A Generic Approach for Allocating Movement Permits During/Outside Curfew Period during COVID-19

Yaser Chaaban
Department of Computer and Information Sciences
Faculty of Sciences and Arts, Taibah University, AlUla Branch
Al-Madinah, Saudi Arabia

***Abstract*—During the coronavirus (COVID-19) pandemic, different exciting concepts around solutions, technical components, smartphone applications, and novel wireless services emerged, which were needed in order to adjust to the new lifestyle standards. In this context, social distancing was imposed to prevent or decrease further transmission of COVID-19. In other words, research results have shown that slowing the spread of COVID-19 is the most efficient way to save people's lives and relieve the burden on health-care systems. This social distancing can be tracked using cell phone movement/data. This paper presents a new approach/algorithm for allocating and optimizing/adapting the movement permits during/outside curfew periods inside workplaces, buildings, companies and institutions. This approach is an effective tool to reduce the spread of COVID-19 by promoting health safety during the pandemic, especially in places where social distancing can be difficult. Consequently, this paper presents a technological solution to automate the process of distributing movement permits in workplaces. The research results showed that the proposed strategy of social distancing inside buildings is effective enough to flatten the curve. Furthermore, health authorities do not have to mandate stay-at-home orders to slow the spread of COVID-19. Consequently, this paper introduces a solution for the resource sharing problem (resource allocation problem), where multiple agents (people or robots) of a system move reliably in their environment. The biggest concern of these agents is to avoid collisions (infections). As a result, the experiments performed in this paper showed the high performance of the designed algorithm complying with COVID-19 social distancing regulations.**

Keywords—COVID-19 pandemic; social distancing; resource allocation problem; movement permits

I. Introduction

Nowadays, scientific research is announcing the opening of the application period around the world for the institutional funding initiative of strategic studies. Many universities or organizations have new priorities and research topics so proposals must align with the coronavirus (COVID-19) pandemic topic. In all instances, strategic research was approved by ministries for research and innovation, ministries of high education, and organizations such as the World Health Organization (WHO) [1]. Simultaneously, the future of distance education and E-Learning became more interesting and necessary during the COVID-19 pandemic. Furthermore, digital technologies in health services, new techniques in infectious diseases management, comprehensive approaches for COVID-19 prevention, screening and management, and state-of-the-art diagnostics of COVID-19 became critical enablers for our society. All these topics represent current challenges in population health. In this context, many brands, organizations, and governments are trying to find digital platforms such as special websites, mobile-friendly websites, virtual reality, and smartphone applications to manage particularly the new lifestyle standard of COVID-19 [2].

Another idea related to the concept of "flatten the curve" [3] is the social distancing strategy. Social distancing also called "physical distancing", can be tracked using cell phone data or movement. This data is important to identify how many times was a certain region, area, or intersection walked on.

In essence, this paper focuses on designing a control algorithm for providing human agents inside workplaces with planned movement permits. This will enable agents to move safely in their shared environment during the COVID-19 pandemic. Accordingly, avoiding infections (collisions) and reducing journey times are needed by implementing a social distancing strategy.

This research paper is organized as follows: Section 2 provides a survey of related work concerning social distancing policies implemented in response to COVID-19. In Section 3, the roles of software applications developed during COVID-19 are discussed. Section 4 raises the question: Are technology solutions ready for life after COVID-19? After that the objectives of the proposed approach/algorithm will be briefly discussed in Section 5. Section 6 describes the scenario of this paper, where agents trying to move in a shared environment inside workplaces during COVID-19. Moreover, the new approach will be presented in Section 7, containing trajectory planning and the developed algorithm. Section 8 presents the evaluation of the proposed algorithm using two metrics. After that, Section 9 concludes this paper. Finally, the future work is explicated in Section 10.

II. Related Works

As mentioned above, various projects and research topics are dealing with the COVID-19 pandemic around the world. In this section, it is concentrated on related works that focus on implementing a social distancing policy due to the COVID-19 pandemic. It is also worth mentioning here that some concepts, algorithms, or policies related to social distancing are relevant to this work. Another aspect that is of interest is the allocation of movement permits during COVID-19, particularly in those places where ensuring social distancing is difficult.

In honeybee systems, honeybees cannot practice any social distancing. However, they work together in very close spaces as a healthy community. Consequently, honeybees work together to fight diseases, whereas humans can learn how they can tackle the COVID-19 pandemic [4]. Furthermore, a beehive gives humanity important insights during pandemics, where all complex interactions are carried out in order to preserve a healthy life of groups within the beehive [5].

During the COVID-19 pandemic, the concept of "flattening the curve" became the most discussed and researched subject by governments and researchers. It is a public health concept reducing the spread of the COVID-19. Siobhan Roberts presented an interesting article towards flattening the COVID-19 curve. He showed that reducing the spread of the COVID-19 infection is as necessary as halting it [6]. Although the best goal would be to completely eradicate the pandemic, slowing it down is also extremely critical for managing COVID-19. In this context, the author presented some mitigation measures towards the COVID-19 pandemic, e.g., social distancing to keep the population apart in time and space, actual self-isolation and quarantine [6].

Another related work in this context was introduced by Jackson Ryan in [7]. The author discussed the relation between social distancing and the concept of "flatten the curve" during the COVID-19 pandemic. He mentioned that many medical experts recommended more stringent social distancing, restricting/canceling mass gatherings, suspending business for many companies in certain sectors, etc. This means that such recommendations would help to impose social distancing during an epidemic and consequently slowing the spread of disease [7].

