Minimal Contact Optimal Routing across Connected Networks for Stochastically Arriving Agents: A Mixed Integer Linear Programming Framework

Apoorva Narula, Prasanna Ramamoorthy, and Varun Ramamohan

***Abstract*—This paper presents an optimization framework for the real-time generation of minimal contact routes for randomly arriving agents within a connected network. The study focuses on minimizing interactions between agents visiting specific nodes, considering overlaps in their durations of occupancy of the nodes as contacts. The effectiveness of the routing strategies is demonstrated through stochastic simulation experiments, highlighting the potential of this approach for enhancing efficiency and safety in dynamic network environments. This makes use of Mixed Integer Linear Programming and Robust Optimization techniques. The proposed framework has applications in logistics for automated warehousing and pandemic-driven customer routing in supermarkets, to name a few.**

***Index Terms*—Minimal Contact Routing, Traveling Salesman Problem, Mixed Integer Linear Programming, Robust Optimization, Automated Guided Vehicles, Collision Avoidance**

I. Introduction and Literature Review

This study considers a system in which agents arrive stochastically at a fully connected network, where a path exists between all pairs of nodes. There is no unique fixed path between any two nodes; instead, as all nodes are interconnected, an agent can choose the combination of edges they prefer to reach the desired node. Each arriving agent possesses a list of nodes they intend to visit, and our objective is to assign paths to these agents in a way that minimizes their interactions or contacts with other agents in the network.

A contact between two agents is defined as the overlap in their duration of stays on a particular node. Therefore, contacts between agents are only considered on nodes and not in the edges or paths of the network. It is important to note that due to the assumption of the presence of free-ranging Automated Guided Vehicles (AGVs) – that is, AGVs not dependent on preinstalled guide paths and able to decide on their path by choosing a combination of edges – concerns about collision avoidance along network edges are less likely to arise. This observation is highlighted by Duinkerken et al. (2006) and Xin et al. (2020).

The objective of this study is to formulate strategies for providing paths to newly arriving agents to the network such that the contacts between these agents is minimized, while also minimising the distance travelled by the agents. Furthermore, the study also considers stochasticity in system parameters such as agent traveling speeds and node dwell times, which are likely to occur in real-world operations.

In recent years, there has been a growing interest in addressing collision-free routing issues across diverse domains. For instance, Herrero-Perez et al. (2010) propose a decentralized navigation control approach for solving navigation conflicts between AGVs. Each AGV calculates its own paths, which eliminates the need for zone control along the whole workspace and drastically simplifies the modeling of behaviors, increasing the performance of the MHS. However, there are situations that can induce blockages and deadlocks, mainly because of limited sensing capabilities and AGV traffic jams. Korsah et al. (2013) discuss the Multi-robot task allocation (MRTA) problems , which are the problems of determining which robots should execute which tasks in order to achieve the overall system goals in a multi-robot system. The features and complexity of MRTA problems are dictated by the requirements of the particular domain under consideration. These problems can range from those involving instantaneous distribution of simple, independent tasks among members of a homogenous team, to those requiring the time-extended scheduling of complex interrelated multi-step tasks for members of a heterogeneous team related by several constraints. Lee et al. (2014) and Bullo et al. (2011) also address a similar problem in the task-allocation context. Guillaume et al. (2017) present an approach to decentralized motion planning and scheduling for Automated Guided Vehicles (AGVs) within a flexible manufacturing system. The core strategy integrates a motion planner with a scheduler, empowering each AGV to dynamically update its destination resource during navigation, acilitating in collision avoidance. The proposed methodology unfolds in two key steps. Firstly, collision avoidance planning is addressed by planning presumed trajectories, avoiding conflicts identified by a central supervisor. This proactive planning sets the stage for more effective decentralized scheduling by AGVs. Subsequently, the second step leverages the exchange of presumed trajectories with neighboring AGVs, enabling the computation of collision-free trajectories based on a priority policy. This approach ensures that AGVs navigate safely, sidestepping potential collisions and

V. Ramamohan is with the Department of Mechanical Engineering, Indian Institute of Technology Delhi, New Delhi 110016, India. E-mail: varunr@mech.iitd.ac.in.

A. Narula is with the Department of Mechanical Engineering and P. Ramamoorthy is with the Department of Management Studies, both at the Indian Institute of Technology Delhi.

optimizing overall system efficiency. In this study, stochastic elements are incorporated into the agent's traveling speed and the time spent by the agent on a node, referred to as the node dwell time. These stochastic factors are considered in the path assignment formulations utilized by a central supervisor. Spensieri et al. (2015) address the challenge of coordinating collision-free movements and scheduling welding tasks within an automotive assembly line. Similarly, Xin et al. (2020) explore the collision-free path planning and routing of multiple robotic effectors on an assembly line.

In addition to the above, a few more approaches used by researchers to address similar problems have been further described. Shi et al. (2018) propose a collision-free path planning algorithm for unmanned aerial vehicle (UAV) delivery. It uses the A* algorithm to optimize the route through off-line planning and considers waiting time, collision avoidance constraint, and battery constraint. The algorithm also uses a data structure called CurrentSchedule for checking and updating the availability of airspace. The simulation results show desirable runtime performance and could aid the decision-making of no-fly zone policy and infrastructure of UAV delivery. The paper further mentions that a fundamental challenge to online algorithms is the modelling of uncertainties during flight. Kim et al. (2007) proposed a probabilistic trajectory model in 3D space that considers multiple trajectories with different degrees of derivations to the baseline trajectory. The probability of conflicts is calculated for each potential trajectory, and the one with the lowest probability of conflicts is selected. However, the proposed algorithm in this work is an off-line algorithm that plans the path prior to departure and does not explicitly consider stochasticity in any parameter. In comparison to Shi et al. (2018) and Kim et al. (2007), the robust formulations presented in this study differ in terms of the context of producing minimal contact routes. It considers contacts on the destinations only. For example, while modeling a supermarket using our formulations, the generated paths would minimize the queuing of the agents on each of the destinations, to prevent possible reneging and long-time agent-to-agent contact, which would be relevant to prevent disease spread in a pandemic situation, such as COVID-19. As another example, when the case of modeling an AGV automated warehouse is considered using proposed formulations, the paths generated would prevent congestion in the system due to the queuing of AGVs at a node, which can lead to collisions between the AGVs, leading to some damage in the same. Furthermore, unlike Shi et al. (2018) and Kim et al. (2007), our formulations are capable of accounting for stochasticity in parameters such as agent traveling speed and node dwell times and significantly decrease contacts under uncertainty.

Xin et al. (2020) discusses about a collision-free routing problem for multi-robot systems. The authors propose a time-space network (TSN) model and a genetic algorithm to solve the problem. The proposed genetic algorithm for the Mixed Integer Programming problem based on the TSN formulation uses a two-dimensional encoding scheme to represent the robot paths and satisfy the constraints of a predefined task sequence and collision avoidance. The algorithm consists of the main procedures of selection, crossover, and mutation, and uses a dedicated fitness function to evaluate the quality of the solutions. As a metaheuristic algorithm, the GA inherently involves some level of stochasticity in the selection, crossover, and mutation procedures. The randomness in these procedures allows the algorithm to explore the solution space and avoid getting stuck in local optima, however, unlike our formulation, the model does not specifically account for stochasticity in the travelling speed of the robots or the time spent by them on any destination in the network. Our robust MILP-based approach captures this using box uncertainty sets assigned to the stochastic parameters as mentioned previously.

