ACCSE 2022

The Seventh International Conference on Advances in Computation, Communications and Services

ISBN: 978-1-61208-964-5

June 26th –30th, 2022

Porto, Portugal

ACCSE 2022 Editors

Ratan Lal, Northwest Missouri State University, USA

ACCSE 2022

Forward

The Seventh International Conference on Advances in Computation, Communications and Services (ACCSE 2022), held between June 26th and June 30th, 2022, continued a series of events targeting the progress made in computation, communication and services on various areas in terms of theory, practices, novelty, and impact. Current achievements, potential drawbacks, and possible solutions are aspects intended to bring together academia and industry players.

The rapid increase in computation power and the affordable memory/storage led to advances in almost all the technology and services domains. The outcome made it possible advances in other emerging areas, like Internet of Things, Cloud Computing, Data Analytics, Smart Cities, Mobility and Cyber-Systems, to enumerate just a few of them.

We take here the opportunity to warmly thank all the members of the ACCSE 2022 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to ACCSE 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the ACCSE 2022 organizing committee for their help in handling the logistics of this event.

We hope that ACCSE 2022 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of Computation, Communications and Services.

ACCSE 2022 Chairs

ACCSE 2022 Publicity Chairs

José Miguel Jiménez, Universitat Politècnica de València (UPV), Spain
Laura Garcia, Universitat Politècnica de València (UPV), Spain

ACCSE 2022
Committee

ACCSE 2022 Publicity Chairs

José Miguel Jiménez, Universitat Politècnica de València (UPV), Spain
Laura Garcia, Universitat Politècnica de València (UPV), Spain

ACCSE 2022 Technical Program Committee

Safa'a AbuJarour, University of Potsdam, Germany
Kishwar Ahmed, University of South Carolina Beaufort, USA
Muhamad Erza Aminanto, University of Indonesia, Indonesia / NICT, Japan
Maxim Bakaev, Novosibirsk State Technical University, Russia
Abdul Basit, State Bank of Pakistan (Central Bank of Pakistan), Pakistan
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Ali Behfarnia, University of Tennessee at Martin, USA
Jagadeesha R Bhat, St. Joseph Engineering College, Mangalore, India
Freimut Bodendorf, Institute of Information Systems - University of Erlangen-Nuremberg, Germany
An Braeken, Vrije Universiteit Brussel, Belgium
Arun Das, Visa Inc., USA
Mounîm A. El Yacoubi, Telecom SudParis / Institut Polytechnique de Paris, France
Alessandro Farasin, Istituto Superiore Mario Boella (ISMB), Turin, Italy
Barbara Gili Fivela, University of Salento, Italy
Aviel Glam, Technion - Israel Institue of Technology | RAFAEL - Advanced Defence System Ltd., Israel
Josefa Gómez, University of Alcalá, Spain
Robert C. Green II, Bowling Green State University, USA
Béat Hirsbrunner, University of Fribourg, Switzerland
Mehdi Hosseinzadeh, Washington University in St. Louis, USA
Fu-Hau Hsu, National Central University, Taiwan
Michael Huebner, BTU Cottbus-Senftenberg, Germany
Sergio Ilarri, University of Zaragoza, Spain
Ilias Iliadis, IBM Research - Zurich Laboratory, Switzerland
Tomayess Issa, Curtin University, Australia
Ajin Joseph, IIT Tirupati, India
Keiichi Kaneko, Tokyo University of Agriculture and Technology, Japan
Abbas Khosravi, Deakin University, Australia
Ratan Lal, Northwest Missouri State University, USA
André Langer, Chemnitz University of Technology, Germany
Yiu-Wing Leung, Hong Kong Baptist University, Kowloon Tong, Hong Kong
Shigang Li, Hiroshima City University, Japan
Yongbo Li, Facebook Inc., USA
Christopher Mansour, Mercyhurst University, Erie, USA
Alfonso Mateos Caballero, Universidad Politécnica de Madrid, Spain
Vinod Muthusamy, IBM T.J. Watson Research Center, USA
Hidemoto Nakada, AIST, Japan
Isabela Neves Ferraz, Universidade de Brasília, Brazil

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Optimal Latency Guarantee for Multiple Concurrent Packets with New IP

Lijun Dong, Richard Li
Futurewei Technologies Inc.
2220 Central Expressway
Santa Clara, CA, U.S.A
Email: {lijun.dong, richard.li}@futurewei.com

***Abstract*— Precise end-to-end latency guarantee is a network service that is required by many emerging and future applications. However, today's Internet built on the best effort principle cannot provide such service, despite of the existing Quality of Service (QoS) mechanisms. Enabled by the New IP framework, the deadline for the packets could be revealed to the network nodes and leveraged to calculate the residual latency budget and average per-hop latency constraint. Correspondingly, the packet forwarding order (i.e., the placement positions of the packets) in the outgoing queue could be deliberately manipulated to satisfy the deadline constraints for as many packets as possible, while achieving the minimum average stay time in a network node. Algorithms based on backtracking, branch and bound are proposed to address the optimal scheduling problem.**

Keywords*— *in-time guarantee; multiple packets; New IP; best effort; contract; metadata; high precision communication; QoS; precise latency; backtracking; branch and bound.

I. INTRODUCTION

Today's Internet is based on the Best Effort (BE) principle. BE is a network service that attempts to deliver packets to their intended destinations, but does not provide any guarantee of the Quality of Service (QoS), e.g., whether the packet gets dropped, or the packet reaches the destination within certain deadline. The use of BE was adopted because rather than guaranteed delivery, BE can be more efficient for some earlier services, and for the network as a whole. For example, in real-time audio or video transfers, a small percentage of packets getting lost is tolerable (i.e., does not affect the sound or video conspicuously), and recovering the lost and corrupted packets results in immoderate overhead that reduces network performance. However, for the emerging Internet applications, such as remote surgery, cloud-based autonomous driving, industrial Internet, each piece of information must be delivered precisely, referred to as High Precision Communication (HPC) [1][2]. In-time guarantee regarding the latency performance [3] is one of the most important yet barely explored territory. It refers to a network service that ensures the delivery of a packet, a group of packets, or all packets in a flow within bounded time frame. Remote surgery application requires that all messages between the master console and remote robots are delivered through the networks within the specified deadlines. If all messages are specified with the same deadline, then the latency guarantee is ensured at the flow level. If each message is specified with an independent deadline, then the latency guarantee is ensured at the packet level.

Although IntServ [4] and DiffServ [5] were proposed to improve upon BE, neither of them is suitable to the above emerging applications. The IntServ QoS model works in small networks, which is hard to be implemented in a large scale or be used in the global Internet. The DiffServ QoS model differentiates the service priorities at class level, which is not satisfactory to the in-time guarantee requirement at the flow level, not to mention the packet level. In [6][7], the authors proposed a new class called Latency Guarantee Service (LGS) on top of already defined classes in DiffServ. The flows that belong to this LGS class will have the highest priority to be transmitted after being admitted. The maximum latency that may be incurred at each intermediate hop is calculated to ensure that the total end-to-end latency of an admitted LGS flow will not exceed its deadline. However, the proposal still only works at class level of granularity, and the end-to-end latency estimation is very raw at its upper bound, thus the network resource may not be efficiently used.

Some deadline-aware transport schemes have been proposed to in Data Centers, such as Deadline-Aware Data center TCP (D2TCP) [7], Deadline Driven Delivery (D3) [9], and Preemptive Distributed Quick (PDQ) [10], which perform flow scheduling to complete serving the most significant number of flows before their deadlines. D3 is a deadline-aware transport scheme, in which senders calculate the requesting rate for flows before the actual flow transmission starts, and the on-path switches towards the destination take the role in helping make decisions on the sending rate for each active flow in a First-Come-First-Served (FCFS) manner. PDQ uses two policies, Early Deadline First (EDF) and Shortest Job First (SJF), where the latter is used to break ties for scheduling. PDQ may preempt a flow that is currently being served (i.e., the active flow) if the deadline of a new arriving flow is tighter than that of the currently active flow. In a more recent work [11], the authors proposed the Preemptive Efficient Queuing (PEQ), which takes both the deadlines and sizes of the flows into account for efficient scheduling of flows in a data-center network. However, even though all those works considered deadlines, they are at the flow level and the major goal is still to optimize the flow throughput. They cannot guarantee the transmission latency of a particular packet or a group of packets in a flow to be within the bounded time frame. On the other hand, with the existing scheduling polices (e.g., FCFS, EDF, SJF), some of the concurrent flows can fail when the deadline expires.

In this paper, we propose to leverage the New Internet Protocol (New IP) framework, such that each intermediate router on the forwarding path is able to process the packet at per-hop basis and schedule all concurrent latency-sensitive packets intelligently in order to achieve the shortest total stay time for those packets in the router. We need to point out that New IP serves as one embodiment of the proposal. We do not exclude other tentative implementation possibilities, such as IPv6

 ISBN: 978-1-61208-964-5

(Internet Protocol Version 6) extension headers, or IPv4 (Internet Protocol Version 4) options. The rest of the paper is arranged as follows: Section II introduces the New IP framework; Section III describes the proposed in-time guarantee mechanisms and algorithms; Section IV gives performance comparisons; Section V concludes the paper.

II. NEW INTERNET PROTOCOL (NEW IP)

New Internet Protocol (New IP) [12] [13] has been proposed to address issues of the three major building blocks of the current Internet, i.e., statistical multiplexing, best-effort paradigm, and an IP address-based reachability. New IP is a data plane technology that defines a new network packet specification, and new service capabilities enabled in the network nodes.

A New IP packet starts with the *Header Specification*, which specifies the boundary of the following *Shipping Specification*, *Contract Specification* and *Payload Specification*.

The *Shipping Specification* intends to change the current fixed types of addressing (i.e., IPv4 or IPv6) to being able to include all types of addresses in a flexible manner and accommodate different reachability scenarios.

The *Contract Specification* provides a series of apparatuses to facilitate new network capabilities, their functionalities and regulative conditions at the finest packet-level granularity. The network and routers fulfill the contract, with the assumption that the contract has been agreed between the packet sender/receiver and the network. New IP contract could be constructed from multiple contract clauses, each of which might include Action, Event/Condition and the associated Metadata. A Contract Clause depicts the processing that network nodes (which are upgraded to support New IP) would carry out on the packet when it traverses the network according to the predefined triggering event or condition. The Metadata contains semantics about the packet, the sender/receiver context information, or the network statistics, etc.

The *Payload Specification* divides the packet payload into multiple portions, such that when network congestion happens, the network nodes could drop some portions of the payload and allow the receiver to consume the residual information. This type of communication is named as Qualitative Communication [14] [15], which helps to mitigate re-transmission overhead and delay when faced with slow or congested network conditions.

III. OPTIMAL LATENCY GURANTEE FOR MULTIPLE CONCURRENT PACKETS

A. Single Packet Scenario

We consider the simplest scenario, in which the network will need to guarantee the in-time delivery of a particular packet. In other words, this particular packet could have the highest priority when being scheduled in the outgoing queue, compared to other packets without such requirement. We consider the end-to-end in-time delivery requirement is set as: $latency \leq d$, in which $latency$ denotes the incurred end-to-end latency for the packet delivery, d denotes the deadline constraint. The first router that the packet reaches is able to apprehend the number of hops information between itself and the destination, which is denoted as *n*. In other words, there are *n* number of hops between the first router and the destination, which means *n* number of routers are involved in the packet forwarding. The exemplary topology is shown in Figure 1.

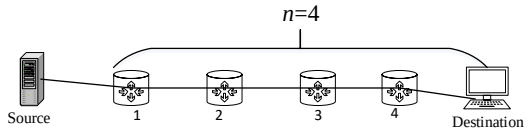

Figure 1. Example topology

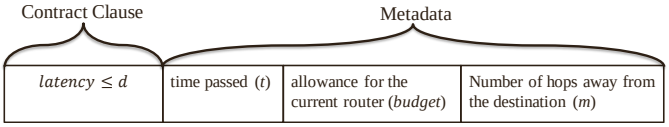

Figure 2. New IP header for packet with in-time guarantee requirement

The source can specify the in-time delivery requirement in the New IP header with the contract clause set to: $latency \leq d$. The metadata carries the following information, as shown in Figure 2:

- Time passed (t): it represents how much time has passed since the packet is sent out from the source. It is initialized to 0 by the source.
- Number of hops away from the destination (m): it represents the number of hops between the current router to the destination.
- Allowance for the current router ($budget$): it denotes the time duration that is allowed for the current router between the time when the packet arrives at the router and the time when the last bit of the packet gets transmitted to the next hop.

When the packet reaches the router 1, *t* is set to the time used to transport the packet from the source to the router 1 and *m* is set to *n*. The current router is allowed to have the time $budget$ to forward the packet to the next router, which is calculated as:

$$budget = \frac{d - t}{m} \quad (1)$$

The time $budget$ has two aspects: (1) If the router uses less time than $budget$, then it does not affect the following routers, but gives them more time budget to use. (2) If the router uses more time than the budget, it will affect the rest of the routers. However, it does not mean the packet has to be dropped if the budget cannot be met. The hybrid policy used in the router for the particular packet is that it puts the packet at the highest priority, but tries its best.

When the packet reaches the intermediate routers (e.g., the router 2, 3 and 4), t is set to be the time used to transport the packet from the source to the current router, m is deducted by 1 every time the packet is being forwarded by a router, $budget$ is calculated accordingly. When an intermediate router finds that the residual time $(d - t)$ is not enough for it to transfer the packet to the destination, the packet is dropped and the in-time guarantee fails because it is not a realistic requirement. The

intermediate router needs to reply the source with a response message, which is also designed in the embodiment of New IP. The metadata contains the following information as shown in Figure 3:

- $dsuggested$ is the suggested deadline based on the current situation, i.e., the number of hops from the current router to the destination and the average latency incurred at previous routers. $dsuggested$ is calculated based on the below equation, where σ gives a small amount of extra time added to the suggested deadline configuration.

$$dsuggested = \frac{t * n}{n - m} + \sigma \tag{2}$$

- *Unsuccessful* flag is to indicate that the packet delivery failed due to the reason that the specified deadline cannot be met.
- *Number of hops away from the source* is set to be $(n - m)$, which is used to notify the source at which intermediate router the packet gets dropped and the deadline expires.

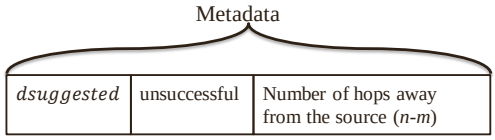

Figure 3. New IP header for reply message from an intermediate router

B. Multiple Concurrent Packets Scenario

In reality, routers might need to guarantee the in-time delivery for packets from multiple flows. We assume a router receives packets from the ingress ports with the in-time guarantee contract and latency related metadata specified as proposed in Section III.A. There is a dedicated queue for latency guaranteed packets for each outgoing port, called Latency Guarantee Queue (**LGQ**), as shown in Figure 4. All packets forwarded by the router with latency guarantee contract clause are put in the LGQ. The packets in the LGQ have the highest priority to be scheduled compared to other packets without deadline constraints.

