i_{Rank}: A Variable Order Metric for DEDS Subject to Linear Invariants

Elvio Gilberto Amparore[1], Gianfranco Ciardo[2], Susanna Donatelli[1(✉)], and Andrew Miner[2]

[1] Dipartimento di Informatica, Università di Torino, Torino, Italy
{amparore,donatelli}@di.unito.it
[2] Iowa State University, Ames, IA, USA
{ciardo,asminer}@iastate.edu

Abstract. Finding good variable orders for decision diagrams is essential for their effective use. We consider Multiway Decision Diagrams (MDDs) encoding a set of fixed-size vectors satisfying a set of linear invariants. Two critical applications of this problem are encoding the state space of a discrete-event discrete state system (DEDS) and encoding all solutions to a set of integer constraints. After studying the relations between the MDD structure and the constraints imposed by the linear invariants, we define i_{Rank}, a new variable order metric that exploits the knowledge embedded in these invariants. We evaluate i_{Rank} against other previously proposed metrics on a benchmark of 40 different DEDS and show that it is a better predictor of the MDD size and it is better at driving heuristics for the generation of good variable orders.

Keywords: Decision diagrams · Variable order metrics and computation

1 Introduction

Decision diagrams (DDs) are a popular data structure to encode large sets of structured data, for example vectors whose elements take values over finite domains, but it is well-known [10] that the size of the DD strongly depends on how the structure of the data (its "variables") is mapped to the structure of the DD (its "levels"). The problem of determining the association of variable(s) to levels is the "variable ordering problem" and it is known that finding an optimal order is an NP-complete problem [9] for any DD class, including binary DDs (BDDs [10]) and multiway DDs (MDDs [19]). This has given rise to a variety of metrics (to compare the effectiveness of two orders without actually building the corresponding DDs) and of heuristics (to compute sub-optimal orders, often by attempting to optimize a given metric). DDs play a central role in many system verification tools [4,11,14,20,22], where they typically support state space exploration. Tools often make use of general-purpose DD libraries [8,18,26,27]. Libraries typically support dynamic reordering to improve the current order at

T. Vojnar and L. Zhang (Eds.): TACAS 2019, Part II, LNCS 11428, pp. 285–302, 2019.
https://doi.org/10.1007/978-3-030-17465-1_16

run-time, while the definition of an initial order (static ordering) is typically up to the verification tool, which can rely on domain knowledge. The two problems are synergistic: reordering works better if the initial order is at least fairly good.

Our research seeks to find good variable order metrics and good variable order heuristics for MDDs encoding sets of fixed-size vectors, when these vectors satisfy some linear invariants. We want to answer whether it is possible to *leverage invariant information to define effective metrics and heuristics for variable order*. Two applications where this is important are *encoding the state space of a discrete-event discrete state system* and *encoding all solutions to a set of integer constraints*. In this paper, we concentrate on the first problem, but also address a special case of the second. Specifically, we study the relationship between MDDs and linear invariants with integer coefficients, and define two new metrics, PF and $\mathrm{i}_{\mathrm{Rank}}$, and associated heuristic and meta-heuristic. PF and $\mathrm{i}_{\mathrm{Rank}}$ exploit the constraint imposed by the invariants. Our evaluation shows that $\mathrm{i}_{\mathrm{Rank}}$ is superior to any other metric we consider, in all experiments we performed.

We do not discuss the state-of-the art on heuristics, see [7] for a full survey, but only metrics and on how metric optimization can guide a meta-heuristic. After the necessary background in Sect. 2, Sects. 3 and 4 define the metrics PF and $\mathrm{i}_{\mathrm{Rank}}$, based on a number of observation and propositions on the relation between MDD and invariants. Section 5 experimentally evaluates the two metrics against several other metrics on 40 different models, considering thousands of variable orders. Section 6 summarizes our results and discusses future work.

2 Background

Let $\mathbb{B} = \{\bot, \top\}$, $\mathbb{N}$ and $\mathbb{Z}$ denote the set of booleans, natural numbers, and integers, respectively. All other sets are denoted by calligraphic letters, e.g., $\mathcal{A}$.

2.1 Discrete-Event Discrete-State System and Their State Space

A *discrete-event discrete-state system* can be generally described by providing:

(1) The set of *potential* states $\mathcal{S}_{pot}$, defining the type of system states. We assume $\mathcal{S}_{pot} = \mathbb{N}^{\mathcal{V}}$; i.e., a state $\mathbf{m}$ is a valuation of a finite set $\mathcal{V}$ of natural variables.
(2) The state $\mathbf{m}_{init} \in \mathcal{S}_{pot}$, describing the *initial* state of the system.
(3) The relation $\mathcal{R} \subseteq \mathcal{S}_{pot} \times \mathcal{S}_{pot}$, describing the state-to-state transitions; if $(\mathbf{m}, \mathbf{m}') \in \mathcal{R}$, the system can move from $\mathbf{m}$ to $\mathbf{m}'$ in one step. We assume that $\mathcal{R}$ is defined by a finite set of *events* $\mathcal{E}$ and a function $\mathit{Effect} : \mathcal{E} \times \mathcal{S}_{pot} \to (\mathcal{S}_{pot} \cup \{\circ\})$, specifying the unique state $\mathbf{m}'$ reached if event e occurs in state $\mathbf{m}$, none if $\mathit{Effect}(e, \mathbf{m}) = \circ$ (e is *disabled* in $\mathbf{m}$). We write $\mathbf{m} \xrightarrow{e} \mathbf{m}'$ iff $\mathit{Effect}(e, \mathbf{m}) = \mathbf{m}' \neq \circ$.

The *reachable* states are $\mathcal{S}_{rch} = \{\mathbf{m} : \exists e_1, \ldots, e_n \in \mathcal{E}, \mathbf{m}_{init} \xrightarrow{e_1} \cdots \xrightarrow{e_n} \mathbf{m}\}$ and, for such a system, an *invariant* is a boolean function $f : \mathcal{S}_{pot} \to \mathbb{B}$ with the

property that it evaluates to $\top$ in all reachable states: $\mathbf{m} \in \mathcal{S}_{rch} \Rightarrow f(\mathbf{m})$, while it may be either $\top$ or $\bot$ in the unreachable states $\mathcal{S}_{pot} \setminus \mathcal{S}_{rch}$.

We specify DEDSs as Petri nets, because of their widespread use and the large body of literature on Petri net invariants. In Petri net terminology, the evaluation of variables describes the number of *tokens* in the set $\mathcal{P}$ of *places* (thus the state, or *marking*, is a vector in $\mathbb{N}^{\mathcal{P}}$), the events $\mathcal{E}$ correspond to the *transitions* $\mathcal{T}$, while two $\mathbb{N}^{\mathcal{P}\times\mathcal{T}}$ matrices $\mathbf{C}^-$ and $\mathbf{C}^+$ define the system evolution. $\mathit{Effect}(t, \mathbf{m}) = \mathbf{m}+\mathbf{C}^+[\mathcal{P}, t]-\mathbf{C}^-[\mathcal{P}, t]$ (transition firing) iff $\mathbf{m} \geq \mathbf{C}^-[\mathcal{P}, t]$, otherwise $\mathit{Effect}(t, \mathbf{m}) = \circ$, i.e., t is disabled in $\mathbf{m}$, where $\geq$ is interpreted componentwise. The *incidence* matrix $\mathbf{C} = \mathbf{C}^+ - \mathbf{C}^-$ is the net change to the marking caused by firing transition t is $\mathbf{C}[\mathcal{P}, t]$. Figure 1 shows two Petri nets used as running example. Places are shown as circles, transitions as bars, and $\mathbf{C}^-$ ($\mathbf{C}^+$) as incoming (outgoing) arcs for transitions with the corresponding value in $\mathbf{C}^-$ ($\mathbf{C}^+$) shown on the arc (omitted if 1). The incidence matrix and initial marking are next to the nets.

