour problem can be seen in Fig. 10. All virtual machine running Apros OpenFaaS optimization jobs were equally powerful, i.e. same speed as promised by the cloud provider. In reality, the speeds were completely different, due to the noisy neighbour problem. Values for Fig. 10 were taken after running 400 CMA-ES generations (i.e. fitness function evaluations) on each virtual machine.

D. CONNECTING CMA-ES WITH A DIGITAL TWIN

See Fig. 11 for an overview of how the CMA-ES utilized the initial condition and the Apros 6 functions to solve the fitness functions. Apros 6 requires a model that represents the district heating network. Before simulation, an initial condition can be provided to the model that represents the network in a particular state. During optimization, we utilize a tracking Apros 6 instance, i.e. a digital twin, of the district heating network. This digital twin is kept synchronized with the real district heating network, completely independently from the optimization algorithm and Apros 6 OpenFaaS functions. When CMA-ES begins an optimization calculation, it requests the most up-to-date initial condition that the digital twin can provide, i.e. it captures the current state of real network in a binary initial condition blob that can be provided to the Apros 6 functions as a starting point.

E. CMA-ES IMPLEMENTATION DETAILS

In our case, CMA-ES performs as many iterations (generations) as possible during a 1-hour (real-time) time-window. A 1-hour time-window was chosen for multiple reasons. Firstly, the changes in forecasts should be taken into account

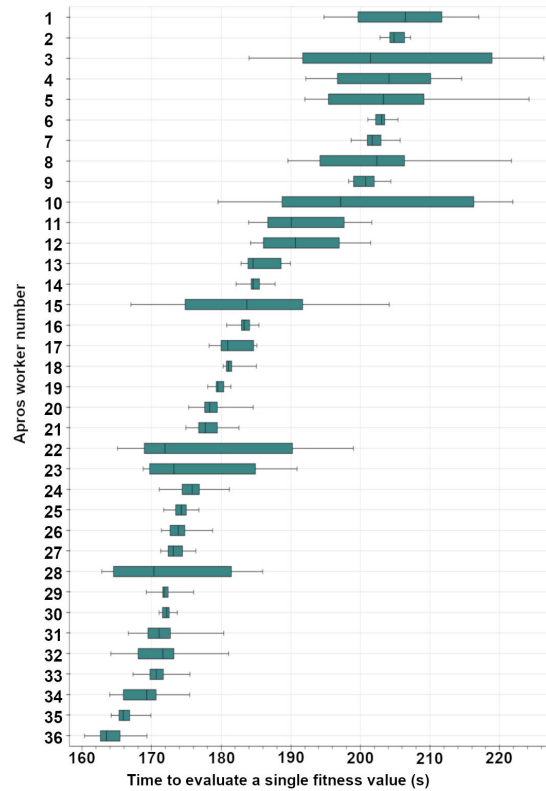


FIGURE 10. Visualization of the noisy neighbour problem, each Apros worker is running on a different virtual machine.

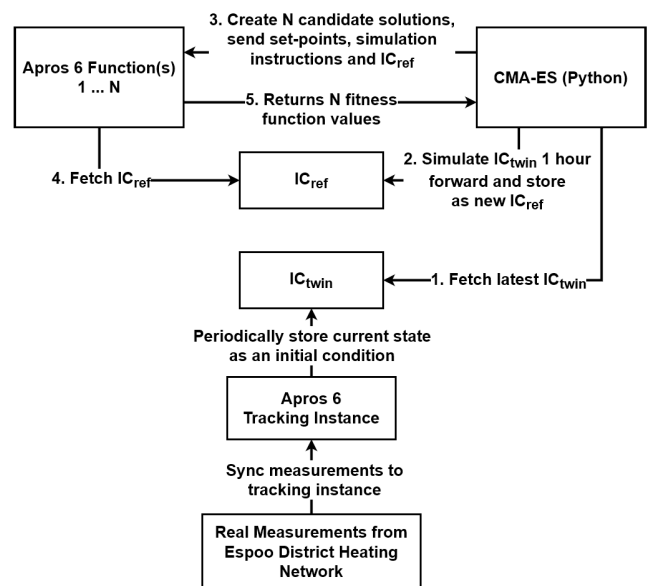


FIGURE 11. Implementation of CMA-ES in a digital twin setting. The algorithm utilized the initial condition of a tracking Apros 6 instance as the starting-point for the model.

as quickly as possible. By running an optimization algorithm continuously once an hour, the forecast data will at most be 1-hour old for any of the candidate solutions suggested by CMA-ES. Secondly, the way power plants are allowed

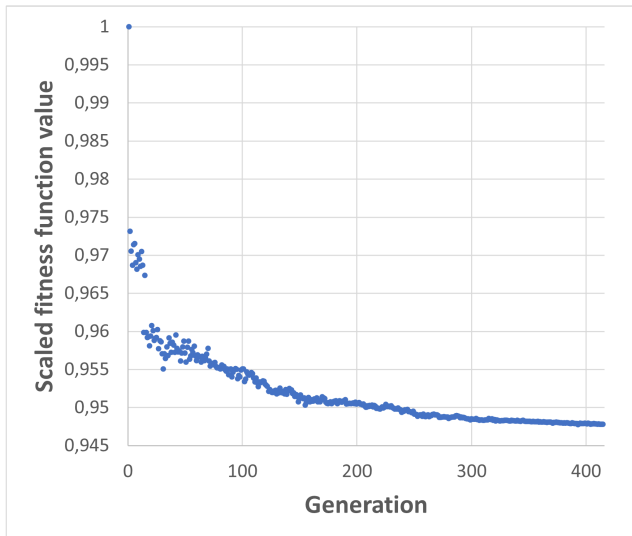


FIGURE 12. The best CMA-ES candidate solution for each generation.

to operate can change at any time, i.e. enforced maximum capacity or shutdowns due to maintenance. These types of changes effectively invalidate any candidate solution suggested by CMA-ES, until CMA-ES takes them into account. Further, the change cannot be taken into account mid-optimization, since it changes what the function is attempting to optimize, invalidating any progress that has been done up until that point.

