1. DE comments

the merit of the proposed technique should be clarified either in a general setting or on specific classes of systems like Petri nets or finite automata, and comparisons with extant tools and algorithms should be made.

2. AE Comments

The contribution and usefulness of the proposed approach is not convincing and should be improved.

1. The authors should convincingly demonstrate the benefit of the proposed MPR over other formalism like the automata and PN while also considering the complexity burden of solving the resulting mathematical program.

2. Furthermore, the discussion about the application of the approach should be significantly strengthened. Currently, this is rather vague and there is no comparisons with existing approaches that are used to address these problems.

3. Reviewer 1

However, I think that the authors should clarify better their contribution with respect to the standard translation of other modelling formalisms into mathematical programming (see references below). From a certain point of view, mathematical programming is also a modelling formalism, even it is usually considered more a computational tool, anyway it is evident that it cannot replace finite state automata or Petri nets to model DES. Hence, my question is, there is a way to translate, as for example a PN using the proposed technique? If yes, is it more efficient? If not, what is the effective application domain of the proposed techinique, in addition to queueing systems?

E. Di Marino, R. Su, F. Basile, Makespan optimization using Timed Petri Nets and Mixed Integer Linear Programming Problem, IFAC-PapersOnLine, Volume 53, Issue 4, 2020, pp. 129-135.

Basile F., Chiacchio P., Tommasi G.D. On K-diagnosability of Petri nets via integer linear programming Automatica, 48 (9) (2012), pp. 2047-2058.

Bemporad A., Morari M. Control of systems integrating logic, dynamics, and constraints Automatica, 35 (3) (1999), pp. 407-427.

4. Reviewer 2

1. The main criticisms I can to formulate about this MPR approach is that (i) it produces only fixed-length simulations of a system, and (ii) it results in high-dimension linear programs (O(NxE) where N is the length of the execution and E the number of event types). This puts stringent restrictions on the possible applications of this encoding. For instance, steady-state behavior can only be approximated, at the expense of very large MPR systems. This casts a serious doubt about the usefulness of the whole approach, in comparison with a DES simulator, implementing an operational semantics of the particular DES considered in the paper.

For this reason, I recommend the rejection of the paper, and encourage the authors to replace part of the queueing network case-studies by an in-depth technical discussion of the possible applications of this MPR encoding.

2. Detailed Comments ==================

The paper reads clearly. There are only very few typos (see at the end).

From a technical point of view, there is one thing that needs to be clarified, page 8: How do you choose constant M? Understandably, it should be large enough. However, I would expect to find in the paper a discussion about a lower bound on M.

The queueing networks case-studies are carefully detailed. Perhaps some space could be gained here, and used Section 4, to allow a detailed and technical discussion about the applications of the MPR encoding.

Here is a list of typos:

p 5, Section 2.3, line 2: ...proposed. This...