A *p-flow* is a vector $\boldsymbol{\pi} \in \mathbb{Z}^{\mathcal{P}} \setminus \{\mathbf{0}\}$ such that $\boldsymbol{\pi}^T \cdot \mathbf{C} = \mathbf{0}$, and its *support* is $\mathit{Supp}(\boldsymbol{\pi}) = \{v \in \mathcal{P} : \boldsymbol{\pi}[v] \neq 0\}$. A p-flow $\boldsymbol{\pi}$ implies a *linear invariant* of the form $\forall \mathbf{m} \in \mathcal{S}_{rch} : \boldsymbol{\pi}^T \cdot \mathbf{m} = \boldsymbol{\pi}^T \cdot \mathbf{m}_{init}$, where $\boldsymbol{\pi}^T \cdot \mathbf{m}_{init} = Tc(\boldsymbol{\pi})$ is obviously a constant value, the *token count* of the invariant, which depends only on $\mathbf{m}_{init}$. If clear from the context, $\boldsymbol{\pi}$ may refer to either a p-flow or the implied invariant.

P-flows with no negative entries are called *p-semiflows*. Let $\mathcal{F}$ be the set of p-flows, $\mathcal{F}^+$ the set of p-semiflows, and $\mathcal{F}^- = \mathcal{F} \setminus \mathcal{F}^+$ the p-flows that are not p-semiflows. Since multiplying a p-flow by a non-zero integer results in a p-flow, these sets are either empty or infinite. Figure 1 shows the *minimal* p-flows (defined later) as column vectors, with the token count below the vector.

A p-semiflow $\boldsymbol{\pi}$ describes a *conservative invariant*, which implies a *bound* $\mathbf{m}[v] \leq Tc(\boldsymbol{\pi})/\boldsymbol{\pi}[v]$ on the number of tokens in each place v of the support of $\boldsymbol{\pi}$ for any reachable marking $\mathbf{m}$. Column "bnd" in Fig. 1 reports these bounds. The two p-semiflows in Fig. 1(A) express the following invariants:

$$f_1: \quad \forall \mathbf{m} \in \mathcal{S}_{rch},\ \mathbf{m}[P_0] + \mathbf{m}[P_1 b] + \mathbf{m}[P_2 b] + \mathbf{m}[P_3 b] = 2$$
$$f_2: \quad \forall \mathbf{m} \in \mathcal{S}_{rch},\ \mathbf{m}[P_0] + \mathbf{m}[P_1 a] + \mathbf{m}[P_2 a] + \mathbf{m}[P_3 a] = 2.$$

These in turn imply that the number of tokens in each place is bounded by 2. We assume that each place v is bounded by some $n_v \in \mathbb{N}$, and redefine $\mathcal{S}_{pot}$ as $\times_{v\in\mathcal{P}}[0, 1, \ldots, n_v]$. This ensures that $\mathcal{S}_{rch}$ is finite and therefore can be encoded in a (large enough but finite) MDD. This is the case if the Petri net is covered by conservative invariants, i.e., each place is in the support of some p-semiflow.

Work on Petri net invariants has mainly targeted $\mathcal{F}^+$ rather than $\mathcal{F}$, possibly because it is easier to compute properties, like the bounds of places, with $\mathcal{F}^+$. On the other hand, $\mathcal{F}$ can be characterized by a basis (whose size equals to the dimension of the null space of $\mathbf{C}$, thus cannot exceed the smaller of $|\mathcal{P}|$ and $|\mathcal{T}|$), while $\mathcal{F}^+$ can only be characterized by a *minimal generator*, the smallest set of vectors that can generate its elements through non-negative integer linear combinations of its elements. It has been shown [15] that this set is finite, is unique (thus we can denote it as $\mathcal{F}^+_{min}$), and consists of all *minimal* p-semiflows

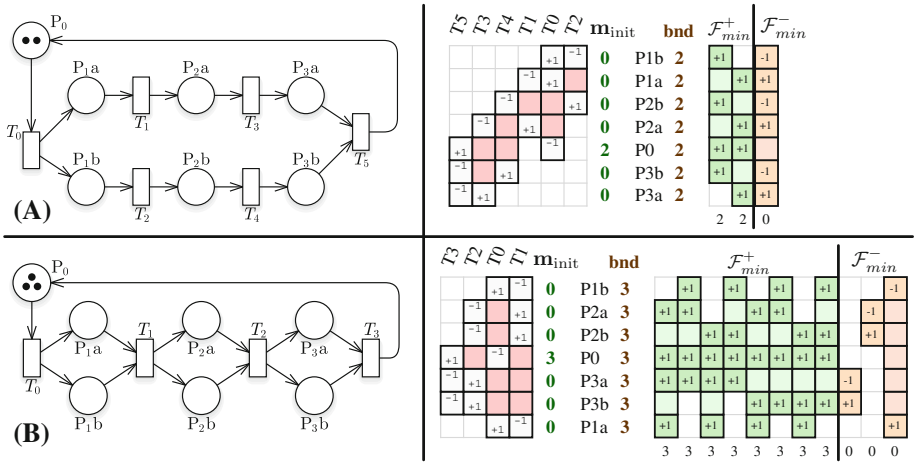

Fig. 1. Two Petri nets, their incidence matrices, and their p-flows.

(where a p-semiflow is minimal if the g.c.d. of its coefficients is 1 and its support does not strictly contain the support of another p-semiflow). However, $\mathcal{F}^+_{min}$ may have size exponential in $|\mathcal{P}|$. A classic example of this is a Petri net sequence of fork and join models with $n+1$ transitions and $2n+1$ places whose $\mathcal{F}^+_{min}$ has size 2^n. Figure 1(B) shows the case $n=3$. The reader can find in [15] full details and a thorough analysis of the cost of computing $\mathcal{F}^+_{min}$.

In addition to $\mathcal{S}_{rch}$, we can define $\mathcal{S}_{sat} = \{\mathbf{m} \in \mathbb{N}^{\mathcal{P}} : \forall \boldsymbol{\pi} \in \mathcal{F}, \mathbf{m} \cdot \boldsymbol{\pi} = Tc(\boldsymbol{\pi})\}$. Obviously $\mathcal{S}_{rch} \subseteq \mathcal{S}_{sat}$. We let $\mathcal{S}$ refer to either when the distinction is not relevant. Note that $\mathcal{S}_{sat}$ is a superset of the *linearized reachability set* [21] $\{\mathbf{m} \in \mathbb{N}^{\mathcal{P}} : \exists\, \mathbf{y} \in \mathbb{N}^{\mathcal{T}}, \mathbf{m} = \mathbf{m}_{init} + \mathbf{C} \cdot \mathbf{y}^T\}$, used in Petri net theory to devise a semi-decidable procedure for safety properties.

2.2 Multiway Decision Diagrams

Definition 1 (MDD). Given a *global domain* $\mathcal{X} = \times_{k=1}^{L} \mathcal{X}_k$, where each *local domain* $\mathcal{X}_k$ is of the form $\{0, 1, \ldots, n_k\}$ for some $n_k \in \mathbb{N}$, an (ordered, quasi-reduced) MDD over $\mathcal{X}$ is a directed acyclic graph with exactly two terminal nodes, $\top$ and $\bot$, at *level* 0 (we write $\top.lvl = \bot.lvl = 0$), with each non-terminal node p at some level $p.lvl = k \in \{1, \ldots, L\}$ having one outgoing edge for each $i \in \mathcal{X}_k$, pointing to a node $p[i]$ at level $k-1$ or to $\bot$, and with no *duplicates* (there cannot be nodes p and q at level k with $p[i] = q[i]$ for all $i \in \mathcal{X}_k$) or *redundant* nodes (node p at level k is redundant if $p[0] = p[i]$ for all $i \in \mathcal{X}_k$) pointing to $\bot$. The function $f_p : \mathcal{X} \to \mathbb{B}$ encoded by an MDD node p is recursively defined as $f_p(i_1, \ldots, i_L) = f_{p[i_k]}(i_1, \ldots, i_L)$ if $p.lvl = k > 0$, and $f_p(i_1, \ldots, i_L) = p$ if $p.lvl = 0$. Interpreting f_p as an indicator function, p also encodes the set $\mathcal{S}_p \subseteq \mathcal{X}$, defined as $\mathcal{S}_p = \{(i_1, \ldots, i_L) : f_p(i_1, \ldots, i_L)\}$. This is the set of variable assignments compatible with the paths from p to $\top$. □

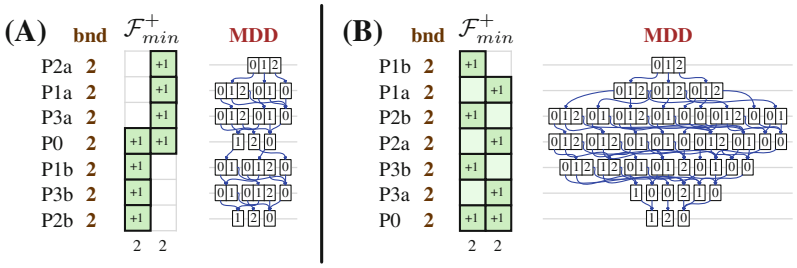

Fig. 2. P-semiflows and MDD for two variable orders for the net in Fig. 1(A).

