Author/s:
Sardar, AA;Rao, AS;Alpcan, T;Das, G;Routray, P;Manivannan, M;Palaniswami, M

Title:
Application-Aware Real-time Network Resource Allocation for Industry 5.0

Date:
2024-01-01

Citation:
Sardar, A. A., Rao, A. S., Alpcan, T., Das, G., Routray, P., Manivannan, M. & Palaniswami, M. (2024). Application-Aware Real-time Network Resource Allocation for Industry 5.0. IEEE Transactions on Cognitive Communications and Networking, 11 (4), pp.1-13. https://doi.org/10.1109/TCCN.2024.3504474.

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474

Application-Aware Real-time Network Resource Allocation for Industry 5.0

Asif Ahmed Sardar, Aravinda S. Rao, *Senior Member, IEEE,* Tansu Alpcan, *Senior Member, IEEE,* Goutam Das, Prasanna Routray, Manivannan M., and Marimuthu Palaniswami, *Life Fellow, IEEE*

***Abstract*—Industry 5.0 integrates advanced technologies like Automated Guided Vehicles (AGVs) and Augmented/Virtual Reality (AR/VR) with human expertise, requiring ultra-reliable communication for safe and efficient manufacturing. Network resource allocation in this context is challenging, demanding efficient support for diverse applications while meeting stringent performance targets, including 99.9999% availability. This study presents a novel application-aware resource allocation scheme for an Industry 5.0 system connected to a 5G network. Our approach dynamically adapts to industrial application states, bridging network optimization and real-time factory operations. The solution comprises (1) a learning-based framework for safety and productivity-conscious allocation policies, (2) a heuristic real-time resource allocation policy addressing the computational scalability problem of learning-based methods, and (3) statistical analysis for bandwidth requirement estimation. Simulation results show our method achieves 99.9999% availability while reducing bandwidth usage by nearly 50% compared to traditional methods. This work contributes to more efficient and scalable Industry 5.0 systems, potentially doubling the number of supported industrial components within the same network infrastructure.**

***Index Terms*—Industry 5.0, Augmented Reality, Virtual Reality, Maintenance 5.0, Network Resource Allocation**

I. Introduction

The 5G wireless network advancement presents a fast, reliable, and secure communication system. Integration of 5G network in the Industry 5.0 system can revolutionize the smart manufacturing processes regarding productivity, flexibility, and scalability [1]. A fast communication system supported by a 5G network makes it possible to offload the computationally expensive industry control functions from the factory floor to an edge server. Moving controllers from the factory floor to the edge server reduces hardware costs on the floor, increases flexibility and mobility of production processes, and allows efficient allocation of storage capacity and computational power according to the needs on the factory floor [2], [3], [4]. Another unique feature of the Industry 5.0 system is the capability of remote maintenance support, which does not require physical human presence on the factory floor, but rather s separate place near or within the factory premises [5]. This human-centric approach, where collaboration among human workers, industry robots, accompanying software, and emerging technologies such as Internet-of-Things (IoT), cloud computing, Augmented Reality (AR) and Virtual Reality (VR) are needed for maintenance in an Industry 5.0 setting is referred to as Maintenance 5.0 [5]. Integration of the 5G network in the Industry 5.0 system provides fast communication that allows industry workers and technicians remote access to the factory floor [1]. The reliable communication offered by the 5G network, along with a suitable resource allocation policy, is equivalent to allowing human workers to seamlessly interact with machines on the factory floor and the proper functioning of various industrial applications.

Industry 5.0 applications have two crucial requirements: satisfy *availability* and increase *scalability*. The performance metric *Availability* is defined as the *percent of the time that the end-to-end communication service is delivered* on the factory floor. The 3GPP Study on Communication for Automation has set the availability target to be greater than 99.9999% (commonly termed *six-nines availability*) [1]. This availability directly affects industrial applications. For example, delay in video transmission can lead to poor motion control decisions and possible collisions on the factory floor, and delays in haptic controls can disrupt remote human access to the factory floor, risking faulty decisions by human workers. Therefore, it is necessary to implement local area networks within factory premises and provide a private 5G network uninterrupted by external factors [1] to meet the performance goals of an industrial automation process. A proper network resource allocation policy is critical to ensure the optimal use of available network resources to avoid wastage and guarantee high performance. Industry 5.0 environments are highly dynamic and heterogeneous. Designing an allocation policy capable of sustaining the Quality-of-Service (QoS) requirements with a very high degree of availability is very important for the successful operation of an Industry 5.0 system. Although some of the existing works [6], [7], [8], [9], [10], [11] in the literature discuss network resource allocation problem in the Industry 5.0 system, none of them show if those policies can ensure 99.9999% availability. Our work shows that some of these policies fail to achieve this requirement.

The *scalability* of an Industry 5.0 system is essential for effectively handling fluctuations in customer demand [1], [12]. A scalable industrial system possesses the capability to

This research was supported in part by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP190102828).

A. A. Sardar, A. S. Rao, T. Alpcan, and M. Palaniswami are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria - 3010, Australia (e-mail: asardar@student.unimelb.edu.au, aravinda.rao@unimelb.edu.au, tansu.alpcan@unimelb.edu.au, palani@unimelb.edu.au).

G. Das is now with the G.S. Sanyal School of Telecommunications, Indian Institute of Technology Kharagpur, India. (e-mail: gdas@gssst.iitkgp.ac.in).

P. Routray and Manivannan M. are with the Touch Lab, Center for Virtual Reality and Haptics, Indian Institute of Technology Madras, India (e-mail: prasanna.routray97@gmail.com, mani@iitm.ac.in).

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474

adjust production capacity effortlessly through the addition or removal of manufacturing resources. The scalability requirements of the Industry 5.0 system arise from the need to accommodate more industrial components to the 5G network and the computational complexity of allocating network resources to different industrial components. So, we have to successfully meet the performance requirements while using fewer network resources to improve the scalability of an industrial process and make the resource allocation method computationally efficient. In conventional communication systems, data injection sources into the wireless network lack regulation and operate independently of the network allocation process. In an industrial system where the performance target is stringent, the resource allocator should be aware of the state of the industrial applications and allocate resources according to the needs of various industrial applications. In this work, we design a computationally efficient application-aware resource allocator that improves the usage efficiency of network resources, thus allowing more industrial components to be added to the wireless system. In the following subsections, we shall discuss some of the relevant works and finally discuss the novelties of our work.

A. Related works

In recent literature, the challenges associated with remote human interaction via AR-VR in Industry 5.0 systems have been explored - (a) safety and comfortability of the remote human workers [13], [14], (b) providing an immersive experience to human workers aiming to enhance the productivity of the industrial applications [15], [16]. Integration of 5G has led to extensive studies on moving industrial controller applications to the edge server, such as - designing architectures for edge-based computing for AR-VR applications [17], [18] and deploying motion controllers for the Automated Guided Vehicles (AGVs) on the edge server [19]. These objectives are intricately related to the availability of sufficient network resources and a corresponding allocation policy, which are not discussed in those works. Our work aims to bridge that gap in the literature. Some network resource allocation policies for Industry 5.0 system are available in literature [8], [6], [7], [9], [10], [11]. None of these works consider the randomness associated with remote human intervention on the factory floor. These resource allocation policies do not consider the state of the industrial application while allocating bandwidth to different components. Our resource allocation method can achieve 99.9999% availability using nearly 50% fewer resources compared to these existing policies.

Very few works in literature have connected the network resource allocation problem with the varying demands of the industrial applications [20], [21]. In [20], the authors propose a learning-based channel selection framework aware of the data backlog, energy capacity, and transmission reliability of the industrial components connected to the wireless network. This resource allocation method does not consider the cycle times of the industrial applications and does not ensure 99.9999% availability. The authors of [21] design a dynamic network management framework that can tackle the varying constraints on the round-trip delays corresponding to different industrial applications. None of these works shows how these resource allocation methods can achieve the 99.9999% availability target required by Industry 5.0 systems. *In our work, we develop a resource allocation method aware of the states of the industrial applications and can achieve* 99.9999% *availability.* Most existing works on the scalability of industrial systems aim to design mathematical models for increasing industrial capacity to meet the market demand [22], [23]. In [24], the authors proposed a novel access protocol for low-power wide area networks that can accommodate more IoT nodes under a single base-station and showed its suitability in an industrial setting. The effect of increased scalability on the availability of the industrial systems was not considered in [24]. Our work explores the mutual effects between scalability and availability, which is missing in the existing works.

B. Contributions

In our previous work [25], we analyzed an Industry 4.0 system with a VR-enabled maintenance training system. Our current work on an Industrial 5.0 system with AR-VR-based remote access capability has some key differences in comparison to [25] as listed below:

(a) **Scenario:** In [25], VR 360° video is sent from the factory floor to the remote maintenance training room throughout the industrial operation. Meanwhile, in current work, remote access events may happen at any point in time to change industry parameters to provide customized services or maintenance purposes. So, the video feed must be sent only during the access events.

