An MPC scheme for traffic coordination in open and irreversible, zone-controlled, guidepath-based transport systems

Spyros Reveliotis

***Abstract*—Zone-controlled, guidepath-based transport systems (ZC-GBTS) is a modeling abstraction that has been used extensively for the modeling of the safe interaction of a number of agents that circulate in a constricted medium. The traffic scheduling problem in these transport systems is very hard, and in some recent work of ours we have proposed a Model Predictive Control (MPC) scheme for simplifying this problem. The detailed implementation of this MPC scheme depends on certain structural and operational properties of the underlying ZC-GBTS. In this work, we detail the aforementioned MPC scheme for a ZC-GBTS sub-class that is characterized as "open and irreversible"; the presented results leverage some earlier similar developments of ours for the sub-class of "open and reversible" ZC-GBTS.**

***Note to Practitioners* – Open and irreversible, zone-controlled, guidepath-based transport systems is a natural abstraction of the traffic dynamics that take place in many unit-load material handling systems (MHS), like the automated guided vehicle (AGV) systems that are used in various production and logistics environments, and the overhead monorail systems that are used in most semiconductor manufacturing facilities. In these environments, vehicles are circulating in a "guidepath network" that is defined either by the physical structure of the employed MHS (as in the case of the overhead monorail systems) or more artificially, in an effort to isolate the traffic of these vehicles from the surrounding environment (as in the case of the AGV systems). Furthermore, in order to ensure safe and collision-free motion for the traveling vehicles, the various edges of this guidepath network are further divided into zones, and it is stipulated that each zone can be allocated to at most one agent at any time. This restriction renders the considered transport systems susceptible to deadlock, and therefore, their traffic controller must control the generated traffic for time-based performance considerations, like the maximization of the system throughput or the minimization of the experienced delays, but also for ensuring traffic liveness, i.e., the ability of every vehicle to complete successfully its current assignment and engage in similar assignments in the future. The resulting problem is very complex, and the current manuscript provides a complete and computationally efficient solution to it.**

Keywords: Guidepath-based transport systems; traffic liveness enforcement; deadlock avoidance; discrete event systems, model predictive control

S. Reveliotis is with the School of Industrial & Systems Engineering, Georgia Institute of Technology, email: spyros@isye.gatech.edu. He was partially supported by NSF grant ECCS-1707695.

I. Introduction

Zone-controlled, guidepath-based transport systems – to be referred to as ZC-GBTS in the following – is a modeling abstraction that has been used extensively for the modeling of the safe interaction of a number of agents that circulate in a constricted medium. Within the Robotics & Automation community, the most prominent application of these models concerns the modeling, analysis and control of the traffic dynamics that are generated by the automated unit-load material handling systems (MHS) that are used in various production and distribution facilities [1], [2]. But similar models have been used in the programming of the animation that is supported by modern video games [3], and for the coordination of the qubit circulation in quantum computing [4]. In all these operational contexts, the system agents that model the various circulating entities, execute concurrently some dynamically (re-)defined "mission trips" that request them to visit a number of locations of the underlying guidepath network in a specified order. Furthermore, due to safety considerations, these mission trips must be executed in a way that observes some "separation" requirements for the traveling agents. Finally, an additional typical requirement is that the agent mission trips must be executed in a way that optimizes some time-based performance objective, like the minimization of the required time for meeting all the imposed visitation requirements by all agents.

The resulting traffic scheduling problem is very hard. In fact, to the best of our knowledge, almost all of the past literature has addressed only a simpler version of this traffic scheduling problem, where each traveling agent must be moved from its current location to a single destination (which is distinct for each agent). And even this problem version has been shown to be NP-Hard [5], [6]. Some representative results on this simplified version of the considered problem that have been shown to perform well in the face of this high complexity, and define the corresponding state-of-art in the current literature, can be found in [7], [8], [4]. Furthermore, the recent publication of [4] provides a very extensive and systematic coverage of this entire literature.

In view of the situation that is described in the previous paragraph, [4] also proposed a *Model Predictive Control (MPC)* [9] scheme that seeks to simplify the original traffic scheduling problem that was described in the opening paragraph of this document, by decomposing it into a sequence of subproblems. These subproblems are formulated and solved

iteratively, every time that a traveling agent has reached its next destination, and they seek to transfer the traveling agents from their current locations to their most immediate destinations according to the sequences of the visitation requirements that define their remaining "mission trips", while minimizing the required transfer time.[1] Hence, each of these subproblems falls within the framework of the corresponding traffic scheduling problems that have been addressed by the existing literature, and it can be solved by applying and/or adapting the corresponding methodology that is currently available for those simpler problems.

However, an important, novel concern that arises in this decomposing setting, is the preservation of the *"liveness"* of the overall traffic; i.e., the preservation of the ability of all traveling agents to complete successfully their current mission trips and engage in other mission trips in the future, in spite of the myopic nature of the solutions that are effected by each of the aforementioned subproblems. In the context of the presented MPC framework, this "traffic-liveness" requirement is formally expressed by the requirement for well-posedness and solvability of the various subproblems that are formulated by this framework.

Some past investigations pertaining to various notions of "liveness" for the ZC-GBTS addressed in this work, and the corresponding liveness-enforcing supervisory control problem, can be found in [10], [11], [12], [13], [14], [15]. But these developments have not connected explicitly the "traffic liveness" problem that is addressed in them to the traffic scheduling formulations that also address the time-based performance optimization of the underlying transport system. Furthermore, the recent work of [16] has shown that the "liveness preservation" problem that arises in the aforementioned MPC scheme for the considered class of ZC-GBTS, depends substantially upon certain structural and operational characteristics of the underlying transport system. More specifically, two particular elements that have been shown to be critical in determining the solution of this "liveness preservation" problem are: (i) the presence of a "home" location in the guidepath network that can accommodate all traveling agents; and (ii) the ability of the traveling agents to reverse the direction of their motion in their current edge. ZC-GBTS that possess the aforementioned "home" location have been characterized as *"open"* in the corresponding literature, while the remaining ones are said to be *"closed"*. Also, ZC-GBTS where the agents can reverse the direction of their motion in their current edge, are characterized as *"reversible"*, while the remaining ones are said to be *"irreversible"*.

A complete implementation of the considered MPC scheme for *open and reversible* ZC-GBTS has been provided in [4]. In this particular case, the issue of preserving traffic liveness is automatically resolved by the fact that in open and reversible ZC-GBTS, the state space that traces the agent distribution to the edges of the guidepath network is strongly connected; i.e., it is always possible to reach any desired placement of the system agents on the edges of the underlying guidepath network from any given initial placement. This fact further implies that in the implementation of the considered MPC scheme for open and reversible ZC-GBTS, the formulation and solution of the subproblems involved needs to focus only on the time-based performance optimization of the underlying transport system; i.e., each of these subproblems must only provide a set of feasible and non-conflicting routes for the system agents that will take them from their current locations in the guidepath network to their next immediate destinations while minimizing the corresponding makespan.

The main purpose of this work is to extend the MPC implementation of [4] to the *open and irreversible* case. As we explain in later parts of this paper, the sought extension is nontrivial because of deadlocking effects that arise in this new case, and compromise the strong liveness properties of its reversible counterpart. The presence of these potential deadlocks requires the restriction of the underlying traffic, and the selection of the target states for the various scheduling subproblems to be formulated by the considered MPC framework, in certain subspaces of the corresponding state space from which these deadlocks are effectively and efficiently avoidable. We provide the relevant characterizations of the target subspaces and the necessary control logic.

More specifically, the rest of the paper proceeds as follows: In the next section, we define formally the considered class of transport systems and demonstrate the practical relevance of this definition by highlighting its straightforward applicability to the operational context of the unit-load MHS that are used in various production and distribution environments. Section III provides a formal characterization of the traffic scheduling problem that arises in the considered transport systems, and it outlines the MPC framework developed in [4] for the solution of the corresponding scheduling problem defined in the context of the open but reversible ZC-GBTS. Furthermore, the last part of this section motivates the remaining part of the paper by pointing out the new challenges that arise from the presumed irreversibility of the agent motion in the ZC-GBTS under consideration. Section IV introduces a set of results that concern the notion of liveness in open and irreversible guidepath-based transport systems, and overviews some results from [10], [16] regarding the support of liveness-enforcing supervision for those environments in an effective and computationally efficient manner. On the other hand, Section V capitalizes upon those past results in order to develop the main results of the paper, i.e., the adaptation of the MPC scheme of [4] to the considered ZC-GBTS class, in a way that ensures the liveness of the underlying traffic and remains computationally tractable. Finally, Section VI concludes the paper and outlines some directions for future work.

Closing this introductory section, we also notice, for completeness, that a preliminary version of this work has appeared, under the same title, in the proceedings of the 15th IEEE International Conference on Automation Science and Engineering (IEEE CASE 2019). That write up provides a much cruder exposition of the presented material, and it lacks a significant part of the background material, most of the motivational and elucidating examples, and a large part of the closing discussion that are provided in this work.

[1] In the corresponding terminology, this required transfer time is known as the "makespan" of the generated traffic schedule.

II. The considered guidepath-based transport systems

A formal characterization of the structure and the traffic dynamics of the considered transport systems: An instance of the particular sub-class of the guidepath-based transport systems considered in this work can be formally defined by a pair $(\mathcal{A}, G)$, where the elements of this pair denote, respectively, (a) the set of the system vehicles (or "agents") circulating in it, and (b) the guidepath graph $G = (V, E \cup \{h\})$ that is traversed by these agents.

Graph G is assumed to be undirected, connected, and with a minimum vertex degree of 2.[2] The edges $e \in E$ of G model the "zones" of the underlying guidepath network. These edges can be traversed by a traveling agent $a \in \mathcal{A}$ in either direction, and they can hold no more than one agent at a time. On the other hand, edge h models the "home" zone of the guidepath network. This edge is connected to the rest of the guidepath network through a single vertex v_h (i.e., edge h is a self-loop of G), and it can hold an arbitrary number of agents, including those agents that have not initiated or have completed their assigned missions. Finally, in the considered application contexts, it is also natural to assume that two vertices v_1, v_2 of graph G are connected by more than one zones, and therefore, in stricter terms, graph G is actually a multi-graph; but this feature does not impact substantially our subsequent developments, and we shall keep referring to G as a graph in the sequel.

