IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Genetic-based Approach for Minimum Initial Marking Estimation in Labeled Petri Nets

HICHEM KMIMECH[1], ACHRAF JABEUR TELMOUDI[1], LAYTH SLIMAN [2], AND LOTFI NABLI.[1]
[1]University of Monastir, Research Laboratory of Automatic Signal and Image Processing
[2]EFREI ÉCOLE D'INGÉNIEUR DES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION, France
Corresponding author: H. KMIMECH (e-mail: hichemkmimech@gmail.com).

ABSTRACT Computing the minimum initial marking (MIM) in labeled Petri nets (PN) while considering a sequence of labels constitutes a difficult problem. The existing solutions of such a problem suffer from diverse limitations. In this paper, we proposed a new approach to automatically compute the MIM in labeled PNs in a timely fashion. We adopted a genetic-based algorithm to model the MIM problem. The choice of such an algorithm is justified by the nature of the MIM process which belongs to the NP-hard class. We experimentally showed the effectiveness of our approach and empirically studied the initial marking quality in particular.

INDEX TERMS Labeled Petri Nets, Minimum Initial Marking, Label sequence, Genetic algorithms, Optimization

I. INTRODUCTION

LABELED Petri Nets (PNs) have been proposed as a fundamental modeling method for discrete event (states oriented) systems in a wide variety of applications such as manufacturing systems, process-based systems, and so on [1], [2]. One of the major studied problems in labeled PNs concerns the minimum Initial Marking (IM) computation while considering a given label sequence [2], [3]. This estimation is carried out by computing the Firing Sequence (FS) of a well-known length, while ensuring a minimum use of resources. The required FS computation process is interesting as it is used in different domains and applications (manufacturing systems, services-based systems, etc.). However, the process of minimum IM estimation in Labeled PNs is a difficult and an NP-hard problem [2].

The minimum IM problem in (labeled) PNs has been receiving a particular attention in the recent decades. Different approaches have been proposed [2], [4]–[6]. While being interesting, these approaches have different limitations and do not go further in ensuring a minimal use of resources. The main objective of this paper is to develop a new approach to automatically compute the minimum IM in labeled PNs in a timely fashion. We adopt a Genetic Algorithm (GA) to address the minimum IM problem. In fact, the GA is a powerful tool to deal with combinatorial problems, which is successfully applied in many other research domains (services compositions, manufacturing systems, cloud resource allocation, and so on.). This success motivates our choice of GA to solve the minimum IM computation problem, which has similar properties in terms of large scale and problem complexity.

The rest of this paper is organized as follows: In Section II, we present a review of the state of the art and the basic concepts of the labeled PN. In Section III describes the minimum IM estimation problem in labeled PNs and its modeling process using genetic algorithms. The empirical studies of the proposed approach is provided in Section IV. In particular, we accentuate the benefits of our approach compared to the related work. Section V concludes this paper and presents some future works.

II. BACKGROUND AND RELATED WORK

In this section, we introduce first an overview of genetic algorithms (GA). Then, we present a summary of the related work. Finally, we describe the basic concept of the LPN that will be used throughout the paper.

A. OVERVIEW OF GENETIC ALGORITHMS

Genetic algorithms (GA) are adaptive, heuristic-based search methods [7], [8]. They are usually used to solve NP-hard

optimization problems like ours. As stated in [8], the implementation of GA basically requires the following elements:

- Data encoding: Through this step, each element in the search space defined by specific data structure, called a genotype or chromosome (genomes) [7]. We distinguish several encoding methods, such as binary, integer, real. The choice of encoding is tightly related to the problem to be solved.
- Generation of the initial population: this step consists of generating of an initial set of individuals to evolve at random according to a uniform distribution.
- Defining of evaluation function: The evaluation function (fitness function) assigns a quality value for each individual in the population. This value allows to measure the performance of each individual to assess its quality and if it is well adapted to its environment [8].
- Choice of genetic operators: This is the set of operation to be applied to individuals: selection, crossing and mutation. The first operator allows select individuals that will be merged to produce new individuals. The mutation makes it possible to change a genome of individuals with a certain probability. The goal of such a change is to circumvent the problem of local optimums and ensure the diversity of the population [8].