(b) **Human-machine interactions:** In [25], no real-time maintenance action events are considered during the lifetime of the industrial system. So, human-machine interaction is absent in that scenario. However, in this work, the human workers intervene in real-time according to the needs of the factory floor. So, the network resource allocation policy must consider whether or not a remote access event is happening on the factory floor.

(c) **Solution framework:** In [25], we have formulated an optimization-based framework for designing resource allocation policy. However, in this work, the inherent spatio-temporal randomness associated with the remote access events makes it impossible to design an optimization-based resource allocation scheme. So, we intend to focus on learning-based methods convenient for dynamic systems.

(d) **Selective allocation to industrial components**: In [25], we allocate resources to all the cameras to create VR 360° view of the whole factory floor throughout the Industrial process. However, in this work, we selectively choose the cameras and AGVs for resource allocation based on the requirements on the factory floor.

This work considers two crucial industrial applications of an Industry 5.0 system: (1) motion control of the AGVs and (2) remote human access on the factory floor. In the motion controller application, AGVs interact with the motion controller on the edge server, which is a machine-to-machine interaction. In contrast, remote human access via the AR-VR system is a human-machine interaction involving remote industrial workers and mobile robots on the factory floor. We can influence the motion controller by managing video input and selectively sending motion decisions to AGVs while ensuring safety on the factory floor. However, haptic control decisions are made by human workers based on information received from the mobile robot operating on the factory floor. So, for remote human access events, human-to-machine interaction occurs, and we do not have a direct or indirect way of controlling the interaction without potentially affecting remote access by industrial workers. We design a network resource allocation method that looks into the requirements of these industrial applications.

The **main contributions** in this work are as follows:

- **Application-aware resource allocation:** We connect the network resource allocation problem with the performance of industrial applications on the floor. This novel approach to making the allocation process application-aware increases the efficiency of resource usage by nearly 50% compared to existing works [8], [6], [7] and improves the scalability of the factory by accommodating more components to the wireless communication system.
- **Statistical method for resource estimation:** We propose a statistical approach to find the lower and upper bounds of the resource requirement for 99.9999% availability for motion control and remote access applications, which is new in Industry 5.0 settings.
- **Computationally efficient heuristic policy:** We design an application-aware heuristic resource allocation policy with quadratic time complexity concerning the number of AGVs and linear time complexity in the number of cameras. It resolves the scalability issue of learning-based methods by reducing the search space for possible resource allocation decisions.
- **Benchmark against Learning-based policy:** The randomness associated with remote human interaction with machines on the factory floor, as well as the variability of the channel conditions inside the factory, means that the best possible sequence of resource allocations over time that can achieve the safety and delay requirements, cannot be obtained by traditional optimization problems. We use several learning-based methods to benchmark our real-time policy and highlight the computational benefit of the heuristic method.

II. System Model

Consider a fully automated factory floor with AR-VR-assisted remote human access to the factory floor, shown in Fig. 1. There are M AGVs moving between their respective source and destination on predefined paths, each having a maximum achievable velocity of $v_{\max}$. C number of cameras are mounted on the walls of the factory floor at strategic locations for monitoring the factory floor. Each camera has a radius of observation beyond which it cannot monitor the floor. It is assumed that every part of the factory floor is observable with at least one camera. AGVs must maintain a minimum distance to prevent collision events, called the *collision avoidance radius* and denoted by r_a. AGVs have sensors onboard to avoid imminent collisions. As shown in Fig. 1, human workers can access the factory from the remote workers' room via VR headsets that support telepresence. A remote access event can happen for various reasons such as - maintenance purposes, tuning industry parameters etc. During a remote access event, the industry worker communicates with the mobile robot on the factory floor via various haptic sensors. The mobile robot has an object detector, impedance rendering capability, and a camera that can stream video from the factory floor to the remote workers' room [26]. Motion control of the AGVs and the mobile robot is performed by a motion controller, based on the video feed obtained from the monitoring cameras. The motion controller is placed on an edge server.

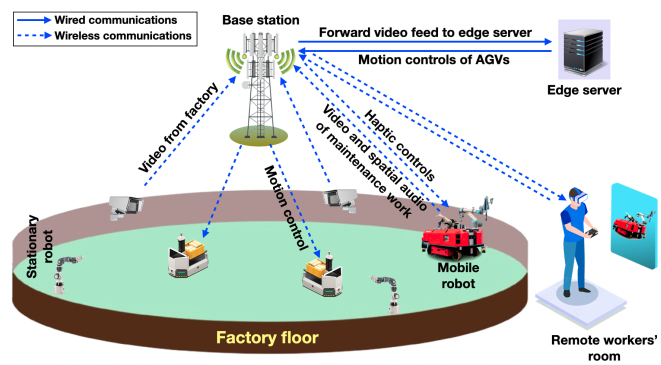


Fig. 1: An Industry 5.0 system with AR-VR-assisted access system for industrial workers at the remote room.

A 5G base station is placed on the industry premises to provide a private and uninterrupted communication system on the factory floor [1]. This work uses the 5G New Radio (NR) access technology for wireless communication on the factory floor. According to 3GPP TS 38.101-1 V18.3.0, a Resource Block (RB) is a block of 12 consecutive subcarriers in the frequency domain. The total number of RBs available to the industrial system is denoted by N_{RB}, which is bounded by the physical structure of 5G NR. The 5G base station allocates RBs to the industrial system components at a time interval of T_S. In the following sections, we develop a statistical method for estimating resource requirements, define the key parameters that describe the industrial system mentioned above, and design methods to find a suitable network resource allocation policy that can satisfy the end-to-end delay requirements of the motion control application and the remote access events. The performance of the Industry 5.0 operation is characterised by the cycle times of the involved applications and the overall availability of the system.

A. Cycle Time

The *cycle time* of a particular industrial application consists of the time to generate a control sequence from the controller affecting that specific application, transmission delay over the wireless network, and the time to receive confirmation that the control command has been implemented [1]. According to the

3GPP Study on Communication for Automation in Vertical Domains (TR 22.804), the end-to-end delay or cycle times of Industry 5.0 applications must be supported by the wireless communication system on the factory floor 99.9999% of time [1]. The cycle times of the AR-VR based remote access and motion control applications are denoted by Δ_{ARVR} and Δ_{Motion}, respectively.

The cycle time for the AR-VR assisted remote access events includes the communication delay between the remote workers' room and the mobile robot, the control delay incurred by human perception, and the delay in implementing haptic controls at the mobile robot. The average delay of human actions and the average delay in implementing haptic controls are represented by δ_{Perc} and δ_{Haptic}. As a result, the maximum allowable transmission delay for the remote human access application can be denoted as $\Delta_1 = \Delta_{\text{ARVR}} - \delta_{\text{Perc}} - \delta_{\text{Haptic}}$. For the motion control application, the end-to-end delay consists of the maximum uplink delay for transmitting camera video feed to the edge server, the processing delay to decide the motion controls, the downlink transmission delay to send the controls to the AGVs, and the delay to implement these controls. The average processing delay and the average delay in implementing the motion controls are denoted by δ_{Proc} and δ_{Motion}, respectively. So, the maximum allowable transmission delay (denoted by Δ_2) for the motion control application is given by $\Delta_2 = \Delta_{\text{Motion}} - \delta_{\text{Proc}} - \delta_{\text{Motion}}$.

B. Availability of Industrial Applications

Availability in the context of 'communication service availability' indicates that a system is deemed available only when it meets all the QoS criteria, including latency, data rate, etc. As such, it is usually measured by the proportion of time in which a system functions correctly. Industry environments demand extremely high availability to ensure uninterrupted production processes. This work considers two industrial applications whose availability is calculated as follows.

- *Motion control application:* Availability of the motion control application depends on the safety on the factory floor. In the simulation process, availability (99.9999%) corresponding to the motion control of the AGVs is calculated by the percentage of time duration of the industrial process in which no possible collision event occurs. By achieving this high level of availability, we lower the chance of end-to-end delay violation for the application to nearly zero, ensuring the desired control performance [1].
- *Remote access application:* Availability of the remote access events is defined as the percentage of time cycle time constraint if the application is not violated.

Clearly, to maintain a high degree of availability, we have to satisfy the cycle times of different industrial applications. In the following section, we develop a real-time network resource allocation policy aiming to meet the performance requirements of the industrial system. Based on this policy, we also show how we can estimate the resource requirement of the Industry 5.0 system. To the best of our knowledge, this work represents the first attempt to quantify *availability* specifically for motion control and remote access applications in industrial settings. By proposing these measurement methods, we aim to provide a more precise and application-specific understanding of availability in Industry 4.0 contexts.

