Parameterised Bounds on the Sum of Variables in Time-Series Constraints*

N. Beldiceanu[1], M. I. Restrepo[1], and H. Simonis[2]

[1] TASC (LS2N-CNRS), IMT Atlantique, FR – 44307 Nantes, France
nicolas.beldiceanu@imt-atlantique.fr mrestrep@uco.fr
[2] Insight Centre for Data Analytics, University College Cork, Ireland
helmut.simonis@insight-centre.org

Abstract. For two families of time-series constraints with the aggregator `Sum` and features `one` and `width`, we provide parameterised sharp lower and upper bounds on the sum of the time-series variables wrt these families of constraints. This is important in many applications, as this sum represents the cost, for example the energy used, or the manpower effort expended. We use these bounds not only to gain a priori knowledge of the overall cost of a problem, we can also use them on increasing prefixes and suffixes of the variables to avoid infeasible partial assignments under a given cost budget. Experiments show that the bounds drastically reduce the effort to find cost limited solutions.

1 Introduction

Time series is an increasingly important format of data in many applications, from financial to scientific. Time series are sequences of values taken at successive equally spaced points in time. Two traditional topics are time series forecasting [16] and time series pattern recognition [19, 15]. A more recent topic is the generation of time series satisfying a given set of constraints. Indeed, in an industrial or commercial setting, time series are constrained by physical laws or organisational regulations. In this case, when time series correspond to a resource produced or consumed, the question of maximising or minimising the sum of the elements of a time series becomes important. This article focuses on this issue.

Context and motivation From a constraint perspective work on time-series constraints was introduced in [13] to formalise the notions of exact and approximate similarity between time-series patterns and data. More recently, some authors have proposed quantitative regular expressions [2, 1] as a way to *(i)* formalise and identify common types of time-series patterns [9, 18], and to *(ii)* express time-series constraints, which are then used to generate constrained time series. To improve propagation, implied constraints and cuts were derived in [6, 7, 3].

* This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 which is co-funded under the European Regional Development Fund as well as from the Gaspard-Monge program.

These ideas have been used to solve real-life problems including the analysis of the output of electric power stations over multiple days [11], the solution of a staff scheduling problem in a service company [5], power management for large-scale distributed systems [10] and the generation of typical energy consumption profiles of a data centre [12, 14]. Most of these problems require the incorporation of an objective function which is represented by the sum of the decision variables. Hence, computing bounds on such sum is an important issue.

Time-series constraints A time-series constraint $\gamma(X, R)$ is a constraint which restricts an integer result variable R to be the result of some computations over a sequence of n integer variables X. The components of a time-series constraint we reuse from [9] are a *pattern* σ, a *feature* f, and an *aggregator* g. A pattern is described by a *regular expression* over the alphabet $\Sigma = \{\text{`} < \text{'}, \text{`} = \text{'}, \text{`} > \text{'}\}$ whose language $\mathcal{L}_\sigma$ does not contain the empty word. For instance, in [4] the `Plateau` pattern is characterised by the expression ` <=*> `. A feature and an aggregator are functions over integer sequences.

- A time-series constraint with the aggregator `Sum` and the feature `one` restricts R to be the number of occurrences of pattern σ in X.
- A constraint with the aggregator `Sum` and the feature `width` restricts R to be the sum of the widths of the maximal occurrences of pattern σ in the integer sequence. The *width* of an occurrence of σ is the number of time-series variables included in σ minus a constant corresponding to the sum of two integer trimming values. For instance, consider a time series $X = \langle 0, 3, 3, 0 \rangle$ with one occurrence of $\sigma =$ `Plateau` $=$ ` <=*> `; the `width` of the occurrence of σ is equal to 2, as the two integer trimming values of σ are equal to 1.

Motivating example Assume we have to generate a time series of size $n = 14$ with $R = 5$ increasing terraces, i.e. $\sigma =$ ` <=+< `, while maximising the sum of the 14 variables, each restricted to be in $[\ell, u] = [2, 6]$. Ignoring the 5 terraces leads to an upper bound of 84, while, as shown in Part (D) of Fig. 1, considering the 5 terraces gives a sharp upper bound $n \cdot u - p \cdot (2 \cdot t + s + 1) - r \cdot (2 \cdot s + 3) = 59$. The procedure for deriving the formulas $p = \min(R, \lfloor \frac{n-2 \cdot R}{2} \rfloor)$, $s = \lfloor \frac{R}{\max(1,p)} \rfloor$, $t = \frac{s \cdot (s+1)}{2}$, $r = R \mod \max(1, p)$, is presented in Section 3. Our goal is to *find a method to derive such formula for different patterns.*

Focus and contributions of this paper We focus on the $g\_f\_\sigma(X, R)$ families of time-series constraints with g being `Sum`, with f being either `one` or `width`, and with σ being a pattern described by a regular expression over the alphabet $\Sigma = \{\text{`} < \text{'}, \text{`} = \text{'}, \text{`} > \text{'}\}$. Our contributions consist of *parameterised sharp upper and lower bounds on the sum of the time-series variables* for the SUM_ONE_$\sigma(X, R)$ (also denoted as NB_$\sigma(X, R)$) and the SUM_WIDTH_$\sigma(X, R)$ families provided all X variables are in the interval $[\ell, u]$. The parameters in the bounds correspond to the sequence length, the values ℓ and u, and the regular expression σ. The limits ℓ and u are typically given by physical limitations of the system, which are time independent, and therefore apply to all variables. The parameterised

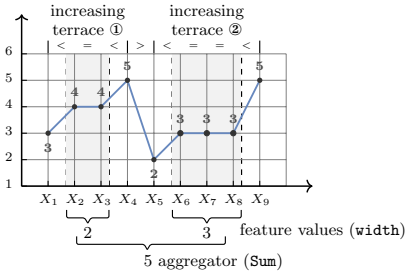
bounds are valid provided some condition on the regular expression σ holds, which in practice is true for 80% of the 22 regular expressions of [4]. Note that an approach encoding the full problem with an automaton would lead to a pseudo-polynomial algorithm since such automaton would have $O(n^2u^3)$ states: assuming $\ell = 0$, each state would record the values of R (from 0 to n), of X_{i-1} (from 0 to n), of the partial sum $X_1 + \cdots + X_i$ (from 0 to $u \times n$), and would have up to u outgoing transitions.

Outline of the paper Sec. 2 presents a background on time-series constraints. Sec. 3 introduces our contribution, a unique per family expression that defines upper and lower bounds on the sum of the time-series variables wrt the time-series constraints. Sec. 4 evaluates the impact of the bounds. Sec. 5 concludes.

