Dear DE, AE and reviewers,

Thanks to your valuable comments, this submission has undergone a major revision. In this revised version, the MP models are developed based on general timed Petri nets (TPN) instead of the proposed representation of discrete event dynamic systems (DEVS) that was used in the previous submission. This change reduces the difficulty of applying the proposed approach and can be better understood by people familiar with DEVS. However, since the TPN that we consider does not model firing cancellation, this manuscript does not cover the systems with event cancellation, which the previous one did. The coauthors and I believe that the benefit of TPN-based MP formulation provides tighter connection to the DEVS literature and is more important than extending to system with cancellation.

The concerns about application, usefulness, benefit and computational complexity were mentioned by the editors and reviewers for several times. To answer such concerns, we have seen many applications of the use of mathematical programming (MP) for the representation of system dynamics in literature. Most of the works couples the system dynamics and optimization, as in Di Marino, Su, and Basile (2020), Weiss and Stolletz (2015), Bemporad and Morari (1999), Alfieri, Matta, and Pastore (2020). All those works solve problems in industrial applications, thus, the applicability has been proved. The properties of DEVS can also be analyzed through MPs, as in Basile, Chiacchio, and De Tommasi (2012).

The usefulness of this work can be justified in two aspects, namely generalization and convenience. TPN is a widely–used general modeling diagram of DEVS, and the proposed work would be easy to apply for people in this field.

The computational complexity of using MP to optimize DEVS has been addressed in many works, and they dealt the issue with classic MP techniques such as Benders decomposition, linear approximation, columns-row generation. We cannot deny that the computation issue really exists, but the ways to overcome are also vast. Furthermore, the translation is bidirectional. One can translate simulation into MP to optimize a system, and also translate the resulting MP into simulation. The value of translating MP into simulation is that the research on MP (such as duality and sensitivity) can increase the value of simulation, which is currently usually regarded only as a performance evaluation tool.

In the following, we answer the comments in details.

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.

Answer: In the revised manuscript our approach has been reformulated to have it working under general settings of timed Petri net. With an example of the $G/G/m$ queue, the differences between the proposed model and the state-of-the-art MP model are discussed.

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.
Answer: In the revised manuscript, the MPR formulation is equivalent to the simulation realization of TPN.

We justify the benefit and computational burden of the proposed approach as follows:

The benefits of using MP models to represent discrete event systems are many. As already mentioned, when coupled with optimization, the MP-based algorithms can be faster and reach better solution than black–box optimization algorithms. Furthermore, black–box approaches have limited capability in solving constrained optimization problems, while MP–based approaches can easily deal with them (Zhang & Matta, 2020). The vast theoretical and methodological results developed in the MP field can be introduced into the study of DEVS through simulation. For instance, using sensitivity analysis of MP models, gradient estimates can be easily derived as suggested in (Chan & Schruben, 2008). In fact, this paper does not propose to totally replace simulation with MP, but to enrich the toolbox for analyzing and optimizing DEVS. Finally, the application of MP models of DEVS is not limited to optimization but also to show formal properties of the system under study. For instance, Basile et al. (2012) proposed the sufficient and necessary condition of K-diagnosability of TPN based on MP analysis.

The major concern about the application of MP is the computational complexity, as the solution procedure can be time–consuming or even unbearable. However, many approaches from optimization community are available to improve the efficiency. Linear programming approximation (Alfieri & Matta, 2012), Benders decomposition (Weiss & Stolletz, 2015), row–column generation (Alfieri et al., 2020), have been studied to solve optimization problems in real contexts based on mathematical programming representation of DEVS.

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.
Answer: With an example of the $G/G/m$ queue, the differences between the proposed model and the state-of-the-art model are discussed.

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 technique, in addition to queueing systems?

References: Basile et al. (2012), Basile et al. (2012), Bemporad and Morari (1999)

Answer: As the reviewer suggests, in the revised manuscript we propose the translation of TPN into MP.

The difference between the proposed approach and the suggested references are detailed in the following.

