Caíno-Lores, S., García, A., García-Carballeira, F., Carretero, J. (2016). Efficient Design Assessment in the Railway Electric Infrastructure Domain Using Cloud Computing. *Integrated Computer-Aided Engineering,* 24(1), pp. 57-72

DOI: 10.3233/ICA-160532

Efficient Design Assessment in the Railway Electric Infrastructure Domain using Cloud Computing

Silvina Caíno-Lores, Alberto García, Félix García-Carballeira and Jesús Carretero
Computer Science and Engineering Department,
University Carlos III of Madrid, Av. Universidad 30, 28911 Leganés, Spain

Abstract. Nowadays, the modelling, design, evaluation and testing stages involved in the development of railway infrastructures are extensively assisted by computer simulators. Moreover, some expert systems take a step further to improve and propose designs, taking the user's knowledge as a baseline. These systems can generate and assess a large number of complex scenarios, which yields the execution of numerous, and potentially very complex simulations. Railway infrastructures rely heavily on these applications to analyze potential deployments prior to their installation. In this paper, we propose the *railway power consumption simulator* model (RPCS), a cloud-based model for the design, simulation and evaluation of railway electric infrastructures. This model integrates the parameters of an infrastructure within a search engine that generates and evaluates a set of simulations to achieve optimal designs, according to a given set of objectives and restrictions. The knowledge of the domain is represented as an ontology, which translates the elements in the infrastructure into an electric circuit, which is simulated to obtain a wide range of metrics in each element of the infrastructure. In order to support the execution of thousands of scenarios in a scalable, efficient and fault-tolerant manner, we also propose an architecture to deploy the model in a Cloud environment. To illustrate how our model would adapt to a specific problem, we describe a case study that aims to maximize energy savings, while maintaining a high power provisioning quality. Using our model, we were able to obtain the optimal substation distribution that allowed the infrastructure to operate under normal and faulty conditions. Additionally, we include the economic costs that arose from the externalization of the computations to the Amazon Elastic Compute Cloud, which were minimized by our dimensioning model and the usage of spot instances.

Keywords: Railway electric provisioning, infrastructure optimization, simulation model, cloud computing

1. Introduction

Power dimensioning and energy saving have been traditionally two main issues regarding the deployment of electrical grids. Since their conception in the Industrial Age, power grids are designed and deployed following a trade-off between supporting high quality provisioning to the consumers, and saving as much energy as possible. Railway electric lines, as a particular case of electric grids, are also concerned about these issues, trying to supply a steady flow of energy to the moving trains, but not exceeding the power required by them.

Within this context, simulators have been the main tools to design and test railway electric lines. Prior to its installation, a particular deployment can be tested on a simulator, modelling the infrastructure and the train traffic in order to check the behavior of the system. Simulators like the ones introduced in (1) and (2) are able to analyze in a per-instant basis whether the power supplied to the trains is enough of not, if there are voltage drops or overvoltages, etc.

Nevertheless, as computer systems evolve, the role of simulators must evolve from merely imitators of the real-world, to expert systems with the ability to make decisions and complement the user knowledge with metrics in order to achieve the best solutions. In previous works (3), we stated that modern simulators should be capable of proposing and evaluating new designs, taking into account all possible issues that may affect, or even determine, the final validity of a solution. This search across the problem domain is driven by expert's knowledge implemented within the simulator, in the form of generation and evaluation rules that may reduce the number of simulations performed, or give those solutions a score indicating their fitness according to specific criteria, and a search algorithm that leads further evolution of selected solutions in order to improve them.

However, as the number of scenarios to be simulated increases, so does the computing resources required to perform the whole search. Exploring the search space of the problem in several dimensions (e.g. sweeping different parameters of the simulation) leads to an explosive number of simulations that have to be performed. In this context, Cloud Computing raises as an option to achieve virtual on-demand resources, with minimal management effort. In fact, Cloud Computing brings the opportunity to tailor the hardware resources according to specific user needs or simulation characteristics, whereas the pay-per-use model frees the user from the burden of maintaining the infrastructure once the simulation has been performed. All things considered, outsourcing the infrastructure to Cloud Service Providers (CSP) is a meaningful decision when there is a high variation in the number of simulations, or in the resources needed to execute them.

In a previous work (4) we introduced the core elements of the simulation model described in this paper. With this baseline, the aim of this paper is to present a full cloud-based simulation model for the railway energy provisioning domain. This extended version of the model supports the simulation, evaluation and optimization of railway infrastructure designs according to a series of user-defined metrics, restrictions and optimization criteria. In order to cope with the numerous simulations that may result from this process, and their inherently massive computational requirements, this model takes advantage of the elasticity of cloud infrastructures. For the purpose of demonstrating the capabilities of this model, a case study is conducted by stating a multi-objective optimization (MMO) problem. The results will show the solutions achieved, the search process, and the economic costs billed by the cloud service provider (CSP) after the whole process is completed.

The paper is structured as follows: Sec. 2 discusses relevant publications related to the topics in this work; Sec. 3 introduces the main aspects of the proposed model, including the ontology used to represent the railway domain, the simulator's structure, and the elements that constitute the search engine; Sec. 4 describes a use case that particularizes the application of this model, along with the formalization of this case as an MOO problem; Sec. 5 exposes the cloud-based architecture that reinforces the scalability and parallelism of the simulation model; Sec. 6 describes the evaluation conducted and the results obtained, specifying the target infrastructure and its impact on the economic cost of the execution; finally, Sec. 7 provides key ideas as conclusions and some insight in future work.

2. Related work

The research community has been aware of the need for optimal planning of power distribution systems as a whole (5). In particular, many of the relevant works in the field are focused on providing a near-optimal solution in a computationally efficient manner. To achieve this, different artificial intelligence (AI) techniques have served as a baseline for the implementation of the aforementioned decision making process, such as particle swarms (6), genetic algorithms (7,8), ant colonies (9), simulated annealing (10), artificial neural networks (11), multi-agent systems (12,13), and evolutionary algorithms (14). These methodologies provide a holistic approach in which the simulator proposes consistent, well-suited solutions to a particular problem.

Ontologies have been traditionally used to model and represent the simulated domain in expert systems. The authors themselves hold broad experience in this topic and in railway-related domains, as shown by previous works that support the need for ontology-driven decision support systems (15,16). Nevertheless, given the wide variety of sub-domains within the railway domain, it is hard to find an ontology which includes all the required elements and contemplates all the facets of a particular engineering problem. (17) represents a first attempt to join experts and create an ontology of railway objects, which was addressed as a great challenge. (18) presented a methodology to develop and infer railway ontologies, but the ontology described as case study was far from the electric problem. Dedicated ontologies for railway power dimensioning are not easy to find, as most of them are focused on scheduling and operation, like the one introduced by (19). (20) proposed in their work an ontology that includes, but is not focused on, the railway power dimensioning problem, leaving relevant aspects aside.

It may occur that the optimization process depends on different conflicting criteria, leading to an MOO problem. Works like (21–25) approach the design of simulation models from an MOO perspective, which allows to define several optimization metrics such as minimization of power losses, overall deployment cost, system failure index, or maximization of energy savings, etc. This approach has been also translated to the field of railway power supply systems, especially along with the previously cited evolutionary techniques. For instance, (26,27) consider a trade-off between failure recovery and load sharing is exposed and tackled as an MOO problem.