In this regard, several projects dealing with contact tracing applications. In [8], a new survey of COVID-19 contact tracing applications was conducted by Nadeem Ahmed et al. These smartphone applications are used to trace all recent contacts of people that were newly identified as positive infected with COVID-19. The authors presented an overview of several proposed smartphone tracing application examples. Additionally, they mentioned some future research directions for designing such applications towards improving tracing, large adoption by individuals, and safety performance [8]. Based on those, centralized/ decentralized/hybrid approaches (system architectures) used for developing tracing applications of COVID-19 were discussed. Additionally, the survey reported the pros and cons of each architecture, diverse attack and protection models, different implementation complexity and operating costs [8].

A more detailed discussion of such applications will be given in the section "Software application roles and responsibilities".

Regarding the implementation of social distancing policies to prevent the COVID-19 pandemic, several approaches from the literature were pointed out. Interesting work was analyzed to study the benefits and also the costs in the case of enforcing social distancing with the aim to flatten the curve of COVID-19 [9]. Linda Thunström and others pointed out that using social distancing, on one hand, will help save lives during the pandemic. Moreover, on the other hand, social distancing leads to very high costs for society as a result of decreased economic activities. The authors used epidemiological and economic forecasting to analyze the benefit-cost of controlling COVID-19 [9].

Another related work in this context was introduced by Arnab Ghorai and others [10]. The authors suggested a digital solution that enforces a social distancing policy using the Deep Learning technique. This technique detects any violation of the imposed social distancing using a tolerated threshold (pre-defined limit on several participants). To achieve this observation, a CCTV camera was installed so that a video stream could be captured. Accordingly, detecting and tracking the humans' motions are carried out by utilizing the PoseNet model (a machine learning model) [10].

Furthermore, the University of Texas M. D. Anderson Cancer Center presented an article dealing with the implementation of social distancing policies [11]. The article showed the relation between practicing social distancing and sustained reduction in COVID-19 transmission. It is a significant study that includes 50 US states and 134 nations, where its result indicates that social distancing can be enforced as an effective public health tool [11].

In addition to these studies, Per Nilsen and others introduce a new framework for analyzing a social distancing policy implementation in the context of the battle against the COVID-19 [12]. This framework aims to merge knowledge from two fields, where elements from implementation science and policy research can be combined. Accordingly, the author suggested a protocol for a comparative study of two countries, Sweden and Denmark. The comparative study reveals similarities and differences between both countries towards preventing the spread of the COVID-19 [12].

Although there are many studies/works made in the literature aiming to reduce the spread of COVID-19, a study of developing a novel algorithm, which allocates movement permits inside workplaces during/outside curfew period, does not exist yet (it is at least extremely rare reported in the literature).

Furthermore, most efforts introduced or suggested in the literature are trying to help the practice of social distancing towards "flattening the curve" during the COVID-19 pandemic. Such attempts will ultimately cost humanity and the economy a lot, tragic human consequences and economic uncertainties, which lead to disastrous results.

This work represents the first approach towards introducing an algorithm by enabling individual human agents to move safely inside workplaces avoiding infections according to COVID-19 social distancing guidance.

III. Software Application Roles and Responsibilities

There is a wide variety of software applications, which are developed to protect people during the COVID-19 pandemic, to manage public health emergencies, or to reduce the number of COVID-19 infections. One of the more interesting roles of these applications was providing instant and live information about the number of COVID-19 infections in a city or a country. In this case, early detection of possible infections will be insured directly after a user of this application shows COVID-19 symptoms. Furthermore, some software applications are trying to distribute movement permits to people that request them during curfew in necessary cases. Additionally, these applications update the permit requesters status during the curfew period to warn them as soon as they are closing to isolated or infectious areas. The more important role of those applications could be played by reporting all suspected cases of COVID-19 so that other people can receive healthcare services timely.

An interesting example of such an application is Tawakkalna App [13]. It is an official app issued by the Saudi Data and Artificial Intelligence Authority (SDAIA) in cooperation with the Saudi Ministry of Health. The goal of this app is to prevent the spread of COVID-19. Tawakkalna App makes it easy to issue movement permits during the curfew period. Additionally, it can be utilized for various health care purposes during the COVID-19 pandemic [13].

In this context, Corona-Warn is another aspect of the intended applications. For example, the Corona-Warn-App in Germany [14] is an app that traces infection chains of COVID-19 in Germany. This application uses a decentralized approach to warn users about exposure to COVID-19. The German government asked SAP and Deutsche Telekom subsidiary T-Systems to develop a Corona-Warn-App in May 2020. They are trying to integrate further information into the application, such as: the latest pandemic situation. The developers have summarized the steps of how the application works. Firstly, collect nearby identifiers using the Exposure Notification System on smartphones, where it should be regularly scanned for identifiers of other smartphones using Bluetooth, and consequently the identifiers can be stored locally. Secondly, communicate test results (optional) of users with symptoms. The tested users utilize the QR code that they received during their test to access their results themselves. Thirdly, distribute list of keys of COVID-19 confirmed users. Users with positive test results can voluntarily upload their last 14 days' temporary keys to the server. Fourthly, check for exposure to COVID-19 confirmed users. In this meantime, the Exposure Notification System will locally check if there is any matching between the downloaded list of users' keys which were tested positive and the locally collected Rolling Proximity Identifiers. If there is any successful match (exposure), the app user will receive immediately suitable support, if necessary [14].

IV. Are Technology Solutions Ready for Life after Corona?

Life after COVID-19 is new and not the same as before. Therefore, technology solutions should be the premier provider of services in society after COVID-19. In this context, movement permits during the COVID-19 pandemic is one of the more interesting solutions.

To understand the main goal, the next comparison is performed between the public health strategy, flatten the curve, and establishing a robust technical computer system.