The power of GA lies in its randomness. Indeed, the GA execution start with generating an initial population. This population consists of a set of chromosomes where each one of them is evaluated throughout fitness function. The best individuals in this population are selected to undergo genetic operations (crosses, mutations) and a new population of solutions is produced for the next generation. This process continues generation after generation, until the stopping criterion is reached, often the maximum number of generations [7].

B. RELATED WORK

Several approaches are proposed to deal with the minimum IM (initial marking) computation problem in Petri nets. The work presented in [9]–[11] introduces a technique for identifying the markings set that are compliant with a sequence of labels. This technique considers the case in which nondeterminism is due to the presence of transitions that share the same label and that can be simultaneously enabled. The authors of [12] developed a recursive algorithm for estimating the least-cost transition firing sequence(s) based on the observation of a sequence of labels produced by transition activity in a given labeled Petri net.

In [2], Lingxi *et al.* introduce a recursive algorithm finding all of the initial markings that allow for the firing of at least one sequence of transitions that is compliant with the sequence of labels and has the least total number of tokens. In addition, they define heuristics that can further reduce the complexity of this algorithm at the cost of obtaining a subset or an approximation of the minimum initial markings.

In [13], te authors deals with the problem of identifying a Petri net that models the unobservable behaviour of a system from the knowledge of its dynamical evolution. We assume that a partial Petri net model that represents the observable behaviour of a system is given in which all the transitions are observable. An identifier monitors the system evolution and records the observed transition sequence (and possible corresponding markings). Some unobservable transitions modelling the unknown system behaviour are identified from the transition sequence by formulating and solving integer linear programming problems. These identified unobservable transitions together with the given partial Petri net model characterize the whole system, including observable and unobservable behaviour.

The work presented in [14] proposes a labelled PN-based approach to specify system's fault diagnosis. In such an approach, the faults are defined by unobservable transitions and the unobservable subnet is acyclic. A new specification is introduced called an overall fault status, which indicates the occurrence of faults from a global system perspective.

In [15], the authors address the marking estimation problem in labelled Petri PN whose initial marking is known to belong to a given convex set, in the presence of silent transitions (i.e., transitions labelled with the empty word) and indistinguishable transitions(i.e.,transitions sharing the same label with other transitions).First, we demonstrate that all sets of markings consistent with a given sequence of observations can be described in linear algebraic terms (as a union of convex sets); subsequently, this observation is used to construct (offline) a marking observer under appropriate boundedness assumptions. Using the marking observer we show how to derive, at design time, a state feedback control law under the assumption that all transitions sharing a label can be enabled or disabled simultaneously as a group; this way, the most burdensome part of the computations is performed offline.

In [6], Ziyue *et al.* propose an estimation method of the labeled PN marking using the representative marking graph concept. This method takes into account the unobservable transitions by classifying them into two groups: pseudo-observable and strictly unobservable transitions. The unobservable reach of a marking can be characterized by the union of the strictly unobservable reach of several basis markings, called representative markings, in the unobservable subnet. The set of consistent markings can be characterized by a linear algebraic system based on those representative markings.

The authors of [16] propose a heuristic algorithm for the minimum initial marking problem of Petri nets. Given a Petri net and a firing count vector X, this algorithm is able to find an initial marking, with the minimum number of tokens, for which there is a transitions sequence such that each transition t appears exactly X(t) times. In [17], Maria Paola *et al.* proposed a recursive algorithm for initial marking estimation given a sequence of observed labels in a Markovian PN setting. This algorithm considers the case of unobservable transitions.

More recently, [18] proposes an extension of [2] with unobservable transitions, while introducing algorithms for the MIM estimation (MIM-UT). In particular, it assume that the

Petri net structure is given and the unobservable transitions in the net are contact-free. An algorithm is developed to find the set of MIM(s) with polynomial complexity in the length of the observed label sequence. Two heuristic algorithms are also proposed to reduce the computational complexity.