Nowadays, many scientific areas make use of the Cloud Computing paradigm to overcome scalability issues in simulations, and increase their performance. In particular, computing frameworks, like MapReduce (28,29), have been increasingly used as building-blocks for distributed large scale simulators in a wide range of areas (30,31). Railway simulators have also been affected by this trend, integrating MapReduce and Cloud environments to existing techniques with promising results in large datasets and scenarios (32). Finally, (33,34) demonstrate the economic feasibility of migrating scientific or engineering simulations to the Cloud, even though making use of Cloud resources entails paying for such resources to the CSP.

From the related works researched by the authors, only a few stay close to the present work, in terms of usefulness and capabilities. (11) proposes a fast approximation based on neural networks in order to plan power supply investments. On the contrary, our approach is independent from the underlying AI techniques, so different search strategies can be implemented with minimal modifications. (12) implements an agent-based smart power router, which can flexibly integrate network areas and optimally manage power flows. Nevertheless, this approach is outside the railway domain, so it does not take into account the particular railway domain characteristics. (1) and (25) both propose an optimization problem for AC railway power systems, with well-developed and consistent models. These models, however, do not consider as many details regarding the infrastructure as our model does. Despite (35) follows an approach similar than ours by presenting an integrated framework for simulation and solution evaluation for scenario optimization, it is not oriented towards infrastructure design, but train scheduling. Finally, neither of these proposals is based on Cloud Computing, nor they can make use of elastic computing infrastructures according to simulation sizes and deadlines.

3. The RPCS model

The objective of this model is to support the assessment of potential infrastructures in order to develop new routes, increase train traffic across the tracks, or test failure situations where services have to be operated in degraded mode. RPCS is able to model a problem in order to simulate, evaluate and optimize a set of potential solutions for the problem of designing and deploying electric infrastructure on railway lines, according to a given optimization criteria and restrictions. In this section we describe in detail the main elements that compose the model: an ontology of the railway electric infrastructure domain, a simulation kernel, and a search engine. Figure 1 shows an overview of the model, including these elements and their relations. The ontology drives a translation of real infrastructure components into elements of an electric circuit: voltage sources, branches, and consumers (current sources). This information is used by the simulation kernel, which uses the knowledge of the railway infrastructure and its elements (tracks, feeders, electrical substations and trains) to build an electric circuit, which is solved afterwards. Finally, the search engine generates and evaluates solutions varying a set of parameters, performing the search across the solution space to meet user-defined restrictions and objectives. Further information on the domain's ontology, the simulation kernel, and the architecture of the search engine is provided in the following subsections.

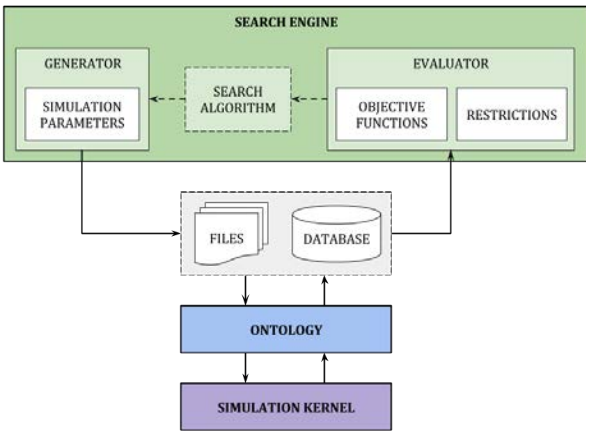

Figure 1: RPCS model high-level architecture.

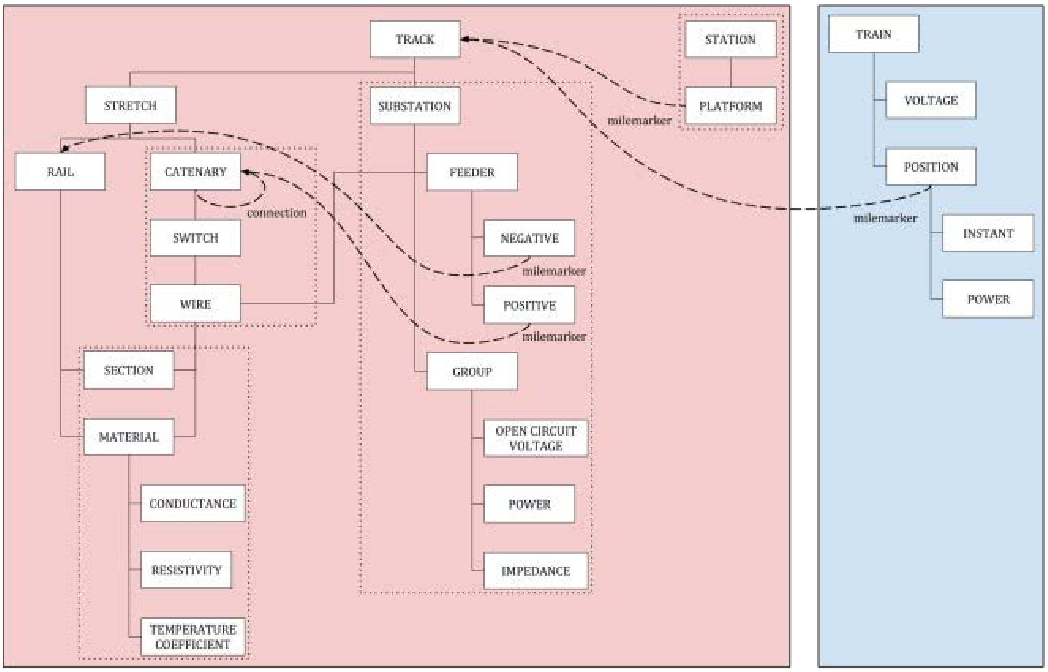

Figure 2: Ontology for the railway power provisioning domain considered in RPCS.

3.1. Ontology

The main role of the domain ontology is to propose a taxonomy to translate real infrastructure elements into an electric circuit. This electric circuit should reflect the actual behavior of the system (i.e. trains, tracks, electrical substations) accurately, in order to approach the results of the analysis as close to reality as possible. The main reason behind the need for an ontology is that it brings the formalization required to interact with the problem domain outside the simulation kernel, helping to set out optimization problems, defining heuristics and restrictions, and performing searches across the problem space using different algorithms.

The proposed ontology represents the domain of a railway installation containing a set of railway stations linked by tracks at a specific milemarker. A number of trains circulate with a pre-defined profile along these stations, and through the tracks. These trains consume electric energy during their operation, and this energy is provided by electric substations placed strategically within the infrastructure. The electricity is transported from the electric stations to the trains by means of catenaries deployed over the tracks, and returns to the substations through the rails. As a whole, these elements conform an electric circuit that changes constantly over time, due to the train movement and consumption variations.

