Minimum-Width Confidence Bands via Constraint Optimization*

Jeremias Berg[1], Emilia Oikarinen[2], Matti Järvisalo[1], and Kai Puolamäki[2]

[1] HIIT, Department of Computer Science, University of Helsinki, Finland
[2] Finnish Institute of Occupational Health, Helsinki, Finland

Abstract. The use of constraint optimization has recently proven to be a successful approach to providing solutions to various NP-hard search and optimization problems in data analysis. In this work we extend the use of constraint optimization systems further within data analysis to a central problem arising from the analysis of multivariate data, namely, determining minimum-width multivariate confidence intervals, i.e., the minimum-width confidence band problem (MWCB). Pointing out drawbacks in recently proposed formalizations of variants of MWCB, we propose a new problem formalization which generalizes the earlier formulations and allows for circumvention of their drawbacks. We present two constraint models for the new problem in terms of mixed integer programming and maximum satisfiability, as well as a greedy approach. Furthermore, we empirically evaluate the scalability of the constraint optimization approaches and solution quality compared to the greedy approach on real-world datasets.

1 Introduction

The use of constraint programming systems has recently proven to be a successful approach to providing solutions to various NP-hard search and optimization problems in data mining and machine learning, using a variety of constraint optimization paradigms such as constraint programming (CP), mixed integer programming (MIP), Boolean satisfiability (SAT), maximum satisfiability (MaxSAT), and answer set programming (ASP). Compared to the more typical in-exact, problem-specific local search style algorithms, the benefits of constraint reasoning and optimization lie on one hand in the ability to provide *provably optimal solutions*, translating into more accurate solutions to the data analysis task at hand, and on the other hand by generality of algorithmic solutions resulting from the declarative approach, allowing for capturing different problem variants simply by enforcing additional or slightly modified constraints.

In this work we extend the use of constraint optimization systems further within data analysis to a central problem arising from the analysis of multivariate data, namely, determining minimum-width multivariate confidence intervals. Confidence intervals are commonly used to summarize distributions over reals, e.g., to denote ranges of data, to specify accuracies of estimates of parameters, or in Bayesian settings to describe

* This work was financially supported by Academy of Finland (grants 251170 COIN, 276412, 284591, 288814); Tekes (Revolution of Knowledge Work); and DoCS Doctoral School in Computer Science and Research Funds of the University of Helsinki.

the posterior distribution. Represented with an upper and a lower bound, confidence intervals are also easy to interpret together with the data. In contrast to p-values, which only convey information about statistical significance—a problem which has been long and acutely recognized in many disciplines [4, 20, 23, 25]—confidence intervals give information on both the statistical significance of the result as well as the effect size. In fact, since statistically significant results can be meaningless in practice due to the small effect size, the problem with p-values has been long recognized [23, 25]. The proposed solution is not to report p-values at all, but use confidence intervals instead [20]. Optimizing the width of multivariate confidence intervals is NP-hard, and furthermore, expectedly even hard to approximate [11], which motivates the use of constraint optimization systems for the task.

The problem of estimating the confidence interval of a distribution based on a finite-sized sample from the distribution has been extensively studied, see e.g. [7]. However, most effort has focused on describing a single univariate distribution over real numbers, and there are surprisingly few approaches to multivariate confidence intervals. In the time series domain, multivariate confidence intervals [9, 11], namely *confidence bands*, have been defined in terms of the *minimum-width envelope* (MWE). The only exact approach to MWE (in this paper denoted by MWCB(k) as motivated later on) that we are aware of is the very recent integer programming model of [21]. However, as explained in [10], solving MWE can result in very conservative confidence bands when there are local deviations from what constitutes as normal behaviour in the data at hand. To overcome this, an alternative definition as what we will refer to as MWCB(k, s) was recently introduced in [10], and a greedy approach to solving this variant was provided; to the best of our understanding no exact algorithms for MWCB(k, s) have been proposed. However, as we will shortly explain, MWCB(k, s) can result in optimal solutions exhibiting extremely narrow parts in the confidence band even when there is no clear explanation for this behaviour in the data.

In this paper, we focus on the combinatorial variant of the minimum-width confidence band problem, and specifically, on constraint optimization approaches to obtaining optimal solutions to a new variant MWCB(k, s, t) of the multivariate minimum-width confidence interval problem. In more detail, *our contributions* are the following. (i) We demonstrate that minimum-width (k, s)-confidence bands as defined in [10] tend to have very narrow parts without a clear intuitive meaning. (ii) We propose an alternative definition to overcome this undesirable property, denoted as the MWCB(k, s, t) problem. (iii) As a novel application domain of declarative constraint optimization, we provide two constraint models for MWCB(k, s, t) in terms of MIP and MaxSAT as the constraint languages of choice at this time. (iv) We also provide a greedy algorithm for MWCB(k, s, t), which provides more scalability, but also allows for analyzing the benefits of exact constraint optimization for the problem in terms of the quality of obtained solutions. (v) To this end, we present an overview of an empirical evaluation on the scalability of the constraint optimization approaches and solution quality.

The rest of this paper is organized as follows. We start with the previously proposed variants MWCB(k) and MWCB(k, s) of the multivariate confidence interval problem, pointing out their drawbacks, and motivated by these we propose the focus of this work, the MWCB(k, s, t) problem (Section 2). We then introduce constraint optimiza-

tion models for MWCB(k, s, t) in terms of MIP and MaxSAT (Section 3), as well as a first greedy approach to MWCB(k, s, t) for comparing with the constraint models (Section 4). Overview of an empirical evaluation of the constraint models is presented in Section 5 using real-world time series datasets, and related work discussed further in Section 6.

2 The Minimum-Width Confidence Band Problem

We consider a set of n data vectors x_i, each of length m, represented by a matrix $X \in \mathbb{R}^{n \times m}$. Let $x_{ij} \in X$ denote the j-th element of x_i. The data X can, for instance, represent time series data, i.e., each x_i is a sequence of values taken at successive points in time. (In the general setting, preprocessing may be necessary as one needs to make sure that the variables or at least their scales are comparable.)

A *confidence band* is defined as a pair (l, u) of vectors, where $l, u \in \mathbb{R}^m$ and $l_j \leq u_j$ for all j. The size of the confidence band $CB = (l, u)$ is $\text{SIZE}(CB) = \sum_{j=1}^{m}(u_j - l_j)$. In order to capture the relationship between a confidence band and a dataset, we use the concept of an *error* of a confidence band w.r.t. the data at hand. An indicator function (unity if the condition $\square$ is satisfied and zero otherwise) is denoted by $I[\square]$.