III. Real-time Network Resource Allocation

In this section, we develop a real-time network resource allocation policy aiming to satisfy the cycle time constraints of the motion control and remote access applications. By prioritizing cycle time constraints and ensuring timely delivery of critical control signals, our method maintains performance as system complexity grows, making it particularly well-suited for dynamic, safety-critical industrial environments where predictable, explainable outcomes are crucial. Later in this section, we design a statistical method, heavily relying on the allocation policy, to estimate the resource requirement of the Industry 5.0 system.

A. Approximate Delay Functions

To quantify the cycle times defined in Section II-A, we need to approximate the various delay components. We approximate transmission delay curves using the Shannon-Hartley theorem to design a heuristic application-aware network resource allocation method. In our earlier work [25], we have shown that these approximated wireless transmission delays work well in an industry setting with access to the private 5G network. The parameters needed to calculate the approximate delay functions are given in Table I. We can approximate $\delta_{\text{Rem}}\left(z_t; g_t^{\text{Rem}}\right)$ as $s_{\text{Rem}} \Big/ \left[12\Delta f z_t \log_2\left(1 + \frac{P_1 g_t^{\text{Rem}} r_{\text{Rem}}^{-\alpha}}{12\Delta f z_t N_0}\right)\right]$. Expressions of the other approximate delay curves can be written similarly. For the t^{th} round of RB allocation, the set of camera indices from which the video feed is sent to the edge server is denoted by $\mathcal{S}^t_{\text{CAM}}$ and the set of indices of the AGVs to which motion controls are sent is denoted by $\mathcal{S}^t_{\text{AGV}}$. The uplink transmission delay from the camera system to the 5G base station is given by $\delta^{\text{Up},t}_{\text{CAM}} = \max_{k \in \mathcal{S}^t_{\text{CAM}}} \delta^{k,t}_{\text{CAM}}\left(x^k_t; h^k_t\right)$. Similarly, the downlink transmission delay to send motion controls from the 5G base station to selected AGVs can be represented as $\delta^{\text{Down},t}_{\text{AGV}} = \max_{i \in \mathcal{S}^t_{\text{AGV}}} \delta^{i,t}_{\text{AGV}}\left(y^i_t; g^i_t\right)$. The following proposition is crucial for designing heuristic methods for network resource allocation.

Proposition 1. *Let $x_1, \ldots, x_n$ be n positive real numbers such that $x_1 + \ldots + x_n \leq N$. Consider n continuous and differentiable decreasing functions $f_i(x_i)\ \forall\ i \in [1, n]$. If the minimum value of $\max_{i=1}^n f_i(x_i)$ is v at $(x'_1, \ldots, x'_n)$, then $x'_1 + \ldots + x'_n = N$ and $f_1(x'_1) = \ldots = f_n(x'_n) = v$.*

The proof of the proposition is given in our earlier work [25].

B. Heuristic Methods for the Selection of Cameras and AGVs

In contrast to our previous work [25], we do not allocate RBs to all AGVs and cameras in every round of RB allocation. We allocate RBs to those AGVs, the motions of which must be controlled appropriately to prevent possible collisions in

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474

the near future. We also need to allocate RBs to the cameras with those critical AGVs in their line of sight so that video feed from those cameras can be sent to the edge server and correct motion control decisions can be made.

- **Selecting AGVs:** The AGVs on a collision course with obstacles are chosen. The distance-based selection of the AGVs reduces the chances of a collision.
- **Selecting cameras:** The set of cameras covering the motion of selected AGVs are chosen, which translates to *set cover problem*, which is an NP-hard problem with no polynomial time algorithm [27].

C. Real-time Algorithm for Resource Allocation and its Computational Complexity

At each allocation round, we divide the available RBs between the two applications, motion control and AR-VR-assisted remote access. The fraction of RBs allocated for remote access is denoted by ψ. In Algorithm 1, we use the *binary search* method to tune the value of ψ to satisfy the cycle time constraints of both applications. If enough network resources are unavailable to the industrial system, satisfying one or both delay constraints will be impossible. We use Subroutine 1 to allocate network resources for the remote access application and Subroutine 2 to allocate RBs to the AGVs and cameras.

Algorithm 1: Policy for t^{th} round of RB allocation

Calculate maximum allowable transmission delays for Remote access and Motion control applications (see Section II-A).
if No remote access event is going on **then**
    Use Subroutine 2 and stop Algorithm 1.
The sets of selected cameras and AGVs (see Section III-B) are $\mathcal{S}^t_{\text{CAM}}$ and $\mathcal{S}^t_{\text{AGV}}$.
Allocating chunks of RBs to different applications:
• Fraction of RBs allocated to the remote access application = ψ ($0 < \psi < 1$). Initialize ψ to 0.5.
• RBs allocated to remote access = N_{Rem} = $\lceil \psi N_{\text{RB}} \rceil$
• Rest (N_{Motion}) are allocated for motion control
RB_{Previous} and RB_{Current} are empty resource allocation vectors.
do
    Copy RB_{Current} to RB_{Previous}.
    Use Subroutine 1 and Subroutine 2 with RB constraints N_{Rem} and N_{Motion} respectively.
    Store the resource allocations found by Subroutine 1 and Subroutine 2 in RB_{Current}.
    If both allocations are invalid, not enough RBs are available. Break the while loop.
    If one allocation is valid, adjust ψ using *Binary Search* to make room for the other allocation.
while RB_{Previous} and RB_{Current} are not equal;
Return the final RB allocation.

(I) *Subroutine 1:* We allocate a fixed amount of RBs to the remote workers' room and the rest to the mobile robot. We iterate the process until an allocation is found that satisfies constraints on RBs and end-to-end delay. The time complexity of Subroutine 1 is $O(N_{\text{RB}})$.

(II) *Subroutine 2:* We allocate a fixed number of RBs for communication from cameras to the base station and the rest to the AGVs to receive motion controls from the base station. We invoke Subroutine 3 (described in the following point) to find RB allocation to achieve the minimum uplink and downlink transmission delays.

(III) *Subroutine 3:* From Proposition 1, minimum transmission delays can be achieved if the individual component (see Section III-A) dictating uplink delay (similarly for the downlink) has the same value. All transmission delays have the form $a_1/(x\log(1+\frac{a_2}{x})) = \Delta_{\text{t}}$, where x corresponds to the number of RBs for a particular transmission. The solution to this equation is given by $x = -\dfrac{a_1 a_2}{a_1 + a_2\Delta_{\text{t}}\text{ProductLog}\left(-\frac{a_1\exp(-a_1/(a_2\Delta_{\text{t}}))}{a_2\Delta_{\text{t}}}\right)}$, where the ProductLog$(z)$ function, also known as the Lambert W function, gives the solution for $we^w = z$. We fix a target transmission delay (both uplink and downlink) in the range (ϵ, Δ_2) and try to find a corresponding RB allocation based on Proposition 1. ϵ is a very small positive real number and Δ_2 is defined in Section II-A. We use binary search to pinpoint the value of the uplink and downlink transmission delays that allow RB allocation without violating the RB constraint. The time complexity of III-C(III) is $O(C+M)$. So, the total time complexity of III-C(II) is $O(N_{\text{RB}} \cdot (C+M))$.

Now, we evaluate the time complexity of our real-time allocation RB allocation policy (Algorithm 1). The time complexity for selecting the cameras and AGVs depends on the particular heuristic method (see Section III-B). The distance-based selection of AGVs and greedy algorithm for set cover-based selection of cameras have time complexity $O(M^2)$ and $O(CM)$. Therefore, the overall complexity becomes $O(M^2 + CM + N_{\text{RB}} \cdot (C+M))$, which makes it computationally more scalable than RL-based policies.

As the individual resource allocation takes positive integer values and the total available resources is finite, the space of all possible resource allocations is finite. We prove the convergence of our heuristic algorithm by showing the convergence of Subroutines 3, 2, 1 and Algorithm 1 in the following points.

1) *Convergence of Subroutine 3:* The search intervals of the binary search to find the minimum uplink and downlink transmission delays using available resources, are bounded and closed. In each iteration, the intervals are halved. As the number of resource blocks allocated to each component are positive integer, the space of valid resource allocation is finite. So, the halving process of the binary search will find the minimum transmission times using available network resources in $O(\log N_{\text{RB}})$ steps.
2) *Convergence of Subroutine 2:* Subroutine 2 invokes Subroutine 1 $O(N_{\text{Motion}})$ times [N_{Motion} is defined in Algorithm 1], and find a resource allocation using available resources to minimize cycle time of motion control application.