Fig. 1 denotes how the public health strategy, called "flatten the curve", is used to slow the spread of a pandemic. The sharp curve shows the spread of a pandemic caused by a disease, for example, the COVID-19 virus, if no intervention (no proactive measures) strategies are provided through a community. This means that infected cases would rise rapidly if no interventions were made aiming to mitigate the spread. To reduce the daily infections and consequently to flatten the curve, some interventions (proactive measures, non-pharmaceutical) would be necessary, such as social distancing, hand washing, isolation, etc. Therefore, with intervention, the second curve (successfully flattened curve) becomes much flatter in the scenario of the COVID-19 pandemic.

Fig. 2 shows the main idea of establishing a robust technical system, which can tolerate faults, deviations, and disturbances that could be occurred in the system. The approach, in this figure, aims to make the system capable to return to its normal state with minimal central proactive measures (intervention) after disturbances, faults, or other problems [17]. Additionally, it can operate under real-time conditions, because a short response time is critical to the success of such robust systems. Moreover, robust technical systems should continue working effectively to fulfill their major tasks in cases of disturbances. Fault or disturbance tolerance plays an important role in avoiding system failures. This approach considers three cases of the system operation: firstly, operation without disturbance, secondly, operation with disturbance without intervention, and finally, operation with a disturbance with the intelligent intervention [17].

In this figure, the system performance is represented by throughput, as an example. If no disturbances occur, the performance of the system (throughput) should be equal to 1 (i.e., at its best). However, in the case of any disturbance occurrence, the system performance would start to collapse over time, if no corrective actions are taken in time. Otherwise, the system performance will recover over time when any disturbance occurs, only if intelligent and corrective intervention is quick enough. In other words, the system performance lasts longer despite disturbance occurrence [17].

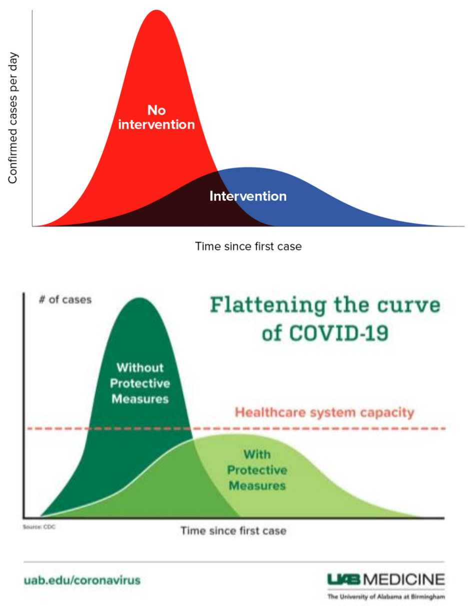

Fig. 1. The Public Health Strategy "Flatten the Curve" [7] [15].

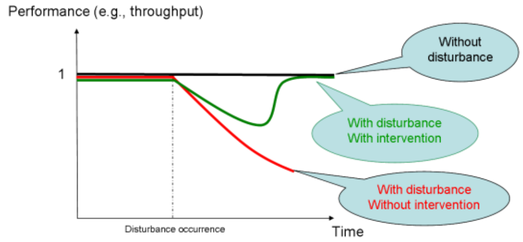

Fig. 2. Robust Technical System with Disturbance Occurrence [16].

Result of comparison

According to the previously introduced comparison of both Fig. 1 and 2, this section summarizes the results of this comparison as follows:

- The disturbance occurrence represents the start of the pandemic.
- Systems continue working effectively and fulfilling their major tasks would represent the peak of hospitalizations.
- System failures represent the peak of a pandemic spread.
- Intelligent intervention represents precautionary measures, giving/adapting movement permits, imposing social distancing, etc.

V. Objectives of the Approach

This section will briefly discuss the objectives of the proposed approach/algorithm in this paper. It forms a connection to utilize computer science algorithms/concepts to adapt our life under the COVID-19 pandemic. A more detailed discussion will be given in the "The Approach" section later.

In the context of this paper, a technology solution is presented to practice social distancing in certain places, where social distancing could be difficult. Such places can be inside workplaces, buildings, companies, and institutions. The goal is to optimize/adapt the movement permits in these places during/outside the curfew period. Additionally, it represents a strategy allowing the agents (people, robots) of the system to move reliably in their environment. For this reason, public health measures (precautionary measures) have to be enforced so that the spread of COVID-19 can be prevented or at least be reduced in workplaces. Furthermore, this paper proposes a solution for the resource allocation problem, called the resource sharing problem, where multiple agents (people, robots) moving in a shared environment trying to avoid collisions (infections).

VI. The Scenario

This section presents the application scenario proposed in this paper, where agents (people, robots) move in a shared environment inside workplaces during the pandemic period. They are trying to cross and move around as fast as possible using movement permits during/outside the curfew period. Furthermore, this section describes the required technique, which may include various capabilities. These capabilities are necessary to ensure that all agents can move safely avoid collisions (infections) in their workplaces.

Since many of the mitigation efforts of COVID-19 use computer models, graphics and simulations, an application scenario should be chosen so that it has a strong relationship with social distancing. Consequently, the application scenario introduced in this paper has a key role in giving/adapting movement permits for agents (people, robots) crossing a shared environment (resource allocation problem) inside buildings, e.g., the workplaces as depicted in Fig. 3. For this reason, a crossway control algorithm based on resource sharing is used. Such scenarios assemble the required concerns, which will be utilized to avoid collisions (infections) in workplaces.