[1] Bemporad and Morari (1999) proposed a mixed integer programming modeling framework for hybrid systems, whose states are mixed integer, in discrete time, while this work deals with continuous–time discrete–state systems.

[2] An MP model was proposed to minimize the makespan of single-server manufacturing systems in Di Marino et al. (2020), while our work is more general and can model multi-server systems.

[3] Basile et al. (2012) proposed the model based on PN, not TPN, because the paper is about diagnosability rather than time-dependent performance measures, such as waiting time or throughput.

The effective application domain of the proposed technique is basically related to optimization. The MP model of TPN can be easily extended to an MP model of TPN optimization. Based on the proposed MP model, an example with the extended MP model to optimize the $G/G/m$ queue is presented in Section 4.3.

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.

Answer: The reviewer's comment is reasonable. We discuss the computational burden as follows in the submission:

The major concern about the application of MP is the computational complexity, as the solution procedure can be time–consuming or even unbearable. However, many approaches from optimization community are available to improve the efficiency. Linear programming approximation (Alfieri & Matta, 2012), Benders decomposition (Weiss & Stolletz, 2015), row–column generation (Alfieri et al., 2020), have been studied to solve optimization problems in real contexts based on mathematical programming representation of DEVS.

In the PhD thesis of one of the coauthors (**?**), a Benders decomposition (BD) algorithm solving integrated simulation-optimization mathematical programming models is proposed. In the BD procedure, the subproblem is the MP representation of simulation realization of TPN. The Benders cuts are derived from the dual solution of the subproblem. The thesis proposed an algorithm to solve the dual solution from simulation, so that the computational complexity is linear to the number of decision variables. The BD algorithm finds good feasible solutions fast and guarantees sample-path optimality in the long run. A working paper is draft to present the computationally efficient BD algorithm. The coauthors and I decide not to combine the algorithm in this manuscript, since we would like to make the topic concentrated on MPR modeling and easy to digest.

2. Detailed Comments

The paper reads clearly. There are only very few typos. Here is a list of typos: p 5, Section 2.3, line 2: ...proposed. This...

Answer: we have correct the typos.

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.

Answer: In the subsections of Section 3, the value of M is discussed each time it is used, as different values of M are required in the constraints where M is needed.

Section 3.1: The value of M in constraints (A1) and (A2) can be set to a value that is larger than the simulation time span as it is needed to bound the finishing time of transitions.

Section 3.2: Also in this case, the value of M can be set to a value that is larger than the simulation time span for constraints (B1) and (B2) as it is used to bound the transition firing time. For constraints (B4), instead, the value of M can be chosen as the upper bound of marking of place p minus $(W^{p,t} - 1)$.

Section 3.3: The value of M in constraints (C3) and (C4) can be set to K, i.e., the total number of iterations.

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.

Answer: We remove the case study on the queueing network and add new examples of comparison and the extended optimization formulation.

References

Alfieri, A., & Matta, A. (2012). Mathematical programming formulations for approximate simulation of multistage production systems. *European Journal of Operational Research*, *219*(3), 773–783.

Alfieri, A., Matta, A., & Pastore, E. (2020). The time buffer approximated buffer allocation problem: A row–column generation approach. *Computers & Operations Research*, *115*, 104835.

Basile, F., Chiacchio, P., & De Tommasi, G. (2012). On k-diagnosability of petri nets via integer linear programming. *Automatica*, *48*(9), 2047–2058.

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

Chan, W. K., & Schruben, L. (2008). Optimization models of discrete-event system dynamics. *Operations Research*, *56*(5), 1218–1237.

Di Marino, E., Su, R., & Basile, F. (2020). Makespan optimization using timed petri nets and mixed integer linear programming problem. *IFAC-PapersOnLine*, *53*(4), 129–135.
Weiss, S., & Stolletz, R. (2015). Buffer allocation in stochastic flow lines via sample-based optimization with initial bounds. *OR spectrum*, *37*(4), 869–902.
Zhang, M., & Matta, A. (2020). Models and algorithms for throughput improvement problem of serial production lines via downtime reduction. *IISE Transactions*, 1–15.