Spensieri et al. (2015) propose a methodology to schedule the operations of different types of machines in automated container terminals in a way that ensures that the machines do not collide with each other while performing their tasks. The methodology takes into account the fact that AGVs can move freely in the terminal, which makes their trajectory planning more complex than that of machines that move along fixed paths. The proposed methodology includes a hierarchical control architecture that decomposes the scheduling problem into two stages: determining the sequence of jobs for each particular piece of equipment and the time window during which each job is processed, and determining the actual time window of each job incorporating collision-free trajectories of AGVs. By using this methodology, the authors aim to improve the efficiency and safety of automated container terminals. The proposed algorithm is a sequential planning approach for collision avoidance in automated container terminals. It first determines the sequence of jobs for each particular piece of equipment by solving a hybrid flow shop scheduling problem. Then it obtains an overall graph sequence based on the job sequences for the pieces of equipment and determine the actual time window of each job incorporating collision-free trajectories of AGVs by solving a collection of mixed integer linear programming problems sequentially. Then it uses a neighborhood variable search metaheuristic to provide a fast initial solution to the proposed sequential planning algorithm for real-time application. The approach of this paper is different from ours in terms of classifying the machines into different types in a deterministic manner and then accounting for the difference in their behavior in their formulation. We, however, consider stochasticity in the behavior of our agents, by considering the time spent by them on each node, along with their traveling speed in the network as independent and identically distributed, sampled from known distributions.

Along with the approaches discussed above, Xin et al. (2015) addresses the problem of optimizing sequences of operations, such as welding, in collaborative manufacturing stations employing multiple industrial robots. The primary objective is to minimize the station cycle time, defined as the duration until the last robot completes its cycle. The approach involves task dispatching among robots and devising collision-free routes and schedules to ensure the completion of predefined tasks. The proposed iterative and decoupled methodology manages the problem's high complexity. Initially, collisions among robots are disregarded, leading to the formulation of a min–max Multiple Generalized Traveling Salesman Problem (MGTSP). Subsequently, after determining and fixing the sets of robot loads, task sequencing and scheduling are performed to prevent conflicts. The first problem is addressed using an exact branch and bound (B&B) method, incorporating different lower bounds derived from solutions to a min–max set partitioning problem and a Generalized Traveling Salesman Problem (GTSP). The second problem assumes synchronous robot movement, introducing a novel transformation of this synchronous problem into a GTSP. Our approach distinguishes itself from these methods in two aspects: firstly, in the context of collision, as we specifically focus on collisions occurring only at nodes. This emphasis aligns with our intended application, which, as an example, could be optimizing logistics for AGV-based warehousing. In contrast, Xin et al. (2015) consider collisions along routes, where pre-installed paths in a manufacturing setup are common, leading to collisions along these predefined paths. Secondly, our approaches also differ in the manner in which we account for stochasticity. In our approach, we account for stochasticity in the system parameters in the robust MILP formulation itself, through rectangular uncertainty sets.

Sen et al. (2021) explored a similar problem and presented an optimization framework for identifying routes through a connected network to minimize or eliminate contacts between agents visiting specific nodes within minimal time. However, their approach assumes constant node dwell times and traveling speeds in the formulation used by them. In our work, we consider stochasticity in agent traveling speeds within the network and the time spent by each agent on a node. To address uncertainties, we design a Robust Optimization-based formulation incorporating box uncertainty constraints, as described by Ben-Tal et al. (2002), to minimize contacts in a stochastic framework. Furthermore, we propose a Time Windows based MILP formulation for the problem and compare the same to the Miller-Tucker-Zemlin (MTZ) formulation for the problem proposed by Sen et al. (2021), and show the efficacy of our model in terms of reducing contacts. Detailed discussions of the TSP and related variants can be found in work by Applegate et al. (2006); Cook (2011); Reinelt (1991); Toth et al. (2014). Details regarding the TSP-Time Windows based formulation (TWP-TW) and TSP-MTZ based formulations can be found in Dumas et al. (1995) and Miller et al. (1960), respectively.

In the context of the aforementioned investigations, our contributions are delineated as follows:
(i) We address the real-time generation of minimal-contact routes for randomly arriving agents, each possessing independent sets of tasks, denoted as node sets for visitation.
(ii) We introduce a methodology for formulating minimal-contact routes through an extension of the Time-Window (TW) formulation within the Traveling Salesman Problem (TSP) framework. This extended formulation demonstrates superior performance compared to previously established formulations utilizing the MTZ framework, as evidenced by both computational runtimes and the aggregate count of contacts, obtained using simulation experiments.
(iii) We explicitly incorporate considerations pertaining to uncertainties associated with agent velocity and node dwell time. We put forth a robust optimization formulation designed to accommodate such uncertainties, thereby enhancing the adaptability and resilience of the proposed routing strategy, to account for randomness in the real world implementation.

II. NETWORK DESCRIPTION AND SIMULATION DETAILS

At the beginning of each simulation experiment, we generate a sequence of interarrival times for agents until an agent's arrival time exceeds the simulation's time horizon, set at four hours in our case. Subsequent to generating arrivals for this agent set, we assign node sets to each agent, representing the nodes the agents plan to visit upon arrival in the connected network. Routes are then created for each agent, considering their node set, anticipated traversal speeds, expected node dwell times, and traversal pattern based on the MILP formulation used used for that simulation experiment.

Stochasticity is introduced into an agent's traversal at two levels:
(a) the speed of movement between nodes, and
(b) the duration spent at each node (node dwell time).
Once a route is assigned to an agent, the actual speeds of movement between each pair of nodes on the route and the dwell times at each node are sampled from their respective distributions, as detailed below. Using this information, the anticipated entry and exit times at each node are recorded for each agent, based on the path assigned to the agent by the formulation. Contacts are identified by checking whether there is any overlap between the actual entry and exit times of agents visiting each node, and this information is recorded separately.

It is crucial to note that the actual entry and exit times are not inputted into the formulations. The formulations receive only the expected node arrival times and dwell times based on the mean values of simulation parameters, such as agent traveling speed and node dwell time, derived from the path assigned to the agent by the formulation.

Furthermore, the network which we consider in our simulation experiments consists of one hundred nodes, including an entry/exit node, within a 100-meter by 100-meter square area. The coordinates of these nodes have been chosen through a uniform distribution in both the length and the width of the square area. This layout represents a fully connected network where each node is interconnected with every other node. The length of the path connecting two nodes is calculated using the Euclidean norm of their coordinate differences.

We now give a description of all the parameters we consider to the capture the behaviour of the agents in the network:

1. **Arrival of Agents**:
   - We model agent arrival using a Poisson distribution. Thus, the inter-arrival times are considered to follow an exponential distribution, as is the case in a Poisson process.
   - For analysis, we assume the mean of the Poisson arrival distribution has an arrival rate $\lambda = 100$ agents/hour, unless mentioned otherwise.