In this scenario, a resource-sharing conflict arises. It is called a resource sharing problem (resource allocation problem). This problem should be resolved to prevent any collision within the shared place. Therefore, the coordination of agents is used later for the evaluation of the proposed scenario. In this context, a route planning algorithm is responsible for calculating the route with the shortest overall distance, if it exists, where safety is not overlooked in the presence of infections during the COVID-19 pandemic. Therefore, crowds in workplaces can be avoided helping to reduce the spread of COVID-19. In this regard, the proposed approach in this paper can use trajectories, where every agent travels using its planned trajectory. Additionally, adapting these trajectories according to the degree of safety is needed. For the special application domain, agents can be categorized into two classes: people (human) and robots.

When agents represent people in the applied scenario, then adapting trajectories (reallocation movement permits) is important to avoid infections in workplaces. However, if agents represent robots in other challenging scenarios, adapting such trajectories (adjusting movement permits) is required to avoid collisions in companies,workplaces or causing traffic jams.

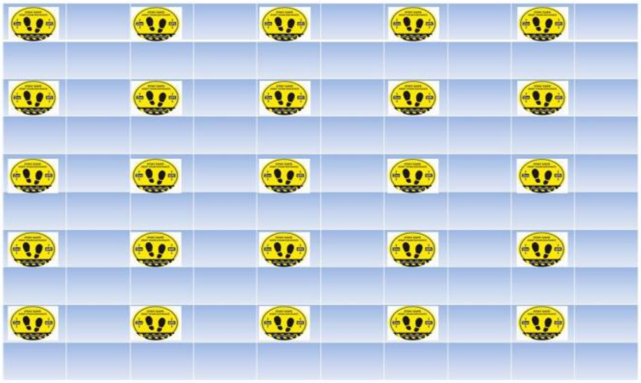

Fig. 3. The Shared Environment.

Finally, nowadays software products are using security cameras, which are based on AI to track social distancing. This is essential to counter the overcrowded facilities so that compliance with health guidelines, such as mask-wearing and social distancing, can be ensured or enforced as long as possible.

VII. The Approach

The new approach presented in this paper is intended to allocate and optimize/adapt all permits of agent's (people, robots) movement during/outside curfew periods. These agents are trying to move safely inside workplaces, buildings, companies, and institutions. In this approach, safety measures will be taken into consideration during the COVID-19 pandemic. This approach distinguishes between two different kinds of agents.

In the first case, agents represent people so that this approach reduces the spread of COVID-19, where policies for promoting the health safety of workplaces during a pandemic are needed. Therefore, preventing infections and avoiding crowds in workplaces is one of the main keys to safety. Furthermore, determining the most suitable social distancing according to the actual conditions will be necessary. Consequently, the next step is observing the maintaining of a suitable social distancing range and readjusting it if needed.

In the second case, agents represent robots that are safely moving in companies/workplaces aiming to avoid collisions. Additionally, agents are trying as quickly as possible to reach their target positions without any collision at all times.

In both cases, the general problem domain of this paper can be represented as a resource allocation problem (resource sharing problem) where several agents must travel from a source to a destination in a shared environment. Therefore, any proposed solution should be able to cope with this problem. The introduced approach in this paper chooses coordination mechanisms that can deal with the resource allocation problem. The coordination of these agents can be achieved through trajectory planning or path planning. Such planning should consider all agents moving in the shared environment and also the environment geometry in the configuration space-time (x, y, t). That means, the environment can be seen as a shared space over time.

According to the scenario presented in this paper, agents (people or robots) are trying to move safely in their environment to avoid collisions (infections). Additionally, every agent should get a reservation representing its route in the environment. This reservation can be configured as a recommendation to agents. It consists of the coordinates (x, y, t). This means that a reservation is a collision-free trajectory that can be used to coordinate the system agents. The agent should obey its planned reservations (trajectory), but this is not always guaranteed. Especially, in dynamic environments agents could have full autonomy and consequently, they have their own decisions and actions. Therefore, agents, in such cases, will not follow their recommended reservations. Consequently, an observation of the shared space (environment) is necessary to detect if agents obey the planned reservations (movement permits). If there is any deviation, then a reallocation process of resources should be initiated immediately so that all movement permits can be re-evaluated and consequently re-issued for all affected agents, if required.

A. *Trajectory Planning*

Nowadays, "safety measures", "keep a safe distance", "please keep your distance", "keep safe social distance", "keep physical distancing" are common signs everywhere. To achieve that, this section introduces the collision-free trajectories, which are required for all agents to move safely. These agents, using their planned trajectories, have to satisfactorily fulfill precautionary measures and health guidelines imposed by governments and safety authorities.

Intuitively, there will not be any conflict if every agent gets its trajectory uniquely. Therefore, a centralized control unit is required to calculate unique trajectories. As mentioned above, the trajectory planning in this paper is used as coordination mechanisms so that resource sharing problems can be solved, where agents move in a shared environment.

Firstly, all agents have local rules to move before they entered the shared environment as depicted in Fig. 4. This means, when people/robots leaving their rooms/base stations /home bases in workplaces and approaching a shared area (well-defined environment/crossroad/company yard), they do not need any planned trajectories from the centralized control unit. Otherwise, planned trajectories should be given for all agents inside the shared environment. These trajectories represent movement permits during/outside the curfew period. Additionally, these trajectories can be recalculated aiming to adapt/optimize the movement permits according to the actual conditions and safety degree.

The centralized control unit (trajectories planner) can simulate the agent's route through the shared environment. This simulation must consider the latest reserved trajectories for other agents using the configuration time-spaces, which is an extension of the configuration space of robots by a time axis. This should be done for every agent approaching the shared environment. The trajectory planner used in this paper is centralized so that it possesses a global view of all shareable resources that leads in turn to allocate collision-free trajectories.