Definition 1. *Given a data vector x_i and a confidence band $CB = (l, u)$, the error of x_i w.r.t. CB is the number of points in x_i that lie outside of CB, i.e.,* $\text{ERROR}(x_i, CB) = \sum_{j=1}^{m} I[x_{ij} < l_j \vee u_j < x_{ij}]$.

There are several possible ways to control the error. In [9, 11] a data vector is considered an *outlier* or *extreme* if it is outside the confidence band in at least one dimension. In the minimum-width confidence band (MWCB(k)) problem the number of extreme data vectors is controlled by a parameter k.

Definition 2 (MWCB(k)). *[9, 11] Given a dataset $X \in \mathbb{R}^{n \times m}$, any confidence band $CB^* \in \arg\min \text{SIZE}(CB)$ over those CB for which $\sum_{i=1}^{n} I[\text{ERROR}(x_i, CB) > 0] \leq k$ is a solution to the MWCB(k) problem.*

The above definition results in well-defined confidence bands which gives the user control over the error in analogy to the family-wise error rate. However, as argued in [10], the resulting confidence bands can be too conservative in cases in which there are local deviations from what constitutes as normal behaviour in the data at hand. To overcome this feature, a relaxed variant of the MWCB(k) problem allowing local deviations from the confidence band was recently proposed [10, 24].

Definition 3 (MWCB(k, s)). *[10] Given a dataset $X \in \mathbb{R}^{n \times m}$ and two integers k and s, any confidence band $CB^* \in \arg\min \text{SIZE}(CB)$ over those CB for which $\sum_{i=1}^{n} I[\text{ERROR}(x_i, CB) > s] \leq k$ is a solution to the MWCB(k, s) problem.*

Now, one may observe that a solution to the MWCB(k, s) problem can be very narrow at places. This was in fact pointed out in [10], where it was further argued that in real datasets with non-trivial marginal distributions and correlation structure this was unlikely to happen, and the confidence band would be approximately of similar

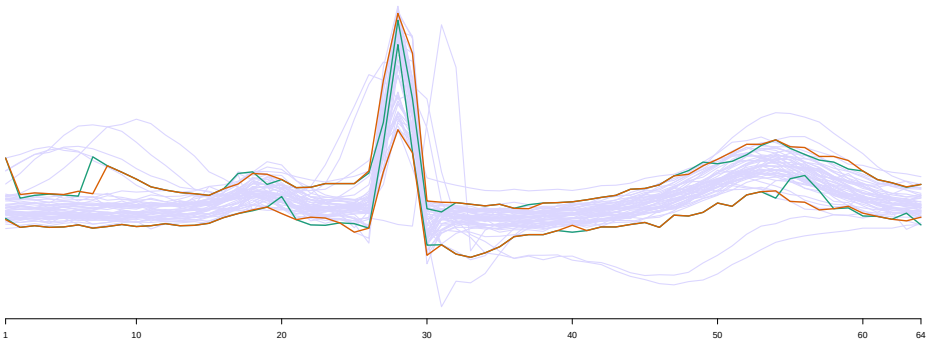

Fig. 1. An example of time series data with $n = 50$ time series of length $m = 64$ (represented with purple lines) for which an MWCB(k, s) confidence band (green lines) has very narrow parts, while the respective MWCB(k, s, t) confidence band (orange lines) does not. Here we have used $k = t = 5$ and $s = 6$, each value representing approximately 10% of the respective dimension.

width across columns. However, we note that optimal solutions to the MWCB(k, s) problem on real data are likely to contain narrow intervals with no clear explanation. For example, consider Fig. 1. The data consists of 50 time series of length 64 sampled from the MITDB data (see Section 5 for details). The green lines represent a confidence band that is a solution to MWCB(5,6) problem. We can observe that the confidence band is very narrow at the peak, i.e., around the time interval [25,30]. One should notice that we use a real data set here to demonstrate the unwanted behaviour, and obviously it is not difficult to craft synthetic instances for which an optimal solution to the MWCB(k, s) problem has extremely narrow parts in the confidence band.

These observations suggest that there should be a mechanism to control the amount of column-wise error in addition to the row-wise constraints, and to this end we propose the concept of a *minimum-width* (k, s, t)*-confidence band* in terms of the MWCB(k, s, t) problem as follows.

Definition 4 (MWCB(k, s, t)). *Given a dataset $X \in \mathbb{R}^{n \times m}$ and integers k, s, and t, any confidence band $CB^* \in \arg\min \text{SIZE}(CB)$ over those $CB = (l, u)$ for which $\sum_{i=1}^{n} I[\text{ERROR}(x_i, CB) > s] \leq k$ and $\sum_{i=1}^{n} I[x_{ij} < l_j \vee x_{ij} > u_j] \leq t$ for all $1 \leq j \leq m$, is a solution to the MWCB(k, s, t) problem*

As straightforward connection between the MWCB(k, s, t) and MWCB(k, s) problems is the following.

Proposition 1. *A confidence band CB for a dataset $X \in \mathbb{R}^{n \times m}$ is a solution to the MWCB(k, s) problem iff it is a solution to the MWCB(k, s, n) problem.*

The additional parameter t gives the user control over the amount of outliers allowed column-wise. If local deviations are likely to not to happen too often, setting the value of t equal to, or slightly larger than, k is a reasonable choice. For example, in Fig. 1, the orange lines represent a confidence band that is a solution to MWCB(5,6,5). One can observe that, indeed, for each time point a majority (90%, i.e., 45 out of 50 to be exact) of the time series are inside the confidence band. Furthermore, at most 10% of

MINIMIZE

$$\sum_{j=1}^{m}(u_j - l_j) \tag{1}$$

SUBJECT TO

$$l_j \leq x_{(t+1)j} \quad \forall j \in \mathcal{M} \tag{2}$$

$$u_j \geq x_{(r_j^{max}-t)j} \quad \forall j \in \mathcal{M} \tag{3}$$

$$l_j - M_j^l d_{ij} \leq x_{ij} \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \tag{4}$$

$$u_j + M_j^u d_{ij} \geq x_{ij} \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \tag{5}$$

$$\sum_{i=1}^{n} d_{ij} \leq t \quad \forall j \in \mathcal{M} \tag{6}$$

$$\sum_{j=1}^{m} d_{ij} - (m-s)y_i \leq s \quad \forall i \in \mathcal{N} \tag{7}$$

$$\sum_{i=1}^{n} y_i \leq k \tag{8}$$

$$y_i, d_{ij} \in \{0,1\} \quad \forall i \in \mathcal{N}, j \in \mathcal{M}$$

$$l_j, u_j \in \mathbb{R} \quad \forall j \in \mathcal{M}.$$

Fig. 2. Mixed integer programming model for MWCB(k, s, t).

the time series deviate from the confidence band in more than 6 time points. Based on experimentation, it seems that for the real datasets we consider in this work, the size of the confidence band is approximately the same for $t \in \{k, k+1, \ldots, 2k\}$ (assuming $k \ll n$), and thus a conservative choice, e.g., $t = (1+\epsilon)k$ for $\epsilon < 1$, seems to be a reasonable one.

