Minimum Initial Marking Estimation of Labeled Petri Nets Based on GRASP Inspired Method (GMIM)

Amir Abdellatif, Achraf Jabeur Telmoudi, Patrice Bonhomme, Lotfi Nabli

Minimum Initial Marking Estimation of Labeled Petri Nets based on GRASP inspired method (GMIM)

Amir Abdellatif [a], Achraf JabeurTelmoudi[b]*, Patrice Bonhomme [c], and Lotfi Nabli[a]

[a] LARATSI Laboratory, ENIM, University of Monastir, 5000 rue Ibn Jazzar, Monastir 5035, Tunisia

[b] LISIER Laboratory, The National Higher Engineering School of Tunis (ENSIT), University of Tunis, 5 Avenue Taha Hussein, BP 56, 1008 Bâb Manara, Tunisia

[c], INSA CVL, LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002, 3 rue de la Chocolaterie, CS 23410, 41034 Blois, France (e-mail: patrice.bonhomme@insa-cvl.fr)

* Corresponding author, (e-mail: achraf-j.telmoudi@ieee.org)

Minimum Initial Marking Estimation of Labeled Petri Nets based on GRASP inspired method (GMIM)

ABSTRACT: This paper deals with the problem of estimating the Minimum Initial Marking (MIM) of Labeled Petri Nets (L-PN). By the observation of a sequence of labels, we determine the set of possible MIMs related to a given L-PN through an approach based on GRASP (Greedy Randomized Adaptive Search Procedure) inspired method – GMIM. The objective is to get the maximum of feasible MIMs by exploring the search space and giving best solutions for real time cyber systems in short time. We consider four basic assumptions during the reasoning: (*i*) the L-PN structure is known; (*ii*) for each transition of L-PN, a label is associated, (*iii*) the label sequence is known, and (*iv*) all transitions of L-PN are observable. We show the validity and efficiency of our approach by applying the proposed GMIM metaheuristic to two validation examples: Initialization of two parallel machines (example widely cited in literature) and resources allocation in a monitoring problem via mobile robot network.

Keywords: Labeled Petri Nets, Minimum Initial Marking, Estimation, GRASP inspired method (GMIM).

1 Introduction

Fast estimating and system state prediction are indispensable in a rapidly changing environment characterized by an economic context unstable with hard competition. The resource allocation problem is among the top challenging problem in project management. In the literature, many approaches are proposed to resolve such problem. Among them, divers methods and methodologies based on the Minimum Initial Marking (MIM) Estimation in Labeled Petri Net (L-PN) [1,2,3,4].

In [8], marking estimation problem of L-PN under a label sequence observation constraint is developed. Authors of this work prove that the determination of the feasible markings set is possible under some conditions on the L-PN structure when the size of the linear system does not depend on the size of the observed label sequence. Basile et al. propose an algorithm for estimating the marking of a Timed Petri Net (T-PN) with unobservable transitions [5]. In this work, due to the "modified state class graph" (MSCG), the behavior and possible evolutions of the examined system can be obtained. In the same context, Abdellatif et al. [9] develop a technique for estimating the MIM of a real-time system under partial observation modeled by a P-Time labeled PN. Li et al., in [10], estimated via a labeled Petri net the least-cost planning sequences of a manufacturing system. The evolution of possible marking(s) is determined through the observation of labels sequences. Indeed, the marking evolution is obtained by following the low cost Transition Firing Sequence(s) (TFS).

The work, in [11], develops a recursive algorithm able to find the MIM in L-PN. Authors propose also two heuristics to obtain an approximated subset of MIMs. The disadvantage of this approach is that the execution of the algorithm takes a long time. In order to reduce complexity and give a better solution in less running time, other published researches, with the same problem such [12], [13], have proposed solutions based on heuristics and metaheuristics. In fact, the problem of running time was solved with genetic algorithm metaheuristic to compute the MIMs in short time [13]. However, there are different limitations to these approaches, among them, producing much smaller solutions against exact methods.

The presented work is motivated by real time problems dealing with control, monitoring and minimum resource allocation in complex and cyber systems. In fact, we search to determine the better solution in less running time. The principal aim of this paper is to propose a new perspective in the field of estimating the initial marking (considered unknown) of L-PN. Compared to the state of the art, the proposed approach in this work is based on the GMIM method inspired from the multi-start metaheuristic GRASP used generally for combinatorial problems [14].

The following four basic assumptions will be considered during the reasoning:

(i) The PN structure is known;

(ii) For each transition of L-PN, a label is associated;

(iii) The label sequence is known;

(iv) All transitions of L-PN are observable.

In the next section, backgrounds on PN and L-PN notations will be presented. Section 3 provides the definition of the considered problem. The GMIM method proposed for obtaining the MIM(s) is presented in section 4. In section 5, a validation will be achieved through two problems.

2 Backgrounds on Labeled Petri Net

A. *Petri Net Notation*

In this section, we assume that readers know the Petri Nets. We will present only basic notions that will be used all over this paper. Further details on PNs are available in [15],[16].

The PN is a structure formed by Place/Transition (P/T net) and arcs, such as PN= (*P*,*T*,*Pre*,*Post*); where:

P is a set of *m* places; *T* is a set of *n* transitions.

$Post\colon P \times T \rightarrow \mathbb{N}$ and $Pre\colon P \times T \rightarrow \mathbb{N}$ are respectively the *Post* and *Pre* incidence functions that specify the arcs; *C*=*Post*−*Pre* is the incidence matrix. The *postset* and *preset* of a node $X \in P \cup T$ are denoted ${}^{o}X$ and X^{o}. A marking is a vector $M\colon P \rightarrow \mathbb{N}$ that assigns to each place p_i of PN an integer positive number M(p_i) (number of tokens on the place p_i).

A net system designated by $\langle PN;\ M_0\rangle$ is a *PN* with an initial marking M_0. A transition t is enabled by a marking M, *iff* $M \geq Pre(:,t)$. When transition t is enabled at M, it may fire and producing the marking $M' = M + C(:,t)$.