In [19], heuristic based algorithms was developed to find the set of minimum initial markings with the complexity that is polynomial in the length of the observed label sequence. While being interesting, the proposed heuristic algorithms allows to find a partial/approximated set of solutions.

The various methods mentioned above come up against the complexity of the MIM problem classified NP-hard and consequently with long computation due to the combinatorial explosion of the solution space. For that purpose, we propose an approach of both to propose an optimal solution or close to the optimal one with less temporal cost.

C. LABELED PETRI NET MODEL

The Petri net (PN) has been developed as a mathematical model of the discrete event (states oriented) systems in a wide variety of applications such as manufacturing systems, process-based systems, and so on [1]. Labeled PN is an extended PN model in which an eventual label is assigned to each transition [2].

Definition 2.1 (Labeled PN): A labeled PN LN can be formally defined as the following triplet: $LN = (N, L, TL)$, where:

- N is a PN model defined as the following quadruplet:$N = (P, T, Pre, Post)$, where:
  - $P = \{p_1, ..., p_n\}$ is a set of places.
  - $T = \{t_1, ..., t_m\}$ is a set of transitions.
  - $Pre : P \times T \rightarrow \mathbb{N}$ is the weight function of the transitions' incoming arcs.
  - $Post : T \times P \rightarrow \mathbb{N}$ is the weight function of the transitions' outgoing arcs.
- $L = \{l_1, ..., l_k\}$ is a set of labels.
- $TL : T \rightarrow L$ is the transition labeling function. It assigns a label for each transition.

It should be emphasized that the weighting value of an incoming (Pre) or an outgoing (Post) arc is equal to 1 when it is not specified in PN model.

Marking a (labeled) PN consists in assigning, for each place p, a finite number of token. We describe a marking M of a labeled PN $LN = (N, L, TL)$ as the following vector:

$$M = \begin{bmatrix} m_0 \\ \vdots \\ m_i \\ \vdots \\ m_n \end{bmatrix} \quad (1)$$

Where m_i (M[p]) denotes the number of taken in the place p.

1) Transition firing rules

The firing of a transitions sequence allows to change the state of the labeled PN from a given marking to another one. For that purpose the following firing rules are used:

- Rule 1: t is said to be active (can be fired) if and only if the following condition is true:

$$\forall p \in P, M[p] \geq Pre(p, t) \quad (2)$$

- Rule 2: the firing of t produces a new marking M' computed as follows:

$$\forall p \in P, M'[p] = M[p] - Pre(p, t) + Post(t, p) \quad (3)$$

2) Initial marking computation of Labeled PN

As claimed in [20], [21], it is possible to compute the required initial marking of a (labeled) PN from which a firing sequence can be executed. This is done according to the aforementioned transitions firing rules and using the procedure computeInitMarking (see Algorithm 1). More precisely, this procedure applies the following formula iteratively to derive the required IM:

$$\forall p \in P, M_j[p] = M_{j-1}[p] + Pre(p, FS[j]) - Post(FS[j-1], p) \quad (4)$$

where j is an iterator that varies from 1 to n. M_j and M_{j-1} denote respectively the marking computed at the iteration j and $j-1$. FS is a one dimensional vector, where each element of it is transition. The length of this vector is the number of the labeled PN transitions.

Algorithm 1 The IM computation algorithm

```
1:  procedure COMPUTEINITMARKING(FS, LN)
2:      for p in P do
3:          M_0(p) ← 0;
4:      end for
5:      m ← getLength(FS);
6:      j ← 1;
7:      while j ≤ m do
8:          for p in P do
9:              M_j[p] ← M_{j-1}[p] + Pre(p, FS[j]) − Post(FS[j − 1], p);
10:         end for
11:         j ← j + 1;
12:     end while
13:     return M_{j-1};
14: end procedure
```

III. MINIMUM IM COMPUTATION IN LABELED PN USING GENETIC-BASED APPROACH

In this section, we present a detailed description of the minimum IM estimation problem in the context of Labeled PNs. First, we introduce a statement and a formulation of the problem. Then, we propose a genetic-based method to model and solve the IM problem.