3 Constraint Optimization Models for MWCB(k, s, t)

Recently, a MIP model for the MWCB(k) problem was proposed in [21]. However, to the best of our knowledge, no efficient exact algorithms for solving the MWCB(k, s) problem (nor the more general MWCB(k, s, t) problem) exist. Two heuristic algorithms are provided in [10], with no guarantee of solution quality. Korpela et. al [10] do provide a MIP model for the special case of one-sided confidence bands. However, this model is only used to show an approximability result and does not yield a practically efficient method, even for the special case.

In the following we present two constraint optimization models for MWCB(k, s, t), one using mixed integer programming and the other using maximum satisfiability. For notation, let $\mathcal{N} = \{1, \ldots, n\}$ and $\mathcal{M} = \{1, \ldots, m\}$. Both of our constraint models use a column ordering for the data X. Thus, we assume that we have an ordering for each of the columns using *dense-rank*[3] (as provided in R) and denote by r_j^{max} the maximum rank in column j. In the following, for a given $r \in \{1, \ldots, r_j^{max}\}$, we use (r) to denote the index i such that x_{ij} has rank r in column j.

[3] In case of ties, both elements get the same rank r and the next greatest element gets rank $r+1$.

3.1 Mixed Integer Programming Model

For our MIP model, we use the *band-wise reduction procedure* suggested in [22], similarly to [21]. However, in our model, instead of looking for whole data vector to exclude from the confidence band, we need to allow the exclusion of individual data points while maintaining both the column-wise and the row-wise constraints.

Our MIP model is presented in detail in Fig. 2. We introduce variables $l_j, u_j \in \mathbb{R}$ for each $j \in \mathcal{M}$ for the confidence band, and the objective (1) is to minimize the size of the confidence band, i.e., the sum of $(u_j - l_j)$'s over all columns j. We introduce $n \times m$ binary variables d_{ij} with the interpretation $d_{ij} = 1$ iff the jth element of x_i is outside the confidence band, i.e., $x_{ij} < l_j$ or $x_{ij} > u_j$. Furthermore, we use n binary variables y_i with the interpretation $y_i = 1$ iff x_i is outside the confidence band in at least s positions, i.e., $\sum_{j=1}^{m} I[x_{ij} < l_j \vee x_{ij} > u_j] > s$.

For the band-wise reduction procedure [22], we can make use of the following observation: since we have the column-wise constraint that at most t data points can be outside the confidence band at each column, we know that the value with rank $t+1$ has to be inside the lower band. Thus, we include the constraints (2). Respective constraints for the upper band are provided in (3). Next, the constraints (4) (resp. (5)) encode the choice that either a value x_{ij} is inside the lower (resp. upper) band or it is outside. Here we use constant vectors $M_l = (M_1^l, \ldots, M_m^l)$ and $M^u = (M_1^u, \ldots, M_m^u)$ defined as

$$M_j^l = x_{(t+1)j} - \min(x_{1j}, \ldots, x_{nj}) \text{ and}$$
$$M_j^u = \max(x_{1j}, \ldots, x_{nj}) - x_{(r_j^{max}-t)j}.$$

Now, if $d_{ij} = 1$, the constraints are de-activated, and if $d_{ij} = 0$ the constraints (4) and (5) together ensure that $l_j \leq x_{ij} \leq u_j$. Here we once more use the property that at most t values can be outside the confidence band. The constraints (6) enforce this. We use the constraints (7) to represent the relationship between d_{ij}'s and y_i. If $y_i = 0$, then at most s variables d_{ij} for each $j \in \mathcal{M}$ can have value 1. On the other hand, if $y_i = 1$, then each constraint (7) reduces to $\sum_{j=1}^{m} d_{ij} \leq m$ which is always satisfied. Finally, the constraint (8) makes sure that at most k data vectors have more that s elements outside the confidence band.

3.2 Maximum Satisfiability

Before presenting our second constraint optimization model for MWCB(k, s, t), we give a brief background on maximum satisfiability. For a more extensive review we direct the reader to [2].

For a Boolean variable x there are two literals, the positive literal x and the negative literal $\neg x$. A clause is a disjunction ($\vee$) of literals and a conjunctive normal form (CNF) formula is a conjunction ($\wedge$) of clauses. Equivalently, a clause is a set of literals and a CNF formula a set of clauses. A truth assignment τ is a function from Boolean variables to $\{0, 1\}$. A truth assignment τ satisfies a clause C ($\tau(C) = 1$) if it assigns a positive literal, $x \in C$ to 1 or a negative literal $\neg x \in C$ to 0, and else τ falsifies the clause ($\tau(C) = 0$). Assignment τ satisfies a CNF formula F if it satisfies all clauses in F. An instance of the (weighted partial) maximum satisfiability (MaxSAT) problem

HARD CLAUSES

$$\text{CNF}(\textstyle\sum_{i=1}^{n} y_i \leq k) \tag{9}$$

$$\text{CNF}((\textstyle\sum_{j=1}^{m} d_{ij} > s) \rightarrow y_i) \quad \forall i \in \mathcal{N} \tag{10}$$

$$\text{CNF}(\textstyle\sum_{i=1}^{n} d_{ij} \leq t) \quad \forall j \in \mathcal{M} \tag{11}$$

$$\text{CNF}(\textstyle\sum_{r=1}^{t+1} l_j^r = 1) \quad \forall j \in \mathcal{M} \tag{12}$$

$$\text{CNF}(\textstyle\sum_{r=r_j^{max}-t}^{r_j^{max}} u_j^r = 1) \quad \forall j \in \mathcal{M} \tag{13}$$

$$\neg d_{(r)j} \rightarrow \textstyle\bigwedge_{h=(r+1)}^{t+1} \neg l_j^h \quad \forall r \in \mathcal{R}_m, j \in \mathcal{M} \tag{14}$$

$$\neg d_{(r)j} \rightarrow \textstyle\bigwedge_{h=r_j^{max}-t}^{(r-1)} \neg u_j^h \quad \forall r \in \mathcal{R}_M^j, j \in \mathcal{M} \tag{15}$$

SOFT CLAUSES

$$(\neg l_j^r \vee \neg u_j^h) \quad \forall j \in \mathcal{M}, \forall r \in \mathcal{R}_m \quad \forall h \in \mathcal{R}_M^j \tag{16}$$

WEIGHTS

$$w((\neg l_j^r \vee \neg u_j^h)) = x_{(h)j} - x_{(r)j} \tag{17}$$

Fig. 3. The base clauses in our MaxSAT encoding for MWCB(k, s, t).

