Electronic Supplement for the Manuscript "On the state liveness of some classes of guidepath-based transport systems and its computational complexity"

Spyros Reveliotis

Abstract

This electronic supplement provides the complete proofs for the results that are presented in the main manuscript.

1 Proof of Proposition 3

(Sufficiency) The sufficiency of the condition of Proposition 3 for the liveness of the traffic of the underlying transport system follows immediately from the content of this condition, the controllability of the considered dynamics, and Definition 2.

(Necessity) Next, let us assume that there is a state s that violates the condition of Proposition 3; i.e., for every strongly connected component Ψ of $R(s)$, there will exist (at least) an agent–edge pair (a, e) such that $\epsilon(a; s') \neq e, \forall s' \in \Psi$.

Consider the set of the maximal strongly connected components of $R(s)$. Since $R(s)$ is a finite set, the set of its maximal strongly connected components is well-defined and finite; let us denote it by $\mathcal{M} = \{\Psi_1, \ldots, \Psi_l\}$.

The elements of $\mathcal{M}$ are partially ordered through the "reachability" relation $\mathcal{R}$ where $\mathcal{R}(\Psi_i, \Psi_j)$ implies that there exists a transition sequence leading from component Ψ_i to component Ψ_j. Also, let $\Psi^{(0)}$ be the maximal strongly connected component that contains state s, and therefore, it constitutes the unique minimal element of the considered partial order.

Let (a_0, e_0) be an agent-edge pair such that $\epsilon(a_0; s') \neq e_0, \forall s' \in \Psi^{(0)}$. If $\epsilon(a_0; s') \neq e_0, \forall s' \in R(s)$, then the considered transport system is not live. If, on the other hand, $\epsilon(a_0; s') = e_0$ for some state(s) $s' \in R(s)$, consider the set $\mathcal{M}^{(1)}$ of the maximal strongly connected components $\Psi_i, i = 1, \ldots, l$, that contain such a state s' and constitute minimal elements in the aforementioned partial order of the corresponding set $\mathcal{M}$ that is defined by $\mathcal{R}$. Clearly, the requirement to place agent a_0 on edge e_0 will bring the traffic state of the underlying transport system to one of the strongly connected components of set $\mathcal{M}^{(1)}$. Let $\Psi^{(1)}$ be one of these components, and (a_1, e_1) be an agent-edge pair such that $\epsilon(a_1; s') \neq e_1, \forall s' \in \Psi^{(1)}$. Then, reasoning as above, we can infer that either the underlying transport system is not live (if $\epsilon(a_1; s'') \neq e_1, \forall s'' \in R(s')$), or the only way for satisfying the corresponding visitation requirement is by transitioning to a maximal strongly connected component $\Psi^{(2)}$ that is reachable from $\Psi^{(1)}$ (and therefore, located higher than $\Psi^{(1)}$ in the underlying order). For component $\Psi^{(2)}$, there will also exist a pair (a_2, e_2) such that $\epsilon(a_2; s') \neq e_2, \forall s' \in \Psi^{(2)}$, which enables the repetition of the above argument with respect to this component. But then, the non-liveness of the underlying transport system can be concluded by the finiteness of the set $\mathcal{M}$ and of all the chains of the elements of this set that are defined by the partial order $\mathcal{R}$.

2 Proof of Theorem 7

In the following, we prove the first part of Theorem 7. The second part of the theorem can be proved in a similar manner, taking into further consideration the necessity of Condition 1 for the liveness of irreversible, dynamically routed guidepath-based transport systems.

(Necessity) In order to establish the necessity of the co-reachability condition of Theorem 7, let s denote a live state of the considered transport system, and further suppose, without loss of generality, that $s \neq s_h$. Also, let a_1 denote an agent with $\epsilon(a_1; s) \neq e_h(a_1)$. Then, there must exist a feasible event sequence σ that takes agent a_1 to its "home" edge $e_h(a_1)$.