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474

3) *Convergence of Subroutine 1:* Subroutine 1 runs for $O(N_{\text{Main}})$ steps [N_{Main} is defined in Algorithm 1], and find a resource allocation scheme that satisfies the cycle time constraint of the remote access event. If enough resource is not available, Algorithm 1 is used to adjust N_{Main}.
4) *Convergence of Algorithm 1:* In Algorithm 1, we partition the total resources between motion control and remote access applications. The search interval $[0, N_{\text{RB}}]$ is bounded and closed. At each step of Algorithm 1, we adjust the partition, use Subroutine 1 to satisfy the cycle time constraint of remote access event and Subroutine 2 to minimise the cycle time of motion control application. Due to the finite search space of resource allocation, at each iteration, we keep halving the interval in which the optimal partition lies and arrive at a final resource allocation in $O((N_{\text{RB}} + \log N_{\text{RB}}) \log N_{\text{RB}})$ steps. If enough resources are unavailable for the Industry 5.0 system, one or both must happen: (i) Subroutine 1 fails to maintain cycle time constraint, (ii) Minimum transmission time found by Subroutine 2 exceeds the cycle time of motion control application.

D. Resource Requirement Analysis

During the planning phase of an Industry 5.0 system, it is essential to get a rough estimate of the number of RBs necessary (N_{RB}) to satisfy performance metrics. The stringent performance demands of the Industry 5.0 system lead us to assess the peak bandwidth needs on the factory floor, notably during remote human access, when reliable communication must be supported between the remote workers' room and the mobile robot. Various system parameters, Probability Distribution Functions (PDFs) and Cumulative Distribution Functions (CDFs) necessary for this analysis are given in Table I.

TABLE I: Parameters for calculating transmission delays

Description	Notation
Number of AGVs on the factory floor	M
Number of RBs assigned to the Industry 5.0 system	N_{RB}
Subcarrier spacing of the RBs	Δf
Path-loss factor inside the factory	α
Noise spectral density inside the factory floor	N_0
Packets sizes of camera video, motion control, mobile robot and remote workers' room	s_{Camera}, s_{Motion}, s_{Mob}, s_{Rem}
Tx. powers of 5G base station, cameras, mobile robot and remote workers' room	P_{B}, P_{Cam}, P_{Mob}, P_{Rem}
Distances from 5G base station to k^{th} camera, i^{th} AGV, mobile robot and remote workers' room	r^k_{Cam}, $r^{i,t}_{\text{AGV}}$, r^t_{Mob}, r_{Rem}
PDFs governing Tx. delays from base station to 5G base station to mobile robot, remote workers' room, k^{th} camera, i^{th} AGV	$p_{\text{Mob}}(\cdot)$, $p_{\text{Rem}}(\cdot)$, $p^k_{\text{CAM}}(\cdot)$, $p^i_{\text{AGV}}(\cdot)$
CDFs governing Tx. delays from base station to 5G base station to mobile robot, remote workers' room, k^{th} camera, i^{th} AGV	$P_{\text{Mob}}(\cdot)$, $P_{\text{Rem}}(\cdot)$, $P^k_{\text{CAM}}(\cdot)$, $P^i_{\text{AGV}}(\cdot)$
Tx. delays from the base station to 5G base station to the mobile robot, remote workers' room, k^{th} camera, i^{th} AGV	$\delta_{\text{Mob}}\left(w_t; g^{\text{Mob}}_t\right)$, $\delta_{\text{Rem}}\left(z_t; g^{\text{Rem}}_t\right)$, $\delta^{k,t}_{\text{CAM}}\left(x^k_t; h^k_t\right)$, $\delta^{i,t}_{\text{AGV}}\left(y^i_t; g^i_t\right)$

Resource block estimation for remote access application: The end-to-end delay violation probability for the remote access application can be written as (refer to Section II-A and Table I for the details of the parameters)

$$
\begin{aligned}
P_{\text{Rem}} &= P\left[\text{End-to-end delay} \leq \Delta_{\text{ARVR}}\right] \\
&= P\left[\text{Total transmission delay} \leq \Delta_1\right] \\
&= \int_{y=0}^{\Delta_1} p_{\text{Rem}}(y) \left(\int_{x=0}^{\Delta_1 - y} p_{\text{Mob}}(x) dx \right) dy \qquad (1) \\
&= \int_{y=0}^{\Delta_1} p_{\text{Rem}}(y) P_{\text{Mob}}(\Delta_1 - y) dy.
\end{aligned}
$$

This end-to-end delay violation probability must be kept below 10^{-6} to satisfy the 99.9999% availability requirement of the Industry 5.0 system. The number of RBs needed to achieve this target is denoted by $N^{\text{Rem}}_{\text{RB}}$. To estimate $N^{\text{Rem}}_{\text{RB}}$, we use the following steps:

- **Set the target probability:** P˙Rem must be kept below 10^{-6} to satisfy the 99.9999% availability requirement of the Industry 5.0 system.
- **Initialize the amount of resource:** Initially, we set the value of $N^{\text{Rem}}_{\text{RB}}$ to $\hat{N}_1$. We allocate the RBs to the industrial components involved with remote access events using Algorithm 1, described in Section III-C.
- **Adjusting the amount of resource:** We calculate P_{Rem} using equation 1. For calculating the transmission delay between the mobile robot and the base station, the distance from the mobile robot to the base station is set to be the maximum distance of all possible locations (where human workers can remotely access) to the base station. This maximises the transmission delay, ensuring the need for the most amount of network resources to achieve a particular end-to-end delay violation probability. If P_{Rem} exceeds 10^{-6}, we double the value of $\hat{N}_1$.
- **Optimise the value of $N^{\text{Rem}}_{\text{RB}}$:** Once the desired P_{Rem} is achieved, we use the binary search method in the range $[1, \hat{N}_1]$ to find the exact value of $N^{\text{Rem}}_{\text{RB}}$ that can achieve the target. This final step guarantees that the minimum amount of resources is used to achieve the desired end-to-end delay violation probability.

Resource block estimation for motion control application: The amount of RBs needed to maintain the delay violation probability of the motion control application to be less than 10^{-6} is denoted by $N^{\text{Motion}}_{\text{RB}}$. The steps to calculate the lower and upper bound of $N^{\text{Motion}}_{\text{RB}}$ can be summarised in the following steps:

- **Simulate AGV movements:** We simulate the movements of AGVs on the factory floor and track the number of possible collision events over time.
- **Identify critical scenarios:** We pinpoint when most AGVs are involved in potential collision events and determine when most cameras are needed for floor monitoring. The sets of indices of AGVs and cameras in these scenarios are denoted by $\tilde{\mathcal{S}}_{\text{AGV}}$ and $\tilde{\mathcal{S}}_{\text{CAM}}$ respectively.
- **Delay probability calculation:** The CDFs that govern the overall uplink and downlink delays can be calculated as $P_{\text{UL}}(\cdot) = \prod_{k \in \tilde{\mathcal{S}}_{\text{CAM}}} P^k_{\text{CAM}}(\cdot)$ and $P_{\text{DL}}(\cdot) = \prod_{i \in \tilde{\mathcal{S}}_{\text{AGV}}} P^i_{\text{AGV}}(\cdot)$. The corresponding PDFs are denoted by $p_{\text{UL}}(\cdot)$ and $p_{\text{DL}}(\cdot)$. The end-to-end delay violation

probability for the motion control application can be written as $P_{\text{Motion}} = \int_{y=0}^{\Delta_2} p_{\text{UL}}(y) P_{\text{DL}}(\Delta_2 - y) dy$.

- **Resource block estimation:** To calculate the upper bound of $N_{\text{RB}}^{\text{Motion}}$, we use $\tilde{\mathcal{S}}_{\text{AGV}}$ and $\tilde{\mathcal{S}}_{\text{CAM}}$ and apply Algorithm 1 (described in Section III-C). Similarly, for the lower bound calculation we use the scenarios when the fewest non-zero number of AGVs are in a collision course and the fewest number of monitoring cameras are needed, and once again apply Algorithm 1.

Combining both, we have a lower and upper bound of the number of RBs (N_{RB}) needed to satisfy the cycle times of remote human access events and motion control applications with 99.9999% certainty. To evaluate the performance of the heuristic method developed in this section, it is important to quantify the deviation from the optimal sequence of the resource allocation process. In Section IV, we model our system in a Reinforcement Learning framework which is suitable for handling complex and dynamic industrial systems. In Section V-A, we use this learning-based framework as the benchmark for performance comparison with Algorithm 1.

IV. Modelling Industry 5.0 System in a Reinforcement Learning Framework

In this section, we use various Reinforcement Learning (RL) methods as the benchmarks for the proposed real-time allocation policy. RL methods offer a powerful framework for addressing complex, dynamic decision-making problems in Industry 5.0 systems. Their ability to learn optimal policies through interaction with the environment makes them particularly well-suited for network resource allocation tasks. However, RL approaches often come with significant computational overhead, especially in large-scale industrial settings. By comparing our computationally efficient approach against RL-based methods, we establish a robust benchmark that not only validates the efficiency of our algorithm but also highlights its practical advantages in real-time industrial applications. This comparison provides insights into the trade-offs between solution quality and computational efficiency, demonstrating how our algorithm balances performance with the stringent time constraints of dynamic Industry 5.0 environments.