(F_h, F_s, w) consists of two CNF formulas, the set of hard clauses F_h and the set of soft clauses F_s, together with a function $w\colon F_s \rightarrow \mathbb{N}$ assigning a positive weight to all soft clauses. Any truth assignment τ that satisfies all hard clauses is a solution to the MaxSAT problem. A solution τ is optimal if it minimizes the sum of the weights of the soft clauses it falsifies, i.e., if $\sum_{C \in F_s}(1 - \tau(C))w(C) \leq \sum_{C \in F_s}(1 - \tau'(C))w(C)$ for all solutions τ'.

Our MaxSAT model makes extensive use of cardinality networks [1]. For our purposes, given a set of literals L, a literal l_B and an integer bound K, a cardinality network produces a set of clauses

$$\text{CNF}(\sum_{l \in L} l > K \rightarrow l_B)$$

that encodes the property that whenever more than K literals from the set L are assigned to 1, then so is the literal l_B. We use $\text{CNF}(\sum_{l \in L} l \leq K)$ as shorthand for the CNF formula $\text{CNF}(\sum_{l \in L} l > K \rightarrow l_B) \wedge (\neg l_B)$. Notice that the clauses in $\text{CNF}(\sum_{l \in L} l \leq K)$ together essentially enforce that at most K literals of the set L can be assigned to 1. As an important special case we also use $\text{CNF}(\sum_{l \in L} l = 1)$ as shorthand for $\text{CNF}(\sum_{l \in L} l \leq K) \wedge (\bigvee_{l \in L} l)$, enforcing that exactly one of the literals in L has to be assigned to 1.

Figure 3 gives the clauses in our MaxSAT encoding. We start be describing the intuition behind the Boolean variables used. Note that for every solution $CB = (l, u)$ to the MWCB(k, s, t) problem and every $j \in \mathcal{M}$, there exists a $r \in \{1, \ldots, t+1\}$ (resp. $r \in \{r_j^{max} - t, \ldots, r_j^{max}\}$) such that $l_j = x_{(r)j}$ (resp. $u_j = x_{(r)j}$). For each column j, we use $\mathcal{R}_m^j$ to denote the set of possible r for which $l_j = x_{(r)j}$ can hold. Since at most t points can lie outside the lower band for any $j \in \mathcal{M}$, we have $\mathcal{R}_m^j =$

$\mathcal{R}_m = \{1, \ldots, t+1\}$. Similarly we use $\mathcal{R}_M^j = \{r_j^{max} - t, \ldots, r_j^{max}\}$ to denote the set of possible indices r for which $u_j = x_{(r)j}$ can hold. We introduce variables l_j^r and u_j^h for $j \in \mathcal{M}$, $r \in \mathcal{R}_m$ and $h \in \mathcal{R}_M^j$ with the interpretation $l_j^r = 1$ (resp. $u_j^h = 1$) iff $l_j = x_{(r)j}$ (resp. $u_j = x_{(h)j}$). In addition, we use the variables d_{ij} and y_i with the same semantics as in the MIP model.

Next we describe the hard clauses enforcing these semantics. The constraints (9) enforce that at most k data vectors are outside the confidence band in more than s elements. The constraints (10) enforce the correct semantics for the y_i variables, i.e., whenever x_i lies outside the confidence band in more than s elements, the variable y_i is also set to true. Next, the constraints (11) enforce that at most t data points lie outside the confidence band in each column. The constraints (12) and (13) enforce that the value of l_j and the value of u_j is uniquely defined in each column j, i.e., exactly one of the l_j^r and u_j^h variables are true for each j. The constraints (14) enforce the correct semantics for the l_j^r variables: whenever a data point $x_{(r)j}$ is inside the lower confidence band l_j, i.e., $d_{(r)j} = 0$, then the value of l_j is at most the value of $x_{(r)j}$. In order to get shorter clauses in the final MaxSAT instance, we use instead an equivalent condition stating that whenever $d_{(r)j} = 0$, the value of l_j is not equal to $x_{(r')j}$ for any $r' \in \{r+1, \ldots, t+1\}$. The constraints (15) enforce a similar condition for the u_j^h variables. The soft clauses (16) enforce that the confidence band defined by the l_j^h and u_j^h variables is of minimum size. For a fixed column j, the clause $(\neg l_j^r \vee \neg u_j^h)$ is falsified if both l_j^r and u_j^h are true, corresponding to $l_j = x_{(r)j}$ and $u_j = x_{(h)j}$. The cost of the clause is set to be $x_{(h)j} - x_{(r)j} = u_j - l_j$, i.e., the contribution of that column to the size of the final confidence band. Notice that due to the hard clauses in the encoding, exactly one soft clause per column will be falsified.

Redundant Constraints The clauses just described are enough to guarantee soundness. However, the encoding also includes redundant clauses meant to improve performance of the MaxSAT algorithms. These are based on the fact that at most t data points can lie outside the confidence band in each column. For a fixed column j this implies that there are certain pairs of indices $r \in \mathcal{R}_m$, $h \in \mathcal{R}_M^j$ for which the variables u_j^h and l_j^r cannot both be set to true.

As an example, the variables u_j^r and l_j^{t+1} for $r = r_j^{max} - t$ cannot be set to true simultaneously, since this would require $2t$ data points, namely $x_{(1)j}, \ldots, x_{(t)j}$ and $x_{(r_j^{max})j}, \ldots, x_{(r_j^{max}-(t-1))j}$ to be outside of the confidence bands in column j. Hence the clause $(\neg u_j^r \vee \neg l_j^{t+1})$ for $r = r_j^{max} - t$ is always satisfied, making it redundant as a soft clause in our encoding. However, we can instead introduce it as a hard clause to improve propagation during search. Generalizing the above observation, for a fixed variable l_j^r we introduce the clause $(\neg l_j^r \vee \neg u_j^h)$ as hard clause instead of a soft one for all $h \in \{r_j^{max} - t, \ldots, r_j^{max} - (t - (r-1))\}$.