A. MINIMUM IM COMPUTATION PROBLEM IN LABELED PN

The minimum IM computation in labeled PNs is an optimization problem where the goal is to generate the firing transitions sequence FS producing the IM having the smallest IM cost, while considering a label sequence.

Definition 3.1: We formally define this problem as the following triplet (LN, LS, IMC), where:

- $LN = (N, L, TL)$ is a labeled PN.
- LS is a label sequence defined as follows: $LS = l_1 l_2 \dots l_k$, where $l_i \in L$ (1≤i≤k).
- IMC is the following optimization function defined using the computeIMC algorithm (see Algorithm 2). Algorithm 2 takes labeled PN and firing sequence as input. It provides the IMC cost of such a firing sequence as output while calling to Algorithm 1 (see line 2) for computing the corresponding initial marking. For a given labeled PN LN and a firing sequence LN, the $IMC(FS, LN)$ value is computed as follows:

$$IMC(FS) = \sum_{p \in P} IM(FS)[p] \tag{5}$$

This formula is defined by Algorithm 2 by a loop 'for'(see lines 4, 5 and 6).

Algorithm 2 The IM computation algorithm

```
1: function COMPUTEIMC(FS, LN)
2:     IM ← computeInitMarking(FS, LN);
3:     IMC ← 0;
4:     for p in P do
5:         IMC ← IMC + IM[p];
6:     end for
7:     return IMC;
8: end function
```

Definition 3.2: A firing sequence of transitions is said to be a valid solution of the minimum IM problem in a labeled PN if and only if the following condition is met:

$$IMC(FS) \leq \delta \tag{6}$$

where δ is the IM quality factor. This factor is a natural number defined by experimentation.

Proposition 3.1: The minimum IM computation process is an NP-hard problem.

Proof: We consider a labeled PN and a label sequence LS of length p. We denote by m the number of transitions sharing the same label. There are m^p possible combinations (each combination represents a firing sequence) in the search space. It is clear that this state-space increases exponentially with the values of m and p. According to [22], finding the optimal (minimizing the IMC function) combination from the possible ones in the search space leads to an NP-hard problem. ■

B. GENETIC-BASED ALGORITHM FOR MINIMUM IM ESTIMATION IN LABELED PN

Several approaches can be used to model the minimum IM (initial marking) problem in Labeled PNs that we defined in the previous section, such as the genetic algorithm [7], [23] and the sequential method. In our work, we aim to propose a genetic-based algorithm (see Algorithm 3) to generate an IM of a given labeled PN in a timely fashion. Such an IM needs to:

- meet all the label sequence constraints;
- minimize the objective function (IMC).

In the sequel, we shall consider a labeled PN including n transitions, T_1,..., T_n and k labels l_1,..., l_k. The transitions sharing a label l are stored in the same array denoted as V_l.

1) Genome encoding

To search for a solution using the genetic algorithms, we have to encode the minimum IM problem with an appropriate genome. As illustrated in Figure 1, the genome is represented by an array of transition indexes (an index is an integer). It is an individual in the population, which represents a firing sequence of the labeled PN (see Figure 1). An element $g[i]$ represents a transition that is encoded as an integer and labeled with $LS[i]$. The size of the array g is equal to the length of the label sequence LS. The positions of the array represent elements of LS.

2) Fitness function and genetic operators

The second step in genetic modeling is the definition of the evaluation (fitness) function and genomes as well as choice of the genetic operators. For the fitness function, we use the following one:

$$F(g) = IMC(g) \tag{7}$$

The genetic operators are the following: selection, crossover and mutation. The selection operator selects the best two genomes for the crossing [23]. We adopted the roulette wheel selection method. This method returns the two genomes with the lowest fitness value for the crossover. The crossover operator allows to merge two genomes to produce two new genomes [23]. In our work, we apply the single-point crossover operator that randomly generates a cut-point (see Figure 2). This cut-point is a natural number belonging to the interval [0, n-1] (where n is the length of the genome).

The mutation operator makes it possible to modify a gene of an existing genome by another [23]. The mutation of a genome is triggered with a very low probability p_m often belonging to the interval [0.001, 0.01]. The appropriate value of the probability p_m is empirically determined. The mutation operator ensures the diversity of the population (a set of genomes). It also ensures that any genome of the search space can be reached.