Furthermore, the "home" structure that is defined by the edges $e_h(a),\ a \in \mathcal{A}$, implies that we can obtain a subsequence σ' of σ that transfers agent a_1 to $e_h(a_1)$ without relocating the agents $a' \in \mathcal{A}$ that are already in their "home" edges $e_h(a')$ in state s. Let the resulting state be denoted by s_1. If $s_1 = s_h$, then, the co-reachability condition of Theorem 7 has been met. Otherwise, select an agent a_2 with $\epsilon(a_2; s) \neq e_h(a_2)$, and repeat the above argument. Since every invocation of this argument increases the number of agents $a \in \mathcal{A}$ that are located in their "home" edges $e_h(a)$ by one, and the entire set of agents $\mathcal{A}$ is finite, it follows that eventually we shall reach a state where every agent $a \in \mathcal{A}$ is located in its "home" edge $e_h(a)$, i.e., state s_h.

(Sufficiency) For this part of the proof, we simply notice that the presumed co-reachability of the considered state s to the "home" state s_h, when combined with the liveness of state s_h that is established by Proposition 6, further implies that satisfaction of the condition of Proposition 3 by state s.

3 Proof of Corollary 8

In view of Proposition 6, it suffices to establish the above result only for states $s \in S$ with $s \neq s_h$. Furthermore, in view of Theorem 7, it suffices to show that the considered state s is co-reachable to the "home" state s_h. For this, we shall order the agents $a \in \mathcal{A}$ with $\epsilon(a; s) \neq e_h(a)$ in increasing distance of their currently held edge $\epsilon(a; s)$ from the "home" vertex v_h; in this ordering, distance is measured by the number of edges of the corresponding shortest paths, and any ties among two or more agents can be handled arbitrarily. Then, for any agent a that is among the closest to the "home" vertex v_h according to the above order, the shortest path(s) leading from edge $\epsilon(a; s)$ to vertex v_h are free in state s. Furthermore, the reversibility of the considered system implies that agent a can always take this shortest path to vertex v_h, and eventually to its "home" edge $e_h(a)$. And this advancement will not relocate any agents $a' \in \mathcal{A}$ that are already in their "home" edges $e_h(a')$ at state s; i.e., the state s' that results from the aforementioned advancement has a smaller number of agents $a \in \mathcal{A}$ with $\epsilon(a; s) \neq e_h(a)$. But then, repeating the above argument a finite number of times, we shall be able to construct an event sequence σ that will lead from state s to the "home" state s_h.

4 Proof of Lemma 11

If edge e is a "hole" in state s, then we simply set $s' \equiv s$. Otherwise, let e' denote an edge containing a "hole" in state s that has the shortest possible distance from edge e, in terms of the number of edges of the corresponding shortest path, and let $p = \langle e \equiv e_0, e_1, \ldots, e_l \equiv e' \rangle$ be one of the shortest paths connecting e and e'. Then, according to the working assumptions, each edge $e_i,\ i = 0, 1, \ldots, l-1$, is occupied by an agent a_i. Consider the state s' that is obtained from state s by advancing each agent $a_i,\ i = 0, \ldots, l-1$, from its current edge e_i to edge e_{i+1}, starting with agent a_{l-1} and working in decreasing order of index i. Then, it is not hard to see that the "hole" at edge e' in state s has moved to edge e in state s'.

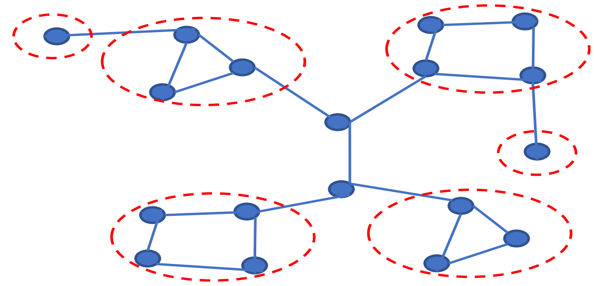

Fig. 1. An example of the decomposing "tree"-like structure that is used in the proof of Theorem12. The solid structure in this figure is the original guidepath graph G, while the subgraphs of G that are annotated in dashed circles, are the nodes of the "tree"-like structure that is induced from G according to the logic that is stated in the proof of Theorem 12.