4 A Greedy Approach to MWCB(k, s, t)

In this section we present a greedy algorithm for finding (typically non-optimal) solutions for the MWCB(k, s, t) problem. The overall idea is to exclude individual data points greedily as long as the row-wise and column-wise constraints remain satisfied.

input : dataset $X \in \mathbb{R}^{n \times m}$, integers k,s,t
output: $CB \in \mathbb{R}^{m \times 2}$

1 **begin**
2 $R \leftarrow$ordering structure for observations in X
3 $\text{rmd}_C \leftarrow$zeros(m); $\text{rmd}_R \leftarrow$zeros(n); $\text{rmd}_{cnt} \leftarrow 0$
4 $G \leftarrow$priorityQueue(gains(R))
5 **while** $G \neq \emptyset$ **do**
6 $(val, j, b) \leftarrow$getMaximumElement(G)
7 $i \leftarrow$idx($R_j, 1, b$)
8 **if** $\text{rmd}_C(j) < t$ *and* ($\text{rmd}_R(i) \neq s$ *or* $\text{rmd}_{cnt} < k$) **then**
9 $R \leftarrow$remove(R_j, b)
10 **if** $rmd_R(i) == s$ **then**
11 rmd_{cnt}++
12 $\text{rmd}_R(i)$++
13 $\text{rmd}_C(j)$++
14 $val \leftarrow$ value($R_j, 1, b$)$-$ value($R_j, 2, b$)
15 G.add(key=val, col=j, bit=b)
16 **for** $j \in 1 : m$ **do**
17 $CB(j, :) \leftarrow$ [value($R_j, 1, 0$), value($R_j, 1, 1$)]
18 **return** CB

Algorithm 1: Greedy algorithm for MWCB(k, s, t).

The general idea is similar to the greedy algorithm proposed in [11], but instead of excluding a data vector fully, we consider excluding a single data point at a time.

The greedy algorithm is presented in pseudocode as Algorithm 1. We use an ordering structure R (line 2) that allows us $O(1)$ time access to the largest and the second to largest (resp. the smallest and the second to smallest) element in each column. The ordering structure consists of a doubly-linked list R_j for each column j, and can be initialized in $O(mn \log n)$ time. We use vectors rmd_C and rmd_R to keep track of the number of excluded values for each column and row, respectively, as well as a counter rmd_{cnt} to keep track of the number of rows for which more than s elements are excluded. Let gains(R) on line 4 be a method returning the possible gains for each of the columns, i.e., $x_{(2)j} - x_{(1)j}$ for the lower band and $x_{(r_j^{max})j} - x_{(r_j^{max}-1)j}$ for the upper band in $O(m)$ time. The values are stored in a priority queue G in $O(m)$ time.

The main part of the algorithm is the while-loop on lines 5–15. The loop is repeated at most $O(mn)$ times and each iteration takes $O(\log m)$ time. We use a Boolean $b \in \{0, 1\}$ to keep track of whether the current gain is obtained from the lower band ($b = 0$) or the upper band ($b = 1$). At each iteration the element with largest gain is used as a candidate for removal. On line 7, idx(R_j, b) returns the index of the currently highest/lowest ranked value in R_j. On line 8, we have the condition under which it is possible to exclude a value from the confidence band (realized by removing the respective element from the ordering structure R). In every case we have to maintain the column-wise constraint, i.e., check that less than t values have been excluded from the column in previous steps. Furthermore, the row-wise constraints can be satisfied in two ways. The first condition $\text{rmd}_R(i) \neq s$ summarizes two cases: if $\text{rmd}_R(i) < s$, it is always safe to exclude the candidate. On the other hand, if $\text{rmd}_R(i) > s$, then the

data vector with index i is already among the k possible outliers with more than s elements excluded, and thus the current value can be excluded as well. The remaining case $\text{rmd}_R(i) == s$ requires that no more than $k-1$ data vectors have more than s elements excluded.

The respective counters are then updated (lines 10–13). On lines 14 and 15, a new candidate gain is computed and pushed to the priority queue. Here $\text{val}(R_j, r, b)$ is the value of the lowest ($b = 0$) or the highest ($b = 1$) ranked element still present in R_j for $r = 1$, resp. the value of the second lowest/highest ranked element for $r = 2$. Finally, the confidence band to be returned is directly obtained from the ordering structure. The overall time complexity for the greedy algorithm is $O(mn \log mn)$ and memory complexity $O(mn)$.

5 Experiments

We present an overview of an empirical evaluation on the scalability of the MIP and MaxSAT models using state-of-the-art solvers on MWCB(k, s, t) instances constructed from real-world time series datasets, as well as on the relative quality of solutions provided by exact constraint optimization and the greedy approach.

For the experimental evaluation, we used the state-of-the-art commercial mixed integer programming system CPLEX version 12.7.1 from IBM [8], and the Maxsat solvers QMaxSAT [12], MSCG [19], and MaxHS [3]. The MaxSAT solvers are representatives of state-of-the-art solvers based on different types of algorithms: QMaxSAT is a so-called model-guided SAT-based solver (using a SAT solver to search for increasingly good solutions until no better solutions can be found), MSCG is a core-guided SAT-based solver (using a SAT solver to extract and rule out unsatisfiable cores of a MaxSAT instance until a satisfying assignment is found), and MaxHS is a hybrid SAT-IP solver for MaxSAT, implementing a so-called implicit hitting set approach. The experiments were run on 2.83-GHz Intel Xeon E5440 quad-core machines with 32-GB RAM and Debian GNU/Linux 8 using a per-instance timeout of 3600 seconds.

We implemented the greedy procedure (recall Section 4) in R. As the greedy procedure has much better running time scalability than the constraint solvers on finding provably-optimal solutions, here we focus on comparing the improvements in solution costs provided by the exact approaches to those provided by the greedy procedure.

5.1 Datasets

For the experiments, we obtained benchmark instances based on the following real-world time series datasets.

Milan temperature data (MILAN) We use the `max-temp-milan` dataset from [11]. The raw data is obtained from Global Historical Climatology Network (GHCN) daily dataset [17, 16] from US National Oceanic and Atmospheric Administration's National Climatic Data Center (NOAA NCDC)[4]. The preprocessed data contains average monthly maximum temperatures for a station located in Milan for the years 1763–2007, resulting in $n = 245$ time series with length $m = 12$.

