DOI 10.15622/ia.23.2.3

A. EBRAHEEM, I. IVANOV

TOWARDS AUTOMATED AND OPTIMAL IIOT DESIGN

Ebraheem A., Ivanov I. **Towards Automated and Optimal IIoT Design.**

Abstract. In today's world, the Internet of Things has become an integral part of our lives. The increasing number of intelligent devices and their pervasiveness has made it challenging for developers and system architects to plan and implement systems of the Internet of Things and Industrial Internet of Things effectively. The primary objective of this work is to automate the design process of Industrial Internet of Things systems while optimizing the quality of service parameters, battery life, and cost. To achieve this goal, a general four-layer fog-computing model based on mathematical sets, constraints, and objective functions is introduced. This model takes into consideration the various parameters that affect the performance of the system, such as network latency, bandwidth, and power consumption. The Non-dominated Sorting Genetic Algorithm II is employed to find Pareto optimal solutions, while the Technique for Order of Preference by Similarity to Ideal Solution is used to identify compromise solutions on the Pareto front. The optimal solutions generated by this approach represent servers, communication links, and gateways whose information is stored in a database. These resources are chosen based on their ability to enhance the overall performance of the system. The proposed strategy follows a three-stage approach to minimize the dimensionality and reduce dependencies while exploring the search space. Additionally, the convergence of optimization algorithms is improved by using a biased initial population that exploits existing knowledge about how the solution should look. The algorithms used to generate this initial biased population are described in detail. To illustrate the effectiveness of this automated design strategy, an example of its application is presented.

Keywords: IIoT, IoT, NGSA-II, TOPSIS, cloud, fog computing, multiobjective optimization, gateway, edge devices.

1. Introduction. One of the drivers of Industry 4.0 is the Industrial Internet of Things (IIoT), the development of which is a consequence of the widespread use of computer technology. Cloud computing is one of the factors driving the success of the Internet of Things (IoT) and Industrial Internet of Things. It allows users to solve computing problems using the resources of connected servers and data centers scattered around the world and working as a single ecosystem [1]. In some cases, the long network distance between edge devices and remote cloud data centers reduces the quality of service, resulting in high latency, high bandwidth usage, and unreliable connections. The concept of fog computing helps solve these problems by bringing computing and storage closer to end users. It can also help reduce unplanned downtime, improve efficiency, and keep the Internet from being flooded with data from a myriad of sources. Thus, fog computing provides services in the same way as the cloud, with better quality parameters that meet the critical requirements of IIoT [2]. Therefore, it can be used as a basis for IIoT systems and models. For fog networks to reach their full potential, good and careful planning is required.

The structure of the paper is as follows: A review of the revised literature is presented in the following section. The architecture of the IIoT system and the suggested techniques for choosing and allocating its components are subsequently specified. After that, the mathematical model is extensively described. The methods related to optimization and decision-making follow. The next steps before the conclusion are the simulation process and outcomes.

2. Related work. The problem of optimizing and synthesizing IIoT systems belongs to a much wider class of problems related to the synthesis of the structure of complex systems, examples of which can be found in [3 – 6]. In this context, special attention was paid to IIoT and cyber-physical systems in [7], where models, methods, and algorithms for synthesizing complex management plans in cyber-physical systems and industrial internet are proposed.

This topic was also addressed by different organizations, alliances and consortiums. As a result, different frameworks and architectures were suggested like RAMI 4.0 (*R*eference *A*rchitectural *M*odel *I*ndustrie 4.0) [8], IIRA (*I*ndustrial *I*nternet *R*eference *A*rchitecture) [9], and IVRA (*I*ndustrial *V*alue *C*hain *R*eference *A*rchitecture) [10]. However, these frameworks do not propose any kind of optimization or automation methods. They rather provide definitions, guidelines and suggestions on how to construct and organize IIoT systems.

Study [11] reviewed the various methods used to optimize IoT networks, and classified them into 8 groups based on the network aspect being optimized (network routing, power consumption, congestion control, heterogeneity, scalability, reliability, quality of service , security). Optimizing and synthesizing IIoT and IoT systems has also been addressed as a problem of network planning and optimization for different technologies like mobile networks [12, 13], LoRaWAN (*Lo*ng *Ra*nge *W*ide *A*rea *N*etwork) [14], SigFox [15], NB-IoT [16].

Paper [17] proposed a model based on a three-layer fog computing architecture that takes into account transmission, propagation, and processing delays, as well as network traffic while keeping the overall deployment cost within the desired budget. Paper [18] uses the same model with modified objective functions, introduces a new optimization method, and compares the results of several optimization algorithms.

The revised literature can be loosely divided into the following categories:

– technology-specific literature addressing certain types of networks or certain layers in the technology stack.

– literature that is more general in nature that addresses the structure of the system and the territorial distribution of its components.

This work falls into the second category and is a continuation and an expansion of the approach presented in [17, 18] by adding the necessary elements to enable an automated buttom-up design approach for IIoT, which, to the best of the authors' knowledge, is not encountered in previous literature.

3. Architecture and methodology. To develop the mathematical model in a simple yet realistic and feasible manner, a four-layer architecture based on fog computing is used (Figure 1).

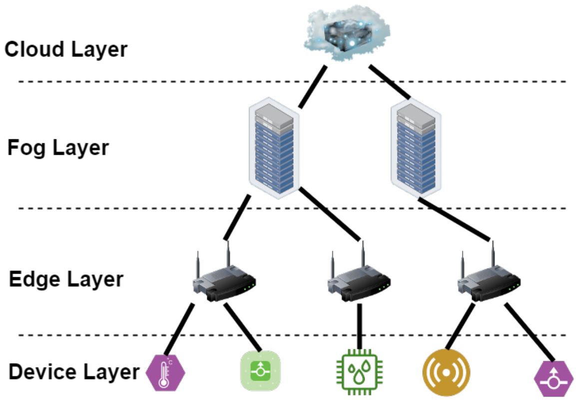

Fig. 1. Fog architecture for IIoT [19]

The layers from bottom to top are:

– layer 1 (the device layer): it includes sensors and actuators used in the system.

– layer 2 (the edge layer): it contains edge gateways that provide edge analysis, data flow multiplexing and throttling, and local data storage.

– layer 3 (the fog layer): it contains fog network nodes, which can be servers or any specialized computing devices.

– layer 4 (the cloud): it is seen as a large network of connected servers operating as a single ecosystem that provides a set of services such as data storage and management, and application hosting [1].

As stated earlier, improving the quality characteristics of IIoT systems is critical and should start early in the design process. To accomplish this purpose, resource allocation at various levels of the aforementioned architecture is used.

The starting point is a set of edge devices, each of which supports one communication technology. These devices have to be connected to edge gateways, the type and location of which are to be selected from a set of possible choices. The gateways must then be connected to a set of fog network nodes, the type and location of which are also to be selected from a set of possible choices, or directly to the cloud. In this work, we consider that the servers of the cloud are located in one data center to which gateways are either connected directly through the internet (the fog layer is bypassed) or indirectly through the fog layer. The fog layer is connected to the cloud using special links.

The approach used is to divide the solution process into two main optimization stages and one intermediate stage (Figure 2).

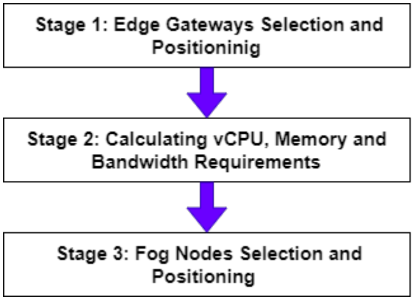

Fig. 2. Solution stages

The types and location of edge gateways are established in the first stage to reduce equipment cost and device-to-gateway distance, which also reduces deployment overhead, power consumption, and enhances connection quality by reducing link errors and packet loss [20]. In preparation for the third stage, the vCPU (virtual Central Processing Unit), memory, and bandwidth requirements are computed in the second stage. The third stage specifies the type and location of the fog network nodes, as well as whether or not to connect to the fog network node or to the cloud directly. The main goal in this stage is to reduce the cost, latency, and traffic traveling over the network.This split simplifies the search for optimal solutions and avoids expanding the dimensionality of the solution space.