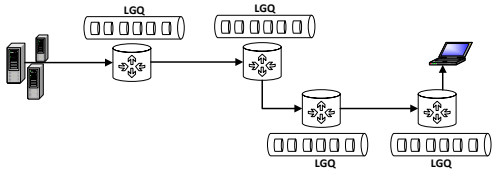

Figure 4. Latency guarantee queue (LGQ)

The time duration a packet stays in the router depends on how the other packets are scheduled, which is called ***stay time*** of the packet in the router. A packet's stay time equals to the total stay time of the packets scheduled before it and its own header processing, propagation and transmission delay in the router. The optimization problem is formulated with the objective to minimize the total stay time of the packets in the router, which have in-time guarantee contract and are going to be forwarded through the same outgoing port. Given there are total K total number of such packets:

$$\min \sum_{k=1}^{K} t_k \tag{3}$$

$$s.j.: t_k \le b_k \text{ for } k = 1 \dots K \tag{4}$$

The problem is that we want to find a permutation of $\{1, \dots K\}$ ($\sigma: \{1, \dots K\} \to \{1, \dots K\}$) which represents the scheduling order of the packets (i.e., the positions of the packets in the queue from front to rear), such that the total stay time can be minimized. For a packet at the k^{th} position, its stay time is calculated as:

$$t_{\sigma(k)} = \sum_{i=1}^{k} P_{\sigma(i)} \tag{5}$$

where $P_{\sigma(i)}$ is the stay time when the packet at the ith position is served. The optimization problem is converted to:

$$\sum_{k=1}^{K} t_{\sigma(k)} = K * P_{\sigma(1)} + (K-1) * P_{\sigma(2)} + \dots + P_{\sigma(K)} \tag{6}$$

TABLE I. PACKETS IN A ROUTER

Identifier	Budget	Stay Time
1	b_1	$P_{\sigma(1)}$
2	b_2	$P_{\sigma(2)}$
3	b_3	$P_{\sigma(3)}$

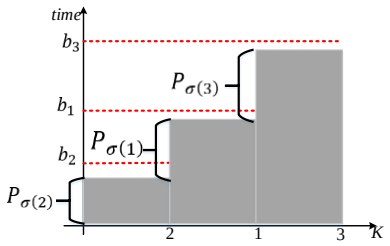

Figure 5. Simple example to illustrate the optimization problem

We use a simple example with the packets as shown in TABLE I. If the permutation is $\sigma: \{1, 2, 3\} \to \{2, 1, 3\}$), then the total stay time of the three packets is calculated as:

$$\begin{aligned} t_1 + t_2 + t_3 &= P_{\sigma(2)} + \left(P_{\sigma(2)} + P_{\sigma(1)}\right) \\ &\quad + \left(P_{\sigma(2)} + P_{\sigma(1)} + P_{\sigma(3)}\right) \\ &= 3 * P_{\sigma(2)} + 2 * P_{\sigma(1)} + P_{\sigma(3)} \end{aligned}$$

The problem is firstly to find all feasible solutions, then find the optimal one that minimizes the area in grey, as shown in Figure 5. We define a feasible solution as a schedule under which all packets' per-hop deadlines could be met.

The algorithm as shown in TABLE II. is proposed to solve the optimization problem by using backtracking method. A typical backtracking algorithm will need the procedures as follows. The constraint is shown in (4) and the objective is shown in (6).

 ISBN: 978-1-61208-964-5

- $discard(constraint, s)$: return true only if the partial scheduling s is not worth going further.
- $accept(constraint, s)$: return true if s is a solution that satisfies all constraints, and false otherwise.
- $first(constraint, s)$: generate the first extension of candidate s, which means the first packet in the queue is selected and added to s.
- $next(constraint, a)$: generate the next alternative extension of a candidate, after the extension a, which means another different packet is selected to be next in the queue.
- $record(constraint, s)$: record the solution that satisfies all constraints.
- $calculate(objective, s)$: calculate the objective result for the solution.
- $select(smallest, objective, s)$: compare the current solution with the selected solution to make sure the selected solution always has the minimal objective result.

TABLE II. BACKTRACKING ALGORITHM

```
backtracking(objective, constraint, s)
1   if discard(constraint, s), then
2       return;
3   if accept(constraint, s), then
4       record(constraint, s)
5       o = calculate(objective, s)
6       select(smallest, objective, s)
7   a = first(constraint, s)
8   while a ≠ NULL do
9       backtracking(objective, constraint, a)
10      a = next(constraint, a)
```

The enumeration procedure can find the optimal solution for any problem with constraints. But it takes too much time, even with the proposed algorithm using backtracking, if the number of packets that needs to be scheduled is large in a router.

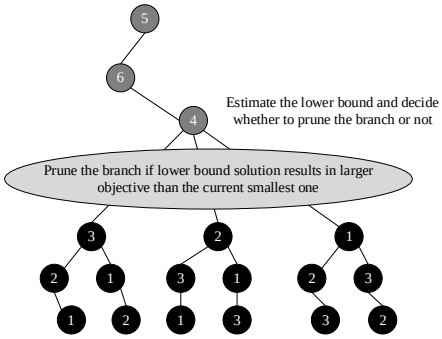

Figure 6. Example of pruning a branch

So, we improve the algorithm by leveraging Branch and Bound concept:

- For a current extension that is partial, estimate the lower bound, stop extending from the current candidate and prune the whole branch rooting from it.
- For the example as shown in Figure 6, the lower bound can easily be calculated by sorting the budgets in decreasing order. For example, for packet 1, 2, 3, the budgets b_1, b_2, b_3 in decreasing order are b_2, b_1, b_3, thus the branch of 2, 1, 3 results in the lower bound of the entire branch extended from 5, 6, 4.
- If this lower bound solution (i.e., 5, 6, 4, 2, 1, 3) cannot obtain a smaller objective than the current smallest one, then the entire branch should not be traversed and can be pruned from the recursive iteration.

In order to significantly reduce the running time of the algorithm, instead of minimizing the objective, the algorithm can be stopped when the first solution that satisfies the constraints is found. Such solution is called a feasible schedule to the packets that are being considered in the algorithm.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed backtracking algorithm. At the end of Section III.B, the proposed Backtracking Algorithm with Branch and Bound (BABB) may stop at a feasible schedule, which is a permutation of the packets that need to be scheduled and satisfies all the constraints specified in (4). It is denoted as One Possible Feasible Schedule (OPFS) in the following of the section.

TABLE III. PACKET SET EXMAPLE

Packets	Deadline	Transmission Time
P1	10	5
P2	14	2
P3	15	1
P4	6	3

TABLE IV. OPFS SCHEDULE BY EXMAPLE

OPFS	Deadline	Transmission Time	Dwell Time
P4	6	3	3
P1	10	5	8
P2	14	2	10
P3	15	1	11

Firstly, we take a look at a simple example of packet set, as shown in TABLE II. The LGQ of the router contains a set of packets, which are associated with the properties of deadline and transmission time. We assume the unit of deadline and transmission time is ms. An OPFS schedule is illustrated in TABLE IV. The average stay time is 8 ms.

TABLE V. BABB SCHEDULE BY EXAMPLE

BABB	Deadline	Transmission Time	Dwell Time
P3	15	1	1
P4	6	3	4
P1	10	5	9
P2	14	2	11

The proposed BABB algorithm is able to find the optimal schedule as shown in TABLE V. The average stay time is 25/4=6.25 ms, which decreases nearly 30% compared to the OPFS schedule.

 ISBN: 978-1-61208-964-5

Besides BABB and OPFS, the following three scheduling schemes are included in the performance evaluation and comparisons:

- First In First Out (FIFO): which is the same as FCFS in [9]. The packets are scheduled according to their arrival time at the outgoing queue of the router. The packet which arrives earliest is scheduled firstly.
- Smallest Transmission Time First (STTF): which is similar to SJF in [10]. The packets are scheduled based on the incremental order of the transmission time. The transmission time is proportional to the packet size if the outgoing link bandwidth is fixed. Thus, in STTF, the minimum-sized packet is scheduled firstly.
- Largest Transmission Time First (LTTF): the packets are scheduled based on the decremental order of the transmission time. Thus, in LTTF, the most bulky packet is scheduled firstly.

The simulator is built in C++. Two types of performances are being evaluated: (1) Packet delivery success rate, which is defined as the ratio of the packets that get scheduled appropriately and meet their corresponding deadline constraints. (2) Average stay time of the packets which could satisfy their deadline constraints under an adopted scheduling scheme.

The packets' transmission time and deadline are deliberately designed to make sure that there is at least one feasible schedule, under which all packets could be transmitted out of the router within their corresponding allowance. The feasible schedule is denoted as OPFS, as introduced above. The transmission time of the packets is randomly generated in the range [1,10] ms, then the deadline of each packet is assigned by adding some extra time compared to its stay time. **Deadline gap ratio** is defined as the ratio between the upper bound of this additional time and the transmission time upper bound. For example, if the deadline gap ratio is 3, then the deadline of a packet is given by adding a random number between [0, 10*3] = [0, 30] ms to the stay time. The packets' order is then shuffled to mimic the arrival time of those packets in the router.

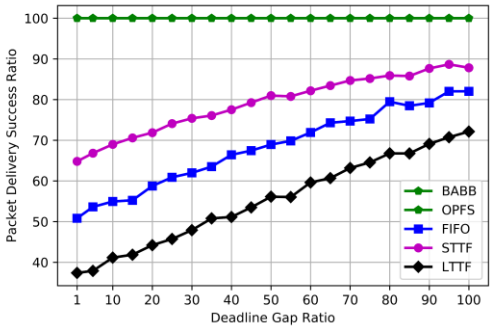

Figure 7. Packet delivery success ratio vs. deadline gap ratio

Figure 7 shows the packet delivery success ratio versus the deadline gap ratio with different packet schedule schemes. With the above simulation configurations, BABB and OPFS can always achieve the 100% packet delivery success ratio. However, if FIFO, STTF or LTTF is used, the stay time of some packets is not able to meet the deadline expectations. No matter how big the deadline gap ratio is, the positions of the packets to be scheduled in the outgoing queue are not appropriately manipulated to satisfy all packets' deadline constraints with FIFO, STTF or LTTF. The packet delivery success ratio of FIFO, STTF or LTTF increases along with the increment of the deadline gap ratio. BABB and OPFS proposed in this paper, on the other hand, guarantee all packets to be able to reach the receivers successfully without missing their deadlines, even when deadlines are very tight.

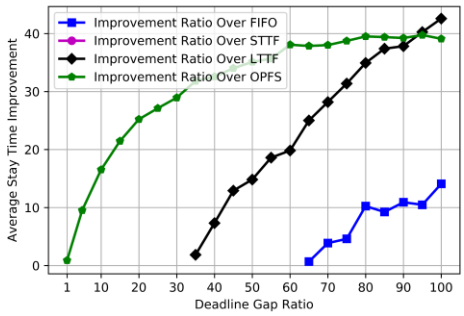

Figure 8. Average stay time improvement ratio vs. deadline gap ratio

Figure 8 shows the average stay time improvement ratio of BABB over the other scheduling schemes versus the deadline gap ratio. It is noticeable that in Figure 8, the improvement ratio of BABB over LTTF is plotted starting from the deadline gap ratio of 35, over FIFO is plotted starting from the deadline gap ratio of 65, while the line of improvement ratio of STTF is missing. The reasons are given as follows: Since we only evaluated the average stay time for those packets which could satisfy their deadline constraints, it means that the dropped packets due to missing deadline are not counted. Thus, the comparison of average stay time is not fair between BABB and FIFO/STTF/LTTF. STTF schedules the packets with the smallest transmission time firstly, those packets with larger transmission time are dropped eventually. According to (6), the total stay time for those successfully transmitted packets with STTF always has the minimal value. When the deadline gap ratio becomes large enough, BABB can improve over FIFO and LTTF. In the scenario that all packets are successfully transmitted, the fair comparison between BABB and OPFS is also shown in Figure 8. BABB achieves the minimal average stay time, while OPFS is the first solution that satisfies the constraints and the backtracking process stops at this point. As a result, BABB always accomplishes better average stay time performance than OPFS by 10% to 40% when the deadline gap ratio increases from 10 to 100.

Next, we evaluate the impact of the number of packets in the outgoing queue to the two types of considered performances. Figure 9 shows the packet delivery success ratio versus the packet number. BABB and OPFS proposed in the paper can always achieve the in-time guarantee for all packets, no matter how many packets are in the outgoing queue, which is very captivating property for latency-sensitive packet forwarding. It means that BABB or OPFS can be used in different types of network nodes, whether they have low or high volume of traffic. For other scheduling schemes, the packet success ratio declines

 ISBN: 978-1-61208-964-5

with the number of packets. When the packet number reaches 200, the packet delivery success ratio of FIFO, STTF or LTTF does not fluctuate much. The packet delivery success ratio of LTTF is the worst, with only 30% success rate.

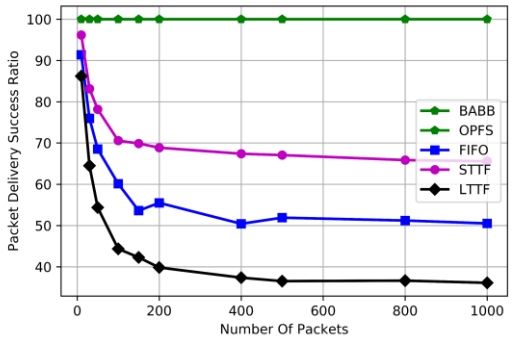

Figure 9. Packet delivery success ratio vs. packet number

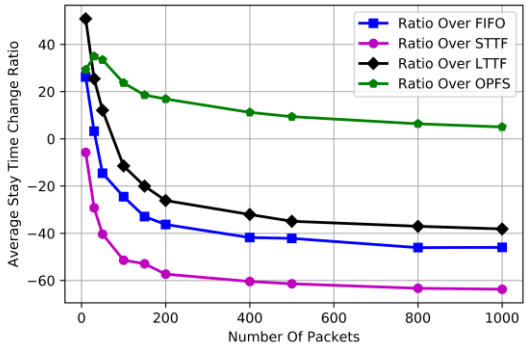

Figure 10. Average stay time change ratio vs. packet number

Figure 10 shows the average stay time change ratio of BABB over other scheduling schemes. As we explained earlier, this comparison only makes sense when the number of successfully delivered packets is the same. However, when the number of packets in the outgoing queue increases, FIFO, STTF and LTTF's packet delivery success ratio drops rapidly. The number of packets counted for the average stay time calculation becomes much less than the one counted in BABB and OPFS. We only draw those points with negative values to show that FIFO, STTF or LTTF are not a desirable scheduling scheme for packets with in-time guarantee, since they would cause too many packets dropping due to missing latency deadline. On the other hand, we can observe that the improvement ratio of BABB over OPFS decreases when there is a very large number of latency-sensitive packets in a router's outgoing queue. Thus, when the number of concurrent packets that require in-time guarantee becomes large, an OPFS scheme is good enough to be adopted to achieve the precise latency performance with reasonable low processing overhead in the router.

V. CONCLUSION

This paper leverages the New IP framework to carry an in-time guarantee contract, as well as the associated deadline constraint and other metadata in the packet, such that each intermediate router on the path from the source to the destination can execute more sophisticated scheduling on multiple packets on the same outgoing port instead of traditional statistical multiplexing. The proposed backtracking solution with bound-and-branch improvement can achieve the minimal average stay time of the packets which require in-time guarantee, and the successful delivery ratio of packets is maximized compared to any other scheduling schemes.

REFERENCES

[1] FG-NET-2030, Sub group 2, "New Services and Capabilities for Network 2030: Description, Technical Gap and Performance Target Analysis," 2019.