A. Application-Aware System State description

In this subsection, we define the *state* of the industrial system. The state description includes the variables necessary to enumerate our system's two objectives, namely (i) maintaining safety on the factory floor and (ii) satisfying end-to-end delay requirements of the motion control application and the AR-VR-assisted remote access applications. The *state* at the beginning of t^{th} iteration of RB allocation is denoted by s_t. The individual components of the state are described in the following points.

- Positions of the AGVs and the mobile robot at the beginning of t^{th} iteration of RB allocation are included in the state description, as any collision event can be judged by these parameters. The set of positions of the AGV and the mobile robot is denoted by $\mathbf{P}_t = \{\mathbf{p}_t^m, \mathbf{p}_t^1, \mathbf{p}_t^2, \ldots, \mathbf{p}_t^M\}$.
- The channel conditions inside the factory floor influence the wireless communication system. Therefore, the channel gain parameters should be in the state description. The channel gain parameters of our industrial system are denoted as follows.
  - The channel gain from the k^{th} camera to the 5G base station: h_t^k
  - The channel gain from the 5G base station to the i^{th} AGV: g_t^i
  - The channel gain of the channel between the maintenance room and the 5G base station: g_t^{Main}
  - The channel gain of the channel between the mobile robot and the 5G base station: g_t^{Mob}
- A binary variable (denoted by ϕ_t) is included in the state description to indicate whether a remote access event is happening on the factory floor. ϕ_t is set to 1 during a remote access event.
- In the case of an ongoing remote access event, the amount of force applied at the place of access and the tolerable threshold for force is denoted by F_t and $\hat{F}_t$, respectively. These parameters are set to zero when no remote access event occurs on the factory floor.

B. Actions - RB allocation parameters

At the beginning of each time slot, the 5G base station allocates RBs to the components of the Industry 5.0 system. The parameters related to the RB allocation must be non-negative integers. These parameters are summarized as follows.

- Number of RBs allocated to the k^{th} camera: x_t^k
- Number of RBs allocated to the i^{th} AGV: y_t^i
- Number of RBs allocated to the mobile robot: w_t
- Number of RBs allocated to the maintenance room: z_t

Our resource allocation framework is based on the 5G New Radio resource allocation architecture, where actions represent the allocation of whole resource blocks to industrial components. The action space is finite and discrete due to three key properties: resource blocks are allocated as indivisible units, allocations must be non-negative integers, and there's an upper bound on total available resources. The values of the parameters listed above are found by the learning-based framework. The total RBs allocated to the factory floor components must not exceed the maximum available RBs. The action parameters corresponding to the t^{th} iteration of RB allocation must satisfy the following inequality constraint.

$$\begin{aligned} &\sum_{k=1}^{C} x_t^k + \sum_{i=1}^{M} y_t^i + w_t + z_t \leq N_{\text{RB}} \\ &\iff f(X_t, Y_t, w_t, z_t) \leq N_{\text{RB}}, \\ &\text{where } X_t = \{x_t^1, \ldots, x_t^C\} \text{ and } Y_t = \{y_t^1, \ldots, y_t^M\} \end{aligned} \tag{2}$$

C. Reward function

The reward function is defined as the negative of the cost function. This reward function must be carefully designed to satisfy the objectives of our industrial system by taking the proper sequence of actions. We describe various penalties that constitute the cost function. Finally, we can add these penalties and negate the value to calculate the reward.

- We penalize the actions if they cause collision events. The number of collision events among the AGVs, between the AGVs and the mobile robot, between stationary objects and the AGVs, and between stationary objects and the mobile robot is denoted by c_t^{11}, c_t^{12}, c_t^{13} and c_t^{14}. The corresponding penalty weights are μ_{11}, μ_{12}, μ_{13}, and μ_{14}. The combined penalty can be written as

$$p_1(\mathbf{P}_t) = \sum_{i=1}^{4} \mu_{1i} c_t^{1i}. \quad (3)$$

In [25], we defined *efficiency* (denoted here by η_t) as the total distance covered by the AGVs in a time frame divided by the distance that the AGVs can move at their maximum speed, and can be written as

$$\eta_t = \frac{1}{W} \left[\sum_{k=1}^{W-1} \eta_{t-k} + \frac{\sum_{i=1}^{M} d_t^i}{v_{\max} M T_S} \right]. \quad (4)$$

The numerator of the second term denotes the total distance covered by M AGVs between t^{th} and $(t+1)^{th}$ RB allocation. If a collision occurs, the involved AGVs stop operating until external intervention, reducing the total distance covered and efficiency. Avoiding collisions ensures all AGVs can move smoothly around the factory, maximizing the distance covered and increasing efficiency. Thus, minimizing collision events maximizes efficiency. The set of distances covered is $\mathbf{D}_t = \{\mathbf{d}_t^1, \ldots, \mathbf{d}_t^M\}$. Assuming the weight of the cost to be μ_2, the corresponding cost is defined as

$$p_2(\mathbf{D}_t) = \mu_2(1 - \eta_t). \quad (5)$$

- If the force exerted at the remote access site exceeds the tolerance limit, we impose a severe penalty with a weight value of μ_3. The corresponding penalty function can be written as

$$p_3(F_t, \hat{F}_t) = \mu_3 \mathbf{1}_{F_t > \hat{F}_t}(F_t - \hat{F}_t). \quad (6)$$

- Actions are heavily penalized if they violate the resource inequality constraint described in (2). The corresponding penalty function has weight μ_4 and is given by

$$p_4(X_t, Y_t, z_t, w_t) = \mu_4 \mathbf{1}_{f(X_t, Y_t, z_t, w_t) > N_{\text{RB}}}. \quad (7)$$

So the overall reward can be written as

$$R_t = -\left[p_1(\mathbf{P}_t) + p_2(\mathbf{D}_t) + p_3(F_t, \hat{F}_t) + p_4(X_t, Y_t, z_t, w_t) \right]. \quad (8)$$

Our heuristic real-time allocation algorithm shares fundamental objectives with learning-based methods, including preventing AGV collisions and maintaining safe operating conditions. Both approaches consider complex state information, but while learning-based methods use this to evolve a policy over time, our heuristic method employs it for immediate, rule-based decision-making. The key distinction lies in how constraints are handled: learning-based methods incorporate them into a cost function, whereas our approach inherently satisfies constraints at each step. The first two components of the reward function are increased as the cycle time of the motion control application is satisfied. Similarly, maintaining the cycle time constraint of the remote access application is critical to increasing the third component of the reward function. The final component of the reward function corresponds to the resource constraint in Algorithm 1.

D. RL methods for benchmarking

To benchmark our heuristic algorithm, we select a diverse set of RL methods that represent different approaches to policy optimization. These methods have shown promising results in various complex decision-making tasks, making them suitable comparators for our network resource allocation problem in an Industry 5.0 environment.

- **Monte Carlo RL:** It is a model-free approach that can handle unknown environment dynamics, which is beneficial in complex Industry 5.0 systems [28]. This method requires complete episodes for learning via extensive exploration, which becomes infeasible as the number of AGVs and cameras increases.
- **Deep Q-Learning (DQL):** DQL is well-suited for handling high-dimensional state spaces common in Industry 5.0 systems [28]. Although DQL can handle the dimensionality problem to a certain extent, the number of outputs (Q-values for all state-action pairs) becomes too high when the number of AGVs and cameras increases, and can become unstable during training.
- **Deep Q-Network (DQN):** DQN provides an alternative approach to solving our network resource allocation problem [28]. In contrast to the deep Q-Learning method, we try to find the mapping between the states and the best possible actions in this approach, which cuts the dimension of the neural network's output enormously. DQN may struggle with very large action spaces and can be sample inefficient, potentially challenging in rapidly changing Industry 5.0 scenarios.
- **Proximal Policy Optimization (PPO):** Although PPO is well-suited for continuous state spaces and can handle discrete actions effectively and offer good sample efficiency and stability, it requires careful hyperparameter tuning and can be computationally intensive for large-scale Industry 5.0 system [29].
- **Advantage Actor-Critic (A2C):** A2C is capable of handling continuous state spaces and discrete actions [29]. Its synchronous nature can lead to more stable training in dynamic environments. However, it converges slower than some other methods and can be sensitive to learning rate settings.
- **Trust Region Policy Optimisation with Lagrangian approach (TRPO-Lagrangian):** TRPO-Lagrangian is a safeRL method, ensuring more stable policy updates, which is valuable in industrial settings with stringent performance requirements [30]. SafeRL, unlike traditional RL, incorporates safety constraints directly into the learning process, ensuring policy improvements without performance degradation. On the flip side, TRPO is

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474

computationally more expensive and converges slowly in a highly dynamic industrial environment.