4. Mathematical Model. The major purpose of the suggested methodology is to give users the ability to select technological solutions (gateways, servers, communication channels) that best match the needs of their IIoT system at every level. While the majority of the revised literature focuses on either the highest three levels of the used model or the lowest two levels, the contribution of this study is the offering of a holistic approach that addresses the system as a whole. The incorporation of technology-specific algorithms or procedures (routing, automatic configuration, etc.) is beyond the scope of this study. The main components of the model (Figure 3) are:

– available resources represented by mathematical sets that could possibly reflect information stored in database tables for example,

– binary decision variables, whose values determine whether a specific resource is allocated or not and where it is installed,

– constraints that guarantee the solution to be feasible and to meet the technical requirements,

– cost functions that have to be either minimized or maximized. In this work, all the cost functions are to be minimized.

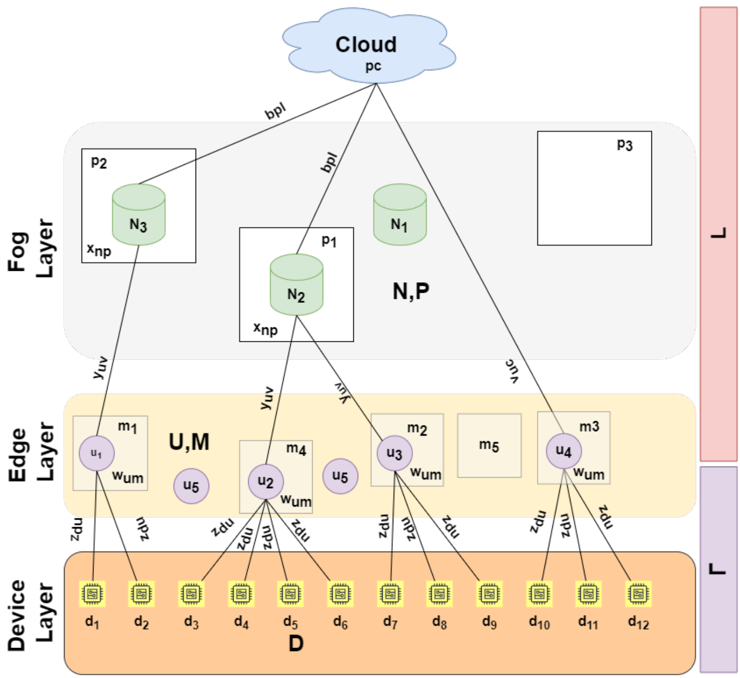

Fig. 3. Architecture and mathematical model of the fog network

The model expands the model proposed in [17, 18] by adding an additional low-level layer, which is the device layer mentioned earlier. It is worth noting that the formulation of the sets U,N,P,L, the decision variables $x_{np}, y_{up}, v_{uc}, b_{pl}$, the constraints from 8-14, and objectives from 3 to 5 described next is based on these two works. The following are the seven mathematical sets employed in this model:

– Γ is the set of communication technologies $\{\gamma_1, \gamma_2, \dots\}$ that can be used. Commnication technology can be something like ZigBee , wireless, and so on. Each technology γ_i is characterized by the maximum number of supported devices d^{γ}_{max} and the carrier frequency f^{γ}. Some technologies support adaptive data rates and some do not, so we assume that devices and gateways can be configured to operate at the same data rate as long as they support the same technology.

– D is the set of edge devices $\{d_1, d_2, \dots\}$. Each edge device d_i is characterized by its communication technology γ^d, location p^d, the number of bytes per second transmitted by the device θ^d, receiver antenna gain G^d_{RX} (dBi), receiver sensitivity S^d_{RX} (dBm), transmitter power output P^d_{TX} (dBm). If θ^d is not precisely known, because the device sends packets on the occurrence of some events and not on a timely basis, worst-case scenario can be used to estimate θ^d. Sometimes θ^d might be implied by the used technology γ^d.

– U is the set of edge gateways that can be installed in the network $\{u_1, u_2, \dots\}$. Each gateway is characterized by the speed of the communication channel κ^u, the cost ξ^u and the set of communication technologies supported by the gateway Γ^u. For each $\gamma^u \in \Gamma^u$ there is a receiver antenna gain $G^{\gamma u}_{RX}$ (dBi), a receiver sensitivity $S^{\gamma u}_{RX}$ (dBm), and a transmitter power output $P^{\gamma u}_{TX}$ (dBm). In our model, these values are known from the very beginning. However, some of the values need to be calculated during stage 2. These values include the total amount of memory λ^u and the number of virtual processors (vCPU) α^u required to process the data sent by the gateway to the fog node or the cloud θ^u which also requires calculation.

– N is the set of fog nodes $\{n_1, n_2, \dots\}$. Each node is characterized by the total available memory λ^n, the number of available vCPUs α^n, the network interface bandwidth θ^n in bytes per second, and the cost ξ^n in currency units.

– L is the set of possible types of links that can be used to connect the nodes of the fog network and the cloud $\{l_1, l_2, \dots\}$. Each channel is characterized by its bandwidth in bytes per second θ^l, and a cost in currency unit per meter ξ^l .

– M is the set of possible places to install gateways $\{m_1, m_2, \dots\}$.

– P is the set of possible places to install the fog nodes $\{p_1, p_2, \dots\}$.

In this model, six decision variables are utilized to allocate the aforementioned resources. This allocation involves the installation of network hardware at specific locations and establishing links between the installed hardware.

– w_{um} is a binary decision variable such that $w_{um} = 1$ if and only if the gateway $u \in U$ is installed at the location $m \in M$.

– z_{du} is a binary decision variable such that $z_{du} = 1$ if and only if the edge device $d \in D$ is connected to the gateway $u \in U$.

– x_{np} is a binary decision variable such that $x_{np} = 1$ if and only if the fog net node $n \in N$ is set to the location $p \in P$.

– y_{up} is a binary decision variable such that $y_{up} = 1$ if and only if gateway $u \in U$ is connected to the location $p \in P$.

– v_{uc} is a binary decision variable such that $v_{uc} = 1$ if and only if the gateway $u \in U$ is connected to the cloud.

– b_{pl} is a binary decision variable such that $b_{pl} = 1$ if and only if the node of the fog network, set at the location $p \in P$, is connected to the cloud by a channel of type $l \in L$.

The first stage in the adopted approach employs the following constraints:

– Constraint 1: the maximum number of devices connected to a gateway via a certain technology must not be exceeded:

$$\sum_{d \in D} z_{du} \delta(d, \gamma) \leqslant q(u, \gamma) d^{\gamma}_{max}, \tag{1}$$

where $\delta(d, \gamma)$ is a binary function that returns 1 if device d supports technology γ, and $q(u, \gamma)$ is a function that returns the number of physical interfaces of type γ in gateway U.

– Constraint 2: this constraint is related to the placement of the gateways. A good approach is to use a link budget constraint, which can be defined as the difference between the gains and losses of the system and must be greater than zero:

$$\left.\begin{array}{l} P^{d}_{TX} + G^{\gamma u}_{RX} - S^{\gamma u}_{RX} - P^{\gamma ud}_{L} > 0 \\ P^{u}_{TX} + G^{d}_{RX} - S^{d}_{RX} - P^{\gamma ud}_{L} > 0 \end{array}\right\} \gamma^{u} \in \Gamma^{u}, u \in U, d \in D, \tag{2}$$

where $P^{\gamma ud}_{L}$ is the path loss from the device to the gateway for the frequency used by communication technology γ. Computing this value requires knowing the precise locations of the transmitter and receiver and the environment in which the signal is traveling. There are various approaches to get close to this value. One approach is to analyze site-specific radio wave propagation using ray tracing techniques, which are used by software like Wireless Insite, that include a collection of RF propagation models (3D ray-tracing, quick ray-based methods, and empirical models) [21]. Matlab communications toolbox provides such capabilities by using RayTracing objects which are propagation models that compute propagation paths using 3-D environment geometry [22]. Empirical models can also be used to estimate the value of the path loss like Cost-231 Walflsch-Ikegami model [23, 24], Hata model, close-in model, floating-intercept model, Longly-Rice model, communications research centre predict model [25]. In cases where there is a line of sight, the free space path loss model can be utilized:

$$P_L = 32.45 + 20log(d) + 20log(f), \tag{3}$$

where d is the distance between the device and the edge gateway in km and f is the frequency in MHz.