A "mission" trip for an agent $a \in \mathcal{A}$ is defined by a sequence of edges $\Sigma_a = \langle e_i \in E \setminus \{h\} \rangle$ that must be visited by agent a in the specified order. More specifically, the edges e_i in sequence Σ_a should be perceived as successive (although not necessarily neighboring) destinations for agent a, and the agent can follow any feasible walk[3] on guidepath graph G when moving from edge e_i to edge e_{i+1}. Furthermore, "home" edge h is an implicit last edge in every sequence Σ_a, since each agent a that has completed its mission trip, must retire at this location.

While traversing an edge $e \in E$ with $e = \{v_i, v_j\}$, an agent a will have a certain direction of motion that will be indicated by the corresponding ordered pair (v_i, v_j) or (v_j, v_i). Furthermore, the irreversibility of the agent motion that is presumed in this work, stipulates that the agents cannot switch the direction of their motion in the edges that are currently allocated to them; hence, an agent a entering edge $e = \{v_i, v_j\}$ from vertex v_i must leave this edge through vertex v_j, and vice versa.[4]

[2]The imposed requirement of a minimal vertex degree of 2 is necessitated by the presumed irreversibility of the agent motion, since an agent a that reaches a vertex v of degree 1 will deadlock at that vertex.

[3]We remind the reader that a walk in an undirected graph G is a sequence $\langle v_0, e_1, v_1, \ldots, v_{i-1}, e_i, v_i, \ldots, v_{k-1}, e_k, v_k \rangle$ where, for all $i = 1, \ldots, k$, edge e_i is incident upon the vertices v_{i-1} and v_i.

[4]From a more practical standpoint, the irreversibility of the agent motion can arise from the limited inherent capabilities of the traveling vehicles, as is the case with many industrial AGV systems, but also from restrictions that are imposed by the operational environment; as more specific examples for this second case, we can mention the restricted maneuverability of the traveling vehicles in the narrow aisles that support their motion, and other restrictions that are enforced from concerns about the safety of the traveling vehicles and their various payloads [1], [17], [18].

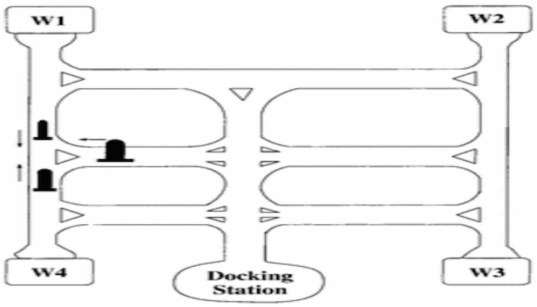

Fig. 1: An abstracting representation of an AGV system and an AGV deadlock.

Finally, since edges $e \in E$ model the zoning scheme that is imposed on the underlying guidepath structure, they are assumed to be of equal length. This assumption, together with the presumed uniformity of the traveling agents $a \in \mathcal{A}$, allow us to further assume that the corresponding edge-traversal times are deterministic and uniform across all edges. This last duration defines a natural *"time unit"* for the considered models, and enables the discretization of the traffic dynamics of the underlying transport system.[5]

In the resulting discretized dynamics of the considered traffic, it is further stipulated that an agent a cannot move in an edge e at time t from a neighboring edge e', unless edge e was empty at time $t-1$. This is a typical assumption for these transport systems that seeks to establish adequate separation among the traveling agents. In more specific terms, this assumption prevents the agent cohabitation on a given edge during the transitional phases that lead from (discrete) epoch $t-1$ to epoch t, and it also implies that two agents cannot "swap" the occupation of two neighboring edges.[6]

AGV systems – a concretizing example of the considered ZC-GBTS: As a concretizing example of the above abstraction of the ZC-GBTS considered in this work, the reader can consider the familiar Automated Guided Vehicle (AGV) systems that are used in various production and distribution facilities [1], [17], [18];[7] a stylized abstraction of such an AGV

[5]Even in the case that the zone-traversal times are non-uniform in terms of the zone set E and/or the agent set $\mathcal{A}$, a discrete-time model for the dynamics of the corresponding traffic can be obtained by utilizing the greatest common divisor of the zone-traversal times, and refining the edge definition for the guidepath network G accordingly.

[6]We also notice, for completeness, that the reusable but exclusive allocation of the zones of the guidepath network to the traveling agents that is presumed in this work, together with all the additional constraints for this allocation that were discussed in the last paragraph, add a "sequential resource allocation" element to the real-time management of the corresponding traffic, besides the allocation of the transport tasks to the system agents and the determination of the corresponding agent routes for the execution of these tasks, that are the primary concerns in the absence of these additional constraints. This "resource allocation" element defines a hard scheduling problem and differentiates substantially the traffic management problems considered in this work from the vehicle routing problems that have been typically studied by the Operations Management literature [19]. In fact, as pointed out in the introductory section, this particular class of scheduling problems has its own distinct presence in the current literature, but the existing literature has focused primarily on the reversible class of ZC-GBTS.

[7]In fact, a similar operational scheme is also implemented by the overhead monorail systems that are used in semiconductor manufacturing facilities [2].

system is depicted in Figure 1. In this case, the traveling agents $a \in \mathcal{A}$ are the system AGVs, which are used to transport materials among the various workstations, and possibly other locations, of the underlying facility. At any point in time, each vehicle might be assigned a sequence of such transport tasks that must be executed in the specified order by visiting the corresponding locations. And the transport-task sequences associated with each vehicle are also dynamically updated as new transport requirements arise in the underlying system.

The guidepath network for these AGV systems is defined either physically (e.g., through some colorful duct tape that is deployed on the shop-floor and must be traced by the vehicle scanners), or virtually (e.g., through some radio signals that must be traced and processed by the vehicle sensors). The exact specification of the system guidepath network, in any of the aforementioned manners, intends to confine the AGV traffic in particular corridors and, in this way, separate it from the remaining activity that takes place in the surrounding environment, due to safety and other efficiency considerations. Finally, the overall layout of this guidepath network also avails of a "docking station" where idling vehicles can retire, and possibly recharge their batteries, receive other maintenance service, etc.

In addition, and in an effort to maintain proper separation (and therefore, prevent collisions) among the traveling vehicles, the various corridors of the guidepath network are split into "zones" that must be occupied by at most one AGV at any point in time; these zones eventually define the edges $e \in E$ of the abstracted guidepath network G. Access of a zone by a traveling vehicle must be authorized by a central traffic controller, and this can happen only if the requested zone is currently empty. Finally, it is also true that in most practical implementations of these AGV systems, vehicles are not expected to back up in their allocated zones, either due to the inherent limitations of their motion dynamics, or due to other safety considerations [1].

Traffic deadlock and the arising need for liveness-enforcing supervision: The motion irreversibility and the other traffic restrictions for the system agents that were defined in the previous parts of this section, when combined with the arbitrary topology of the guidepath graph G and the bidirectional traversal of its edges by the traveling agents, can give rise to deadlocking situations similar to that depicted in Figure 1. More specifically, in the situation that is depicted in Figure 1, each of the three AGVs is blocked in its further advancement by the presence of the other two vehicles that occupy the edges that constitute potential next edges for this particular vehicle.

The development of deadlocks similar to that depicted in Figure 1 will permanently stall the further advancement of the agents involved in it, and it must be proactively prevented by the traffic controller that manages the zone allocation for the traveling agents. The corresponding problem is known as the problem of *liveness-enforcing supervision (LES)* [8] for the considered transport systems, and it is at the core of the main results that are presented at the later parts of the paper.

III. The considered traffic scheduling problem and the simplifying MPC framework of [4]

The first part of this section provides a formal characterization of the basic traffic scheduling problem for open ZC-GBTS that is addressed in this work. The second part of the section introduces the MPC framework that was developed in [4] for the particular instantiation of this scheduling problem when the underlying ZC-GBTS is open and reversible. Finally, the last part of the section discusses the particular challenges for extending the results of [4] to the case of open and *irreversible* ZC-GBTS, and, thus, it motivates the subsequent developments of the paper.

A. The considered traffic scheduling problem

In view of the structural and operational characterization of the considered ZC-GBTS that was provided in the previous section, the traffic scheduling problem that is addressed in this work can be formalized through the following definitions:

Definition 1: A *route* $\mathcal{R}_a$, for any given agent $a \in \mathcal{A}$, over a (discrete) timespan T, is a sequence $\langle (v_i, v_j)_a^t : t = 0, \ldots, T \rangle$ defining, for each period $t \in \{0, \ldots, T\}$, the edge of the guidepath network G that is occupied by agent a during that period, and the agent orientation (or direction of its motion) in this edge. Furthermore, edge $(v_i, v_j)_a^0$ must correspond to the initial zone and orientation of agent a, while any pair of edges $(v_i, v_j)_a^t$, $(v_i', v_j')_a^{t+1}$, $t = 0, 1, \ldots, T-1$, must be neighboring according to the topology of the underlying guidepath network G, and compatible with the presumed reversibility properties of the motion of agent a within its allocated zones.[9] □

Definition 2: A set of routes $\{\mathcal{R}_a\colon a \in \mathcal{A}\}$ is a *feasible routing schedule* $\mathcal{S}$ for agents $a \in \mathcal{A}$ if and only if (*iff*) (i) they satisfy the visitation requirements Σ_a for all agents $a \in \mathcal{A}$ and eventually bring these agents to the "home" zone h, and (ii) they further observe the requirement that an agent a can enter a zone represented by edge e at period t only if this zone was empty at period $t-1$. □

Definition 3: For any given feasible routing schedule $\mathcal{S}$, the maximal traveling time required to meet all the posed visitation requirements across all the corresponding routes $\mathcal{R}_a$, $a \in \mathcal{A}$, and bring all agents back to the "home" zone h, is characterized as the *makespan* of this schedule, and it will be denoted by $T^{\mathcal{S}}$.

Furthermore, a feasible routing schedule $\mathcal{S}$ is *optimal iff* it possesses the minimal makespan across all feasible schedules. □

In the following, we shall denote an optimal routing schedule by $\mathcal{S}^*$ and the corresponding makespan by T^*. Some additional definitions that are very useful in the organization of the subsequent developments, and in the presentation of the derived results, concern the notion of "state" of the considered ZC-GBTS.

[8] In the following, the acronym LES will imply either "liveness-enforcing supervision" or "liveness-enforcing supervisor", depending on the corresponding context of its usage.