Thus, with longer time-windows, there is a higher potential for outdated and invalid candidate-solutions to be suggested by the CMA-ES algorithm. However, with too short time-windows, CMA-ES does not have enough time to perform enough generations to find any significant savings. As seen from Fig. 12, CMA-ES quickly finds better candidate solutions with few generations. Table 4 visualizes the total reduction of the fitness function value, given different amounts of CMA-ES generations. As can be seen in Fig. 12 and Table 4, there is a diminishing return for adding more generations. A compromise of 1-hour was chosen in this project, since the CMA-ES algorithm was observed to perform well enough with the roughly 18 generations it had time to perform during the 1-hour window. The fitness function value was reduced by 4% in that time, compared to a 5.2% reduction after 415 generations. However, the improvements that can be reached are subject to the randomness of the CMA-ES algorithm, as seen in Fig. 13. In practice, the randomness results in cost-savings that vary hour-by-hour by the CMA-ES algorithms, as seen in Fig. 12. Values in Fig. 12 have been scaled according to (2), where MAX was the fitness function value of the initial guess before generation 1.

Because CMA-ES has 1-hour to complete the optimization task, it means that the solution it suggests will be received 1-hour from when it started. We do not want to unnecessarily try to find an optimal way to run the plan during the hour that is lost this way. To avoid this, at the beginning of the optimization, the initial condition for the current state of

TABLE 4. Fitness function value reductions given different amounts of generations.

| Generation | Fitness function value reduction |
|------------|----------------------------------|
| 1 | 2.69% |
| 5 | 2.85% |
| 10 | 3.15% |
| 15 | 4.01% |
| 20 | 3.93% |
| 25 | 3.98% |
| 35 | 4.08% |
| 50 | 4.41% |
| 100 | 4.50% |
| 200 | 4.95% |
| 415 | 5.22% |

the network is taken and simulated 1-hour forward with the current set-points. The new initial condition received this way better represents the state of the network as it will look like when the optimizer suggests its final solution.

The optimization problem attempts to find the cheapest possible way to run the district heating network, based on fuel price forecasts and weather forecasts, for the next 28 hours. The network must at all times have enough pressure in the pipes to continue operation and must satisfy all consumers' demands for heating. The plant also contains a long-term water tank used for heat accumulation when excess and/or cheap fuel is available, which is then utilized when demand is high or fuel is expensive. An optimization function that wants to operate the plant as cheaply as possible would always drain this heat accumulator of all power, since it is "free energy". In order to prevent this, a long-term operational goal for the heat accumulator was specified by the company running the district heating network and the optimizer must always utilize the heat accumulator in a way that respects this plan, or be penalized heavily in the fitness function's evaluation.

A single candidate solution contains 32 data-points, 16 supply temperature set-points and 16 heat accumulator set-points. These 16 values are provided to the Apros 6 model once per hour. Supply temperature values are interpolated, whereas the heat accumulator values always set the desired value for the next hour. The restrictions on the heat accumulator only apply at the very end and at that point, the power remaining in the heat accumulator must equal to or higher than specified by the long-term plan. After the simulation has performed 16 hours of simulation, i.e. used all the set-points provided to it and simulated the model 16 hours forward, it will perform an additional 12 hours worth of simulation based purely on the weather forecast.

F. CMA-ES RANDOMNESS

It is well known that evolutionary algorithms have innate randomness in their search and CMA-ES is no exception to this. To demonstrate the randomness, see Fig. 13 below,

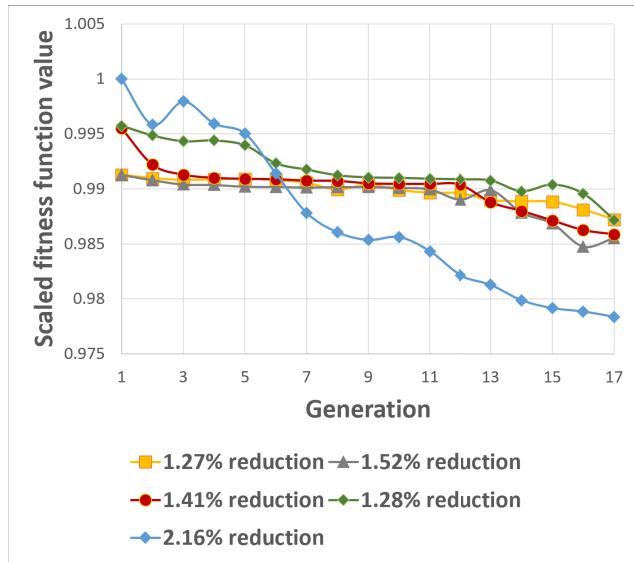


FIGURE 13. CMA-ES algorithm randomness for finding the best candidate solution.

where the CMA-ES algorithm introduced in this paper has been executed five times with the same inputs. The number of generations has been capped at 17, to reflect the typical number of generations our CMA-ES algorithm has time to calculate during the 1-hour window (see section V). The names of the colored series in Fig. 13 reflect the best candidate solution of that series, when compared to the overall worst guess of all the data-points shown in Fig. 13.

The inputs for the CMA-ES algorithm in Fig. 13 were taken between 00.00 9th Aug. 2021 and 04.00 10th Aug. 2021 (a 28-hour period). The inputs include all the necessary data needed by the model and CMA-ES, i.e. weather-data, fuel prices and Espoo district heating network plant availability data for that time-period.

V. RESULTS

The CMA-ES optimization algorithm was used to optimize the operating plan of the district heating network of Espoo, which contains over 800 km of water pipes and over 14 producers, including CHP plants, HOB plants, heat-pumps or waste heat facilities. However, only the 14 biggest producers were modelled in the medium fidelity Apros 6 model used in the optimization.

The results seen in Fig. 14 have been taken from 49 consecutive CMA-ES optimization iterations, each lasting 1 hour, with a population size of 36. During those 1-hour periods, the CMA-ES algorithm was tasked to find as good a candidate solution as possible, with 36 parallel Apros 6 instances, with as many generations as it had time with. In practice, the algorithm usually had time to run 17-18 generations each hour. To mitigate randomness of the CMA-ES algorithm, at the beginning of each optimization, the previous optimization's distribution's mean and the step-length (σ_0) is used as the initial guesses for the CMA-ES algorithm. This is done to ensure the algorithm begins searching in a space that has

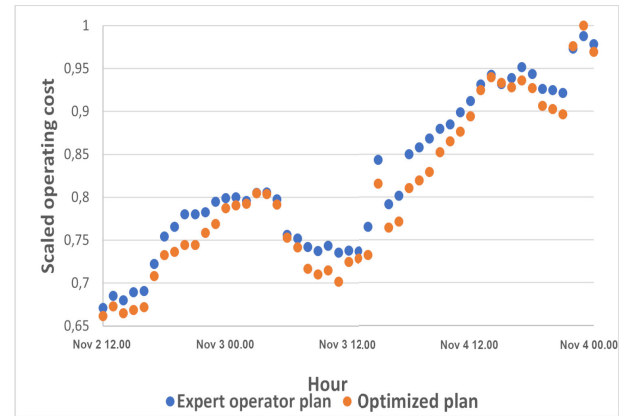


FIGURE 14. Comparison of CMA-ES results and the existing approach (Expert operator plan).