– Constraint 3: each device is connected to exactly one gateway:

$$\sum_{u \in U} z_{du} = 1, d \in D. \tag{4}$$

– Constraint 4: each location has a maximum of one gateway:

$$\sum_{u \in U} w_{um} \leqslant 1, m \in M. \tag{5}$$

– Constraint 5 : a gateway can only be installed if at least one device is actually connected to it:

$$\vartheta(u) = 1, u \in U, \tag{6}$$

$$\vartheta(u) = \begin{cases} 1: & \left(\sum_{m \in M} w_{um} > 0 \text{ and } \sum_{d \in D} z_{du} > 0\right) \\ & \text{ or } \left(\sum_{m \in M} w_{um} = 0 \text{ and } \sum_{d \in D} z_{du} = 0\right), u \in U. \\ .0: & \text{otherwise} \end{cases} \tag{7}$$

– Constraint 6: a device and a gateway can be linked if they support the same communication technology:

$$z_{du} \leqslant \sum_{\gamma \in \Gamma} \gamma(d, \gamma) q(u, \gamma), d \in D, u \in U. \tag{8}$$

– Constraint 7: the traffic passing through a gateway cannot exceed its bandwidth capability:

$$\sum_{d \in D} z_{du} \theta^d \leqslant \kappa^u. \tag{9}$$

The following set of constraints is used in stage 3 described earlier:

– Constraint 8: no more than one fog node can be installed at any given location:

$$\sum_{n \in N} x_{np} \leqslant 1, p \in P. \tag{10}$$

– Constraint 9: each gateway used is connected to only one fog node or to the cloud:

$$\sum_{p \in P} y_{up} + v_{uc} = 1, u \in U. \tag{11}$$

– Constraint 10: each fog node used is connected to the cloud:

$$\sum_{n \in N} x_{np} = \sum_{l \in L} b_{pl}, p \in P. \tag{12}$$

– Constraint 11: the number of vCPUs required by edge gateways does not exceed the number of vCPUs that can be provided by the fog node:

$$\sum_{u \in U} y_{up} \alpha^u \leqslant \sum_{n \in N} x_{np} \alpha^n, p \in P. \tag{13}$$

– Constraint 12: the amount of memory required by the gateways does not exceed the amount of memory that can be provided by the fog node to which they are connected:

$$\sum_{u \in U} y_{up} \lambda^u \leqslant \sum_{n \in N} x_{np} \lambda^n, p \in P. \tag{14}$$

– Constraint 13: the total bandwidth required to connect edge gateways to the fog node does not exceed the bandwidth of the fog node:

$$\sum_{u \in U} y_{up} \theta^u \leqslant \sum_{n \in N} x_{np} \theta^n, p \in P. \tag{15}$$

– Constraint 14: the bandwidth required to send data from the fog node to the cloud does not exceed the bandwidth of the link used:

$$\sum_{n\in N}\sum_{u\in U} y_{up}x_{np}\theta^{u}r^{n} \leqslant \sum_{l\in L} b_{pl}\theta^{l}, p\in P. \tag{16}$$

where r^n is the average percentage of data sent to the cloud from the incoming data at the fog network node.

During the first stage the goal is to:

– Objective 1: minimize the total distance from devices to gateways:

$$\Theta = \min\left(\sum_{d\in D}\sum_{u\in U} z_{du}\mathscr{E}\left(d, \sum_{m\in M} w_{um}m\right)\right), \tag{17}$$

where $\mathscr{E}(a,b)$ is a function that returns the distance between two points a and b.

– Objective 2: minimize the cost of deployment, which is primarily the total cost of the gateways:

$$\Lambda = \min\sum_{m\in M}\sum_{u\in U} w_{um}\xi^{u}. \tag{18}$$

During stage 3, the goal is to:

– Objective 3: minimize network latency:

$$D_T = \min(D_t + D_n + D_p), \tag{19}$$

where:

– D_t is the transmission delay, which can be calculated using the following formula:

$$D_t = \sum_{u\in U}\theta^{u}/\kappa^{u}\left((h_1+1)\left[\sum_{p\in P} y_{up}\right] + (h_1+h_2+1)v_{uc}\right. \\ \left. + (h_2+1)\sum_{n\in N}\sum_{p\in P}\sum_{u\in U} y_{up}x_{np}\theta^{u}r^{n}\left(\sum_{l\in L}\frac{b_{pl}}{\theta^{l}}\right)\right), \tag{20}$$

where h_1 is the average number of hops from edge gateways to fog network nodes, and h_2 is the average number of hops from fog network nodes to the cloud. It is assumed that the average number of transitions from edge gateways to the cloud is (h_1+h_2).

– D_n is the propagation delay, which can be calculated using the following formula:

$$D_n = \sum_{u \in U} \frac{\mathscr{E}(m_u, p_c)}{v} v_{uc} + \sum_{u \in U} \left(\frac{\mathscr{E}(m^u, p^u) + \mathscr{E}(p^u, p^c)}{v} \sum_{p \in P} y_{up} \right), \quad (21)$$

where v is the signal propagation speed, $m^u = \sum_{m \in M} w_{um} m$ is the location of the gateway u, $p^u = \sum_{p \in P} \sum_{n \in N} x_{np} y_{up} p$ is the location at which is installed the fog node to which the gateway u is connected, and p^c is the location at which the data center representing the cloud system (or at least part of it) is installed.

– D_p is the processing delay, which can be calculated using the following formula:

$$D_p = \sum_{u \in U} (h_1 + h_2) k, \quad (22)$$

where k is the average processing delay in seconds per transition.

– Objective 4: minimize the total traffic going to the cloud, which is the sum of the traffic from all the edge gateways connected to the cloud, plus the traffic coming from the various fog network nodes:

$$\mathscr{T} = \min \left(\sum_{u \in U} \left[v_{uc} + \sum_{n \in N} \sum_{p \in P} x_{np} y_{up} r^n \right] \theta^u \right). \quad (23)$$

– Objective 5: minimize costs:

$$\mathscr{C} = \min \left(\sum_{p \in P} \left[\sum_{n \in N} x_{np} \xi^n + \sum_{l \in L} b_{pl} \mathscr{E}(p, p^c) \xi^l \right] \right). \quad (24)$$

At the intermediate stage 2, the following values are calculated for the used gateways:

– the number of bytes per second sent by the gateway θ^u:

$$\theta^u = \sum_{d \in D} z_{du} \theta^d, u \in U, \quad (25)$$

– the number of vCPUs α^u and the amount of RAM λ^u needed to process the data sent by the gateway to the fog nodes. These quantities are difficult to calculate accurately. When visiting the websites of popular cloud providers and technology companies [26 – 28], it turned out that the maximum bandwidth is related to the number of virtual processors and RAM in addition to the type of workload and the required performance in addition to other factors. To determine these values in our model, we assume a linear relationship between the outbound bandwidth of the edge gateway and the number of required virtual processors and RAM.

$$\begin{aligned} \alpha^u &= \kappa_\alpha \theta^u + \alpha_{min} + \alpha_{margin}, \\ \lambda^u &= \kappa_\lambda \theta^u + \lambda_{min} + \lambda_{margin}, \end{aligned} \tag{26}$$

where:

– κ_α is a coefficient that relates throughput to the number of vCPUs.

– α_{min} is the minimum number of vCPUs required for a virtual machine to function properly.

– α_{margin} is the number of vCPUs dedicated to handle unexpected load changes or ensure proper system operation.

– κ_λ is a coefficient that relates bandwidth to the amount of RAM.

– λ_{min} is the minimum amount of RAM required for a virtual machine to function correctly.

– λ_{margin} is the amount of RAM, reserved to handle unexpected changes in load or ensure proper system operation.

The proposed model does not claim to be complete and has a number of limitations:

– Devices are treated as stationary and do not move from one gateway to another. In such cases, a network should be planned to ensure and maximize coverage in areas where assets are moving. However, these cases are not considered by the current model.