MDDs are a *canonical* representation of subsets of $\mathcal{X}$: given MDD nodes p and q at the same level, $\mathcal{S}_p = \mathcal{S}_q$ iff $p = q$. We observe that quasi-reduced MDDs differ from the more common *fully-reduced* MDDs, which allow edges to skip levels by eliminating all redundant nodes, not just those encoding $\bot$. As it will be clear, though, the quasi-reduced MDD encoding the state space of a Petri net covered by invariants *cannot contain redundant nodes*, thus coincides with the fully-reduced MDD for such models. When drawing MDDs, edges point down and we omit node $\bot$, edges pointing to it, and the corresponding cells in the originating node, so that, if node p at level k with $\mathcal{X}_k = \{0, \ldots, 4\}$ is drawn as $\boxed{2}\boxed{3}$, it means that $p[0] = p[1] = p[4] = \bot$. We also omit node $\top$ and edges pointing to it, but not the corresponding cell in the originating node.

MDDs have been successfully employed to generate and store the reachable state space of DEDSs, in particular Petri nets, using fixpoint *symbolic* iterations. The MDD representation of a state space $\mathcal{S}_{rch}$ is computed as the least fixpoint of the equation $\mathcal{Z} = \mathcal{Z} \cup \{\mathbf{m}_{init}\} \cup \{\mathbf{m}' : \mathbf{m} \in \mathcal{Z} \wedge \exists e \in \mathcal{E}, \mathbf{m} \xrightarrow{e} \mathbf{m}'\}$, while the generation of $\mathcal{S}_{sat}$ simply needs to consider one flow (and associated invariant) at a time, thus can be achieved by performing exactly $|\mathcal{F}| - 1$ intersections of the sets of assignments satisfying each individual constraint.

Since we focus on the size of the MDD encoding $\mathcal{S}$, we only consider MDDs with a single root node r, so that $\mathcal{S}_r = \mathcal{S}$. Letting $\mathcal{N}_k$ be the set of MDD nodes al level k, we characterize the MDD size in terms of its nonterminal nodes $\mathcal{N}$, i.e., $|\mathcal{N}| = \sum_{k=1}^{L} |\mathcal{N}_k|$ (although, unlike for BDDs [10] where nodes have exactly two outgoing edges, the number of MDD edges $\sum_{k=1}^{L} |\{(p, i) : p \in \mathcal{N}_k, p[i] \neq \bot\}|$ could also be a meaningful measure of size). The first step to generate $\mathcal{S}$ is to map the places $\mathcal{P}$ of the Petri net to the L levels of the MDD. We limit ourselves to mapping each place to a different level, i.e., requiring a *variable order* $\lambda : \mathcal{P} \to \{1, \ldots, L\}$, where $L = |\mathcal{P}|$. It is known that the choice of λ can exponentially affect the size of MDD and finding an optimal mapping is NP-complete [9]. We stress that we consider only the *final* size of the MDD. In reality, the fixpoint iterations to compute $\mathcal{S}_{rch}$ or the intersections to compute $\mathcal{S}_{sat}$ can lead to an intermediate size of the MDD (*peak* size) that is normally much larger than the final size. However, our work to reduce the final MDD size is largely orthogonal to other strategies (like *saturation* [13] for $\mathcal{S}_{rch}$ construction) aimed at reducing the peak size, thus both can be employed to improve efficiency.

The MDDs in Fig. 2 encode $\mathcal{S}_{rch}$ for the Petri net of Fig. 1(A), for two different variable orders. More precisely Fig. 2 shows, left to right, and for each order, the variable order (with level L at the top), the place bounds, the p-semiflows $\mathcal{F}^{+}_{min}$ (with the token count at the bottom), and the corresponding MDDs. The variable order in (B) is poor, resulting in an MDD with 40 nodes, while that in (A) requires only 19 nodes.

2.3 Metrics for Variable Orders

A metric M is a *perfect* predictor of MDD size if $M(\lambda_1) \leq M(\lambda_2)$ implies $|\mathcal{N}(\lambda_1)| \leq |\mathcal{N}(\lambda_2)|$ for any variable orders λ_1 and λ_2, where $\mathcal{N}(\lambda)$ is the number of nodes in the MDD for $\mathcal{S}$ when using variable order λ; no efficiently-computable perfect predictor is known. Metrics have been defined based on the *span of events* in the incidence matrix $\mathbf{C}$, on the *bandwidth* of $\mathbf{C}$, on the *center of gravity* of events, and on p-semiflows. Metrics that consider the span of each event t (distance between the top and bottom nonzero in $\mathbf{C}^{-}[\mathcal{P}, t]$ or $\mathbf{C}^{+}[\mathcal{P}, t]$ for the given variable order) are the Normalized sum of Event Span (NES), the Weighted NES (WES), Sum of Span (SOS) [24], Sum of Tops (SOT) [11] and Sum of Unique and Productive Spans (SOUPS) [25]. Classic bandwidth reduction techniques from linear algebra were applied to variable order computation for the first time in [23]. The corresponding metrics are Bandwidth (BW), Profile (PROF), or Wavefront (WF), computed on a squared matrix derived from the incidence matrix $\mathbf{C}$. Point-transition spans (PTS) is the metric used as a convergence criterion by the widely used heuristic FORCE [3], an algorithm for multi-dimensional clustering of graphs that has been adapted to variable order generation. A center of gravity for the variables is defined and the orders are measured in terms of *hyperdistance* of the variable from the center of gravity. PTS^{P} [6] is a variation of PTS to consider also the effect of p-semiflows in the PTS variable clustering. Finally, the p-semiflow span (PSF) is the metric optimized by the heuristic defined in [5], which works by ordering the variables according to p-semiflows. PSF is a measure of the proximity of places that belong to the same p-semiflow.

An overview of these metrics can be found in [6], which also studies their coefficient of correlation to determine the predictive power of each metric over a large set of models and of orders. All models in the study are Petri nets, mostly conservative. We now provide some details for SOUPS and PSF which, together with PTS^{P}, have been reported as valuable predictors [6].

SOUPS modifies the *sum of transition spans* (SOS) metric [24] by considering only once the maximal common portion of multiple transition spans having the same effect on the marking and avoids counting the bottom portion of a transition span if it checks but does not change the marking of the corresponding places. SOUPS performs particularly well in conjunction with saturation [13], as it tends to result in even smaller peak MDDs. SOS and SOUPS, just like WES and NES [24] or SOT [11], are easily computed from the matrices $\mathbf{C}^{-}$ and $\mathbf{C}^{+}$.

PSF is computed analogously to SOS, but considering p-semiflow spans instead of transition spans:

$$\mathrm{PSF}(\lambda) = \sum_{\boldsymbol{\pi} \in \mathcal{F}^+_{min}} \Big(\max\{\lambda(v) : \boldsymbol{\pi}[v] \neq 0\} - \min\{\lambda(v) : \boldsymbol{\pi}[v] \neq 0\} + 1 \Big).$$

In our figures, the column for p-flow $\boldsymbol{\pi}$ has a dark cell with $\boldsymbol{\pi}[v]$ in it for each place v in $Supp(\boldsymbol{\pi})$, a light empty cell for each place not in the support but bracketed by places in the support, and a white empty cell for the remaining places not in the support. With this notation, PSF is just the count of the number of non-white squares in the matrix of $\mathcal{F}^+_{min}$.