5 Proof of Theorem 12

Consider first the case where $\mathcal{P}_S = \emptyset$. Then, the condition of Theorem 12 becomes $|\mathcal{A}| \leq |E| - 1$. Also, consider a path $p = \langle e \equiv e_0, e_1, \ldots, e_l \equiv e' \rangle$ from e to e'. Then, the working assumption $\mathcal{P}_S = \emptyset$ implies that there is a path from edge e_1 to a "hole" that does not include edge e_0, and working as in the proof of Lemma 11, we can move this "hole" to edge e_1. Hence, agent a can advance across path p by one edge, to edge e_1. Furthermore, iterative invocation of the above argument implies that there is a routing schedule that can take agent a all the way to edge e'.

When singular paths are present, the entire graph G can be uniquely decomposed to a "tree"-like structure $\mathcal{T}$ as follows: (a) The nodes of $\mathcal{T}$ are (i) the maximal connected subgraphs $G_k, k = 1, \ldots, K$, of G that contain at least one edge and no singular paths, and also (ii) any terminal vertices of the singular paths of G that do not connect to any of the aforementioned subgraphs G_i. (b) The edges of $\mathcal{T}$ correspond to the singular paths of G that interconnect two nodes of $\mathcal{T}$. A concrete example of this graphical decomposition is depicted in Figure 1.

In the context of the "tree"-like structure that was defined in the previous paragraph, ransferring a given agent a from its current edge e to edge e' will involve, in general, the traversal of a "path" Q consisting of some subgraphs G_k and the interconnecting singular paths. Let G_1 denote the maximal subgraph containing edge e. Lemma 11 guarantees that we can move a "hole" to subgraph G_1, and, subsequently, an argument similar to that provided in the first part of this proof further establishes that agent a can move between any pair of edges of the considered subgraph G_1. Also, let G_2 denote the maximal subgraph that is second in the aforementioned "path" Q. The condition $|\mathcal{A}| \leq |E| - 1 - \max_{p \in \mathcal{P}_S}\{|p|\}$ guarantees that (i) it is always possible to empty the singular path p leading from G_1 to G_2 that is required by the traveling agent a, while preserving the accessibility of agent a to this path, and (ii) the agent ability to enter the next required subgraph G_2: all that needs to be done is first to bring agent a to an edge e_1 of G_1 that is adjacent to the singular path p, and subsequently empty the path p of any other agents while ensuring the presence of a "hole" in subgraph G_2. Working in this way, agent a can advance through the entire "path" Q that connects edges e and e' in the aforementioned "tree"-like structure.

6 The polynomial reduction that establishes the result of Theorem 16

The next definition is based on the coverage of the corresponding material in Reveliotis (2017).

Definition – State safety for L-SU-RAS with unit resource capacities: Consider a set of m reusable resources $\mathcal{R} = \{R_1, \ldots, R_m\}$ and another set of n process instances $\Pi = \{J_1, \ldots, J_n\}$ that need to utilize these resources for their execution. More specifically, each process instance $J_j,\ j = 1, \ldots, n$, is defined by a resource sequence $\mathcal{S}_j = \langle R[1;j], \ldots, R[l_j;j]\rangle;\ R[k;j] \in \mathcal{R},\ \forall k \in \{1, \ldots, l_j\}$, that constitutes the corresponding "process plan" and must be interpreted according to the following semantics: Process instance $J_j,\ j = 1, \ldots, n$, currently holds exclusively resource $R[1;j] \in \mathcal{S}_j$ and it further needs the sequential and exclusive allocation of the remaining resources in $\mathcal{S}_j$ in order to advance to its completion. The allocation of the system resources to these process instances is coordinated by a central controller, and a requested resource allocation is feasible only if the considered resource is currently free. Furthermore, a process instance J_j will release its currently allocated resource, $R[k;j]$, only after it has been granted the next required resource, $R[k+1;j]$, in the corresponding process plan $\mathcal{S}_j$. Finally, the system controller will grant any resource allocation requests that satisfy the aforestated conditions one at a time (and will recheck the feasibility of the remaining requests in the RAS state that will result from the execution of the selected allocation). We need to resolve whether there exists a resource allocation sequence for advancing process instances $J_j,\ j = 1, \ldots, n$, through their various processing stages that are defined by the corresponding process plans $\mathcal{S}_j$; more specifically, this resource allocation sequence must be feasible with respect to the aforestated resource allocation protocol, and it must allow each process instance J_j to complete successfully the corresponding process plan $\mathcal{S}_j$.