[9] We also emphasize that the provided definition of route $\mathcal{R}_a$ allows that $(v_i, v_j)_a^t = (v_i', v_j')_a^{t+1}$, for any $t = 0, 1, \ldots, T-1$; i.e., an agent a can stay idle in its current zone $(v_i, v_j)_a^t$ during some period t.

Definition 4: For the needs of the subsequent developments, the *state* $s(t)$ of any open and reversible ZC-GBTS at some period t is defined by the distribution of the agents $a \in \mathcal{A}$ to the zones $E \cup \{h\}$ of the underlying guidepath network G.

In the case of open and irreversible ZC-GBTS, the above notion of "state" must be augmented with the information about the direction of the agents $a \in \mathcal{A}$ that are located in some zone $e \in E$.

It is further assumed that any given state s is *valid*, i.e., every edge $e \in E$ is allocated to no more than one agent in state s.

Finally, we define the *"home" state* s_h to be the state where every agent $a \in \mathcal{A}$ is located in the "home" zone h. □

In the following, we shall denote the entire set of valid traffic states s by S. Also, we shall use the notation $\gamma(a; s)$ to denote the zone of agent $a \in \mathcal{A}$ in state s.[10] Furthermore, for irreversible ZC-GBTS, the state s can be graphically represented by a *labeled partially directed digraph (PDG)*, $\hat{G}(s)$, that is induced from the original undirected graph G by (i) labeling each zone $e \in E$ that is allocated to some agent $a \in \mathcal{A}$ by the name of the corresponding agent, and (ii) turning edge e into a directed edge with its sense of direction indicating the direction of motion of agent a in the corresponding zone. Finally, for any given period t, state $s(t+1)$ is obtained from state $s(t)$ by advancing a subset of agents $\hat{\mathcal{A}} \subseteq \mathcal{A}$ from their current zones, $\gamma(a; s(t))$, to some neighboring zones that are empty in $s(t)$; furthermore, these advancements must ensure the validity of the resulting state s'.

With all the above definitions in place, now we are ready to introduce the basic traffic scheduling problem that is considered in this work.

Definition 5: The basic traffic scheduling problem for the considered ZC-GBTS: Given (i) the current state s_0 of the considered ZC-GBTS, and (ii) the remaining visitation requirements Σ_a for each agent $a \in \mathcal{A}$, determine a min-makespan routing schedule $\mathcal{S}^*$ that will satisfy these visitation requirements and bring all agents back to the "home" zone h. □

In the case of open and reversible GBTS, an optimal schedule $\mathcal{S}^*$, for any given traffic state s_0 and remaining visitation requirements $\Sigma_a,\ a \in \mathcal{A}$, can be obtained, in principle, by formulating and solving a mixed integer program (MIP) [20]. But this MIP formulation becomes intractable even for fairly small instantiations of the considered scheduling problem. As already noticed, in order to cope with these practical computational challenges, [4] has proposed an MPC scheme towards the solution of the aforementioned scheduling problem. The next subsection outlines, briefly, the defining logic of the MPC scheme that was developed in [4], and the main mechanisms that establish the correctness of this logic with respect to the liveness of the generated traffic. On the other hand, Subsection III-C leverages the insights that are provided by the following subsection in order to highlight the new challenges that arise when the agent motion within their designated zones is irreversible. Hence, the material of the next two subsections provides, both, context and motivation for the results that are presented in the rest of this paper.

[10] According to Definition 4, $\gamma(a; s)$ should be understood as a directed edge if the underlying ZC-GBTS is irreversible and agent a is located in a zone other than the "home" zone; otherwise, $\gamma(a; s)$ corresponds to an undirected edge.

1) Construct an initial feasible schedule $\mathcal{S}^{(0)}$ for the considered problem instance.
2) $\mathcal{S} := \mathcal{S}^{(0)}$.
3) $Q := \dot{\mathcal{A}}(\mathcal{S})$.
4) While $(Q \neq \emptyset)$ do
   a) Pick an element $a \in Q$.
   b) Test whether agent a can be used for generating an improving schedule $\hat{\mathcal{S}}$.
   c) If the above test is positive, do
      i) $\mathcal{S} := \hat{\mathcal{S}}$.
      ii) Goto Step 3.
   d) else $Q := Q \setminus \{a\}$.
5) Return $\mathcal{S}$.

Fig. 2: The basic heuristic algorithm of [4] for the solution of the subproblems that are addressed in the corresponding MPC framework that was developed in that work for open and reversible ZC-GBTS.

B. The MPC scheme of [4] for the traffic scheduling problem of open and reversible ZC-GBTS

As stated in the introductory section, the considered MPC framework decomposes the overall scheduling problem to a sequence of subproblems, with each subproblem seeking the effective and efficient routing of the traveling agents towards their *most immediate destinations* in the corresponding zone-sequences $\Sigma_a,\ a \in \mathcal{A}$. More specifically, these subproblems are formulated and solved every time that one of the traveling agents reaches its current destination, and they try to determine a min-makespan schedule that drives the underlying transport system from its current state s to the state s' that is defined by placing each agent $a \in \mathcal{A}$ to its next destination according to its remaining visitation requirements Σ_a.[11]

In [4], the subproblems that are formulated by this MPC scheme, are solved by the heuristic algorithm that is presented in Figure 2. This is essentially a two-phase heuristic algorithm where: (i) The first phase consists of Steps (1) and (2), and constructs an initial feasible routing schedule $\mathcal{S}^{(0)}$. (ii) On the other hand, the second phase consists of Steps (3) and (4), and it constitutes an iterative scheme that seeks the incremental improvement of the generated schedules through a systematic search over a "neighborhood" of feasible schedules that is defined with respect to the current incumbent schedule. Next, we briefly discuss the most salient points of this algorithm.

The construction of the initial schedule $\mathcal{S}^{(0)}$ in the algorithm of Figure 2: In [4], the initial feasible routing

[11] Obviously, the next destination for the particular agent a that has triggered the formulation of the new subproblem by reaching its current destination, has been revised accordingly; for the remaining agents $a' \in \mathcal{A}$, the next destination in the new subproblem remains the same as in the previous subproblem.

schedule for each subproblem that is generated by the corresponding MPC scheme, is obtained through the results that are established by the following two propositions.

Proposition 1: In an open and reversible ZC-GBTS, it is always possible to reach any traffic state $s \in S$ from the "home" state s_h.

A formal proof for the result of Proposition 1 is provided in [4]. This proof employs a notion of "distance" for each zone $e \in E$ from node v_h (i.e., the terminal node of the "home" zone h), which is defined as the smallest number of edges $e' \in E$ that must be traversed in order to reach edge e from node v_h. Then, the gist of the argument that establishes the correctness of Proposition1 is that, when all agents are collected in zone h, it is possible to route the agents $a \in \mathcal{A}$ to their corresponding zones $\gamma(a; s)$ one at a time, giving priority to those agents a that have a destination $\gamma(a; s)$ with the longest distance from the node v_h. Furthermore, a similar type of argument establishes the following proposition of [4].

Proposition 2: In an open and reversible ZC-GBTS, it is always possible to reach the "home" state s_h from any traffic state $s \in S$.

In the case of Proposition 2, a feasible routing schedule is obtained by routing agents $a \in \mathcal{A}$ with $\gamma(a; s) \neq h$ to zone h one at a time, starting with those agents that are located in zones $\gamma(a; s)$ which are closest to node v_h.

Finally, the combination of Propositions 1 and 2 provides the following corollary.

Corollary 1: For an open and reversible ZC-GBTS, the underlying state space S is strongly connected; i.e., there is always a feasible routing schedule leading from any given state $s \in S$ to some other state $s' \in S$.

Corollary 1 establishes the existence of a feasible routing schedule $\mathcal{S}^{(0)}$ for any subproblem that might be generated in the MPC scheme of [4]. Furthermore, the arguments in the proofs of Propositions 1 and 2 also provide the necessary logic for the construction of this initial schedule. As we shall see in the next subsection, these arguments break down in the case of open but irreversible ZC-GBTS. But before turning into this issue, next we briefly outline the implementation of Steps (3) and (4) of the algorithm in Figure 2 in [4]; this part of the algorithm will be reused almost verbatim in the main developments of this paper.

The implementation of the improving step in the algorithm of Figure 2: Given a feasible routing schedule $\mathcal{S}$, the algorithm of Figure 2 tries to find an improving schedule $\hat{\mathcal{S}}$ to this schedule as follows:

The algorithm first determines the set of agents, $\dot{\mathcal{A}}(\mathcal{S})$, that reach their (immediate) destination zones $\gamma(a; s')$ in the current schedule $\mathcal{S}$ at the latest period $\hat{t}$, among all agents $a \in \mathcal{A}$; hence, the agents $a \in \dot{\mathcal{A}}(\mathcal{S})$ determine the makespan of the current schedule $\mathcal{S}$. Next, the algorithm tries to find an agent $a \in \dot{\mathcal{A}}(\mathcal{S})$ that can be rerouted to its destination zone $\gamma(a; s')$ through an alternative route $\mathcal{R}'_a$ that fulfills the following two requirements: (i) $\mathcal{R}'_a$ together with the routes $\mathcal{R}_{a'}$ in the current schedule $\mathcal{S}$ for the remaining agents $a' \in \mathcal{A}$ define a feasible routing schedule $\hat{\mathcal{S}}$ for the considered subproblem. (ii) $\mathcal{R}'_a$ brings agent a to its destination zone $\gamma(a; s')$ faster than the original route $\mathcal{R}_a$ that is employed by schedule $\mathcal{S}$.

As soon as such an agent a is identified, the corresponding schedule $\hat{\mathcal{S}}$ becomes the new incumbent schedule, and the algorithm starts a new iteration that repeats the above logic. Furthermore, the algorithm terminates at the first iteration where such an agent $a \in \dot{\mathcal{A}}(\mathcal{S})$ cannot be identified, returning the current schedule $\mathcal{S}$ as the generated solution.

In order to complete the description of this improving step of the algorithm of Figure 2, we also need to describe the mechanism that is employed by the algorithm in order to check the existence of an improving route $\mathcal{R}'_a$ for any given agent $a \in \dot{\mathcal{A}}(\mathcal{S})$. This search is effected through a dynamic programming ("shortest path"-type [21]) formulation that is defined on a digraph $\mathcal{G}(a; \mathcal{S})$ that encodes compactly all the possible routes that can drive the considered agent a from its current location to its destination edge within $\hat{t} - 1$ periods. More specifically, each node of digraph $\mathcal{G}(a; \mathcal{S})$ represents the placement of agent a on a certain edge e at some period t of the considered timespan $\{0, 1, \ldots, \hat{t} - 1\}$. Furthermore, each node of digraph $\mathcal{G}(a; \mathcal{S})$ is associated with a cost that expresses the number of conflicts that are induced by the corresponding agent placement that is encoded by this node and the currently fixed routes of the remaining agents. Hence, a feasible improving route $\mathcal{R}'_a$ for the considered agent a is any route of zero cost in digraph $\mathcal{G}(a; \mathcal{S})$. The algorithm of [4] selects one of those improving routes $\mathcal{R}'_a$ that brings agent a to its destination zone $\gamma(a; s')$ as soon as possible.