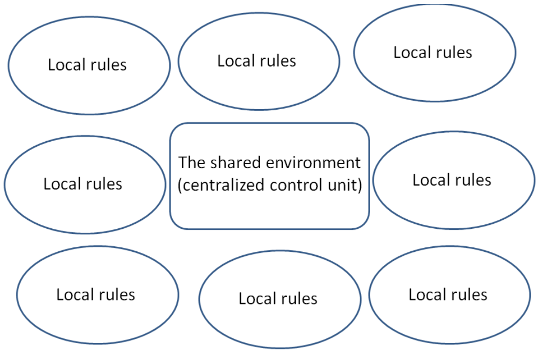

Fig. 4. Local Rules Outside the Shared Environment.

Every calculated route (trajectory) can be represented and planned as a vector (n-tuple) of points. Each point is modeled as space-time entities (objects) so that the (x_i, y_i) coordinates will be occupied at a time moment t_i.

$$\text{calculated route} = \{(x_1,y_1,t_1), (x_2,y_2,t_2),\ldots.., (x_n,y_n,t_n)\} \text{ with } 1\leq i\leq n;\ i,n \in N \qquad (1)$$

B. Algorithm

The algorithm serves as a basis to give an overview of how the centralized control unit works. It reacts according to input possibilities: request permission, agent class (people/robots), social distancing size. After handling the input values, it plans the most suitable routes (trajectories) as movement permits for agents. Furthermore, it can send emergency instructions if needed to send any warning signals to the participants in the case of an emergency. Additionally, it will adapt/recalculate these movement permits in two cases: firstly, if agents do not obey the planned routes. Secondly, if the given social distancing size was changed based on a new defined degree of safety. Accordingly, the input and output of the proposed algorithm can be summarized as follows:

Input: request permission, agent class (people/robots), social distancing size.

Output: movement permits (giving/adapting), reservations, calculated trajectories (new/adapt), emergency instructions.

In this algorithm, the Euclidean distance, which is a straight-line from a given point to the goal point, is the heuristics used to estimate the cost of reaching the goal point. Using these heuristics will ensure finding the shortest path (if it exists) and consequently planning optimal paths inside shared environments. This distance can be determined using the formula as described in the next equation:

$$\min \parallel (x_s, y_s) - (x_g, y_g) \parallel \ ;\ (x_s, y_s)\text{: start point, } (x_g, y_g)\text{: goal point} \qquad (2)$$

Comparing to the algorithm described above, an adapted algorithm, which can be used in this approach, can have the following features:

- The function recalculateRoutesOfAffectedAgents() can be used to recalculate trajectories if some agents violate the planned routes. In this context, the presence of modified policies of social distancing size (SDS) can be taken into account.
- The function ModifySocialDistancingSize (SDS) is used when the current social distancing policy is modified by the state or the city to force new requirements or to give recommendations. This can be possible by reopening after the COVID-19 lockdown but under modified social distancing orders and regulations.

Algorithm: Overview of "Movement Permits During Corona" Algorithm

---

```
AlgorithmMovementPermitsDuringCorona (startPoint,
goalPoint)
BEGIN
Input (RP,AC, SDS); // RP: request a permission,
AC: agent class, SDS: social distancing size.
Initialise (C, RSM, startPoint); //C: calculated
Route, RSM: Resource Sharing Map
                CostSoFar (startPoint) = 0;
bestCalculatedRouteSoFar (startPoint) = startPoint;
updatedRSM = insertToRSM (startPoint), Cost
(startPoint) = 0;
while not isEmpty (updatedRSM) do
        nextPoint =
returnNextPointFromRSM(updatedRSM);
        if (nextPoint = goalPoint) then
                return bestCalculatedRouteSoFar ();
        end if
        for all neighbours in RSM(nextPoint) do
                =
                calculateTheRestPartOfRoute(nextPoin
                t,SDS); RPR
                calculateTheCostOf(RPR);
                bestCalculatedRouteSoFar
                (neighbours) = nextPoint;
        end for
end while
        if (situation = emergency) then
                findTheBeingWell-
                preparedForTheEmergency();
                return bestEmergencyInstructions();
        end if
Output(movement permits, emergency instructions).
END
```

VIII. Performance Evaluation

In this section, an initial evaluation of the algorithm proposed in this paper is presented. It uses the model of a shared environment inside workplaces during the pandemic period as depicted in Fig. 3. As mentioned above, in this paper, agents (people or robots) move within a shared environment. This application scenario served as a test bed for the evaluation of the approach presented in this thesis.

Since the trajectories planning plays an important role in the user application scenario to achieve high throughput (performance), an evaluation of this algorithm is necessary under different loads of agents. In this regard, we assume that there won't be any deviations from planned trajectories. Furthermore, we assume that the social distancing size (SDS) is equal to 1 cell in the modeled shared environment.

In future work, on one hand, the handling of deviations from planned routes should be studied and considered. Moreover, on the other hand, modified social distancing size (SDS) should be taken into account.

Additionally, two metrics were used for evaluating the approach: throughput and mean waiting time. Firstly, throughput presents the total number of agents that left the shared area (environment) over time (#Agents/tick). Secondly, mean waiting time presents the average time (iterations/ticks) needed by agents to cross the shared environment.

All required experiments have been carried out based on a MAS simulation using the "Repast" (Recursive Porous Agent Simulation Toolkit) framework [18].

A. Evaluation Scenarios

To compare the system performance in different evaluation scenarios, two simulation parameters were used for measuring the system performance. Firstly, A_{max} presents the maximum number of all agents in the whole system. Secondly, P_{rate} is the production rate of agents in the whole system (agent flow rate). Modifying the values of A_{max} and P_{rate} will be useful for ensuring effective performance of the proposed approach in different combinations of the simulation parameters. Particularly, this effectiveness of system performance can be guaranteed even in crowding, e.g. during rush hour for employees arriving for their work; where it would be difficult for them to get their work done at their company during the pandemic.