Let $\sigma_i = t_{i1}t_{i2} \dots t_{ik}$ $(t_{ij} \in T, i \in \{1,2,\dots,h\}, j \in \{1,2,\dots,k\})$ be a transition firing sequence; for example, t_{i2} is the second transition of the i^{th} σ. A marking M is reachable in $\langle N;\ M_0\rangle$ iff there exists a firing sequence σ_i such that $M_0[\sigma_i\rangle M$. The set of all sequences that are enabled at the initial marking M_0 is denoted L(N, M_0); i.e., L(N, M_0)={ $\sigma \in T^*|M_0[\sigma_i\rangle$}, with T^* the set of feasible firing sequence. We use $\overrightarrow{0_n}$(respectively, $\overrightarrow{1_n}$) to denote the n×1 vector of all zeros (respectively, ones). The set of all markings reachable from M_0 defines the reachability set of $\langle N;\ M_0\rangle$ and is denoted $R(N, M_0)$.

Let y be the firing vector that corresponds to σ and $y(t)$ is the total number of occurrences of transition t in σ. For a TFS that contains a single transition t, we use $\bar{t}$ to denote its firing vector (i.e. $\bar{t}$ is a unit vector).

B. *Labeled Petri Net*

A Labeled Petri Nets (L-PN) is a t-uple $\langle PN; M_0; \mathcal{LM}\rangle$; where:

$\langle PN; M_0\rangle$ is a *PN* with an initial marking M_0,

$\mathcal{LM}$ is the labels mapping.

A labels mapping $\mathcal{LM}$ is associated with all transitions of the PN:

$$\mathcal{LM}: T \rightarrow \Omega$$

With, Ω is a finite alphabet, and all transitions labels can be detected by an external observer. In this paper, the set of transitions is partitioned into two subsets. The first subset is called distinguishable transitions and denoted T_{dis}; T_{dis} includes all transitions of the L-PN where the assigned label is not shared with at least one of the others transitions. The second subset is called indistinguishable; the subset is denoted T_{ind} and contains each transition sharing the same label $\zeta \in \Omega$ with several transitions of the L-PN. Two transitions t_x, t_z with $t_x \neq t_z$ will be denoted $t_{x,z} \in T_{ind}$ if: $\mathcal{LM}(t_x)$ =$\mathcal{LM}(t_z)$ = ζ. So, $T = T_{dis} \cup T_{ind}$ and $T_{dis} \cap T_{ind} = \emptyset$.

The extension of the label mapping can be realized over sequences, $\mathcal{LM}: T^* \rightarrow \Omega^*$, recursively as follows:

1. $\mathcal{LM}(t_x) = \zeta \in \Omega$ if$t_x \in$ T,
2. Let σ ∈ T^* and $t_x \in$ T then $\mathcal{LM}(\sigma t_x) = \mathcal{LM}(\sigma)\mathcal{LM}(t_x)$,
3. $\mathcal{LM}(\lambda)$=ε where λ is the empty sequence and ε is the empty word.
4. $\mathcal{L}$= $\mathcal{LM}(t_\alpha)\mathcal{LM}(t_\beta)\ldots\mathcal{LM}(t_\gamma)$ (where, $\{t_\alpha, t_\beta, \ldots t_\gamma\} \in T$) an observed label sequence that can give us a finite number of TFS,

3 Problem formulation

The considered problem is a L-PN with unknown initial marking and an observed label sequence $\mathcal{L}$. The observed label sequence $\mathcal{L}$ may generate a finite set of transition firing sequence, where $\sigma_i = t_{i_1}t_{i_2}\ldots t_{i_k}$ (where, $\mathcal{LM}(t_{i_j}) = l_j \in \mathcal{L}$ and j is the index of the j^{th} transition of σ_i). The objective is to find the set of minimum initial marking(s) that: *(i)* enable the firing of at least one sequence of transitions that is coherent with both Ω and the L-PN structure, and *(ii)* is (are) minimum (i.e., the marking(s) has (have) the lowest cumulative number of tokens). It is possible to find more than one MIM for the same given observation.

Example:

Consider the L-PN represented in figure 1 with a unit weight in all arcs, such that $\{p_1, p_2, p_3, p_4, p_5\}$ is the places set and $\{t_1, t_2, t_3, t_4\}$ is the transitions set. A labeling function is defined such that $\mathcal{LM}(t_1)$ =$\mathcal{LM}(t_3) = a$, $\mathcal{LM}(t_2) = b$, $\mathcal{LM}(t_4) = c$.

t_1 and t_3 are indistinguishable transitions; t_2 and t_4 are distinguishable transitions.

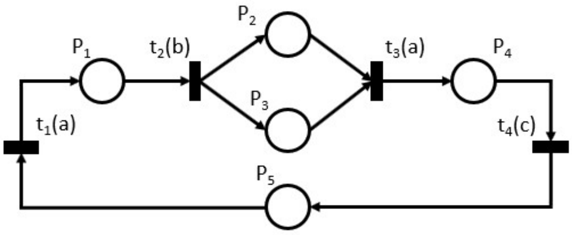


Figure 1: Simple L-PN used to demonstrate basic concepts

Given the following label sequence *£=abac*. The determination of MIM(s) is done through the enumeration of all TFS which is demonstrated in figure 2. According to this sequence, there are four possible *TFS*:$\{t_1t_2t_1t_4; t_1t_2t_3t_4; t_3t_2t_1t_4;\ t_3t_2t_3t_4\}$. The number of TFS is equal to the number of leaves in the tree shown in figure 2. This number increases exponentially compared to the size of the label sequence, and we call it the solution space (**SS**). More precisely, the number of TFS is **SS** = $\prod_{i=1}^{k}\left|T_{\zeta_i}\right|$(where T_{ζ_i}denotes the transitions set associated with label ζ_i, and $\left|T_{\zeta_i}\right|$ is the cardinality of this set); for more detail see [18]. Let's take the label sequence of the first example *£=abac*; **SS**=$|T_a| \times |T_b| \times |T_a| \times |T_c| = 2 \times 1 \times 2 \times 1 = 4$. The exponential growth is related to labels that have more than one transition associated with them $\left|T_{l_i}\right| \geq 2$.