Complexity analysis and empirical evaluation of the algorithm of Figure 2: In [4] it is shown that the initializing phase of the algorithm of Figure 2 has a worst-case computational complexity of $O(|E|^3)$, while the worst-case complexity of the schedule-improving phase is $O(|\mathcal{A}|^3|E|^2\bar{D})$, where $\bar{D}$ denotes the maximal distance between an edge $e \in E$ of the guidepath graph G and the node v_h. Also, [4] provides additional implementations of the improving step of the considered algorithm that can enhance the quality of the derived schedules at an increased computational cost. Finally, extensive numerical experimentation reported in [4] demonstrates that the resulting algorithms can provide very competitive (near-optimal) solutions for the corresponding traffic scheduling problem, even under the presence of a large number (hundreds) of traveling agents and very congested networks (e.g., cases where $|\mathcal{A}|/|E| \approx 1/3$).

C. Extending the algorithm of Figure 2 to the class of open and irreversible ZC-GBTS and the corresponding challenges

In this subsection we discuss which of the developments of [4] that were presented in the previous subsection extend to the case of open and irreversible ZC-GBTS, maybe with some minor modifications, and which of these developments fail to extend to this new class of ZC-GBTS, necessitating, thus, a more extensive modification of the algorithm of Figure 2 in order to be applicable to this new class of transport systems. The identification and the development of these necessary modifications are the main theme of this paper.

We start with the following proposition, which is an adaptation of the result of Proposition 1 to the class of ZC-GBTS.

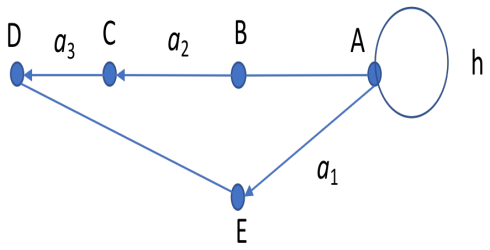

Fig. 3: A counter-example to Proposition 2 for the case of open and irreversible ZC-GBTS.

Proposition 3: Consider an open and irreversible ZC-GBTS at its "home" state s_h, and furthermore, associate with each agent $a \in \mathcal{A}$ a target zone $z(a) \in E \cup \{h\}$ such that

$$z(a_1) = z(a_2) \neq h \implies a_1 = a_2$$

Then, there is a state s reachable from s_h such that each agent $a \in \mathcal{A}$ is located at its target zone $z(a)$. □

The proof of this proposition is very similar to the proof of Proposition 1; i.e., the claimed state s can be obtained by routing agents $a \in \mathcal{A}$ from the "home" zone h to their destination zones $z(a)$ one at a time, while giving priority to those agents with their target zones having the longest distance from node v_h. However, the reader should notice that, in the case of open and irreversible ZC-GBTS, this construction does not guarantee a particular direction of agent a in the corresponding zone $z(a)$, since this direction will be restricted, in general, by the shortest route(s) that can take agent a from zone h to zone $z(a)$; this limitation is also reflected in the statement of Proposition 3.

The reader can also check that, when an initial schedule $\mathcal{S}^{(0)}$ is somehow available, the logic for the improving step in the algorithm of Figure 2 extends almost immediately to the case of the open and irreversible ZC-GBTS. The only necessary modification is that the construction of the digraphs $\mathcal{G}(a;\mathcal{S})$ that are used in the search for improving routes for any agent $a \in \dot{\mathcal{A}}(\mathcal{S})$, must consider only those routes that are compatible with the irreversibility of the agent motion; this can be easily attained by adding also the direction of the agent motion to the informational content of the nodes of the digraph $\mathcal{G}(a;\mathcal{S})$.

On the other hand, the result of Proposition 2 is not extensible to the class of open and irreversible ZC-GBTS. This fact is demonstrated through Figure 3, which also highlights some further problems that arise in this particular case. According to the logic that established the correctness of Proposition 2 for the case of open and reversible ZC-GBTS, the three agents depicted in Figure 3 should be routed to the "home" zone h one at a time, in the sequence $\langle a_1, a_2, a_3 \rangle$, since the corresponding distances of the edges AE, BC and CD from node A are respectively 0, 1 and 2. However, in the presented situation, agents a_1 and a_2 cannot follow the shortest routes that determine the aforementioned distances because of their current orientation in their corresponding zones. Hence, the entire argument that provided Proposition 2 for open and reversible ZC-GBTS, breaks down in the case of open but irreversible ZC-GBTS.

The breakdown of Proposition 2 in the case of open and irreversible ZC-GBTS, further implies the need for an alternative mechanism that will provide the initial routing schedule $\mathcal{S}^{(0)}$ in any potential extension of the algorithm of Figure 2 to this particular ZC-GBTS class.

Even more importantly, Figure 3 demonstrates not only the difficulty of identifying a feasible routing schedule that will lead all traveling agents $a \in \mathcal{A}$ from their current zones in the depicted state s to the "home" zone h, but the potential non-existence of such a routing schedule. Indeed, it is easy to see that, under irreversible agent motion, all three agents depicted in Figure 3 are destined to an unavoidable deadlock. The realization of this additional possibility further implies that we must proactively restrict the operation of any open and irreversible ZC-GBTS so that any traffic state leading to such unavoidable deadlock formations becomes unreachable under the imposed traffic control policies. We provide the necessary control logic in the next two sections.

IV. Live and ordered states in open and irreversible ZC-GBTS

In the last part of the previous section, we saw that, contrary to what is happening in the case of open and reversible ZC-GBTS, the traffic of open and irreversible ZC-GBTS must be proactively controlled in order to ensure the requirement for deadlock freedom and the related requirement for traffic liveness. Hence, in this section we overview a set of definitions and past results that will allow us to provide (i) a formal characterization of the problem of the liveness-enforcing supervision (LES) for the considered class of open and irreversible ZC-GBTS, (ii) the optimal solution of this problem and its computational complexity, and (iii) additional suboptimal solutions to this problem that establish an effective and efficient trade-off between the operational efficiency that is supported by these solutions and their computational tractability. We address each of these three issues in a separate subsection.

A. A formal characterization of traffic liveness for open ZC-GBTS

Traffic liveness in open ZC-GBTS has been studied extensively and systematically in [15], where a formal-linguistic modeling framework has been employed for the representation and the analysis of the qualitative dynamics of the generated traffic. In this subsection, we overview some key findings of that study that are crucial for the needs of this work.

We start this discussion by noticing the under the dynamic and arbitrary nature of the specification of the agent mission trips that was described in Section II, a pertinent operational definition of the notion of "traffic liveness" for the considered class of transport systems is as follows:

Definition 6: The traffic that is generated by an open ZC-GBTS $(\mathcal{A}, G)$ is *live iff* every agent $a \in \mathcal{A}$ retains its ability to reach each edge $e \in E \cup \{h\}$ of G *ad infinitum*. □

In [15] it is shown that the liveness condition of Definition 6 is equivalent to the condition that is stated in the following proposition.

Proposition 4: An open ZC-GBTS is live *iff* every reachable traffic state s is co-reachable to the "home" state s_h. □

We remind the reader that in the formal modeling frameworks considered in [15], a state s is said to be co-reachable to a state s' *iff* state s' is reachable from state s through a feasible event sequence. Then, the "sufficiency" part of Proposition 4 is obvious, since, under the presumed connectivity of the guidepath graph G, once all agents have been collected in the "home" edge h, then any agent $a \in \mathcal{A}$ can be routed to any other edge $e \in E$ and returned back to h. The "necessity" part can be argued from the observations that (i) the "home" edge h is one of the system edges that must always be accessible by any agent $a \in \mathcal{A}$, according to Definition 6, and furthermore, (ii) any agent a located in h does not hinder the circulation of the remaining agents, and therefore, it can stay on this edge while these remaining agents try to reach edge h. Finally, Proposition 4 motivates naturally the following definition.

Definition 7: A traffic state s of an open ZC-GBTS is characterized as *live iff* it is co-reachable to the "home" state s_h. □

In the following, the entire set of live traffic states will be denoted by S_l.

B. Maximally permissive LES for open ZC-GBTS

Proposition 4 also provides a natural characterization of the *maximally permissive* LES for open ZC-GBTS.

Definition 8: At any traffic state s of an open ZC-GBTS, the *maximally permissive LES* will allow the transition of an agent $a \in \mathcal{A}$ from its current zone $\gamma(a; s)$ to a neighboring free zone e *iff* the traffic state s' that will result from this transition is live. □

In view of Definition 8, the most natural implementation of the maximally permissive LES for open ZC-GBTS is through the on-line assessment of the liveness of any tentative state s' that will result from a contemplated advancement of some given set of agents $\hat{\mathcal{A}} \subseteq \mathcal{A}$ to free neighboring edges. Furthermore, Proposition 1 implies that in the case of open and reversible ZC-GBTS, this assessment is resolved in the most trivial manner; i.e., every considered state s' is live, and therefore immediately admissible by the maximally permissive LES. But in the case of open and irreversible ZC-GBTS, the worst-case computational complexity of assessing the reachability condition of Proposition 4 on any given state $s \in S$ is an open research problem; we refer the reader to [15], [22] for a more systematic characterization of this decision problem and some corresponding results. In view of this situation, one can consider the deployment of a suboptimal – i.e., non-maximally permissive – LES, based on some alternative property that will define the admissibility of any given traffic state s; this property (i) must preserve the traffic liveness, and (ii) its assessment on any given traffic state $s \in S$ must incur a polynomial computational cost with respect to the size of the underlying transport system. We present such a property in the next subsection.