[2] FG-NET-2030, "Network 2030 - A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond," May 2019.

[3] L. Dong, L. Han, and R. Li, "Support Precise Latency for Network Based AR/VR Applications with New IP," EAI MobiMedia 2020.

[4] R. Braden, D. Clark, and S. Shenker, "RFC 1663: Integrated Services in the Internet Architecutre: an Overview," IETF, Jun. 1994.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475: An Architecture for Differentiated Services," IETF, Dec. 1998.

[6] L. Han, Y. Qu, L. Dong and R. Li, "A Framework for Bandwidth and Latency Guaranteed Service in New IP Network," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 85–90.

[7] L. Dong and L. Han, "New IP Enabled In-Band Signaling for Accurate Latency Guarantee Service," 2021 IEEE WCNC, pp. 1–7.

[8] B. Vamanan, J. Hasan, and T. Vijaykumar, "Deadline-aware datacenter TCP (D2TCP)," ACM SIGCOMM Computer Communication Review, vol. 42, no. 4, pp. 115–126, 2012.

[9] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, "Better never than late: Meeting deadlines in datacenter networks," SIGCOMM Computer Communication Review, vol. 41, no. 4, pp. 50–61, 2011.

[10] C.-Y. Hong, M. Caesar, and P. Godfrey, "Finishing flows quickly with preemptive scheduling," ACM SIGCOMM Computer Communication Review, vol. 42, no. 4, pp. 127–138, 2012.

[11] V. K. Gopalakrishna, Y. Kaymak, C. Lin and R. Rojas-Cessa, "PEQ: Scheduling Time-Sensitive Data-Center Flows using Weighted Flow Sizes and Deadlines," 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), 2020, pp. 1–6.

[12] R. Li, K. Makhijani and L. Dong, "New IP: A Data Packet Framework to Evolve the Internet : Invited Paper," 2020 IEEE 21st HPSR, 2020, pp. 1–8.

[13] R. Li, A. Clemm, U. Chunduri, L. Dong, and K. Makhijani, "A New Framework And Protocol For Future Networking Applications," ACM Sigcomm NEAT workshop 2018, pp. 21–26.

[14] R. Li, K. Makhijani, H. Yousefi, C. Westphal, L. Dong, T. Wauters, and F. De Turck, "A Framework For Qualitative Communications Using Big Packet Protocol," ACM Sigcomm NEAT workshop 2019, pp. 22–28.

[15] L. Dong and R. Li, "In-Packet Network Coding for Effective Packet Wash and Packet Enrichment," IEEE Globecom Workshops, 2019, pp. 1–6.

[16] L. Dong, K. Makhijani and R. Li, "Qualitative Communication Via Network Coding and New IP : Invited Paper," 2020 IEEE 21st HPSR, 2020, pp. 1–5.

 ISBN: 978-1-61208-964-5

IoT Security: A Basic IoT Hardware Security Framework

Christoph Haar
Hochschule für Telekommunikation Leipzig
Leipzig, Germany
Email: haar@hft-leipzig.de

Erik Buchmann
Leipzig University
Leipzig, Germany
Email: buchmann@informatik.uni-leipzig.de

***Abstract*—More and more Internet of Things (IoT) devices are being used in companies today. The usage harbors great risks, because numerous observations have shown that many IoT devices on the market are insecure. For this reason, well-known security authorities such as the German Federal Office for Information Security (BSI) or the National Institute for Standards and Technologies (NIST) have established standards and guidelines, considering known threats and common security practices for IoT devices. They focus on software security as well as the secure planning and usage of IoT devices. Hardware security on the other hand is less considered. In this paper, we develop a basic IoT hardware security framework that can be implemented into existing security concepts. To reach this goal, we compare three official IoT security standards to identify important hardware threats. After that, we perform a risk identification for four different IoT devices to find out if the mentioned hardware threats really apply to different application scenarios. Based on the results, we develop a basic IoT hardware security framework. Our research has shown that the hardware threats mentioned in the official IoT security standards are of great importance. Because they apply to a wide range of different application scenarios for IoT devices, we implemented them in our basic IoT hardware security framework.**

Keywords – IoT Security Standards, IoT Hardware Threats, Risk Identification, Security Framework.

I. INTRODUCTION

Over the last years, the number of connected IoT devices in enterprises has increased rapidly [1]. They are used to improve the productivity of business processes or to massively reduce costs [2] [3]. On the other hand, there are numerous threats associated with their use [4]–[7]. Observations have shown that many IoT devices on the market are insecure [8]. Attackers can compromise them, spy out internal data and interrupt services. The damage caused by such attacks can be existence-threatening. Official security authorities, such as the German Federal Office for Information Security (BSI), the National Institute for Standards and Technology (NIST) or the European Union Agency for Cybersecurity (ENISA) have already addressed this issue. Numerous free IoT security standards also have been published during the last years. They consider known threats and common security practices. A closer examination reveals that there is no uniform process for IoT hardware security. The hardware is the basis of any IoT device [9]. Thus, there should be a structured process for basic protection. The aim of this paper is to develop a basic IoT hardware security framework that can be used to protect any IoT device on a basic level. For this purpose, we compare three official IoT security standards, published by BSI, NIST and ENISA to identify important IoT hardware threats mentioned in the standards. The result of this comparison serves as the basis for a risk identification. We select four different and commonly used IoT devices and perform a risk identification to be able to find out if the mentioned hardware threats really apply to different application scenarios. Based on the results, we derive a basic IoT hardware security framework that includes the identified risks. Our basic IoT hardware security framework consists of three steps. Following these steps ensures a basic protection of any IoT device regardless of its application scenario.

Paper structure: Section II contains the related work. In Section III, we perform a risk identification and generalize our findings. In Section IV, we define the IoT hardware security framework, followed by a discussion in Section V. Section VI concludes the paper.

II. RELATED WORK

In this section, we introduce three official IoT security standards and compare, which hardware threats are mentioned. In this way, we are able to identify particularly important threats. At the end of this section, we give a brief introduction into risk identification.

A. IoT Hardware Security

The security of IoT devices should start with the security of the hardware because it is the basis of any device [9]. There are already numerous publications, describing hardware threats and suitable security practices for IoT devices [9]–[13]. Also official security standards have been developed and published to ensure a secure usage of IoT devices and all their data, as well as the entire system on which they are operated. Each standard considers hardware security differently.

a) BSI: The BSI describes 47 product and technology-neutral elementary threats in the BSI standard 200-3 [14]. They describe general risks for IT systems, regardless of their application scenario. Not every elementary threat is affecting each part of an IT system. The BSI lists all elementary threats that are addressing a certain element of the IT system in the

 ISBN: 978-1-61208-964-5

IT Grundschutz Compendium [15]. For example, the module "'SYS.4.4 General IoT Devices"' contains an appendix which considers 20 of the 47 elementary threats that are affecting IoT devices as Table I illustrates.

TABLE I
IoT Elementary Threats

BSI Elementary Threats For IoT Devices
G 0.2 Bad Environmental Conditions
G 0.4 Pollution, Dust, Corrosion
G 0.8 Disruption of Power Supply
G 0.9 Failure or Disruption of Communication...
G 0.14 Interception of Information / Espionage
G 0.16 Theft of Devices, Storage Media and...
G 0.18 Poor Planning or Lack of Adaptation
G 0.19 Disclosure of Sensitive Information
G 0.20 Information or Products from an...
G 0.21 Manipulation with Hardware
G 0.23 Access to IT Systems
G 0.24 Destruction of Devices or Storage Media
G 0.25 Failure of Devices or Systems
G 0.26 Malfunction of Devices or Systems
G 0.28 Software Vulnerabilities or Errors
G 0.29 Violation of Laws or Regulations
G 0.30 Unauthorised Use or Administration of...
G 0.38 Misuse of Personal Information
G 0.39 Malware
G 0.40 Denial of Service

These threats apply to all IoT devices regardless of their application scenario or security properties. They consider the hardware and software, as well as a secure planning and usage. Because they are completely unsorted, it is up to the user to identify the hardware related threats.

b) NIST: The *National Institute for Standards and Technology (NIST)* published several drafts for IoT security in 2020 [16]–[20]. They consider the acquisition and implementation of IoT devices in companies and give an overview about important steps that need to be considered, when planning to use IoT devices. They also describe how the data of IoT devices can be protected, as well as the entire system. As Table II illustrates, the NIST does not use elementary threats like the BSI but similar hardware threats are mentioned.

TABLE II
NIST IoT Hardware Threats

NIST Hardware Threats For IoT Devices
Physical Damage
Unauthorized Access
Hardware Manipulation

The NIST specifies the mentioned threats. Physical damage includes vandalism, as well as damage through high or low temperatures and humidity [16]. This is similar to G 0.2 Bad Environmental Conditions and G 0.24 Destruction of Devices or Storage Media. It is also mentioned that IoT devices may have to endure physical damage through extreme temperatures that could be caused by a fire. Unauthorized Access is considered by considering the restriction of network and local interfaces [17]. That means, the IoT device must be able to deactivate local and network interfaces. In this way, open communication interfaces could be deactivated to avoid unauthorized access. This covers the elementary threat G 0.23. Hardware manipulation is addressed by mentioning the use of unique physical identifiers [17]. There is no precise definition of what is meant by unique physical identifier, but there are approaches, such as PUFs that leads to an unique behavior of the device. Any physical manipulation would change this unique behavior and detect the manipulation. This is similar to G 0.21 Manipulation of Hardware.

The hardware threats are mentioned in different sections of the drafts but there is no separate section or even a clear process that defines general steps for protecting the hardware.

c) ENISA: The *European Union Agency for Network and Information Security* (ENISA) [21] published the Baseline Security Recommendations for IoT. This publication contains a hardware security section. It is addressing IoT security challenges and provides general security recommendations when using IoT devices. Many hardware threats are considered as shown in Table III.

TABLE III
ENISA IoT Hardware Threats

ENISA Hardware Threats For IoT Devices
Elemental Threats
Environmental Threats
Physical Damage
Hardware Manipulation
Power Loss
Data Interception

The mentioned threats are also similar the elementary threats from the BSI. ENISA separates the threat physical damage. It is only caused through vandalism. Threats like water and fire are not considered as physical damage but as elemental threats. Environmental threats on the other hand are causing damage through high or low temperatures. Interception is not only a physical threat. It is mentioned that all kinds of data interception has to be considered. That could be the interception of data traffic or stored data but also the interception of electromagnetic radiation emitted by the hardware. Also the usage of hardware that provides security features like specialised security chips to detect physical manipulations is recommended. Disruption of power supply is another mentioned threat. Even though ENISA has introduced a separate section for hardware security, there is still no clearly defined process for hardware protection.

 ISBN: 978-1-61208-964-5

It can be clearly seen that the mentioned hardware threats are very similar in the three security standards. Sometimes the threats are just categorized differently. For example the NIST considers fire as physical damage. For ENISA, on the other hand, it is an elementary threat. However, both standards consider fire as a threat.

Since the hardware threats are so similar in the individual standards, we use the 47 BSI elementary threats as a basis for our risk identification. The elementary threats are also product and technology-neutral and compatible with other international catalogs and standards.

B. Risk Analysis

In 2017, the BSI published the current version of the BSI-Standard 200-3 [14] that defines the steps of a risk analysis. The first step is the risk identification. Threats that are realistic for a certain target object and its application environment are identified by IT-security experts in a brainstorming session. IT-security experts means information security officers, responsible specialists, administrators, users of the target object and if available external expert. The risk identification is a very important step, because not identified threats will lead to a major security gap. It is only possible to classify threats and define appropriate security practices for identified threats. ENISA also starts the risk analysis process by considering security incidents that have become public over the last years. Also a threat taxonomy is illustrated [21]. Other official security authorities like NIST [22] also define a risk management process that starts with a risk identification.

They all understand risk identification as a fundamental step for further risk management activities.

III. RISK IDENTIFICATION

In this section, we perform a risk identification for the hardware of four different IoT devices. In the first step, we select four IoT devices and list all their hardware components. We use the hardware components to determine, which hardware threats are affecting the IoT device. After that, we analyze the 47 elementary threats from the BSI and select those that potentially address the hardware. This is necessary because the elementary threats from the BSI are not limited to the hardware. In the next step, we perform the risk identification. In particular, we systematically analyze for each IoT device which hardware components are affected by the potential hardware threats. Finally, we analyze and generalize our findings.

A. IoT Devices

For our investigation, we select four commonly used IoT devices. The IoT Security Camera [23], the IoT Smoke Detector [24], the IoT Soil Temperature Sensor [25] and the IoT Power Outlet [26]. The application scenarios of the devices are as different as possible. In this way, we are able to determine if the mentioned IoT hardware threats from the BSI really apply to a wide range of different application scenarios.

Table IV gives a brief overview of all hardware components of each IoT device.

TABLE IV
IoT Device Hardware Components

Security Camera	**Smoke Detector**
Cables, Camera, Case, Infrared LED's, Micro SD socket, Microphone, Motherboard, Processor, Sensors	Battery, Case, LED, Motherboard, Processor, Reset Button, Sensors, Speaker
Soil Temp. Sensor	**Power Outlet**
Antenna, Battery, Case, Motherboard, processor, Sensors	Case, Motherboard, Processor, Sensors, Socket Connector

B. Potential IoT Hardware Threats

The elementary threats, defined by the BSI cover a wide range of potential threats for an entire company. They also consider many hardware threats. We do not consider all of them in this paper. Table V summarizes the hardware threats we consider. We consider damage caused by G 0.1 Fire, G 0.2 Bad Environmental Conditions and G 0.3 Water because these threats are always conceivable. For example, Water damage can be caused by simple rain. Fire can be caused by a short circuit and bad environmental conditions can stem from to high or low temperatures. We also consider that an IoT device can be damaged by excessive pollution. Dust and soil can intrude through leaks and damage hardware components. This threat is considered in G 0.4 Soiling, Dust, Corrosion.

TABLE V
Potential IoT Hardware Threats

Potential IoT Hardware Threat
G 0.1 Fire
G 0.2 Bad Environmental Conditions
G 0.3 Water
G 0.4 Soiling, Dust, Corrosion
G 0.8 Disruption of Power Supply
G 0.12 Electromagnetic Interference
G 0.13 Interception of Radiation
G 0.21 Manipulation of Hardware
G 0.23 Unauthorized Entry
G 0.24 Destruction

On the other hand, we do not consider G 0.5 Natural Catastrophes, G 0.6 Catastrophes in the Environment and G 0.34 Attack. These threats do affect the entire hardware but they are extreme events. Normally, IoT devices cannot be protected against such incidents. G 0.8 Disruption of Power Supply considers service interruptions or physical damages caused by a sudden power loss. The reason for this could be a storm that could occur at any time. Thus, we do also consider this threat. The BSI also mentions G 0.12 Electromagnetic Interferences

 ISBN: 978-1-61208-964-5

as an elementary threat. Although it is not mentioned as an elementary threat to IoT devices, the BSI points out that all electronic devices are affected by G 0.12. Furthermore, the BSI emphasizes that also wireless communication like WI-FI can be affected by electromagnetic interferences. Because IoT devices are electronic devices and they do communicate wireless, we consider this threat. Data can be revealed through electromagnetic interferences. The BSI is mentioning this threat in G 0.13 but does not consider it as an elementary threat for IoT devices. ENISA on the other hand mentions that all threats that intentionally or unintentionally reveal data has to be considered. Due to this fact, we also consider G 0.13 Interception of Radiation. This could also be considered as G 0.14 espionage but this security practice also includes non hardware aspects like the interception of data traffic. Due to this fact, we do not consider G 0.14 espionage. We do also not consider G 0.16 Theft of Devices, because the hardware is not necessarily affected by a theft. G 0.21 Manipulation of Hardware means every willful change of the original hardware that leads to an unnoticed change in behavior. Since devices are usually purchased from unknown manufacturers, manipulation of the hardware cannot be ruled out. Thus, we also consider this threat. With G 0.23 Unauthorized Entry, the BSI considers physical access via unprotected communication interfaces like USB ports [11]. Because many IoT devices have such open communication interfaces, we also consider this threat. The BSI differentiates between G 0.41 Sabotage and G 0.24 Destruction. In both cases the aim is to damage the IT systems. Destruction means willful attacks against the device by impact. Sabotage describes the manipulation of the environment that leads to a damage of the IT system. For example closing the ventilation slots of a server, which leads to overheating and finally damage or destroy the server. Since both threats have the same goal, we summarize and consider them in G 0.24 Destruction.