There has been a proposal [16] to use of p-semiflows to *eliminate* some state variables (decision diagram levels) through a greedy heuristic, but later work [12] observed that this leads to a loss of *locality* in the MDD representation of the transition relation, and suggested instead to use p-semiflows to *merge* variables, proving that this always reduces the MDD size. The same paper [12] also proposed to modify the sum-of-transition-tops (SOT) metric so that it considers also a set of linearly independent p-semiflows, but provided no hints about the relative weight given to transitions vs. p-semiflows when computing the metric.

3 MDD and Invariants: The PF Metric

We now begin investigating the relationship between p-flows and the shape of the MDD encoding $\mathcal{S}_{rch}$ and $\mathcal{S}_{sat}$, and introduce the new metric PF.

P-flows and information remembered at level k. The invariant corresponding to a p-flow $\boldsymbol{\pi}$ imposes a constraint on the reachable markings, since it implies a constant weighted sum of the tokens in the places belonging to $Supp(\boldsymbol{\pi})$. Thus, the MDD must "remember" (using distinct nodes at level k) the possible partial weighted sums corresponding to places in the invariant support that are above level k, as long as the invariant is *active*, i.e., its support contains places mapped to levels k or below, and this is true even if the place mapped to level k is not in the support. Thus, intuitively, places in the support should be mapped to levels close to each other. This can be easily seen seen in Fig. 2(B), where the places in the support of the two p-semiflows in $\mathcal{F}^+_{min}$ are not in consecutive levels, resulting in more nodes: the level for $P_2 b$ has 9 nodes, since the MDD must remember the partial sum of tokens of the places in the two branches of the Petri net of Fig. 1(A), and each of them can range from 0 to 2. In the order of Fig. 2(A), all places in the top branch are instead above the level of place P_0, which is in turn above all places in the bottom branch. Thus, level P_0 has only three possible values to remember: whether in the top (and thus in the bottom) branch there are 0, 1, or 2 tokens (and therefore P_0 has 2, 1, or 0 tokens, respectively). This dependence is captured by the metric PSF of Sect. 2.3. The PSF value for order (B) is 13, while it is 8 for order (A), consistent with the intuition that a smaller value of PSF results in a smaller MDD.

P-flows and singletons. The token count of an invariant $\boldsymbol{\pi}$ determines a *single* possible value for the number of tokens in the level "completing" $\boldsymbol{\pi}$ (the lowest level corresponding to a place in $Supp(\boldsymbol{\pi})$), which can then only contain *singletons* (nodes with a single outgoing edge). This is the case for level P_0 in the

MDD of Fig. 2(A) and P_0 in the MDD of Fig. 2(B). Interestingly, level P_3a in the MDD of Fig. 2(B) also contains only singletons. This is due to an invariant generated by a p-flow in $\mathcal{F}^-$:

$$\boldsymbol{\pi}_3 : \quad \mathbf{m}[P_1a] + \mathbf{m}[P_2a] + \mathbf{m}[P_3a] - \mathbf{m}[P_1b] - \mathbf{m}[P_2b] - \mathbf{m}[P_3b] = 0,$$

As p-flows in $\mathcal{F}^-$ have similar implications on the MDD structure as those in $\mathcal{F}^+$, we define a new metric PF, by extending PSF to consider also non-positive p-flows. Give a set of p-flows $\mathcal{F}_{min}$, we can then define:

$$\mathrm{PF}(\lambda) = \textstyle\sum_{\boldsymbol{\pi} \in \mathcal{F}_{min}} \Big(\max\big\{\lambda(p) : \boldsymbol{\pi}(p) \neq 0\big\} - \min\big\{\lambda(p) : \boldsymbol{\pi}(p) \neq 0\big\} + 1 \Big),$$

But what is an appropriate choice for $\mathcal{F}_{min}$? To have a consistent definition of the metric we need $\mathcal{F}_{min}$ to be uniquely and appropriately defined. While p-semiflows are characterized by a unique generator set $\mathcal{F}^+_{min}$, p-flows can be characterized by a *basis*, but the choice of basis is not unique and can lead to meaningless value of PF (for example if we choose a basis where each p-flow has the same span over the places, so that any variable order results in the same value for the PF metric).

Continuing the analogy with PSF, we define $\mathcal{F}_{min}$ as the set of minimal p-flows, i.e., the g.c.d. of their entries is 1 and their support does not strictly include the support of any other p-flow; in addition, to avoid considering both a p-flow and its negative, we assume an arbitrary place order (unrelated to the MDD variable order) and require the first nonzero entry to be positive. We now prove that this set $\mathcal{F}_{min}$ is unique and that it can generate a multiple of any p-flow. In the figures, the set $\mathcal{F}_{min}$ is shown partitioned into $\mathcal{F}^+_{min}$ and $\mathcal{F}^-_{min} = \mathcal{F}_{min} \backslash \mathcal{F}^+_{min}$.

Theorem 1. Set $\mathcal{F}_{min}$ is unique, and it spans all p-flow directions, i.e., given $\boldsymbol{\pi} \in \mathcal{F}$, for some $a \in \mathbb{Z}$, $a\boldsymbol{\pi}$ equals a linear combination of elements in $\mathcal{F}_{min}$.

Proof. To prove uniqueness, it suffices to show that there can be at most one minimal p-flow with a given support. Assume by contradiction that there are two distinct minimal p-flows $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ with $Supp(\boldsymbol{\pi}_1) = Supp(\boldsymbol{\pi}_2) = \mathcal{Q}$, and let $a_1 > 0$ and $a_2 > 0$ be the coefficients in $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ corresponding to the first place $v \in \mathcal{Q}$, respectively. Then, define $\boldsymbol{\pi} = a_2\boldsymbol{\pi}_1 - a_1\boldsymbol{\pi}_2$, so that $\boldsymbol{\pi}[v] = 0$.
If $\boldsymbol{\pi} \neq \mathbf{0}$, then $\boldsymbol{\pi} \in \mathcal{F}$ but $Supp(\boldsymbol{\pi}) \subseteq \mathcal{Q} \backslash \{v\}$, thus $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ cannot be minimal p-flows since their support strictly contains the support of $\boldsymbol{\pi}$, a contradiction.
If $\boldsymbol{\pi} = \mathbf{0}$, then $a_2\boldsymbol{\pi}_1 = a_1\boldsymbol{\pi}_2$, which implies $a_2 = a_1$, since the g.c.d. of both $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ is 1. But then, $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_2$, again a contradiction.
To prove that $\mathcal{F}_{min}$ spans all p-flow directions, consider $\boldsymbol{\pi}'_1 \in \mathcal{F}$. There must exist $\boldsymbol{\pi}_1 \in \mathcal{F}_{min}$ with $Supp(\boldsymbol{\pi}_1) \subseteq Supp(\boldsymbol{\pi}'_1)$; pick $v \in Supp(\boldsymbol{\pi}_1)$ and let $a_1 = \boldsymbol{\pi}_1[v]$ and $b_1 = \boldsymbol{\pi}'_1[v]$, so that $a_1\boldsymbol{\pi}'_1 = b_1\boldsymbol{\pi}_1 + \boldsymbol{\pi}'_2$, with $Supp(\boldsymbol{\pi}'_2) \subseteq Supp(\boldsymbol{\pi}'_1) \setminus \{v\}$. Either $Supp(\boldsymbol{\pi}'_2) = \emptyset$, or it is a p-flow, in which case we can repeat the process to obtain $a_2\boldsymbol{\pi}'_2 = b_2\boldsymbol{\pi}_2 + \boldsymbol{\pi}'_3$, and so on. Eventually, we must reach the case $Supp(\boldsymbol{\pi}'_{n+1}) = \emptyset$, i.e., $\boldsymbol{\pi}'_{n+1} = \mathbf{0}$, at which point we can write $a_1 \cdots a_n\boldsymbol{\pi}'_1 = b_1a_2 \cdots a_n\boldsymbol{\pi}_1 + b_2a_3 \cdots a_n\boldsymbol{\pi}_2 + \cdots + b_n\boldsymbol{\pi}_n$, where $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n \in \mathcal{F}_{min}$, i.e., we can express a multiple of $\boldsymbol{\pi}'_1$ as a linear combination of elements of $\mathcal{F}_{min}$. □

We observe that the size of $\mathcal{F}_{min}$, like that of $\mathcal{F}^{+}_{min}$, is at most exponential in $|\mathcal{P}|$, since the proof of Theorem 1 shows that the elements of $\mathcal{F}_{min}$ must have uncomparable supports.