C. Polynomial-complexity LES for open and irreversible ZC-GBTS through h-ordered states and the Banker's algorithm

In this subsection we present an efficient suboptimal LES for open and irreversible ZC-GBTS that is based on the notion of the "h-ordered" traffic state. A detailed definition of this concept is as follows:

Definition 9: A traffic state s of an open and irreversible ZC-GBTS is *"h-ordered" iff* there exists an ordering $[\cdot] : \{1, \ldots, |\mathcal{A}|\} \rightarrow \mathcal{A}$, of the agent set $\mathcal{A}$, such that, for each agent $a_{[i]},\ i = 1, \ldots, |\mathcal{A}|$, there is a feasible route $\mathcal{R}_{a_{[i]}}$ that can take agent $a_{[i]}$ from its original zone $\gamma(a_{[i]}; s)$ to the "home" zone h, while agents $a_{[j]},\ j = i+1, \ldots, |\mathcal{A}|$, maintain their original positions in state s. □

The entire set of h-ordered states will be denoted by S_{ho}. Any element s of this set can be recognized efficiently through a merging algorithm that runs iteratively on the corresponding PDG $\hat{G}(s)$, and, at each iteration, it collapses into node v_h those incident edges on node v_h that are either undirected or they are directed pointing towards the node v_h. If these iterations collapse the entire graph $\hat{G}(s)$ into node v_h, then the considered state s is ordered. Furthermore, the sequence in which the edges $\gamma(a; s),\ a \in \mathcal{A}$, were absorbed into the node v_h during the algorithm execution, defines an agent ordering the satisfies the condition of Definition 9. On the other hand, if the aforementioned iterations fail to collapse the entire graph $\hat{G}(s)$ into node v_h, then the considered state s is not ordered; in particular, the agents a for which the corresponding zone $\gamma(a; s)$ has not been absorbed into node v_h, block each other in their endeavor to find a free route $\mathcal{R}_a$ that will take them to the target node v_h. Finally, from the above description, it should also be clear that the computational complexity of the execution of this merging algorithm on any given state s is $O(|E|)$.

We must emphasize, however, that even if the above merging algorithm returns a negative result when applied on any given traffic state s, state s might still be live, since there might exist a routing schedule $\mathcal{S}$ that takes all agents $a \in \mathcal{A}$ to the "home" zone h but involves interleaved partial advancements of these agents towards their destination. Hence, S_{ho} might be a strict subset of S_l, and the traffic supervisor that seeks to establish traffic liveness by admitting only h-ordered states, will not be maximally permissive.

We also notice that the merging algorithm presented in the previous paragraphs constitutes an adaptation of Dijkstra's Banker's algorithm [23] to the considered operational context; the reader is referred to [10] for a systematic connection of the notion of "h-ordered traffic state" and the presented merging algorithm to the theory of Banker's algorithm that provides an efficient approach for deadlock avoidance in sequential resource allocation systems.

Furthermore, in [24] it is shown that the set of (h-)ordered states admitted by any efficient realization of Banker's algorithm, can be further expanded through controlled excursions to the complementary subspace of non-h-ordered states, in a way that guarantees the liveness of the underlying system and controls the involved computational cost.

Finally, it is easy to see that the "home" state s_h, as well as the states s where only a single agent $a \in \mathcal{A}$ is located

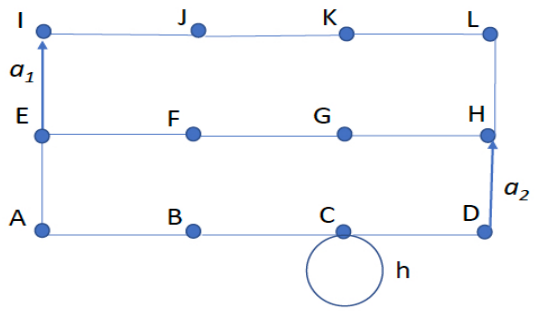

Fig. 4: The initial state s_0 of the guidepath-based transport system considered in the first part of the example of Section V.

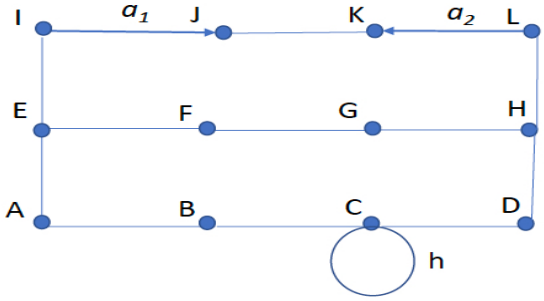

Fig. 5: The final state s' of the guidepath-based transport system considered in the first part of the example of Section V.

on an edge $e \in E$, are h-ordered. Hence, when starting from state s_h, that defines a natural initial state for the considered transport systems, it is possible to satisfy all the visitation requirements Σ_a, for all agents $a \in \mathcal{A}$, while restricting the system operation to its subspace S_{ho}.

Collectively, all the above remarks imply that it is possible to establish live operation for an open and irreversible ZC-GBTS by restricting its operation within its set of h-ordered states, and the resulting LES will also be computationally efficient. In the next section we discuss how this capability can be utilized in the MPC framework that we pursue for this class of transport systems.

V. Adapting the MPC scheme of [4] to open and irreversible guidepath-based transport systems

In this section we show that the MPC scheme of [4], that was discussed in Section III, can be extended to the class of open and irreversible ZC-GBTS in a way that preserves the liveness of the generated traffic. This extension is attained by requesting that the target states s' of the various subproblems that are formulated and solved by this MPC scheme, must belong to the class of live states, S_l. And since the state class S_l might not be easily recognizable, eventually, we focus on the more restrictive class S_{ho}. The following example demonstrates vividly the need for enforcing such a restriction, and provides context and motivation for the subsequent developments.

TABLE I: An optimal routing schedule for the subproblem that is considered in the first part of the example of Section V.

Period	Agent a_1	Agent a_2
1	IJ	HL
2	IJ	LK

Example: Figure 4 depicts an open and irreversible ZC-GBTS with two agents currently on active missions, to be denoted as a_1 and a_2. In the depicted state, s_0, agent a_1 is located on edge EI of the guidepath network, moving in the direction that is indicated in the figure. Similarly, agent a_2 is located on edge DH with the direction of its motion indicated by the corresponding arrow. Finally, the remaining visitation requirements for each of these two agents are, respectively, $\Sigma_{a_1} = \langle IJ, DH \rangle$ and $\Sigma_{a_2} = \langle KL \rangle$. Therefore, the next immediate destination of agent a_1 is the (undirected) edge IJ of the depicted guidepath network, and the next immediate destination of agent a_2 is the (undirected) edge KL.

The reader should notice that the state s_0 depicted in Figure 4 is live since both agents a_1 and a_2 can be routed to the "home" edge h of the underlying guidepath network. In fact, state s_0 is also h-ordered since there are routing schedules that can take any of these two agents to the "home" zone h without moving the other agent from its currently allocated zone. Furthermore, the min-makespan routing schedules for the subproblem that is defined by the considered MPC framework for state s_0 and the aforestated visitation requirements, have a makespan of 2 periods, and such a min-makespan schedule consists of the two agent routes that are described in Table I. The state s' that results from the execution of the routing schedule of Table I is depicted in Figure 5. It can be easily seen that, in state s', the two agents a_1 and a_2 are destined unavoidably to a deadlock, and therefore, the generated traffic is not live.

The proposed adaptation of the MPC scheme of [4] for the class of open and irreversible ZC-GBTS: In view of the above example, the basic problem that is addressed in the rest of this section is the development of a canonical procedure that, if necessary, will redefine the target state for the subproblems formulated in the pursued MPC framework, from the original state s_t that is defined naturally by the most immediate destinations for the traveling agents $a \in \mathcal{A}$ in the corresponding lists Σ_a, to a new state $\hat{s}_t$ that (i) will be live, and (ii) will still provide some progress towards the satisfaction of the posed visitation requirements. Furthermore, as explained in the opening paragraph of this section, we shall also substitute the requirement $\hat{s}_t \in S_l$, by the more restrictive, but also more easily testable requirement $\hat{s}_t \in S_{ho}$; i.e., we shall require that the modified target state $\hat{s}_t$ is h-ordered.

We start our developments with the next proposition, which is an immediate corollary of Proposition 3 for the particular set of states s_t that constitute target states for the various subproblems that are generated by the considered MPC scheme.

Proposition 5: Consider an open and irreversible ZC-GBTS at its "home" state s_h. Furthermore, let s_t denote a traffic state that is defined by placing each agent $a \in \mathcal{A}$ to the edge e_a that

constitutes the next destination for agent a according to the corresponding edge sequence Σ_a, and with the direction of each agent a at the corresponding edge e_a being consistent with the direction that is specified by some shortest path leading from the "home" zone h to edge e_a. Then, state s_t is reachable from the "home" state s_h. □

In view of Proposition 5, the proposed adaptation scheme for the states s_t that constitute the original target states for the subproblems that would be formulated by the original MPC scheme of [4] – i.e., the states defined by the most immediate destinations of the traveling agents in these subproblems – can be epitomized by the following three rules.

Rule 1: If a generated target state s_t is invalid, then, for each group of agents that are conflicting upon a common desitnation edge e, all agents but one must be re-directed to the "home" edge h.

Rule 2: When the system is in the "home" state s_h, let s_t denote the traffic state that is defined by placing the agents $a \in \mathcal{A}$ to their next immediate destinations in the corresponding sequences Σ_a, according to the routing logic that is defined in the proof of Propositions 3 and 5. If state s_t is not valid, apply Rule 1.

Let the state that results from the execution of the previous steps be denoted by s'_t, and the set of agents a with $\gamma(a; s'_t) \neq h$ be denoted by $\mathcal{A}'$. If state s'_t is h-ordered, then, set $\hat{s}_t = s'_t$. In the opposite case, let $\mathcal{A}''$ denote the set of agents that block each other from reaching the "home" zone h in state s'_t.[12] Identify a minimal set of agents $\mathcal{A}''' \subset \mathcal{A}''$ such that the state s''_t that is defined by placing the agents in $\mathcal{A}'' \backslash \mathcal{A}'''$ on their edges $\gamma(a; s'_t)$ and every other agent in zone h, is h-ordered. Finally, define state $\hat{s}_t$ as the traffic state that is obtained by setting the agents in $\mathcal{A}' \setminus \mathcal{A}'''$ to their next immediate destinations with the directions that are specified by the state s'_t, and keeping all the other agents (i.e., the agents in $(\mathcal{A} \setminus \mathcal{A}') \cup \mathcal{A}'''$) at the "home" zone h.