   Node Set Generation for an Agent:
   - We randomly assign each agent a node set, considering a uniform probability distribution across all the nodes.
   - To enhance realism in the abstraction, we limit the total number of nodes assigned to any agent to twelve, excluding the entry and exit gates. This cap limit can be adjusted based on the specific application scenario.
   - Alternatively, the node assignment process can be executed in a way where certain nodes are visited with a higher probability than others. This variation can be tailored to the specific application of these formulations, adapting to different situations.
2. **Traversal Patterns**:
   - The traversal pattern of an agent consists of the order in which an agent visits the nodes in its node set.
   - Traversal patterns to reduce contacts with other agents are generated using the MILP formulations.
3. **Node Dwell Time**:
   - By default, agent dwell times are considered to follow an exponential distribution with a mean of 2 minutes. The exponential distribution is chosen to accommodate scenarios where node dwell time lacks predictive value. Consequently, we view the exponential distribution of node dwell time as a worst-case scenario in this context. Additionally, we conduct a sensitivity analysis where we vary the node dwell distributions to observe how our results fluctuate with different node dwell time distributions.
   - In deterministic scenarios, a fixed node dwell time of 2 minutes is employed.
4. **Agent Speed**:
   - Agent speed is modeled using the normal distribution, and the absolute value of the sampled random variable is considered.
   - We use the mean speed as $v = 15$ meters per minute, with a standard deviation of 9 meters per minute (60% of the mean), unless otherwise specified.
   - Deterministic scenarios employ a constant velocity of 15 meters per minute.

We conduct two sets of simulation experiments. In the first set, we assume fixed values for node dwell times and agent traveling speeds, set equal to their expected values. The second set of simulation experiments introduces stochasticity in these parameters, following the probability distributions described earlier.

For a clearer understanding of the concept of a minimal contact route, an example of such a route is illustrated in Fig. 1, depicting two agents visiting a connected network.

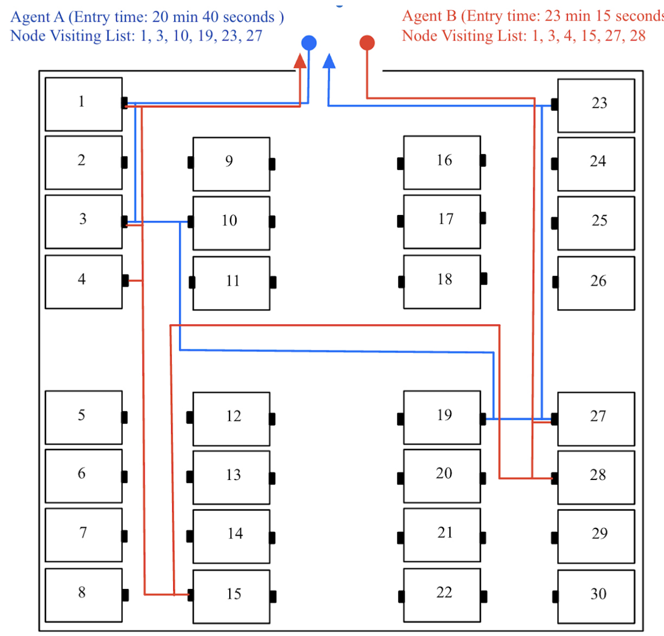


Fig. 1: Example 1 - Contacts in the example path for agents A and B

TABLE I: Node Occupancy for Agents in Example 1 (The values in the example are arbitrarily chosen to describe the simulation experiments better)

Node	Node Occupancy (minutes : seconds)	
	Agent A	Agent B
1	21:18 - 22:58	45:40 - 48:11
3	23:46 - 25:53	42:46 - 44:52
4	—	40:25 - 42:22
10	26:05 - 28:11	—
15	—	35:56 - 38:01
19	30:59 - 32:53	—
23	37:45 - 40:03	—
27	33:05 - 35:09	26:51 - 29:03
28	—	29:28 - 31:47

III. Path Assignment Formulations

In this study, we develop a mixed integer linear programming (MILP) framework for generating optimal minimal contact routes for agents that arrive randomly (i.e., interarrival times are stochastic) to a connected network. We develop two extensions of the time windows (TW) mixed integer programming formulation of the traveling salesman problem (TSP) for generating optimal routes for an agent tasked to traverse its node set (i.e., the set of nodes it is assigned to visit within the network). Each of the TSP TW formulations that we develop have two objectives:

(a) minimize the time spent traversing the agent's node set; and

(b) minimize (if not eliminate) all contacts at nodes with other agents already in the network that may have intersecting node sets.

We develop two TW formulations:
(a) first, to find an optimal route that eliminates all contacts with agents already in the network, referred to in short as the TW-NC formulation (where NC stands for no contact, and TW stands for time windows); and
(b) second, to find an optimal route that minimizes all contacts with agents already in the network. The second formulation is referred to as the TW-MC model, with MC being an acronym for minimal contact. This formulation may yield routes that may have a non-zero number of contacts; however, we incorporate a penalty term for contacts in the objective function along with minimizing the time spent in the network. The weightage given to minimizing contacts relative to minimizing time spent in the network may be adjusted as desired.

The formulations extend the TSP TW formulation proposed by Dumas et al. [10], by incorporating decision variables and constraints that take into account the time at which the agent under consideration reaches each node as well as the time points at which agents already in the network are anticipated to visit and exit nodes in the node set of the agent in question. We also consider uncertainty in two key model parameters: velocity of travel of the agent between nodes and the time spent by an agent at a node, and hence we also develop and include a robust optimization version of the TW-MC formulation with asymmetric box uncertainty sets in our optimal minimal contact routing framework.

In the pursuit of establishing a benchmark for evaluating the path assignment strategies developed by us, we adopt two foundational formulations denoted as Appendix Formulation 1 (MTZ-NC) and Appendix Formulation 2 (MTZ-MC), introduced by Sen et al. [7]. These formulations are extensions of the TSP MTZ formulation, proposed by Miller et al. [9]. The fundamental premise of these benchmark formulations involves considering constant values for both node dwell time and agent traveling speeds in the formulation design. In response to these benchmarks, we introduce the novel formulations TW-NC and TW-MC. In the next section, we offer a comparative analysis between the prposed and benchmark formulations.

A. TSP-NC TW Formulation

This formulation is designed to provide no-contact paths to the agents arriving in the network. Whenever such a no-contact path is not possible, this formulation would lead to an infeasible solution space, leading to the formulation to not converge. This mixed-integer program presents the no-contact path relying on the idea that to prevent contact between two agents, an agent must refrain from reaching a node already occupied by another agent. To achieve this, a set of continuous time variables is introduced, representing the moments when an agent reaches various nodes during their tour. These time variables are constrained to prevent overlap with time windows during which nodes are occupied or blocked by other agents. The blocked time windows for each node are specified as input data, derived from the routes generated by the formulation for agents already within the network, and using the expected value of the node dwell time. The formulation is outlined below.

In the description of the formulations, we consider node 1 as the Entry Gate and node n as the Exit Gate.

CONSTANTS FIXED BEFORE SIMULATION:

M: Large positive real number

b: Expected Node dwell time

v: Expected Agent travelling speed

D_{ij}: Distance between node i and node j

PARAMETERS COLLECTED DURING SIMULATION: $K_i : \{K_{i1}, \ldots, K_{iS_i}\}$ Set of S_i time points indicating arrival time of agents at i^{th} node.