The entities and their relationships are represented in Fig. 2. This ontology is designed to distinguish the elements that belong to the infrastructure, and the mobile elements that use it (i.e. trains). In this context, trains are defined by their electric properties, as well as the profile of its run. This profile is constituted by a collection of records that relate the power consumption of the train with a specific instant and position, expressed as a milemarker of a track. More specifically, we can define the profile of a specific train as indicated by Eq. 1. This profile, $\wp$, is defined as a set of 3-tuples, where t is the instant in the simulated time in which we know the position of the train on the track, m, and its instantaneous power consumption, P.

$$\wp = \{(t, m, P): t \in T, m \in M, P \in \mathbb{R}^+\} \qquad (1)$$

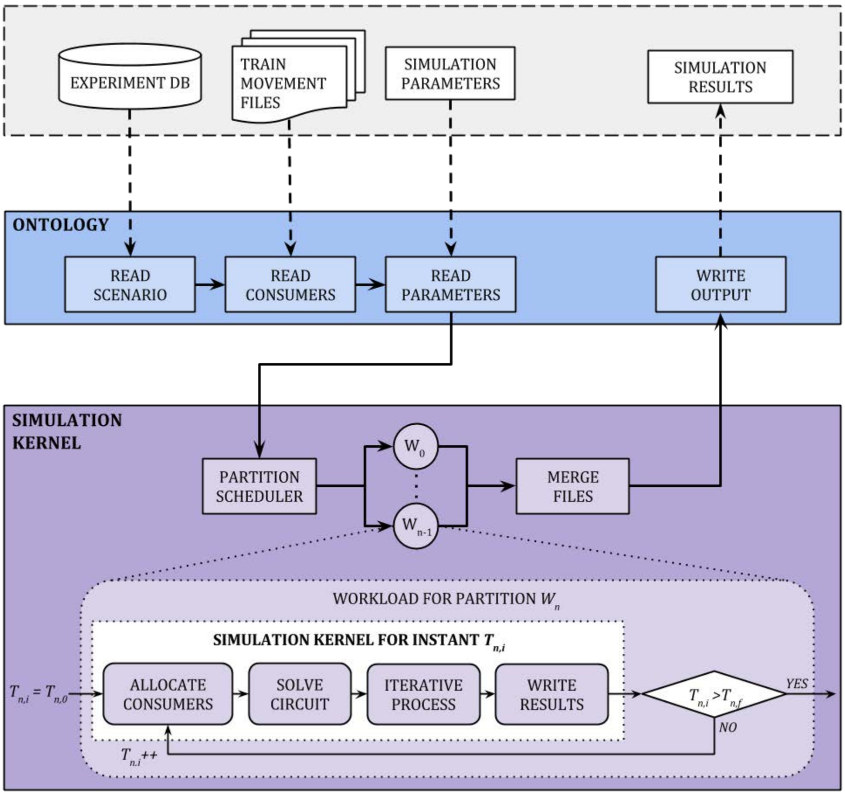

Figure 3: Simulation process scheme.

3.2. Simulation kernel

The aim of the simulation kernel is, provided a number of trains circulating across the line(s), to calculate the electric status of the infrastructure in the form of voltages and currents in the system. Subsequently, this permits to assess whether the amount of power supplied by the electrical substations is sufficient to allow the trains to render without delays, failures, or any other contingency.

The main algorithm that composes the electric circuit requires the definition of a set of electrically independent tracks, voltage sources, and ground connections that link the track to the electric substation. Trains run across the track demanding power at the time and point marked by their profile, thus acting as consumers in a circuit. These entities are translated by the ontology to characterize the resulting electric circuit with its nodes, branches, and voltage/current sources. Subsequent circuit analysis is conducted to obtain data that reflects the state of the circuit and all its components. These results are translated back in order to map the outcome of the analysis to the physical entities of the domain. A summary of this procedure can be seen in Fig. 3.

Note that, due to the fact that the trains are in movement, the system is constantly changing, so every instant the electric circuit must be composed and calculated, varying the position of the consumers. As consequence, modified nodal analysis (MNA) (36) must be performed on every simulated instant, thus requiring a significant amount of computing power to perform the whole simulation. The MNA general formulation can be simplified in this domain considering branches as resistors, and independent voltage sources only. This results in Eq. 2, where G, B and C are matrices of known values obtained from the circuit elements (connections, conductances, etc.), u^n and i^r are the unknown voltages and currents, and, finally, i and e contain the sum of the currents through the passive elements, and the values of the independent voltage sources, respectively.

$$\begin{bmatrix} G & B \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} u_n \\ i_r \end{bmatrix} = \begin{bmatrix} i \\ e \end{bmatrix} \qquad (2)$$

The larger the simulated times and installation, and the larger the larger the number of elements to be simulated (lines, trains, power stations, etc.), the bigger are the MNA matrices that need to be solved.

This simulation kernel relies on two types of input data:

– Shared infrastructure specification data, containing the initial and final time of the simulation, plus a wide range of domain-specific simulation parameters such as station and railway specifications and power supply definition.
– A set of train movements data, structured in a time-based manner, in which each register contains the calculation of speed and distance profiles for a particular train at a specific instant regarding the infrastructure constraints, and most important, the instantaneous power demand, with a one second interval.

3.3. Multi-objective optimization engine

In the previous section we have described how the ontology and simulation kernel can be used to simulate a scenario with several tracks, trains, electric substations, etc. Nevertheless, as we stated before, modern simulators should be capable of proposing and evaluating new designs taking into account possible issues that may affect the final validity of a solution. In order to do so, we broaden the RPCS model turning towards a generic MOO problem that can be tailored to meet user-defined constraints and objectives, supporting experiment generation, evaluation and solution improvement.

The proposed MOO engine consists of the following elements:

– An *experiment generator* that models the parameters that will vary to generate the pool of scenarios to be simulated. This permits to execute not one, but many simulations, each one constituting a variation of the input data –either the infrastructure or the trains–.
– A *solution evaluator* that models how the results of each of the former experiments are going to be assessed. Its main objective is to detect the solutions that meet given constraints and objective functions, introduced in the form of an MOO problem.
– A *search algorithm* in charge of guiding the selection of the optimal solutions. This could also include evolutionary and iterative techniques to improve solutions in further generation-execution-evaluation cycles.

These elements wrap the simulation kernel and the ontology. In this way, the model supports the definition of different search problems by only defining the generation and evaluation procedures, plus choosing the algorithm to perform the search.

Nevertheless, setting out search problems typically entails the execution of not one, but many simulations, thus requiring a high amount of computing resources. The inherent complexity of the simulation process is factored by the number of simulations to be executed, which is also dependent on the selected search algorithm and optimization metrics. For instance, a single-objective search aiming to maximize the overall provisioning quality would require much less simulations than a bi-criterion optimization that also minimizes power losses.

In order to set the basis for the rest of the paper, the next section will describe in detail a case study following our model.

4. Case study: optimizing energy provisioning on railway lines

At this point we focus on the modelization of the trade-off between energy saving and quality of energy provisioning. As trains circulate along the tracks demanding power, voltage oscillations may arise all across the electric circuit, leading to voltage drops or over-voltages. These situations should be avoided, maintaining a constant flow of electric power to the trains. Note that trains do not always consume the same amount of power, and even more, they can return power to the circuit due to regenerative braking technologies. The quality of the power supply refers to the concept of maintaining the system as close to the nominal voltage as possible. While voltage drops can be avoided by adding more electric substations on the tracks, the fluctuating energy consumption in the trains might lead to an excessive voltage in the system, especially if the regenerative braking mechanism is operative. Besides, the more substations to be placed, the more expensive the deployment is, and the more aggregated energy is consumed by the electric substations. This leads to conflicting objectives, thus to a MOO problem: the goal of maintaining a constant power flow, in favor of providing more energy, against the target benefit of saving energy.

Our search problem will be to find the corresponding Pareto frontier of the MOO problem. To evaluate the generated scenarios, we propose a set of restriction rules that must be fulfilled by the design in order to be considered as acceptable, and set of optimization metrics in order to score those accepted solutions. Both sets are obtained by analyzing the European regulations (37–39).

4.1. Problem formalization

As previously described, there are two objectives that guide the optimization process:

- Improving the quality of the power supply.
- Reducing the amount of power consumed by the groups.

We define from these goals the following criteria:

- Maximizing the mean useful voltage per train, O_1.
- Minimizing total amount of energy consumed by the groups, O_2.