[4] http://www.ncdc.noaa.gov/

Table 1. Summary of datasets and the instances generated.

Dataset	sample sizes for n	sample sizes for m
MILAN (n=245, m=12)	50, 100, 150, 200, 245	12
POWER (n=1417, m=24)	200, 400, 600, 800, 1000	24
MITDB (n=2027, m=253)	100, 150, 200, 250, 300	26, 32, 43, 64, 127, 253

UCI-Power data (POWER) The UCI-Power dataset is the individual household electric power consumption data[5] from the UCI machine learning repository [13]. It consists of hourly averages of the variable `active.power`, resulting in a dataset with $n = 1417$ time series with $m = 24$ time points.

Heartbeat data (MITDB) We use the preprocessed datasets `heartbeat-normal` and `heartbeat-pvc` from [11]. These datasets are obtained from the MIT-BIH arrhythmia database available at Physionet [5]. The data contains annotated 30-minute records of normal and abnormal heartbeats [18]. There are 1507 observations in `heartbeat-normal` and 520 observations in `heartbeat-pvc` both with $m = 253$ time points.

As a preprocessing step, we shifted the data so that all values are non-negative. To assess the scalability of our constraint models, we used the datasets to produce instances with varying dimensions. To obtain an instance with n' time series, we randomly sample n' time series from the respective dataset. For the heartbeat data, we create instances with n' observations by sampling at random $0.9n'$ time series from `heartbeat-normal` and $0.1n'$ time series from `heartbeat-pvc`. Furthermore, in order to obtain instances with $m' < 253$ while maintaining the autocorrelation structure, we take every jth time point for $j \in \{2, 4, 6, 8, 10\}$. Table 1 summarizes the datasets and the parameters of the instances sampled from the datasets.

5.2 Results

Due to the large parameter value space, we will present selected views on the results which provide interesting insights into the performance of the MIP and MaxSAT approaches for the problem, as well as quality of solutions obtained.

Scalability of the Exact Approaches We start the overview of the empirical results with the scalability of the exact approaches, i.e., CPLEX on the proposed MIP model and the three considered state-of-the-art MaxSAT solvers on the MaxSAT encoding.

Results from this comparison are provided in Figs. 4 and 5. Figure 4 (left) shows the number of instances solved by each of the four solvers under different per-instance time limits on instances based on the MILAN dataset using the parameter values $k \in \{0.01n, 0.02n, 0.05n\}$, $s \in \{1, 2\}$, and $t \in \{k, k+2\}$, rounding values below 1 to 1, and the rest to the nearest integer. To increase the number of instances, we used $k \in \{1, 2, 3\}$ for the smallest value $n = 50$. This results in 60 instances in total. Out of the three MaxSAT solvers, the model-guided QMaxSAT performs the best. However, CPLEX on the MIP model solves each of the instances very fast, surpassing in performance the MaxSAT solvers on the MaxSAT encoding.

[5] http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption

Based on this, we take a further look at the performance of CPLEX and QMaxSAT as the two most promising out of the considered solvers. Figure 4 (right) shows the number of solved instances generated from all of the three benchmark datasets (MILAN, POWER, and MITDB) using parameter values $k \in \{0.01n, 0.02n, 0.05n\}$, $t \in \{k, k+2\}$ and $s = \{0.01m, 0.02m, 0.05m\}$ (720 instances in total), and gives further support for the fact that CPLEX dominates in performance the MaxSAT approach.

One should note that the value of t has a direct impact on the size of the search space. Values $t < k$ are allowed by the definition, but can result in unintuitive solutions. Thus, we consider values $t \geq k$. As an increase in the value of t can intuitively drastically increase the hardness of an instance (in the worst case all nm values need to be considered for removal), we assessed the effect of t on the solution quality, i.e., the size of the optimal confidence bands. Experimentation with instances from the MITDB and POWER dataset with 200–400 time series showed that increasing the value of t from k to $2k$ for $k = 0.05n$ decreased the size of an optimal solution by less that 1%. Thus, to assess the scalability of our approach, we chose to use the conservative values $t \in \{k, k + 2\}$. One should note, however, that the best value for t depends on the dataset at hand and the expected distribution of local and global outliers.

Figures 5 and 6 give further insights into the scalability of the MIP approach, using instances based on the MITDB dataset with $m = 43$ and $m = 127$. As parameters values we consider $s = \{0.02m, 0.05m\}$ and $t \in \{k, k + 2\}$. For instances with $m = 43$, we use $k \in \{0.01n, 0.02n, 0.05n, 0.10n\}$, and for instances with $m = 127$ we use $k \in \{0.01n, 0.02n, 0.05n\}$.

First, we consider the effect of data dimensions to the solving time. We observe that an increase in the length of the time series m affects the solving time more than an increase in the number of time series n, i.e., the instances based on the MITDB data with $n = 250$ and $m = 127$ are easier to solve than instances with $n = 100$ and $m = 253$ (detailed results for MITDB-$m253$ not reported due to space constraints).

As for the scalability w.r.t. parameter k, typically one would be interested in 95% or 90% confidence intervals. The 95% confidence intervals correspond to setting $k = \lfloor 0.05 \rfloor$, and our MIP model can handle $k = \lfloor 0.05n \rfloor$ with instances up to $n = 300$ and $m = 127$. For the instances based on MITDB data with $m = 253$, instances with up to $n = 100$ and POWER data with $m = 24$ instances with up to $n = 600$ can be solved. In the case corresponding to 90% confidence intervals, MITDB instances with $m = 43$

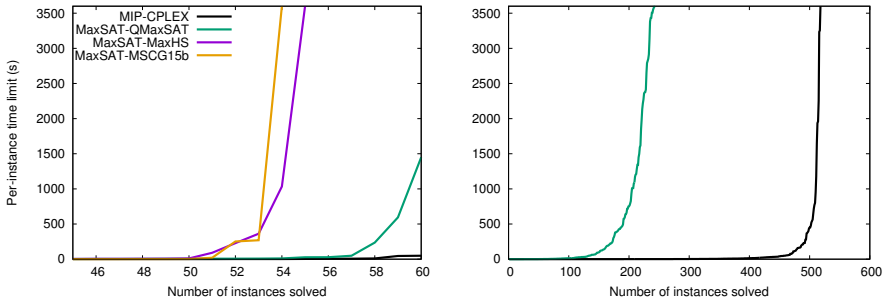

Fig. 4. Comparison of solver scalability. Left: MILAN dataset, right: all datasets.