Both defined metrics for performance evaluation were measured in a simulation interval between 0 and 3000 ticks (time steps). As depicted in Fig. 4, agents can flow in all directions, and consequently, four directions should be taken into account.

In this scenario, the shared environment is represented as a 10x10 grid cell, where every cell can be considered as a reservation tile. For example, the throughput and the mean waiting time (MWT) of the system were measured in the case that the production rate of agents (P_{rate}) in the whole system is 5 agents/tick. Furthermore, the measurement was repeated in the cases that the maximum number of agents in all directions (inside and outside the shared environment) is 40, 60, 80, 100, 120, and 500 agents. It is worth mentioning that the case of 500 agents in all directions can be seen as a crowding threshold (an extreme case) in this scenario. Additionally, a series of experiments were performed by changing the production rate of agents (P_{rate}) in the whole system: 10, 13, 18, 20 agents/tick.

B. Results

1) Throughput measurement: Fig. 5 displays the system throughput per time unit (#Agents/tick) for the evaluation scenario described above using some combinations of both simulation parameters: Amax, Prate. As mentioned earlier, The throughput was measured for a simulation interval between 0 and 3000 ticks.

Fig. 5 shows that from the simulation time/iteration 115 (it is an approximation and not an accurate simulation time) agents start leaving the shared environment. This can be explained by the fact that the shared environment was free from agents at the simulation start. This in turn leads to the system throughput per time unit, which is 0 during the simulation interval between 0 and 115. However, shortly after that, the system throughput per time unit (#Agents/tick) is constantly increasing with the passage of simulation time. This can be traced back to the system throughput per time unit as it goes in fair proportion to the number of agents. However, there is a single exception to this statement only in the extreme case/crowding threshold (500 agents in all directions). In this case, the system performance reaches only a value of around 9400 agents. That is because the maximum number of agents, in this case, is greater than the capacity of the whole designed environment. Consequently, it concludes that the system throughput per time unit is increasing steadily within the environment's capacity (resource consumption) with the number of agents for all possible combinations of both simulation parameters. It is worth reminding that one tick in the used "Repast" simulator means one-time step.

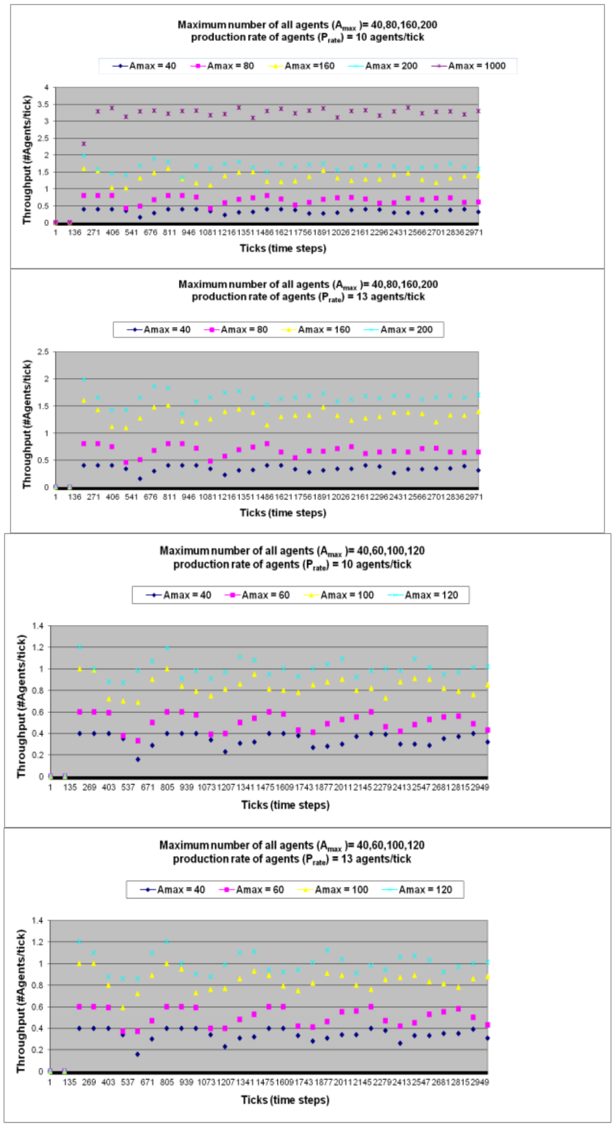

Fig. 5. The System throughput Per Time unit (#Agents/Tick) for the Evaluation Scenario for a Simulation Interval between 0 and 3000 Ticks

2) Mean waiting time measurement (MWT): This metric can be evaluated in many cases as the first metric, throughput, mentioned above. However, it is noteworthy in this context that the mean waiting time is calculated in an extreme case. In this case, A_{max}= 1000 is the maximum number of all agents in the whole system inside and outside the shared environment. This case represents a crowding threshold, where the maximum number of agents is in fact greater than the capacity of the shared environment. Consequently, such measurement will give an environmental protection to authority by calculating the longest waiting time spent by agents crossing a shared environment during the pandemic period.

In this context, several experiments have been carried out aiming to study the mean waiting time in different values of production rate (P_{rate}) of agents. Therefore, the values of the mean waiting time of all agents and their standard deviations were calculated after the simulation end time in the extreme case (A_{max}= 1000) as depicted in Table I.