DECISION VARIABLES:

t_i: Time of arrival of current agent at node i

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ comes before } j \text{ (in route of current agent)} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if agent } k \text{ visits node } i \text{ after current agent} \\ 0 & \text{otherwise} \end{cases}$$

OBJECTIVE FUNCTION::

$$\min \ t_n - t_1$$

CONSTRAINT SET - 1, SUB-TOUR ELIMINATION: For all distinct nodes, we impose a constraint ensuring that either one node precedes the other in the agent's path or vice versa. This precedence relation serves to prevent the occurrence of sub-tours in the paths taken by the agents.

$$\begin{aligned} &x_{ij} + x_{ji} = 1 \quad \forall i \neq j, i, j \in [1, n] \\ &x_{1i} = 1 \quad \forall i \in [2, n] \\ &x_{ii} = 0 \quad \forall i \in [1, n] \\ &x_{i,n} = 1 \quad \forall i \in [1, n-1] \end{aligned}$$

CONSTRAINT SET - 2: The following constraints are established to put in place a connection between the continuous time variables and the ordering binary variables. This connection allows these variables to be utilized for analyzing the arrival times of agents at various nodes. Subsequently, these variables are further constrained to prevent overlapping in the time spent on each node.

$$t_j - t_i \geq \left(\left\lceil \frac{D_{ij}}{v} \right\rceil + b \right) x_{ij} - M \left(1 - x_{ij}\right)$$
$$\forall i \in [1, n]; j \in [1, n]; i \neq j$$

$$t_j - t_1 \geq \left\lceil \frac{D_{j1}}{v} \right\rceil, \quad \forall j \in [2, n]$$

CONSTRAINT SET - 3, NO CONTACT CONSTRAINTS: Now, utilizing the continuous time variables updated in the preceding set of constraints, we impose additional constraints to avoid any overlap in the time spent by the agent at the nodes along their paths.

$$t_i + b \leq K_{ik} y_{ik} + M\left(1 - y_{ik}\right) \quad \forall i \in [2, n-1]; k \in [1, S_i]$$
$$t_i - b \geq K_{ik}\left(1 - y_{ik}\right) - M y_{ik} \quad \forall i \in [2, n-1]; k \in [1, S_i]$$

B. TSP-MC TW Formulation

DECISION VARIABLES:

t_i : time of arrival of the current agent at node i

$$x_{ij} : \begin{cases} 1 & \text{if node } i \text{ is visited before } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} : \begin{cases} 1 & \text{if agent } k \text{ visits node } i \text{ after the current agent} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{ik} = \begin{cases} 1 & \text{if contact occurs between the previously} \\ & \quad \text{arrived agent } k \text{ and the agent at node } i \\ 0 & \text{otherwise} \end{cases}$$

OBJECTIVE FUNCTION:

$$\min \; t_n - t_1 + P_e \times \sum_{i=2}^{n-1} \sum_{j=1}^{S_i} \delta_{ij}$$

MINIMAL CONTACT CONSTRAINTS: Instead of the No-Contact constraints in the previous formulation, we describe a formulation here that aims to minimize these contacts between individual agents. This is achieved by introducing indicator variables, which take the value of 1 in the presence of contact between agents at a node and 0 otherwise.

$$t_i \geq [K_{ik} - b]\delta_{i,k} - M\left(1 - \delta_{i,k}\right), \forall i \in [2, n-1]; k \in [1, S_i]$$
$$t_i \leq [K_{ik} + b]\delta_{i,k} + M\left(1 - \delta_{i,k}\right), \forall i \in [2, n-1]; k \in [1, S_i]$$

$$t_i \geq [K_{ik} + b](1 - \delta_{i,k}) - M\left(\delta_{i,k}\right)$$
$$\forall i \in [2, n-1]; k \in [1, S_i]$$

$$t_i \leq [K_{ik} - b](1 - \delta_{i,k}) + M\left(\delta_{i,k}\right)$$
$$+ M(1 - y_{ik}), \forall i \in [2, n-1]; k \in [1, S_i]$$

C. Robust Optimization Formulation

In this specific formulation, we integrate the consideration of uncertainty in both node dwell times and agent traveling speeds through the utilization of an asymmetric box uncertainty approach. This incorporation of uncertainty allows for a more comprehensive and realistic depiction of the stochastic nature inherent in these variables. To gauge the degree of asymmetry within the box uncertainty set, we introduce the parameter α. α serves as an indicator of the asymmetry level present in the uncertainty set for a given stochastic variable. A higher value of α signifies an increased robustness in the model's capacity to handle the randomness associated with that specific stochastic variable. Through the adoption of an asymmetric box uncertainty approach and the introduction of the parameter α, we enhance the model's capability to accommodate and effectively manage uncertainty in node dwell times and agent traveling speeds.

CONSTANTS FIXED BEFORE SIMULATION:

$\overline{b_i}$: Average dwell time on node$i$

$\overline{v_{i,j}}$: Average velocity in going from node i to node j,

b_i: Dwell time on node $i, \forall i \in \{1, \ldots, n\}$,
where $b_i \in [\overline{b_i} - \gamma_i^b, \overline{b_i} + \alpha\gamma_i^b]$

P_c: Penalty applied for contact between customers at a node,

P_t: Penalty applied for time spent inside the system

$v_{i,j}$: Velocity in going from node i to node j,
where $v_{i,j} \in [\overline{v_{i,j}} - \gamma_{i,j}^v, \overline{v_{i,j}} + \alpha\gamma_{i,j}^v]$

OBJECTIVE FUNCTION::

$$\min \; P_t \times (t_n - t_1) + (P_c + \gamma^c) \times \left(\sum_{i=2}^{n-1} \sum_{j=1}^{S_i} \delta_{ij} \right)$$

BOOK KEEPING CONSTRAINTS: The subsequent constraints are introduced to establish a relationship between the continuous time variables and the ordering binary variables, taking into account values for agent traveling speed and node dwell times derived from the box uncertainty set. We incorporate the upper limit for agent speeds and extended node dwell times from the box uncertainty to enhance our model's ability to prevent contacts more effectively. This approach is justified by the understanding that higher speeds and prolonged node dwell times would result in increased contacts, and by incorporating these upper limits, the model is better equipped to manage and minimize such occurrences.

$$t_j - t_i \geq \left(\frac{D_{ij}}{\overline{v_{i,j}} + \alpha\gamma_{i,j}^v} + \overline{b_i} + \alpha\gamma_i^b \right) x_{ij} - M\left(1 - x_{ij}\right)$$
$$\forall i \in [1, n]; j \in [1, n]; i \neq j$$

$$t_j - t_1 \geq \frac{D_{j1}}{\overline{v_{i,j}} + \alpha\gamma_{i,j}^v} \quad \forall j \in [2, n]$$

MINIMAL CONTACT CONSTRAINTS:

$$t_i \geq [K_{ik} - \overline{b_i} - \alpha\gamma_i^b] \times \delta_{i,k} - M\left(1 - \delta_{i,k}\right)$$
$$\forall i \in [2, n-1]; k \in [1, S_i]$$
$$t_i \leq [K_{ik} + \overline{b_i} + \alpha\gamma_i^b] \times \delta_{i,k} + M\left(1 - \delta_{i,k}\right)$$
$$\forall i \in [2, n-1]; k \in [1, S_i]$$
$$t_i \geq [K_{ik} + \overline{b_i} + \alpha\gamma_i^b] \times (1 - \delta_{i,k}) - M\left(\delta_{i,k}\right) - M(y_{ik})$$
$$\forall i \in [2, n-1]; k \in [1, S_i]$$
$$t_i \leq [K_{ik} - \overline{b_i} - \alpha\gamma_i^b] \times (1 - \delta_{i,k}) + M\left(\delta_{i,k}\right) + M(1 - y_{ik})$$
$$\forall i \in [2, n-1]; k \in [1, S_i]$$