– Energy consumption is not taken into account in an explicit way. This model aims to minimize the distance between IoT devices and gateways, which can lead to situations where some technology-specific settings that allow for less energy consumption can be applied.

It is also worth mentioning that the system structure and the software operation requirements are two important correlated factors in the context of territorially distributed systems. On the one hand, the system architecture can affect how efficiently data is processed. On the other hand, the structure of the system may be determined by the requirements for software operation. While the proposed method does not take into consideration the technicalities of the used software, it aims to enhance the quality of service factors, which usually positively impact the software performance.

The environmental factors also affect the quality of the obtained solutions, by affecting the link budget of the communication systems. Rain, terrain, and interference from other sources can all affect the quality of the communication link. While this might sometimes be accounted for by choosing an appropriate propagation model and good quality equipment (antennas, cables,....), it is hard to take into consideration all the factors. The authors recommend using equation 2 with a positive margin or safety factor $P_{margine}$:

$$\left.\begin{array}{l} P_{TX}^{d}+G_{RX}^{\gamma u}-S_{RX}^{\gamma u}-P_{L}^{\gamma ud}>P_{margine} \\ P_{TX}^{u}+G_{RX}^{d}-S_{RX}^{d}-P_{L}^{\gamma ud}>P_{margine} \end{array}\right\} \gamma^{u}\in\Gamma^{u}, u\in U, d\in D. \quad (27)$$

5. Searching for optimal solutions. The problem is formulated as a multiobjective optimization problem. Optimality is understood as a situation where no objective function can be improved without worsening at least one other (Pareto optimality). In the absence of any additional information, it cannot be said that one of the Pareto optimal solutions is better than the other. This means that a slight increase in costs may lead to a decrease in traffic or latency, or an increase in delay may be accompanied by a decrease in traffic or costs. In such situations, it is important to find as many solutions as possible on the Pareto front. By isolating one specific optimal solution at a time or by scalarizing the problem, certain common optimization approaches offer to reduce a multi-objective optimization problem to a single-objective optimization problem. When such techniques are employed, they must be used repeatedly in order to acquire the greatest number of solutions, which increases the time costs. However, there exist techniques that aim to guarantee convergence to a Pareto-optimal set while maintaining diversity in solutions [29]. They simultaneously consider all objective functions reserving the nature of the original problem. Evolutionary algorithms are one such method. This article uses the Non-dominated Sorting Genetic Algorithm II (NSGA II), which is a very famous and widely used variant of the genetic algorithm. According to [30], by solving several test problems using the NSGA-II, it was found that this method outperforms other algorithms under testing, such as PAES (Pareto Achieved Evolution Strategy) and SPEA (Strength Pareto evolutionary algorithm), in terms of finding a diverse set of solutions. The algorithm follows the general scheme of the genetic algorithm with modified crossover and selection. By definition, A solution p_1 is said to dominate the other solution p_2 if p_1 is no worse than p_2 for all objectives and p_1 is strictly better than p_2 in at least one objective.

The crowding distance measures the density of solutions around an individual in the objective space based on the distances between neighboring solutions in each dimension. It is used in NSGA-II to select diverse solutions for the next generation. For each objective function, the population is initially sorted in ascending order of magnitude. Intermediate solutions are given a distance value equal to the absolute normalized difference in the function values of two adjacent solutions, while the boundary solutions are given an infinite distance value. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective [29].

The selection of individuals is carried out using a binary tournament selection, which involves several tournaments between two individuals chosen at random from the population each time. The winner of each tournament is selected for crossover. The steps of NSGA-II can be summarized as follows [31]:

1. initialize the NSGA-II parameters like the population size N, the generation counter $t = 0$, and the offspring population obtained by applying crossover and mutation $Q_t = \emptyset$.

2. create the initial parent population P_t according to the procedures described later on.

3. merge the parent and offspring populations for maintaining elitism $R_t = P_t \cup Q_t$.

4. sort the population of candidate solutions R_t into different fronts $F_1, F_2, ..., F_r$ using non-dominated sorting.

5. calculate the crowding distance of each candidate solution in each front F_i.

6. using binary tournament selection, choose the parent population for the next generation P_{t+1} based on the non-domination status and crowding distance .

7. apply crossover and mutation on P_{t+1}, generate offspring population Q_{t+1} for next generation.

8. increment t.

9. check for termination criteria which might be a maximum number of generations or a satisfactory level of convergence. If the termination criteria is not met repeat steps from 3 to 9.

Figure 4 illustrates the main steps of the algorithms as described in the original article [29].

In step 1 during the generation of the initial population, individuals that represent *pseudo-solutions* are generated. A *pseudo-solution* might not satisfy all the constraints but its *structure* is similar to a feasible solution.

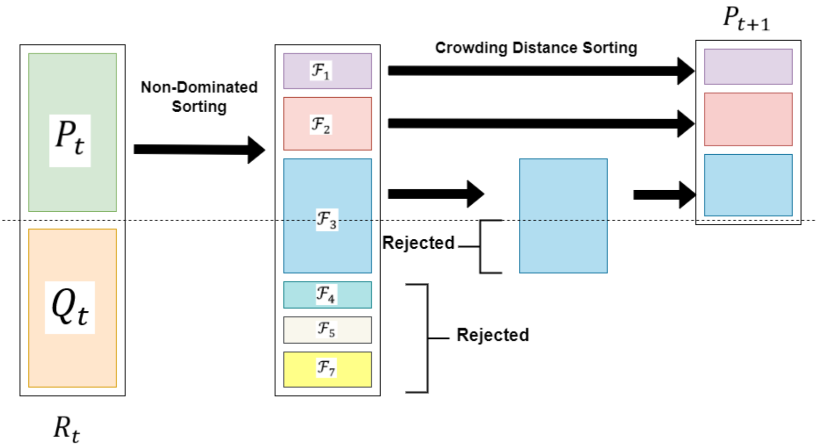

Fig. 4. NSGA II procedure [29]

This helps reduce the time required to reach feasible and optimal solutions. The following procedure is used:

1. determine the set of communication technologies Γ_{used} used by all the devices in D.

2. filter out any gateway in U that does not support any $\gamma \in \Gamma_{used}$. As a result, U can be written as $(U = U_{usefull} \cup U_{useless})$.

3. determine the maximum number of gateways $i_{u,max}$ that can be installed, which is the minimum between the size of $U_{usefull}$ and the size of M.

4. choose a random number $i_u \in \{1, \ldots, i_{u,max}\}$ of gateways from $U_{usefull}$ and install them at random locations from M. As a result, we obtain w_{um}.

5. assign each device to an installed gateway that supports its communication technology. As a result, we obtain z_{du}.

6. repeat 3, 4 and 5 until the required number of individuals in the initial population is reached.

A similar procedure is used to generate the initial population for the second step:

1. choose a subset $U_{cloud} \subset U_{installed} = \{u \in U : \exists m \in M, w_{um} \neq 0\}$. For each $u \in U_{cloud}$ set $v_{uc} = 1$.

2. if the size of U_{cloud} is equal to the size of $U_{installed}$ set $x_{np} = 0, \forall n \in N, p \in P$ and $b_{pl} = 0, \forall p \in P, l \in L$ then go the previous step if the number of individuals is not reached.

3. if the size of U_{cloud} is less than to the size of $U_{installed}$ determine the maximum number of fog nodes $j_{n,max}$ that can be installed, which is the minimum between the size of P, the size of N and the size of $U_{installed} \setminus U_{cloud}$.

4. choose a random number $j_n \in \{1, ..., j_{n,max}\}$ of fog nodes $N_{installed} \subset N$ and install them at random locations $P_{occupied} \subset P$. As a result, we obtain x_{np}.

5. assign to each location $p \in P_{occupied}$ its closest gateway first to make sure all fog nodes are used then assign the rest of the gateways in the same way. As a result we obtain y_{up}.

6. for every $p \in P_{occupied}$ assign a random $l \in L$. As a result we obtain b_{pl}.

7. repeat the previous steps until the required number of individuals in the initial population is reached.

In this work, the algorithm implementation within the pymoo framework [32] is used. Pymoo is a python framework that supports modern single-objective and multi-objective optimization algorithms.