C. Implementation

In this step, we implement the risk identification. That means, we analyze which hardware components are affected by the potential hardware threats. Furthermore, we check for each of the four IoT devices whether it has the affected hardware component. **G 0.1 Fire** A fire cannot be assigned to a specific hardware component. It could lead to a damage of all hardware components of any IoT device. Therefore, we consider all four IoT devices as affected.

G 02. Bad Environmental Conditions All four devices have a clear defined operating temperature. That makes their entire hardware affected by G 0.2 Bad Environmental Conditions. The security camera can be operated between -10 and +55 degrees. The smoke detector and the power outlet on the other hand can only be operated above 0 degrees up to +40 degrees. The soil temperature sensor has the largest operating temperature range from -40 to +80 degrees. If the devices are operated outside the specified operating temperature, all hardware components could be damaged. Thus, we consider all four devices as affected.

G 03. Water In case of a water intrusion, non electric components would not be affected. For example, all four devices have a plastic case. This case can be wet but it would not be damaged by the water. The damage would caused to electronic components. All four devices have electronic components like sensors, processors or LEDs. Thus, we consider all as affected.

G 0.4 Soiling, Dust, Corrosion Like G 0.3, this threat does not affect non electric components like the plastic case or buttons. Soiling, dust and corrosion would only cause damage to electronic components. For example the sensors of all four IoT devices could be disturbed by to much pollution or the processor could overheat. Furthermore, the electronic components could rust after moisture has entered the device. Because all four devices have electronic components, we consider all as affected.

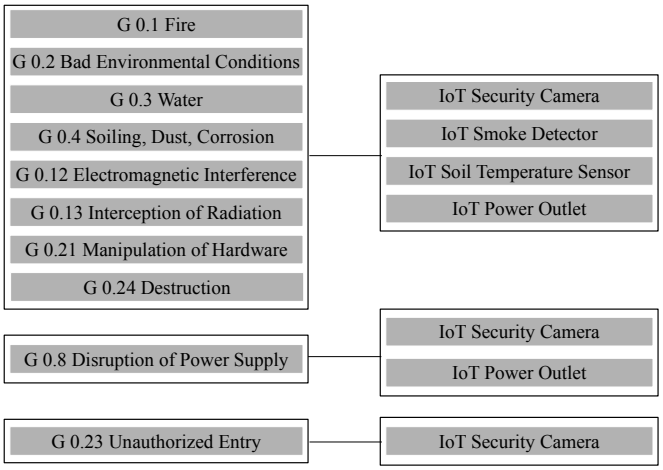

Fig. 1. IoT Device Threats

G 0.8 Disruption of Power Supply The IoT security camera and the IoT power outlet have a power connection. These devices do not have batteries what makes them dependent from external power sources. As soon as the power supply is interrupted, both devices will be turned off immediately. Due to this fact, we consider them as affected. The IoT smoke detector and the IoT temperature sensor are battery operated. That means, they are not connected to power sources and therefore unaffected by this threat.

G 0.12 Electromagnetic Interference All electronic devices can be disturbed by electromagnetic interferences. That means, every electronic hardware component is affected. Since each of the four IoT devices is an electronic device, we consider them as affected.

G 0.13 Interception of Radiation All electronic devices emit radiation. That means, every electronic hardware component is affected. Since each of the four IoT devices is an electronic device, we consider them as affected.

G 0.21 Manipulation of Hardware Manipulation of hardware can also not be assigned to a specific hardware com-

 ISBN: 978-1-61208-964-5

ponent. For example, the case of all 4 devices could be manipulated in such a way that water intrudes and cause damage. The sensors could also be manipulated so that incorrect measurement results are transmitted. Thus, we consider all hardware components of each device to be affected.

G 0.23 Unauthorized Entry The IoT security camera is the only device that has an open communication interface. It is possible to connect SD cards. An attacker could use this SD card socket to gain unauthorized access to the IoT security camera or the entire network. Due to this fact, we consider the IoT security camera as affected. The other three devices not have any open communication interfaces, thus we consider them as not affected.

G 0.24 Destruction It is always possible for an attacker to intentionally destroy any hardware component of each of the four devices. Thus, we consider all four devices as affected.

Figure 1 summarizes the results of our risk identification.

It can be seen that each of the four devices is addressed by at least one threat. G 0.23 is only affecting the IoT camera. This is because it is the only device with an open communication interface. G 0.8 Disruption of Power Supply is affecting the IoT camera and the IoT power outlet because they are connected to the buildings electricity. All other threats are affecting each of the four IoT devices. With our risk identification, we were able to confirm that the threats mentioned in the official IoT security standards apply to different application scenarios. In the next step, we generalize our results to be able to use them as a basis for our framework.

a) Generalization: As we can see, a hardware threat can only affect an IoT device, if it has the addressed hardware component. For example, G 0.23 Unauthorized Entry can only affect IoT devices that have open communication interfaces like USB ports. Figure 2 illustrates hardware components that are affected by the potential IoT hardware threats.

We were able to determine that G 0.1, G 0.2, G 0.21 and G 0.24 are affecting every hardware component. That means, as soon as a device exists, *all components* are addressed. G 0.3, G 0.4, G 0.12 and G 0.13 are affecting all *electronic components* in general. Since every IoT device communicates electronically, it also consists of electronic components. Due to this fact, all IoT devices are affected. G 0.8 is addressing *power connections* to the building's electricity. That means, battery operated devices are generally not affected. G 0.23 is affecting all IoT devices with *open communication interfaces*.

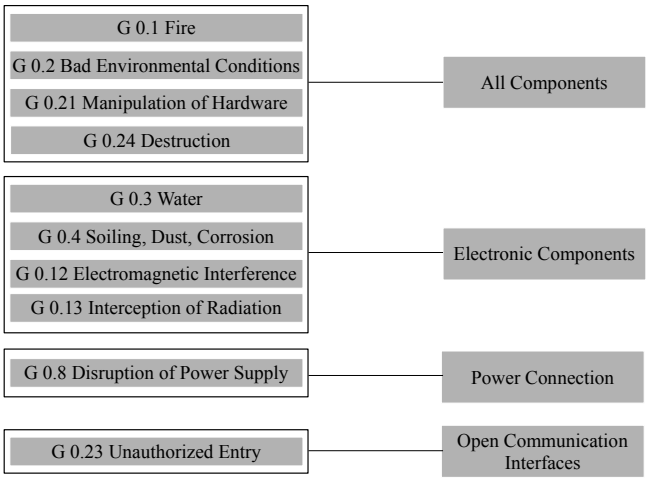

Fig. 2. Affected Hardware Components

IV. FRAMEWORK DEFINITION

With our risk identification, we found out, which of the hardware threats mentioned in the official IoT security standards apply to different IoT devices. In this way, we were able to confirm that these threats are of particular importance for the hardware of IoT devices. These threats must either be considered for all IoT devices or at least for a large number of different application scenarios. Due to this fact, we define a basic IoT hardware security framework that considers these threats in this section. There is a total of four hardware threats (G 0.1, G 0.2, G 0.21 and G 0.24) that apply to all hardware components of each of the four IoT devices. Because they are also mentioned in the official IoT security standards, they should definitely be considered when securing IoT devices. Same holds for the four hardware threats (G 0.3, G 0.4, G 0.12 and G 0.13) that are affecting all electronic hardware components of each of the four IoT devices. Because every IoT device has electronic components, they should also definitely be considered. There are two hardware threats that only apply, in case the IoT device has a certain hardware component. G 0.8 is only affecting devices that have a power connection. G 0.23 requires open communication interfaces like USB ports or SD card slots for example. In our framework, it has to be checked, if the IoT device have these components. If the device not have the hardware components, G 0.8 and G 0.23 not have to be considered. This process is illustrated by the following pseudocode.

```
for EACH IoT-Device x do
    SECURE G 0.1, G 0.2, G 0.3, G 0.4,
    G 0.12, G 0.13, G 0.21, G 0.24 ON x
    if x has power connection then
        SECURE G 0.8 ON x
    end if
    if x has open communication interface then
        SECURE G 0.23 ON x
    end if
end for
```

x is representing a certain IoT device which goes through the framework. SECURE indicates a function. If SECURE is ON, the hardware threat is affecting the IoT device and security practices has to be considered for a certain hardware threat like G 0.8 for example. Otherwise, the hardware threat is not affecting the device and no security practices has to be implemented for this threat.

 ISBN: 978-1-61208-964-5

V. DISCUSSION

The official security authorities consider different aspects of IoT security e.g., a secure planning, implementation and usage of IoT devices, data security, as well as software and hardware security. Our comprehensive review and comparison of three official IoT security standards, published by the BSI, NIST and ENISA has shown that the mentioned hardware threats are very similar within these standards. With our risk identification, we were also able to confirm that the mentioned hardware threats indeed affect a wide range of different application scenarios for IoT devices. Thus, it is meaningful to define a framework that includes these threats. However, suggesting appropriate security practices for these threats is not part of our framework, because they are already described in the BSI module SYS.4.4. It is also important to mention that further security measures are necessary. Our framework serves as a basic hardware protection. It includes IoT hardware threats that are affecting different IoT devices, regardless of their application scenarios or security requirements. Additional threats must be identified for each IoT device. In this way, our framework can be included into other security activities. For example, the BSI defines steps for a risk analysis in the BSI standard 200-3 [14] to identify additional threats and security practices according to specific application scenarios and security requirements. Our basic IoT hardware security framework could be implemented before the risk analysis. In this way it can be embedded into existing security concepts.

VI. CONCLUSION

This paper was motivated by the fact that official IoT security standards do not consider a uniform procedure for a basic hardware protection of IoT devices. The aim was to develop a basic IoT hardware security framework that can be implemented into existing security concepts. For this purpose, we analyzed three official IoT security standards, publishes by the BSI, NIST and ENISA. We compared which hardware threats are mentioned. These threats seem to be of great importance for IoT security in general. By performing a risk identification for four different IoT devices, we checked whether these threats really apply to different application scenarios. We were able to confirm the importance of these threats. In the next step, we used them to develop our basic IoT hardware security framework. This framework consists of a total of 10 hardware threats that are affecting different application scenarios for IoT devices. It can be used as a basic hardware protection for IoT devices, and it can be included into existing security concepts.

REFERENCES

[1] Business Wire, "Strategy analytics: Internet of things now numbers 22 billion devices but where is the revenue?" 2021.

[2] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[3] R. Gupta and R. Gupta, "ABC of Internet of Things: Advancements, benefits, challenges, enablers and facilities of IoT," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–5.

[4] W. Zhao, S. Yang, and X. Luo, "On threat analysis of IoT-based systems: A survey," in *2020 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 2020, pp. 205–212.

[5] V. Hassija *et al.*, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[6] K. N. Mallikarjunan, K. Muthupriya, and S. M. Shalinie, "A survey of distributed denial of service attack," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2016, pp. 1–6.

[7] M. Hosseinzadeh, B. Sinopoli, and E. Garone, "Feasibility and detection of replay attack in networked constrained cyber-physical systems," in *2019 57th annual allerton conference on communication, control, and computing (Allerton)*. IEEE, 2019, pp. 712–717.

[8] J. Kleinhans, "Internet of insecure things," https://www.stiftung-nv.de/sites/default/files/internet_of_insecure_things.pdf, Accessed March 2022, 2017.

[9] S. Sidhu, B. J. Mohd, and T. Hayajneh, "Hardware Security in IoT Devices with Emphasis on Hardware Trojans," *Journal of Sensor and Actuator Networks*, vol. 8, no. 3, p. 42, 2019.

[10] J. Milosevic, N. Sklavos, and K. Koutsikou, "Malware in IoT software and hardware," *Conference: Workshop on Trustworthy Manufacturing and Utilization of Secure Devices*, 2016.

[11] T. Alladi, V. Chamola, B. Sikdar, and K.-K. R. Choo, "Consumer IoT: Security Vulnerability Case Studies and Solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 17–25, 2020.

[12] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017, pp. 32–37.

[13] L. I. P. Technik, "SONOFF S55 Wi-Fi Smart Waterproof Socket," 2020.

[14] Federal Office for Information Security BSI, "BSI-Standard 200-3: Risk Analysis based on IT-Grundschutz," *https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Grundschutz/International/bsi-standard-2003_en_pdf.pdf, Accessed March 2022*, 2017.

[15] ——, "BSI IT Grundschutz Compendium Edition 2019," *https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Grundschutz/International/bsi-it-gs-comp-2019.pdf, Accessed March 2022*, 2019.

[16] National Institute of Standards and Technology, "IoT Device Cybersecurity Guidance for the Federal Government," *https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-213-draft.pdf, Accessed March 2022*, 2020.

[17] ——, "IoT Device Cybersecurity Capability Core Baseline," *https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259A.pdf, Accessed March 2022*, 2020.

[18] ——, "IoT Non-Technical Supporting Capability Core Baseline," *https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259B-draft.pdf, Accessed March 2022*, 2020.

[19] ——, "Creating a Profile Using the IoT Core Baseline and Non-Technical Baseline," *https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259C-draft.pdf, Accessed March 2022*, 2020.

[20] ——, "Profile Using the IoT Core Baseline and Non-Technical Baseline for the Federal Government," *https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259D-draft.pdf, Accessed March 2022*, 2020.

[21] European Union Agency for Cybersecurity, "Baseline Security Recommendations for IoT," https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot/, Accessed March 2022, 2017.

[22] National Institute of Standards and Technology, "Guide for Conducting Risk Assessments ," *https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf, Accessed March 2022*, 2012.

[23] Reolink, "Most Popular 5MP PoE Security IP Camera," https://reolink.com/gb/product/rlc-410/ Accessed March 2022, 2021.

[24] X-Sense Innovations Co., Ltd., "X-Sense XS01-WT Wi-Fi Smoke Detector," https://www.x-sense.com/products/x-sense-xs01-wt-wi-fi-smoke-alarm Accessed March 2022, 2021.

[25] Sigfox Foundation, "Remote Signals Soil Temperature Monitor," https://partners.sigfox.com/products/remote-signals-soil-temperature-monitor Accessed March 2022, 2021.

[26] Allterco Robotics, "Make Your Home Smart," https://shelly.cloud/documents/catalogues/catalogue.pdf Accessed March 2022, 2021.