TABLE 5. Comparison of total operating cost (scaled) for the time period Nov 2 12.00 to Nov 4 12.00.

| Expert operator plan | Optimized plan |
|----------------------|----------------|
| 1.0 | 0.978 |

previously been found to be good, ensuring the algorithm doesn't start from scratch each hour.

As can be seen in section IV-F and Fig. 13, the algorithm typically improves each generation, but not always towards the same solution. Given the relatively small number of generations (17-18) that our CMA-ES algorithm has time with, this randomness does play a large part to explain the differences of achieved operating-cost improvements seen in Fig. 14.

The best candidate solution produced by the CMA-ES algorithm after each 1-hour optimization has been compared with the expert operator plan, see Fig. 14 and Fig. 15. In real district heating networks automatic operating-cost optimization solutions have not yet been taken into use. Instead, human operators of the district heating networks decide hourly setpoints for the heat storage and the supply temperature of the CHP and HOB plants. This is also the case with our Espoo case-study network. In this paper, the term 'expert operator plan' means those human-decided setpoints for a period of one or more days. The real costs cannot be disclosed in this paper due to confidentiality. Instead, scaled values have been used in Fig. 14, obtained using (2).

$$V_{\text{scaled}} = \frac{V_{\text{actual}}}{\text{MAX}}, \quad (2)$$

where MAX was the highest operating cost observed between 12.00 2nd Nov. 2022 and 12.00 4th Nov. 2022. For that same time-period, we reached about 2% cost savings compared to industrial state-of-the-art (i.e. the expert operator plan), see Table. 5.

VI. DISCUSSION

In this paper, we have presented an approach and architecture that fulfils the research questions' goals. In this chapter, those research questions will be discussed, and results are analysed.

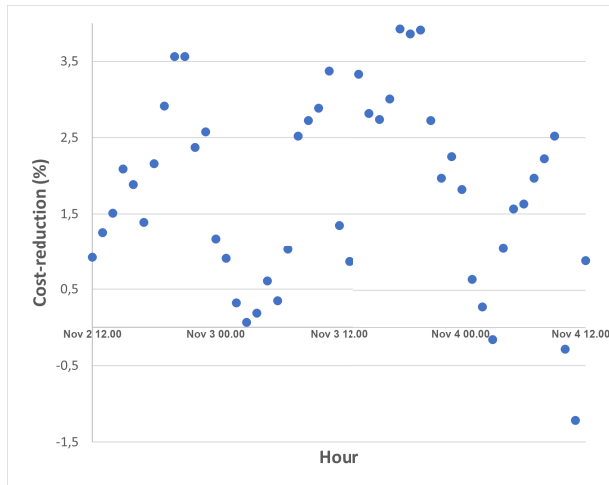


FIGURE 15. Cost saving when using the best CMA-ES candidate solution plan instead of the expert operator plan.

1. How can we use simulation and optimization algorithms to minimize the heating energy production cost of a district heating network with several production plants?

Firstly, a model that represents the network, including its pipes, consumers and producers, is needed. The model should take into account delays inherent to the network, caused by the potentially hundreds of kilometres of pipes that transport heated water. Secondly, an optimization algorithm that can simulate this model is needed, so that the optimization algorithm can learn from the model. Evolutionary algorithms can be used as the optimization algorithm, in which case the problem needs to be described as a fitness function. This fitness function should be designed in such a way that it minimizes the cost for operating the plant. In a district heating network, this can be done by modifying the supply temperature (i.e. the outgoing water temperature) of producers: A lower temperature reduces costs. However, a balance must be found, as the supply temperature needs to be kept high enough so that the heat demands of the consumers is met. Further, if a heat storage tank exists in the network, the fitness function should be rewarded for charging the tank when fuel prices are low and discharging it when fuel prices are high.

2. What kind of ICT architecture can support the use of a digital twin as the fitness function of an evolutionary algorithm that optimizes a large-scale industrial process?

An evolutionary algorithm needs to be able to utilize parallel computing when evaluating the fitness function of the digital twin model, due to the computational complexity and slow simulation speeds associated with a model of a large-scale industrial process. Further, a digital twin of this model must track the state of the real industrial process, i.e. it must receive real measurements from sensors in the physical system, in real-time. Copies of the digital twin model should be deployed in parallel in such a manner that the evolutionary algorithm can use them for fitness function evaluation. Additionally, the state of the digital twin model must be exportable so that the parallel copies of the digital

twin can be initialized with the correct state of the real physical system, at the beginning of each hour.

3. What are the trade-offs between computational complexity and fidelity, and how can the computational challenges be mitigated with parallel computing and cloud computing?

As computational complexity increases the time it takes to evaluate a single fitness function also increases. A high-fidelity model of an industrial process will be computationally complex, but provide the most accurate results compared to the real process. As fidelity is lowered, complexity is reduced and fitness function evaluation is faster. A trade-off must be done between the detail of the model and the speed at which the model can be simulated. Computational complexity can also occur due to the need to evaluate multiple fitness functions at the same time. Parallel computing can further be utilized to take advantage of additional resources beyond a single server and cloud computing allows these resources to be bought on-demand. Further, cloud computing allows each fitness function evaluation to be performed on their own virtual machine. As with anything in the cloud, however, the noisy neighbour problem should be taken into account, and further work on this is discussed in section VII.

As with anything in the cloud, however, the noisy neighbour problem should be taken into account.

4. How does the decision to use an evolutionary computation impact the architecture and computational challenges?

In evolutionary algorithms there is the concept of a population. Each individual in the population has a different fitness value, which is typically evaluated at the same time for the entire population. Thus, a parallel computing architecture benefits the evolutionary algorithm greatly, especially if the fitness function evaluation is time-consuming.