To automatically select the best compromise solution among those obtained, the Technique for Order of Preference by Similarity to the Ideal Solution (TOPSIS) method is used. It is a multi-criteria decision making (MCDM) technique that compares a set of alternatives based on predefined criteria. According to this method, the chosen optimal solution must have the shortest Euclidean distance from the positive-ideal solution and the longest Euclidean distance from the negative-ideal solution [33]. In this work, the TOPSIS method is implemented in python based on [34]:

– Construct a normalized objective matrix with m rows and n columns:

$$t_{ij} = \frac{f_{ij}}{\sqrt{\sum_{i=1}^{m} f_{ij}^2}}. \tag{28}$$

– Construct a weighted normalized objective matrix by multiplying each column by a weight w_j corresponding to the importance of the objective function:

$$v_{ij} = t_{ij} w_j. \tag{29}$$

– Determine the positive ideal solution, A^+, and negative-ideal solution, A^- by finding the best value of each objective function. Where maximization is required, the best value is the largest value within the column of the objective matrix and where minimization is required (which is the case for all the objective functions used in our case), the best value is the smallest value in the column. Mathematically, the positive ideal solution is given by:

$$A^{+} = \left\{ \left(\max_{i}(v_{ij}) \,\middle|\, j \in J \right), \left(\min_{i}(v_{ij}) \,\middle|\, j \in J' \right) \middle| i \in 1,2,\ldots,m \right\}$$
$$= \left\{ v_1^{+}, v_2^{+}, \ldots, v_j^{+}, \ldots, v_n^{+} \right\}, \quad (30)$$

where J is the set of indexes of maximization objectives and J' is the set of indexes of minimization objectives in the overall set $\{1,2,3,4,...,n\}$. Next, find the worst value of each objective, which is the smallest and largest value within the column of the objective matrix for maximization and minimization objectives respectively. These values constitute the negative-ideal solution is given by:

$$A^{-} = \left\{ \left(\min_{i}(v_{ij}) \,\middle|\, j \in J \right), \left(\max_{i}(v_{ij}) \,\middle|\, j \in J' \right) \middle| i \in 1,2,\ldots,m \right\}$$
$$= \left\{ v_1^{-}, v_2^{-}, \ldots, v_j^{-}, \ldots, v_n^{-} \right\}. \quad (31)$$

– Calculate the Euclidean distance between each solution and the positive-ideal and negative-ideal solutions:

$$S_{i+} = \sqrt{(\sum_{j=1}^{n} \left(v_{ij} - v_j^{+} \right)^2}, i = 1,2,3,\ldots,m. \quad (32)$$

$$S_{i-} = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^{-} \right)^2}, i = 1,2,3,\ldots,m. \quad (33)$$

– Calculate the closeness of each optimal solution:

$$C_i = \frac{S_{i-}}{S_{i-} + S_{i+}}, \quad (34)$$

when $S_{i-} = 0$, $C_i = 0$ and solution i is the closest to the negative ideal. When $S_{i+} = 0$,$C_i = 1$ and solution i is the closest to the positive ideal. The solution having the largest C_i is the recommended solution.

The order in which the previously mentioned algorithms and procedures are applied is shown in Figure 5.

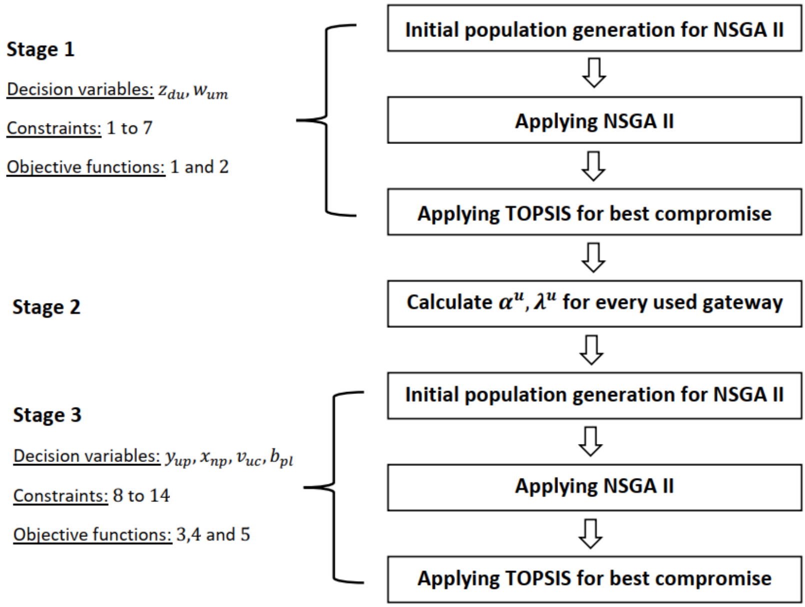

Fig. 5. The sequence of applying NSGA II and TOPSIS

6. Simulation and Results. A simple example is provided to demonstrate the results of applying the proposed method. However, the method was successfully applied to much larger and more complex cases. The success of the application refers to the convergence of the method and the quality of obtained solutions. Some of the numerical values chosen for the simulation are similar to what might be found in a real scenario, but some values, such as the position of the fog nodes and the cloud data center, are chosen to make it easier to visually demonstrate the concept. In real life, they would exist at much greater distances from the edge of the system. The example uses a set of abstract communication technologies that are characterized by the maximum number of supported devices d^{γ}_{max} and the used frequency f^{γ} (Table 1). These communication technologies are supported by both the set of edge devices and the set of gateways. The edge devices in this example are listed in Table 2 and are described by the supported interface γ^d, the number of bytes sent per second θ^d, and their location $p^d = (x^d, y^d)$. It is worth noting that for simplicity and convenience of demonstration, it is assumed that all components of the IIoT system can be distributed over a geographical area whose dimensions are $4000m \times 4000m$.

Table 1. Parameters of the used communication technologies

	d^{γ}_{max}	f^{γ} MHz
γ_0	32	915
γ_1	128	433.92
γ_2	128	17.12

Table 2. Parameters of the edge devices

[h!]	γ^d	θ^d (bytes per second)	x^d (m)	y^d (m)	G^d_{RX} (dBi)	S^d_{RX} (dBm)	P^d_{TX} (dBm)
d_0	γ_0	512	1391	1945	2	-70	14
d_1	γ_0	25600	1408	1461	2	-70	14
d_2	γ_0	6400	488	1578	2	-72	14
d_3	γ_0	12800	834	518	2	-71	14
d_4	γ_0	12800	594	387	2	-75	14
d_5	γ_0	3200	949	1628	2	-73	14
d_6	γ_0	25600	904	235	2	-72	14
d_7	γ_1	25600	401	3323	2	-82	22
d_8	γ_1	3200	200	3306	2	-81	20
d_9	γ_2	51200	3348	2960	2	-90	20
d_10	γ_2	12800	159	2917	2	-86	22
d_11	γ_0	12800	3267	3273	2	-89	22
d_12	γ_2	25600	3406	3512	2	-91	22
d_13	γ_2	12800	3800	2899	2	-88	22
d_14	γ_1	6400	3748	3546	2	-89	22

The gateways to choose from are listed in Table 3. In this case, there are 5 gateways described by their link speed κ^u, cost ξ^u, and supported communication technologies Γ^u. A gateway can support multiple technologies and multiple instances of the same technology.

Gateways can be physically installed in 6 possible locations, the coordinates of which are indicated in Table 4. The proposed method selects the best gateway and the best location given the given constraints at the first stage of optimization. It also determines which devices should be connected to the selected gateway.

The fog nodes to choose from are listed in Table 5. In this case, there are 4 nodes that can be described using available memory λ^n, number of available virtual processors α^n, network bandwidth interface θ^n and cost ξ^n.