 ISBN: 978-1-61208-964-5

Identifying Significant Parameters of the US Bridges

Prasad Chetti
School of CSIS
Northwest Missouri State University
Maryville, MO, USA
Email: pchetti@unomaha.edu

Hesham Ali
College of Information Science and Technology
UNO, Omaha, NE, USA
Email: hali@unomaha.edu

***Abstract*— Various bridge parameters, either alone or in combination with other parameters, significantly affect the performance of the bridge decks in various regions. Identifying such parameters and/or their interaction effects allows the bridge authorities to understand and assess the bridge decks' performance in different regions. This paper analyzes 1,732 same-age US bridges from the national bridge inventory dataset with respect to various independent parameters along with bridge deck condition rating as the dependent parameter. The Kruskal-Wallis test was performed on these bridge decks for various independent parameters. Multiple pairwise comparisons between groups were also performed using the Wilcoxon test. Results show that material, design, and region affect the deck condition ratings of the bridges both individually and while interacting with each other. Further, it is observed that bridges made of concrete material with stringer multi-beam girder design, and which reside in the Highplains region perform the worst, whereas the prestressed concrete bridges with the same design and which reside in the same region perform the best.**

Keywords*— National Bridge Inventory (NBI) Dataset; Kruskal-Wallis Test; Bridge Condition Ratings.

I. INTRODUCTION

The national bridge inventory database is being maintained by the U.S. Federal HighWay Administration (FHWA) since 1992. This database has the information of more than 600,000 bridges which includes both culverts and highway bridges. Information of each bridge is recorded with more than 100 parameters. Each bridge in the U.S. is inspected once every two years [7][8]. In 2017, U.S. bridges received a C+ grade for their overall performance [1]. Bridge decks', superstructures', and substructures' condition is verified by the bridge engineer based on the inspection frequency of the bridge [2]. Bridge decks are assigned a rating value between '0' and '9', as shown in Table 1. Rating condition '9' is assigned to an excellent condition rating bridge and '0' is assigned to a failed bridge. Bridges deteriorate for various reasons which include age of the bridge, material used to construct it, design used, average daily traffic on the bridge, geographical region of the bridge, etc. Population analysis models in association with correlation networks were applied on the civil infrastructures and in financial markets to show that various significantly enriched parameters effect the dependent parameters [2]-[6][11]. Deterioration models were applied to estimate the bridge condition ratings [9] and stochastic systems have been analyzed using stochastic/probabilistic model checking [12][13]. Some research in the recent past showed that climatic region also plays an important role in the deterioration of bridges [10]. However, the research presented in [10] was limited to one parameter, such as concrete material. The U.S. is geographically a huge country and divided into six climatic regions based on the varying environmental conditions, as shown in Fig. 1. All climatic regions have their own environmental conditions. For example, the Southeast region has dry or hot temperatures, and the Northeast region has very cold temperatures.

The National Bridge Inspection (NBI) database [7] consists of the information of each bridge along with more than 100 parameters. These parameters could be divided into both input and outcome parameters [2]-[5]. This paper attempts to study the effect of the independent parameters on a dependent parameter, including the interaction effects of the independent parameters on the dependent parameter. The three independent parameters considered for this study are: material, design, and region. The only dependent parameter considered was the deck condition rating. The aim of this study is to see how same-age bridges perform when the independent parameters are stand alone and when they interact with each other. Further, this study estimates the mean deck condition ratings while comparing the groups of independent parameters when they are alone or in interaction with each other. A one-way analysis of variance (one-way) ANOVA was required to see if the independent parameter groups' means are the same or not. For this, the Kruskal-Wallis test (since the data is not normal) was used for identifying the effects of the independent parameters on the dependent parameter. If the groups' means are the same, then

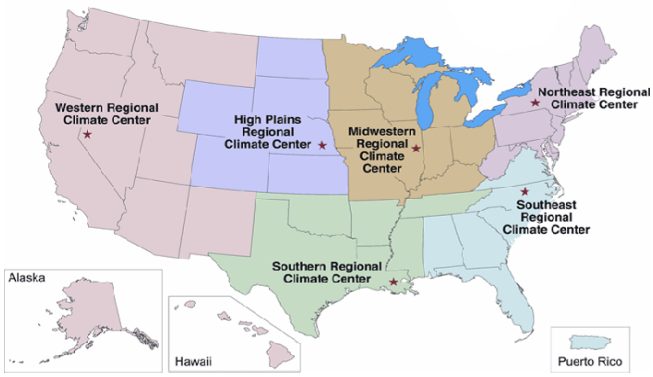

Figure 1. Six climatic regions encompassing the United States.

TABLE 1. DESCRIPTION OF CONDITION RATINGS OF BRIDGES

Condition Rating	Description
9	Excellent condition
8	Very good condition
7	Good condition
6	Satisfactory condition
5	Fair condition
4	Poor condition
3	Serious condition
2	Critical condition
1	Imminent failure condition
0	Failed condition

there is no effect of the independent parameter on the dependent parameters. Otherwise, there is an effect. If the groups' means are not the same, then we need another test to see which group performs better or worse. For this, the Wilcoxon signed-rank test was used for comparing the means of the groups.

II. METHODOLOGY

The dataset used for this study is from the NBI database [7]. The bridges across the USA that were built between the years 1991 and 1993 with an age of 27 years and were not rebuilt (bridges that did not go through any kind of maintenance so far) are selected for this study. The resulting dataset has a total of 1,732 bridges. Three independent parameters and one dependent parameter are considered for this study. The three independent parameters are: material, design, and region, and the dependent parameter is the deck condition rating. The objective is to find whether the independent parameter influences the dependent parameter. This process is done in two steps. First, the Kruskal-Wallis (KW) test is applied individually on each of these independent parameters to see if they influence the dependent parameter. Second, if there is an influence of the independent parameter on the dependent parameter, then the Wilcoxon pairwise comparison test is applied on the independent parameter groups to see the differences within them. The second step is applied only if the independent parameter effects the dependent parameter in the first step. This two-step process is repeated with the interactions of the independent parameters to see their influence on the dependent parameter. The purpose of applying the KW test and the Wilcoxon test on the interactions is to see if the interactions of independent parameters' groups lead to better mean values of the deck condition ratings than the independent parameters' groups mean values alone.

The following hypotheses are tested on all independent parameters and interactions with the deck condition ratings as the independent parameter.

1. The null and alternate hypotheses on materials is given below.

   H_0: The means of deck condition ratings of all material types are equal

   H_a: The means are not equal

2. The null and alternate hypotheses on designs is given below.

   H_0: The means of deck condition ratings of all design types are equal

   H_a: The means are not equal

3. The null and alternate hypotheses on regions is given below.

   H_0: The means of deck condition ratings of all regions are equal

   H_a: The means are not equal

4. The null and alternate hypotheses on material * design * region is given below.

   H_0: The means of deck condition ratings of all material*design*region are equal

   H_a: The means are not equal

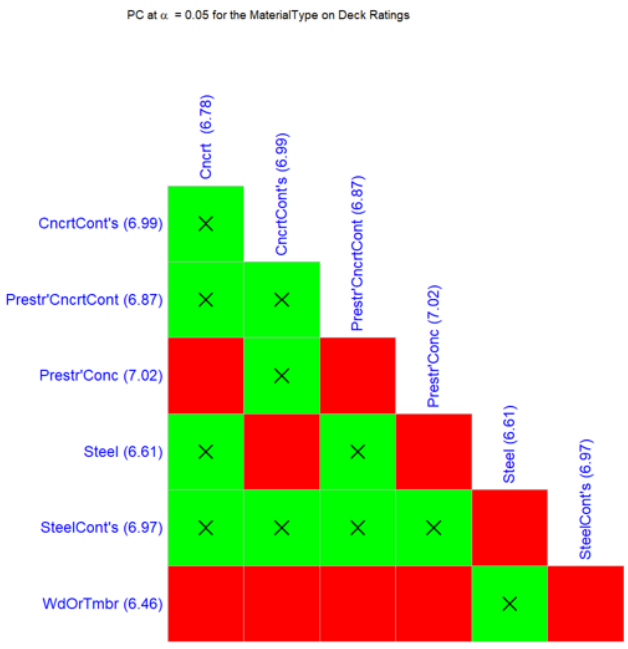

Figure 2. Pairwise comparison among the groups of materials

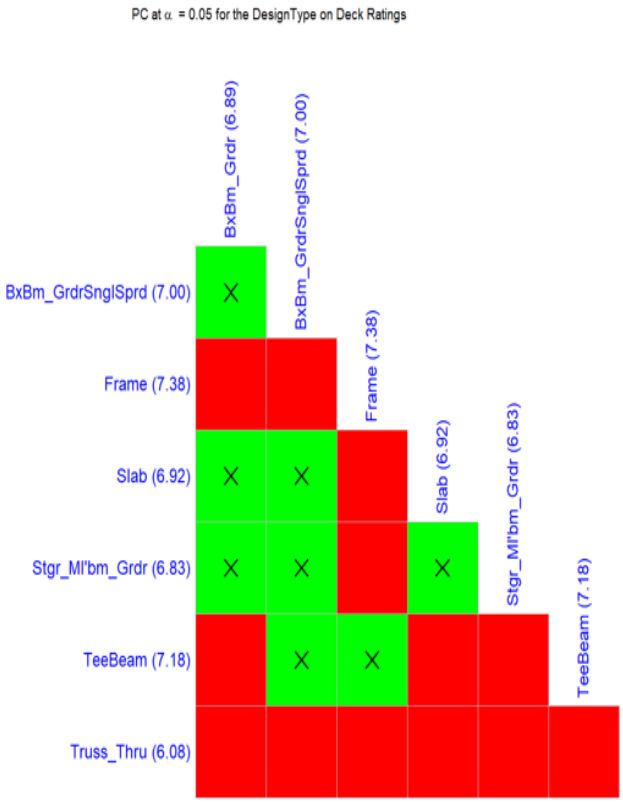

Figure 3. Pairwise comparison among the groups of the designs

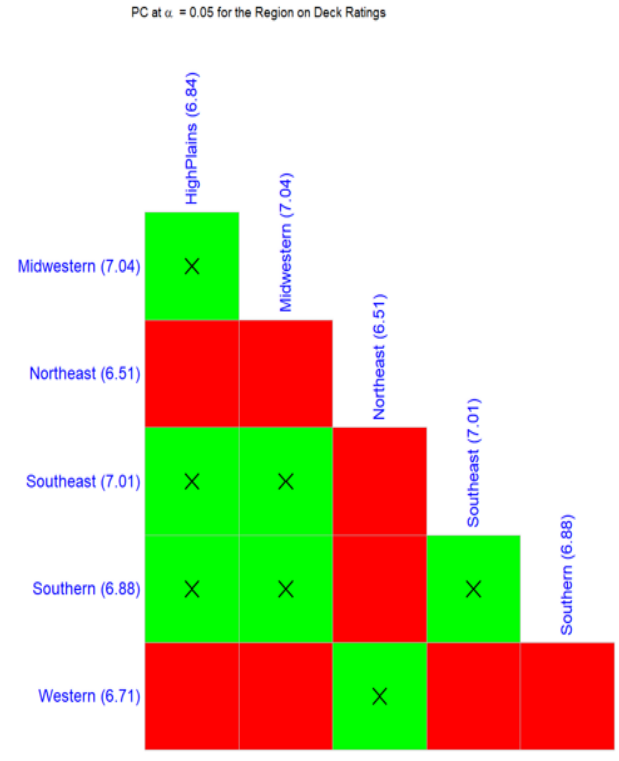

Figure 4. Pairwise comparison among the groups of the region

At the significance level $\alpha = 0.05$, p-values are < 0.001 for all the above hypotheses. Therefore, the null hypotheses are rejected in all the cases. This clearly indicates that the independent parameters influence the dependent parameter.

III. EXPERIMENTAL RESULTS

This section demonstrates different experimental results for all the given hypotheses.

Initially, the KW test was applied on the material types. The mean values of all the material types are not equal. Hence, the null hypotheses were rejected, and it was concluded that the material types influence the deck condition rating. The Wilcoxon test was applied on the groups of material types to see which material is performing better. Fig. 2 shows various material types and their corresponding mean deck condition rating values (shown inside parentheses) after 27 years. From Fig. 2, we see that the prestressed concrete (given as Prestr'Conc) material type is performing the best compared to all other material types with the average deck condition rating value of 7.02. The lowest performing material type is Wood or Timber (given as WdOrTmbr). Fig 2 also shows two types of boxes while comparing the material type groups. Boxes shown with red color (or boxes without 'x' mark in them) are the material groups whose mean deck condition rating values are significantly different. For example, prestressed concrete material's mean deck condition rating value is 7.02, which is significantly different from the concrete material's (given as Cncrt) average deck condition rating value, which is 6.78. Similarly, the green boxes with 'x' symbol in them indicate that there is no significant difference between the mean values of the two material types. For example, concrete continuous (given as CnctrCont's) and concrete materials do not have a significant difference in their mean deck condition ratings. Seven material types were compared in this test. Out of seven material types, Wood or Timber material is significantly different from five other material types, as shown in Fig. 2. This indicates that Wood or Timber material is behaving differently from almost all other materials in terms of performing lower. Further, there is no performance difference among steel continuous material and concrete, concrete continuous, prestressed concrete, and prestressed concrete continuous materials.

The second independent parameter tested is the design type. Seven different designs were tested, as shown in Fig. 3. From Fig. 3, we see that the Frame type is performing the best with the mean deck condition ratings value 7.38, and Truss-Thru design is performing the worst with the value 6.08. Further, Truss-Thru design's performance is significantly different from all other designs.

The third independent parameter is the region. The US is geographically divided into six different regions, as shown in Fig 4. The KW-test was applied first, and the results show

 ISBN: 978-1-61208-964-5

that the region effects the deck condition ratings. The Wilcoxon test results are shown in Fig 4. The Midwestern

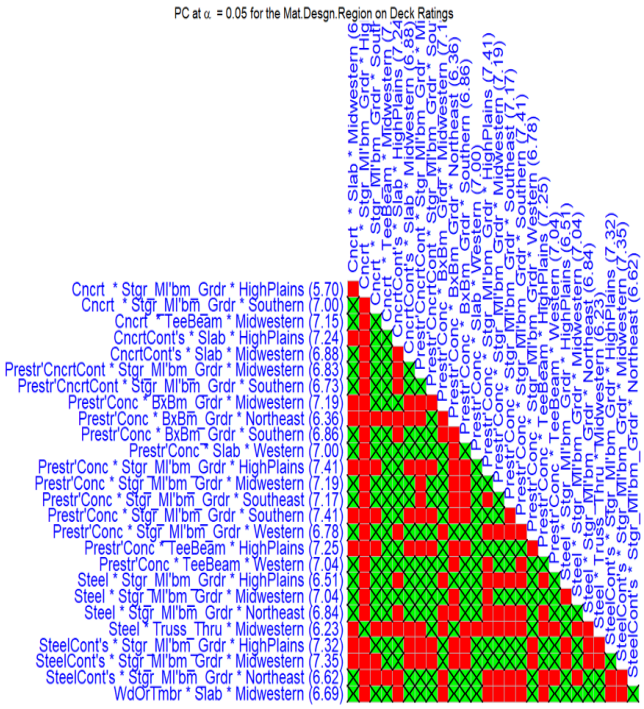

Figure 5. Pairwise comparison among the groups of the interactions of materials, designs, and regions

region is performing the best with the average deck condition rating value of 7.04, and the Northeast region is performing the worst with the value of 6.51. Further, the Northeast region's performance is significantly different from all other regions, except the Western region. The Wilcoxon test result of interactions of all the three independent parameters is shown in Fig 5. The results show that the bridges made of concrete material with stringer multi-beam girder design and that reside in the Highplains region perform the worst with an average deck condition rating value of 5.70, whereas the prestressed concrete bridges with the same design that reside in the same region perform the best with the value of 7.41. Similarly, prestressed concrete bridges with stringer multibeam-girder design that reside in the Southern region are also performing the best after 27 years.