2 Background

We present the background to define bounds on the sum of the time-series variables wrt the time-series constraints with aggregator `Sum` and features `one` and `width`. A time-series constraint imposed on a sequence of n integer variables $X = \langle X_1, ..., X_n \rangle$ and a result variable R is described by a feature f, an aggregator g, and a pattern σ as mentioned in the introduction. Let $S = \langle S_1, ..., S_{n-1} \rangle$ be the *signature* of a time series X, which is defined by: $(X_i < X_{i+1} \Leftrightarrow S_i = \text{`} < \text{'}) \wedge (X_i = X_{i+1} \Leftrightarrow S_i = \text{`} = \text{'}) \wedge (X_i > X_{i+1} \Leftrightarrow S_i = \text{`} > \text{'})$ for all $i \in [1, n-1]$. If a sub-signature $\langle S_i, ..., S_j \rangle$ is a maximal word matching σ in the signature of X, then the subsequence $\langle X_{i+b_\sigma}, ..., X_{j+1-a_\sigma} \rangle$ is called a *σ-pattern*, and the subsequence $\langle X_i, ..., X_{j+1} \rangle$ is called an *extended σ-pattern*. The constants b_σ and a_σ respectively trim the left and right borders of an extended σ-pattern to obtain a σ-pattern from which a feature value is computed. They are useful when there is the need to perform computations from only a part of the occurrence of σ, as shown in Ex. 1. As in [9], we assume σ-patterns not to overlap.

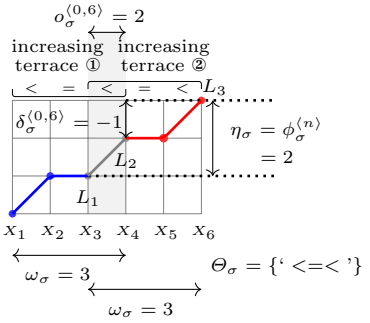
Example 1. Consider the σ = `IncreasingTerrace` = ' $<=^+<$ ' regular expression with $a_\sigma = b_\sigma = 1$ and the time series X shown in the figure in the right over the interval $[2, 5]$ and with signature $S = \langle <, =, <, >, <, =, =, < \rangle$. A σ-pattern called increasing terrace within X is a subset whose signature is a maximal occurrence of σ in the signature of X. Time series X contains two increasing terraces, labelled ① and ②, namely $\langle 3, 4, 4, 5 \rangle$ and $\langle 2, 3, 3, 3, 5 \rangle$ with widths 2 and 3, respectively. Hence, the aggregation of the number of occurrences using the aggregator `Sum` is 2 and the aggregation of their widths using `Sum` is 5. The corresponding time-series constraints are NB_$\sigma(X, R)$ and SUM_WIDTH_$\sigma(X, R)$, respectively. △



Regular-expression characteristics were introduced as a way to parameterise the bounds on the result value of time-series constraints [8] and to derive AMONG implied constraints [6] in a systematic way. We now present a brief definition of the characteristics we reuse in this paper and illustrate them with one example.

- The *size* of σ, denoted by ω_σ, is the length of a shortest word in the language $\mathcal{L}_\sigma$ of the regular expression σ.
- The *height* of σ, denoted by η_σ, is the smallest difference between the domain upper and lower limits, i.e. $u - \ell$, such that there is a *ground time series* (all X_i are fixed) over $[\ell, u]$ whose signature has at least one occurrence of σ.
- The *range* of σ wrt n, denoted by $\phi_\sigma^{\langle n \rangle}$, is the minimum difference between the maximum and the minimum values in an extended σ-pattern of width n.
- The *set of inducing words* of σ, denoted by Θ_σ, is a subset of $\mathcal{L}_\sigma$ such that for every word v in $\mathcal{L}_\sigma$, there exists a word $w = w_1 w_2 ... w_k$ in Θ_σ such that every w_i is non-empty and every v in $\mathcal{L}_\sigma$ can be represented as $v_1 w_1 v_2 w_2 ... v_k w_k v_{k+1}$ with every v_i being a word in $\{\text{`} < \text{'}, \text{`} = \text{'}, \text{`} > \text{'}\}^*$.
- The *overlap* of σ wrt $\langle \ell, u \rangle$, denoted by $o_\sigma^{\langle \ell, u \rangle}$, is the maximum number of time-series variables that belong simultaneously to two consecutive extended σ-patterns of a time series among all time series over $[\ell, u]$. If such maximum is not bounded, then $o_\sigma^{\langle \ell, u \rangle}$ is undefined.
- The *smallest variation of maxima* of σ wrt $\langle \ell, u \rangle$, denoted by $\delta_\sigma^{\langle \ell, u \rangle}$, corresponds to the smallest difference between the maximum values of two consecutive extended σ-patterns that have at least one common time-series variable.
- The *set of supporting time series* of a word v in $\mathcal{L}_\sigma$ wrt $\langle \ell, u \rangle$, denoted by $\Omega_\sigma^{\langle \ell, u \rangle}(v)$, is a set of time series where each element of $\Omega_\sigma^{\langle \ell, u \rangle}(v)$ is a time series over $[\ell, u]$ whose signature is v.

Example 2. Consider the $\sigma = \texttt{IncreasingTerrace} = \text{`} <=^+< \text{'}$ regular expression and the sequence $X = \langle 3, 4, 4, 5, 5, 6 \rangle$. The figure on the right illustrates regular-expression characteristics associated with X. The common time-series variables of the two consecutive extended σ-patterns are coloured in grey. The first (resp. second) extended σ-pattern is shown in blue (resp. red). Points L_1 and L_2 correspond to the overlap $o_\sigma^{\langle \ell, u \rangle}$. The difference between the y-coordinates of points L_2 and L_3 corresponds to the value of $\delta_\sigma^{\langle \ell, u \rangle}$. △



We reuse in Sec. 3 the notions of interval without restart and superposition of two words from [8] that we now recall. An *interval without restart* consists of a subsequence such that every two consecutive extended σ-patterns within this subsequence have $o_\sigma^{\langle \ell, u \rangle} > 0$ common time-series variables. The intervals without restart are always disjoint. Consequently, two consecutive extended σ-patterns

belonging to distinct intervals without restart do not share any time-series variables. The *superposition of two words* v and w in $\mathcal{L}_\sigma$ wrt $\langle \ell, u \rangle$ is the signature q of some ground time series over $[\ell, u]$ that contains at least two σ-patterns. For instance, the word $z =$ ' <=<=< ' is the superposition of the two increasing terraces in the figure from Ex. 2.

3 Bounds on the Sum of the Time-Series Variables

Consider a regular expression σ, an integer interval $[\ell, u]$, and a time series $X = \langle X_1, \ldots, X_n \rangle$, with every X_i ranging over $[\ell, u]$. We present a method to derive upper bounds on the sum of the X_i for $\text{NB\_}\sigma(X, R)$ and $\text{SUM\_WIDTH\_}\sigma(X, R)$. Wlog lower bounds are obtained in a similar way.

3.1 New Regular-Expression Characteristics

We present in this section two new regular-expression characteristics that will be used to maximise the sum of the time-series variables, while at the same time *(i)* constructing a fixed number of pattern occurrences, or *(ii)* building a number of pattern occurrences achieving a given total width. We first motivate and give the intuition of such characteristics in the context of the `IncreasingTerrace` = ' <=$^+$< ' pattern before providing their formal definitions.