3) Genetic algorithm

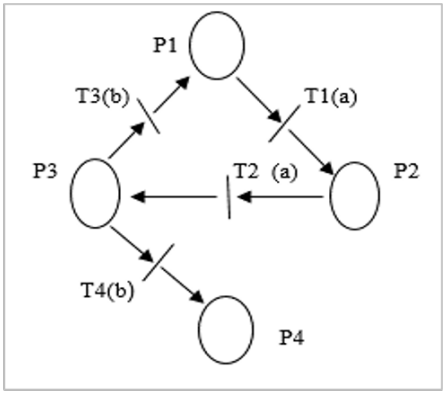
The genetic algorithm we used to model and find a solution for the minimum IM problem is performed in three steps (see

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2967342, IEEE Access

(a) A label sequence LS=aabab

T1	T1	T3	T1	T4

(b) A solution in the simulated annealing research space is an array of transitions index.



(c) A sample labeled PN

FIGURE 1: The genome encoding method of the minimum IM problem

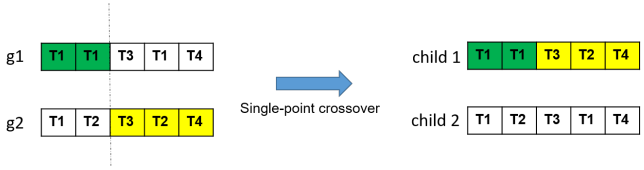


FIGURE 2: The used single-point crossover operator

algorithm 3). The first step is an initialization (from line 1 to line 3). The second step (from line 4 to line 10) attempts to find a valid solution (firing sequence of the labeled PN). The third step (from line 11 to line 13) is a finalization in which we verify the existence of a valid solution for the problem.

Algorithm 3 A genetic-based algorithm

```
1:  initialize(population);
2:  evaluate(population);
3:  cgen← 1;
4:  while cgen ≤ maxgen ∧ checkValidity()=false do
5:      selection();
6:      crossover();
7:      mutation();
8:      evaluate(population);
9:      cgen← cgen+1 ;
10: end while
11: if stopCondition() = true then
12:     return argmin_{g∈population} F(g);
13: end if
```

At the first step, we randomly generate an initial population including a fixed number of genomes. Next, we compute the fitness value of each genome. We initialize a generation counter, $cgen$, to 1. The second step is materialized by a loop with the following condition:

$$cgen \leq maxgen \wedge checkValidity() = false \quad (8)$$

The $checkValidity$ function checks the validity condition given in Definition 3.2. At each iteration of the loop, we apply the selection, crossover and mutation operators. Next, each new genome generated is evaluated. At the end of each iteration, the value of cgen is incremented by 1. The execution of the loop stops in the two following cases:

- the value of the variable $cgen$ is larger than the maximum number of iterations $maxgen$. In this case, there is no solution for the problem.
- the algorithm returned a valid solution for the problem ($checkValidity() = true$).

Once the second step has completed, the genome validation condition is rechecked. If this condition is true ($checkValidity() = true$), our algorithm returns the valid genome with the lowest fitness value.

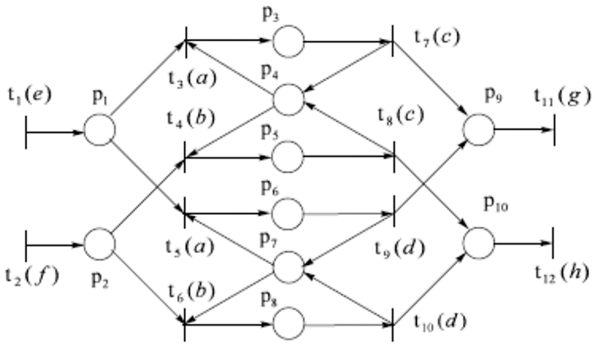


FIGURE 3: The studied labeled PN modeling two parallel machine

IV. EMPIRICAL STUDIES

In this section, we propose the following empirical studies to evaluate the effectiveness of the proposed approach:

- Studying the impact of the quality factor δ
- Studying the benefits of the proposed approach compared to the related work.