5. How does the proposed architecture and approach for handling the trade-offs perform in a case study?

The architecture presented in this paper performs well on the case study of the district heating network of Espoo. The network contains over 800 km of water pipes and over 14 heat producers, including CHP-plants, HOB-plants, heat pumps and waste heat facilities. A medium fidelity Apros 6 model was chosen over a high-fidelity model, in order to speed up fitness function evaluation. In this medium fidelity model, only the 14 biggest heat producers were modelled. In addition, closely located consumers and pipes were combined to simplify the network model. 36 parallel Apros 6 instances were used by a CMA-ES optimization algorithm, connected to an OpenFaaS instance. During the chosen 1-hour time-window, an average of 18 generations could be evaluated by CMA-ES and operating costs were reduced on average by 2%, compared to industrial state-of-the-art.

The total operating costs were calculated by running two operating plans on the medium fidelity Apros 6 model of the network, the expert operator plan and the optimized plan. These costs were compared and form the basis of the cost-saving results discussed in this paper. To reach this result, 36 virtual machines running 36 Apros 6 optimization instances was used by the CMA-ES algorithm. A few

additional virtual machines were used for running the digital twin Apros 6 instances and the rest of the software components, including the Python CMA-ES implementation. Separate virtual machines were needed for each Apros 6 instance due to performance reasons, in order to reach sufficient simulation speeds. As mentioned before, we cannot disclose the real operating costs of the case-study network in this paper due to confidentiality. However, we can disclose that reducing the operating costs of the district heating network by an average of 2% is a significant cost-saving reduction compared to the costs of the required virtual machines from EC2 on-demand Windows instances on AWS.

VII. CONCLUSION

In the paper, we have presented an architecture and approach for utilizing the up-to-date state of a digital twin, represented by a process simulation model, as the starting point for optimization computations. The architecture allows an evolutionary algorithm, in our case study CMA-ES, to leveraging parallel computations and cloud computing for the evaluation of fitness functions. A function-as-a-service approach connects the evolutionary algorithm's requests with the calculation capacity of multiple parallel process simulation nodes, which perform the fitness function evaluation. The case study presented in this paper shows that this approach is effective for optimizing a large-scale industrial process.

In daily use, observed between Nov. 2nd 2022 and Nov. 4th 2022, the optimized operating plan for the district heating network of Espoo reduces costs by an average of 2% and up-to 4%, compared to the expert operator plan. Reaching a 2% reduction in operating costs for the district heating network of Espoo significantly out-weights the infrastructure costs needed by the architecture described in this paper. These cost-savings were reached when both the optimized plan and the expert operator plan were simulated on the medium fidelity model and the simulation results were compared. The optimized plan would have to be executed on the real district heating network to translate into actual monetary savings, which in the end is the final decision of the network operators.

In a world with increasing energy prices and higher demands, optimizing the utilization of the available energy becomes increasingly important. The presented architecture and approach could be utilized for the optimization of other types of large industrial processes, not only district heating networks. The key benefit of this approach is that parallel computation allows optimizations to be performed on cases that would otherwise be infeasible to optimize. One emerging category of applications for the apparatus developed in this paper could be demand response capable industrial processes. Such processes would reschedule their operations so that energy intensive actions would occur during electricity market intervals with low prices.

Future work includes the elimination of mitigation of the noisy neighbour problem. Virtual machines hosted on platforms such as AWS are subject to the noisy-neighbour problem. Further, the cost-to-performance for on-demand

virtual machines, especially Windows Server types, makes them an expensive choice. Since the CMA-ES generation always waits for the slowest instance, a single cloud instance experiencing a noisy-neighbour will impact the performance of the whole CMA-ES algorithm. To eliminate the noisy neighbour problem, an on-premise dedicated cluster should be considered, instead of virtual machines from a public cloud-provider. Given only the removal of the noisy neighbour problem, as visualized in Fig. 10, all fitness function evaluations could be reduced in less than 170 seconds. This change alone would increase the number of generations the CMA-ES algorithm would have time to do from 18 to 21, without using more powerful hardware.

Future work is also required for industrial deployment of the proposed optimization method. The optimization results obtained in this paper assumes that the data measurements received from the real district heating network are reliable and accurate. In practice, due to the nature of a daily production system, faults can occur in system that causes the incoming data to be faulted or missing. Further, our results also assume that the medium fidelity model behaves as the real physical system, meaning that an operating plan simulated with the model would give similar cost-savings in the real process. During live operation, steps were taken to mitigate these challenges: Data outside accepted min/max boundaries, and data that was not been received in a certain time-period, cause the optimization result to be flagged as unreliable. Additionally, the validity of the medium fidelity model was checked continuously by the digital twin instance with KPIs, which compare the simulated results of the tracking digital twin model with the real measurements of the network. Discrepancies outside acceptable limits flagged the optimization result as unreliable. Still, there are many questions that need to be answered:

1. How to know if an optimized operating plans is actually sensible and reliable, when applied to the real network?
2. How to know if the digital twin model and the forecasts used by the optimization algorithm, match reality accurately enough so that decisions based on them can be relied on?
3. How to know if received real measurements are reliable and intact?

ACKNOWLEDGMENT

The authors would like to thank Fortum for giving their permission to write a paper about their Espoo district heating network, also would like to thank Tuomas Kallio and Sami Tuuri for their efforts developing and maintaining the Apros 6 district heating network model used by the optimization, also would like to thank Jani Simomaa, Henrik Milan, and Tuukka Lehtonen for their help implementing the various software components that were developed as part of this work, and also would like to thank Fortum's Project Manager Juha Kuronen, who helped with the overall vision of the project and its scope.

The funders had no involvement in study design; in the collection, analysis and interpretation of data; in the writing

of the report; and in the decision to submit the article for publication.

AUTHOR CONTRIBUTION

Miro Eklund: conceptualization, software, methodology, visualization, investigation, writing—original draft. Seppo Sierla: visualization, writing—review and editing, conceptualization. Timo Korvola: software, methodology. Hannu Niemistö: software, methodology, investigation. Jouni Savolainen: software, investigation. Tommi Karhela: supervision, project administration, and funding acquisition.