In this example, it is easily to calculate the initial marking and find its all results with respect to *£*: $\{[00012]^T; [00001]^T; [11101]^T; [11100]^T\}$. Without difficulty, we can notice that the MIM is the following$[00001]^T$ because it admits the lowest cumulative number (=1) of token(s) summed over all places.

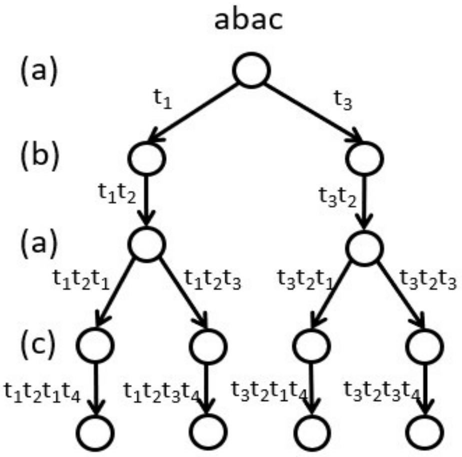


Figure 2: Enumeration of all firing sequences

Obviously, the problem is upper bounded by a polynomial function in terms of the total number of possible firing vector. In this work, our aim is to propose a novel efficiency method reducing the complexity and the state explosion problem. In the next section, the proposed GMIM metaheuristic inspired from the GRASP method will be presented.

Further details on the principle of GRASP (Greedy Randomized Adaptive Search Procedure) method are available in [14].

4 GMIM metaheuristic for MIM estimation

The objective is to have a consistent set of potential solutions (a couple of TFS and MIM) respecting the behaviour of real time cyber systems. The proposal of GMIM algorithm is based on the knowledge of the problem studied in order to generate and ameliorate iteratively the alternative solutions resulting from combinations of existing sequences.

The principle of this method is based on two main functions: the initial marking computation (fitness function) and the mutation operator to generating new TFS.

A. *Fitness Function*

In the literature various methods are proposed to compute the initial marking of TFS in PN. In the sequel we use the exact method proposed by *Giua et al.* (in [6] and [7]) to compute the MIM for a given TFS.

We consider this problem as the following triple (R,TFS,MIM); where:

1. *R*=(*P*,*T*,*Pre*,*Post*) is a PN.
2. TFS: $\sigma_i = t_{i_1} t_{i_2} \ldots t_{i_k}$ (where ($\sigma_i \in$ TFS) for i= {1,..., h}and ($t_{i_j} \in T$) for j = {1,..., k}).
3. MIM is the smallest total number of tokens that can have a TFS.

Inspired from the famous method proposed in [6] and [7] to compute recursively the MIM of a PN, *Li & Hadjicostisin* (in [11]) proposed the following formula to resolve the same problem in a L-PN:

$$M_0^{j+1} = max\{M_0^j + C.y_{j-1}, Pre(:, t_{ij})\} - C.y_{j-1} \qquad (1)$$

Where:

y_{j-1} is the vector of the first $j-1$ elements of σ_i,

M_0^1 is an $\overrightarrow{0_n}$(i.e., *n*-dimensional vector of zeros),

y_0 is an $\overrightarrow{0_n}$(i.e., *m*-dimensional vector of zeros),

M_0^j is the minimum initial marking estimated before firing t_{ij} and,

M_0^{j+1} is the initial marking estimated after firing t_{ij}.

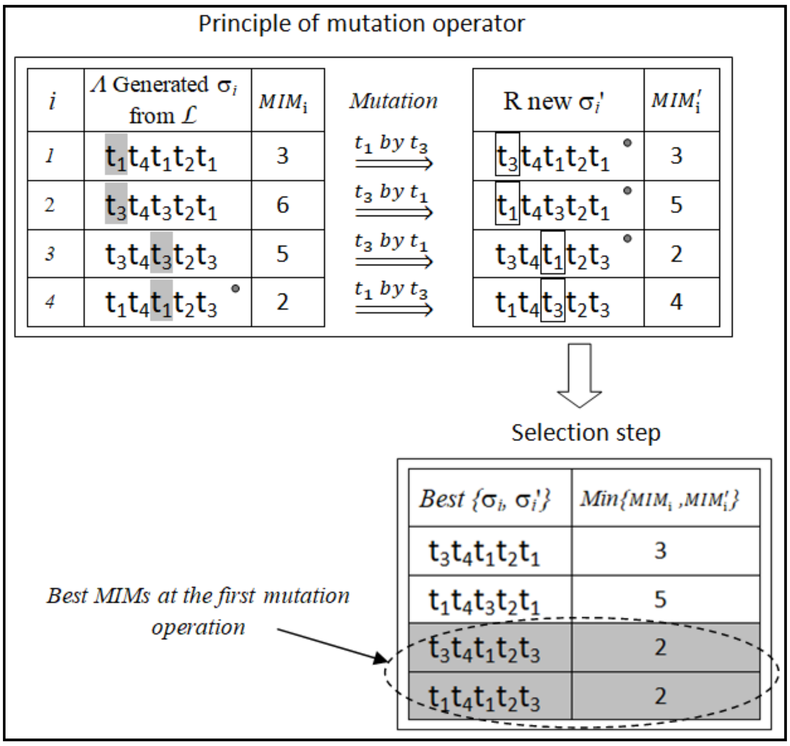
In the GMIM algorithm, equation (1) will be used as the fitness function to calculate the MIM of each generated σ_i.

B. Mutation operator

Considering the L-PN of figure 1 and the following label sequence *ℒ=abaca,* figure 3 presents the evolution of the steps performed in one cycle of the mutation function.

In the beginning, we create a random population (of size *Λ*) of TFS coherent with ℒ and we put them in the first table (in the left of figure 3). The last column calculates the MIM_i corresponding to σ_i by using the equation (1).