These studies were performed on a computer with an Intel Core i5 processor, 4 GB of RAM, and running under a Windows 8 professional edition. For each experiment, our

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2967342, IEEE Access

genetic-based algorithm was executed 50 times and the average value was finally reported. As a case study, we used a real labeled PN model (see Figure 3) and the labeling sequence:

$$LS = eeffabcddcbabbccaaddgghhdcbaeeffabcddcba$$

Some of the parameters of our genetic-based algorithm needed to be configured. These parameters are listed below:

- the population size.
- the mutation probability p_m.
- the maximum number of generations $maxgen$.
- the quality factor δ.

The population size was fixed to 100 for all the experimentation. The mutation probability p_m was fixed to 0.001, while the maximal number of generations was set to 100. The best value of the quality factor δ was also determined by experimentation (more details will be shown in Section IV-A).

A. STUDYING THE IMPACT OF THE QUALITY FACTOR

The goal of this experiment is to determine the best value of the quality factor δ. The best value of δ is the smallest value that allows to generate a solution for the MIM problem in a timely fashion way. For that purpose, the following experimentation approach is adopted:

- step 1: we use the label sequence of length 40 (see Figure 3b).
- step 2: we vary the value of δ from 0 to 12.
- step 3: we compute the number of found solutions.

Figure 4 reports the results of this experiment. As shown by this graph, the number of feasible solutions decreases when δ decreases. This can be explained by the fact that with δ becoming stricter, the existence probability for a feasible solution declines. Some of the solutions are rejected since they don't satisfy the stop condition of the genetic algorithm. Starting from the value 9, the genetic algorithm succeeds to generate a solution. There is no solution when δ is strictly smaller than 9. At the same time, when the value of δ increases, the quality of the generated solution can be decreased. For instance, the IMC value of the solution generated with the value 9 of δ is better than that generated with the value 12.

As a conclusion, the quality of the generated solution is influenced by the value of δ. In what follows, we choose the value 9 of δ. This value ensures a good IMC value for the generated solution in a timely fashion way.

B. STUDYING THE BENEFITS OF THE PROPOSED APPROACH

In this section, we present an evaluation of the benefits of the proposed approach compared to Lingxi's heuristic algorithm that we presented in Section II-B. The evaluation is carried out as follows:

- step 1: we vary the length of the label sequence from 0 to 40.

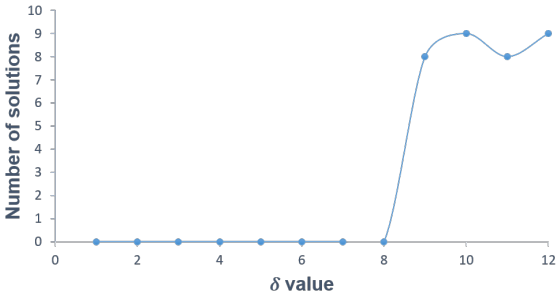


FIGURE 4: Studying the impact of the quality factor

- step 2: we determine the computation times, number of estimated minimum IM (MIM) and the quality of the generated solution of the two approaches.

The sequential method starts by generating all the transition sequences that are compliant with the label sequence and it then selects the appropriate one having the minimum quality cost. As said above, the number of all possible transition sequences increases exponentially with the length of the label sequence. Subsequently, exhaustively creating all these sequences in order to obtain the optimal minimum IM is very time-consuming, which makes it an impractical method. This confirms the empirical results reported in Figure 5.

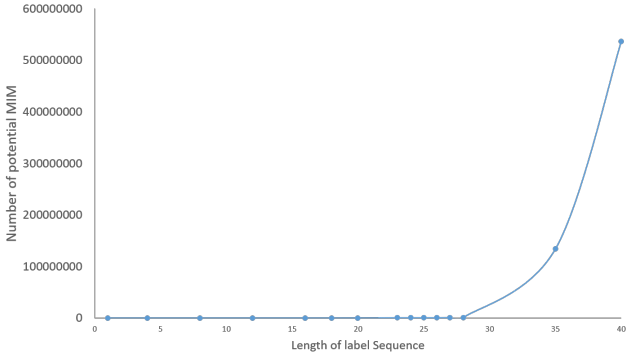


FIGURE 5: The evolution of the number of potential MIM with the length of labels sequence.