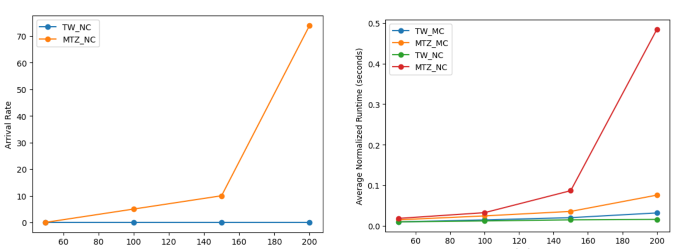
IV. Results and Discussion

The simulation experiments were conducted utilizing the 13th Generation Intel Core i9 processor, operating on a

Windows 11 Home platform, with 16 GB of RAM and a 1 TB Solid State Drive (SSD), complemented by an NVIDIA® GeForce RTX 4070 graphics card, all housed within the Acer Predator Helios 16 Gaming Laptop. The Mixed Integer Linear Programming (MILP) formulations were coded and executed using the Gurobi optimizer [20], for the simulation experiments which discuss.

A. Comparing MTZ MC, MTZ NC, TW MC, TW NC formulations in the Deterministic Scenario

In this section, we evaluate the performance of the MTZ MC, MTZ NC, TW MC, and TW NC formulations under deterministic conditions, where each agent travels at a constant speed and spends an identical amount of time at each node, consistent with the average values of these quantities. We observe instances of infeasibility in the MTZ NC formulation, and these instances increase with the rise in the arrival rate. Infeasibilities represent scenarios where the feasible region becomes a null space, a situation encountered in the MTZ NC formulation when an arriving agent within the network struggles to identify a path that avoids contact with other agents. This limitation arises from the central supervisor's inability, when utilizing the MTZ NC formulation for path assignment, to accommodate an agent waiting at a node. This constraint hinders the assignment of routes in a way that includes the agent waiting on their node long enough to prevent contact with other agents, leading to the observed infeasibilities. In contrast, TW NC formulations allow for additional stay on nodes, preventing infeasibilities. Results in Figure 2a show MTZ NC formulation experiencing increased infeasibilities with rising arrival rates, while TW NC formulation remains free of infeasibilities. This difference difference between MTZ NC and TW NC formulations can be seen in Constraint Set 3 of MTZ NC and Constraint Set 2 of TW NC formulations. Next, we proceed to compare the average normalized runtimes of these implemented formulations, which are normalized with respect to the number of nodes visited by an agent. The corresponding experimental outcomes are shown in Figure 2b. It is observed that the Time Windows (TW) formulations exhibit markedly lower runtimes.



(a) Comparing Infeasibilities for MTZ NC and TW NC formulations

(b) Comparing Runtimes for MTZ NC, TW NC, MTZ MC, and TW MC formulations

Fig. 2: Comparing the TW and MTZ formulations based on runtimes and infeasibilities

When we compare the Minimal Contact Formulations, i.e., formulations C and D, we try to understand which among MTZ and TW formulations is better at minimizing the contacts. As shown in Figure 3, we observe that the TW formulation leads to a lesser number of contacts, except in the case when there is no penalty for the contacts. Like the prevention of infeasibilities, this too can be attributed to the flexibility of TSP TW MC formulation, where TW formulation is able to reduce the number of contacts by allowing the agent to wait for a small duration of time before moving to the next node. In Figure 4, we observe how both the TW MC and MTZ MC formulations lead to similar normalized node dwell times, which is the average time spent by the agent in the network divided by the total number of nodes visited by the agent. We find these values to be similar to each other numerically.

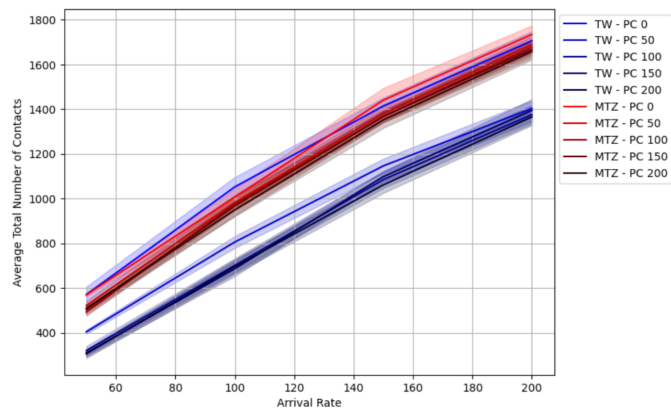
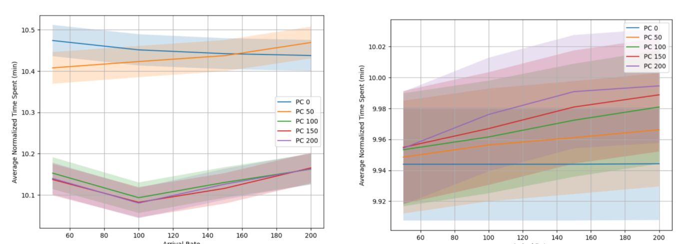


Fig. 3: Comparison of the total number of contacts under no randomness (along with 95% confidence intervals) for TW and MTZ formulations with varied arrival rates (m/s) [50, 100, 150, 200] and penalty per contact (PC) [0, 50, 100, 150, 200]



(a) TW MC Formulation

(b) MTZ MC Formulation

Fig. 4: Comparing Average Normalized Node Dwell times for the Minimal Contact Routing formulations

B. Considering Randomness

In the stochastic setting, we present the results obtained from stochastic simulation experiments to discern which formulations, among MTZ MC and TW MC, result in fewer contacts. The standard deviation (SD) assigned to the normal distribution governing agent traveling speeds is varied, specifically set at 20% and then 60% of the mean value. Notably, the case with a higher standard deviation correlates with a lower number of contacts. This outcome is attributed to increased variability in traveling speeds, reducing the likelihood of overlap at nodes among agents. To elaborate, when sampling agent speeds, the absolute value of a random variable drawn from a normal distribution with parameters $Normal(15, \sigma^2)$ is taken. A higher σ corresponds to a greater probability of obtaining traveling speeds on the left side of the mean, closer to zero. Consequently, smaller traveling speeds translate to longer traveling times, preventing collisions at nodes by allowing for delayed arrivals, particularly for slower-traveling agents. The results for this experiment can be seen in Figure 5.