After that, each sequence presented in the left table undergoes a mutation on only one random transition by another transition having the same label. Then, we place the new firing sequence σ_i'in the line *i* of the second table (in the right of the figure) and we apply the same equation to compute MIM_i'.
The table of selection step will be completed by (*Best* {σ_i, σ_i'}, $Min\{im_i, im_i'\}$).


Principle of mutation operator
Selection step
Best MIMs at the first mutation operation

i	*Λ* Generated σ_i from ℒ	MIM_i	*Mutation*	R new σ_i'	MIM_i'
1	$t_1t_4t_1t_2t_1$	3	t_1 *by* t_3	$t_3t_4t_1t_2t_1$ ●	3
2	$t_3t_4t_3t_2t_1$	6	t_3 *by* t_1	$t_1t_4t_3t_2t_1$ ●	5
3	$t_3t_4t_3t_2t_3$	5	t_3 *by* t_1	$t_3t_4t_1t_2t_3$ ●	2
4	$t_1t_4t_1t_2t_3$ ●	2	t_1 *by* t_3	$t_1t_4t_3t_2t_3$	4

Best {σ_i, σ_i'}	*Min*{MIM_i, MIM_i'}
$t_3t_4t_1t_2t_1$	3
$t_1t_4t_3t_2t_1$	5
$t_3t_4t_1t_2t_3$	2
$t_1t_4t_1t_2t_3$	2

Figure 3: Mutation operator and selection of best MIMs

C. GMIM metaheuristic algorithm

The formal description of our GMIM algorithm for MIM estimation is shown in figure 4 and Algorithm 1. The whole principle of the proposed GRASP inspired metaheuristic (GMIM) for MIM estimation is given in Figure 4.

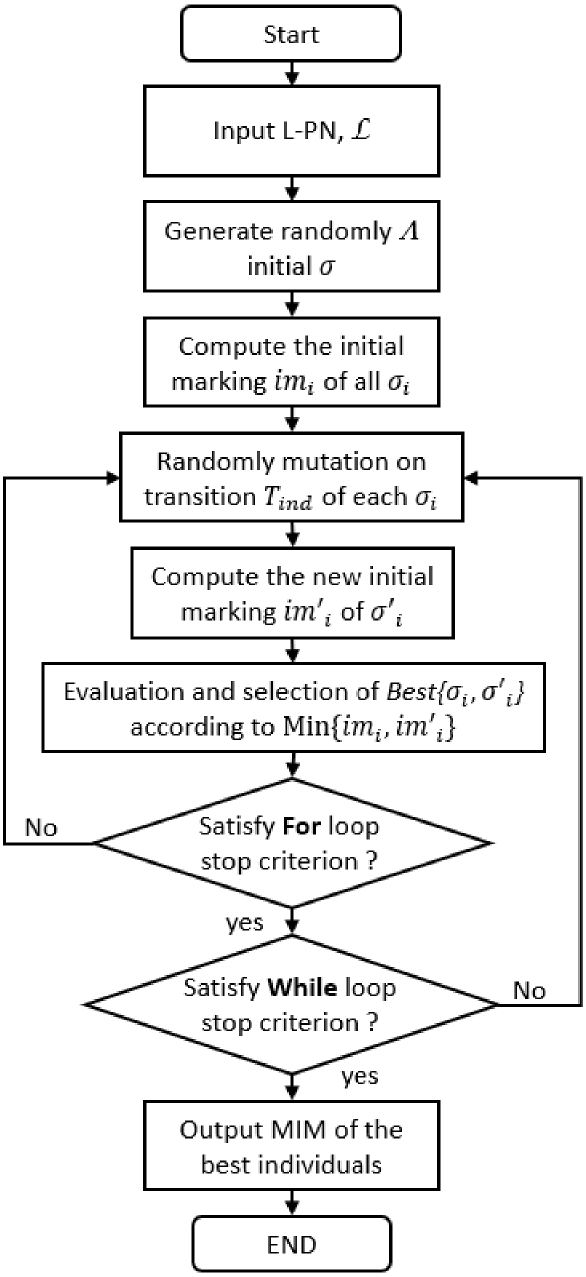

Start
Input L-PN, $\mathcal{L}$
Generate randomly Λ initial σ
Compute the initial marking im_i of all σ_i
Randomly mutation on transition T_{ind} of each σ_i
Compute the new initial marking im'_i of σ'_i
Evaluation and selection of $Best\{\sigma_i, \sigma'_i\}$ according to $\mathrm{Min}\{im_i, im'_i\}$
Satisfy For loop stop criterion ?
No
yes
Satisfy While loop stop criterion ?
No
yes
Output MIM of the best individuals
END

Figure 4: Flowchart of the proposed GMIM

The algorithm 1 is explained in the following lines. Given an L-PN with and observed sequence of labels $\mathcal{L}$= $l_1 l_2 ... l_q$ of length q.

The proposed algorithm begins by storing information in a data structure R of Λ *transition firing sequences* (σ) coherent with $\mathcal{L}$(Step 5). At steps(6, 7 and 8), an evaluation of the marking for initial population (one by one) will be done by performing the Equation (1). im_i is the minimum initial marking of σ_i. The solution of the first execution of this equation provides only the necessary tokens allowing the firing of TFS created in R.