- The first characteristic corresponds to the *maximum weight* of the inducing word of a regular expression σ. For example, given ' <=$^+$< ' and the domain value u, the maximum weight is the maximum value which can be achieved by a supporting time series of the inducing word ' <=< ', i.e. $(u-2)+(u-1)+(u-1)+u = 4 \cdot u - 4$.
- The second characteristic corresponds to the *weight of the overlap of the inducing word* of a regular expression σ with itself. We need to know this quantity to evaluate the maximum weight that can be achieved by a supporting time series of a stretch of overlapping inducing words. For example, given ' <=$^+$< ' and the domain value u, the maximum weight of the overlap highlighted in grey in '<=<=<' of two consecutive inducing words ' <=< ' is equal to $(u-2)+(u-1) = 2 \cdot u - 3$.

Definition 1 (Maximum weight of σ). *Consider a regular expression σ with exactly one word $v \in \Theta_\sigma$ with length ω_σ, and an integer interval domain $[\ell, u]$. The* maximum weight *of σ wrt $\langle u \rangle$, denoted by $\lambda_\sigma^{\langle u \rangle}$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z}$ to $\mathbb{Z}$. It is defined by $\lambda_\sigma^{\langle u \rangle} = u \cdot (\omega_\sigma + 1) - \nu_\sigma$, where ν_σ is the weight variation of σ. The function ν_σ maps an element of $\mathcal{R}_\Sigma$ to $\mathbb{Z}$,*

$$\nu_\sigma = \min_{t \in \Omega_\sigma^{\langle \ell, u \rangle}(v)} \left[(\omega_\sigma + 1) \cdot \max_{X_i \in t} X_i - \sum_{X_i \in t} X_i \right],$$

where t is a supporting time series of $v \in \Theta_\sigma$ wrt $\langle \ell, u \rangle$ denoted by $\Omega_\sigma^{\langle \ell, u \rangle}(v)$, and R_Σ denotes the set of regular expressions over the alphabet Σ.

Definition 2 (Total weight of the overlap of σ). *Consider a regular expression σ with exactly one word v in Θ_σ, and an integer interval domain $[\ell, u]$. The* total weight of the overlap *of σ wrt $\langle u \rangle$, denoted by $\alpha_\sigma^{\langle u \rangle}$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z}$ to $\mathbb{Z}$. It is defined by $\alpha_\sigma^{\langle u \rangle} = u \cdot o_\sigma^{\langle \ell, u \rangle} - \xi_\sigma$, where ξ_σ is the weight variation of the overlap of σ, defined by*

$$\xi_\sigma = \begin{cases} \min\limits_{t \in \Omega_\sigma^{\langle \ell, u \rangle}(v,v)} \left[o_\sigma^{\langle \ell, u \rangle} \cdot \max_{X_i \in t} X_i - \sum_{X_i \in t_o} X_i \right], & \text{if } \Gamma_\sigma^{\langle \ell, u \rangle}(v, v) \neq \emptyset \\ 0, & \text{otherwise.} \end{cases}$$

where $\Gamma_\sigma^{\langle \ell, u \rangle}(v, w)$ is the shortest superposition of words v and v in Θ_σ, $\Omega_\sigma^{\langle \ell, u \rangle}(v, v)$ is the supporting time series set for the shortest superposition between v and v wrt $\langle \ell, u \rangle$, and t_o is a subsequence of t corresponding to the overlap of two consecutive extended σ-patterns from $\Gamma_\sigma^{\langle \ell, u \rangle}(v, v)$.

Example 3. Consider $\sigma_1 =$ `StrictlyDecreasingSequence`, $\sigma_2 =$ `Peak` $=$ ' $<(<|=) * (>|=) * >$ ', and $\sigma_3 =$ `IncreasingTerrace` $=$ ' $<=^+<$ ', and the interval $[0, 3]$. Table 1 presents the values for the weight variation and the total weight regular-expression characteristics of the inducing words and the overlap of σ_1, σ_2 and σ_3. △

3.2 Time-Series Constraints With Feature ONE

We show how to derive bounds on the sum of the time-series variables for the NB_$\sigma(X, R)$ constraint family, provided all variables are in an interval $[\ell, u]$.

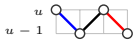
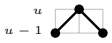
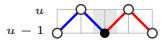
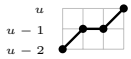
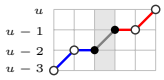
σ	word type	Θ_σ	length	illustration	new characteristics: weight variation	new characteristics: total weight
`Strictly-Decreasing-Sequence`	inducing word	' $>$ '	2		1	$2u - 1$
	overlap	-	0		0	$0u - 0$
`Peak`	inducing word	' $<>$ '	3		2	$3u - 2$
	overlap	-	1		1	$u - 1$
`Increasing-Terrace`	inducing word	' $<=<$ '	4		4	$4u - 4$
	overlap	-	2		3	$2u - 3$

Table 1. Regular-expression characteristics for `StrictlyDecreasingSequence`, `Peak`, `IncreasingTerrace`; column "length" gives the number of variables in the time series of interest, i.e. the number of filled dots in the column "illustration".

– First, Ex. 4 provides the basis for understanding the intuition of the method.
– Second, we list the properties required by a regular expression to use the intuitions we just described for deriving an upper bound.
– Finally, based on these properties, we give a greedy method to construct a time series that maximises the sum of its variables wrt the NB_$\sigma(X, R)$ family of time-series constraints.

From an intuition to a methodology

Example 4 (Intuition for constructing a time series reaching the upper bound). Fig. 1 gives three examples of how to build a time series that maximises the sum of its variables, while reaching a given number of pattern occurrences. Part (A) gives three constraints of the form NB_σ_i with σ_1 = `StrictlyDecreasingSequence` = ' $>^+$ ', σ_2 = `Peak` = ' $<(<\mid=)*(>\mid=)*>$ ', and σ_3 = `IncreasingTerrace` = ' $<=^+<$ ', respectively enforcing 3 occurrences of σ_1, 3 occurrences of σ_2, and 5 occurrences of σ_3.

– Since strictly decreasing sequences cannot overlap, Part (B) shows a time series with three intervals without restart where each interval corresponds to a strictly decreasing minimum size sequence positioned at its highest level, the remaining variables X_7, X_8 being set at their maximum value.
– Even if consecutive peaks may overlap, their maximal values may remain at the same level, Part (C) shows a time series with a single interval without restart containing three minimum size peaks positioned at their highest level, the remaining variables Y_8, Y_9 being set at their maximum value.
– As two consecutive intersecting increasing terraces are necessarily offset in height, Part (D) shows a time series containing the maximum number of possible intervals without restart, given that 5 increasing terraces have to be positioned in a sequence of size 14. The 5 terraces ①, ②, ..., ⑤ are distributed in two intervals without restart in the most balanced way, i.e. 3 and 2 terraces, by placing them at the highest possible level. △

To build a time series whose sum of variables is maximum, while having R maximal occurrences of the pattern σ, we proceed as follows.