V. Simulation Results

We can broadly divide the simulation results section into five subsections. In the first subsection, we compare the performance of network resource allocation using various learning methods described in Section IV-D and the heuristic approach developed in Section III-C. In the second subsection, we show how selective RB allocation of cameras and AGVs leads to better scalability in terms of the number of AGVs and cameras associated with an industrial process, compared to our previous work [25]. In the third subsection, we show how well our theoretical estimates perform compared to actual resource requirements. In the fourth subsection, we compare the performance of the network resource allocation policy described in Algorithm 1 with our method discussed in [25], and also with some of the existing policies for allocating network resources in an industrial setting [6], [7], [8]. In the final subsection, we show how maintaining a high degree of availability can ensure the proper functioning of industrial applications.

We consider an industry floor with an area of 1 km$\times$1 km for the simulation setup. The field of view of each camera is 100 m. The time-stamps of the remote access events are generated using an exponential distribution with the parameter 0.005, that is, on average a remote access event occurs at an interval of 16.67 minutes. The remote access locations by human workers are selected using a uniform random distribution defined over the factory floor area. We consider a particular type of remote access work that requires a human worker to remove an obstacle weighing 500 g remotely. The grip force applied by the human worker is in the range 10 N and 15 N [31]. The other details of the simulation setup are provided in Table II. We simulate this Industry 5.0 system using OMNet++ version 6.0 integrated with SimuLTE, which provides a 5G network backbone. While the values of different parameters used in the simulation are practical values taken from cited sources, the simulation setup is not connected to any real system. We have used a computer setup with 128 GB RAM and an Intel Xeon processor with 20 cores. For simulating the Industry 5.0 system using the heuristic method of RB allocation, we have used a MacBook Pro with a 3.1 GHz Dual-Core Intel Core i5 processor, Intel Iris Plus Graphics 650 1536 MB, and 8 GB RAM.

TABLE II: Simulation setup

Parameter	Value
Monitoring camera resolution	1920 $\times$ 1080 at 60 fps
Size of packets generated by cameras	25 Kb
Size of packets generated by mobile robot	65 Kb
Size of the motion control messages	500 bytes [2]
Cycle time of motion control application	100 ms [1]
Cycle time of remote access application	45 ms [32]
Values of δ_{Proc} and δ_{Motion} (Sec. IV-A)	1 ms and 0.5 ms [1]
Values of δ_{Perc} and δ_{Haptic} (Sec. IV-A)	25 ms [33] and 1 ms [1]
Subcarrier spacing (Δf) of the RBs	15 kHz
Maximum speed of AGVs and mobile robot	2 m/s

A. Comparison of Learning-based Methods versus Heuristic Method

In this subsection, we compare the performance of various RL methods (Section IV-D) and heuristic algorithm (Section III-C) in an Industry 5.0 system with varying numbers of AGVs and monitoring cameras. We use distance-based and set cover-based selection for AGVs and cameras. In Fig. 2, we use several RL methods, namely Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), Trust Region Policy Optimization (TRPO) and Deep Q-network (DQN), and compare their performance with Algorithm 1. Each episode of the simulation consists of 10^5 steps. The reward is calculated as described in Section IV-C. When the number of AGVs and cameras is relatively low (Fig. 2(a)), it appears that A2C is the fastest to approach the optimal reward that can be found by it. As the number of AGVs and cameras is increased (Fig. 2(b) and Fig. 2(c)), DQN outperforms the other methods in terms of converging towards the optimal reward. TRPO-Lagrangian is a safe RL method developed by OpenAI [30]. For our system, this method does not surpass the other RL methods even after 1000 episodes, as shown in Fig. 2(a).

For simpler factory settings (Fig. 2(a)), the RL methods can achieve better reward compared to our heuristic algorithm, but as we increase the number of AGVs and cameras, the RL methods converge very slowly due to the higher dimensionality of the state space, and running these algorithms for a thousand episodes becomes untenable. Algorithm 1 on the other hand achieves better reward than the RL methods with far less computation (Fig. 2(b) and Fig. 2(c)).

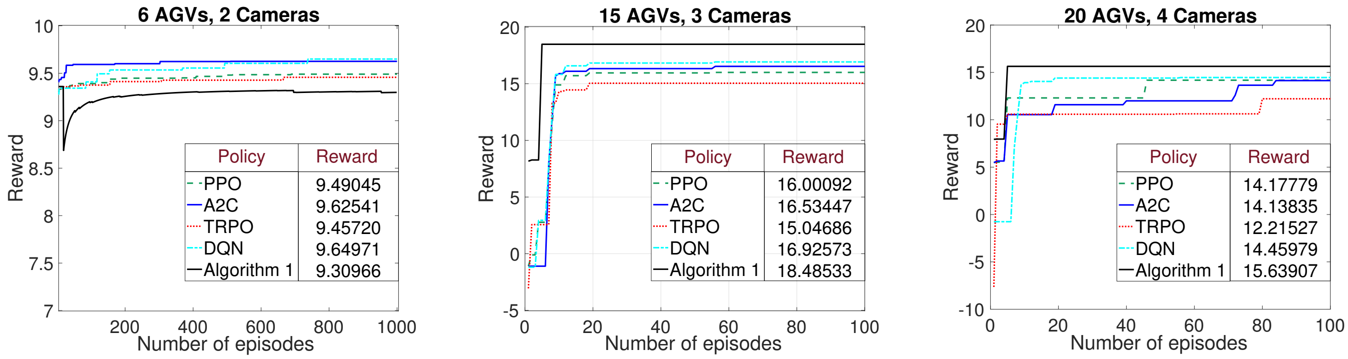
If we compare the reward values for different numbers of AGVs and cameras in Fig. 2(a)-(c), it seems that the reward goes up as we increase the number of AGVs to a certain extent and then it falls as we further increase the number of AGVs. The area of operation remains the same. If we keep on adding AGVs, it restricts their motion after a certain point and hampers smooth operation and this explains the trend of reward values.

B. Application-aware resource allocation

In Fig. 3a, we vary the number of AGVs and compare the average number of AGVs and cameras polled during a single phase of RB allocation using different methods. Due to the scalability issue associated with learning-based methods, we cannot go beyond 12 and 24 AGVs for DQL and DQN methods. It is evident from Fig. 3a that we do not need to allocate RBs to all AGVs and cameras all the time. Selective RB allocation based on safety requirements on the factory floor leads to more efficient use of valuable network resources. Our heuristic method does not look at the future states of the factory floor and is inherently myopic. Therefore, it slightly outperforms the learning-based techniques. However, we can use the heuristic approach for RB allocation in an Industry 5.0 system with more AGVs.

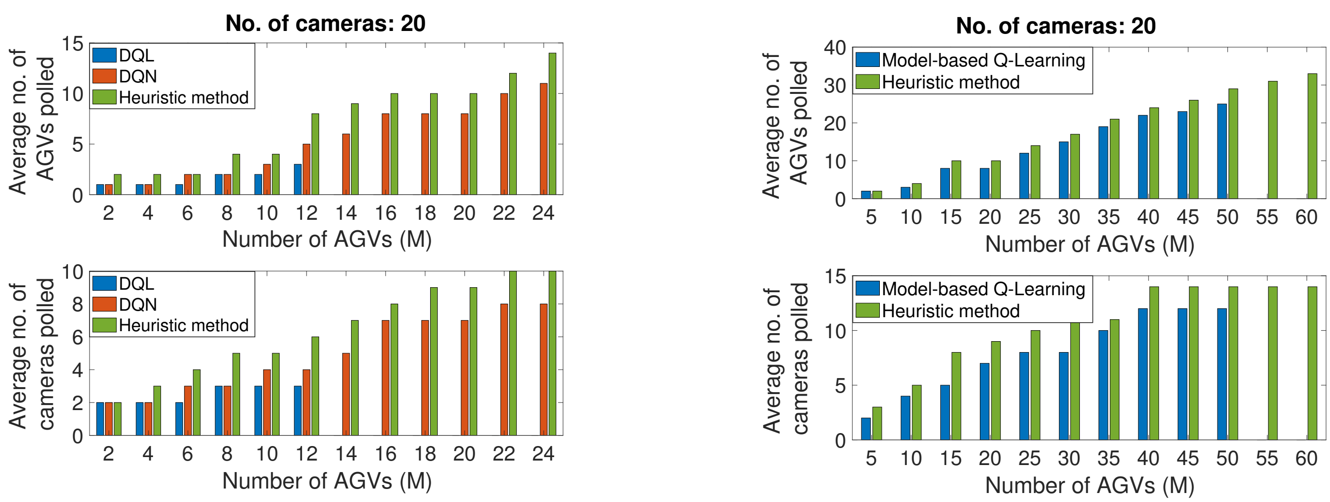
While RL approaches offer powerful solutions for many complex problems, the specific nature of our industrial floor resource allocation scenario presents unique challenges that favor our proposed heuristic method. The dynamic nature of

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474



(a) (b) (c)

Fig. 2: These figures compare the performance of various reinforcement learning methods and a heuristic algorithm in terms of reward over episodes. For an industry setting with fewer AGVs and cameras, RL methods achieve higher rewards compared to the heuristic algorithm (Algorithm 1). Due to the computationally intensive nature of RL methods, the heuristic algorithm achieves better reward when more AGVs and cameras are added.