The results also show that our genetic-based approach outperforms the two others methods (sequential method and Lingxi's heuristic algorithm) not only in terms of computation times/number of estimated MIM, but also in terms of IM cost thanks to the quality factor δ (see figures 6 and 7). Our approach is able to provide a good (in terms of cost) minimum initial marking of a labeled PN in a timely fashion.

It is worth mentioning that the computation time does not depend on the number of transitions sharing the same labels. It depends basically on the following parameters: the population size, the maximal number of generations (maxgen) and the length of the sequence of labels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2967342, IEEE Access

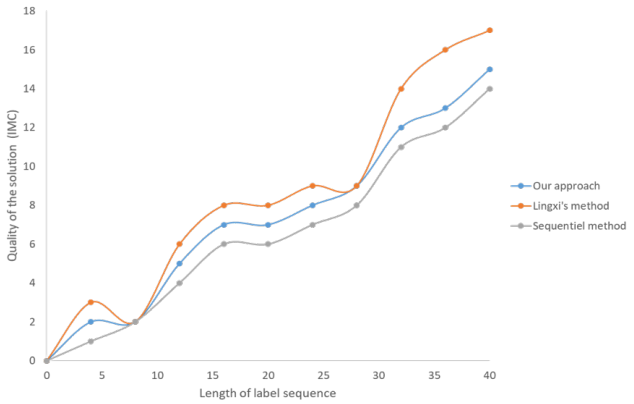


FIGURE 6: Results of studying the benefits of our approach compared to the Lingxi's heuristic algorithm and sequential method in terms of quality of the solution.

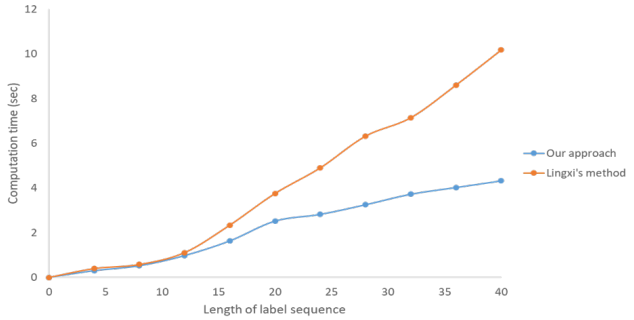


FIGURE 7: Results of studying the benefits of our genetic-based approach compared to the Lingxi's heuristic algorithm in terms of computation time of the solution.

V. CONCLUSION

In this paper, we first defined the problem of minimum IM computation in the labeled PNs. In particular, we proposed a quality factor of the initial marking. The goal of this factor is to increase the cost of the PN initial marking. Second, we proposed a genetic-based method to model and solve the IM problem. Finally, we conducted empirical studies to evaluate the effectiveness of the proposed work compared to the related approaches.

For future work, we are making progress on applying the proposed method on real case studies. We plan to study the problem of computing the firing sequence that allows to a minimum resources consumption. This problem is closely similar to the one studied in this paper.