– [MAXIMISING THE NUMBER OF VARIABLES SET TO u] We minimise the overall size taken by all R maximal occurrences of σ in order to set all remaining variables to their maximum value u.
– [POSITIONING PATTERN OCCURRENCES AS CLOSE AS POSSIBLE TO u] We try to position the R maximal occurrences of σ at their maximum height wrt to u. Unfortunately, as shown in Ex. 4 for the `IncreasingTerrace` pattern, this is not always possible: in Part (D) of Figure 1, only the terraces labelled with ① and ② are placed at their highest possible level. This can occur for patterns such that $o_\sigma^{\langle \ell,u \rangle} \neq 0$ and $\delta_\sigma^{\langle \ell,u \rangle} \neq 0$, when R is too large wrt the size of the time series. In this case, the R pattern occurrences are distributed in a balanced way over as many intervals without restart as possible.

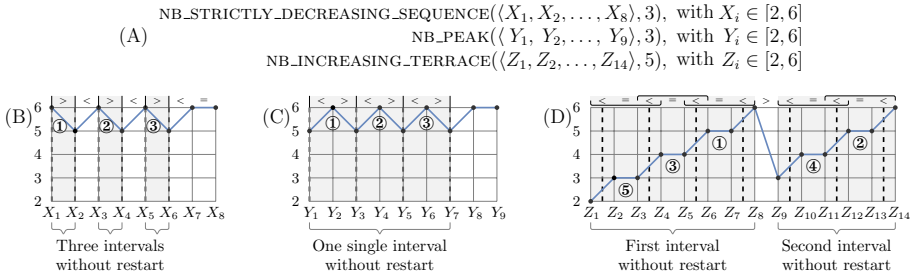


Fig. 1. (A) Three constraints and their corresponding time series that maximise the sum of the time-series variables respectively containing (B) three strictly decreasing sequences, (C) three peaks, and (D) five increasing terraces

– [SELECTING EACH PATTERN OCCURRENCE] Finally, each maximal occurrence of σ corresponds to a supporting time series $X_1, X_2, \ldots, X_{\omega_\sigma+1}$ of a word v of $\mathcal{L}_\sigma$ verifying simultaneously all the following conditions:
  i. v is a word whose size is as short as possible; hence its size is $\omega_\sigma + 1$.
  ii. $X_1, \ldots, X_{\omega_\sigma+1}$ minimises the variation wrt the maximum value of its variables, i.e. $(\omega_\sigma + 1) \cdot \max_{i \in [1,\omega_\sigma+1]} X_i \; - \; \sum_{i \in [1,\omega_\sigma+1]} X_i$.

Required properties of regular expressions. As shown before, building in a greedy way a time series t that maximises the upper bound on the sum of the time-series variables wrt a time-series constraint with aggregator `Sum` and feature `one`, requires finding R maximal words of $\mathcal{L}_\sigma$, such that the superposition of these R words wrt an integer interval domain $[\ell, u]$ *simultaneously optimises* several regular-expression characteristics. To define these properties, we use two regular-expression characteristics presented in Sec. 2 and Sec. 3.1: the set of inducing words and the weight variation of word v, denoted by Θ_σ and $\nu_\sigma(v)$.

Prop. 1 The language of σ does not include the word ' $=^+$ ', i.e., ' $=^+$ ' $\notin \mathcal{L}_\sigma$.
Prop. 2 Regular expression σ has only one inducing word, i.e., $\mid \Theta_\sigma \mid = 1$.
Prop. 3 The weight variation wrt the maximum domain value u of the only inducing word of σ, denoted by v, is lower than or equal to the weight variation of any other word in the language of σ, i.e., $\nu_\sigma(v) \leq \nu_\sigma(w)$, for each $w \in \mathcal{L}_\sigma : w \neq v$.

Prop. 1 guarantees that when the number of time-series variables included in the R maximal occurrences of σ is lower than the sequence length n, the time series t can be completed by setting all its reminder variables in the maximal domain value u. Prop. 2 guarantees that the smallest possible number of time-series variables is used to include R maximal occurrences of pattern σ in time series t. Prop. 3 ensures that the weight variation of a σ occurrence is minimised. Hence, the upper bound on the sum of the time-series variables associated with the R occurrences of σ in t is maximal. We show in Lemma 2 that these three properties give a sufficient condition for getting a sharp upper bound on the sum of time-series variables wrt a NB_$\sigma(X, R)$ constraint.

Structure of a time series achieving the upper bound on the sum of the time-series variables. Following the description of the methodology presented in Ex. 4, Lemma 2 defines the structure of a time series achieving the upper bound on the sum of time-series variables wrt a $\text{NB\_}\sigma(X, R)$ time-series constraint. For regular expressions with $o_\sigma^{\langle \ell,u \rangle} \neq 0$ and $\delta_\sigma^{\langle \ell,u \rangle} \neq 0$ (e.g., Part (D) of Fig. 1), we present an intermediary lemma (Lemma 1) which defines the *maximal number of intervals without restart* containing R maximal occurrences of σ in a time series X achieving the upper bound on the sum of its variables.

Lemma 1. *Consider a regular expression σ, a time series $X = \langle X_1, \ldots, X_n \rangle$ with every X_i ranging over the same integer interval domain $[\ell, u]$, a $\text{NB\_}\sigma(X, R)$ constraint with $R \geq 0$. When $o_\sigma^{\langle \ell,u \rangle} \neq 0$, $\delta_\sigma^{\langle \ell,u \rangle} \neq 0$ and Prop. 2 holds, the maximal number of intervals without restart, denoted by p, is defined by*

$$p = \min\left(R, \left\lfloor \frac{n - R \cdot (\omega_\sigma + 1 - o_\sigma^{\langle \ell,u \rangle})}{o_\sigma^{\langle \ell,u \rangle}} \right\rfloor\right). \tag{1}$$

Proof. When $o_\sigma^{\langle \ell,u \rangle} \neq 0$ and $\delta_\sigma^{\langle \ell,u \rangle} \neq 0$ the R σ-patterns might be contained in one or more intervals without restart. Since each interval without restart contains at least one σ-pattern, p cannot exceed R. Wlog assume that we have only one σ-pattern in the first $p-1$ intervals without restart and $R - p + 1$ in the last one; we remark that moving one σ-pattern from an interval without restart containing more than one σ-pattern to another interval without restart, does not change the overall number of time-series variables belonging to the R σ-pattern occurrences. By Prop. 2 we use the only inducing word of σ, hence:

- In the first $p-1$ intervals without restart the total number of time-series variables used is $(p-1) \cdot (\omega_\sigma + 1)$.
- In the last interval without restart the total number of time-series variables used is $(R - p + 1) \cdot (\omega_\sigma + 1) - (R - p) \cdot o_\sigma^{\langle \ell,u \rangle}$.