Table 3. Parameters of the gateways

	κ^u (Mbs)	ξ^u ($)	Γ^u				
			γ	$q(u,\gamma)$	$G_{RX}^{\gamma u}$ dBi	$S_{RX}^{\gamma u}$ dBm	$P_{TX}^{\gamma u}$ dBm
u_0	95	20000	γ_0	1	3	−120	26
			γ_1	1	4	−110	24
u_1	92	25000	γ_1	1	4	110	24
			γ_2	1	25	3	-100
u_2	32	18000	γ_1	1	4	110	24
			γ_2	1	25	3	-100
u_3	67	15000	γ_1	1	4	110	24
u_4	85	12000	γ_0	1	3	-120	26

Table 4. Coordinates of the possible locations for installing gateways

	x^m (mm)	y^m (mm)
m_0	921	1254
m_1	339	3093
m_2	3592	3239
m_3	489	2830
m_4	3847	3225
m_5	1026	956

Table 5. Parameters of fog nodes

	λ^n (Gigabyte)	α^n	θ^n (megabit per second)	ξ^n ($)
n_0	256	96	3598	1000000
n_1	128	48	2768	520000
n_2	256	24	3486	600000
n_3	128	96	2265	4800000

Nodes can be physically installed in 4 possible locations, the coordinates of which are indicated in Table 6. The proposed method will select the best variant and the best location, taking into account the given constraints, at the second stage of optimization. It will also determine which gateways should be connected to the selected node. The data center or cloud system is usually far from the edges of the IIoT network, but for demonstration purposes, it is considered to be set to $(x^c, y^c) = (2000, 3800)$.

Table 6. Coordinates of the possible locations for installing fog network nodes

	x^p (mm)	y^p (mm)
p_0	2569	3084
p_1	1679	2589
p_2	1209	2896

Four possible communication channels or links can be used to connect the fog node to the data center. For each communication channel, the transmission rate θ^l and the price ξ^l are listed in Table 7. The price represents the cost of deployment in dollars per meter.

Table 7. Parameters of the communication links

	θ^l (megabits per second)	ξ^l (\$/m)
l_0	10	6
l_1	100	14
l_2	1000	24
l_3	10000	50

The initial data on the system are depicted in Figure 6.

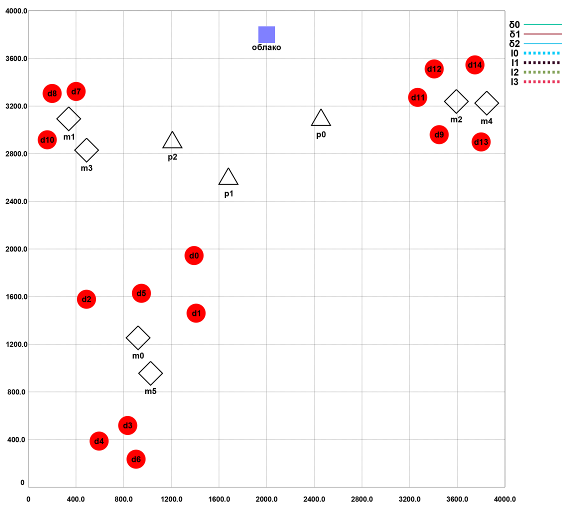

Fig. 6. Initial data about the IIoT system

In the first optimization step, NSGA II is applied first. As a result, we get 3 optimal solutions after 200 generations. As for the path loss, we consider that there is a direct line of sight between the devices and gateways for simplicity. Hence, the free space path loss model is considered.

In this example, we have no preference as to which solution to use, so TOPSIS is applied, resulting in the following solution:

$$w_{um} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$z_{du} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

These results are shown in Figure 7.

Note that, for example, $w_{22} = 1$ means that u_2 is installed at m_2, and $z_{04} = 1$ means that d_0 is connected to u_4. At the intermediate stage, additional parameters of the installed gateways are determined, as in Table 8.

As a result of applying NSGA II at the second stage of optimization, 7 optimal solutions were obtained. In this example, we have no preference as to which solution to use, so TOPSIS is applied again, resulting in the following solution:

$$y_{up} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, x_{np} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0, \end{bmatrix}, b_{pl} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0, \end{bmatrix}, v_{uc} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (35)$$

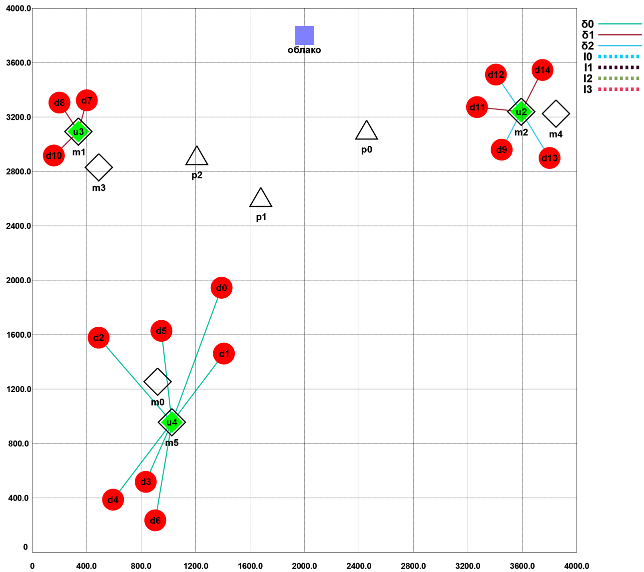

Fig. 7. Visualization of the results of the first stage

Table 8. Additional parameters for the installed gateways

	λ^u (Gigabytes)	α^u	θ^u (bytes per second)
u_0	-	-	-
u_1	-	-	-
u_2	2	2	3720
u_3	2	2	1520
u_4	2	2	5756

The final results are shown in Figure 8.

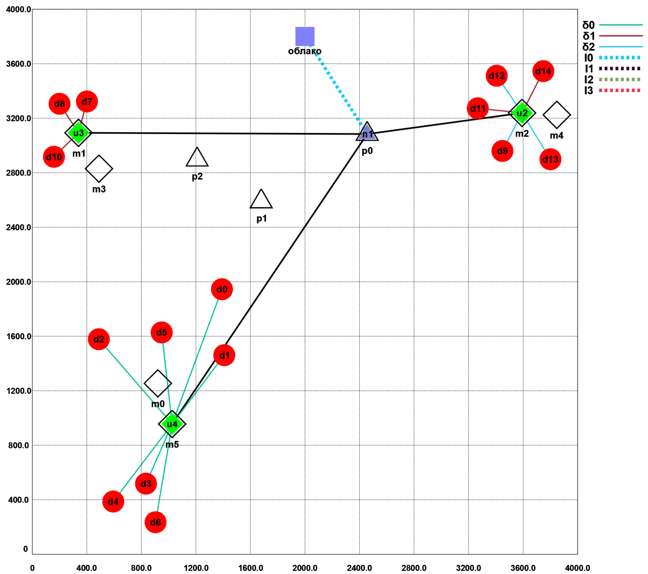

Fig. 8. Visualization of the results of the third stage

Note that, for example, $y_{20} = 1$ means that u_2 is connected to a node that is installed at p_0, and $x_{10} = 1$ means that n_1 is assigned to p_0. $b_{00} = 1$ means that a channel of type l_0 is used to connect the fog node at p_0 to the cloud.

Thus, the proposed method automatically allocates network resources using TOPSIS and NSGA II. The user is only required to provide a database of available resources and possible places to install these resources. The algorithm then decides what to use and where to install it.

7. Conclusion. Although the proposed technique does not consider mobility or technology-specific factors, it has the potential to ensure IIoT performance and effectiveness and decrease the dependency on the knowledge of system architects by looking for the best non-dominated Pareto solutions. The importance of this work lies in its holistic approach and generality. It not only handles specific architectural layers as most methods presented in the revised literature do but also includes all layers in the system hierarchy. The generality comes from the technique being agnostic to the type of IoT technology by using only general properties that characterize each level component. While the presented example demonstrates the applicability of the method on small-scale problems, applying the method to larger-scale problems has also yielded similar results. The next step in this research is to introduce more hardware acceleration by exploiting the potentials of graphics cards, collect statistical data about the results of applying the proposed technique in different projects, which might help further verify its usefulness and practicality, and comparing the results and performance indicators of using the proposed model with optimization algorithms other that NSGA-II.