As a result, despite having big values of agents, the designed system has low mean waiting times (Latencies) with small standard deviations. That applies in different production rates of agents, where the worst mean waiting time was Φ 4.290499947151464 ± 1.3562567367584307 in case of the biggest P_{rate} applied in simulation experiments of this paper.

Furthermore, this exploratory qualitative study also measured total waiting times (aggregated latencies) and mean waiting times (mean latencies) in normal cases (not only the extreme case), e.g., by A_{max}=200, P_{rate}=8 during the whole simulation period.

Firstly, Fig. 6 shows the total waiting times (aggregated latencies) in the case of A_{max}=200, P_{rate}=8 during the whole simulation. The experimenter here can see that the worst total waiting times which agents experience is about 250 ticks as depicted in Fig. 6.

Secondly, Fig. 7 shows the mean waiting times (mean latencies) in the case of A_{max}=200, P_{rate}=8 at any time during the experiment simulation. The designer here will see that the worst mean waiting time which can be measured is ca. 1.3 ticks as depicted in Fig. 7.

Thirdly, Fig. 8 shows the maximum waiting times (maximum latencies) in the case of A_{max}=200, P_{rate}=8 during the whole experiment. These measurements showed that the biggest waiting time which can be experienced is 7 ticks as depicted in Fig. 8.

TABLE I. MEAN WAITING TIMES (LATENCIES) OF ALL AGENTS AND THEIR STANDARD DEVIATIONS AFTER THE SIMULATION END TIME IN THE EXTREME CASE

A_{max}	P_{rate}	**mean waiting time (Latency in ticks)**	**standard deviations**
1000	5	2.7831321540062435	0.9451051160462923
	7	3.5283895921237694	1.4656030736537114
	10	3.872145144076841	1.836589145239349
	13	4.147975811584978	1.5825953904261338
	15	4.290499947151464	1.3562567367584307

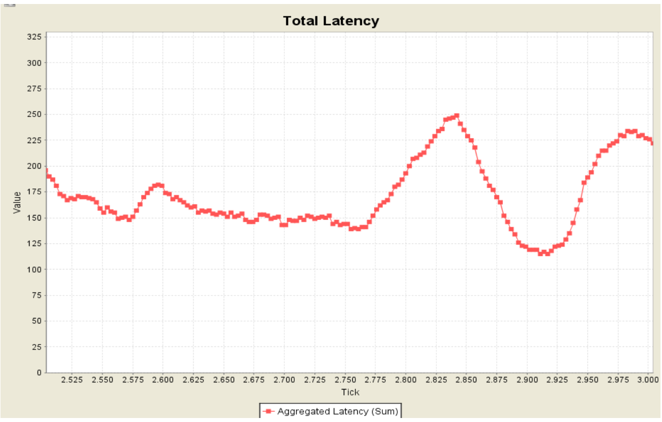

Fig. 6. Total mean Waiting Time (Aggregated Latencies) by Amax=200, Prate=8.

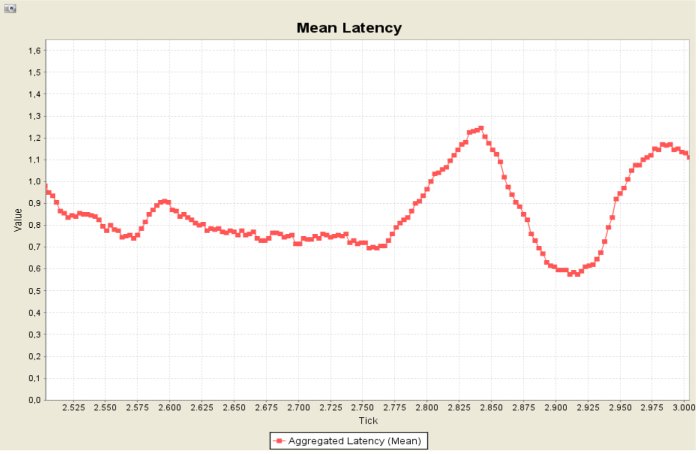

Fig. 7. Aggregated mean Waiting Time (mean Latency) by Amax=200, Prate=8.

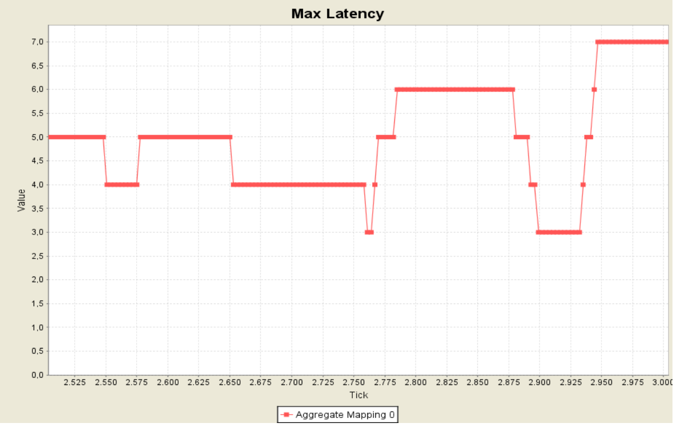

Fig. 8. Maximum Waiting Times (Maximum Latencies) by Amax=200, Prate=8.