4 MDD and Invariants: The i_{Rank} Metric

As we shall see in Sect. 5, both PSF and PF exhibit significant correlation with the MDD size. However, there are cases where they do not perform well, especially when $\mathcal{F}_{min}$ is large. Consider for example the Petri net of Fig. 1(B), and the three MDDs corresponding to different variable orders in Fig. 3. This Petri net has many minimal p-flows, $|\mathcal{F}^{+}_{min}| = 8$ and $|\mathcal{F}_{min}| = 11$. The three p-flows in $\mathcal{F}^{-}_{min}$ relate the places inside each fork-and-join subnet ($P_i a = P_i b$, for $i = 1, 2, 3$), while the eight p-semiflows in $\mathcal{F}^{+}_{min}$ relate the tokens in the three fork-and-join subnets with those in place P_0. The order in Fig. 3(B) produces the smallest MDD size (37 nodes against the 49 nodes of order (A) and 69 of (C)), but it is the one with the worst (highest) value of PSF. On the other side also PF fails to chose the order with the smallest MDD: the smallest value for PF is 55 for order (A), which is only the second best for MDD size. One reason is that, when $\mathcal{F}_{min}$ contains many related, dependent constraints affecting a given MDD level, counting all of them may confuse the metric. On the other hand, we have seen that considering instead a basis depends strongly on the choice of vectors included in the basis, with a meaningless metric in the worst case.

We then propose i_{Rank}, a new variable order metric which, like PSF and PF, is based on linear invariants but, unlike PSF and PF, *is unaffected by redundant minimal p-flows* and *is independent of the choice of the specific p-flows* being considered, as long as they constitute a generator set. i_{Rank} focuses on the number $\rho(k)$ of *linearly independent partial p-flows* that are still active at level k. The definition of i_{Rank} requires a deeper understanding of the relationships among the MDD structure and the p-flows, as illustrated next.

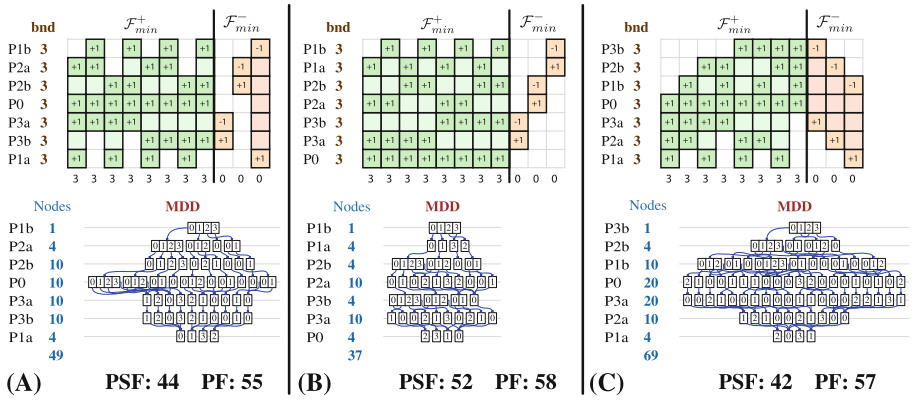

Fig. 3. Three variable orders for the Petri net of Fig. 1(B), and the resulting MDDs.

Given an MDD with root node r and two MDD nodes $p, q \neq \bot$ with $p.lvl = k$ and $q.lvl = h$, let $\mathcal{A}_p$ (for above) be the set of paths from r to p, $\mathcal{B}_p$ (for below) the set of paths from p to $\top$, and $\mathcal{C}_{p,q}$ the set of paths from p to q:

$$\mathcal{A}_p = \{(i_L, \ldots, i_{k+1}) : r[i_L] \cdots [i_{k+1}] = p\}$$
$$\mathcal{B}_p = \{(i_k, \ldots, i_1) : p[i_k] \cdots [i_1] = \top\}$$
$$\mathcal{C}_{p,q} = \{(i_k, \ldots, i_{h+1}) : p[i_k] \cdots [i_{h+1}] = q\},$$

thus $\mathcal{A}_r = \mathcal{B}_\top = \{()\}$, $\mathcal{A}_\top = \mathcal{B}_r = \mathcal{S}_r$, $\mathcal{A}_p = \mathcal{C}_{r,p}$, $\mathcal{B}_p = \mathcal{C}_{p,\top}$, $\mathcal{C}_{p,q} = \emptyset$ if $q.lvl \geq p.lvl$. When using an MDD to store $\mathcal{S}$ with a given variable order λ, the sets of paths defined by $\mathcal{A}_p$, $\mathcal{B}_p$, and $\mathcal{C}_{p,q}$ also denote sets of submarkings, by interpreting i_k as the number of tokens in place $v = \lambda^{-1}(k)$, and so on.

Theorem 2 [28]**.** The nodes at level k can be used to define a partition of $\mathcal{S}_r$: $\bigcup_{p \in \mathcal{N}_k} \mathcal{A}_p \times \mathcal{B}_p = \mathcal{S}_r$, and $\forall p, q \in \mathcal{N}_k, p \neq q \Rightarrow \mathcal{A}_p \times \mathcal{B}_p \cap \mathcal{A}_q \times \mathcal{B}_q = \emptyset$.

We can relate MDD nodes and the p-flows by proving that all submarkings described by $\mathcal{C}_{p,q}$, therefore by $\mathcal{A}_p$, have the same partial sum for any given p-flow. Given nodes p and q with $p.lvl = k > q.lvl = h$, $\sigma = (i_k, \ldots, i_{h+1}) \in \mathcal{C}_{p,q}$, and a p-flow $\boldsymbol{\pi} \in \mathcal{F}$, we let the partial sum of submarking σ for invariant $\boldsymbol{\pi}$ be:

$$Sum(p, q, \sigma, \boldsymbol{\pi}) = \textstyle\sum_{p.lvl \geq j > q.lvl} i_j \cdot \boldsymbol{\pi}[\lambda^{-1}(j)].$$

In particular, for any $\sigma \in \mathcal{C}_{r,\top} = \mathcal{S}_{rch}$, we have $Sum(r, \top, \sigma, \boldsymbol{\pi}) = Tc(\boldsymbol{\pi})$.

We can now introduce two fundamental properties enjoyed by an MDD encoding a state space subject to a set of p-flows $\mathcal{F}$, which will pave the way to the definition of our new metric called $\mathrm{i}_{\mathrm{Rank}}$.

Theorem 3. Assume a set of states $\mathcal{S}$ subject to the set of p-flows $\mathcal{F}$ is encoded by an MDD rooted at r. Then, all paths between a given pair of nodes have the same partial sum for any given invariant: $\forall \sigma, \sigma' \in \mathcal{C}_{p,q}, \forall \boldsymbol{\pi} \in \mathcal{F}$, $Sum(p, q, \sigma, \boldsymbol{\pi}) = Sum(p, q, \sigma', \boldsymbol{\pi})$. We can therefore write $Sum(p, q, \boldsymbol{\pi})$.

Proof. Consider two nodes p and q, with $p.lvl = k > q.lvl = h$, two paths σ and σ' from p to q, and any $\sigma_a \in \mathcal{A}_p$ and $\sigma_b \in \mathcal{B}_q$, so that both $(\sigma_a, \sigma, \sigma_b)$ and $(\sigma_a, \sigma', \sigma_b)$ describe markings in $\mathcal{S}$. Then, for any p-flow $\boldsymbol{\pi} \in \mathcal{F}$, we have that $Sum(r, \top, (\sigma_a, \sigma, \sigma_b), \boldsymbol{\pi}) = Sum(r, \top, (\sigma_a, \sigma', \sigma_b), \boldsymbol{\pi}) = Tc(\boldsymbol{\pi})$. However, $Sum(r, \top, (\sigma_a, \sigma, \sigma_b), \boldsymbol{\pi}) = Sum(r, p, \sigma_a, \boldsymbol{\pi}) + Sum(p, q, \sigma, \boldsymbol{\pi}) + Sum(q, \top, \sigma_b, \boldsymbol{\pi}) = Sum(r, \top, (\sigma_a, \sigma', \sigma_b), \boldsymbol{\pi}) = Sum(r, p, \sigma_a, \boldsymbol{\pi}) + Sum(p, q, \sigma', \boldsymbol{\pi}) + Sum(q, \top, \sigma_b, \boldsymbol{\pi})$, thus we must have $Sum(p, q, \sigma, \boldsymbol{\pi}) = Sum(p, q, \sigma', \boldsymbol{\pi})$. □

An even stronger property holds if the MDD encodes $\mathcal{S}_{sat}$: then, every node in the MDD is completely identified by a unique pattern of partial p-flow sums.