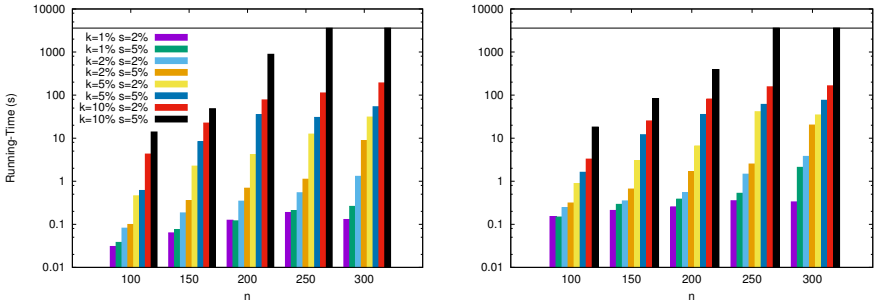

Fig. 5. The solving times for the instances sampled from the MITDB data with $m = 43$ using the MIP model. Left: $t = k$, right: $t = k + 2$.

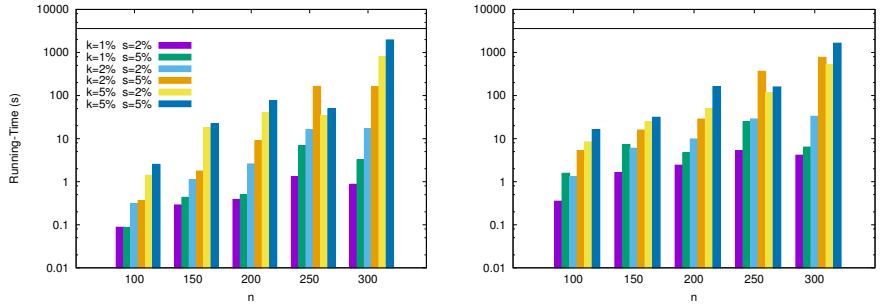

Fig. 6. The solving times for the instances sampled from the MITDB data with $m = 127$ using the MIP model. Left: $t = k$, right: $t = k + 2$.

up to $n = 200$ (with $s = 0.05m$) can be solved before timeout. In Figs. 5 and 6 the effect of parameter s on solving time seems smaller than that of parameter k. This is to some extend to due to the fact that for the actual values used, namely $1 \leq s \leq 6$, the number of possible combinations stays reasonable. For larger s, the effect becomes more visible.

Finally, as expected, typically an increase on the value of parameter t results in an increase in solving times. However, in contrast to the other parameters, the solving times do not monotonically increase upon increasing t. In fact, there are some instances with $t > t'$ for which it is faster to solve the MWCB(k, s, t) problem than the MWCB(k, s, t') problem, e.g., the MITDB-$m43$ instance with $n = 200$.

Overall, based on the empirical results, CPLEX on the proposed MIP model for MWCB(k, s, t) scales reasonably well on the real-world datasets under various parameter value combinations.

Solution Quality: Exact vs Greedy Finally, we look at the relative quality of solutions obtained on one hand using the exact MIP approach and, on the other hand, using the greedy algorithm presented in Section 4. Here we focus on the question of whether the higher computational cost of exact optimization pays off by offering in cases better solutions than the greedy approach.

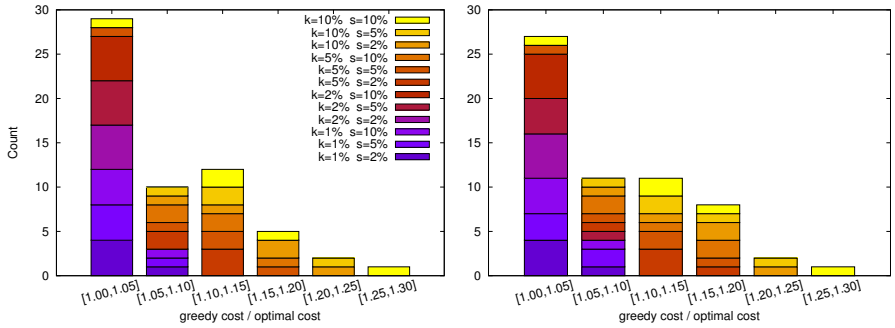

Fig. 7. Comparison of the relative cost of greedy and optimal solutions of solutions for the MITDB instances with $m = 43$ and $n \in \{100, 150, 200, 250, 300\}$. Left: $t = k$, right: $t = k + 2$. The other parameter values used were $k \in \{0.01n, 0.02n, 0.05n, 0.1n\}$ and $s \in \{0.02m, 0.05m, 0.1m\}$. The relative cost is provided for the 110 (out of 120) instances for which a provably optimal solution is found in 3600 seconds.

As witnessed by the results presented in Fig. 7, the optimal solutions are in cases non-negligibly better than those provided by the greedy approach. In more detail, the histograms in Fig. 7 show the counts of the relative costs of greedy and optimal solutions, defined as $\text{SIZE}(CB_{gr})/\text{SIZE}(CB_{opt})$, for instances based on the MITDB ($m = 43$) dataset with $t = k$ (left) and $t = k + 2$ (right).

We observe that there are greedy solutions that have a cost of up to approximately 127.5% of that of the optimal solution, while on average the cost of the greedy solution is 108% of the optimum for the MITDB-$m43$ instances.

Furthermore, we observed that the MIP approach can provide solutions with a low cost (without proving them optimal) often much faster than what it takes for CPLEX to prove the solutions found optimal. In detail, for 91 out of the 120 instances considered in Fig. 7, CPLEX provided a provably optimal solution in less than one minute on our MIP model. For 28 out of the remaining 29 instances, CPLEX provided within 60 seconds solutions with 7 % lower cost on average compared to the solutions provided by the greedy algorithm. Thus we observed that even in cases in which an optimal solution cannot be found fast, our MIP model can be typically used to obtain better than greedy solutions relatively fast.

These observations motivate the exact approach presented in this work, as well as future work on ways of further improving the scalability of exact approaches for the MWCB(k, s, t) problem. On the other hand, if solutions to very large instances of MWCB(k, s, t) are needed very fast, our greedy algorithm is also a viable option.

6 Related Work

The univariate confidence interval of a distribution based on a finite-sized sample from the distribution has been extensively studied (see, e.g., [7]). However, there are surprisingly few approaches to multivariate confidence intervals and most of the effort has

been focused on describing univariate distributions. Another alternative are the *confidence regions* (see, e.g., [6]), which however require making assumptions about the underlying distributions or which cannot be described simply by upper and lower bounds; e.g., confidence regions for multivariate Gaussian data are ellipsoids.