Algorithm: GRASP Minimum Initial Marking (GMIM)

Input: A free labeled Petri net PN and an observable label sequence $\mathcal{L}= l_1 l_2 ... l_q$of length q,

Output: The MIM that has (have) the lowest cumulative number of tokens over all places.

```
1.  Var $IMPROVEMENT$=TRUE
2.  Var $\mathrm{NO_{IMP}} = 0$
3.  MIM = ∞
4.  V_sol = ∅
5.  Generate randomly a set ‘$R$’(of size$\Lambda$) of $\sigma$ coherent with $\mathcal{L}$
6.  For each transition firing sequence $\sigma \in R$ Do
7.      Compute $im$using Equation (1) ($im$ : initial marking belonging to $\sigma$ )
8.  End
9.  While $IMPROVEMENT$ Do
10.     For each transition firing sequence $\sigma \in R$ Do
11.         Look for any $T_{ind}$and apply a random mutation for $\sigma$:
12.         $\sigma \rightarrow \sigma'$ (i.e., $\sigma'$ is the new firing sequence after mutation)
13.         Compute $im'$using Equation (1) ($im'$ : initial marking belonging to $\sigma'$ )
14.         If $im' \geq im$ Then $\mathrm{NO_{IMP}} = \mathrm{NO_{IMP}}+1$
15.         End
16.     End
17.     If $NO_{IMP} = |R|$
18.         $IMPROVEMENT =$ FALSE
19.     End
20. End
21. For all marking in $R$
22.     If $im$ == MIM
23.         V_sol = V_sol ∪ $im$
24.     Else if $im$ < MIM
25.         V_sol = ∅
26.         V_sol = MIM = $im$
27.     End
28. End
```

At step 9, a while loop controlled by the variable $IMPROVEMENT$ is introduced. $IMPROVEMENT$ is a Boolean variable initialized at $TRUE$ in step 1, it will be set to

$FALSE$ in step 18 when no improvement appears all over the population. The while loop continuing, through a randomly mechanism of mutation, running as long as $IMPROVEMENT$ is performed. That means, when the current solution im'_i (in terms of marking) it become better than the previous one im_i (i.e., at the current step (say, i),a new marking is obtained with $im'_i < im_i$; im_i is stored at the previous step (say, $i - 1$)). At step 10, each individual σ_i undergo a mutation operation performed to indistinguishable transition (i.e. we look for label that have more than associated with it, and then we do a random permutation between transitions). The principle of mutation and selection of best solution (*Best* {σ_i,σ_i'}, *Min*$\{im_i,im'_i\}$) is described in the previews subsection.

With:

IMPROVEMENT is a Boolean variable initialized at $TRUE$ in step 1. It will be set to $FALSE$ in step 18 when no improvement accrues all over the population. NO_{IMP} will be incremented when no improvement performed to the new firing sequence marking. MIM variable is initialized to ∞, and V_sol is empty befor we start looking for MIM.

4 Validation examples

The efficiency estimation properties of the proposed GMIM metaheuristic were tested on two examples: initialization of two parallel machines and resources allocation in a monitoring platform via mobile robot network. The required goal is to affirm the performance of the GRASM inspired metaheuristic by referring to the evolutionary strategies proposed in [11] (*Algorithm Li & Hadjicostis, 2013* and *Heuristic A-B Li & Hadjicostis, 2013*). Simulations were performed on a computer with an Intel Core i7 processor, 8 GB of RAM, and running under a Windows 10.

A. Application 1: Initialization of two parallel machines

The figure 5 represents the L-PN model of two parallel working machines [17, p. 132]. All transitions of L-PN are observable. We maintain the same label sequence of length 40, generated randomly in [11], *£=eeffababcddcbabbccaaddgghhdcbaabcdabcdha*. Then, the aim is to estimate the set of minimum initial marking using GMIM and making a comparison with the results of generated solutions provided by the proposed methods of (*Li & Hadjicostis, 2013*) in [11].

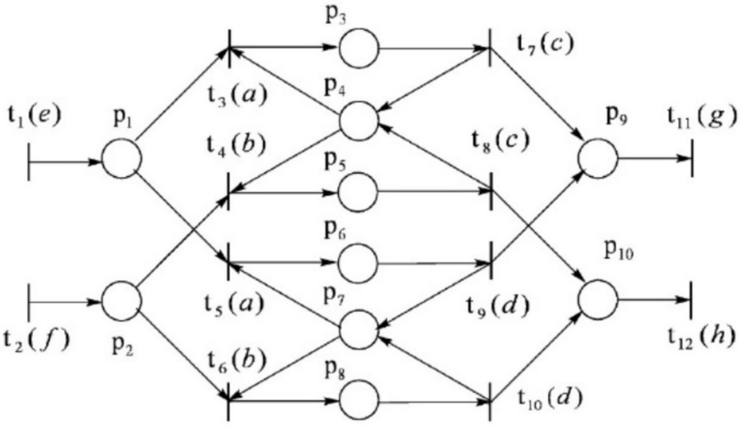


Figure 5: L-PN Model of two parallel working machines

The execution of GMIM is performed with a simulation strategy based on the increment of the size Λ of the initial set R_i of *TFS* (generated randomly). In fact, the first simulation begins with 10 initial marking with respect to $\mathcal{L}$**,** (i.e., Λ_1=10). In the second simulation, we increase Λ to 20 (i.e., Λ_2=20); the set of new initial marking R_2 is generated independently of R_1.We continue to increase Λ_i until the number of MIM will be remarkably stable. For each level (i.e., Λ_i) we accrue twenty simultaneous executions, each one with new randomly input sequences. The simulation results are given by Table 1 and Figure 6.

Table 1: Comparative table

	R_size	*N_ DMIM*	*N_MIME*	*N_MIM*	*Time*
Algorithm of Li & Hadjicostis	-	5	12679	276	149(s)
Heuristic A-B of Li & Hadjicostis	-	3	5760	106	110(s)
GMIM Algorithm	50	2.1	50	4.8	0.25(s)
	100	3.5	100	16.5	0.6(s)
	150	4.6	150	32.9	1(s)
	200	4.8	200	53.9	1.5(s)
	250	5	250	81.5	2.24(s)
	500	5	500	236.7	6.5(s)
	1000	5	1000	684.8	15.82(s)
	2000	5	2000	1579.4	47.8(s)

- *"R_Size"* is the size of randomly generated sequences set,
- *"N_ DMIM"* is the average of the number of different output sequences of MIMs at each *R_Size* level after 20 simulations. For example, for *R_Size=100,* we

obtained as solutions: 4 sub-results with 5 possible sequences of MIMs, 7 sub-results with 4 possible sequences of MIMs, 6 sub-results with 3 possible sequences of MIMs, 2 sub-results with 2 possible sequences of MIMs, and one sub-result with one possible sequence of MIM; so, *N_ DMIM* is calculated as follow:

$$N\_DMIM = \frac{4\times5+7\times4+6\times3+2\times2+1\times1}{20} = 3.5\ ,$$

- *"N_MIME"* is the number of the MIMs estimated after 20 simulations in each *R_Size,*
- *"N_MIM"* is the average of the MIMs obtained after 20 simulations. It is computed with same calculation principle of *N_ DMIM.*
- *"Time"* is the running time.