References

1. Microsoft Azure official web site. Available at: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-the-cloud. (accessed 02.01.2023).
2. Basir R., Qaisar S., Ali M., Aldwairi M., Ashraf M.I., Mahmood A., Gidlund M. Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges. Sensors. 2019. vol. 19(21). no. 4807.
3. Tsvirkun A.D. Osnovy sinteza struktury slozhnyh system [Fundamentals of synthesis of the structure of complex systems]. M.: Nauka, 1982. 200 p. (In Russ.).
4. Tsvirkun A.D., Akinfiev V.K., Solov'ev M.M. Modelirovanie razvitija krupnomasshtabnyh sistem: (Na primere toplivno-jenergeticheskih otraslej i kompleksov) [Modeling the development of large-scale systems: (On the example of fuel and energy industries and complexes)]. M.: Ekonomika, 1983. 176 p. (In Russ.).
5. Akinfiev V.K., Cvirkun A.D. Metody i instrumental'nye sredstva upravlenija razvitiem kompanij so slozhnoj strukturoj aktivov [Methods and tools for managing the development of companies with a complex asset structure.]. M.: IPU RAN, 2020. 307 p. (In Russ.).
6. Tsvirkun A.D., Akinfiev V.K., Filippov V.A. Imitacionnoe modelirovanie v zadachah sinteza struktury slozhnyh system [Simulation modeling in problems of synthesis of the structure of complex systems]. M.: Nauka, 1985. 173 p. (In Russ.).
7. Potryasayev S.A. Sintez tehnologij i kompleksnyh planov upravlenija informacionnymi processami v promyshlennom internete. dis. d-r teh. nauk. [Synthesis of technologies and compex plans for managing information processes in the industrial Internet. doct. diss. in technical sciences.]. St. Petersburg, 2020. (In Russ.).
8. International society of automation official website. Available at: https://www.isa.org/intech-home/2019/march-april/features/rami-4-0-reference-architectural-model-for-industr. (accessed 13.09.2023).
9. Industry IoT consortium official website. Available at: https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf. (accessed 12.09.2023).
10. Industrial value chain initiative official website. Available at: https://docs.iv-i.org/doc_161208_Industrial_Value_Chain_Reference_Architecture.pdf. (accessed 14.09.2023).
11. Srinidhi N.N., Kumar S.D., Venugopal K.R. Network optimizations in the Internet of Things: A review. Engineering Science and Technology, an International Journal. 2019. vol. 22. no. 1. pp. 1–21.
12. Ceselli A., Premoli M., Secci S. Mobile Edge Cloud Network Design Optimization. IEEE/ACM Transactions on Networking. 2017. vol. 25. no. 3. pp. 1818–1831.
13. Chimmanee K., Jantavongso S. Practical mobile network planning and optimization for Thai smart cities: Towards a more inclusive globalization. Research in Globalization. 2021. vol. 3. no. 100062.
14. Gava M.A., Rocha H.R.O., Faber M.J., Segatto M.E.V., Wortche H., Silva J.A.L. Optimizing Resources and Increasing the Coverage of Internet-of-Things (IoT) Networks: An Approach Based on LoRaWAN. Sensors. 2023. vol. 23(3). no. 1239.

15. Purnama A.A.F., Nashiruddin M.I. SigFox-based Internet of Things Network Planning for Advanced Metering Infrastructure Services in Urban Scenario. IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). 2020. pp. 15–20.
16. Nashiruddin M.I., Purnama A.A.F. NB-IOT network planning for advanced metering infrastructure in Surabaya, Sidoarjo, and gresik. 8th International Conference on Information and Communication Technology (ICoICT). 2020. pp. 1–6.
17. Haider F., Zhang D., St-Hilaire M., Makaya C. On the Planning and Design Problem of Fog Computing Networks. IEEE Transactions on Cloud Computing. 2018. vol. 9. no. 2. pp. 724–736.
18. Zhang D., Haider F., St-Hilaire M., Makay C. Model and algorithms for the planning of Fog Computing Networks. IEEE Internet of Things Journal. 2019. vol. 6. no. 2. pp. 3873–3884.
19. Ebraheem A., Ivanov I.A. Internet of Things: Analysis of Parameters and Requirements. International Conference on Smart Applications, Communications and Networking (SmartNets). 2022. pp. 01–04.
20. Kaur S., Mir R.N. Base station positioning in Wireless Sensor Networks. International Conference on Internet of Things and Applications (IOTA). 2016. pp. 116–120.
21. REMCOM official web site. Available at: https://www.remcom.com/wireless-insite-em-propagation-software. (accessed 04.07.2023).
22. Mathworks official web site. Available at: https://mathworks.com/help/comm/ref/rfprop.raytracing.html. (accessed 04.07.2023).
23. Alqudah Y.A. On the performance of Cost 231 Walfisch Ikegami model in deployed 3.5 GHz network. The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE). 2013. pp. 524–527.
24. Correia L.M. A view of the COST 231-Bertoni-Ikegami model. 3rd European Conference on Antennas and Propagation. 2009. pp. 1681–1685.
25. Zhang J., Gentile C., Garey W. On the Cross-Application of Calibrated Pathloss Models Using Area Features: Finding a way to determine similarity between areas. IEEE Antennas and Propagation Magazine. 2019. vol. 62. no. 1. pp. 40–50.
26. Rackspace technology official web site. Available at: https://docs.rackspace.com/blog/different-types-of-oci-servers-in-the-cloud. (accessed 12.05.2023).
27. Google cloud official web site. Available at: https://cloud.google.com/compute/docs/machine-resource. (accessed 12.05.2023).
28. Amazon web services official web site. Available at: https://aws.amazon.com/ec2/instance-types. (accessed 12.05.2023).
29. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002. vol. 6. no. 2. pp. 182–197.
30. Yusoff Y., Ngadiman M., Zain A. Overview of NSGA-II for optimizing machining process parameters. Procedia Engineering. 2011. vol. 15. pp. 3978–3983.
31. Palaparthi A., Riede T., Titze I.R. Combining Multiobjective Optimization and Cluster Analysis to Study Vocal Fold Functional Morphology. IEEE Transactions on Biomedical Engineering. 2014. vol. 61. no. 7. pp. 2199–2208.
32. Blank J., Kalyanmoy D. Pymoo: Multi-objective optimization in python. IEEE Access. 2020. vol. 8. pp. 89497–89509.
33. Halicka K. Technology Selection Using the TOPSIS Method. Foresight and STI Governance. 2020. vol. 14. no. 1. pp. 85–96.

34. Sarraf A., Mohaghar A., Bazargani H. Developing TOPSIS method using statistical normalization for Selecting Knowledge Management Strategies. Journal of Industrial Engineering and Management. 2013. vol. 6. no. 4. pp. 860–875.

Ebraheem Ali — Ph.D. student, HSE University. Research interests: industrial internet of things, control theory, software technologies and development of information systems. The number of publications — 4. aebrakhim@hse.ru; 34, Tallinskaya St., 123592, Moscow, Russia; office phone: +7(495)772-9590 [15166].

Ivanov Ilya — Ph.D., Associate professor, academic supervisor of the programme (internet of things and cyber-physical systems), HSE University. Research interests: internet of things, cyber-physical systems, control and diagnostics of electronic devices. The number of publications — 105. i.ivanov@hse.ru; 34, Tallinskaya St., 123592, Moscow, Russia; office phone: +7(495)772-9590 [15166].

УДК 004.021 **DOI** 10.15622/ia.23.2.3

А. Эбрахим, И.А. Иванов

НА ПУТИ К АВТОМАТИЗИРОВАННОМУ И ОПТИМАЛЬНОМУ ПРОЕКТИРОВАНИЮ СИСТЕМ IIOT

Эбрахим А., Иванов И.А. **На пути к автоматизированному и оптимальному проектированию систем IIoT.**

Аннотация. В современном мире Интернет вещей стал неотъемлемой частью нашей жизни. Растущее число умных устройств и их повсеместное распространение усложняют разработчикам и системным архитекторам эффективное планирование и внедрение систем Интернета вещей и промышленного Интернета вещей. Основная цель данной работы – автоматизировать процесс проектирования промышленных систем Интернета вещей при оптимизации параметров качества обслуживания, срока службы батареи и стоимости. Для достижения этой цели вводится общая четырехуровневая модель туманных вычислений, основанная на математических множествах, ограничениях и целевых функциях. Эта модель учитывает различные параметры, влияющие на производительность системы, такие как задержка сети, пропускная способность и энергопотребление. Для нахождения Парето-оптимальных решений используется генетический недоминируемый алгоритм сортировки II, а для определения компромиссных решений на Парето-фронте – метод определения порядка предпочтения по сходству с идеальным решением. Оптимальные решения, сгенерированные этим подходом, представляют собой серверы, коммуникационные каналы и шлюзы, информация о которых хранится в базе данных. Эти ресурсы выбираются на основе их способности улучшить общую производительность системы. Предлагаемая стратегия следует трехэтапному подходу для минимизации размерности и уменьшения зависимостей при исследовании пространства поиска. Кроме того, сходимость оптимизационных алгоритмов улучшается за счет использования предварительно настроенной начальной популяции, которая использует существующие знания о том, как должно выглядеть решение. Алгоритмы, используемые для генерации этой начальной популяции, описываются подробно. Для иллюстрации эффективности автоматизированной стратегии приводится пример ее применения.