As described in European normative UNE-EN-50388 (39), for direct current installations the mean useful voltage is defined as the mean of all voltages at the pantograph of each train in the geographic zone, along all simulation steps. This measure indicates the quality of the power supply. The lower the mean useful voltage is, the less energy is transferred from the supply stations to the trains, on average.

For the formalization of this problem, let *T* be the set of trains in the whole system, and *G* be the set of groups in the network. The first objective is defined in Eq. 3, where U^t_{mu} is the mean useful voltage per train, and U_{max1} constitutes the maximum permanent voltage.

$$\mathbf{max}\, O_1 = \frac{U^t_{mu} -,\ 800}{U_{max1} -,\ 800} \quad \forall t \in T \qquad (3)$$

The second objective is formulated in Eq. 4, where E^i_g is the energy consumed per group, in *kW/h*.

$$\mathbf{min}\, O_{,} = \sum_{i=1}^{G} E^i_g \quad i \neq g, \forall g \in G \qquad (4)$$

The problem is subject to the following constraints:

- According to the normative (39), the mean useful voltage per train, U^t_{mu}, must never be lower than *2800V*, and it shall not surpass the maximum permanent voltage, U_{max1}.

$$2\square00 \leq U^t_{mu} \leq U_{max1} \qquad (5)$$

- No sharp voltage drops or over-voltages shall exist on normal (i.e. non-failure) operating conditions (37). Therefore, instantaneous voltages should be in the range of non-permanent conditions on every instant of the simulation. This derives Eq. 6a and Eq. 6b.

$$U_{min1} \leq U_t \leq U_{max,} \quad \forall t \in T \qquad (6a)$$
$$U_{min1} \leq U_g \leq U_{max,} \quad \forall g \in G \qquad (6b)$$

The mean voltages on trains and the simulated zone, shall be within the limits of permanent operating conditions, even if voltages fall beyond that limits for a moment during the simulation [2, 11]. This yields Eq. 7a and Eq. 7b.

$$U_{min1} \leq U^t_{mu} \leq U_{max,} \quad \forall t \in T \qquad (7a)$$
$$U_{min1} \leq U_{muz} \leq U_{max1} \qquad (7b)$$

4.2. Experiment generation rules

We selected as benchmark a standard railway scenario described in the European normative EN-50641 (38). This proposal of normative establishes the requirements for the validation of simulation tools used for the design of traction power supply systems. Therefore, it is meaningful to apply such normative to assess the capabilities of a model for a railway power infrastructure simulator.

The problem's search space will be generated by simulating a wide set of experiments that take as baseline the former CENELEC test case, and differ in the placement of the electrical substations along the tracks –i.e. the con-

Table 1: Variations of electrical substations placement for the MOO optimization.

Electrical substation	E_1	E_2	E_3
Milemarkers (km) (initial, final, Δ)	(0, 20, 1)	(20, 40, 1)	(40, 50, 1)

nection milemarker of the substation to the track–. By modifying the location of the substations, we vary the electric circuit, thus we obtain different measures of instantaneous and mean voltages, as well as the consumed potency. Therefore, substation placement has a direct impact on the power supply quality and energy savings. Furthermore, we want to assess fault tolerance in the base design, so we also need to consider the situation in which a substation is unavailable.

4.2.1. Base simulation scenarios

First we need to determine the generation rules for the scenarios with all of the substations available. For each substation, E_k, the initial and final points of the interval in which they can be placed must be defined –E_{kini} and E_{kfin}, respectively–, along with the distance between each planned position for the genera tion of the experiment set, Δ_k. Each substation can be assigned to any of the points within the $[E_{kini}, E_{kfin}]$ interval, leading to the number of base scenarios indicated in Eq. 8, where M is the number of substations to be manipulated.

$$N_{base} = \prod_{k=1}^{M} \left(\frac{E_{kfin} - E_{kini}}{\Delta_k} \right) \qquad (8)$$

This definition of the parameters involved in our target experiments leads to N_{base} different base scenarios. Each of them constitute an independent experiment.

4.2.2. Fault-tolerance assessment

In order to assess fault tolerance, we can eliminate one of the substations at a time in the designs that result from the rules in Sec. 4.2.1. Therefore, we obtain a set of faulty scenarios, each of them with a single substation missing. As these faulty cases might lead to scenarios that do not permit proper operation, we can also consider extended scenarios that included an additional group to increase the capacity of the system.

Given the former, we take into account the possible combinations of $M - 1$ substations, with and without the added group in each of the remaining substations. This yields that, for every base scenario, we would obtain as many scenarios as indicated in Eq. 9.

$$N_{fault} = M \cdot \binom{M}{M-1} + 1 \qquad (9)$$

From these equations we can see that, the finer the grain of the planned experiments, and the larger the number of substations, the more simulations have to be executed in order to generate the solution space. Considering that the number of experiments to be simulated grows exponentially with the number of substations in the worst case, as indicated by Eq. 10, the overall computing resources required to generate the solution space of the MOO would quickly outscale those typically available in current desktop computers.

$$N = N_{base} \cdot N_{fault} \qquad (10)$$

For instance, considering the previous case study, we can generate a set of base scenarios using the variations of the positions indicated in Tab. 1, displacing each substation from one kilometer to the next, without overlapping their ranges. Following Eq. 10, this leads to an overall number of 1040 experiments to be executed. Since each experiment is composed of 4,800 simulation steps –one per simulated instant, corresponding to 1h and 20m of simulated time–, it would be required to solve 4,800 equation systems like the one introduced in Eq. 2, per experiment. Therefore the total workload would yield the resolution of 4,992,000 equation systems. The potentially huge experiment pool we can generate is the main reason behind the cloud-oriented architecture presented in the following section.

5. Cloud-based design and deployment

Cloud Computing paradigm brings us several features that can be useful in the context of a simulation framework:

- Virtual unlimited scalability of resources. Execution is not tied to the local infrastructure, and more computing power may be allocated on-demand.
- Flexibility by means of adapting computing resources. We can allocate more or less computing power depending on the size of the simulation, the deadline for obtaining the results, and the available budget to conduct the experimentation.

Nevertheless, in order to take advantage of the Cloud Computing paradigm, it is essential to propose an efficient way to distribute the workload among multiple nodes. Otherwise, as the simulation problem increases in size, we would not be able to scale up the infrastructure by adding more nodes. Therefore, the problem has been decomposed at two levels of abstraction to overcome this issue:

- *Simulation concurrency*. The way some search problems are performed yields an implicit opportunity for parallelization, provided that some of the simulations can be conducted independently, with no dependences between them
- *Domain decomposition*. RPCS target simulations consist of a period of simulated time. Within this period, each instant must be represented as an electric circuit to be solved. Because the train positions and their consumption are known all along the simulated time, there is no dependency between one instant and the following, so multiple instants can be solved in parallel.

In our approach both levels of concurrency are exploited to maximize scalability and performance.