REFERENCES

- [1] B. Ghahremani, M. Bitaraf, and H. Rahami, "A fast-convergent approach for damage assessment using CMA-ES optimization algorithm and modal parameters," *J. Civil Struct. Health Monit.*, vol. 10, no. 3, pp. 497–511, Jul. 2020, doi: [10.1007/s13349-020-00397-1](https://doi.org/10.1007/s13349-020-00397-1).
- [2] S.-E.-K. Fateen, A. Bonilla-Petriciolet, and G. P. Rangaiah, "Evaluation of covariance matrix adaptation evolution strategy, shuffled complex evolution and firefly algorithms for phase stability, phase equilibrium and chemical equilibrium problems," *Chem. Eng. Res. Des.*, vol. 90, no. 12, pp. 2051–2071, Dec. 2012, doi: [10.1016/j.cherd.2012.04.011](https://doi.org/10.1016/j.cherd.2012.04.011).
- [3] M. W. Iruthayarajan and S. Baskar, "Covariance matrix adaptation evolution strategy based design of centralized PID controller," *Expert Syst. Appl.*, vol. 37, no. 8, pp. 5775–5781, Aug. 2010, doi: [10.1016/j.eswa.2010.02.031](https://doi.org/10.1016/j.eswa.2010.02.031).
- [4] A. Hill, J. Laneurit, R. Lenain, and E. Lucet, "Online gain setting method for path tracking using CMA-ES: Application to off-road mobile robot control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 7697–7702, doi: [10.1109/IROS45743.2020.9340830](https://doi.org/10.1109/IROS45743.2020.9340830).
- [5] S. S. Reddy, B. K. Panigrahi, R. Kundu, R. Mukherjee, and S. Debchoudhury, "Energy and spinning reserve scheduling for a wind-thermal power system using CMA-ES with mean learning technique," *Int. J. Electr. Power Energy Syst.*, vol. 53, pp. 113–122, Dec. 2013, doi: [10.1016/j.jepes.2013.03.032](https://doi.org/10.1016/j.jepes.2013.03.032).
- [6] Z. Li, X. Wang, W. Du, M. Yang, Z. Li, and P. Liao, "Data-driven adaptive predictive control of hydrocracking process using a covariance matrix adaption evolution strategy," *Control Eng. Pract.*, vol. 125, Aug. 2022, Art. no. 105222, doi: [10.1016/j.conengprac.2022.105222](https://doi.org/10.1016/j.conengprac.2022.105222).
- [7] R. Aringhieri and F. Malucelli, "Optimal operations management and network planning of a district heating system with a combined heat and power plant," *Ann. Oper. Res.*, vol. 120, nos. 1–4, pp. 173–199, 2003, doi: [10.1023/A:1023334615090](https://doi.org/10.1023/A:1023334615090).
- [8] D. Quaggiotto, J. Vivian, and A. Zarrella, "Management of a district heating network using model predictive control with and without thermal storage," *Optim. Eng.*, vol. 22, no. 3, pp. 1897–1919, Sep. 2021, doi: [10.1007/s11081-021-09644-w](https://doi.org/10.1007/s11081-021-09644-w).
- [9] R. Krug, V. Mehrmann, and M. Schmidt, "Nonlinear optimization of district heating networks," *Optim. Eng.*, vol. 22, no. 2, pp. 783–819, Jun. 2021, doi: [10.1007/s11081-020-09549-0](https://doi.org/10.1007/s11081-020-09549-0).
- [10] J. Lappalainen, "Extending mechanistic thermal-hydraulic modelling and dynamic simulation for new industrial applications," Doctoral dissertation, Dept. Chem. Eng., Aalto Univ., Espoo, Finland, 2019. [Online]. Available: <http://urn.fi/URN:ISBN:978-952-60-8759-7>
- [11] J. Yu, Y. Song, D. Tang, and J. Dai, "A digital twin approach based on nonparametric Bayesian network for complex system health monitoring," *J. Manuf. Syst.*, vol. 58, pp. 293–304, Jan. 2021, doi: [10.1016/j.jmsy.2020.07.005](https://doi.org/10.1016/j.jmsy.2020.07.005).
- [12] Q. Wang, W. Jiao, and Y. Zhang, "Deep learning-empowered digital twin for visualized weld joint growth monitoring and penetration control," *J. Manuf. Syst.*, vol. 57, pp. 429–439, Oct. 2020, doi: [10.1016/j.jmsy.2020.10.002](https://doi.org/10.1016/j.jmsy.2020.10.002).
- [13] T. Moi, A. Cibicik, and T. Rølvåg, "Digital twin based condition monitoring of a knuckle boom crane: An experimental study," *Eng. Failure Anal.*, vol. 112, May 2020, Art. no. 104517, doi: [10.1016/j.engfailanal.2020.104517](https://doi.org/10.1016/j.engfailanal.2020.104517).
- [14] D. Mourtzis, T. Togiias, J. Angelopoulos, and P. Stavropoulos, "A digital twin architecture for monitoring and optimization of fused deposition modeling processes," *Proc. CIRP*, vol. 103, pp. 97–102, Jan. 2021, doi: [10.1016/j.procir.2021.10.015](https://doi.org/10.1016/j.procir.2021.10.015).
- [15] Y. Li, Y. Rezgui, and H. Zhu, "Dynamic simulation of heat losses in a district heating system: A case study in Wales," in *Proc. IEEE Smart Energy Grid Eng. (SEGE)*, Aug. 2016, pp. 273–277, doi: [10.1109/SEGE.2016.7589537](https://doi.org/10.1109/SEGE.2016.7589537).
- [16] S. Syri, H. Mäkelä, S. Rinne, and N. Wirgentius, "Open district heating for Espoo city with marginal cost based pricing," in *Proc. 12th Int. Conf. Eur. Energy Market (EEM)*, May 2015, pp. 1–5, doi: [10.1109/EEM.2015.7216654](https://doi.org/10.1109/EEM.2015.7216654).
- [17] J. Yang, N. Zhang, A. Botterud, and C. Kang, "On an equivalent representation of the dynamics in district heating networks for combined electricity-heat operation," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 560–570, Jan. 2020, doi: [10.1109/TPWRS.2019.2935748](https://doi.org/10.1109/TPWRS.2019.2935748).
- [18] J. Dorfner and T. Hamacher, "Large-scale district heating network optimization," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1884–1891, Jul. 2014, doi: [10.1109/TSG.2013.2295856](https://doi.org/10.1109/TSG.2013.2295856).
- [19] C. Delmastro, G. Mutani, and L. Schranz, "The evaluation of buildings energy consumption and the optimization of district heating networks: A GIS-based model," *Int. J. Energy Environ. Eng.*, vol. 7, no. 3, pp. 343–351, Sep. 2016, doi: [10.1007/s40095-015-0161-5](https://doi.org/10.1007/s40095-015-0161-5).
- [20] S.-J. Zhou, M.-C. Tian, Y.-E. Zhao, and M.-G. Guo, "Dynamic modeling of thermal conditions for hot-water district-heating networks," *J. Hydrodyn.*, vol. 26, pp. 531–537, Aug. 2014, doi: [10.1016/S1001-6058\(14\)60060-3](https://doi.org/10.1016/S1001-6058(14)60060-3).
- [21] G. Rudolph, "Convergence of evolutionary algorithms in general search spaces," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 50–54, doi: [10.1109/ICEC.1996.542332](https://doi.org/10.1109/ICEC.1996.542332).
- [22] T. Back and H.-P. Schwefel, "Evolutionary computation: An overview," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 20–29, doi: [10.1109/ICEC.1996.542329](https://doi.org/10.1109/ICEC.1996.542329).
- [23] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*. Berlin, Germany: Springer, 2006, pp. 75–102, doi: [10.1007/3-540-32494-1_4](https://doi.org/10.1007/3-540-32494-1_4).
- [24] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 312–317, doi: [10.1109/ICEC.1996.542381](https://doi.org/10.1109/ICEC.1996.542381).
- [25] D. Molina, A. LaTorre, and F. Herrera, "An insight into bio-inspired and evolutionary algorithms for global optimization: Review, analysis, and lessons learnt over a decade of competitions," *Cogn. Comput.*, vol. 10, no. 4, pp. 517–544, Aug. 2018, doi: [10.1007/s12559-018-9554-0](https://doi.org/10.1007/s12559-018-9554-0).
- [26] K. Varelas, A. Auger, D. Brockhoff, N. Hansen, O. A. ElHara, Y. Semet, R. Kassab, and F. Barbaresco, "A comparative study of large-scale variants of CMA-ES," in *Proc. Int. Conf. Parallel Problem Solving Nature*, Coimbra, Portugal, 2018, pp. 3–15, doi: [10.1007/978-3-319-99253-2_1](https://doi.org/10.1007/978-3-319-99253-2_1).
- [27] C. Yang, J. Ding, Y. Jin, C. Wang, and T. Chai, "Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1046–1057, Jul. 2019, doi: [10.1109/TASE.2018.2865593](https://doi.org/10.1109/TASE.2018.2865593).
- [28] G. Yu, T. Chai, and X. Luo, "Multiobjective production planning optimization using hybrid evolutionary algorithms for mineral processing," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 487–514, Aug. 2011, doi: [10.1109/TEVC.2010.2073472](https://doi.org/10.1109/TEVC.2010.2073472).
- [29] P. Loganthurai, V. Rajasekaran, and K. Gnanambal, "Evolutionary algorithm based optimum scheduling of processing units in Rice industry to reduce peak demand," *Energy*, vol. 107, pp. 419–430, Jul. 2016, doi: [10.1016/j.energy.2016.04.027](https://doi.org/10.1016/j.energy.2016.04.027).
- [30] X. Wu and Z. Cao, "An improved multi-objective evolutionary algorithm based on decomposition for solving re-entrant hybrid flow shop scheduling problem with batch processing machines," *Comput. Ind. Eng.*, vol. 169, Jul. 2022, Art. no. 108236, doi: [10.1016/j.cie.2022.108236](https://doi.org/10.1016/j.cie.2022.108236).
- [31] O. R. Castro, R. Santana, J. A. Lozano, and A. Pozo, "Combining CMA-ES and MOEA/DD for many-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1451–1458, doi: [10.1109/CEC.2017.7969474](https://doi.org/10.1109/CEC.2017.7969474).
- [32] E. BouDaher and A. Hoorfar, "Electromagnetic optimization using mixed-parameter and multiobjective covariance matrix adaptation evolution strategy," *IEEE Trans. Antennas Propag.*, vol. 63, no. 4, pp. 1712–1724, Apr. 2015, doi: [10.1109/TAP.2015.2398116](https://doi.org/10.1109/TAP.2015.2398116).