In the time series domain, multivariate confidence intervals [9, 11], namely *confidence bands* have been defined in terms of the *minimum-width envelope* (MWE) problem: a time series is within a confidence band if it is within the confidence interval of *every time point*, also see [14, 15, 21] for similar approaches. While this definition has desirable properties, it can result in very conservative confidence bands if there are local deviations from what constitutes as normal behaviour. To overcome this limitation, an alternative definition was recently introduced in [10, 24], where a data vector is within a confidence band if it is outside the confidence intervals of at most s elements, yielding the MWCB(k, s) problem extended further in this work.

MWCB(k, s) becomes quickly unfeasible as data/parameter values grow, as each of the points is potential for exclusion. Furthermore, as explained in Section 2, solutions to MWCB(k, s) can be problematic. In terms of greedy procedures for obtaining confidence intervals, the closest work to ours is [10] which focuses on MWCB(k, s). The quality of solutions of our greedy algorithm for MWCB(k) and MWCB(k, s) compared to those in [11, 10] depends on the data. In the typical case of $n > m$, [11] has higher time complexity than us. In terms of using exact constraint optimization to determining confidence bands, the only and closest work to our is [21] where a MIP model is provided for MWCB(k); we generalize here to MWCB(k, s, t). Our approach applies also to the special case of MWCB(k), although for capturing at the same time the more general setting considered here we use $n \times m$ binary variables (as compared to n binary variables in [21]).

7 Conclusions

We focused on the combinatorial optimization problem of determining tight (minimum-width) multivariate confidence bands as a central yet NP-hard optimization problem in data analysis. Pointing out drawbacks in earlier characterizations of the problem, we proposed a generalization MWCB(k, s, t) circumventing some of the earlier drawbacks. We proposed two constraint models allowing for exactly solving instances of MWCB(k, s, t), as well as a greedy algorithm for the problem. We studied the scalability of mixed integer programming and maximum satisfiability solvers on the respective constraint models, and observed that mixed integer programming especially provides good scalability on MWCB(k, s, t) instances based on real-world data. The greedy algorithm, on the other hand, can provide relatively good solutions very fast. However, we also showed empirically that the optimal solutions provided by the exact constraint-based approach can at times provide noticeably better solutions than the greedy approach, and can also provide relatively fast better quality solutions (without proving optimality). The study of potential alternative characterizations (e.g., objective functions) of the minimum-width confidence band problem which would still have the same benefits as MWCB(k, s, t) compared to MWCB(k) and MWCB(k, s) is one interest aspect for further work.

References

1. Asín, R., Nieuwenhuis, R., Oliveras, A., Rodríguez-Carbonell, E.: Cardinality networks: a theoretical and empirical study. Constraints 16(2), 195–221 (2011)
2. Biere, A., Heule, M., van Maaren, H., Walsh, T.: Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185. IOS Press (2009)
3. Davies, J., Bacchus, F.: Exploiting the power of MIP solvers in MaxSAT. In: Proc. SAT. Lecture Notes in Computer Science, vol. 7962, pp. 166–181. Springer (2013)
4. Gardner, M.J., Altman, D.G.: Confidence intervals rather than p values: estimation rather than hypothesis testing. British Medical Journal (Clinical Research Edition) 292(6522), 746–750 (1986)
5. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101(23), e215–e220 (2000)
6. Guilbaud, O.: Simultaneous confidence regions corresponding to Holm's step-down procedure and other closed-testing procedures. Biometrical Journal 50(5), 678 (2008)
7. Hyndman, R.J., Fan, Y.: Sample quantiles in statistical packages. The American Statistician 50(4), 361–365 (1996)
8. IBM ILOG: CPLEX optimizer (2017), `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`
9. Kolsrud, D.: Time-simultaneous prediction band for a time series. Journal of Forecasting 26(3), 171–188 (2007)
10. Korpela, J., Oikarinen, E., Puolamäki, K., Ukkonen, A.: Multivariate confidence intervals. In: Proc. SDM. pp. 696–704. SIAM (2017)
11. Korpela, J., Puolamäki, K., Gionis, A.: Confidence bands for time series data. Data Mining and Knowledge Discovery 28(5-6), 1530–1553 (2014)
12. Koshimura, M., Zhang, T., Fujita, H., Hasegawa, R.: QMaxSAT: A partial Max-SAT solver. Journal of Satisfiability, Boolean Modeling and Computation 8(1/2), 95–100 (2012)
13. Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
14. Liu, W., Jamshidian, M., Zhang, Y., Bretz, F., Han, X.: Some new methods for the comparison of two linear regression models. Journal of Statistical Planning and Inference 137(1), 57–67 (2007)
15. Mandel, M., Betensky, R.A.: Simultaneous confidence intervals based on the percentile bootstrap approach. Computational Statistics & Data Analysis 52(4), 2158–2165 (2008)
16. Menne, M., Durre, I., Korzeniewski, B., McNeal, S., Thomas, K., Yin, X., Anthony, S., Ray, R., Vose, R., Gleason, B., Houston, T.: Global Historical Climatology Network — Daily (GHCN-Daily), version 3.11 (2012)
17. Menne, M., Durre, I., Vose, R., Gleason, B., Houston, T.: An overview of the Global Historical Climatology Network-Daily Database. Journal of Atmospheric and Oceanic Technology 29, 897–910 (2012)
18. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH arrhythmia database. IEEE Engineering in Medicine and Biology Magazine 20(3), 45–50 (2001)
19. Morgado, A., Dodaro, C., Marques-Silva, J.: Core-guided MaxSAT with soft cardinality constraints. In: Proc. CP. Lecture Notes in Computer Science, vol. 8656, pp. 564–573. Springer (2014)
20. Nuzzo, R.: Scientific method: Statistical errors. Nature 506, 150–152 (2014)
21. Schüssler, R., Trede, M.: Constructing minimum-width confidence bands. Economics Letters 145, 182–185 (2016)

22. Staszewska-Bystrova, A., Winker, P.: Constructing narrowest pathwise bootstrap prediction bands using threshold accepting. International Journal of Forecasting 29(2), 221–233 (2013)
23. Trafimow, D., Marks, M.: Editorial. Basic and Applied Social Psychology 37(1), 1–2 (2015)
24. Wolf, M., Wunderli, D.: Bootstrap joint prediction regions. Journal of Time Series Analysis 36(3), 352–376 (2015)
25. Woolston, C.: Psychology journal bans P values. Nature 519, 9 (2015)