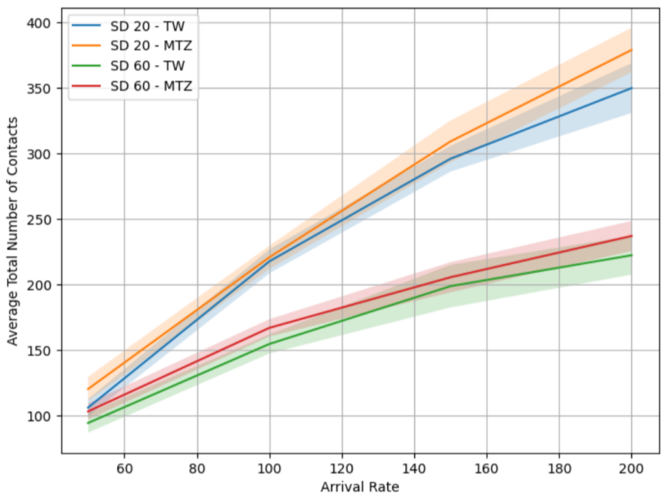
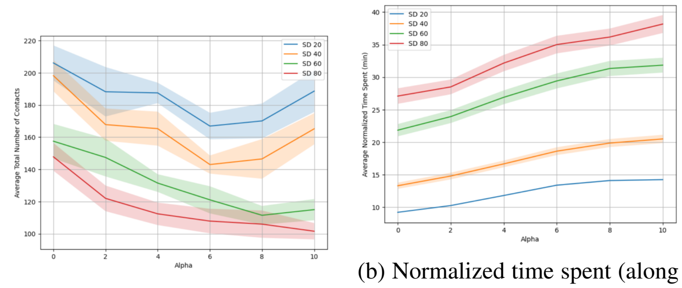


Fig. 5: Comparison of the total number of contacts under randomness (along with 95% confidence intervals) for TW and MTZ formulations with varied arrival rates (m/s) [50, 100, 150, 200] and penalty per contact (PC) = 100

Now, we use the robust version of the TSP MC TW Formulation with an asymmetric box uncertainty set to mitigate contacts in the stochastic setting. As discussed before, the parameter α (Alpha) controls the asymmetry of the box uncertainty set used in the robust formulation. From the simulation results, averaged across ten random seeds and depicted in Fig.6, we observe a decrease in the total number of contacts as α increases, with the minimum contacts observed for α values between 8 and 10. This trend is attributed to the higher values of α, which account for increased randomness in traveling time and node dwell time within the system. Consequently, the formulation strategically assigns paths to agents to minimize contacts in the presence of uncertainty, while it results in longer lengths of stay in the network. Longer paths are assigned to delay an agent's arrival at nodes shared with other agents. This rationale explains the observed behavior in Fig.6, where a statistically significant decrease in contacts is evident, albeit with an associated increase in the time agents spend in the system as α increases. Moreover, to gain a deeper understanding of the behavior of these formulations, we explore a scenario where we possess precise information about all the future node arrival times of the agents, and integrate this information into the formulation. The results of this scenario are presented in Fig. 7, revealing that having such information offers marginal to no discernible benefit in reducing the total number of contacts. This observation raises a pertinent question regarding the practicality and commercial viability of investing in sensors to monitor the exact node

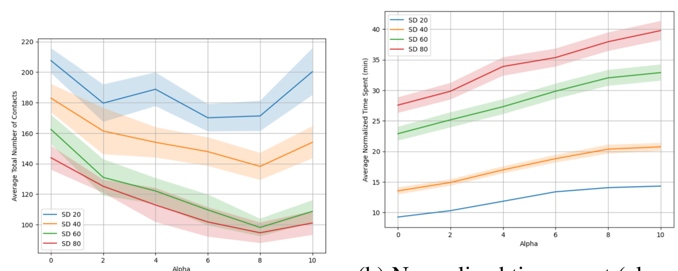


(a) Variation of contacts with Alpha (along with 95% confidence interval)

(b) Normalized time spent (along with 95% confidence intervals) on each node in the scenario without randomness)

Fig. 6: Results for Robust Optimization based Formulation, considering exponential node dwell time

arrival times of agents, such as Automated Guided Vehicles (AGVs) in nodes (stations within warehouses). Our simulation experiments suggest a negligible benefit, highlighting the limited insights that exact node arrival times provide in terms of understanding the duration of stay of an agent at a given node.



(a) Variation of contacts with Alpha (along with 95% confidence interval)

(b) Normalized time spent (along with 95% confidence intervals) on each node in the scenario without randomness)

Fig. 7: Results for Robust Optimization based Formulation when exact future node arrival times are known by the formulations, considering exponential node dwell time

To check whether our results hold when we choose distributions other than the exponential distribution for the node dwell time, we change the distribution of the node dwell times to normal, triangular, gamma, and exponential distributions. We take the absolute values of the samples generated from these distributions as the node dwell time. The parameters of these distributions, as also highlighted in Fig. 8, are chosen such that ninety percent of the samples generated are between $x = 1$ and $x = 5$. The differences in the total number of contacts between these experiments can be attributed to the difference in the left and right tails of these distributions. We see a high overlap between the normal and the triangular distribution cases, and hence observe similar total number of contacts and normalized time spent trends. Here too, as seen in Fig. 9 and Fig. 10, we observe that, with the increase in α, there is a decrease in the total number of contacts, with least contacts observed for α between 8 and 10. We also see that there is an increase in the normalized time spent per node as the α increases, the reasoning for which is the same as described earlier. As before, here too in in Fig. 11 and Fig. 12, we observe negligible to no benefit of knowing the exact node arrival times of the agents. This is because the exact node arrival times don't capture much information regarding the amount of time the agent spends on that node, hence not being very useful in terms of reducing contacts.

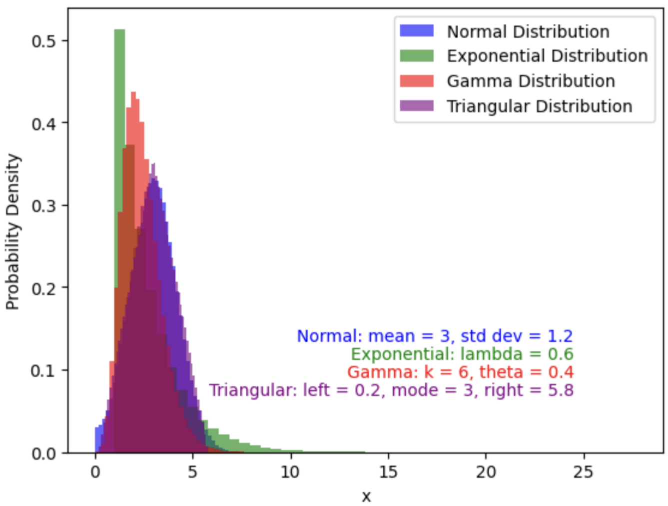
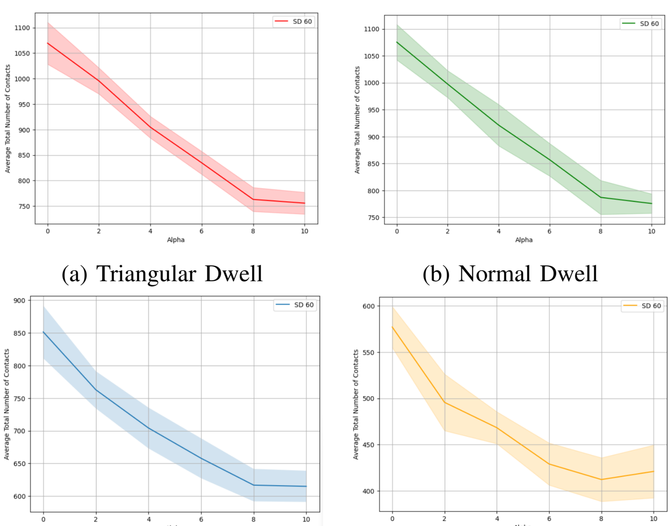