Since the total number of time-series variables used by the R σ-patterns must be lower than or equal to n we have:

$$(p-1) \cdot (\omega_\sigma + 1) + (R - p + 1) \cdot (\omega_\sigma + 1) - (R - p) \cdot o_\sigma^{\langle \ell,u \rangle} \leq n.$$

By isolating p, and since p is an integer, we obtain $p \leq \left\lfloor \frac{n - R \cdot (\omega_\sigma + 1 - o_\sigma^{\langle \ell,u \rangle})}{o_\sigma^{\langle \ell,u \rangle}} \right\rfloor$, which is thus the second term inside the min term in Equation (1). □

Lemma 2. *Consider a regular expression σ that has Prop. 1, Prop. 2 and Prop. 3. Then for any integer number $n \geq 2$ and given number of occurrences of σ $R \geq 0$, there exists a word z with an associated ground time series t of length n over $[\ell, u]$ achieving the upper bound on the sum of the X_i time-series variables.*

Proof. We first construct a word z composed by the concatenation of two words, a prefix, denoted by $\overrightarrow{z}$, containing R maximal occurrences of σ, and a suffix,

denoted by $\overleftarrow{z}$, containing zero occurrences of σ. Second, we prove that there exists a supporting time series wrt $[\ell, u]$ with signature z that maximises the sum of the time-series variables.

Part A: construction of the word z. When building word z, if $o_\sigma^{\langle \ell,u \rangle} = 0$, each pair of consecutive σ-patterns does not share any time-series variables. Hence, each extended σ-pattern belongs to a different interval without restart and $p = R$. If $o_\sigma^{\langle \ell,u \rangle} \neq 0$ and $\delta_\sigma^{\langle \ell,u \rangle} = 0$, all pairs of consecutive extended σ-patterns share $o_\sigma^{\langle \ell,u \rangle}$ time-series variables. Hence, time series t has a single interval without restart that contains all σ-patterns and $p = 1$. By Lemma 1, if $o_\sigma^{\langle \ell,u \rangle} \neq 0$ and $\delta_\sigma^{\langle \ell,u \rangle} \neq 0$, all σ-pattern occurrences of time series t are contained in $p \geq 1$ intervals without restart. There exists R words of $\mathcal{L}_\sigma$, a prefix word $\overrightarrow{z}$ including the R words, and a concatenation of $\overrightarrow{z}$ with a suffix word $\overleftarrow{z}$ such that all the conditions of Prop. 1, Prop. 2 and Prop. 3 are satisfied. We construct the signature of the time series, denoted by z, by first building the signature z_k (with $k \in [1, p]$) of every interval without restart of t by imposing the following conditions:

- [**Structure of each interval without restart**] Each word z_k (with $k \in [1, p]$) has c_k occurrences of σ and is defined by

$$\begin{cases} z_k = v^{c_k}, & c_k = 1, & \text{if } o_\sigma^{\langle \ell,u \rangle} = 0 \\ z_k = v^{c_k}, & c_k = R, & \text{if } o_\sigma^{\langle \ell,u \rangle} \neq 0 \text{ and } \delta_\sigma^{\langle \ell,u \rangle} = 0 \\ z_k = vw^{c_k - 1}, & \begin{cases} c_k = s+1, & \text{if } k \leq p' \\ c_k = s, & \text{otherwise} \end{cases} & \text{otherwise} \end{cases} \tag{2}$$

  where $v \in \Theta_\sigma$, v^k denotes the concatenation of k occurrences of v, vw is the superposition between v and v, $s = \lfloor \frac{R}{\max(1,p)} \rfloor$, and $p' = R \bmod \max(1, p)$.

- [**Combining the intervals without restart: structure of $\overrightarrow{z}$**] Word $\overrightarrow{z}$ is defined by

$$\begin{cases} w^{R-1}v, & \text{if } o_\sigma^{\langle \ell,u \rangle} = 0 \\ z_1, & \text{if } o_\sigma^{\langle \ell,u \rangle} \neq 0 \text{ and } \delta_\sigma^{\langle \ell,u \rangle} = 0 \\ z_1 \text{`} < \text{'} \ldots \text{`} < \text{'} z_p, & \text{if } o_\sigma^{\langle \ell,u \rangle} \neq 0 \text{ and } \delta_\sigma^{\langle \ell,u \rangle} > 0 \\ z_1 \text{`} > \text{'} \ldots \text{`} > \text{'} z_p, & \text{if } o_\sigma^{\langle \ell,u \rangle} \neq 0 \text{ and } \delta_\sigma^{\langle \ell,u \rangle} < 0 \end{cases} \tag{3}$$

  where $v \in \Theta_\sigma$, and word w belonging to $\{\text{`}v > \text{'}, \text{`}v = \text{'}, \text{`}v < \text{'}\}$ is not a proper factor of any word in $\mathcal{L}_\sigma$ and its height is η_σ.

- [**Completing the set of intervals without restart: structure of $\overleftarrow{z}$**] Word $\overleftarrow{z}$ with length m is defined by

$$\begin{cases} \varepsilon, & \text{if } m = 0 \\ \text{`} <=^* \text{'}, & \text{if } m > 0 \text{ and `} > (=|>)^* \text{'} \text{ is a suffix of } v \\ \text{`} =^+ \text{'}, & \text{otherwise} \end{cases} \tag{4}$$

Part B: proving that there is a ground time series t with signature z that maximises the sum of the time-series variables. Since we assume that regular expression σ has Prop. 1, time-series variables in $\overleftarrow{z}$ can be assigned to the maximal domain value u without creating a new occurrence of pattern. Hence, to prove the maximality on the sum of the X_i variables belonging to t, it suffices to show that there exists a ground time series over $[\ell, u]$ obtained with the signature of word $\overrightarrow{z}$ achieving the upper bound on the sum of its variables. For space reasons we only consider the case where $\delta_\sigma^{\langle \ell,u \rangle} \neq 0$. We define two ground time series t^* and t' such that their signatures contain R σ-pattern occurrences and p intervals without restart:

- t^* corresponds to the ground time series with signature $\overrightarrow{z}$ satisfying Equation (2) and where the first σ-pattern occurrence of each interval without restart is at level 0, i.e. the level closest to the maximal domain value u.
- t' corresponds to any other ground time series where the number of σ-patterns located at level 0 is strictly less than p.