IV. CONCLUSION

This paper analyzed the deterioration of 1,732 US bridges with 27 years of age for each bridge. Three independent parameters, namely, material, design, and region along with their interactions have been considered to see their effect on the dependent outcome condition rating parameter, namely, the deck rating. However, the region parameter was not part of the original NBI database. It was added in this study as the climatic regions play an important role in the deterioration of bridges. As the deck ratings data is not normal, the Kruskal-Wallis test was applied to see if the independent parameters influence the dependent parameter. The Wilcoxon test was applied for the multiple pairwise comparisons between the groups of the independent parameters to see how the groups are performing significantly different. Corrplot package in R language was used to visualize the significant differences among the groups of the independent parameters.

The results show that material, design, and region influence the deck condition ratings of the bridges both individually and in interaction with each other. Further, it is observed that the bridges made of concrete material with stringer multi-beam girder design that reside in the Highplains region perform the worst, whereas the prestressed concrete bridges with the same design that reside in the same region perform the best. Similarly, prestressed concrete bridges with stringer multibeam-girder design that reside in the Southern region are also performing the best after 27 years. Hence, by using the Kruskal-Wallis test followed by the Wilcoxon test, it is concluded that the bridges with concrete material and having stringer multibeam-girder design are not suitable for the Highplains region. The prestressed concrete material is more suitable, as it performs the best.

REFERENCES

[1] ASCE report card on the US bridges, 2017, [Online] https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Bridges-Final.pdf [accessed June 2022].

[2] P. Chetti and H. Ali, "Estimating the Inspection Frequencies of Civil Infrastructures using Correlation Networks and Population Analysis", International Journal on Advances in Intelligent Systems, vol.13 1&2, pp. 151-162, 2020.

[3] P. Chetti and H. Ali, "Impact of Daily Traffic on Various Bridge Decks in Different Climatic Regions", EWSHM, 2022.

[4] P. Chetti and H Ali, "Analyzing the Structural Health of Civil Infrastructures Using Correlation Networks and Population Analysis", In *Proceedings of the Eighth International Conference on Data Analytics, Porto, Portugal*, 2019.

[5] P. Chetti, H. Ali, R. Gandhi, B. Ricks, D. Ghersi, and L. Najjar, "A New Approach for Analyzing Safety and Performance Factors in Civil Infrastructures Using Correlation Networks and Population Analysis". In the Proceedings of the 13th International Workshop on Structural Health Monitoring (IWSHM), 2021.

[6] Z. Hatami, P. Chetti, H. Ali, D. Volkman, A Novel Population Analysis Approach for Analyzing Financial Markets under Crises – 2008 Economic crash and Covid-19 pandemic, 2021.

[7] Federal Highway Administration (FHWA), [Online] https://www.fhwa.dot.gov/bridge/nbi/ascii.cfm, [accessed June 2022].

[8] Federal Highway Administration (FHWA), [Online] https://www.fhwa.dot.gov/bridge/mtguide.pdf, [accessed June 2022]

[9] A. Hatami and G. Morcous, "Developing deterioration models for Nebraska bridges", No. M302, 2011.

[10] A. M. Chyad, O. Abudayyeh, F. Zakhil, and O. Hakimi, "Deterioration rates of concrete bridge decks in several climatic regions", In 2018 IEEE International Conference on Electro/Information Technology (EIT), 2018, pp. 0065-0068.

 ISBN: 978-1-61208-964-5

[11] Z. Hatami, H. Ali, D. Volkman, and P. Chetti, “A New Approach for Analyzing Financial Markets Using Correlation Networks and Population Analysis” In Proceedings of the 24th International Conference on Enterprise Information Systems, vol. 1, 2022, pp. 569-577.

[12] R. Lal, W. Duan, and P. Prabhakar, “Bayesian statistical model checking for continuous stochastic logic”, In 2020 18th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE), 2020, pp. 1-11.

[13] R. Lal, “Scalable safety verification of stochastic hybrid systems” (Doctoral dissertation, Kansas State University), 2021 [Online] https://krex.k-state.edu/dspace/handle/2097/41399 [accessed June 2022].

 ISBN: 978-1-61208-964-5

Optimal Multi-Robot Path Planning for Trash Pick and Drop in Hospitals

Ratan Lal
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, USA
email: rlal@nwmissouri.edu

Rehaman Naguru Abdur
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, USA
email: s544913@nwmissouri.edu

***Abstract*—In this paper, we consider a hospital environment, where each patient's room is very contagious. Hence, we consider the problem of picking trash from each patient's room and dropping it in a big container through multiple robots. Here, we assume that all the robots are small and can pick only one trash bag at a time. Our main objective is to find a plan for the robots that can minimize the total consumed energy (distance, time). Our broad approach is to express the environment in the form of a graph and reduce the problem as an instance of the Multiple Traveling Salesman problem. Then, we encode the reduced problem into the Mixed-Integer Linear Programming (MILP) and solve the encoding using the MILP solver. Next, we perform our approach for hospitals of varied sizes and pick-drop tasks. Our experimental results show that our method is scalable. Finally, we simulate an execution of the optimal plan in the Virtual Robot Experimentation Platform (V-REP) simulator.**

***Index Terms*—multi robots; path planning; trash pick and drop; mixed integer linear programming.**

I. Introduction

Path planning [32] is a well-known problem which is widely used in various applications, such as task spanning [10], evacuation [24], search and rescue [2, 3, 13], coverage [1, 4, 23], precision weeding [31], pesticide spraying [8, 17, 18], transportation in the hospital [21, 28] and medicine delivery [14, 15, 22]. The planning algorithm has been extended from single mobile robot to multiple robots through different techniques, such as cell decomposition approaches [11, 12, 23, 26, 35], potential field approaches [7, 27, 29], and road map approaches [5, 6, 16]. Although different aspects of multi-robot path planning for a hospital have been explored, there is a pressing need of multi-robot for picking dustbin bags from the patient's room and dropping them into a big container to avoid contagious diseases, such as COVID-19.

In this paper, we consider a multi-robot path planning for trash picking and dropping in a hospital environment motivated by a real need. Here, we want to use multiple robots to pick trash bags from the desired patient rooms and drop them into a big container. We assume that robots are small and can collect the garbage from only one dustbin bag at a time. Our objective is to find a plan for the robots that covers all the desired trash bags while minimizing the total distance traveled by all the robots. Hence, this is an optimization problem.

Our broad approach is to reduce the problem into an instance of the Multiple Traveling Salesman (MTS) problem. The MTS problem is a well-known problem where given a graph between cities, multiple salesmen need to visit all the cities exactly once and return to their initial position with the minimum traveling distance. Our approach for reduction is based on the shortest distance graph algorithm. First, we transform a given hospital environment $\mathcal{E}$ into a weighted graph $\mathcal{G}_{\mathcal{E}}$. In the graph $\mathcal{G}_{\mathcal{E}}$, we capture all the valid line segments of the environment where robots could move. Note that the size of $\mathcal{G}_{\mathcal{E}}$ depends on the types of hospitals. Here, we consider three types of hospitals, namely, small, medium and large. Next, we capture each pick and drop task as a pair of one trash bag and one big container. Then, we transform the graph $\mathcal{G}_{\mathcal{E}}$ into another weighted graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ based on a given set of tasks Υ. In the graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$, we capture only robots' initial locations, all the desired trashes and containers' location and create the following edges: (a) edges from robots' location to trashes' location; (b) edges between trashes' location and containers' location. We compute the weight of the edges by applying the shortest distance graph algorithm on $\mathcal{G}_{\mathcal{E}}$. Note that the size of the graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ depends on the number of tasks. Also, for the MTS problem, each city (vertex) must be visited exactly once by one of the salesmen. However, in the graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$, a robot may need to visit the same container more than once. So, we need to ensure that each vertex corresponding to a big container is visited exactly once by one of the robots. Therefore, we transform $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ into another graph $\mathcal{G}_M$, where if a big container is common among multiple tasks, we create a copy of the vertex corresponding to the container and add the respective edges to the copied vertex. Next, we introduce a dummy vertex to $\mathcal{G}_M$ for the robots to return to their initial location. Finally, we encode the reduced problem into the Mixed-Integer Linear Programming (MILP) similar to the one given in [19].

We have implemented our method in the Python toolbox for finding the optimal plan for the robots, where we have used the NetworkX tool for the graph construction and the GNU Linear Programming Kit (GLPK) to solve the MILP encoding for an instance of the MTS problem. Next, we extract a real optimal plan for the robots from a solution returned by the GLPK solver and by applying the shortest path graph

 ISBN: 978-1-61208-964-5

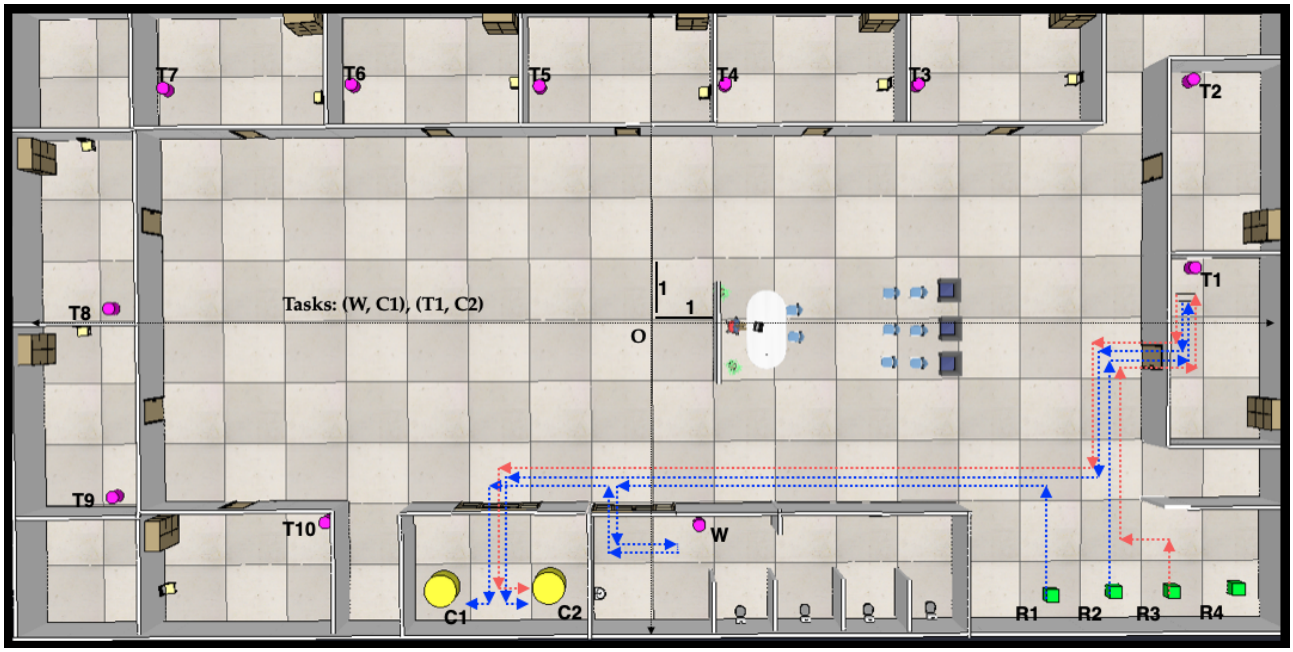

Figure 1: Hospital Environment

algorithm on $\mathcal{G}_{\mathcal{E}}$. Finally, we deploy the plan in the Virtual Robot Experimentation Platform (V-REP) simulator [25].

Our main contributions of the paper are given below.

- We have presented a task-based graph reduction for reducing the pick and drop problem into an instance of the MTS problem.
- Since the task-based graph depends on the tasks, the size of encoding is less and the GLPK solver returns faster.
- We have simulated the plan in the V-REP simulator.

Remark 1: We use the collision avoidance protocol integrated with the V-REP simulator while executing the plan.

II. Related Work

Multi-robot path planning has been explored for hospitals. Different techniques, such as sub-dimensional expansion [30], Integer Linear Programming [33], Artificial Potential Field (AFP) [29], and Enhanced Genetic Algorithm (EGA) [20] have been explored.

In this paper, we explore selected trash pick and drop tasks for the hospitals. Some of the research works in the area of trash collection have been explored. A prototype for the garbage collection based on Convolutional Neural Networks (CNNs) has been studied [9], where a CNN is integrated with a robot to detect and classify different types of garbage. However, our work is focused on finding an optimal path for the robots to pick the trash from patient's room and drop into a big container. Although the pick and drop problem has been studied in urban settings [34], there are limited works in this direction. Hence, further investigation is needed to avoid contagious diseases.

III. Motivation

In this section, we demonstrate the problem by considering a hospital environment shown in Figure 1, where there are eleven small dustbin containers, namely, T1, T2, ..., T10 located in each patient's room and W situated in the washroom represented by magenta color; two big trash containers, namely, C1 and C2 represented by yellow color; and four robots, namely, R1, R2, R3, and R4 represented by green color. Assume that each container has a small dust bag, and each (small) robot can pick only one bag at a time. Note that all the bags do not always have dust. Hence, we consider the selected pick and drop problem, where we want the robots to pick the bags from only required dustbins and drop them into the big containers. For example, in Figure 1, we want the robots to pick the bag from washroom W and drop it into C1 and pick the bag from patient dustbin T1 and drop it into C2. For the problem, each pair of pick and drop points is considered a task. For example, (W, C1) is a task.

Here, given a list of tasks (W, C1), (T1, C2), we want to find a path for each robot to solve the selected pick and drop problem. Note that multiple paths may exist for the robots. For example, for the tasks, the following two paths solve the problem, namely, (a) blue path from R1 and R2; (b) blue path from R1 and red path from R3. Hence, our objective is to find a path for each robot such that the total distance traveled by all the robots can be minimized.