An optimized traffic schedule leading from state s_h to the defined state $\hat{s}_t$ can be obtained by first constructing an initial feasible schedule according to the logic that underlies Proposition 5, and subsequently optimizing further this schedule through the iteration that is defined Steps 3 and 4 in the algorithm of Figure 2.

Rule 3: When the system gets in a state $s \neq s_h$ where some agent a has just reached its current destination, keep executing the current routing schedule $\mathcal{S}$ until reaching a state s' with $s' \in S_{ho}$; this will always be possible, since, according to the proposed control scheme, the target state for the currently executed routing schedule $\mathcal{S}$ is h-ordered. Furthermore, if state s is h-ordered, then, obviously, $s' = s$.

Once state s' has been reached, assess the validity of the states s_t that are defined by advancing each agent $a \in \mathcal{A}$ to its next immediate destination edge e_a in the corresponding edge sequence Σ_a; more than one such states s_t can be obtained by choosing a different orientation for placing each agent $a \in \mathcal{A}$ in the corresponding destination edge e_a, but all these states will have the same validity since this concept does not depend on the aforementioned orientations. If the considered states are found to be invalid, then apply Rule 1, and eventually let $\mathcal{A}'$ be the set of agents that retain an immediate destination edge $e_a \neq h$ after the potential execution of this rule.

Compute an optimized routing schedule $\tilde{\mathcal{S}}$ that routes all agents $a \in \mathcal{A}'$ to their next immediate destinations e_a, and all the remaining agents to the "home" zone h. This schedule can be obtained through a two-phase computational logic similar to that executed by the algorithm of Figure 2. More specifically: (i) During the first phase, an initial schedule will be obtained by first routing all agents $a \in \mathcal{A}$ to the "home" zone h according to the merging algorithm of Section IV-C, and subsequently the agents $a \in \mathcal{A}'$ will be routed to their destination edges e_a according to the logic of Proposition 5. (ii) In the second phase, the schedule obtained from the first phase will be incrementally improved according to the corresponding logic of the algorithm of Figure 2.

Let s'_t denote the traffic state that will result from the execution of the optimized schedule $\tilde{\mathcal{S}}$ that is computed in the previous step. The state s'_t obtained from the above computation will be set to the test "$s'_t \in S_{ho}$?", and the optimized schedule $\tilde{\mathcal{S}}$ will be accepted only if the resultant state s'_t is h-ordered.

If $s'_t \notin S_{ho}$, s'_t must be revised to a new h-ordered state $\hat{s}_t$. This revision is performed through the following steps: (i) First, traffic state s'_t is revised to the traffic state $\tilde{s}_t$ that is obtained from state s'_t by resetting the directions of the agents $a \in \mathcal{A}'$ in the corresponding target edges e_a as requested in the statement of Proposition 5. (ii) Next, state $\tilde{s}_t$ is further processed by applying to it the corresponding logic in Rule 2; i.e., (a) first, we need to determine the subset $\mathcal{A}''$ of $\mathcal{A}'$ containing all the agents that block each other from reaching the "home" zone h in state $\tilde{s}_t$; (b) if $\mathcal{A}'' = \emptyset$ (i.e., the considered state $\tilde{s}_t$ is h-ordered), then, we set $\mathcal{A}''' = \emptyset$; otherwise, we need to identify a minimal set of agents $\mathcal{A}''' \subset \mathcal{A}''$ such that the state s''_t that is defined by placing the agents in $\mathcal{A}'' \setminus \mathcal{A}'''$ on their edges $\gamma(a; \tilde{s}_t)$ and every other agent in zone h, is h-ordered; (c) finally, we shall define the traffic state $\hat{s}_t$ as the traffic state that is obtained by setting the agents in $\mathcal{A}' \setminus \mathcal{A}'''$ to their next immediate destinations with the directions that are specified by the state $\tilde{s}_t$, and keeping all the other agents (i.e., the agents in $(\mathcal{A} \setminus \mathcal{A}') \cup \mathcal{A}'''$) at the "home" zone h.

State $\hat{s}_t$ can be reached from the initial state s' through an optimized schedule $\mathcal{S}$ that again is con-

[12] As explained in Section IV-C, these agents will be identified by the merging algorithm that assesses the h-ordered structure of the considered state s'_t.

structed through a two-phase process: (i) The first phase constructs an initial schedule by first routing all agents to the "home" zone h according to the merging algorithm of Section IV-C, and subsequently routing the agents $a \in \mathcal{A}' \setminus \mathcal{A}'''$ to their destination zones according to the logic of Proposition 5. (ii) The second phase optimizes further this initial schedule by executing the iteration of Steps 3 and 4 in the algorithm of Figure 2.

Figure 6 provides a flowchart of the basic logic that is encoded in Rules 1–3. This flowchart essentially constitutes the necessary algorithm for the formulation and the solution of the different subproblems that will be generated by the MPC framework for open and irreversible ZC-GBTS that is developed in this work. Furthermore, the next theorem establishes that when applied on any open and irreversible ZC-GBTS, this new MPC scheme will preserve the liveness of the generated traffic.

Theorem 1: The sequence of target states, $\hat{s}_t^{(1)}, \hat{s}_t^{(2)}, \hat{s}_t^{(3)}, \ldots$, generated by the MPC scheme for open and irreversible ZC-GBTS that is defined by the algorithm of Figure 6 and Rules 1–3, has the following properties:

1) It consists of valid h-ordered states.
2) When the system is started from the "home" state s_h, each target state $\hat{s}_t^{(i)},\ i = 1, 2, \ldots$, is reachable from the state $\hat{s}_0^{(i)}$ that constitutes the initial state of the corresponding subproblem.
3) Any finite set of visitation requirements that is defined by a set of finite lists $\{\Sigma_a,\ a \in \mathcal{A}\}$, will be satisfied by the considered MPC scheme in finite time.

Proof: The satisfaction of Property #1 in the above theorem is guaranteed immediately by the construction logic for the considered states $\hat{s}_t^{(1)}, \hat{s}_t^{(2)}, \hat{s}_t^{(3)}, \ldots$, that is defined by the flowchart of Figure 6 and Rules 1–3.

In order to establish Property #2, we discern two cases:

Case 1: If state $\hat{s}_0^{(i)} = s_h$, then the reachability of the state $\hat{s}_t^{(i)}$ from state $\hat{s}_0^{(i)}$ is guaranteed by the defining logic of the state $\hat{s}_t^{(i)}$ in Rule 2 (see also the left side of the flowchart of Figure 6) and Proposition 5.

Case 2: If state $\hat{s}_0^{(i)} \neq s_h$, then, the h-ordered property of this state is guaranteed from the first part of Rule 3 (or, equivalently, by the very first step in the right side of the flowchart of Figure 6). Hence, there is a feasible routing schedule leading from this state to the "home" state s_h that can be detected by the merging algorithm of Section IV-C. Furthermore, the existence of a feasible schedule $\tilde{\mathcal{S}}$ that is to be computed in this case, is established by the feasibility of each of the steps for the computation of this schedule that are detailed in the statement of Rule 3. Hence, if the state s'_t that will result from the execution of schedule $\tilde{\mathcal{S}}$ is h-ordered, then Property #2 of Theorem 1 has been established for this case, as well. In the opposite case, the considered algorithm will have to execute the last 3 steps in the right side of the flowchart of Figure 6, revising the state s'_t to the state $\hat{s}_t$ as discussed in that part of the flowchart, and seeking a schedule S leading from the initial state $\hat{s}_0^{(i)} = s'$ to the new target state $\hat{s}_t$; the feasibility of this last schedule should also be obvious from the definition of the involved states s' and $\hat{s}_t$ and the construction methods that are indicated in the statement of the corresponding part of the algorithm (i.e., Step 3 in that block).

Since Cases 1 and 2 cover exhaustively the entire set of traffic states S, Property #2 of Theorem 1 has also been proved.

Finally, Property #3 of Theorem 1 is implied immediately from Property #2 upon noticing that the execution of the routing schedule of each subproblem results in the satisfaction of at least one visitation requirement by some traveling agent $a \in \mathcal{A}$. □

Complexity considerations: Regarding the computational complexity of the algorithm of Figure 6, we make the following remarks: First of all, it should be clear that testing the validity of any given traffic state s and obtaining the corresponding set $\mathcal{A}'$ is a task of complexity $O(|\mathcal{A}|)$. Also, as discussed in Section IV-C, the test "$s \in S_{ho}$?" and the computation of the corresponding set $\mathcal{A}''$ is a task of worst-case complexity $O(|E|)$. In addition, it should be clear that the tasks of computing the schedules $\mathcal{S}$ and $\tilde{\mathcal{S}}$ that appear in the flowchart of Figure 6, are of polynomial computational complexity with respect to the size of the underlying ZC-GBTS, as well, since each of the more detailed steps that are involved in those computations (as listed in that flowchart), have already been shown to be of polynomial computational complexity in the earlier parts of this document.

The only parts in the flowchart of Figure 6 that might present super-polynomial complexity with respect to the size of the underlying transport system, are those involving the extraction of a minimal agent set $\mathcal{A}'''$ from the corresponding parent set $\mathcal{A}''$, since this extraction will require some partial enumeration of the subsets of the parent set $\mathcal{A}''$. From past experience with similar implementations of Banker's algorithm in other application contexts, it is expected that this enumeration will be very benign. On the other hand, if we insist to keep polynomial the computational complexity of the entire algorithm, we can give up the request for minimality of the sought set $\mathcal{A}'''$, and simply set $\mathcal{A}''' = \mathcal{A}'' \setminus \{a\}$ for some element $a \in \mathcal{A}''$; it is easy to check that this alteration will maintain the correctness of Theorem 1, but it might increase the set of agents $a' \in \mathcal{A}''$ that will have to be rerouted to the "home" zone h by the considered subproblem.

Example (cont.): Next, we provide a more concrete exposition of the algorithm that is defined by the flowchart of Figure 6, by tracing the execution of this algorithm on the example transport system that was introduced at the beginning of this section. Figure 7 will facilitate the following discussion.