To obtain the total weight of a ground time series, i.e. the upper bound on the sum of the time-series variables, we first define the maximum weight of a σ-pattern located at level e by $\underbrace{\lambda_\sigma^{\langle u \rangle}}_{\text{A}} - \underbrace{(\omega_\sigma + 1)\cdot \mid \delta_\sigma^{\langle \ell,u \rangle} \mid \cdot e}_{\text{B}}$, and the weight of the overlap between two consecutive σ-patterns located at levels e and $e+1$ by $\underbrace{\alpha_\sigma^{\langle u \rangle}}_{\text{C}} - \underbrace{o_\sigma^{\langle \ell,u \rangle} \cdot \mid \delta_\sigma^{\langle \ell,u \rangle} \mid \cdot e}_{\text{D}}$. Terms A and C, defined in Sec. 3.1, correspond to the maximum weight of a σ-pattern and to the total weight of the overlap between two consecutive σ-patterns, respectively. B and D are two correction terms which respectively adjust the weight of a σ-pattern and the weight of the overlap between two consecutive σ-patterns, caused by a change in the level of a σ-pattern occurrence.

The total weight of a ground time series t, denoted by W_t, is the sum of the weights of the R σ-patterns minus the sum of the weights of the $R-p$ overlaps between consecutive pairs of σ-patterns. Hence, W_t is defined by

$$W_t = \Big(R\cdot\lambda_\sigma^{\langle u \rangle} - (\omega_\sigma+1)\underbrace{\sum_{k=1}^{p}\sum_{e=i_k}^{j_k}\Delta_e}_{B_T}\Big) - \Big((R-p)\cdot\alpha_\sigma^{\langle u \rangle} - o_\sigma^{\langle \ell,u \rangle}\underbrace{\sum_{k=1}^{p}\sum_{e=i_k}^{j_k-1}\Delta_e}_{D_T}\Big), \quad (5)$$

where $\Delta_e = \mid \delta_\sigma^{\langle \ell,u \rangle} \mid \cdot e$, and i_k, j_k are the highest and the lowest levels of the σ-patterns in interval without restart $k \in [1,p]$, respectively. Note that in Equation (5), the only terms that depend on the level of the σ-pattern occurrences are the correction terms B_T and D_T. Let $i_k = 0$ and $j_k = c_k - 1$ be the levels of the highest and the lowest σ-pattern occurrence in interval $k \in [1,p]$ for t^*. For t' we assume that at least one $i_k > 0$ with $k \in [1,p]$. Therefore, we compare the terms B_T and D_T for t^* and t' in the following way:

$$\sum_{k=1}^{p}\sum_{e=i_k}^{j_k}\Delta_e \geq \mid \delta_\sigma^{\langle \ell,u\rangle} \mid + \sum_{k=1}^{p}\sum_{e=0}^{c_k-1}\Delta_e \quad (6) \qquad \sum_{k=1}^{p}\sum_{e=i_k}^{j_k-1}\Delta_e \geq \mid \delta_\sigma^{\langle \ell,u\rangle} \mid + \sum_{k=1}^{p}\sum_{e=0}^{c_k-2}\Delta_e \quad (7)$$

Our objective is to show that $W_{t^*} > W_{t'}$, i.e. the total weight of t^* is strictly greater than the total weight of t'. Hence, by using Equation (5) to define W_{t^*} and $W_{t'}$, by including Inequalities (6) and (7) in $W_{t'}$ and by factorising, we have

$$o_\sigma^{\langle \ell,u\rangle}\sum_{k=1}^{p}\sum_{e=0}^{c_k-2}\Delta_e - (\omega_\sigma+1)\sum_{k=1}^{p}\sum_{e=0}^{c_k-1}\Delta_e \; > \tag{8}$$
$$o_\sigma^{\langle \ell,u\rangle}\left(\mid \delta_\sigma^{\langle \ell,u\rangle} \mid + \sum_{k=1}^{p}\sum_{e=0}^{c_k-2}\Delta_e\right) - (\omega_\sigma+1)\left(\mid \delta_\sigma^{\langle \ell,u\rangle} \mid + \sum_{k=1}^{p}\sum_{e=0}^{c_k-1}\Delta_e\right)$$

By factorising Inequality (8), we have

$$\omega_\sigma + 1 > o_\sigma^{\langle \ell,u\rangle} \tag{9}$$

Since the size of σ is always greater than or equal to the overlap of σ, i.e. $\omega_\sigma \geq o_\sigma^{\langle \ell,u\rangle}$, Inequality (9) holds and $W_{t^*} > W_{t'}$. □

Upper bound on the sum of the time-series variables. Consider a NB_$\sigma(X,R)$ family of time-series constraints with every X_i ranging over the same interval $[\ell,u]$. Theorem 1 provides an upper bound on the sum of the time-series variables wrt the time-series constraint.

Theorem 1. *Consider a regular expression σ satisfying the conditions of Prop. 1, Prop. 2 and Prop. 3. The upper bound on the sum of the time-series variables for the* NB_$\sigma(X,R)$ *family is defined by*

$$\sum_{k=1}^{p}\sum_{e=0}^{c_k-1}(\lambda_\sigma^{\langle \ell,u\rangle} - (\omega_\sigma+1)\cdot\Delta_e) - \sum_{k=1}^{p}\sum_{e=0}^{c_k-2}(\alpha_\sigma^{\langle \ell,u\rangle} - o_\sigma^{\langle \ell,u\rangle}\cdot\Delta_e) + m\cdot u, \tag{10}$$

where m is defined by: $m = n - \left\lceil \sum_{k=1}^{p}\sum_{e=0}^{c_k-1}(\omega_\sigma+1) - \sum_{k=1}^{p}\sum_{e=0}^{c_k-2} o_\sigma^{\langle \ell,u\rangle} \right\rceil$.

Proof. It uses the construction of the proof of Lemma 2. □

This upper bound is valid for all 22 regular expressions of [4], except for `Inflexion`, `Zigzag`, `Steady` and `SteadySequence`, since the first two regular expressions do not satisfy the condition in Prop. 2 and the last two regular expressions do not satisfy the condition in Prop. 1.

3.3 Time-Series Constraints with Feature WIDTH

For patterns σ satisfying Prop. 1 and Prop. 2 we sketch a method to derive bounds on the sum of the time-series variables for the SUM_WIDTH_$\sigma(X,R)$ family, provided all X_i (with $i \in [1,n]$) variables are in an interval $[\ell,u]$. To build a time series t whose sum of variables is maximum, while having R as the sum of the widths of the occurrences of the pattern σ, we use a two-step procedure.

– [STEP 1: NORMALISING THE PATTERN OCCURRENCES] For each σ pattern, we define a transformation $\mathcal{T}_\sigma$ whose repeated application from any initial signature $S_{initial}$ leads to a target signature S_{target}. $S_{initial}$ and S_{target} have the same value for R, and no matter the value of $S_{initial}$, this signature will converge to a signature S_{target} with the same number of σ-pattern occurrences. A single application of $\mathcal{T}_\sigma$ from a signature S to a signature S' has the following properties:
  i. S and S' share the same sum of the widths for their σ patterns.
  ii. The largest sum of the X_i variables compatible with S is less than or equal to the largest sum of the X_i variables compatible with S'.