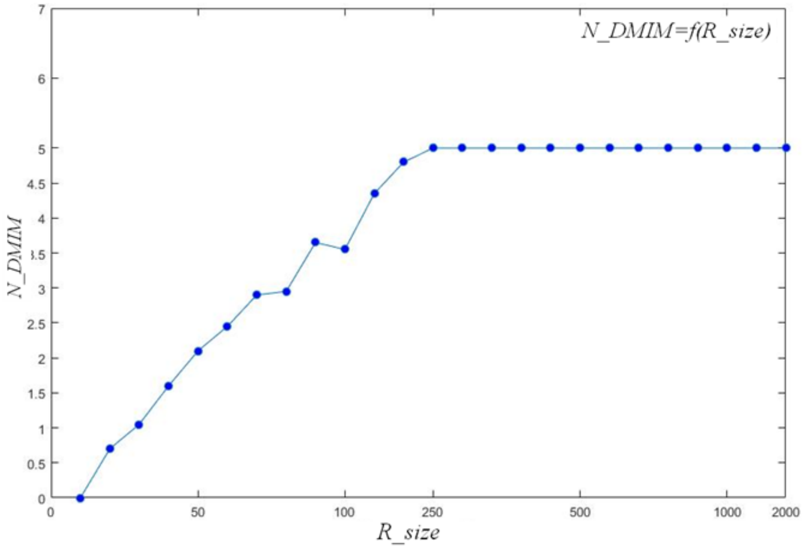
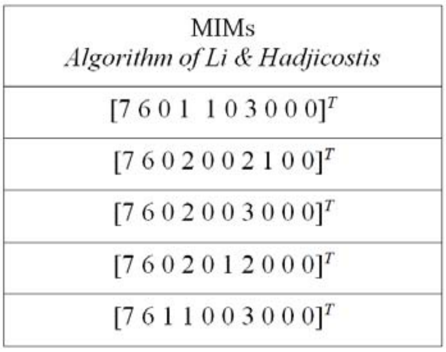


Figure 6: GMIM simulation (application 1) – *R_size* vs. *N_ DMIM*

Before discuss the results of our proposed GMIM metaheuritic, we provide in Figure 7 all possible MIMs obtained by the *Algorithm of Li & Hadjicostis* in [11].



MIMs *Algorithm of Li & Hadjicostis*
$[7\ 6\ 0\ 1\ \ 1\ 0\ 3\ 0\ 0\ 0]^T$
$[7\ 6\ 0\ 2\ 0\ 0\ 2\ 1\ 0\ 0]^T$
$[7\ 6\ 0\ 2\ 0\ 0\ 3\ 0\ 0\ 0]^T$
$[7\ 6\ 0\ 2\ 0\ 1\ 2\ 0\ 0\ 0]^T$
$[7\ 6\ 1\ 1\ 0\ 0\ 3\ 0\ 0\ 0]^T$

Figure 7: All possible MIMs obtained in [11]

According to results shown in table 1 and figure 6, it is clear that when we increase the size of the initial population (*R_Size*), the effectiveness of the proposed metaheuristic becomes more improved in terms of the number of MIMs solutions. Indeed, moving from the first simulation (*R_size*=50) to the simulation with *R_size*=250 a concrete improvement is shown. In fact, the *N_ DMIM* and *N_MIM* increase, respectively, from 2.1 to 5 and from 4.8 to 81.5. Despite that, we can remark that the running time (*Time* =2.24 s) is always applicable for real time systems. In figure 6, from the simulation with *R_size*=250 to the one with *R_size*=2000, we observe easily that there is a stability In terms of the number of *N_ DMIM*(=5); i.e., the *N_ DMIM* provided by the *Algorithm of Li& Hadjicostis* [11]. Also, the efficiency of our GMIM metaheuristic appears in the higher number of possible estimated MIM; it was increased to provide more possible MIM solutions than the proposed *Heuristic A-B* and *Algorithm of Li & Hadjicostis* [11]. In terms of fast estimation, it is clear that running time increases when we amplify the size of *R*. despite that, the running time in all simulations remains so much lower than that of *Algorithm and Heuristic of Li & Hadjicostis* [11] (respectively, 149 s and 110 s).

The major advantage of our approach is that even with a small size of initial number of sequences, we can obtain the 5 different MIMs solutions provided in figure 7. For example, four sub-results with five possible sequences of MIMs were obtained for *R_Size*=100 after 20 simulations (i.e., in 20% of all simulations with *R_Size*=100).

A. *Application 2: problem of monitoring by mobile robot network*

A complex system application is required to prove the performance of the proposed metaheuristic. In this section, we propose to resolve a surface monitoring problem based on mobile robot network.

Let a surface divided on finite number *n* of sub-surfaces. For each sub-surface, a physical parameter Φ_j should be monitored; $\Phi\colon\{\Phi_1, \Phi_2, \dots, \Phi_m\}$ is a set of finite alphabet, when Φ_j is the alphabet affected to j^{th} parameter of the set Φ. Each mobile robot have the ability to move in its environment with not attachment to a physical location.

The goal is to know the minimum number of mobile robots as well as their ideal location at t=0; to ensure the monitoring of the total surface with respect to a label sequence $\mathcal{L}$.

As application, we consider a square divided on 9 sub-surfaces (figure 8). Each sub-square is characterized by one parameter belonging to Φ, where Φ: $\{a, b, c, d, e, f\}$.

The aim of this application example is to monitoring the square surface, via mobile robot network, according to a global constraint presented by the label sequence $\mathcal{L}= fbebaedfcbdbcbabcbcd$, of size 20 (SS $\simeq$ $1{,}347\ 10^{13}$ possible solutions).