Hence, we remind the reader that the considered transport system is currently in state s_0 that is reproduced in the left part of Figure 7, and the remaining visitation requirements for the two traveling agents a_1 and a_2 are those stated on the left side of this figure. Since $s_0 \neq s_h$, the algorithm will execute the right part in the flowchart of Figure 6. As already pointed out in the earlier part of this example, state s_0 is h-ordered, and therefore, the algorithm will set $s' := s_0$. Also, since the next immediate destinations for agents a_1 and a_2 are respectively zones IJ and KL, it is clear that all the possible

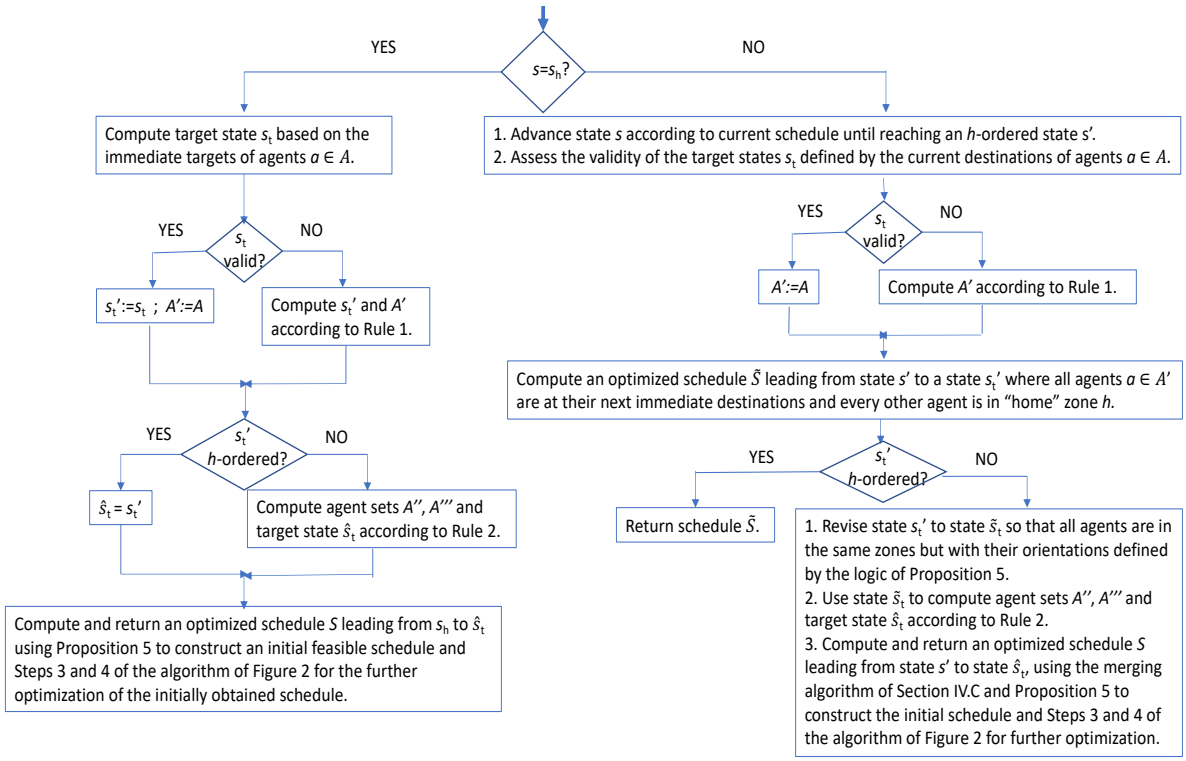

Fig. 6: A flowchart representing the adaptation of the heuristic algorithm of [4] for the generation and solution of the subproblems to be addressed by the MPC framework for open and irreversible ZC-GBTS that is developed in this work.

target states s_t that are defined by these destinations, are valid. The optimized schedule $\tilde{S}$ that will be computed next by the algorithm, will result in state s'_t that is depicted in Figure 5. As discussed in the earlier part of this example, this state is not h-ordered; in fact, it is not even live. Hence, the algorithm will proceed to the execution of the block in the bottom-right corner of the flowchart of Figure 6. In the considered case, we have $\tilde{s}_t = s'_t$ since the original orientations of agents a_1 and a_2 in their corresponding destination edges are those that correspond to the shortest paths that lead from node C to those edges. Next, we obtain an h-ordered state $\hat{s}_t$ from the non-live state $\tilde{s}_t (= s'_t)$, by redirecting agent a_2 to the "home" zone h. Finally, we compute an optimized schedule $\mathcal{S}$ that will lead from the initial state s_0 to the revised target state $\hat{s}_t$.

The optimized schedule $\mathcal{S}$ will place agent a_1 at its next destination edge IJ after one period. At that time, agent a_2 will be in zone GH with the direction depicted in the right part of Figure 7, on its (shortest) path to the "home" zone h. Since agent a_1 has reached its next destination, the algorithm of Figure 6 will be executed again, this time on the state s that is depicted at the right part of Figure 7. Since $s \neq s_h$, the algorithm will execute again through the right part of the flowchart of Figure 7. It can be seen that the considered state s is an h-ordered state, and therefore, we shall have $s' = s$. Also, this time, the two agent destinations are the zones HD and KL, which are distinct, and therefore, the corresponding states s_t are valid. Computing an optimized schedule $\tilde{S}$ leading from the considered state $s (= s')$ to a state that places agents a_1 and a_2 to their current destination edges, leads to a state s'_t where $\gamma(a_1; s'_t) = (H, D)$ and $\gamma(a_2; s'_t) = (K, L)$. Furthermore, it can be checked that state s'_t is h-ordered, and therefore, the algorithm exits returning the schedule $\tilde{S}$ for this second subproblem.

The execution of the schedule $\tilde{S}$ will bring the system to a state q with $\gamma(a_1; q) = (H, D)$ and $\gamma(a_2; q) = (I, J)$, where agent a_1 has reached its next destination, and therefore, at this point the algorithm of Figure 6 will be invoked again with state s being the aforementioned state q and the immediate destinations for the two agents a_1 and a_2 being, respectively, the zones h and KL. Repeating the previous steps for this new case, it is easy to see that the algorithm will produce an optimized schedule $\tilde{S}$ that will lead from the considered state q to the new state s'_t with $\gamma(a_1; s'_t) = h$ and $\gamma(a_2; s'_t) = (K, L)$, which is clearly h-ordered. Furthermore, this new state s'_t will be the starting state s for the next invocation of the algorithm of Figure 6, for the formulation and solution of the last subproblem that must generated for this example, and will bring agent a_2 to the "home" zone h, as well.

Closing the discussion on the considered example, we want also to point out that, as revealed in this discussion, the redirection of some agents to the "home" zone h by the algorithm of Figure 6 does not necessarily imply that these agents will actually be taken to this "home" zone. Indeed, in the considered example, even though agent a_2 was redirected to "home" zone h by the solution of the first subproblem, it just moved to the neighboring zone GH during the execution of the corresponding schedule, and subsequently, it only moved on its optimal route towards its next destination(s). Hence, the redirections to the "home" zone h that are potentially

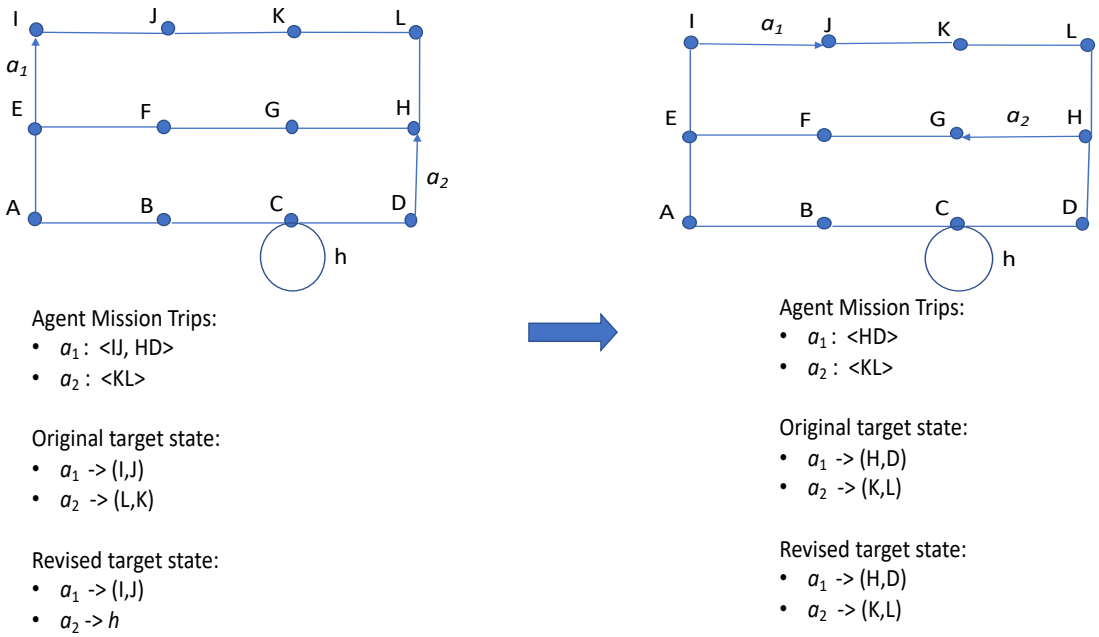

Fig. 7: A partial execution of the algorithm of Figure 6 on the example transport system that was introduced at the beginning of Section V.

effected by the considered algorithm, should be perceived more as "diversions" of those agents from their optimal routes to their current destinations instead of temporary storage of these agents to the "home" zone h. Of course, these diversions might elongate the trips of the corresponding agents to their destinations, as it is also demonstrated by the considered example, but this is the price to pay for ensuring traffic liveness in a tractable manner.

Discussion: Rules 1–3 together with the flowchart of Figure 6 provide a very basic framework for extending the MPC scheme of [4] to the class of open and irreversible ZC-GBTS while ensuring the liveness of the resulting traffic. This framework can be further detailed and enhanced by considering some additional implementational aspects of it.

More specifically, a particular issue that might need some further discussion is the determination of the agent sets $(\mathcal{A}\backslash\mathcal{A}')$ and $\mathcal{A}'''$ that might have to be redirected to the "home" zone h during the execution of the considered algorithm, in order to resolve potential conflicts due to a common destination zone for a subset of agents, or in order to establish an h-ordered structure for the sought target state $\hat{s}_t$. In general, a pertinent selection of these sets might require additional information that will be more specific to the operational context of the underlying transport system; as a more concrete example of such context-specific information, one can mention concerns about the criticality of certain "mission" trips or a prioritization scheme that is imposed over the set of traveling agents. Such additional information can be brought into the considered algorithm by assigning a set of "weights" to the traveling agents, and then, the selection of the aforementioned sets $(\mathcal{A} \setminus \mathcal{A}')$ and $\mathcal{A}'''$ can be framed as the problem of minimizing the total weight of the agents that will be included in these sets. The resulting optimization problem subsequently can be addressed through "branch and bound"-type of search processes that will be conducted over the power set of the agent subset that are engaged in the considered conflict.