Theorem 4. Let $\mathcal{S}_{sat}$ be encoded by an MDD rooted at r. Then, the nodes at level k have different partial sums:

$$\forall p, p' \in \mathcal{N}_k,\ p \neq p' \ \Rightarrow\ \exists \boldsymbol{\pi} \in \mathcal{F},\ Sum(r, p, \boldsymbol{\pi}) \neq Sum(r, p', \boldsymbol{\pi}).$$

Proof. Remember that $\mathcal{S}_{sat} = \{\mathbf{m} \in \mathbb{N}^{\mathcal{P}} : \forall \boldsymbol{\pi} \in \mathcal{F}, \mathbf{m} \cdot \boldsymbol{\pi} = Tc(\boldsymbol{\pi})\}$. Assume that distinct nodes $p, p' \in \mathcal{N}_k$ satisfy $\forall \boldsymbol{\pi} \in \mathcal{F} : Sum(r, p, \boldsymbol{\pi}) = Sum(r, p', \boldsymbol{\pi})$. Since the MDD is canonical, p and p' must encode different sets, thus there must be a σ in $\mathcal{B}_p \setminus \mathcal{B}_{p'}$ or in $\mathcal{B}_{p'} \setminus \mathcal{B}_p$ (w.l.o.g. assume the former case). Then, considering any $\sigma_a \in \mathcal{A}_p$ and $\sigma'_a \in \mathcal{A}_{p'}$, we have $(\sigma_a, \sigma) \in \mathcal{S}_{sat}$ and $(\sigma'_a, \sigma) \notin \mathcal{S}_{sat}$. But $(\sigma_a, \sigma) \in \mathcal{S}_{sat}$ implies $\forall \boldsymbol{\pi} \in \mathcal{F}, Sum(r, \top, (\sigma_a, \sigma), \boldsymbol{\pi}) = Tc(\boldsymbol{\pi})$ and, since $Sum(r, \top, (\sigma_a, \sigma), \boldsymbol{\pi}) = Sum(r, p, \sigma_a, \boldsymbol{\pi}) + Sum(p, \top, \sigma, \boldsymbol{\pi}) = Sum(r, p', \sigma'_a, \boldsymbol{\pi}) + Sum(p, \top, \sigma, \boldsymbol{\pi}) = Sum(r, \top, (\sigma'_a, \sigma), \boldsymbol{\pi})$, and this holds for any $\boldsymbol{\pi}$ in $\mathcal{F}$, we should also have $(\sigma'_a, \sigma) \in \mathcal{S}_{sat}$, a contradiction. □

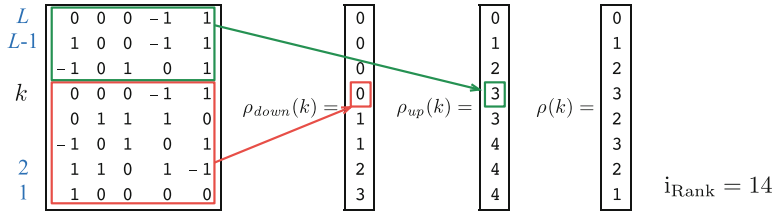

Fig. 4. Computations of the rank weights from a matrix $\mathbf{F}$ with $\text{rank}(\mathbf{F}) = 4$.

Theorem 4 implies that every node in the MDD encoding $\mathcal{S}_{sat}$ is completely identified by a unique pattern of partial p-flow sums. However, not every p-flow is relevant at a given level k of the MDD, and, more importantly, the portions from level L to level k of different p-flows may encode the same information, i.e., may be *linearly dependent*, yet these redundant portions contribute to the computation of the PF metric. i_{Rank}, then, attempts to estimate the number of *possible* combinations of partial path sums that may actually be found in the nodes at level k of the MDD, taking into account these linear dependencies.

To this end, we consider the $|\mathcal{P}| \times |\mathcal{F}_{min}|$ matrix $\mathbf{F}$ (rows ordered according to λ, columns in any order) describing the p-flows in $\mathcal{F}_{min}$, and define the number $\rho_{up}(k)$ of linearly independent partial p-flows up to level k:

$$\rho_{up}(k) = \text{rank}\big(\mathbf{F}[L : k+1, \cdot]\big),$$

where $\mathbf{F}[L : k+1, \cdot]$ is the submatrix of $\mathbf{F}$ with rows L through $k+1$ (level k is excluded because we are counting the partial sums *reaching* level k). $\rho_{up}(k)$ counts both p-flows active at level k and those that are not, as the lowest place in their support is mapped to a level above k (p-flow already "closed" at level k). The number $\rho_{down}(k)$ of linearly independent closed p-flows at level k is obtained by subtracting the rank of submatrix $\mathbf{F}[k : 1, \cdot]$ from the rank of the entire matrix $\mathbf{F}$:

$$\rho_{down}(k) = \text{rank}(\mathbf{F}) - \text{rank}\big(\mathbf{F}[k : 1, \cdot]\big).$$

Then, the value we are seeking is the difference of these two quantities:

$$\rho(k) = \rho_{up}(k) - \rho_{down}(k).$$

Figure 4 depicts the definition of $\rho_{up}(k)$ and $\rho_{down}(k)$. The rectangles in the invariant matrix $\mathbf{F}$ represents the portions used to compute the ranks for level k. The values of $\rho_{up}(k)$, $\rho_{down}(k)$, and $\rho(k)$, for all levels k, are listed on the right.

The value of the $\mathrm{i}_{\mathrm{Rank}}$ metric is then the sum of all the $\rho(k)$ values:

$$\mathrm{i}_{\mathrm{Rank}} = \textstyle\sum_{1 \le k \le L} \rho(k),$$

which can be thought of as an estimate of the number of independent factors affecting the number of MDD nodes at the various levels. Thus, we should expect that a linear increase in $\mathrm{i}_{\mathrm{Rank}}$ implies an exponential increase in the MDD size. The main advantage of $\mathrm{i}_{\mathrm{Rank}}$ is that it does not suffer in the presence of an excessive number of p-flows (as do PF and PSF). Indeed, since the metric is computed on the rank of $\mathbf{F}$ and on the rank of sets of rows of $\mathbf{F}$, and since these ranks do not change while adding linear combinations of p-flows (larger $\mathbf{F}$) or by removing p-flows (smaller $\mathbf{F}$) as long as we remove only linear dependent vectors, we have a metric that is rather robust. Additionally, it is also fairly inexpensive to compute, $O(\min\{\mathcal{P}, \mathcal{T}\}^3)$.

5 Experimental Assessment of the Metrics

We now experimentally assess the efficacy of PF and $\mathrm{i}_{\mathrm{Rank}}$: since the relationship between p-flows and MDD nodes is stronger for $\mathcal{S}_{sat}$ than for $\mathcal{S}_{rch}$ (Theorem 4), we expect higher correlation when the MDD encodes the former. We also seek to determine whether these metrics can be used to drive iterative heuristics or meta-heuristics that compute variable orders. All experiments are on different sets of orders for 40 models taken from the Petri Net Repository [2]. The experiments have been conducted using the GreatSPN tool [1,4], which uses the Meddly library [8]. All MDDs generated had fewer than one million nodes. We follow the evaluation procedure of [6] and compute the Spearman coefficient of correlation (CC), whose interpretation is: $[1, 0.8]$ means very strong correlation, $[0.6, 0.8]$ strong correlation, $[0.4, 0.6]$ moderate correlation, and so on decreasing. Negative values indicate anti-correlation.