The following assumption is considered for this example:

A robot can supervise all parameters, and must always be in monitoring state. I.e., a robot can only move from a sub-surface to another having direct borders with it. For example: a robot in S_1 can move only to S_2 and S_4; a robot in S_5 can move only to S_2, S_4 , S_6 and S_8.

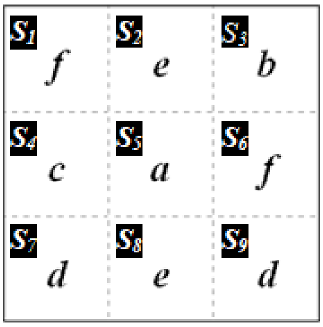


Figure 8: Surface to be monitored

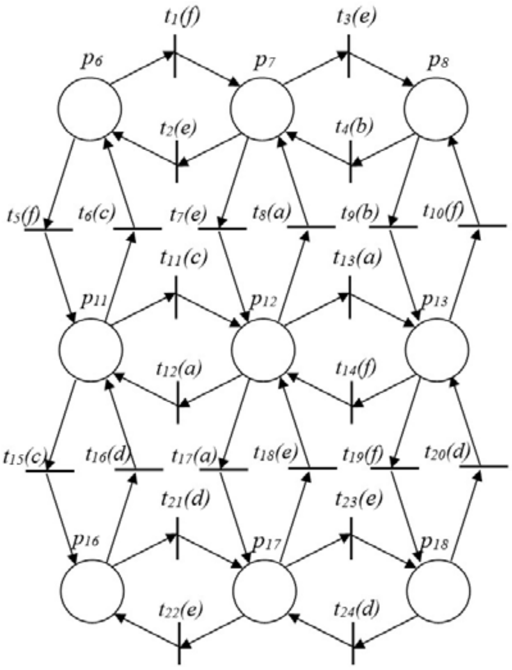


Figure 9: Surface S modeled with L-PN

The Labeled Petri Net of figure 9 represents the modeling of our structure with respect to all constraints already mentioned.

The objective is to minimize the number of robots (tokens) in the initial state as well as their positions (places). We execute the GMIM metaheuristic with the same simulation strategy described in the first application. Indeed, we begin the first simulation with 100 initial marking with respect to $\mathcal{L}$, (i.e., Λ_l=100). We continue the increase (with a $\Delta\Lambda_i$=100) until the number of MIM will be stable. For each Λ_i we accumulate 20 successive executions with new randomly input sequences.

The simulation results are presented in the table2 and figure 10.

Table 2: Comparative table

	R_size	*N_ DMIM*	*N_MIME*	*N_MIM*	*Time*
Algorithm of Li & Hadjicostis	-	-	-	-	-
Heuristic A-B of Li & Hadjicostis	-	8	28622	875	382(min)
GMIM Algorithm	100	3.4	100	4.8	0.43(s)
	200	8.7	200	34	1.7(s)
	300	10.7	300	68.4	2.8(s)
	400	12.6	400	145.8	4.6(s)
	600	13.9	600	286	7.5(s)
	800	14	800	493.4	12.3(s)
	1000	14	1000	600.8	20.1(s)
	1500	14	1500	1107.3	41(s)
	2000	14	2000	1470.7	55.9(s)
	4000	14	4000	3200	2.34(min)

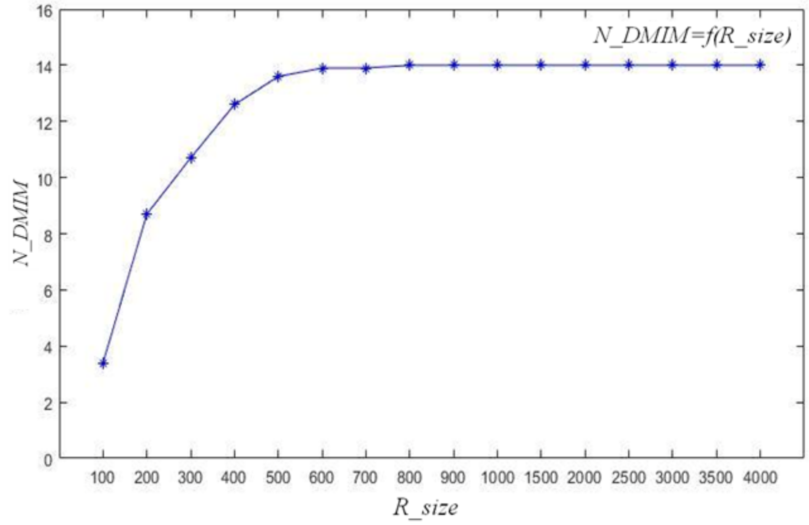
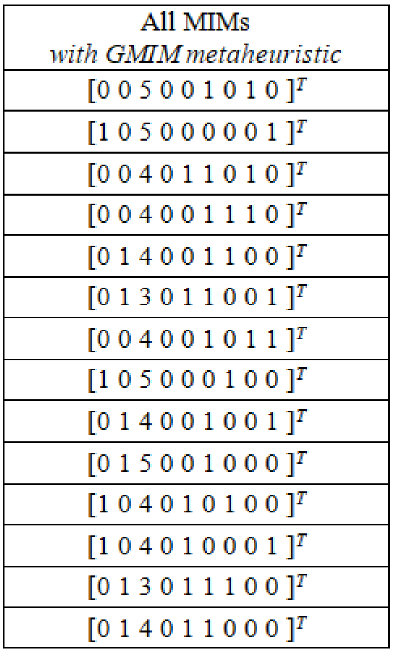


Figure 10: Figure 6: GMIM simulation (application 1) – *R_size* vs. *N_ DMIM*

We note that we can’t reach a final result by running the *Algorithm of Li & Hadjicostis* with a simple computer. On the other hand, the running time to simulate *Heuristic A-B* *of Li & Hadjicostis* takes more than 6 hours to provide frustrating results (only 8 estimated MIMs).