Ключевые слова: IoT, IIoT, NGSA-II, TOPSIS, облако, туманные вычисления, многокритериальная оптимизация, шлюз, пограничные устройства.

Литература

1. Официальный сайт Microsoft Azure. URL: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-the-cloud (дата обращения: 02.01.2023).
2. Basir R., Qaisar S., Ali M., Aldwairi M., Ashraf M.I., Mahmood A., Gidlund M. Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges. Sensors. 2019. vol. 19(21). no. 4807.
3. Цвиркун А.Д. Основы синтеза структуры сложных систем. М.: Наука, 1982. 200 с.
4. Цвиркун А.Д., Акинфиев В.К., Соловьев М.М. Моделирование развития крупномасштабных систем: (На примере топливно-энергетических отраслей и комплексов). М.: Экономика, 1983. 176 с.
5. Акинфиев В.К., Цвиркун А.Д. Методы и инструментальные средства управления развитием компаний со сложной структурой активов. М.: ИПУ РАН, 2020. 307 с.

6. Цвиркун А.Д., Акинфиев В.К., Филиппов В.А. Имитационное моделирование в задачах синтеза структуры сложных систем. М.: Наука, 1985. 173 с.
7. Потрясаев С.А. Синтез технологий и комплексных планов управления информационными процессами в промышленном интернете: дис. д-р тех. наук. СПБ., 2020.
8. Официальный сайт Международного общества автоматизации. URL: https://www.isa.org/intech-home/2019/march-april/features/rami-4-0-reference-architectural-model-for-industr (дата обращения: 13.09.2023).
9. Официальный сайт промышленного IoT-консорциума. URL: https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf (дата обращения: 12.09.2023).
10. Официальный сайт инициативы в области промышленной цепочки создания стоимости. URL: https://docs.iv-i.org/doc_161208_Industrial_Value_Chain_Reference_Architecture.pdf (дата обращения: 14.09.2023).
11. Srinidhi N.N., Kumar S.D., Venugopal K.R. Network optimizations in the Internet of Things: A review. Engineering Science and Technology, an International Journal. 2019. vol. 22. no. 1. pp. 1–21.
12. Ceselli A., Premoli M., Secci S. Mobile Edge Cloud Network Design Optimization. IEEE/ACM Transactions on Networking. 2017. vol. 25. no. 3. pp. 1818–1831.
13. Chimmanee K., Jantavongso S. Practical mobile network planning and optimization for Thai smart cities: Towards a more inclusive globalization. Research in Globalization. 2021. vol. 3. no. 100062.
14. Gava M.A., Rocha H.R.O., Faber M.J., Segatto M.E.V., Wortche H., Silva J.A.L. Optimizing Resources and Increasing the Coverage of Internet-of-Things (IoT) Networks: An Approach Based on LoRaWAN. Sensors. 2023. vol. 23(3). no. 1239.
15. Purnama A.A.F., Nashiruddin M.I. SigFox-based Internet of Things Network Planning for Advanced Metering Infrastructure Services in Urban Scenario. IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). 2020. pp. 15–20.
16. Nashiruddin M.I., Purnama A.A.F. NB-IOT network planning for advanced metering infrastructure in Surabaya, Sidoarjo, and gresik. 8th International Conference on Information and Communication Technology (ICoICT). 2020. pp. 1–6.
17. Haider F., Zhang D., St-Hilaire M., Makaya C. On the Planning and Design Problem of Fog Computing Networks. IEEE Transactions on Cloud Computing. 2018. vol. 9. no. 2. pp. 724–736.
18. Zhang D., Haider F., St-Hilaire M., Makay C. Model and algorithms for the planning of Fog Computing Networks. IEEE Internet of Things Journal. 2019. vol. 6. no. 2. pp. 3873–3884.
19. Ebraheem A., Ivanov I.A. Internet of Things: Analysis of Parameters and Requirements. International Conference on Smart Applications, Communications and Networking (SmartNets). 2022. pp. 01–04.
20. Kaur S., Mir R.N. Base station positioning in Wireless Sensor Networks. International Conference on Internet of Things and Applications (IOTA). 2016. pp. 116–120.
21. Официальный сайт REMCOM. URL: https://www.remcom.com/wireless-insite-em-propagation-software (дата обращения: 04.07.2023).
22. Официальный сайт Mathworks. URL: https://mathworks.com/help/comm/ref/rfprop.raytracing.html (дата обращения: 04.07.2023).
23. Alqudah Y.A. On the performance of Cost 231 Walfisch Ikegami model in deployed 3.5 GHz network. The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE). 2013. pp. 524–527.

24. Correia L.M. A view of the COST 231-Bertoni-Ikegami model. 3rd European Conference on Antennas and Propagation. 2009. pp. 1681–1685.
25. Zhang J., Gentile C., Garey W. On the Cross-Application of Calibrated Pathloss Models Using Area Features: Finding a way to determine similarity between areas. IEEE Antennas and Propagation Magazine. 2019. vol. 62. no. 1. pp. 40–50.
26. Официальный сайт Rackspace. URL: https://docs.rackspace.com/blog/different-types-of-oci-servers-in-the-cloud (дата обращения: 12.05.2023).
27. Официальный сайт Google Cloud. URL: https://cloud.google.com/compute/docs/machine-resource (дата обращения: 12.05.2023).
28. Официальный сайт Amazon Web Services. URL: https://aws.amazon.com/ec2/instance-types (дата обращения: 12.05.2023).
29. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002. vol. 6. no. 2. pp. 182–197.
30. Yusoff Y., Ngadiman M., Zain A. Overview of NSGA-II for optimizing machining process parameters. Procedia Engineering. 2011. vol. 15. pp. 3978–3983.
31. Palaparthi A., Riede T., Titze I.R. Combining Multiobjective Optimization and Cluster Analysis to Study Vocal Fold Functional Morphology. IEEE Transactions on Biomedical Engineering. 2014. vol. 61. no. 7. pp. 2199–2208.
32. Blank J., Kalyanmoy D. Pymoo: Multi-objective optimization in python. IEEE Access. 2020. vol. 8. pp. 89497–89509.
33. Halicka K. Technology Selection Using the TOPSIS Method. Foresight and STI Governance. 2020. vol. 14. no. 1. pp. 85–96.
34. Sarraf A., Mohaghar A., Bazargani H. Developing TOPSIS method using statistical normalization for Selecting Knowledge Management Strategies. Journal of Industrial Engineering and Management. 2013. vol. 6. no. 4. pp. 860–875.

Эбрахим Али — аспирант, Национальный исследовательский университет «Высшая школа экономики». Область научных интересов: промышленный интернет вещей, теория управления, технология разработки программных комплексов. Число научных публикаций — 4. aebrakhim@hse.ru; улица Таллинская, 34, 123592, Москва, Россия; р.т.: +7(495)772-9590 [15166].

Иванов Илья Александрович — канд. техн. наук, доцент, научный руководитель программы (интернет вещей и киберфизические системы), Национальный исследовательский университет «Высшая школа экономики». Область научных интересов: интернет вещей, киберфизические системы, контроль и диагностика электронных устройств. Число научных публикаций — 105. i.ivanov@hse.ru; улица Таллинская, 34, 123592, Москва, Россия; р.т.: +7(495)772-9590 [15166].