IV. Preliminaries

Notations: Let $\mathbb{R}_{\geq 0}$, $\mathbb{R}$ and $\mathbb{N}$ denote the set of positive real numbers, the set of real numbers, and the set of natural numbers, respectively. We use $[n]$ to denote the set

 ISBN: 978-1-61208-964-5

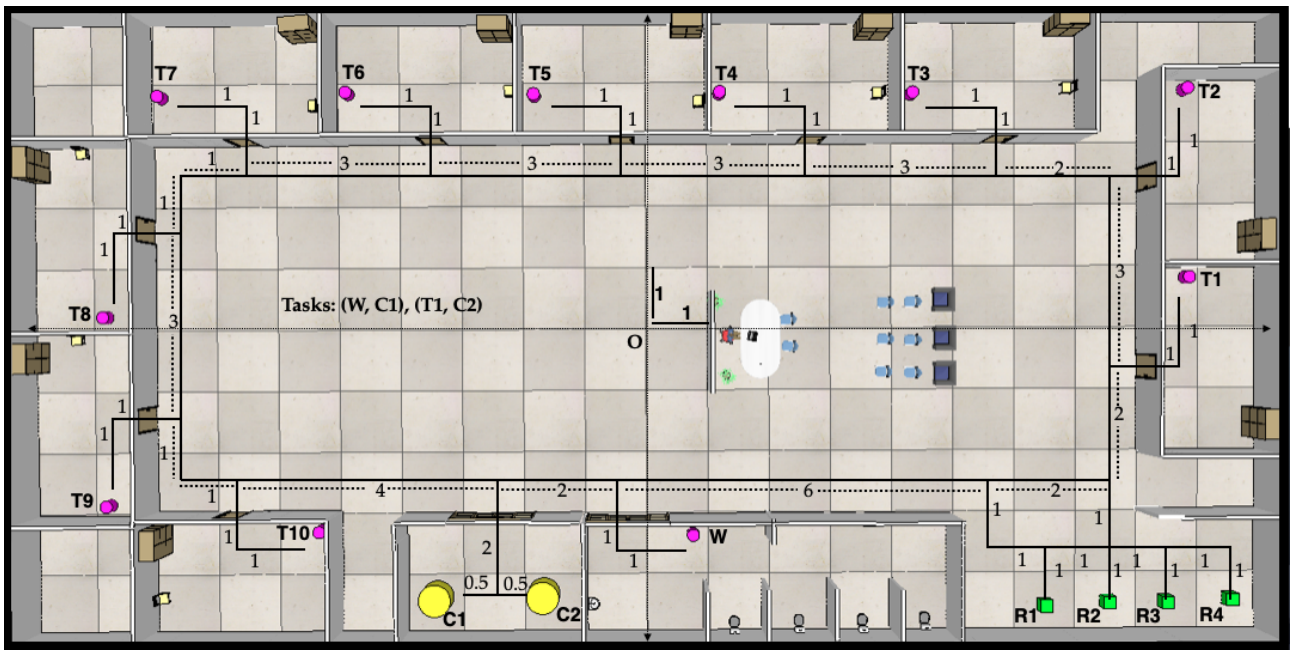

Figure 2: Hospital Environment ($\mathcal{E}$)

$\{1, 2, \ldots, n\}$. Given a set $\mathcal{S}$, we use $|\mathcal{S}|$ to denote the number of elements in $\mathcal{S}$.

Euclidean and Manhattan Distance: Given two $2-$dimensional points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, the Euclidean distance between p_1 and p_2 denoted by $d_E(p_1, p_2)$, is defined as follows:

$$d_E(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

The Manhattan distance between p_1 and p_2 denoted by $d_M(p_1, p_2)$, is defined as follows:

$$d_M(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|.$$

Weighted Graph: A weighted graph is defined as a tuple $\mathcal{G} = (V_0, V, E, W)$ where

- V is a set of vertices;
- $V_0 \subseteq V$ is a set of initial vertices;
- $E \subseteq V \times V$ is a set of edges;
- $W : E \to \mathbb{R}_{\geq 0}$ is a weight function that captures the length for each edge.

A path denoted by σ for a given weighted graph $\mathcal{G}$ is a sequence of vertices $v_0, v_1, v_2, \ldots, v_n \in V$ such that $(v_{i-1}, v_i) \in E$ for $i \in [n]$. We use $cost(\sigma)$ to denote the cost of the path σ, that is,

$$cost(\sigma) = \sum_{i=1}^{n} W(v_{i-1}, v_i).$$

Complete Paths: Given a weighted graph $\mathcal{G}$, a complete path is a set of paths $\varrho = \{\sigma_i\}_{i=1}^{n}$ satisfying the following conditions:

- $n = |V_0|$;
- for each path $\sigma_i = v_0^i, v_1^i, \ldots, v_m^i$, $v_0^i \in V_0$.

Succinctly, we use $cost(\varrho)$ to denote the cost of the complete path ϱ, that is,

$$\sum_{\sigma \in \varrho} cost(\sigma).$$

V. Problem Formulation

In this section, we formally describe the pick and drop problem. First, we define an environment for the hospitals given as below.

Definition 1: [Environment] An environment is a tuple $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, where

- $\mathcal{R} \subseteq \mathbb{R}^2$ is a set of robots' initial location;
- $\mathcal{T} \subseteq \mathbb{R}^2$ is a set of dustbins' location;
- $\mathcal{C} \subseteq \mathbb{R}^2$ is a set of big trash containers' location;
- $Edges \subseteq \mathcal{P} \times \mathcal{P}$ where $\mathcal{P} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{C}$, that captures a set of edges on which robots could move;
- $\mathcal{D} : Edges \to \mathbb{R}_{\geq 0}$ is a distance function that determines the length for each edge.

Example 1: Consider the hospital environment shown in Figure 2. It can be represented as a tuple $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, where

- $\mathcal{R} = \{(6.5, -4.5), (7.5, -4.5), (8.5, -4.5), (9.5, -4.5)\}$;
- $\mathcal{T} = \{(8.5, 0.5), (8.5, 3.5), (4.5, 3.5), (1.5, 3.5), (-1.5, 3.5), (-4.5, 3.5), (-7.5, 3.5), (-8.5, 0.5), (-8.5, -2.5), (-5.5, -3.5), (0.5, -3.5)\}$;
- $\mathcal{C} = \{(-3.5, -4.5), (-1.5, -3.5)\}$;
- $Edges$ are all the pairs of end points of each solid black line segment shown in Figure 2;
- For each line segment $e \in Edges$, $\mathcal{D}(e)$ is shown in Figure 2. For example, $\mathcal{D}((6.5, -4.5), (6.5, -3.5)) = 1$.

Next, we define a pick and drop task given as below:

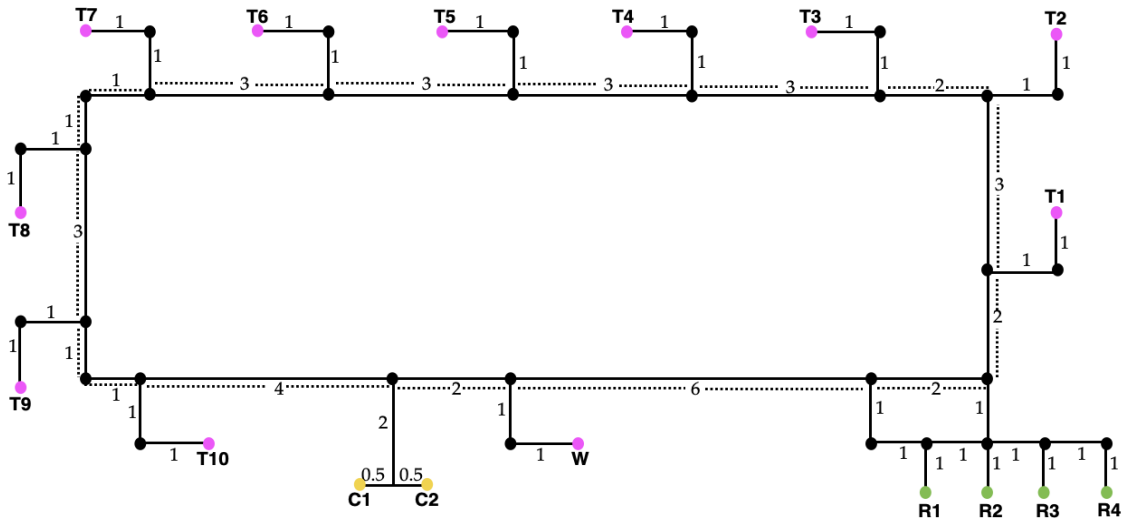

Figure 3: Graph ($\mathcal{G}_\mathcal{E}$)

Definition 2: [Task] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, a task is defined as a pair $\tau = (t, c)$, where

- $t \in \mathcal{T}$ is a location of a dustbin;
- $c \in \mathcal{C}$ is a location of a big container.

Next, we define a plan for a robot to accomplish a set of tasks in an environment given as below.

Definition 3: [Plan] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, an initial robot's location $r \in \mathcal{R}$, a plan for the robot is a sequence of locations $\rho = l_0, l_1, l_2, \ldots, l_n$ satisfying the following conditions:

- $l_0 = r$ and $(l_{i-1}, l_i) \in Edges$ for $i \in [n]$;
- for each $l_i = t_k$ for some $i \in [n]$ and $k \in [p]$, $\nexists\ j \in [n]$, satisfying $j \neq i$ and $l_j = t_k$;
- for each $l_i = t_k$ for some $i \in [n]$ and $k \in [p]$, $\exists\ j$, $j > i$ satisfying $l_j = c_k$ and $\nexists\ m$, $i < m < j$ such that $l_m = t_{k'}$ for some $k' \in [p]$.

The cost of the plan ρ can be computed by the following formula:

$$cost(\rho) = \sum_{i=1}^{n} \mathcal{D}((l_{i-1}, l_i)).$$

Definition 4: [Complete Plan] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, a complete plan is defined as a sequence of plan $\Gamma = \{\rho_i\}_{i=1}^{|\mathcal{R}|}$ such that for each task (t_k, c_k), there exists a plan $\rho = l_0, l_1, \ldots, l_n \in \Gamma$ starting from some robot's initial location $r \in \mathcal{R}$ such that $l_i = t_k$ for some $i \in [n]$.
The cost of the complete plan $\Gamma = \{\rho_i\}_{i=1}^{|\mathcal{R}|}$ can be computed by the following expression:

$$\sum_{i=1}^{|\mathcal{R}|} cost(\rho_i).$$

Note that multiple complete plans are possible. Hence, our aim is to find a complete plan whose cost is minimum. We formally define the optimization problem given as below.

Problem 1: [Problem] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$ and a set of tasks $\Upsilon = \{\tau_k\}_{k=1}^{p}$, find a complete plan $\Gamma = \{\rho_i\}_{i=1}^{|\mathcal{R}|}$ such that

$$\sum_{i=1}^{|\mathcal{R}|} cost(\rho_i) \text{ is minimum.}$$

VI. Our Approach

In this section, we present a procedure to reduce the problem as an instance of the MTS problem. The reduction procedure consists of three steps. First, we express a given environment $\mathcal{E}$ as a weighted graph $\mathcal{G}_\mathcal{E}$ that captures all possible robots' movements, where the set of initial vertices will be all robots' initial location. Second, given a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, we reduce $\mathcal{G}_\mathcal{E}$ into another graph $\mathcal{G}_\mathcal{E}^\Upsilon$ that consists of only those vertices, which are either t_i, c_j, or robots' initial location for $i, j \in [p]$. The edges for $\mathcal{G}_\mathcal{E}^\Upsilon$ are constructed as follows: (a) for each vertex t_i, we create edges from t_i to all c_j's; (b) for each vertex c_j, we create edges from c_j to all t_i's; (c) for each robots' initial location r, we create edges from r to all c_j's. Then, we compute the weight for each edge by computing the shortest distance between the end points of the edge in the graph $\mathcal{G}_\mathcal{E}$. Finally, we reduce the problem as an instance of the MTS problem by appropriate modification to $\mathcal{G}_\mathcal{E}^\Upsilon$. Next, we provide the details about each step.

A. Graph Representation for Environments

In this section, we provide a formal construction of the weighted graph capturing all the line segments of an environment in which robots could move.

Definition 5: [Construction of $\mathcal{G}_\mathcal{E}$] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, the construction of $\mathcal{G}_\mathcal{E} = (V_0, V, E, W)$ is given below:

- $V_0 = \mathcal{R}$;
- $V = \bigcup_{(u,v)\in Edges} \{u, v\}$;
- $E = Edges$;
- $W(e) = \mathcal{D}(e)$.

Next, we demonstrate the construction of $\mathcal{G}_\mathcal{E}$ with an example.

Example 2: Consider the hospital environment $\mathcal{E}$ shown in Figure 2. The constructed graph $\mathcal{G}_\mathcal{E}$ corresponding to the environment $\mathcal{E}$ is shown in Figure 3.

B. Tasks based Reduction of $\mathcal{G}_\mathcal{E}$

In this section, we reduce the graph $\mathcal{G}_\mathcal{E}$ based on a given set of tasks. Given a set of tasks Υ, and a graph $\mathcal{G}_\mathcal{E}$ corresponding to an environment $\mathcal{E}$, we construct a task-based graph $\mathcal{G}_\mathcal{E}^\Upsilon$ from the graph $\mathcal{G}_\mathcal{E}$ that consists of only those vertices related to robots' initial location, trashes' location, containers' location. The formal construction of the graph $\mathcal{G}_\mathcal{E}^\Upsilon$ is given as below.

Definition 6: [Construction of $\mathcal{G}_\mathcal{E}^\Upsilon$] Given a graph $\mathcal{G}_\mathcal{E} = (V_0, V, E, W)$ corresponding to an environment $\mathcal{E}$ and a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^p$, the construction of $\mathcal{G}_\mathcal{E}^\Upsilon = (V_0', V', E', W')$ is given as below:

- $V_0' = V_0$;
- $V = V_0' \cup T \cup C$, where $T = \cup \bigcup_{(t,c)\in\Upsilon} \{t\}$ and $C = \cup \bigcup_{(t,c)\in\Upsilon} \{c\}$;
- $E = E_1 \cup E_2 \cup E_3$ where
  - $E_1 = \{(v, t) \mid v \in V_0, t \in T\}$ captures edges from robots' initial location to the dustbins associated with the tasks;
  - $E_2 = \{(t, c) \mid t \in T,\ c \in C\}$ captures edges between each dustbin and container associated with the tasks;
  - $E_3 = \{(c, t) \mid c \in C,\ t \in T,\}$ captures edges between each container and dustbin associated with the tasks;
- for each edge $(u, v) \in E'$, $W'(u, v)$ captures the shortest distance between u and v in $\mathcal{G}_\mathcal{E}$.

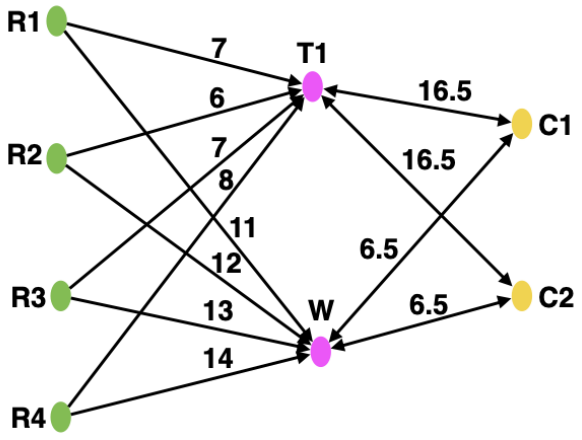

Figure 4: Task based Graph ($\mathcal{G}_\mathcal{E}^\Upsilon$)

Next, we illustrate the construction of the task-based graph with an example.

Example 3: Consider the graph $\mathcal{G}_\mathcal{E}$ shown in Figure 3 corresponding to the hospital environment $\mathcal{E}$ shown in Figure 2 and a set of tasks $\Upsilon = \{(W, C_1), (T1, C2)\}$. The constructed graph $\mathcal{G}_\mathcal{E}^\Upsilon$ corresponding to the graph $\mathcal{G}_\mathcal{E}$ and Υ is shown in Figure 4.