REFERENCES

[1] T. Murata, "Petri nets: Properties, analysis and applications," Proceedings of the IEEE, vol. 77, no. 4, pp. 541–580, 1989.
[2] L. Li and C. N. Hadjicostis, "Minimum initial marking estimation in labeled petri nets," IEEE Transactions on Automatic Control, vol. 58, no. 1, pp. 198–203, 2013.
[3] T. WATANABE, T. TANIDA, M. YAMAUCHI, and K. ONAGA, "The minimum initial marking problem for scheduling in timed petri nets," IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, vol. 75, no. 10, pp. 1407–1421, 1992.
[4] J. Rodríguez-Beltrán and A. Ramfrez-Trevino, "Minimum initial marking in timed marked graphs," in Systems, Man, and Cybernetics, 2000 IEEE International Conference on, vol. 4. IEEE, 2000, pp. 3004–3008.
[5] T. WATANABE, Y. MIZOBATA, and K. ONAGA, "Minimum initial marking problems of petri nets," IEICE TRANSACTIONS (1976-1990), vol. 72, no. 12, pp. 1390–1399, 1989.
[6] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Marking estimation in labelled petri nets by the representative marking graph," IFAC-PapersOnLine, vol. 50, no. 1, pp. 11 175–11 181, 2017.
[7] M. Srinivas and L. M. Patnaik, "Genetic algorithms: a survey," Computer, vol. 27, no. 6, pp. 17–26, 1994.
[8] K.-F. Man, K. S. Tang, and S. Kwong, Genetic algorithms: concepts and designs. Springer Science & Business Media, 2001.
[9] A. Giua, D. Corona, and C. Seatzu, "State estimation of λ-free labeled petri nets with contact-free nondeterministic transitions," Discrete Event Dynamic Systems, vol. 15, no. 1, pp. 85–108, 2005.
[10] A. Giua, J. Júlvez, and C. Seatzu, "Marking estimation of petri nets based on partial observation," in American Control Conference, 2003. Proceedings of the 2003, vol. 1. IEEE, 2003, pp. 326–331.
[11] D. Corona, A. Giua, and C. Seatzu, "Marking estimation of petri nets with silent transitions," in Decision and Control, 2004. CDC. 43rd IEEE Conference on, vol. 1. IEEE, 2004, pp. 966–971.
[12] Y. R. L. Li and C. N. Hadjicostis, "Least-cost transition firing sequence estimation in labeled petri nets," in IEEE Conference on Decision and Control. IEEE, 2006, pp. 416–421.
[13] G. Zhu, Y. Wang, and Y. Wang, "Model identification of unobservable behavior of discrete event systems using petri nets," Journal of Control Science and Engineering, vol. 2019, 2019.
[14] G. Zhu, L. Feng, Z. Li, and N. Wu, "Online fault diagnosis of discrete event systems modeled with labeled petri nets using an overall fault status," 2018.
[15] M. P. Cabasino, C. N. Hadjicostis, and C. Seatzu, "Marking observer in labeled petri nets with application to supervisory control," IEEE Transactions on Automatic Control, vol. 62, no. 4, pp. 1813–1824, 2016.
[16] M. Yamauchi and T. Watanabe, "A heuristic algorithm for the minimum initial marking problem of petri nets," in Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on, vol. 1. IEEE, 1997, pp. 245–250.
[17] M. P. Cabasino, C. N. Hadjicostis, and C. Seatzu, "Initial marking estimation in labeled petri nets in a probabilistic setting." in CDC, 2014, pp. 6725–6730.
[18] K. Ruan, L. Li, and W. Wu, "Minimum initial marking estimation in labeled petri nets with unobservable transitions," IEEE Access, vol. 7, pp. 19 232–19 237, 2019.
[19] L. L. K. Ruan and W. Wu, "Minimum initial marking estimation in labeled petri nets with unobservable transitions," IEEE Access, vol. 7, pp. 19 232–19 237, 2019.
[20] A. Giua and C. Seatzu, "Observability of place/transition nets," IEEE Transactions on Automatic Control, vol. 47, no. 9, pp. 1424–1437, 2002.
[21] A. Giua, "Petri net state estimators based on event observation," in Decision and Control, 1997., Proceedings of the 36th IEEE Conference on, vol. 4. IEEE, 1997, pp. 4086–4091.
[22] M. R. Garey and D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman & Co., 1990.
[23] H.-L. Fang, "Genetic algorithms in timetabling and scheduling," Ph.D. dissertation, Citeseer, 1994.

HICHEM KMIMECH is urrently working as Professor at the Department of computer science , University of Sousse. His research interest include Artificial Intelligence, Computer Engineering and Control Systems Engineering. Their current project is 'Control and optimization in industry and service'.

PROF. LOTFI NABLI is currently working as Full Professor at the Department of Electronic Engineering, University of Monastir. His research interests include Artificial Intelligence, Computer Engineering and Control Systems Engineering.