  To find the time series with the largest sum of the X_i variables compatible with signature S we first perform generalised arc consistent (GAC) in the induced constraint satisfaction problem. Second, we fix all X_i variables to their respective maximal value. Note that for a binary constraint of the type $<$, $=$ or $>$, we can always set its two variables to their respective maximal values, while satisfying the constraint in question.
– [STEP 2: NORMALISATION OUTSIDE THE PATTERN OCCURRENCES] We modify S_{target} to S_{final} so that all X_i variables that do not belong to an extended σ-pattern of S_{final} can be set to their maximum value u.

We define two transformations, denoted by $\mathcal{T}^1_\sigma$ and $\mathcal{T}^2_\sigma$. For space reasons, we sketch the two transformations but we only illustrate $\mathcal{T}^1_\sigma$ in Ex. 5.

– $\mathcal{T}^1_\sigma$ transforms $S_{initial}$ into a sequence S_{target} containing the smallest possible words in $\mathcal{L}_\sigma$, i.e. inducing words whose widths are equal to $\pi_\sigma = \omega_\sigma + 1 - a_\sigma - b_\sigma$. $\mathcal{T}^1_\sigma$ works for $\sigma =$ `DecreasingSequence`, `IncreasingSequence`, `StrictlyIncreasingSequence`, and `StrictlyDecreasingSequence`, and for `Gorge` and `Summit` when $n \geq p_\sigma^{\langle R \rangle} \cdot (\omega_\sigma + 1) - (p_\sigma^{\langle R \rangle} - 1) \cdot o_\sigma^{\langle \ell, u \rangle}$, i.e. there is enough space to create $p_\sigma^{\langle R \rangle} = \lfloor \frac{R}{\pi_\sigma} \rfloor$ inducing words of σ. The upper bound on the sum of X_i variables when $\mathcal{T}^1_\sigma$ is used is

$$\underbrace{p_\sigma^{\langle R \rangle} \cdot \lambda_\sigma^{\langle \ell, u \rangle} - (p_\sigma^{\langle R \rangle} - 1) \cdot \alpha_\sigma^{\langle \ell, u \rangle}}_{\text{I}} + \underbrace{(R \bmod \pi_\sigma) \cdot (u - (\eta_\sigma + 1))}_{\text{II}} + \underbrace{m \cdot u}_{\text{III}},$$

where $m = n - (p_\sigma^{\langle R \rangle} \cdot (\omega_\sigma + 1) - (p_\sigma^{\langle R \rangle} - 1) \cdot o_\sigma^{\langle \ell, u \rangle} + R \bmod \pi_\sigma)$. Term I corresponds to the maximum weight of the concatenation of $p_\sigma^{\langle R \rangle}$ occurrences of the only inducing word of σ. Term II is related to a correction term which is used when it is not possible to obtain a sum of the widths equal to R with $p_\sigma^{\langle R \rangle}$ inducing words. Term III corresponds to the maximum weight of the variables that do not belong to any σ-pattern occurrence. In Part (C) of Fig. 2 points ●, ◎ and ○ respectively contribute to terms I, II and III.
– $\mathcal{T}^2_\sigma$ transforms $S_{initial}$ into a sequence S_{target} containing one occurrence of σ. $\mathcal{T}^2_\sigma$ works for 10 other σ-pattern including `IncreasingTerrace` and `Peak`. The upper bound on the sum of X_i variables when $\mathcal{T}^2_\sigma$ is used is $\lambda_\sigma^{\langle R, u \rangle} + m \cdot u$,

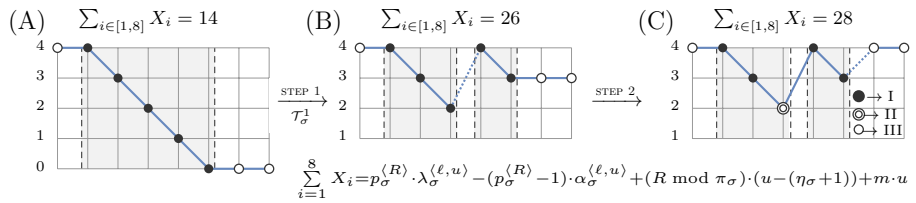


Fig. 2. Transforming an initial time series to a final time series that maximises the sum of the X_i variables, where both time series share the same value, i.e. $R = 5$, for the sum of the widths of the strictly decreasing sequences

where $m = n-(R+a_\sigma+b_\sigma)$, and $\lambda_\sigma^{\langle R,u\rangle}$ is the maximum weight of the regular expression σ where words in $\mathcal{L}_\sigma$ have a fixed length of $R + a_\sigma + b_\sigma - 1$.

Example 5. Fig. 2 gives an example of how to build a time series that maximises the sum of its variables while reaching a given sum of the widths of the pattern occurrences. The constraint used is SUM_WIDTH_$\sigma(\langle X_1, \ldots, X_8\rangle, 5)$ with $\sigma =$ ' $>^+$ ', $a_\sigma = b_\sigma = 0$, and $X_i \in [0, 4]$. Part (A) shows an initial time series with the largest sum of the X_i variables compatible with signature $S_{initial} = \langle =, >, >, >, >, =, =\rangle$. Part (B) presents a time series with the largest sum of the X_i variables compatible with $S_{target} = \langle =, >, >, <, >, =, =\rangle$. S_{target} is obtained after applying $\mathcal{T}_\sigma^1$ to $S_{initial}$ by changing the fourth signature variable from ' $>$ ' to ' $<$ '. Note that $S_{initial}$ and S_{target} share the same value for R and that the largest sum of the X_i variables compatible with $S_{initial}$ is less than the largest sum of the X_i variables compatible with S_{target}. Part (C) shows a final time series with the largest sum of the X_i variables compatible with $S_{final} = \langle =, >, >, <, >, <, =\rangle$, which is obtained by applying STEP 2 to S_{target}, i.e. by changing the sixth signature variable from ' $=$ ' to ' $<$ '. This allows one to obtain a larger value for the sum of the X_i, i.e. 28 instead of 26. △

4 Evaluation

As a test for our procedure, we run all time-series constraints from the NB_σ and the SUM_WIDTH_σ families for synthetic time series with length between 5 and 60, and for all possible result values (in all 45,835 runs), and find a single optimal solution minimising the sum of the time-series variables. The individual constraints use a state-of-the-art implementation, combining optimised automata [5], bounds on the result variables [8], glue-matrix constraints linking all prefixes and suffixes [5], and selected redundant linear constraints based on Farkas lemma [17]. For the variable assignment, we compare four search methods shown below, while using the bounds obtained for cost variables.

Search This is the default search in SICStus, the variables are assigned in natural order, enumerating the values from the smallest to the largest.

Custom This implements a custom search routine based on assigning the signature variables first. The same method is used for all test cases.

Search Impose This uses the default search in SICStus, but first assigns the cost variable to its smallest value. As the bounds are sharp, the first solution found is optimal.

Custom Impose We use the custom search method, but also initially impose the lower bound of our method for each constraint.