Note that for the MTS problem, each vertex has to be visited exactly once by the salesmen. However, in the pick and drop problem, a vertex corresponding to the big container may need to be visited more than once. Hence, we convert the graph $\mathcal{G}_\mathcal{E}^\Upsilon$ into another graph $\mathcal{G}_M$ for reducing the pick and drop problem as an instance of the MTS problem.

C. Construction of $\mathcal{G}_M$ from $\mathcal{G}_\mathcal{E}^\Upsilon$

In this section, we present the construction of the graph $\mathcal{G}_M$ from the graph $\mathcal{G}_\mathcal{E}^\Upsilon$. First, for each vertex in the graph $\mathcal{G}_\mathcal{E}^\Upsilon$ corresponding to a big container c associated with the task, if the maximum number of tasks having c is k, then we create $k-1$ copies of vertex c. Then, we add all the incoming and outgoing edges associated with the container c to all the copied vertices. Next, we introduce a dummy vertex d for all the robots to have a common initial point. Finally, we add edges between d and all the robots' location and from big containers to d with distance 0. The formal construction for the graph $\mathcal{G}_M$ is given below.

Definition 7: [Construction of $\mathcal{G}_M$] Given a task based graph $\mathcal{G}_\mathcal{E}^\Upsilon = (V_0', V', E', W')$, the construction of $\mathcal{G}_M = (V_0^m, V^m, E^m, W^m)$ is given as below. Let $C = \cup \bigcup_{(t,c)\in\Upsilon} \{c\}$.

- $V_0^m = \{d\}$;
- $V^m = V_0^m \cup V' \cup V_c$, where $V_c = \bigcup_{c\in C} \{c_i \mid 1 \le i < k,$ k is the number of tasks in which c appears$\}$;
- $E^m = E' \cup \{(u, v) \mid u \in V_0^m,\ v \in V_0'\} \cup \{(u, v) \mid u \in V_0' \cup V_c \cup C,\ v \in V_0^m\}$;
- $W^m(e) = W(e)$ if $e \in E'$ otherwise $W^m(e) = 0$.

Example 4: Consider the task based graph $\mathcal{G}_\mathcal{E}^\Upsilon$ shown in Figure 4. The constructed graph $\mathcal{G}_M$ corresponding to $\mathcal{G}_\mathcal{E}^\Upsilon$ is shown in Figure 8.

Now, the graph $\mathcal{G}_M$ can be used for an instance of the MTS problem, where vertices corresponding to the robots can be considered as salesmen. The dummy vertex can be treated as a source and target vertex for each robot. Finally, we use the graph $\mathcal{G}_M$ to encode the pick and drop problem into the Mixed Integer Linear Programming (MILP) and extract the optimal plan for the robots.

VII. Experimental Analysis

In this section, we present the analysis of our method for different number of tasks in three kinds of hospitals, namely, small, medium, and large shown in Figures 5, 6, and 7, respectively. We have implemented our method in the Python toolbox, where we have used the NetworkX tool for the graph construction; the GLPK solver for solving the optimization problem. Finally, we use the V-REP [25] simulator to simulate the optimal plan.

In Table I, *#Robots* denotes the number of robots for small, medium, and large hospitals. *#Tasks* represents the number

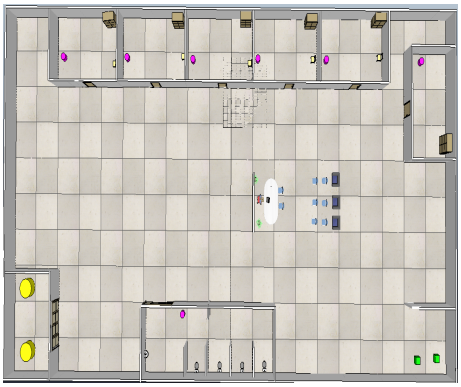

Figure 5: Small Hospital

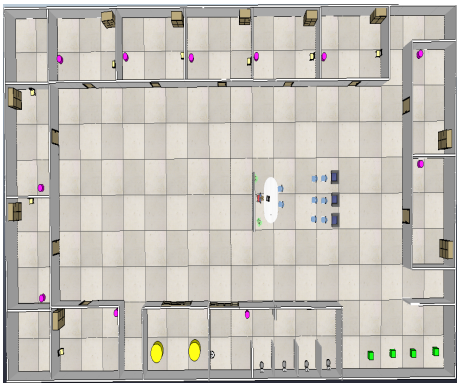

Figure 6: Medium Hospital

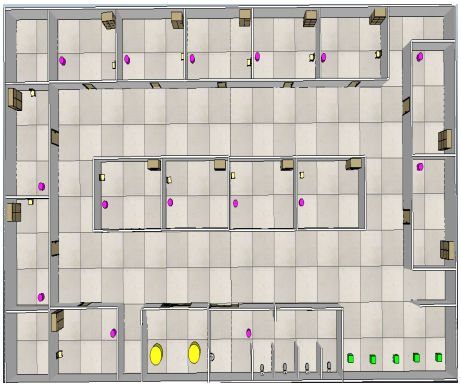

Figure 7: Large Hospital

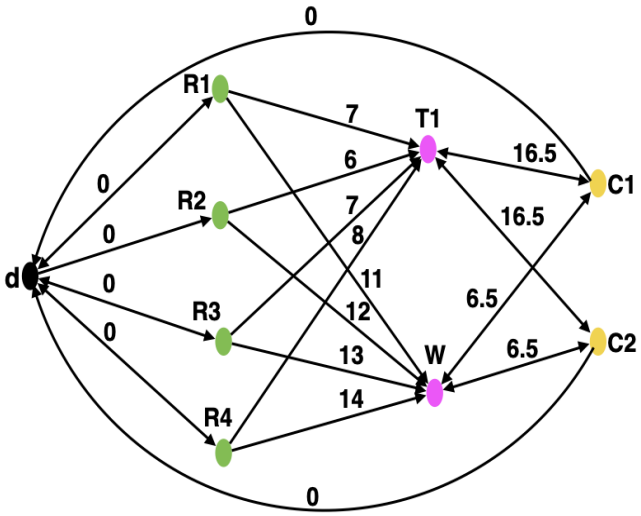

Figure 8: Graph ($\mathcal{G}_M$)

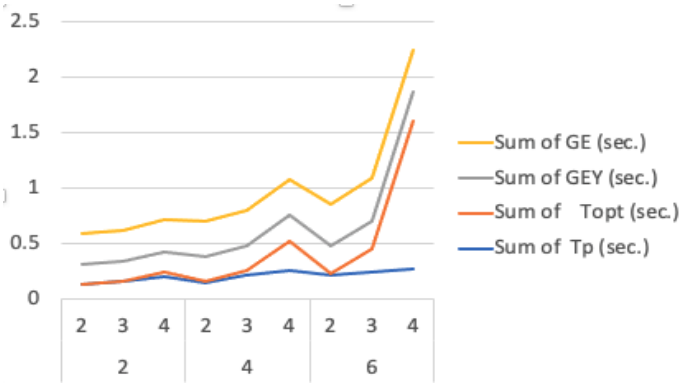

Figure 9: Computational Analysis

of tasks. $T_{\mathcal{G}_\mathcal{E}}$, $T_{\mathcal{G}_\mathcal{E}^\Upsilon}$, T_{opt}, and T_p are the times taken for constructing the graph corresponding to a given environment, a task-based graph, and finding the optimal solution by the GLPK solver, and extracting an optimal path for the robots, respectively. *cost* denotes the optimal cost, that is, the total distance traveled by all the robots. The experimental results are presented in Table I.

TABLE I: COMPUTATIONAL ANALYSIS

#Robots	*#Tasks*	$T_{\mathcal{G}_\mathcal{E}}$(sec.)	$T_{\mathcal{G}_\mathcal{E}^\Upsilon}$(sec.)	T_{opt}(sec.)	T_p(sec.)	*cost*
2 (small)	2	0.28	0.18	0.001	0.13	72
	3	0.28	0.18	0.005	0.16	113.5
	4	0.28	0.19	0.041	0.20	136
4 (med.)	2	0.32	0.23	0.004	0.15	62
	3	0.32	0.23	0.031	0.22	101.5
	4	0.32	0.24	0.255	0.26	142
6 (large)	2	0.38	0.25	0.019	0.21	60
	3	0.38	0.25	0.216	0.24	98.5
	4	0.38	0.26	1.337	0.27	138

In Table I, we have observed that the time taken to construct the graph $\mathcal{G}_E$ gradually grows for a fixed number of tasks when we increase the size of hospitals, as can be seen in the first row of small, medium, and large hospital, respectively. A similar observation can be seen for the task-based graph. The time taken by the GLPK solver slowly grows when we increase the number of tasks for a fixed environment, as can be seen in the first three rows for the small hospital. A similar observation can be seen for the path extraction. Also, *cost* increases when we increase the number of tasks for the same environment. We have plotted the data in Figure 9 corresponding to the data presented in Table I, where all the observations mentioned above can be clearly seen. Overall, our method is scalable.

VIII. CONCLUSION

In this paper, we have investigated optimal multi-robot path planning for a selected pick and drop problem. We have reduced the problem as an instance of the MTS problem by representing a hospital environment as a weighted graph and transforming the weighted graph into a task-based graph with the help of the shortest distance graph algorithm. We have used the GLPK solver to solve the optimization problem. Finally, we have performed the experiments for different types of hospitals, namely, small, medium, and large. In the future, we will investigate optimal multi-robot path planning for multi-objective tasks.

 ISBN: 978-1-61208-964-5

IX. Acknowledgments

This work is partially supported by Northwest Missouri State University.

References

[1] A. Barrientos, et al., "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," Journal of Field Robotics vol. 28, pp. 667–689, 2011.

[2] J. L. Baxter, E. Burke, J. M. Garibaldi, and M. Norman, "Multi-robot search and rescue: A potential field based approach," Autonomous robots and agents, pp. 9–16, 2007.

[3] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello, "Mixed-initiative planning and execution for multiple drones in search and rescue missions," International Conference on Automated Planning and Scheduling, vol. 25, pp. 315–323, 2015.

[4] H. Choset, "Coverage for robotics–a survey of recent results," Annals of mathematics and artificial intelligence, vol. 31, pp. 113–126, 2001.

[5] C. M. Clark, S. M. Rock, and J.-C. Latombe, "Motion planning for multiple mobile robots using dynamic networks," IEEE International Conference on Robotics and Automation, vol. 3, pp. 4222–4227, 2003.

[6] A. Cowley, C. J. Taylor, and B. Southall, "Rapid multi-robot exploration with topometric maps," IEEE International Conference on Robotics and Automation, pp. 1044–1049, 2011.

[7] R. Gayle, W. Moss, M. C. Lin, and D. Manocha, "Multi-robot coordination using generalized social potential fields," IEEE International Conference on Robotics and Automation, pp. 106–113, 2009.

[8] M. Gonzalez-de-Soto, L. Emmi, M. Perez-Ruiz, J. Aguera, and P. Gonzalez-de-Santos, "Autonomous systems for precise spraying–Evaluation of a robotised patch sprayer," Biosystems engineering, vol. 146, pp. 165–182, 2016.

[9] S. Gupta, H. Kruthik, C. Hegde, S. Agrawal, and S. B. Prashanth, "GarBot: Garbage Collecting and Segregating Robot," Journal of Physics: Conference Series, vol. 1950, pp. 012023, 2021.

[10] D. Halperin, J.-C. Latombe, and R. H. Wilson, "A general framework for assembly planning: The motion space approach," vol. 26, pp. 577–601, 2000.

[11] N. Hazon and G. A. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," IEEE international conference on robotics and automation, pp. 735–741, 2005.

[12] N. Hazon, F. Mieli, and G. A. Kaminka, "Towards robust on-line multi-robot coverage," IEEE International Conference on Robotics and Automation, pp. 1710-1725, 2006.

[13] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," International Conference on Advanced Robotics, pp. 193–200, 1997.

[14] A. Joy, R. Varghese, A. Varghese, A. M. Sajeev, J. Raveendran, A. Thomas, and K. Saran, "Medrobo medicine delivering and patient parameter monitoring robot," International Conference on Advanced Computing and Communication Systems, vol. 1, pp. 1808–1812, 2021.

[15] S. Kavirayani, D. S. Uddandapu, A. Papasani, and T. V. Krishna, "Robot for delivery of medicines to patients using artificial intelligence in health care," IEEE Bangalore Humanitarian Technology Conference, pp. 1–4, 2020.

[16] S. Kumar and S. Chakravorty, "Multi-agent generalized probabilistic RoadMaps: MAGPRM," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3747–3753, 2012.

[17] R. Lal and P. Prabhakar, "Time-optimal multi-quadrotor trajectory planning for pesticide spraying," IEEE International Conference on Robotics and Automation, pp. 7965–7971, 2021.

[18] R. Lal, A. Sharda, and P. Prabhakar, "Optimal multi-robot path planning for pesticide spraying in agricultural fields," IEEE 56th Annual Conference on Decision and Control, pp. 5815–5820, 2017.

[19] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," Journal of the ACM, vol. 7, pp. 326–329, 1960.

[20] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," Expert Systems with Applications, vol. 115, pp. 106–120, 2019.

[21] A. G. Ozkil, Z. Fan, S. Dawids, H. Aanes, J. K. Kristensen, and K. H. Christensen, "Service robots for hospitals: A case study of transportation tasks in a hospital," IEEE international conference on automation and logistics, pp. 289–294, 2009.

[22] A. Patel, P. Sharma, and P. Randhawa, "MedBuddy: The Medicine Delivery Robot," International Conference on Reliability, Infocom Technologies and Optimization, pp. 1–4, 2021.

[23] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," IEEE International Conference on Robotics and Automation, vol. 4, pp. 3462–3468, 2004.

[24] S. Rodriguez and N. M. Amato, "Behavior-based evacuation planning," IEEE International Conference on Robotics and Automation, pp. 350–355, 2010.

[25] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1321–1426, 2013.

[26] T. Siméon, S. Leroy, and J.-P. Lauumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," IEEE transactions on robotics and automation, vol. 18, pp. 42–49, 2002.

[27] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," IEEE International Conference on Robotics and Automation, vol. 2, pp. 1217–1222, 2002.

[28] M. Takahashi, T. Suzuki, H. Shitamoto, T. Moriguchi, and K. Yoshida, "Developing a mobile robot for transport applications in the hospital domain," Robotics and Autonomous Systems, vol. 58, pp. 889–899, 2010.

[29] H. G. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," IEEE International Conference on Robotics and Automation, pp. 4132–4137, 2005.

[30] G. Wagner and H. Choset, "M*: A complete multirobot path planning algorithm with performance bounds," IEEE/RSJ international conference on intelligent robots and systems, pp. 3260–3267, 2011.

[31] M. Weis, et al., "Precision farming for weed management: techniques," Gesunde Pflanzen, vol. 60, pp. 171–181, 2008.

[32] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi- robot coordination," International Journal of Advanced Robotic Systems, vol. 10, pp. 399, 2013.

[33] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," IEEE Transactions on Robotics, vol. 32, pp. 1163–1177, 2006.

[34] S. Yu, J. Puchinger, and S. Sun, "Van-based robot hybrid pickup and delivery routing problem," European Journal of Operational Research, vol. 298, pp. 894–914, 2022.

[35] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3852–3857, 2005.

 ISBN: 978-1-61208-964-5