Figure 5 compares the correlation of $\mathrm{i}_{\mathrm{Rank}}$ and PF to that of the metrics of Sect. 2.3. Although all experiments have been performed, for sake of space only 6 metrics are considered in the tables. We have chosen to include PSF, PF and $\mathrm{i}_{\mathrm{Rank}}$ (for obvious reasons), plus the best among the $\mathbf{C}$ span metrics (SOUPS), and two versions of PTS (PTS and PTS^P, without and with p-flow) since PTS is the metrics implicitly optimized by the widely used FORCE heuristic. No bandwidth metric is reported since they all exhibit at best a moderate correlation. Each row represents a metric, columns report the CC of the metrics with the MDD encoding $\mathcal{S}_{sat}$ (columns [A] and [B]) and $\mathcal{S}_{rch}$ (columns [C] and [D]) for two different sets of orders. The CC of a single model for a single metric is computed from the bivariate series relating, for each variable order λ, the MDD size built using λ with the value of the metric for that λ. ICC is the CC computed over the set of orders λ in $\mathcal{V}_{\mathrm{IMPR}}$ and BCC is computed over $\mathcal{V}_{\mathrm{BEST}}$. The sets $\mathcal{V}_{\mathrm{IMPR}}$ and $\mathcal{V}_{\mathrm{BEST}}$ are built from 1,000 initial random orders by generating sequences of

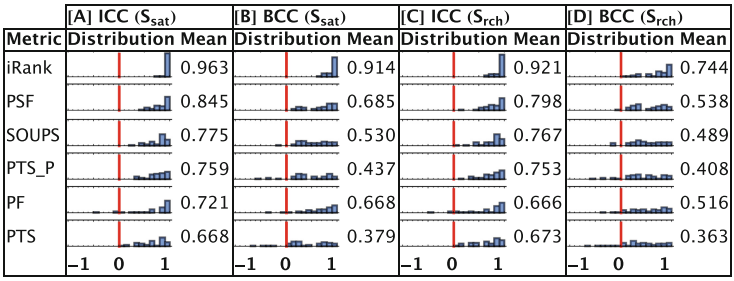

Fig. 5. Two correlation coefficients for different metrics for $\mathcal{S}_{sat}$ and $\mathcal{S}_{rch}$

increasingly better orders (in terms of MDD final size) until a convergence criterion is satisfied; $\mathcal{V}_{\text{IMPR}}$ retains all orders while $\mathcal{V}_{\text{BEST}}$ retains only the last orders in each sequence (thus exactly 1,000 orders). This construction is explained in [6], where it was observed that $\mathcal{V}_{\text{BEST}}$ tends to contain mostly good orders, and $\mathcal{V}_{\text{IMPR}}$ a mixture of good and bad orders. The above sets have been built for each of the 40 models. For each combination, we report the mean CC (over all models) and the CC distribution for the 40 models; the x axis is partitioned into 20 bins, so the y axis indicates the number of models whose CC falls into each bin. All plots have the same scale on the y axis. and the height of the bar at 0 is fixed at 36 for all rows.

The results of Fig. 5 indicate that i_{Rank} has the highest correlation for both ICC and BCC and for both $\mathcal{S}_{rch}$ and $\mathcal{S}_{sat}$. i_{Rank} is better than the second best by 12% (ICC on $\mathcal{S}_{sat}$) and up to 28% (BCC on $\mathcal{S}_{rch}$). The comparison with PTS (the metric used as a convergence criteria by the widely used FORCE heuristic) is even more striking. It is also evident that in none of the four cases PF performs better than PSF, supporting our observation that considering more p-flows is not always (or even usually) a good idea. Figure 5 also indicates that all metrics have better CC when the MDD encodes $\mathcal{S}_{sat}$ (column [A] vs. [C], and column [B] vs. [D]). This is not surprising for i_{Rank}, given Theorem 4, but it also holds for all other metrics. This could be due to the fact that, since $\mathcal{S}_{rch} \subseteq \mathcal{S}_{sat}$, the MDD for $\mathcal{S}_{rch}$ encodes additional constraints not captured by any of the metrics.

Comparing columns [A] and [B] (and columns [C] and [D]) of Fig. 5, we observe that ICC is higher than BCC for all metrics, meaning that they have better correlation when the set of considered orders is $\mathcal{V}_{\text{IMPR}}$ (mix of good and bad orders) rather than $\mathcal{V}_{\text{BEST}}$ (mostly good orders). This is related to the use of the Spearman CC, which quantifies how well the i-th largest value of the metric correlates with the i-th largest value of the MDD size: certainly with $\mathcal{V}_{\text{BEST}}$ we tend to have more MDDs of similar size, making it more difficult to discriminate.

The experiments reported in Fig. 6 serve to evaluate whether the metrics can be used as an objective function inside a simulated annealing procedure (columns [A] and [C]) or as a meta-heuristic to select one among the orders produced during the simulated annealing (columns [B] and [D]). Given an initial variable order and a metric m the procedure searches an "optimal" order through

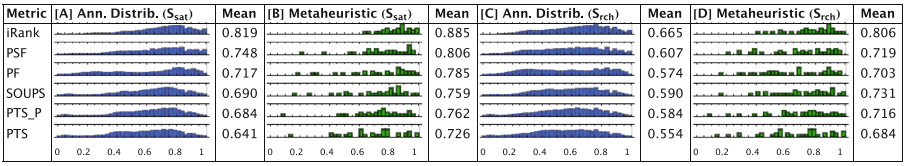

Fig. 6. Evaluation of metrics on simulated annealing produced orders

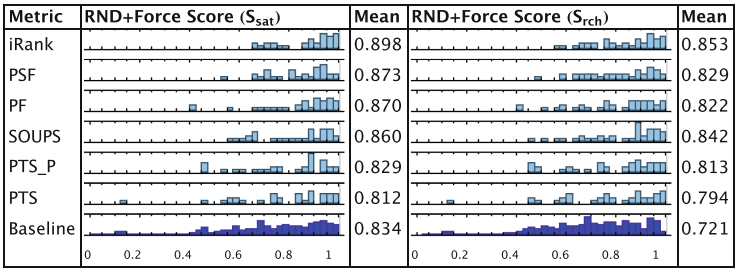

Fig. 7. Evaluation of metrics on FORCE-produced orders.

a simulated annealing procedure [17], aimed at minimizing the value of m. We employ a standard simulated annealing procedure, described in [6]. Unlike the construction of the set of orders used for the computation of ICC and BCC in Fig. 5, no MDD is built during the construction of the candidate variable order. For each metric, the simulated annealing procedure is run 1,000 times, from different initial orders, and Fig. 6 reports, in columns [A] and [C], the mean and distribution of the "score" of the MDDs built using the 1,000 orders produced by the 1,000 runs of the simulated annealing for each metric m, for the 40 models. The score is the distance from the size of the smallest MDD built, normalized on the distance between the smallest and the largest MDD size built (see [6], Eq. 5), obviously computed separately for each model. A value of 1 for order λ for a given model indicates that the smallest MDD seen for that model was built using λ. A value of 0 indicates the worst order. Column [A] refers to the MDDs storing $\mathcal{S}_{sat}$, while column [C] refers to $\mathcal{S}_{rch}$. Again, $\mathrm{i}_{\mathrm{Rank}}$ performs better than any other metrics in both cases.

Columns [B] and [D] instead report the results of using each metric m as a meta-heuristic: for each model a single order is chosen (the order with the best value for metric m), and the 40 resulting scores are plotted. This corresponds to using metrics in practice to select a given order for a model. Again, $\mathrm{i}_{\mathrm{Rank}}$ shows the best performance, indicating that it can select good candidate orders.

Figure 7 shows the evaluation of a meta-heuristic also defined in [6], based on FORCE. Each metric m is used to drive the selection of the "best" variable order among a set of variable orders produced using FORCE from an initial set of 1,000 random orders. This is done for each of the 40 models. The last row is the baseline (40×1,000 points, all computed using FORCE), while all other

histograms are built out of 40 MDD sizes, one per model. A mean value greater than the baseline mean indicates that the metric selects the best orders among the ones computed by FORCE. A mean below the baseline indicates otherwise. Again, when we employ i_{Rank} to select the order to use, we get a better score than with any other metric for both $\mathcal{S}_{sat}$ (left column) and $\mathcal{S}_{rch}$ (right column).