Furthermore, some additional elements that can be factored in the "weighting" scheme that is mentioned above are (i) the proximity of the considered agents to the "home" zone h, and (ii) the delay that has been experienced by each traveling agent in its current "mission" due to prior similar diversions. The last criterion is especially important for avoiding phenomena of "indefinite postponement" with respect to the "missions" of certain agents, and, thus, establishing a notion of "fairness" in the underlying operation.

We also want to emphasize that, as it is also the case in the corresponding results of [4], the routing schedules that are presented in the proofs of Proposition 5 and Theorem 1 should be perceived as "certificates" for the feasibility of the traffic-scheduling subproblems that are generated by the considered MPC scheme, and for the liveness of the resulting traffic. The actual agent routes to be followed towards the agent destinations that are defining these subproblems, eventually will be determined through the solution of these subproblems by optimizing techniques similar to those that have been developed in [4]. This fact is also captured in the statement of Rules 1–3 and in the flowchart of Figure 6, and it was demonstrated very vividly by the example of the algorithm execution that was provided in the previous parts of this section.

In fact, stretching further the remarks in the previous paragraph, we can also envision instantiations of the transport systems and the corresponding traffic management scheme considered in this work, where the "home" zone h might not be able to accommodate physically all the system agents, but only up to a certain number of these vehicles. On the other hand, the applied control logic will still treat zone h as having a capacity of $|\mathcal{A}|$. The plausibility of such instantiations is due to the fact that in an operational setting where (i) the transport capacity of the underlying system is highly utilized, and the fleet of vehicles is not (much) larger than the actual number of vehicles that can simultaneously circulate in the guidepath network, it is quite unlikely to have many of these vehicles located at the "home" zone h simultaneously. The actually needed capacity of h as a fraction of $|\mathcal{A}|$ will depend, in general, on various factors like the experienced volume and patterns of the transport requirements, the resulting utilization and the traffic patterns for the traveling agents, and the topology of the guidepath network. The empirical assessment of the plausibility of such a relaxed operational scheme, and of the resulting gains in the required capacity of the "home" zone h, would be an interesting follow-up to this work.

We can also consider the selection of alternative destinations for those agents that are redirected to the "home" zone h in any given subproblem, that might be more accessible from the current location of these agents and will still preserve the liveness and/or the h-ordered structure of the resulting target state $\hat{s}_t$. This problem currently is pretty open, and its systematic investigation is part of our intended future work in this area.

Finally, in the last part of this discussion, we want to introduce some further possibilities of the presented work that go beyond the characterization of the induced dynamics and the elaboration on the implementational details of the methodology that has been developed in Section V. The first of these possibilities is defined by the following two observations: The presented methodology essentially tries to redefine, if necessary, the target state for the subproblem that is formulated and solved at each iteration of the presented MPC framework, in a way that will guarantee the liveness of the generated traffic. On the other hand, the solution of the resulting subproblems basically reuses the methodology of [4] for solving the corresponding subproblems formulated and solved by this MPC framework when applied to the traffic management of open but reversible ZC-GBTS. But then, the developments of this section can also be coupled with any existing methodology for open and reversible ZC-GBTS that addresses the considered traffic management problems under some performance objectives other than the schedule makespan. Clearly, such a possibility expands further the practical scope and the application potential of this work; we refer the reader to [8] and the references cited therein for the characterization of such an alternative set of objectives for the considered traffic scheduling problems and the provision of some methodology for their solution in the context of reversible ZC-GBTS.

Furthermore, the availability of a systematic methodology for addressing the traffic scheduling problem considered in this work also enables an optimized selection of system attributes like the size of its agent fleet and the length of the zones of the guidepath network, while assuming a fixed topology of the guidepath network and a certain structure for the processes that will generate the expected transport tasks. The optimization of these parameters can be performed through a high-fidelity simulation (or emulation) of the system operation under various settings of the aforementioned parameters, while using the MPC control scheme developed in this work for the actual operation of the underlying transport system. The further development of this idea can be based on simulation-driven optimization methods like those presented in [25]. And, of course, this possibility is true not only for the open and irreversible ZC-GBTS considered in this work, but also for their reversible counterparts that are studied in [4].

VI. Conclusion

This paper has developed an MPC scheme for the real-time traffic management in open and irreversible, zone-controlled, guidepath-based transport systems. Special emphasis has been placed on the preservation of the liveness of the generated traffic, in view of the deadlock and livelock effects that are experienced in the context of these systems. To the best of our knowledge, this is the first effort to provide such a systematic solution to the considered traffic management problem, and the presented framework is building upon and leverages an extensive body of results that have been developed in our past studies of the corresponding transport systems.

Our future work will seek to (i) further address the implementational details and variations that are outlined in the closing discussion of Section V, and (ii) extend the considered MPC scheme to *closed* ZC-GBTS – i.e., to ZC-GBTS that do not possess a "home" zone – by leveraging some results on the liveness characterization and analysis of closed ZC-GBTS that are presented in [13], [26].

References

[1] S. S. Heragu, *Facilities Design (3rd ed.)*. CRC Press, 2008.

[2] D. Pillai, "The future of semiconductor manufacturing: Factory integration breakthrough opportunities," *IEEE Robotics & Automation Magazine*, vol. 13-4, pp. 16–24, 2006.

[3] P. Surynek, "Towards optimal cooperative path planning in hard setups through satisfiability solving," in *Proc. 12th Pacific Rim Int. Conf. Artif. Intell.*, 2012, pp. 564–576.

[4] G. Daugherty, S. Reveliotis, and G. Mohler, "Optimized multi-agent routing for a class of guidepath-based transport systems," *IEEE Trans. on Automation Science and Engineering*, vol. 16, pp. 363–381, 2019.

[5] J. Yu and S. M. LaValle, "Optimal multi-robot path planning on graphs: Structure and computational complexity," *ArXiv: http://arxiv.org/pdf/1507.03289v1.pdf*, 2015.

[6] H. Ma, C. Tovey, G. Sharon, S. Kumar, and S. Koenig, "Multi-agent path finding with payload transfers and the package-exchange robot-routing problem," in *AAAI 2016*, 2016, pp. 3166–3173.

[7] T. Standley and R. Korf, "Complete algorithms for cooperative pathfinding problems," in *Proc. 22nd Intl. Joint Conf. Artif. Intell.*, 2011.

[8] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Trans. on Robotics*, vol. 32, pp. 1163–1177, 2016.

[9] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. London, UK: Springer, 2015.

[10] S. A. Reveliotis, "Conflict resolution in AGV systems," *IIE Trans.*, vol. 32(7), pp. 647–659, 2000.

[11] N. Wu and M. Zhou, "Resource-oriented Petri nets in deadlock avoidance of AGV systems," in *Proceedings of the ICRA'01*. IEEE, 2001, pp. 64–69.
[12] M. P. Fanti, "Event-based controller to avoid deadlock and collisions in zone-controlled AGVS," *Inlt. Jrnl Prod. Res.*, vol. 40, pp. 1453–1478, 2002.
[13] E. Roszkowska and S. Reveliotis, "On the liveness of guidepath-based, zoned-controlled, dynamically routed, closed traffic systems," *IEEE Trans. on Automatic Control*, vol. 53, pp. 1689–1695, 2008.
[14] S. Reveliotis and E. Roszkowska, "On the complexity of maximally permissive deadlock avoidance in multi-vehicle traffic systems," *IEEE Trans. on Automatic Control*, vol. 55, pp. 1646–1651, 2010.
[15] S. Reveliotis and T. Masopust, "Some new results on the state liveness of open guidepath-based traffic systems," in *27th Mediterranean Conference on Control and Automation (MED 2019)*. IEEE, 2019.
[16] S. Reveliotis, "Preservation of traffic liveness in MPC schemes for guidepath-based transport systems," in *IEEE CASE 2018*. IEEE, 2018.
[17] T. Ganesharajah, N. G. Hall, and C. Sriskandarajah, "Design and operational issues in AGV-served manufacturing systems," *Annals of OR*, vol. 76, pp. 109–154, 1998.
[18] I. F. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 170, pp. 677–709, 2006.
[19] H. N. Psaraftis, "Dynamic vehicle routing problems," *Vehicle Routing: Methods and Studies*, vol. 16, pp. 223–248, 1988.
[20] L. A. Wolsey, *Integer Programming*. NY, NY: John Wiley & Sons, 1998.
[21] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1993.
[22] S. Reveliotis and T. Masopust, "Efficient liveness assessment for traffic states in open, irreversible, dynamically routed, zone-controlled guidepath-based transport systems," *IEEE Trans. on Automatic Control*, vol. 65, 2020.
[23] E. W. Dijkstra, "Cooperating sequential processes," Technological University, Eindhoven, Netherlands, Tech. Rep., 1965.
[24] M. Lawley, S. Reveliotis, and P. Ferreira, "The application and evaluation of Banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems," *Intl. Jrnl. of Flexible Manufacturing Systems*, vol. 10, pp. 73–100, 1998.
[25] M. C. Fu (ed.), *Handbook of Simulation Optimization*. NY, NY: Springer, 2015.
[26] S. Reveliotis, "On the state liveness of some classes of guidepath-based transport systems and its computational complexity," *Automatica*, vol. (to appear).

Spyros Reveliotis received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, the M.Sc. degree in computer systems engineering from Northeastern University in Boston, and the Ph.D. degree in industrial engineering from the University of Illinois at Urbana-Champaign.

He is a Professor with the School of Industrial and Systems Engineering, Georgia Institute of Technology, in Atlanta, GA. His main research interests are in discrete event systems theory and its applications.

Dr. Reveliotis is an IEEE Fellow and a member of INFORMS. He has served on the editorial boards of many journals and conferences in his areas of interest. Some of his latest editorial positions are those of a Senior Editor for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, an Associate Editor for the Journal of Discrete Event Dynamic Systems, and the Editor-in-Chief of the Editorial Board at the IEEE Conference on Automation Science and Engineering (CASE). He has also served as the Program Chair at the 2009 IEEE CASE Conference, and the General Co-Chair of the 2014 edition of the same conference. Finally, he has been a recipient of a number of awards, including the 2014 Best Paper Award of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.