Figure 11 presents the set of all possible MIMs obtained science the simulation with Λ_f=800. In fact, the optimal number of mobile robots needed to monitoring this surface is 7 (=Σ tokens in one MIM) with 14 possibilities of emplacement in the initial sate.



All MIMs *with GMIM metaheuristic*
$[0\ 0\ 5\ 0\ 0\ 1\ 0\ 1\ 0]^T$
$[1\ 0\ 5\ 0\ 0\ 0\ 0\ 0\ 1]^T$
$[0\ 0\ 4\ 0\ 1\ 1\ 0\ 1\ 0]^T$
$[0\ 0\ 4\ 0\ 0\ 1\ 1\ 1\ 0]^T$
$[0\ 1\ 4\ 0\ 0\ 1\ 1\ 0\ 0]^T$
$[0\ 1\ 3\ 0\ 1\ 1\ 0\ 0\ 1]^T$
$[0\ 0\ 4\ 0\ 0\ 1\ 0\ 1\ 1]^T$
$[1\ 0\ 5\ 0\ 0\ 0\ 1\ 0\ 0]^T$
$[0\ 1\ 4\ 0\ 0\ 1\ 0\ 0\ 1]^T$
$[0\ 1\ 5\ 0\ 0\ 1\ 0\ 0\ 0]^T$
$[1\ 0\ 4\ 0\ 1\ 0\ 1\ 0\ 0]^T$
$[1\ 0\ 4\ 0\ 1\ 0\ 0\ 0\ 1]^T$
$[0\ 1\ 3\ 0\ 1\ 1\ 1\ 0\ 0]^T$
$[0\ 1\ 4\ 0\ 1\ 1\ 0\ 0\ 0]^T$

Figure 11: All possible MIMs obtained with application of GMIM

In fact, it is clear to interpret, from the table 2, the major advantage of our metaheuristic in terms of running time. By applying GMIM with initial population of 200 TFS the running time represents only 0,0074% compared to the running time of the *Heuristic A-B of Li & Hadjicostis.*

7 Conclusion

In this paper, a GMIM metaheuristic was developed for MIM estimation in L-PN. The proposed approach is inspired on the principle of the GRASP multi-start metaheuristic.

Parting from an initial phase based on the creation of a randomly feasible TFS respecting a given label sequence, an algorithm with two principal functions (mutation and fitness) is implemented until a local minimum solutions will be found during a local search phase. The complexity of GMIM depends on the L-PN model structure.

The proposed GMIM metaheuristic was applied on two problems: Initialization of two parallel machines and resources allocation in a monitoring mobile robots platform. With comparison to the state-of-the-art approaches, the simulation results show major improvement of our proposed GMIM metaheuristic mainly in faster running time.

References

[1] T. Watanabe, Y. Mizobata, and K. Onaga, “Legal firing sequences and minimum initial markings for petri nets,” in IEEE International Symposium on Circuits and Systems,. IEEE, 1989, pp. 323–326.

[2] A. Giua, C. Seatzu, and D. Corona, “Marking estimation of Petri nets with silent transitions,” IEEE Transactions on Automatic Control, vol. 52, no. 9, pp. 1695–1699, 2007.

[3] M. P. Cabasino, A. Giua, and C. Seatzu, “Marking estimation of petri nets with arbitrary transition labeling,” IFAC Proceedings Volumes, vol. 40, no. 6, pp. 229–234, 2007.

[4] M. P. Cabasino, A. Giua, A. Paoli, and C. Seatzu, “Decentralized diagnosis of discrete event systems using labeled Petri nets,” IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems, vol. 43, no. 6, pp. 1477 – 1485, 2013.

[5] F. Basile, M. P. Cabasino, and C. Seatzu, “Marking estimation of time Petri nets with unobservable transitions,” in 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA),, 2013, pp. 1–7.

[6] A. Giua, “Petri net state estimators based on event observation,” Proceedings of the 36th IEEE International Conference on Decision and Control, pp. 4086 – 4091, December 1997.

[7] A. Giua and C. Seatzu, “Observability of place/transition nets,” IEEE Transactions on Automatic Control, vol. 47, no. 9, pp. 1424–1437, September 2002.

[8] A. Giua, J. Júlvez, and C. Seatzu, “Marking estimation of Ppetri nets based on partial observation,” in Proceedings of the American Control Conference, vol. 1, 2003, pp. 326–331.

[9] A. Abdellatif, , P. Bonhomme, , A. J. Telmoudi, and L. Nabli. “Towards a Minimum Initial Marking Estimation Procedure for P-Time Labelled Petri net Systems”. In : 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2019. p. 42-47.

[10] L. Li and C. N. Hadjicostis, “Least-cost planning sequence estimation in labelled Petri nets,” Transactions of the Institute of Measurement and Control, vol. 33, no. 3-4, pp. 317–331, 2011.

[11] 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, 2012.

[12] M. Yamauchi and T. Watanabe, “A heuristic algorithm for the minimum initial marking problem of petri nets,” in 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 1, Oct 1997, pp. 245–250.

[13] H. Kmimech, A. J. Telmoudi, L. Sliman, and L. Nabli. "Genetic-based Approach for Minimum Initial Marking Estimation in Labeled Petri Nets". IEEE Access, 2020.

[14] T. A. Feo and M. G. Resende, “Greedy randomized adaptive search procedures,” Journal of global optimization, vol. 6, no. 2, pp. 109–133, 1995.

[15] T. Murata, “Petri nets, properties, analysis and applications,” Proceedings of the IEEE, vol. 77, pp. 541–580, 1989.

[16] C. G. Cassandras and S. Lafortune, “Discrete event systems: The state of the art and new directions,” in Applied and computational control, signals, and circuits. Springer, 1999, pp. 1–65.

[17] J.-M. Proth and X. Xie, “Petri Nets: A Tool for Design and Management of Manufacturing Systems,” New York: Wiley, 1996.

[18] Y. Ru and C. N. Hadjicostis, “State estimation in discrete event systems modeled by labeled petri nets,” in Proceedings of the 45th IEEE Conference on Decision and Control. IEEE, 2006, pp. 6022–6027.