- [33] S. Limmer and D. Fey, "Investigation of strategies for an increasing population size in multi-objective CMA-ES," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 476–483, doi: [10.1109/CEC.2016.7743832](https://doi.org/10.1109/CEC.2016.7743832).
- [34] S. M. Bazaz, M. Lohtander, and J. Varis, "5-dimensional definition for a manufacturing digital twin," *Proc. Manuf.*, vol. 38, pp. 1705–1712, Jan. 2019, doi: [10.1016/j.promfg.2020.01.107](https://doi.org/10.1016/j.promfg.2020.01.107).
- [35] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Adv. Model. Simul. Eng. Sci.*, vol. 7, p. 13, Mar. 2022, doi: [10.1186/s40323-020-00147-4](https://doi.org/10.1186/s40323-020-00147-4).
- [36] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decis. Support Syst.*, vol. 145, Jun. 2021, Art. no. 113524, doi: [10.1016/j.dss.2021.113524](https://doi.org/10.1016/j.dss.2021.113524).
- [37] M. Dalibor, N. Jansen, B. Rumpe, L. Wachtmeister, and A. Wortmann, "Model-driven systems engineering for virtual product design," in *Proc. ACM/IEEE 22nd Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Sep. 2019, pp. 431–436, doi: [10.1109/MODELS-C.2019.00069](https://doi.org/10.1109/MODELS-C.2019.00069).
- [38] Z. Lei, H. Zhou, W. Hu, G.-P. Liu, S. Guan, and X. Feng, "Toward a web-based digital twin thermal power plant," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1716–1725, Mar. 2022, doi: [10.1109/TII.2021.3086149](https://doi.org/10.1109/TII.2021.3086149).
- [39] G. S. Martínez, "Simulation-based digital twins of industrial process plants: A semi-automatic implementation approach," Doctoral dissertation, Dept. Elect. Eng. Automat., Aalto Univ., Espoo, Finland, 2019. [Online]. Available: <http://urn.fi/URN:ISBN:978-952-60-8537-1>
- [40] W. D. Chen, H. M. Hasanien, and K. J. Chua, "Towards a digital twin approach—Experimental analysis and energy optimization of a multi-bed adsorption system," *Energy Convers. Manage.*, vol. 271, Nov. 2022, Art. no. 116346, doi: [10.1016/j.enconman.2022.116346](https://doi.org/10.1016/j.enconman.2022.116346).
- [41] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, and A. Calinescu, "Digital twins: State of the art theory and practice, challenges, and open research questions," *J. Ind. Inf. Integr.*, vol. 30, Nov. 2022, Art. no. 100383, doi: [10.1016/j.jii.2022.100383](https://doi.org/10.1016/j.jii.2022.100383).
- [42] Y. Wang, X. Kang, and Z. Chen, "A survey of digital twin techniques in smart manufacturing and management of energy applications," *Green Energy Intell. Transp.*, vol. 1, no. 2, Sep. 2022, Art. no. 100014, doi: [10.1016/j.geits.2022.100014](https://doi.org/10.1016/j.geits.2022.100014).
- [43] C. Alcaraz and J. Lopez, "Digital twin: A comprehensive survey of security threats," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1475–1503, 3rd Quart., 2022, doi: [10.1109/COMST.2022.3171465](https://doi.org/10.1109/COMST.2022.3171465).
- [44] S. S. Kamble, A. Gunasekaran, H. Parekh, V. Mani, A. Belhadi, and R. Sharma, "Digital twin for sustainable manufacturing supply chains: Current trends, future perspectives, and an implementation framework," *Technol. Forecasting Social Change*, vol. 176, Mar. 2022, Art. no. 121448, doi: [10.1016/j.techfore.2021.121448](https://doi.org/10.1016/j.techfore.2021.121448).
- [45] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Y. C. Nee, "Enabling technologies and tools for digital twin," *J. Manuf. Syst.*, vol. 58, pp. 3–21, Jan. 2021, doi: [10.1016/j.jmsy.2019.10.001](https://doi.org/10.1016/j.jmsy.2019.10.001).
- [46] F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji, "Digital twin modeling," *J. Manuf. Syst.*, vol. 64, pp. 372–389, Jul. 2022, doi: [10.1016/j.jmsy.2022.06.015](https://doi.org/10.1016/j.jmsy.2022.06.015).
- [47] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018, doi: [10.1016/j.ifacol.2018.08.474](https://doi.org/10.1016/j.ifacol.2018.08.474).
- [48] J. Bönsch, M. Elstermann, A. Kimmig, and J. Ovtcharova, "A subject-oriented reference model for digital twins," *Comput. Ind. Eng.*, vol. 172, Oct. 2022, Art. no. 108556, doi: [10.1016/j.cie.2022.108556](https://doi.org/10.1016/j.cie.2022.108556).
- [49] S. Zeb, A. Mahmood, S. A. Hassan, M. J. Piran, M. Gidlund, and M. Guizani, "Industrial digital twins at the Nexus of NextG wireless networks and computational intelligence: A survey," *J. Netw. Comput. Appl.*, vol. 200, Apr. 2022, Art. no. 103309, doi: [10.1016/j.jnca.2021.103309](https://doi.org/10.1016/j.jnca.2021.103309).
- [50] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017, doi: [10.1109/ACCESS.2017.2657006](https://doi.org/10.1109/ACCESS.2017.2657006).
- [51] H. Dang, M. Tatipamula, and H. X. Nguyen, "Cloud-based digital twinning for structural health monitoring using deep learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 3820–3830, Jun. 2022, doi: [10.1109/TII.2021.3115119](https://doi.org/10.1109/TII.2021.3115119).