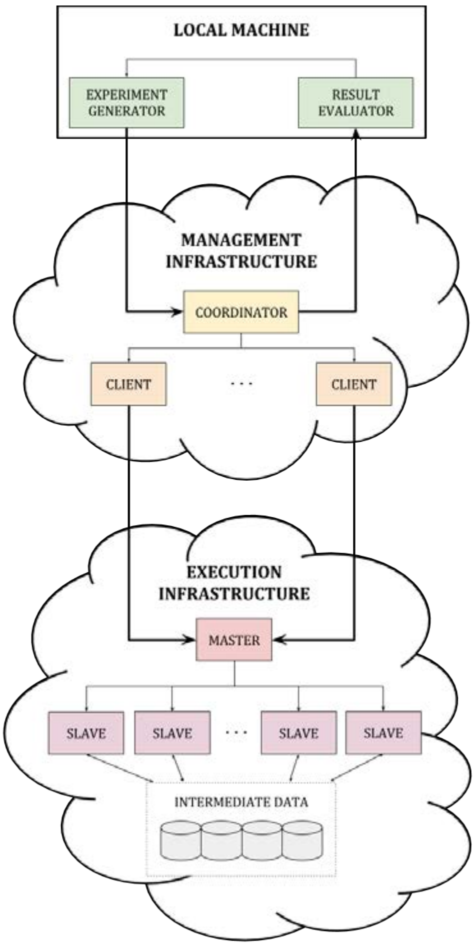

Figure 4: Cloud deployment architecture for the RPCS model.

The simulation domain -i.e. the simulated time– is distributed between the Cloud nodes. This distribution is conducted by dividing the simulated period in a number of independent instants, simulating each instant independently through the simulation kernel, and finally merging the results. Furthermore, multiple simulations are conducted in parallel. All concurrent simulations are distributed in the same way, thus in the virtual cluster instants from different simulations are scheduled together, leading to an efficient resource usage.

From this perspective, we can consider that the whole simulation set –which we will refer to as *experiment pool* in the next paragraphs– can be partitioned into smaller subsets (*experiment partitions*) that can be distributed in the infrastructure. Regarding the domain decomposition, each experiment is composed of a series of independent *jobs*, which are composed of *tasks* that handle specific simulation instants. The methodology used to perform this transformation is described in (40), as well as all the evaluation performed in order to assure scalability when dealing with a large number of experiments.

5.1. Architecture definition

The high-level design of the architecture to adapt the model to the cloud environment is shown in Fig. 4. There are two main master-worker schemes within this architecture: one in charge of interacting with the search engine and managing the experiment pool resulting from the generation process; and another that is responsible for the

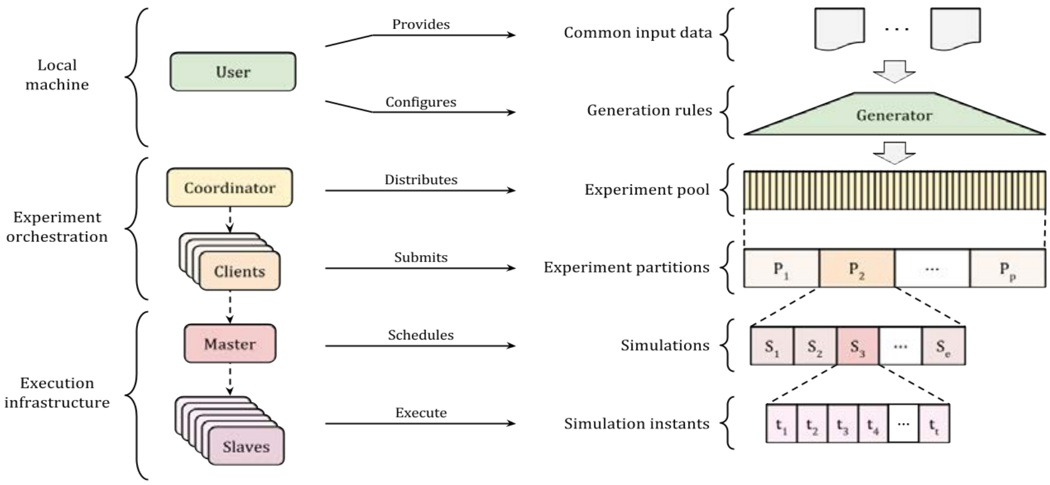

Figure 5: Summary of infrastructure entities and their assigned role.

execution of the tasks associated with each experiment. The former, which we will refer to as the *experiment orchestration subsystem*, is composed of the following elements:

- The *coordinator* is an entity that distributes the experiment pool generated by the experiment generator of the search engine. This large set is divided into smaller subsets, resulting in several experiment partitions that will be delegated to other nodes. Besides experiment distribution, the coordinator receives the results of every experiment, so it is able to execute the evaluation and search stages of the search engine.
- There are one or more *client* nodes, each of them being assigned with one or more experiment partitions, so that they can concurrently request the execution of every element in this subset simultaneously.

Once the experiment pool is properly distributed, each client requests the execution of its experiments to the *execution infrastructure*, which is constituted by the following entities:

- A *master* node entitled to the orchestration of the set of simulated instants that compound the execution of a distributed experiment.
- One or more *slaves* that run the simulations assigned by the master, reporting to it both their progress and the resulting data.

Figure 5 shows the hierarchy of the former entities and their relation to the experiments within the RPCS model. As we have seen, the experiment generator holds the necessary information to generate the experiment pool, which is partitioned and distributed by the coordinator according to its knowledge of the underlying infrastructure. Afterwards, each client manages its experiment subset by requesting it to be run in the execution infrastructure. Therefore, the management infrastructure acts as a link between the search engine and the execution infrastructure, in which the clients link the scheduled experiments to the nodes that perform the executions.

5.2. Cloud infrastructure dimensioning model

Once we have clearly described the cloud-based RPCS architecture design, we need to consider the amount of resources that are necessary to accomplish a successful execution. Having an accurate estimation of the kind and number of nodes that we need to meet a deadline or minimize the total execution time is very interesting to estimate the amount resources and their cos on on-demand environments like clouds.

We provide guidelines that could help to dimension the size of the infrastructure –i.e. the number and type of cloud instances– that is required for a particular use case, starting from a profile of a sample experiment. This model permits to optimize the performance, the operational cost of the whole process, or a trade-off between them. This estimation is performed for the two sub-infrastructures described in the previous section: coordinator-clients and master-slaves.

5.2.1. Definitions and assumptions

First of all, we define the concept of a task. A task is a single instant of the simulated time of one of the experiments which are part of the experiment pool. The input data associated to a task is the smallest autonomous piece of data which can be processed by an execution of the simulation kernel. Next, we consider the following assumptions:

1. All the slaves shall be equal in terms of memory and number of cores. The same occurs with the clients.
2. The sets of instance types for the slaves, *I*, and the clients, *C*, are finite, and may overlap.
3. The execution time required to process a task, $t_e \in \mathbb{R}^+$, is known and homogeneous for all the tasks. This is the time required to execute the simulation kernel and simulate a single instant of an experiment.
4. The amount of memory required to process a task, $m_e \in \mathbb{R}^+$, is known and homogeneous for all the tasks. This is the memory needed to allocate and simulate one single simulation kernel and merge the results.
5. The amount of memory required to submit and manage the map-reduce jobs of an experiment, $m_s \in \mathbb{R}^+$, is known. This value is usually platform-dependent, thus homogeneous for all experiments.
6. The total number of experiments (i.e. *submissions*), $n_s \in \mathbb{N}$, is known.
7. The number of tasks per submission, $n_{e\square} \in \mathbb{N}$, is known and homogeneous for every experiment.

Regarding the instance types, both sets are defined by the CSP. An instance in the set of slave types, *I*, is defined by its price per unit of time, p_I, its amount of memory, m_I, and its number of cores, c_I. An instance in the set of client types, *C*, is also defined by its price, p_C, and its amount of memory, m_C.