(a) Model-free learning (b) Model-based learning

Fig. 3: These figures show the average number of AGVs and cameras polled during resource block allocation for different methods as the number of AGVs increases. Selective resource block allocation based on safety requirements leads to more efficient use of network resources, with the heuristic method showing better scalability than learning-based approaches.

our environment, where the number of active AGVs and cameras can fluctuate, challenges the adaptability of pre-trained models. Our heuristic method addresses these issues through its polynomial time complexity with respect to the number of AGVs and cameras, ensuring scalability as the system grows. The heuristic method's adaptability to changing environmental conditions without requiring extensive retraining gives it a significant advantage in dynamic industrial settings. This balance of efficiency, scalability, and adaptability makes our proposed method particularly well-suited for the real-time resource allocation challenges in modern, flexible manufacturing environments. We can find the final policy using less computational burden when we incorporate the heuristic method discussed in Section III-C as a model for determining policy using the learning-based frameworks. In Fig. 3b, we have added more AGVs in this model-based learning setup, and it is clear that the learning-based method outperforms our heuristic method of resource allocation. Model-based learning techniques are superior in performance compared to the model-free learning methods. The selective allocation of AGVs and cameras based on industrial states leads to less resource usage. It allows for more industrial components to be connected to the wireless network using the available resources, improving the scalability of the industrial system.

C. Theoretical Estimates

In Fig. 4, we plot the theoretical estimates of upper and lower bounds of N_{RB}, which are calculated using the statistical method described in Section III-C, and compare them with the actual RB requirement to achieve 99.9999% availability. In Fig. 4, we use 5 and 10 cameras for monitoring the factory floor. The estimated lower bound based on selective RB allocation to AGVs and cameras provides a strict bound. To calculate the upper bound, we allocate RBs to all components. So, the gap between the upper bound and the actual requirement is quite significant as the number of AGVs and cameras increases. We need far fewer RBs to get 99.9999% availability if the RB allocator makes decisions based on the

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2024.3504474

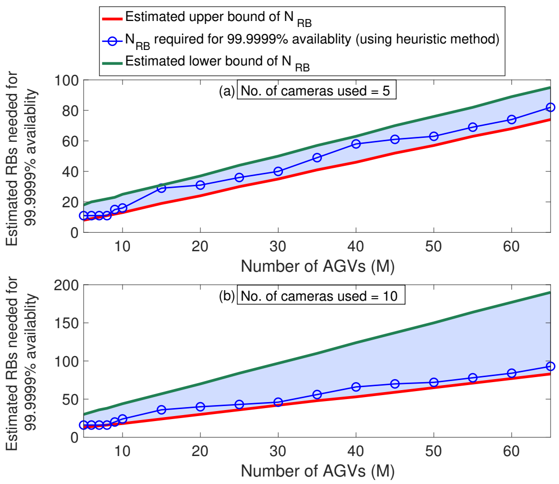


Fig. 4: This figure plots the theoretical upper and lower bounds of required resource blocks (RBs) against the actual requirement for achieving 99.9999% availability as the number of AGVs increases. The selective RB allocation approach requires significantly fewer RBs than allocating to all components, improving system scalability.

industrial application. As a result, we can accommodate more AGVs and other industrial components into the 5G network, thus increasing the scalability of the Industry 5.0 system.

D. Comparison with Existing Resource Allocation Policies

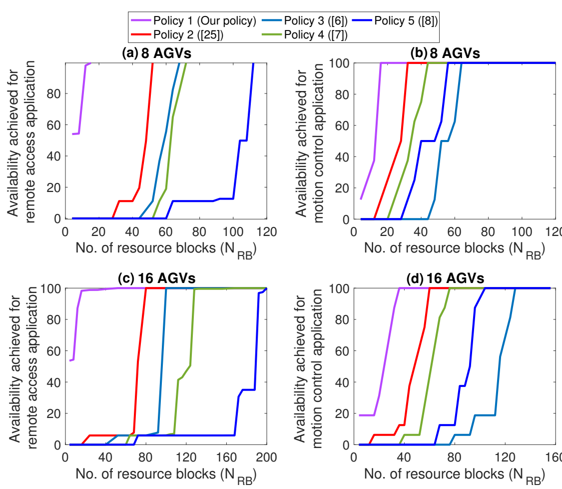


Fig. 5: These figures compare the performance of the proposed policy (Policy 1) with existing network resource allocation schemes (Policy 2 [25], Policy 3 [6], Policy 4 [7], and Policy 5 [8]). The application-aware allocation in Policy 1 achieves 99.9999% availability with nearly 50% fewer resource blocks compared to other policies, enhancing bandwidth usage efficiency and scalability.

In Fig. 5, we compare the performance of our real-time heuristic method for RB allocation (Policy 1), with the policy described in our work [25] (Policy 2) and some of the existing RB allocation policies in literature for industry settings, namely - [6] (Policy 3), [7] (Policy 4), and [8] (Policy 5). In [25] (Policy 2), we formulate a myopic optimization problem for RB allocation aiming to maximize the efficiency of the industrial process and minimize the end-to-end delay and provide an algorithm to solve the problem. In Policy 3, the industrial applications are prioritised for resource allocation based on QoS demands and queuing delays. In Policy 4, an optimization framework has been designed for resource allocation to maximize energy efficiency. The authors suggest designing fixed network slices for eMBB and URLLC applications targeting to maximize throughput and spectral efficiency. In all policies except Policy 1, the RB allocation process is detached from the state of industrial applications. In Fig. 5a and Fig. 5b, we use 5 cameras to monitor, and the number of AGVs are 8 and 16. Due to the application-aware allocation in Policy 1, we achieve 99.9999% availability for motion control and remote access applications with fewer RBs (nearly 50% reduction compared to the best policy) compared to the other policies. Increased bandwidth usage efficiency allows us to accommodate more industrial devices in the 5G network, thus improving scalability.

E. Availability and Scalability

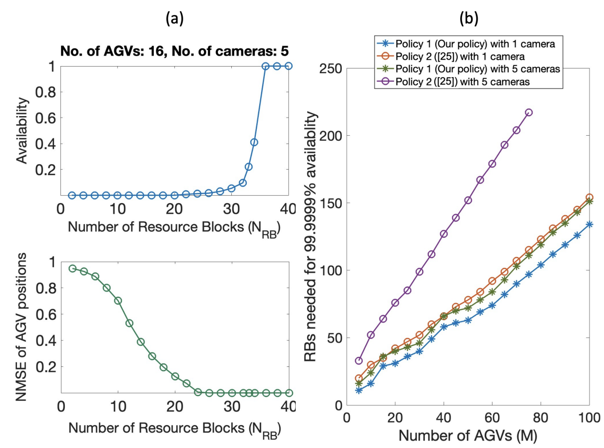


Fig. 6: The figure shows how *availability* and location data accuracy (NMSE) depend on the number of available network resources and compares resource usage efficiency between the proposed policy and our previous work [25]. Increased network resources improve availability and data accuracy, while the selective allocation approach allows for connecting more AGVs with the same amount of resources, enhancing system scalability.

In Fig. 6(a), we show how the *availability* of our industry 5.0 framework depends on the number of available network resources and also how the *availability* factor affects the accuracy of the industrial data maintained at the edge server. We plot the *availability* and the Normalised Mean Square Error (NMSE) of the location data of the AGVs concerning the number of resource blocks allocated for the industrial system. The actual position of the i^{th} AGV and the position data estimated at the edge server are denoted by $\mathbf{p}_t^i$ and $\hat{\mathbf{p}}_t^i$ respectively. Then the NMSE of the positions of the AGVs throughout the duration of the industrial system can be calculated as

$$\mathrm{NMSE} = \frac{1}{MT}\sum_{t=1}^{T}\sum_{i=1}^{M}\frac{\|\mathbf{p}_t^i - \hat{\mathbf{p}}_t^i\|_2^2}{\|\mathbf{p}_t^i\|_2^2}. \tag{9}$$

As we increase network resources, more industrial components can be connected to the wireless network while increasing the probability of maintaining the cycle times of the motion control and remote access applications. This leads to improved *availability* (top half of Fig. 6(a)). With access to more bandwidth, it is possible to establish efficient and reliable communication between the factory floor and the edge server and the cycle time constraint of motion control application is satisfied with a very high degree of availability. So, the positional data of the AGVs at the edge server is timely updated, leading to better accuracy of the perception by the motion controller about the actual industrial system and the NMSE of the positions of the AGVs goes down (bottom half of Fig. 6(a)).

In Fig. 6(b), we compare the resource usage for the heuristic policy described in this work and the policy designed in our previous work [25]. By selectively allocating resources to the AGVs and cameras, as done in the proposed policy, we can improve the usage efficiency of the network resources. As a result, more AGVs can be connected to the wireless network using the same amount of resources, compared to our previous work.