- [52] T. Hoebert, W. Lepuschitz, E. List, and M. Merdan, "Cloud-based digital twin for industrial robotics," in *Int. Conf. Ind. Appl. Holonic Multi-Agent Syst.*, in Lecture Notes in Computer Science, vol. 11710, 2019, pp. 105–116, doi: [10.1007/978-3-030-27878-6_9](https://doi.org/10.1007/978-3-030-27878-6_9).
- [53] W. Hofmann and F. Branding, "Implementation of an IoT- and cloud-based digital twin for real-time decision support in port operations," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2104–2109, 2019, doi: [10.1016/j.ifacol.2019.11.516](https://doi.org/10.1016/j.ifacol.2019.11.516).
- [54] H. Pargmann, D. Euhansen, and R. Faber, "Intelligent big data processing for wind farm monitoring and analysis based on cloud-technologies and digital twins: A quantitative approach," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2018, pp. 233–237, doi: [10.1109/ICCCBDA.2018.8386518](https://doi.org/10.1109/ICCCBDA.2018.8386518).
- [55] H. Zhou, C. Yang, and Y. Sun, "Intelligent ironmaking optimization service on a cloud computing platform by digital twin," *Engineering*, vol. 7, no. 9, pp. 1274–1281, Sep. 2021, doi: [10.1016/j.eng.2021.04.022](https://doi.org/10.1016/j.eng.2021.04.022).
- [56] V. Yussupov, J. Soldani, U. Breitenbücher, A. Brogi, and F. Leymann, "FaaS: your decisions: A classification framework and technology review of function-as-a-service platforms," *J. Syst. Softw.*, vol. 175, May 2021, Art. no. 110906, doi: [10.1016/j.jss.2021.110906](https://doi.org/10.1016/j.jss.2021.110906).
- [57] D. Barcelona-Pons and P. García-López, "Benchmarking parallelism in FaaS platforms," *Future Gener. Comput. Syst.*, vol. 124, pp. 268–284, Nov. 2021, doi: [10.1016/j.future.2021.06.005](https://doi.org/10.1016/j.future.2021.06.005).
- [58] M. Sewak and S. Singh, "Winning in the era of serverless computing and function as a service," in *Proc. 3rd Int. Conf. Conver. Technol. (IC2T)*, Apr. 2018, pp. 1–5, doi: [10.1109/IC2T.2018.8529465](https://doi.org/10.1109/IC2T.2018.8529465).
- [59] P. Leitner, E. Wittern, J. Spillner, and W. Hummer, "A mixed-method empirical study of function-as-a-service software development in industrial practice," *J. Syst. Softw.*, vol. 149, pp. 340–359, Mar. 2019, doi: [10.1016/j.jss.2018.12.013](https://doi.org/10.1016/j.jss.2018.12.013).
- [60] A. U. Gias and G. Casale, "COCOA: Cold start aware capacity planning for function-as-a-service platforms," in *Proc. 28th Int. Symp. Modeling, Anal., Simulation Comput. Telecommun. Syst. (MASCOTS)*, Nov. 2020, pp. 1–8, doi: [10.1109/MASCOTS50786.2020.9285966](https://doi.org/10.1109/MASCOTS50786.2020.9285966).
- [61] J. Shen, T. Yang, Y. Su, Y. Zhou, and M. R. Lyu, "Defuse: A dependency-guided function scheduler to mitigate cold starts on FaaS platforms," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2021, pp. 194–204, doi: [10.1109/ICDCS51616.2021.00027](https://doi.org/10.1109/ICDCS51616.2021.00027).
- [62] Z. Li, Q. Zhang, X. Lin, and H.-L. Zhen, "Fast covariance matrix adaptation for large-scale black-box optimization," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2073–2083, May 2020, doi: [10.1109/TCYB.2018.2877641](https://doi.org/10.1109/TCYB.2018.2877641).
- [63] N. Hansen, Y. Akimoto, and P. Baudis, "CMA-ES/pycma on Github," Zenodo, Tech. Rep., Feb. 2019. [Online]. Available: <https://github.com/CMA-ES/pycma>, doi: [10.5281/zenodo.2559634](https://doi.org/10.5281/zenodo.2559634).
- [64] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the Brownian motion," *Phys. Rev.*, vol. 36, no. 5, pp. 823–841, Sep. 1930, doi: [10.1103/PhysRev.36.823](https://doi.org/10.1103/PhysRev.36.823).
- [65] Y. A. Kutoyants and J. A. Kutojanc, *Statistical Inference for Ergodic Diffusion Processes*. London, U.K.: Springer-Verlag, 2004.
- [66] E. Silvennoinen, K. Juslin, M. Hänninen, O. Tiihonen, J. Kurki, and K. Porkholm, "The APROS software for process simulation and model development," VTT Tech. Res. Centre Finland, Espoo, Finland, Res. Rep. 618, 1989. [Online]. Available: <https://cris.vtt.fi/en/publications/the-apros-software-for-process-simulation-and-model-development>
- [67] S. Patankar, *Numerical Heat Transfer and Fluid Flow*, 1st ed. Boca Raton, FL, USA: CRC Press, 1980, doi: [10.1201/9781482234213](https://doi.org/10.1201/9781482234213).
- [68] L. Tomás, C. Vázquez, J. Tordsson, and G. Moreno, "Reducing noisy-neighbor impact with a fuzzy affinity-aware scheduler," in *Proc. Int. Conf. Cloud Autonomic Comput.*, Sep. 2015, pp. 33–44, doi: [10.1109/ICAC.2015.14](https://doi.org/10.1109/ICAC.2015.14).
- [69] W. Lloyd, S. Pallickara, O. David, M. Arabi, and K. Rojas, "Mitigating resource contention and heterogeneity in public clouds for scientific modeling services," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, Apr. 2017, pp. 159–166, doi: [10.1109/IC2E.2017.29](https://doi.org/10.1109/IC2E.2017.29).