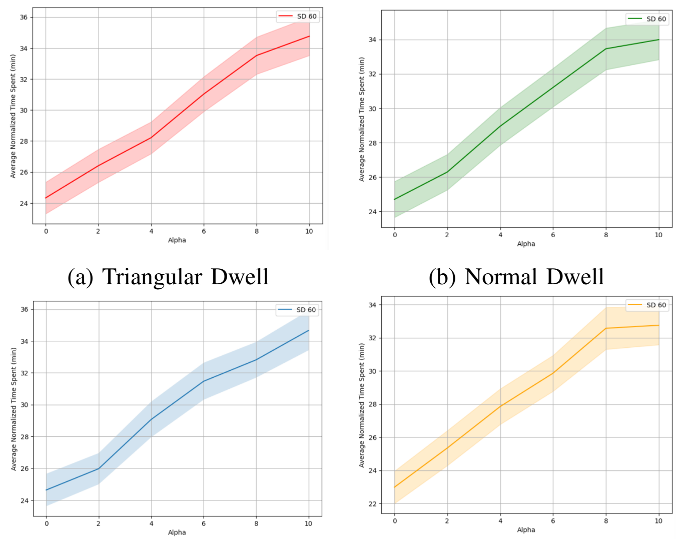
Fig. 8: Distribution of Node Dwell Times when Triangular, Normal, Gamma, and Exponential Dwell times are considered



(a) Triangular Dwell (b) Normal Dwell

(c) Gamma Dwell (d) Exponential Dwell

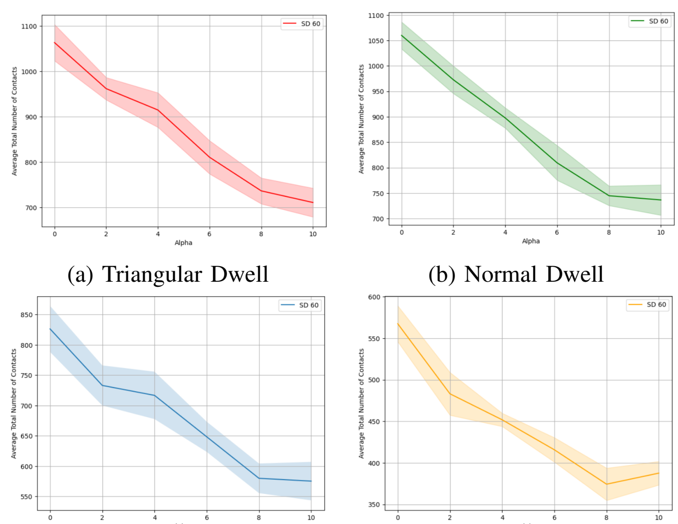
Fig. 9: Variation of Average Total Number of Contacts with Alpha for Robust Formulation



(a) Triangular Dwell (b) Normal Dwell

(c) Gamma Dwell (d) Exponential Dwell

Fig. 10: Variation of Normalized Time Spent per Node with Alpha for Robust Formulation



(a) Triangular Dwell (b) Normal Dwell

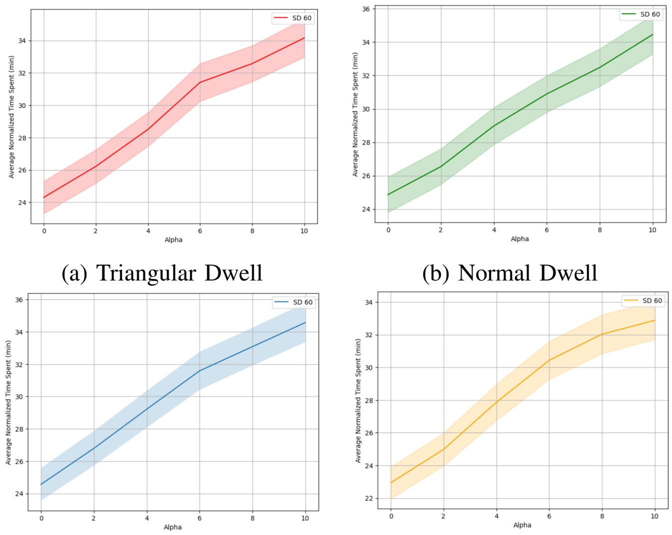
(c) Gamma Dwell (d) Exponential Dwell

Fig. 11: Variation of Average Total Number of Contacts with Alpha for Robust Formulation, when exact future node arrival times are known

V. Conclusion

In conclusion, our study reveals that the Time-Windows (TW) formulation outperforms the Miller-Tucker-Zemlin (MTZ) formulation in terms of reducing contacts and achieving shorter normalized runtimes. This superiority is consistent across experiments involving fixed node dwell times and agent arrival rates. The TW formulation, providing agents the flexibility to choose between waiting on a node or selecting alternative paths, results in fewer contacts within the network. Furthermore, this flexibility leads to the TW NC formulation to eliminate infeasibilities, whereas MTZ NC formulation led to infeasibilities, which increased with the agent arrival rate.

When introducing randomness in node dwell times and agent traveling speeds, both formulations exhibit similar contact counts. Although the TW formulation tends to yield fewer contacts on average, this difference lacks statistical significance at the 5% level. The application of the Robust Optimization Formulation leads to a statistically significant reduction in contacts compared to the TW MC and TW NC



(a) Triangular Dwell (b) Normal Dwell

(c) Gamma Dwell (d) Exponential Dwell

Fig. 12: Variation of Normalized Time Spent per Node with Alpha for Robust Formulation, when exact future node arrival times are known

formulations.

Furthermore, we explore scenarios where precise future node arrival times of agents are known and integrated into the formulation. Surprisingly, we find that possessing this information offers marginal to no benefit in terms of reducing total contacts. This observation prompts questions about the practicality and cost-effectiveness of investing in sensors to monitor exact node arrival times of agents, especially in systems resembling our simulation experiments.

Appendix A
The Benchmark MTZ formulations

To set a reference point, we examine two formulations labeled as Appendix Formulation 1 and Appendix Formulation 2, put forth by Sen et al. (2021). These formulations primarily tackle the deterministic scenario, presuming a scenario where traveling speeds and node dwell times are fixed. They draw inspiration from the MTZ formulation for the traveling salesman problem, introducing integer decision variables to independently address the elimination of sub-tours.

Appendix Formulation 1: TSP-NC MTZ Formulation

PARAMETERS FOR SIMULATION:

M: Large positive real number

b: Expected node dwell time

v: Expected Agent travelling speed

D_{ij}: Distance between node i and node j

PARAMETERS COLLECTED DURING SIMULATION: $K_i : \{K_{i1}, \ldots, K_{iS_i}\}$ Set of S_i time points indicating arrival time of agents at i^{th} node.