5.2.2. Balancing the infrastructure

The first consideration that must be reflected in the model is the need for balance among the different elements of the architecture. We first need to define a series of key concepts:

1. Let $N_e \in \mathbb{N}$ be the number of *schedulable tasks* within the whole cluster, this is, the amount of tasks we can concurrently execute in an infrastructure while preserving a maximum degree of parallelism among them. A formalization of the concept is introduced in Eq. 11, where $n_i \in \mathbb{N}$ is the number of instances in the targeted cluster.
$$N_e = n_i \min\left\{\left\lfloor\frac{\sum_i \delta_i m_I}{m_e}\right\rfloor, \sum_i \delta_i c_I\right\} \quad (11)$$
Additionally, in Eq. 10 we introduce an auxiliary variable, $\delta_i \in \{0,1\}$, which indicates whether the instance type $i \in I$ is selected or not. Since we are considering that all the slaves belong to the same instance type, we force δ_i to be one only once by introducing the restriction in Eq. 12.
$$\sum_i \delta_i = 1 \quad (12)$$
2. Let $N_s \in \mathbb{N}$ be the number of *schedulable submitters* within the whole cluster, this is, the amount of experiments we can spawn among all the client machines simultaneously. Equation 13 represents this idea, in which $n_c \in \mathbb{N}$ is the number of clients in the management infrastructure.
$$N_s = n_c \left\lfloor\frac{\sum_j \gamma_j m_C}{m_s}\right\rfloor \quad (1\square)$$
Here we also need an auxiliary variable, $\gamma_j \in \{0,1\}$, to control the selection or not of each client type $j \in C$. Equation 14 also restricts γ_j to select only one instance type.
$$\sum_j \gamma_j = 1 \quad (14)$$
3. Let $n_e \in \mathbb{N}$ be the *total number of entries* to be executed for the whole experiment set, which is computed as follows:
$$n_e = n_{e\square} n_s \quad (15)$$

Notice that $n_e \geq N_e$ and $n_s \geq N_s$, hence N_e might not equal $n_{e\square} N_s$. In particular, we can derive three different cases according to the relation between N_e and N_s:

– If $N_e > n_{e\boxtimes} N_s$, we are in a situation in which we can execute more tasks than the ones we are able to submit, hence the execution infrastructure is underutilized and we are incurring in unnecessary costs for the slaves.
– If $N_e = n_{e\boxtimes} N_s$, the resources of the selected clients and slaves instances match perfectly the amount of submittable and executable tasks, leading to an optimal infrastructure in terms of balance.
– If $N_e < n_{e\boxtimes} N_s$, the clients are in the position of submitting too many tasks to the execution infrastructure because we can schedule too many experiments at once. In this circumstance, the execution infrastructure becomes overloaded, while we have too many resources in the management infrastructure.

As we have seen, the optimal balance between both infrastructures is attained when we can schedule as many tasks as we can submit. Despite this situation is not easy to reach, it is highly desirable to waste as little resources as possible in one of the subsystems. Therefore, we need to balance the resources of both subsystems. This leads to the constraint represented in Eq. 16, which states that we shall minimize the difference between the available execution slots and the schedulable tasks. In Eq. 16, $\tilde{N}_e$ is the number of tasks that we can execute taking into consideration the resource limitation in the submitter side.

Table 3: Problem definition.

Parameter	Value	Parameter	Value
t_e	1	n_s	1040
m_e	7168 MB	m_s	256 MB
n_e	4992000	α	0.5
$n_{e\boxtimes}$	4800	β	20

Table 4: Problem solution proposed by the model.

Variable	Value	Variable	Value
δ_0	0	γ_0	0
δ_1	1	γ_1	0
$\delta_,$	0	$\gamma_,$	1
δ_3	0	γ_3	0
n_i	400	n_c	1
$\left\lceil \frac{T}{u} \right\rceil$	9 h	C	886.41 $

$$\tilde{N}_e = \min\{N_e, n_{e\boxtimes} N_s\} \qquad (16)$$

Analogously, we can define Eq. 17, which computes $\tilde{N}_s$, the actual number of submitters we can schedule considering the resources in the execution infrastructure.

$$\tilde{N}_s = \left\lceil \frac{\tilde{N}_e}{n_{e\boxtimes}} \right\rceil \qquad (17)$$

5.2.3. Optimizing performance

One of the primary objectives of dimensioning the infrastructure is to select the proper resources to reduce the execution time. In order to understand the aspects that affect performance in this architecture we need to consider the platform overhead. Let $\alpha, \beta \in \mathbb{R}^+$ be two parameters that represent the compute overhead factor of the underlying platform for spawning processes, assigning resources and other managerial tasks, for the slaves and clients, respectively. Both parameters are considered constant, but could be refined to reflect platform-specific features.

Given the former, we can define the total execution time of an experiment set running on the cloud architecture as indicated by Eq. 18. Naturally, *the execution time is desired to be as low as possible*.

$$\min_{N_e, N_s} T = \frac{n_e}{\tilde{N}_e}(t_e + \alpha) + \frac{n_s}{\tilde{N}_s}\beta \qquad (18)$$

This minimization problem could be modified by letting T be a fixed value, in order to find suitable instances to meet a specific deadline. This deadline-oriented planning can be very beneficial to minimize costs in pay-as-you-go infrastructures. Therefore, this heuristic can be used to: (a) provide a VM configuration and calculate the expected execution time using that configuration, or (b) select the best VM configuration in order to meet a given deadline.

Table 2: Description of the selected EC2 instances

Instance	Type	Virtual CPUs	Memory (GB)	Storage (GB)	Price/h ($)	$i \in I$	$j \in C$
m3.large	Generalist	2	7.5	32 (SSD)	0.140	0	0
m2.2xlarge	Memory optimized	4	34.2	850	0.490	1	1
m2.xlarge	Memory optimized	2	17.1	420	0.245	2	2
c3.xlarge	Compute optimized	2	3.75	2x40 (SSD)	0.210	3	3

5.2.4. Minimizing the operational costs

In order to provide a mean to control economical costs and keep an execution on a budget, we include an additional objective in the cluster planning: *monetary costs must be minimized*. Assuming the prices for the slave and client instances types are known -- $p_I, p_C \in \mathbb{R}^+$, respectively--, we can compute the final cost of an execution as shown in Eq. 19, where $u \in \mathbb{R}^+$ constitutes the time unit in which the provider charges for the resources.

$$\min_{n_i, n_c, \delta_i, \gamma_j} C = \left\lceil \frac{T}{u} \right\rceil \left(n_i \sum_i \delta_i p_I + n_c \sum_j \gamma_j p_C \right) \quad (19)$$

6. Evaluation

6.1. Experimental setup

Table 5: Execution cost of the experiments using spot instances against the theoretical price.

	Time (hours)	Spot ($)	Full ($)	Savings (%)
Execution	9	66.436	903.420	92.646
Storage	-	44.896	44.896	0.000
Total		111.332	948.316	88.260

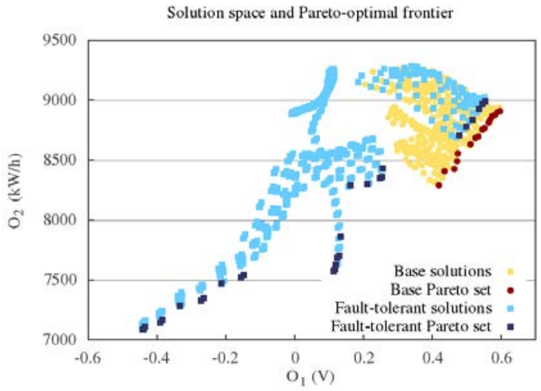

Figure 6: Solution space and Pareto-optimal frontier for the CENELEC experiment set, including base solutions and fault-tolerant solutions supporting voltage drops

We applied our cloud dimensioning model to a subset of the full Amazon EC2 instance offer. Table 2 describes the main characteristics of these instance classes, which constitute representatives of Amazon's instance flavors: with a focus on memory resources (*memory optimized*), computing power (*compute optimized*), or a balance between the former (*general purpose*). Considering our case study, we set the model's parameters as shown in Tab. 3. Following the guidelines of the optimization model, shown in Tab. 4, the resulting cloud infrastructure consisted of a general purpose *m3.large* node as dedicated master, four hundred *m2.xlarge* machines as slaves, and three *m2.2xlarge* client instances, each controlling a partition of the whole experiment pool, while one of them did also act as coordinator. Despite the model suggests that a single client would suffice, we replicated the client for fault-tolerance reasons, as we relied on spot instances to reduce the operational costs of the whole process. Additionally, only one master machine was considered, but this could be extended to implement backup mechanisms in order to further improve fault tolerance.