IX. CONCLUSION

The tightened measures against COVID-19 are supported by a clear majority all over the world these days. This in turn led many authorities in the world to restrict movement and range of motion or even impose nationwide shutdowns. In this case, lockdown also caused a shut down of social life during the COVID-19 pandemic. Therefore, studies were initiated aiming to prevent the imposition of such lockdowns and consequently to mitigate the bad economic impacts on society and businesses. In this context, this paper presented a new approach/algorithm as a technical solution towards achieving a desired target safety margin despite COVID-19 disturbances. This approach was trying to adapt the social distancing by movement permits inside workplaces, institutions, companies, and big buildings. On other hand, the paper proposed a possibility for solving the problem called "resource sharing problem" or "resource allocation problem". It supposes that the designed system consists of many agents representing people or robots aiming to reserve suitable resources of their shared environment. The evaluation of the proposed algorithm that manages all movement permits (resource allocations) showed the high performance of this algorithm (high throughput values and low waiting times or latencies). Additionally, all experiments were based on various loads of agents to ensure their safe and reliable operation.

X. Future Work

Comparing to the algorithm evaluation described above, an extension is necessary to incorporate detailed results and expanded experiments. Especially, detecting and handling possible deviations from routes planned from the algorithm would be the first step in future work. Furthermore, taking into account that social distancing size (SDS) can be modified due to changed policies and procedures would be the second step in the future work as well. After that, the paper extension will be satisfactory evidence of compliance with the applicable requirements of Corona-stipulated social distancing.

References

[1] The World Health Organization (WHO). https://www.who.int/ [access July 2020].

[2] Healthcare Market Research. “Technology Solutions - Coronavirus (COVID-19) Case Study”. Pages : 18 - Publisher : GlobalData - Report code : ASDR-536189 , 2020. [access October 2020]. Available from https://www.asdreports.com/market-research-report-536189/technology-solutions-coronavirus-covid-case-study.

[3] IEEE ProComm. “Flatten the Curve: Why certain messages catch on”. IEEE Professional Communication Society (ProComm). 05 May, 2020. https://procomm.ieee.org/flatten-the-curve-why-certain-messages-catch-on/.

[4] R. Bonoan and P. Starks. “Honey bees can’t practice social distancing, so they stay healthy in close quarters by working together”. The Conversation. Aug 14, 2020. theconversation.com.

[5] R. Bonoan and, P. Starks. “Honey Bees Stay Healthy In Such Close Quarters”. The innerself . August , 2020. innerself.com.

[6] S. Roberts. “Flattening the Coronavirus Curve”. The New York Times. ISSN 0362-4331, 27 March 2020. [access July 2020]. https://www.nytimes.com/article/flatten-curve-coronavirus.html.

[7] J. Ryan. “Coronavirus pandemic: How social distancing can help flatten the curve”. CNET, 16 March 2020. [access July 2020]. https://www.cnet.com/news/coronavirus-pandemic-how-social-distancing-can-help-flatten-the-curve/.

[8] N. Ahmed, R. A. Michelin, W. Xue, S. Ruj, R. Malaney, S. S. Kanhere, et al. “A Survey of COVID-19 Contact Tracing Apps” in IEEE Access, vol. 8, pp. 134577-134601, 2020, doi: 10.1109/ACCESS.2020.3010226. https://ieeexplore.ieee.org/document/9144194.

[9] L. Thunström, S. C. Newbold, D. Finnoff, M. Ashworth, and J. F. Shogren. “The Benefits and Costs of Using Social Distancing to Flatten the Curve for COVID-19” in Journal of Benefit-Cost Analysis, 11(2):1-27,April 2020, doi: 10.1017/bca.2020.12. https://doi.org/10.1017/bca.2020.12.

[10] A. Ghorai, S. Gawde, and D. Kalbande. “Digital Solution for Enforcing Social Distancing”. Proceedings of the International Conference on Innovative Computing & Communications (ICICC) May 31, 2020, Available at SSRN: https://ssrn.com/abstract=3614898 or http://dx.doi.org/10.2139/ssrn.3614898.

[11] University of Texas M. D. Anderson Cancer Center. “Implementation of social distancing policies correlates with significant reduction in SARS-CoV-2 transmission: Study of 50 US states and 134 nations reinforces social distancing as effective public health tool”. ScienceDaily. Available at: www.sciencedaily.com/releases/2020/07/200730161025.htm.

[12] P. Nilsen, I. Seing, C. Ericsson, O. Andersen, N. Thórný Stefánsdóttir, T. Tjørnhøj-Thomsen, et al. “Implementing social distancing policy measures in the battle against the coronavirus: protocol of a comparative study of Denmark and Sweden”. In Implementation Science Communications 1, 77 (2020). https://doi.org/10.1186/s43058-020-00065-x.

[13] Saudi Data and Artificial Intelligence Authority (SDAIA). “Tawakkalna App”. About Tawakkalna, 2020. https://ta.sdaia.gov.sa/en/index [access July 2020].

[14] The German government, SAP and Deutsche Telekom subsidiary T-Systems. “Corona-Warn-App App in Germany”. About Corona-Warn-App in Germany, 2020. https://www.coronawarn.app/en/ [access October 2020].

[15] H. Herfurth. “What exactly does it mean to 'flatten the curve'? UAB expert defines coronavirus terminology for everyday life”. April 28, 2020, The University of Alabama at Birmingham (UAB). https://www.uab.edu/news/youcanuse/item/11268-what-exactly-does-it-mean-to-flatten-the-curve-uab-expert-defines-coronavirus-terminology-for-everyday-life.

[16] Y. Chaaban and C. Müller-Schloer. “A Survey of Robustness in Multi-Agent Systems”, in Cognitive13: proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications, 2013, pp. 7-13.

[17] Y. Chaaban. “A Methodology for Designing Robust Central/Self-Organising Multi-Agent Systems”, in International Journal of Computer Information Systems and Industrial Management Applications, Vol. 6, 2014, pp. 571-581

[18] RePast (Recursive Porous Agent Simulation Toolkit) framework. http://repast.sourceforge.net/, [accessed, September 30, 2020].