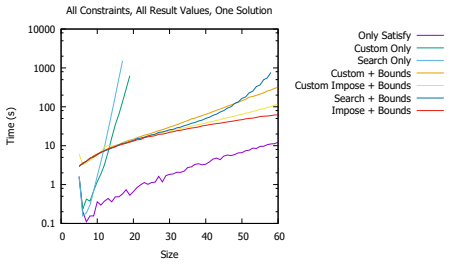


Fig. 3: Comparing baseline solutions with different search strategies

Since the conjunction of arithmetic constraints encoding bounds can propagate poorly, which results in some poor performance in the context of optimisation, we do not propagate the bounds directly; we rather use a table constraint to link the cost and result variables with a pre-computed table of all possible pairs. We compare against three baseline solutions. The first one (Only Satisfy) finds a single feasible solution, the second one (Search Only) solves the optimisation problem without bounds on the cost and with the default search routine, the third one (Custom Only) uses the custom search, again without the bounds on the cost variable. All experiments were run with SICStus Prolog 4.3.5 on a single core of a Windows 10 laptop with an Intel Core i7 CPU running at 2.9 GHz and with 64 Gb of memory. We stop the search if, for a given size, the time to run all its instances exceeds 600s, or if we reach size 60.

As we observed that both families NB_σ and SUM_WIDTH_σ behave similarly in our benchmarks, the results are shown in Fig. 3, using a log scale for the y axis. We see that without the new bounds on the cost even a custom search routine does not find solutions for all cases if the size exceeds 18. Adding the bounds significantly increases the size of the problem one can handle. The custom search outperforms the default search for larger sizes, and further improvements are possible if we impose the lower bound before starting the search on the time-series variables. The best search combines imposing the lower bound with the default search, which seems to impose only a very limited overhead compared to the Only Satisfy base line, which only finds feasible solutions.

5 Conclusion

On the one hand, the theoretical contribution of this paper consists of *parameterised sharp bounds* on the sum of the time-series variables for two families of time-series constraints. Future work may look how to extend this work to any linear cost function, e.g. linear functions where all coefficients are not set to one.

On the other hand, the practical insight of this paper is related to the importance of encoding all *arithmetic constraints* representing a bound as a table constraint in order to get all the benefits from the bounds. An interesting avenue for future research is related to the derivation of bounds for the conjunction of time-series constraints.

References

1. Abbas, H., Rodionova, A., Bartocci, E., Smolka, S.A., Grosu, R.: Quantitative regular expressions for arrhythmia detection algorithms. In: Feret, J., Koeppl, H. (eds.) Computational Methods in Systems Biology - 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27-29, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10545, pp. 23–39. Springer (2017)
2. Alur, R., Fisman, D., Raghothaman, M.: Regular programming for quantitative properties of data streams. In: Thiemann, P. (ed.) Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9632, pp. 15–40. Springer (2016)
3. Arafailova, E., Beldiceanu, N., Carlsson, M., Flener, P., Rodríguez, M.A.F., Pearson, J., Simonis, H.: Systematic derivation of bounds and glue constraints for time-series constraints. In: International Conference on Principles and Practice of Constraint Programming. pp. 13–29. Springer (2016)
4. Arafailova, E., Beldiceanu, N., Douence, R., Carlsson, M., Flener, P., Rodríguez, M.A.F., Pearson, J., Simonis, H.: Global constraint catalog, volume II, time-series constraints. arXiv preprint arXiv:1609.08925 (2016)
5. Arafailova, E., Beldiceanu, N., Douence, R., Flener, P., Rodríguez, M.A.F., Pearson, J., Simonis, H.: Time-series constraints: Improvements and application in cp and mip contexts. In: International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems. pp. 18–34. Springer (2016)
6. Arafailova, E., Beldiceanu, N., Simonis, H.: Among implied constraints for two families of time-series constraints. In: International Conference on Principles and Practice of Constraint Programming. pp. 38–54. Springer (2017)
7. Arafailova, E., Beldiceanu, N., Simonis, H.: Generating linear invariants for a conjunction of automata constraints. In: International Conference on Principles and Practice of Constraint Programming. pp. 21–37. Springer (2017)
8. Arafailova, E., Beldiceanu, N., Simonis, H.: Deriving generic bounds for time-series constraints based on regular expressions characteristics. Constraints **23**(1), 44–86 (2018)
9. Beldiceanu, N., Carlsson, M., Douence, R., Simonis, H.: Using finite transducers for describing and synthesising structural time-series constraints. Constraints **21**(1), 22–40 (2016)
10. Beldiceanu, N., Feris, B.D., Gravey, P., Hasan, S., Jard, C., Ledoux, T., Li, Y., Lime, D., Madi-Wamba, G., Menaud, J.M., et al.: Towards energy-proportional clouds partially powered by renewable energy. Computing **99**(1), 3–22 (2017)
11. Beldiceanu, N., Ifrim, G., Lenoir, A., Simonis, H.: Describing and generating solutions for the edf unit commitment problem with the modelseeker. In: International Conference on Principles and Practice of Constraint Programming. pp. 733–748. Springer (2013)
12. Eeckhout, L., De Bosschere, K., Neefs, H.: Performance analysis through synthetic trace generation. In: 2000 IEEE International Symposium on Performance Analysis of Systems and Software. ISPASS (Cat. No. 00EX422). pp. 1–6. IEEE (2000)
13. Goldin, D.Q., Kanellakis, P.C.: On similarity queries for time-series data: constraint specification and implementation. In: International Conference on Principles and Practice of Constraint Programming. pp. 137–153. Springer (1995)

14. Kegel, L., Hahmann, M., Lehner, W.: Template-based time series generation with loom. In: EDBT/ICDT Workshops. vol. 1558. Citeseer (2016)
15. Lin, J., Williamson, S., Borne, K.D., De Barr, D.: Pattern recognition in time series. In: Way, M.J., Scargle, J.D., Ali, K.M., N, S.A. (eds.) Advances in Machine Learning and Data Mining for Astronomy. CRC (2016)
16. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to Time Series Analysis and Forecasting. Willey, 2nd edn. (2016)
17. Rodríguez, M.A.F., Flener, P., Pearson, J.: Implied constraints for automaton constraints. In: Gottlob, G., Sutcliffe, G., Voronkov, A. (eds.) Global Conference on Artificial Intelligence, GCAI 2015, Tbilisi, Georgia, October 16-19, 2015. EPiC Series in Computing, vol. 36, pp. 113–126. EasyChair (2015)
18. Rodríguez, M.A.F., Flener, P., Pearson, J.: Automatic generation of descriptions of time-series constraints. In: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 102–109. IEEE (2017)
19. Shokoohi-Yekta, M., Chen, Y., Campana, B.J.L., Hu, B., Zakaria, J., Keogh, E.J.: Discovery of meaningful rules in time series. In: Cao, L., Zhang, C., Joachims, T., Webb, G.I., Margineantu, D.D., Williams, G. (eds.) Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015. pp. 1085–1094. ACM (2015)