6 Conclusion and Future Work

We considered the problem of defining and evaluating variable orders for MDDs encoding either the reachable states of a DEDS ($\mathcal{S}_{rch}$) or the states satisfying a set of linear invariants ($\mathcal{S}_{sat}$). We studied the relation between the MDD size and structure and the linear invariants, and proposed two new metrics: PF, a trivial extension to PSF; and i_{Rank}. Through a set of experiments, metrics have been evaluated both as predictors of the MDD size and as drivers for two heuristics (and associated meta-heuristics). The experiments follow the procedure proposed in [6], as defining a good and fair procedure to compare metrics and MDD sizes for a set of models is a nontrivial task. The results show that i_{Rank} is better than any other metrics we found in the literature.

The definition of i_{Rank}, and PF, assumes that linear invariants are available. For DEDSs specified as Petri nets, the linear invariants are derived from the p-flows, the left annullers of the incidence matrix, an integer matrix describing how an event modifies a state. Clearly, whenever a DEDS can be specified through a similar matrix, the application of our method is straightforward, as in the case of various formalisms used in system modeling and verification. For other formalisms, this may be less immediate, but our method only assumes a set of linear invariants on the state space, regardless of how they are computed.

In our experiments, we considered only *conservative* Petri nets, where each place appears in at least one invariant. This allowed us to compare with previously defined metrics that exploit linear invariants generated from p-semiflows. If no invariants are available, or if most places are not part of any invariant, PF and i_{Rank} could perform very poorly. If a net is not conservative, a subset of places may "lose" tokens, "gain tokens", or both. The last two cases cause $\mathcal{S}_{rch}$ to be infinite, but the first case can still be managed by our approach, thanks to p-flows. As an example, consider the net obtained from the net in Fig. 1(B) by removing the arc from transition T_3 back to P_0 Such a net does not have any p-semiflow, but all the places between each pair of fork-and-join belong to a p-flow, allowing us to apply our method. A further extension could consider invariants where the weighted sum of tokens in a subset of places is less than or equal a constant (instead of just equal).

Several directions for additional exploration remain. First, i_{Rank} does not consider the initial state of the DEDS, but the number of nodes at a given level depends on the token count of the p-flows, and this may be especially important when the p-flows have significantly different token counts. Then, the efficient computation of i_{Rank} is obviously important, as heuristics using it could probably evaluate it many times. The computation could be expensive since it involves matrix rank computations.

Finally, we are interested in extending i_{Rank} to more general constraints, which can still provide hints on good variable orders; for example, a constraint "if $A = 3$ then $B = C$" imposes no limitations on C along paths where $A \neq 3$, (assuming A is above B and B is above C in the MDD), but, requires to remember the value of B until reaching C along paths where $A = 3$.

References

1. The GreatSPN tool homepage. http://www.di.unito.it/~greatspn/index.html
2. MCC: The Model Checking Competition. https://mcc.lip6.fr
3. Aloul, F.A., Markov, I.L., Sakallah, K.A.: FORCE: a fast and easy-to-implement variable-ordering heuristic. In: Proceedings of GLSVLSI, pp. 116–119. ACM, New York (2003)
4. Amparore, E.G., Balbo, G., Beccuti, M., Donatelli, S., Franceschinis, G.: 30 years of GreatSPN. In: Fiondella, L., Puliafito, A. (eds.) Principles of Performance and Reliability Modeling and Evaluation. SSRE, pp. 227–254. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30599-8_9
5. Amparore, E.G., Beccuti, M., Donatelli, S.: Gradient-based variable ordering of decision diagrams for systems with structural units. In: D'Souza, D., Narayan Kumar, K. (eds.) ATVA 2017. LNCS, vol. 10482, pp. 184–200. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68167-2_13
6. Amparore, E.G., Ciardo, G., Donatelli, S.: Variable order metrics for decision diagrams in system verification (2018, submitted for publication). http://www.di.unito.it/~amparore/metrics_STTT.pdf
7. Amparore, E.G., Donatelli, S., Beccuti, M., Garbi, G., Miner, A.: Decision diagrams for Petri nets: which variable ordering? In: Transactions on Petri Nets and Other Models of Concurrency XI. Springer, Heidelberg (2019, to appear)
8. Babar, J., Miner, A.: Meddly: multi-terminal and edge-valued decision diagram library. In: International Conference on Quantitative Evaluation of Systems, pp. 195–196. IEEE Computer Society, Los Alamitos (2010)
9. Bollig, B., Wegener, I.: Improving the variable ordering of OBDDs is NP-complete. IEEE Trans. Comput. **45**(9), 993–1002 (1996)
10. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Trans. Comput. **35**(8), 677–691 (1986)
11. Ciardo, G., Jones, R.L., Miner, A.S., Siminiceanu, R.: Logical and stochastic modeling with SMART. Perf. Eval. **63**, 578–608 (2006)
12. Ciardo, G., Lüttgen, G., Yu, A.J.: Improving static variable orders via invariants. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 83–103. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73094-1_8
13. Ciardo, G., Zhao, Y., Jin, X.: Ten years of saturation: a Petri net perspective. In: Jensen, K., Donatelli, S., Kleijn, J. (eds.) Transactions on Petri Nets and Other Models of Concurrency V. LNCS, vol. 6900, pp. 51–95. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29072-5_3
14. Cimatti, A., Clarke, E.M., Giunchiglia, F., Roveri, M.: NuSMV: a new symbolic model verifier. In: Halbwachs, N., Peled, D. (eds.) CAV 1999. LNCS, vol. 1633, pp. 495–499. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48683-6_44
15. Colom, J.M., Silva, M.: Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal p-semiflows. In: Rozenberg, G. (ed.) ICATPN 1989. LNCS, vol. 483, pp. 79–112. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-53863-1_22

16. Davies, I., Knottenbelt, W., Kritzinger, P.S.: Symbolic methods for the state space exploration of GSPN models. In: Field, T., Harrison, P.G., Bradley, J., Harder, U. (eds.) TOOLS 2002. LNCS, vol. 2324, pp. 188–199. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46029-2_12
17. Du, K.L., Swamy, M.N.S.: Search and Optimization by Metaheuristics. Springer, Basel (2016). https://doi.org/10.1007/978-3-319-41192-7
18. Lind-Nielsen, J.: BuDDy Manual, July 2003. http://buddy.sourceforge.net/manual/main.html
19. Kam, T., Villa, T., Brayton, R.K., Sangiovanni-Vincentelli, A.: Multi-valued decision diagrams: theory and applications. Multiple-Valued Logic **4**(1), 9–62 (1992)
20. Kwiatkowska, M., Norman, G., Parker, D.: PRISM: probabilistic model checking for performance and reliability analysis. Perform. Eval. **36**(4), 40–45 (2009)
21. Martinez, J., Silva, M.: A simple and fast algorithm to obtain all invariants of a generalized Petri net. In: Girault, C., Reisig, W. (eds.) Application and Theory of Petri Nets. INFORMATIK, vol. 52, pp. 301–310. Springer, Heidelberg (1982). https://doi.org/10.1007/978-3-642-68353-4_47
22. McMillan, K.L.: Symbolic Model Checking. Kluwer Academic Publishers, Norwell (1993)
23. Meijer, J., van de Pol, J.: Bandwidth and wavefront reduction for static variable ordering in symbolic reachability analysis. In: Rayadurgam, S., Tkachuk, O. (eds.) NFM 2016. LNCS, vol. 9690, pp. 255–271. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40648-0_20
24. Siminiceanu, R.I., Ciardo, G.: New metrics for static variable ordering in decision diagrams. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 90–104. Springer, Heidelberg (2006). https://doi.org/10.1007/11691372_6
25. Smith, B., Ciardo, G.: SOUPS: a variable ordering metric for the saturation algorithm. In: Proceedings of the International Conference on Application of Concurrency to System Design (ACSD). IEEE Computer Society Press (2018)
26. Somenzi, F.: Efficient manipulation of decision diagrams. STTT **3**(2), 171–181 (2001)
27. Thierry-Mieg, Y.: Symbolic model-checking using ITS-tools. In: Baier, C., Tinelli, C. (eds.) TACAS 2015. LNCS, vol. 9035, pp. 231–237. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46681-0_20
28. Wan, M., Ciardo, G., Miner, A.S.: Approximate steady-state analysis of large Markov models based on the structure of their decision diagram encoding. Perf. Eval. **68**, 463–486 (2011)