Next we use the above decision problem in order to establish the result of Theorem 16.

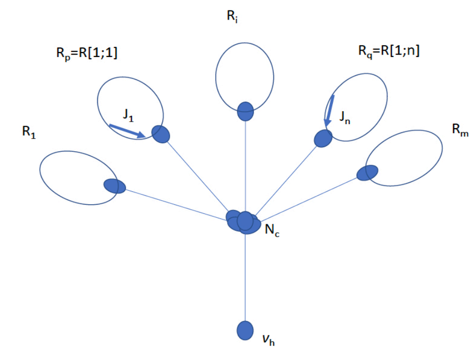

Fig. 2. The traffic state s' that is constructed in the reduction of the proof of Theorem 16.

Proof: First we show that the considered problem belongs in NP. Hence, consider a given traffic state s where the traveling agents $a \in \mathcal{A}$ must execute the walks $W_a(s)$ that correspond to their remaining "mission" trips at state s. Then, defining an event to be the advancement of a single agent by one step through its corresponding "mission" trip, any event sequence σ that leads to the completion of the "mission" trips of all agents, will have a fixed length equal to the sum of all the steps in the random walks $W_a,\ a \in \mathcal{A}$; this sum is polynomially related to the problem data. Furthermore, the validity of such a sequence σ can be assessed through simulation, and this task is also of polynomial complexity with respect to the size of the underlying traffic system. Therefore, the considered problem is in NP.

In order to establish the NP-completeness of the considered problem, we shall reduce to it the L-SU-RAS state-safety problem that was introduced in the previous definition. So, consider an instance of this second problem, and let s denote the corresponding RAS state. The traffic state s' that will be constructed by the proposed reduction is depicted in Figure 2. The corresponding guidepath network possesses a central node N_c and $m+1$ edges $e_i,\ i = 1, \ldots, m+1$, that are incident to this node in a "hub & spoke" sense. At the second node of each edge $e_i,\ i = 1, \ldots, m$, there is a "self-loop" edge that corresponds to resource R_i. On the other hand, the second node of edge e_{m+1} is the "home" vertex v_h (we have not drawn explicitly the "home" edges $e_h(a),\ a \in \mathcal{A}$, in the figure). Each process instance $J_j,\ j = 1, \ldots, n$, is represented in the constructed traffic state s' by an agent a_j located at the "self-loop" edge that corresponds to resource $R[1;j]$; this is indicated by representing this "self-loop" edge as a directed edge. Finally, the agent corresponding to process instance J_j must visit each of the "self-loop" edges that correspond to the resources $R[k;j],\ k = 2, \ldots, l_j$, according to the sequence that is specified by the corresponding process plan $\mathcal{S}_j$, and furthermore, it cannot visit any other edge that is not absolutely necessary for the realization of this process plan; these requirements define completely the walk that must be executed be each agent $a \in \mathcal{A}$ in order to complete its current "mission" trip and retire to its "home" edge $e_h(a)$.

It is not difficult to see that, under the aforestated specification of the route to be followed by each traveling agent, the construction of the previous paragraph defines a bisimulation between the original RAS dynamics and the dynamics of the induced traffic system. Hence, the original RAS state s will be safe if and only if the induced state s' is live. Furthermore, it is clear that the size of the employed representation of the constructed state s' is related polynomially to the size of the employed representation for the RAS state s. Hence, the claim of Theorem 16 is true.

References

Reveliotis, S. (2017), 'Logical Control of Complex Resource Allocation Systems', *NOW Series on Foundations and Trends in Systems and Control* **4**, 1–223.