DECISION VARIABLES:

t_i: Arrival time of agent at node i

u_i: Order in which node i is visited

$$x_{ij} = \begin{cases} 1 & \text{if node } i \text{ is visited immediately before node } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if agent } k \text{ visits node } i \text{ after the current agent} \\ 0 & \text{otherwise} \end{cases}$$

OBJECTIVE FUNCTION (To minimize)::

$$\sum_{i=1}^{n+1} \sum_{j=1}^{n+1} D_{ij} \times x_{ij}$$

CONSTRAINT SET - 1:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in [2, n]$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \in [1, n-1]$$

CONSTRAINT SET - 2, SUB-TOUR ELIMINATION CONSTRAINTS FOR TSP:

$$u_1 = 1$$
$$u_n = n$$
$$u_i \geq 2 \quad \forall i \in [2, n-1]$$
$$u_i \leq n-1 \quad \forall i \in [2, n-1]$$
$$u_i - u_j + n \times x_{ij} \leq n-1 \quad \forall i \in [1, n-1], \forall j \in [2, n]$$

CONSTRAINT SET - 3, CONSTRAINTS FOR KEEPING TRACK OF TIME:

$$t_0 = 0$$

$$t_j - t_i \leq M(1 - x_{ij}) + b + \left\lceil \frac{D_{ij}}{v} \right\rceil \forall i \in [1, n-1]; j \in [2, n]$$

$$t_j - t_i \leq -M(1 - x_{ij}) + b + \left\lceil \frac{D_{ij}}{v} \right\rceil \forall i \in [1, n-1]; j \in [2, n]$$

CONSTRAINT SET - 4, NO CONTACT CONSTRAINTS:

$$t_i + b \leq K_{ik} y_{ik} + M(1 - y_{ik}) \forall i \in [2, n]; k \in [1, S_i]$$
$$t_i - b \geq K_{ik}(1 - y_{ik}) - M y_{ik} \forall i \in [2, n]; k \in [1, S_i]$$

Appendix Formulation 2: TSP-MC MTZ Formulation

DECISION VARIABLES:

t_i: Arrival time of agent at node i

u_i: Order in which node i is visited

$$x_{ij} = \begin{cases} 1 & \text{if node } i \text{ is visited immediately before node } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if agent } k \text{ visits node } i \text{ after the current agent} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{ik} = \begin{cases} 1 & \text{if contact occurs between the previously} \\ & \quad \text{arrived agent } k \text{ and the agent at node } i \\ 0 & \text{otherwise} \end{cases}$$

OBJECTIVE FUNCTION (To minimize)::

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \times x_{ij} + P_e \times \sum_{i=2}^{n-1}\sum_{j=1}^{S_i} \delta_{ij}$$

CONSTRAINT SET - 1:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in [2, n]$$
$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \in [1, n-1]$$

CONSTRAINT SET - 2, SUB-TOUR ELIMINATION CONSTRAINTS FOR TSP:

$$u_1 = 1$$
$$u_n = n$$
$$u_i \geq 2 \quad \forall i \in [2, n-1]$$
$$u_i \leq n-1 \quad \forall i \in [2, n-1]$$
$$u_i - u_j + (n)x_{ij} \leq n-1 \quad \forall i \in [1, n-1], \quad \forall j \in [2, n]$$

CONSTRAINT SET - 3, CONSTRAINTS FOR KEEPING TRACK OF TIME:

$$t_1 = 0$$
$$t_j - t_i \leq M(1 - x_{ij}) + b + \left[\frac{D_{ij}}{v}\right] \forall i \in [1, n-1]; j \in [2, n]$$
$$t_j - t_i \geq -M(1 - x_{ij}) + b + \left[\frac{D_{ij}}{v}\right] \forall i \in [1, n-1]; j \in [2, n]$$

CONSTRAINT SET - 4, MINIMAL CONTACT CONSTRAINTS:

$$t_i \geq [K_{ik} - b]\delta_{i,k} - M(1 - \delta_{i,k}), \forall i \in [2, n]; k \in [1, S_i]$$
$$t_i \leq [K_{ik} + b]\delta_{i,k} + M(1 - \delta_{i,k}), \forall i \in [2, n]; k \in [1, S_i]$$
$$t_i \geq [K_{ik} + b](1 - \delta_{i,k}) - M(\delta_{i,k}) - M(y_{ik})$$
$$\forall i \in [2, n]; k \in [1, S_i]$$

$$t_i \leq [K_{ik} - b](1 - \delta_{i,k}) + M(\delta_{i,k}) + M(1 - y_{ik})$$
$$\forall i \in [2, n]; k \in [1, S_i]$$

References

[1] Applegate, D. L., Bixby, R. E., Chvatal, V., and Cook, W. J. (2006). The traveling salesman problem: A computational study. Princeton University Press.

[2] Ben-Tal, A., and Nemirovski, A. (2002). Robust optimization – methodology and applications. Math. Program. 92, 453–480.

[3] Bullo, F., Frazzoli, E., Pavone, M., Savla, K., and Smith, S. L. (2011). Dynamic vehicle routing for robotic systems. Proceedings of the IEEE, 99(9), 1482–1504.

[4] Cook, W. J. (2011). In pursuit of the traveling salesman: Mathematics at the limits of computation. Princeton University Press.

[5] Duinkerken, M. B., Ottjes, J. A., and Lodewijks, G. (2006). Comparison of routing strategies for AGV systems using simulation. In Proceedings of the 2006 winter simulation conference (pp. 1523–1530).

[6] Dumas, Y., Desrosiers, J., Gelinas, E., and Solomon, M. M. (1995). An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. Operations Research, 43(2), 367-371.

[7] Guillaume, D., Michael, D., Abdelghani, B., Damien, T., and Mohamed, D. (2017). Decentralized motion planning and scheduling of AGVs in FMS. Transactions on Industrial Informatics.

[8] Gurobi Optimization, L. (2021). Gurobi optimizer reference manual. Retrieved from http://www.gurobi.com

[9] Herrero-Perez, D., and Martinez-Barbera, H. (2010). Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems. IEEE Transactions on Industrial Informatics, 6(2), 166–180.

[10] Korsah, G. A., Stentz, A., and Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. The International Journal of Robotics Research, 32(12), 1495–1512.

[11] Kwang-Yeon Kim, Jung-Woo Park, and Min-Jea Tahk (2007). UAV collision avoidance using probabilistic method in 3-D. International Conference on Control, Automation and Systems, Seoul, pp. 826-829.

[12] Lee, D.-H., Zaheer, S. A., and Kim, J.-H. (2014). A resource-oriented, decentralized auction algorithm for multirobot task allocation. IEEE Transactions on Automation Science and Engineering, 12(4), 1469–1481.

[13] Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. Journal of the ACM, 7(4), 326–329.

[14] Miyamoto, T., and Inoue, K. (2016). Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. Computers and Industrial Engineering, 91, 1–9.

[15] Reinelt, G. (1991). TSPLIB—a traveling salesman problem library. ORSA Journal on Computing, 3(4), 376–384.

[16] Sen, D., Ramamohan, V., and Ramamoorthy, P. (2021). Optimal Minimal-Contact Routing of Randomly Arriving Agents Through Connected Networks. In Proceedings of the Winter Simulation Conference (WSC), Phoenix, AZ, USA, 2021 (pp. 1-12).

[17] Shi, Z., and Ng, W. K. (2018). A collision-free path planning algorithm for unmanned aerial vehicle delivery. In 2018 international conference on unmanned aircraft systems (icuas) (pp. 358–362).

[18] Spensieri, D., Carlson, J. S., Ekstedt, F., and Bohlin, R. (2015). An iterative approach for collision-free routing and scheduling in multirobot stations. IEEE Transactions on Automation Science and Engineering, 13(2), 950–962.

[19] Toth, P., and Vigo, D. (2014). Vehicle routing: Problems, methods, and applications. SIAM.

[20] Xin, J., Meng, C., Schulte, F., Peng, J., Liu, Y., and Negenborn, R. R. (2020). A time-space network model for collision-free routing of planar motions in a multirobot station. IEEE Transactions on Industrial Informatics, 16(10), 6413–6422.

[21] Xin, J., Negenborn, R. R., Corman, F., and Lodewijks, G. (2015). Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. Transportation Research Part C: Emerging Technologies, 60, 377–396.