MIRO EKLUND was born in Finland, in 1993. He received the M.Sc. degree in computer engineering from Åbo Akademi University, in 2018. He is currently pursuing the Ph.D. degree in researching digital twins from the Department of Information Technology, Åbo Akademi University, in 2020.

Since 2016, he has been working with Semantum Ltd. Since 2020, he has been at Semantum. His areas of expertise include cloud-computing architectures, software development, and modeling and simulation tools. He has been working on various projects in the modeling and simulation domain, since 2015.



SEPPO A. SIERLA (Senior Member, IEEE) was born in 1977. He received the M.Sc. degree in technology with the major in embedded systems and the D.Sc. degree in technology from the Helsinki University of Technology, in 2003 and 2007, respectively, and the Docent degree in software design for industrial automation from the School of Electrical Engineering, Aalto University, Finland, in 2013.

From 2003 to 2011, he was a Research Scientist with Aalto University and the Helsinki University of Technology. Since 2011, he has been a University Lecturer with Aalto University. His research interests include applications of simulation, machine learning, and ICT technologies to the energy sector.

Dr. Sierla was the Vice Chair of the IEEE Finland Section (2022–2023) and before that, he was the Section Treasurer (2020–2021). Since 2020, he has been the Co-Chair of the “Industrial Digitalization, Digital Twins in Industrial Applications” Track in the IEEE INDIN Conference Series. He is currently the Finance Chair of several IEEE conferences INDIN 2019, MLSP 2020, ISGT 2021 Europe, ISIT 2022, and ISIE 2023.



HANNU NIEMISTÖ was born in 1980. He received the Ph.D. degree in mathematics, especially in finite model theory from the University of Helsinki, in 2007. After receiving his Ph.D. degree, he continued as a Research Scientist with VTT. His research interests include simulation, formal methods, co-simulation, and industrial information management. He moved to Semantum, in 2016, and has worked on projects involving optimization, simulation, and data transformations.



TIMO KORVOLA received the M.Sc. degree in mathematics and computer science from the Helsinki University of Technology (now part of Aalto University), in 1994, and the Ph.D. degree in mathematical physics from the University of Helsinki, in 2003. His thesis was about the Cahn-Hilliard equation. Since 2005, he has been working with VTT as a Research Scientist in projects involving mathematical modeling, optimization, and simulation.



JOUNI SAVOLAINEN received the M.Sc. (Tech.) degree in chemical engineering and mathematics from the Helsinki University of Technology and the D.Sc. (Tech.) degree from the Department of Electrical Engineering and Automation, Aalto University, in 2019.

He worked as a Senior Scientist with the VTT Technical Research Centre of Finland. He works as a Chief Modeling and Simulation Specialist with Semantum Ltd. He has worked with several industrial domains e.g., energy, forest, and marine industries. His professional and academic research interests include dynamic process simulation and its applications, combining it with other mathematical methods.



TOMMI A. KARHELA received the Doctor of Science degree in technology from the Helsinki University of Technology, in 2002.

He worked for ten years as a Research Professor in computer simulation technology with the Technical Research Centre of Finland—VTT, until 2018. He had also a parallel position as a Professor of Practice with Aalto University for five years (2013–2018). He is currently working as a Partner and the Chairperson of the Board at a start-up software company at Semantum specialized in engineering automation and simulation-based digital twins. He has also a parallel position as Visiting Professor with Aalto University. He has received the VTT Award, in 2010, for exceptional performance and the Finnish Automation Award, in 2011, for the work with the Simulation Integration Environment—Simantics.

...