Spot instances are constituted by spare resources in the EC2 infrastructures, thus they are priced according to their availability, which varies over time. Additionally, their price is based on a bidding mechanism, in which users set the maximum price they are willing to pay for a specific instance type. The allocated resources are available to the user unless the instance price exceeds this bid. It is important to remark that we can take the risk of losing some spot instances due to the application's inherent fault tolerance. Because the implementation of the RPCS model is built on top of the Hadoop MapReduce computing platform (41), it inherits its features regarding task independence, replication and robustness. In particular, the platform's replication strategy forces input data to be present in different node s, so the whole computation is able to proceed in the case some of the nodes fail.

6.2. Results

Table 5 shows the amount that was charged for the overall execution, considering that the simulation took 8.43 hours, including the time spent during system setup. This configuration and start-up stage is tightly related to the number of nodes in the virtual cluster. This stage is arguably auxiliary, and its relevance in the final price could be diluted over time if the cluster is partially reused for subsequent experimentations. In any case, we are currently considering several improvements aimed towards reducing the complexity and time required for this process.

The results we obtained for the base and faulty scenarios were parsed and evaluated according to the metrics defined in Sec. 4.1. The Pareto-optimal frontiers for the former datasets are shown in Fig. 6, along with the other solutions that resulted from the subsequent simulations. The solutions that belong to the Pareto-optimal frontiers highlighted in Fig. 6 are the ones that meet the optimization criteria developed in Sec. 4.1. The final selection could balance the supply quality (O_1) and the wasted energy (O_2), or be directed towards emphasizing one of the optimization objectives. From the resulting data we are able to draw the following conclusions:

- *Base scenarios*. While any solution in the Pareto-optimal frontier of the base configurations meets the optimization criteria, we can highlight two particular cases: the ones that fully optimize each of the objectives. In order to maximize the provisioning quality, O_1, the substation configuration shall place, taking as reference the beginning of the track, E_1 at 16*km*, E_2 at 25*km*, and E_3 at 40*km*. To minimize the consumed power, O_2, the best configuration is the one that places E_1 at 4*km*, E_2 at 23*km*, and E_3 at 46*km* of the beginning of the track.
- *Fault-tolerant scenarios*. Among the fault-tolerant solutions, the one that maximizes O_1 situates E_2 at 36*km* of the beginning of the track, while E_3 is located at 40*km*. In this case, E_1 is the substation that could fail or be eliminated without leaving the infrastructure out of operation. Additionally, E_3 would require an additional group to provide a better quality in the power provisioning. The scenario that minimizes O_2 corresponds to a design that is able to operate with two substations –E_1 and E_3–, the first placed at the beginning of the track, and the second 40*km* away from the former. In Fig. 6 we can clearly see three different solution subsets, one of them partially overlapping the base solutions set. We are currently working towards analyzing the correlation between these results, in order to detect the relation between substation placement and the electric profile of this particular traffic configuration.

7. Conclusions

In this paper, we have presented RPCS, a model for the design, simulation and evaluation of electric railway infrastructure deployments. It supports the assessment of simulation scenarios with respect to case-specific restrictions and constraints, generating a whole set of experiments to evaluate, and optionally evolve, in order to provide near-optimal solutions to a wide range of problems.

The model is constituted by three core elements: an ontology, a simulator, and a multi-objective optimization engine. The first element translates the entities within the railway infrastructure into the items that constitute an electric circuit. The second supports the simulation of this circuit in a per instant basis by means of the MNA technique. Finally, the third element allows the definition of experiment generation guidelines to support the execution of multiple simulations concurrently. The results are assessed according to an MOO problem to find the solution set. This procedure can be iterated to evolve into subsequently better solutions.

Since the experiment generation procedure might lead to hundreds, or even thousands of scenarios to be simulated, scalability and elasticity of the model are key features by design. We achieve this by orienting the model towards cloud-based distributed infrastructures, which provide the possibility to scale up or down according to the experiment pool size. The model also supports scalability regarding the simulation size, as a single experiment can be run in multiple nodes to overcome resource limitations in standalone machines. Hence, we are able to achieve parallelism among different simulations, and within a single experiment, which results in very high scalability and resource usage efficiency. Additionally, we propose a dimensioning model to estimate the proper cluster size that would minimize cost, execution time, or a trade-off between them.

In order to show the capabilities of this model, we have illustrated the process of defining a search problem focused on the trade-off between saving energy while providing a high quality provisioning, applied to a standard railway scenario defined in an European normative. The evaluation was conducted on Amazon EC2, running a virtual cluster of hundreds of nodes executing the computing tasks. Given that it is a pay-per-use infrastructure, we aimed to minimize our operational costs by using spare spot instances, which are available at a reduced fare, but are subject to be terminated by the provider as prices vary. Remarkably, our model's fault tolerance capabilities allowed us to reduce the expenses drastically with no significant performance loss.

As future work, we intend to minimize the system setup and configuration time, which has a major impact in costs for large clusters. Currently the model considers only DC systems, but its extension to AC systems is now work in progress. Finally, we are investigating the possibility to expand the model to support efficient stream processing, so that the generation, execution and evaluation procedures could be conducted periodically to adjust elements of the infrastructure dynamically –such as train position and speed–, according to a specified problem.

Acknowledgements

This work has been partially funded under the grant TIN2013-41350-P of the Spanish Ministry of Economics and Competitiveness, and the COST Action IC1305 ”Network for Sustainable Ultrascale Computing Platforms” (NESUS).

References

1. Pilo E, Rouco R, Fernandez A, Hernandez-Velilla A. A simulation tool for the design of the electrical supply system of high-speed railway lines. In: IEEE Power Engineering Society Summer Meeting, 2000. 2000. p. 1053–8 vol. 2.

2. Sanz Bobi J de D, Jorreto Marcos F, Garzón Núñez J. A Simulation Tool for Sizing Electrical Railway Lines. 2007 Jan 1;269–77.

3. García A, Gómez C, García-Carballeira F, Carretero J. Enhancing the structure of railway infrastructure simulators. In: Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization (OPT-i). 2014. p. 352–63.

4. Carretero J, Caino S, Garcia-Carballeira F, Garcia A. A Multi-Objective Simulator for Optimal Power Dimensioning on Electric Railways using Cloud Computing: In SCITEPRESS - Science and and Technology Publications; 2015. p. 428–38.

5. Pilo E, Mazumder SK, Gonzalez-Franco I. Smart Electrical Infrastructure for AC-Fed Railways With Neutral Zones. IEEE Trans Intell Transp Syst. 2015 Apr;16(2):642–52.