VI. Conclusions and Future Works

This work presents RB allocation policies for an Industry 5.0 system with AR-VR-assisted remote human access. We propose a real-time heuristic allocation method that is aware of the states of the industrial applications, which leads to improved network resource usage efficiency and enhanced scalability of the industrial system. We also develop a statistical method to estimate upper and lower bounds for resource requirements to achieve the desired availability using our policy. Compared to some of the existing policies, our policy requires 50% less RBs to achieve 99.9999% availability. We model the Industry 5.0 system in an RL framework to compare the performance of our policy, which has a significant computational advantage over RL methods.

In our future work, we intend to incorporate different Industrial applications such as - floor monitoring using various sensors and analyse how we can build a real-time resource allocation policy to meet the desired performance targets.

References

[1] G. Brown, P. Analyst, and H. Reading, “Ultra-Reliable Low-Latency 5G for Industrial Automation,” *Technol. Rep. Qualcomm*, vol. 2, no. 52065394, pp. 1–11, 2018.

[2] L. Grosjean, O. Dobrijevic, K. Landerns, R. Kullng, P. Falco, S. Azhar, A. Rostami, S. Schmitt, N. Knig, P. Mohanram, B. Sayrac, F. Parzysz, G. Madhusudan, A. M. G. Serrano, S. Cerovic, S. Destouet-Roblot, and S. Inca, “Forward Looking Smart Manufacturing Use Cases, Requirements,” pp. 1–74, 2020.

[3] P. Ferrari, S. Rinaldi, E. Sisinni, F. Colombo, F. Ghelfi, D. Maffei, and M. Malara, “Performance Evaluation of Full-cloud and Edge-cloud Architectures for Industrial IoT Anomaly Detection Based on Deep Learning,” in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 & IoT)*. IEEE, 2019, pp. 420–425.

[4] R. Bacci di Capaci and C. Scali, “A Cloud-based Monitoring System for Performance Assessment of Industrial Plants,” *Industrial & Engineering Chemistry Research*, vol. 59, no. 6, pp. 2341–2352, 2020.

[5] A. Cortés-Leal, C. Cárdenas, and C. Del-Valle-Soto, “Maintenance 5.0: Towards a Worker-in-the-Loop Framework for Resilient Smart Manufacturing,” *Applied Sciences*, vol. 12, no. 22, pp. 1–34, 2022.

[6] W. U. Rehman, T. Salam, A. Almogren, K. Haseeb, I. U. Din, and S. H. Bouk, “Improved Resource Allocation in 5G MTC Networks,” *IEEE Access*, vol. 8, pp. 49 187–49 197, 2020.

[7] X. Qi, S. Khattak, A. Zaib, and I. Khan, “Energy Efficient Resource Allocation for 5G Heterogeneous Networks Using Genetic Algorithm,” *IEEE Access*, vol. 9, pp. 160 510–160 520, 2021.

[8] T. Ma, Y. Zhang, F. Wang, D. Wang, and D. Guo, “Slicing Resource Allocation for eMBB and URLLC in 5G RAN,” *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–11, 2020.

[9] I. O. Sanusi, K. M. Nasr, and K. Moessner, “Radio Resource Management Approaches for Reliable Device-to-device (D2D) Communication in Wireless Industrial Applications,” *IEEE transactions on cognitive communications and networking*, vol. 7, no. 3, pp. 905–916, 2020.

[10] H. Peng, W. Tärneberg, and M. Kihl, “Latency-Aware Radio Resource Allocation Over Cloud RAN for Industry 4.0,” in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–8.

[11] S. Jamil, M. Rahman, M. S. Abbas, and Fawad, “Resource Allocation Using Reconfigurable Intelligent Surface (RIS)-Assisted Wireless Networks in Industry 5.0 Scenario,” *Telecom*, vol. 3, no. 1, pp. 163–173, 2022. [Online]. Available: https://www.mdpi.com/2673-4001/3/1/11

[12] R. Accorsi, M. Bortolini, F. G. Galizia, F. Gualano, and M. Oliani, “Scalability analysis in industry 4.0 manufacturing,” in *Sustainable Design and Manufacturing 2020: Proceedings of the 7th International Conference on Sustainable Design and Manufacturing (KES-SDM 2020)*. Springer, 2021, pp. 161–171.

[13] C. Aceta, I. Fernández, and A. Soroa, “KIDE4I: A Generic Semantics-Based Task-Oriented Dialogue System for Human-Machine Interaction in Industry 5.0,” *Applied Sciences*, vol. 12, no. 3, pp. 1–26, 2022.

[14] V. Cillo, G. L. Gregori, L. M. Daniele, F. Caputo, and N. Bitbol-Saba, “Rethinking companies’ culture through knowledge management lens during Industry 5.0 transition,” *Journal of Knowledge Management*, vol. 26, no. 10, pp. 2485–2498, 2022.

[15] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, and L. Wang, “Industry 5.0: Prospect and retrospect,” *Journal of Manufacturing Systems*, vol. 65, pp. 279–295, 2022.

[16] S. Li, P. Zheng, J. Fan, and L. Wang, “Toward Proactive Human-Robot Collaborative Assembly: A Multimodal Transfer-Learning-Enabled Action Prediction Approach,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 8, pp. 8579–8588, 2021.

[17] D. Mourtzis, V. Siatras, J. Angelopoulos, and N. Panopoulos, “An Augmented Reality Collaborative Product Design Cloud-Based Platform in the Context of Learning Factory,” *Procedia Manufacturing*, vol. 45, pp. 546–551, 2020.

[18] Y. Wang, T. Yu, and K. Sakaguchi, “Context-Based MEC Platform for Augmented-Reality Services in 5G Networks,” in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–5.

[19] W. Nakimuli, J. Garcia-Reinoso, J. E. Sierra-Garcia, P. Serrano, and I. Q. Fernández, “Deployment and Evaluation of an Industry 4.0 Use Case over 5G,” *IEEE Communications Magazine*, vol. 59, no. 7, pp. 14–20, 2021.

[20] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, “Learning-based context-aware resource allocation for edge-computing-empowered industrial iot,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4260–4277, 2019.

[21] E. C. Balta, M. H. Mamduhi, J. Lygeros, and A. Rupenyan, “Controller-Aware Dynamic Network Management for Industry 4.0,” in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2022, pp. 1–6.

[22] G. X. Wang, S. H. Huang, Y. Yan, and J. J. Du, “Reconfiguration schemes evaluation based on preference ranking of key characteristics of reconfigurable manufacturing systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 89, pp. 2231–2249, 2017.

[23] A. Cerqueus, X. Delorme, and A. Dolgui, “Analysis of the Scalability for Different Configurations of Lines,” *Reconfigurable Manufacturing Systems: From Design to Implementation*, pp. 139–160, 2020.

[24] T. Adame, A. Bel, and B. Bellalta, “Increasing LPWAN Scalability by Means of Concurrent Multiband IoT Technologies: An Industry 4.0 Use Case,” *IEEE Access*, pp. 46 990–47 010, 2019.

[25] A. A. Sardar, A. S. Rao, T. Alpcan, G. Das, and M. Palaniswami, “Network Resource Allocation for Industry 4.0 with Delay and Safety Constraints,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, pp. 223–237, 2023.

[26] R. Masoni, F. Ferrise, M. Bordegoni, M. Gattullo, A. E. Uva, M. Fiorentino, E. Carrabba, and M. Di Donato, “Supporting Remote

Maintenance in Industry 4.0 through Augmented Reality," *Procedia manufacturing*, vol. 11, pp. 1296–1302, 2017.

[27] N. Alon, D. Moshkovitz, and S. Safra, "Algorithmic construction of sets for k-restrictions," *ACM Transactions on Algorithms (TALG)*, vol. 2, no. 2, pp. 153–177, 2006.

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[29] H. Yang, X.-Y. Liu, S. Zhong, and A. Walid, "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," in *Proceedings of the first ACM international conference on AI in finance*, 2020, pp. 1–8.

[30] A. Ray, J. Achiam, and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning," *arXiv preprint arXiv:1910.01708*, vol. 7, no. 1, pp. 1–25, 2019.

[31] G. P. Slota, M. L. Latash, and V. M. Zatsiorsky, "Grip Forces During Object Manipulation: Experiment, Mathematical Model, and Validation," *Experimental Brain Research*, vol. 213, pp. 125–139, 2011.

[32] B. Picano and R. Fantacci, "End-to-End Delay Bound Analysis for VR and Industrial IoE Traffic Flows under Different Scheduling Policies in a 6G Network," *Computers*, vol. 12, no. 3, pp. 1–15, 2023.

[33] M. C. Potter, B. Wyble, C. E. Hagmann, and E. S. McCourt, "Detecting meaning in RSVP at 13 ms per picture," *Attention, Perception, & Psychophysics*, vol. 76, pp. 270–279, 2014.