6. del Valle Y, Venayagamoorthy GK, Mohagheghi S, Hernandez J-C, Harley RG. Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. IEEE Trans Evol Comput. 2008 Apr;12(2):171–95.

7. Ramirez-Rosado IJ, Bernal-Agustin JL. Genetic algorithms applied to the design of large power distribution systems. IEEE Trans Power Syst. 1998 May;13(2):696–703.

8. Carrano EG, Takahashi RHC, Cardoso EP, Saldanha RR, Neto OM. Optimal substation location and energy distribution network design using a hybrid GA-BFGS algorithm. Gener Transm Distrib IEE Proc-. 2005 Nov;152(6):919–26.

9. Gomez JF, Khodr HM, De Oliveira PM, Ocque L, Yusta JM, Villasana R, et al. Ant colony system algorithm for the planning of primary distribution circuits. IEEE Trans Power Syst. 2004 May;19(2):996–1004.

10. Parada V, Ferland JA, Arias M, Daniels K. Optimization of electrical distribution feeders using simulated annealing. IEEE Trans Power Deliv. 2004 Jul;19(3):1135–41.

11. Abrahamsson L, Soder L. Railway Power Supply Investment Decisions Considering the Voltage Drops - Assuming the Future Traffic to Be Known. In: 15th International Conference on Intelligent System Applications to Power Systems, 2009 ISAP ’09. 2009. p. 1–6.

12. Nguyen PH, Kling WL, Ribeiro PF. Smart Power Router: A Flexible Agent-Based Converter Interface in Active Distribution Networks. IEEE Trans Smart Grid. 2011 Sep;2(3):487–95.

13. Prýmek M, Horák A. Multi-agent Approach to Power Distribution Network Modelling. Integr Comput-Aided Eng. 2010 Dec;17(4):291–303.

14. Strbac G, Djapic P. A genetic based fuzzy approach to optimisation of electrical distribution networks. In: Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995 GALESIA First International Conference on (Conf Publ No 414). 1995. p. 194–9.

15. Saa R, Garcia A, Gomez C, Carretero J, Garcia-Carballeira F. An ontology-driven decision support system for high-performance and cost-optimized design of complex railway portal frames. Expert Syst Appl. 2012 Aug;39(10):8784–92.

16. Garcia A, Gomez C, Saa R, Garcia-Carballeria F, Carretero J. Optimizing the process of designing and calculating railway catenary support infrastructure using a high-productivity computational tool. Transp Res Part C Emerg Technol. 2013 Mar;28:1–14.

17. Bjørner D. Train: The railway domain. In: Building the Information Society [Internet]. Springer; 2004 [cited 2016 Jan 22]. p. 607–11.

18. Mohan A, Arumugan G. Constructing Railway Ontology using Web Ontology Language and Semantic Web Rule Language. Int J Comput Tech Appl. 2011;2(2):314–21.

19. Liu X, Du H, Liu N. Research on high-speed railway ontology integration method based on semantic relationships. In: 2011 International Symposium on IT in Medicine and Education (ITME). 2011. p. 295–8.

20. Masucci V, Adinolfi F, Servillo P, Dipoppa G, Tofani A. Ontology-Based Critical Infrastructure Modeling and Simulation. In: Palmer C, Shenoi S, editors. Critical Infrastructure Protection III. Springer Berlin Heidelberg; 2009. p. 229–42.

21. Augugliaro A, Dusonchet L, Favuzza S, Sanseverino ER. Voltage regulation and power losses minimization in automated distribution networks by an evolutionary multiobjective approach. IEEE Trans Power Syst. 2004 Aug;19(3):1516–27.

22. Carrano EG, Soares LAE, Takahashi RHC, Saldanha RR, Neto OM. Electric distribution network multiobjective design using a problem-specific genetic algorithm. IEEE Trans Power Deliv. 2006 Apr;21(2):995–1005.

23. Mendoza F, Bernal-Agustin JL, Dominguez-Navarro JA. NSGA and SPEA Applied to Multiobjective Design of Power Distribution Systems. IEEE Trans Power Syst. 2006 Nov;21(4):1938–45.

24. Ramirez-Rosado IJ, Dominguez-Navarro JA. Possibilistic model based on fuzzy sets for the multiobjective optimal planning of electric power distribution networks. IEEE Trans Power Syst. 2004 Nov;19(4):1801–10.

25. Soler M, Lopez J, Mera Sanchez de Pedro JM, Maroto J. Methodology for Multiobjective Optimization of the AC Railway Power Supply System. IEEE Trans Intell Transp Syst. 2015 Oct;16(5):2531–42.

26. Chang CS, Wang W, Liew AC, Wen FS, Srinivasan D. Genetic algorithm based bicriterion optimisation for traction substations in DC railway system. In: , IEEE International Conference on Evolutionary Computation, 1995. 1995. p. 11 – .

27. Chang CS, Wang W, Liew AC, Wen FS. Bicriterion optimisation for traction substations in mass rapid transit systems using genetic algorithm. Electr Power Appl IEE Proc -. 1998 Jan;145(1):49–56.

28. Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. Commun ACM. 2008 Jan;51(1):107–13.

29. Sun Z, Shen J, Young J. A novel approach to data deduplication over the engineering-oriented cloud systems. Integr Comput Aided Eng. 2013 Jan 1;45–57.

30. Radenski A. Using MapReduce Streaming for Distributed Life Simulation on the Cloud. In MIT Press; 2013 [cited 2016 Jan 22]. p. 284–91.

31. Kim B, Lee SJ, Kim TG, Song HS. Mapreduce based experimental frame for parallel and distributed simulation using hadoop platform. In: 28th European Conference on Modelling and Simulation. Brescia, Italy; 2014. p. 664–9.

32. Liu Z, Liu F, Zhang B, Ma F, Gao S. Research on cloud computing and its application in railway. J Beijing Jiaotong Univ. 2010;35(5):14–9.

33. Angeli D, Masala E. A cost-effective cloud computing framework for accelerating multimedia communication simulations. J Parallel Distrib Comput. 2012 Oct;72(10):1373–85.

34. Deelman E, Singh G, Livny M, Berriman B, Good J. The Cost of Doing Science on the Cloud: The Montage Example. In: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing. Piscataway, NJ, USA: IEEE Press; 2008. p. 50:1–50:12.

35. Azadeh A, Ghaderi SF, Izadbakhsh H. Integration of DEA and AHP with computer simulation for railway system improvement and optimization. Appl Math Comput. 2008 Feb 1;195(2):775–85.

36. Ho C-W, Ruehli AE, Brennan PA. The modified nodal approach to network analysis. IEEE Trans Circuits Syst. 1975 Jun;22(6):504–9.

37. AENOR. UNE-EN 50163: Railway Applications - Supply voltages of traction systems. AENOR; 2004.

38. CENELEC. EN 50641: Railway Applications - Fixed installations – Requirements for the validation of simulation tools used for the design of traction power supply systems. CENELEC: European Committee for Electrotechnical Standardization; 2015.

39. AENOR. UNE-EN 50388: Railway Applications - Power supply and rolling stock - Technical criteria for the coordination between power supply (substation) and rolling stock to achieve interoperability. AENOR; 2012.

40. Caíno-Lores S, Fernández AG, García-Carballeira F, Pérez JC. A cloudification methodology for multidimensional analysis: Implementation and application to a railway power simulator. Simul Model Pract Theory. 2015 Jun;55:46–62.

41. White T. Hadoop: The Definitive Guide. O'Reilly Media, Inc.; 2012. 687 p.