A Parallel Algorithm for Finding All Minimal Maximum Subsequences via Random Walk

H.K. Dai(✉) and Z. Wang

Computer Science Department, Oklahoma State University, Stillwater, Oklahoma 74078, USA
{dai,wzhu}@cs.okstate.edu

Abstract. A maximum(-sum) contiguous subsequence of a real-valued sequence is a contiguous subsequence with the maximum cumulative sum. A minimal maximum contiguous subsequence is a minimal contiguous subsequence among all maximum ones of the sequence. We have designed and implemented a domain-decomposed parallel algorithm on cluster systems with Message Passing Interface that finds all successive minimal maximum subsequences of a random sample sequence from a normal distribution with negative mean. Our study employs the theory of random walk to derive an approximate probabilistic length bound for minimal maximum subsequences in an appropriate probabilistic setting, which is incorporated in the algorithm to facilitate the concurrent computation of all minimal maximum subsequences in hosting processors. We also present a preliminary empirical study of the speedup and efficiency achieved by the parallel algorithm with synthetic random data.

Keywords: All maximum subsequences · Theory of random walk · Message passing interface · Parallel random access machine model

1 Preliminaries

Algorithmic and optimization problems in sequences and trees arise in widely varying domains such as bioinformatics and information retrieval. Large-scale (sub)sequence comparison, alignment, and analysis are important research areas in computational biology. Time- and space-efficient algorithms for finding multiple contiguous subsequences of a real-valued sequence having large cumulative sums help identify statistically significant subsequences in biological sequence analysis with respect to an underlying scoring scheme – an effective filtering preprocess even with simplistic random-sequence models of independent residues.

For a real-valued sequence $X = (x_\eta)_{\eta=1}^{n}$, the cumulative sum of a non-empty contiguous subsequence $(x_\eta)_{\eta=i}^{j}$, where i and j are in the index range $[1, n]$ with $i \leq j$, is $\sum_{\eta=i}^{j} x_\eta$ (and that of the empty sequence is 0). All subsequences addressed in our study are contiguous in real-valued sequences; the terms "subsequence" and "supersequence" will hereinafter abbreviate "contiguous subsequence" and "contiguous supersequence", respectively. A maximum(-sum) subsequence of X is one with the maximum cumulative sum. A minimal maximum subsequence of X is a

A.-H. Dediu et al. (Eds.): LATA 2015, LNCS 8977, pp. 133–144, 2015.
DOI: 10.1007/978-3-319-15579-1_10

minimal subsequence (with respect to subsequential containment) among all maximum subsequences of X.

Very often in applications it is required to find many or all pairwise disjoint subsequences having cumulative sums above a prescribed threshold. Observe that subsequences having major overlap with a maximum subsequence tend to have good cumulative sums. Intuitively, we define the sequence of all successive minimal maximum subsequences $(S_1, S_2, \ldots)$ of X inductively as follows: (1) The sequence S_1 is a (non-empty) minimal maximum subsequence of X, and (2) Assume that the sequence $(S_1, S_2, \ldots, S_i)$ of non-empty subsequences of X, where $i \geq 1$, has been constructed, the subsequence S_{i+1} is a (non-empty) minimal subsequence (with respect to subsequential containment) among all non-empty maximum subsequences (with respect to cumulative sum) that are disjoint from each of $\{S_1, S_2, \ldots, S_i\}$.

Efficient algorithms for computing the sequence of all successive minimal maximum subsequences of a given sequence are essential for statistical inference in large-scale biological sequence analysis. In biomolecular sequences, high (sub)sequence similarity usually implies significant structural or functional similarity. When incorporating good scoring schemes, this provides a powerful statistical paradigm for identifying biologically significant functional regions in biomolecular sequences [8], such as transmembrane regions and deoxyribonucleic acid-binding domains [6] in protein analyses. The non-positivity of the expected score of a random single constituent tends to delimit unrealistic long runs of contiguous positive scores.

We design and implement a domain-decomposed parallel algorithm on cluster systems with Message Passing Interface that finds all successive minimal maximum subsequences of a random sample sequence from a normal distribution with negative mean. A brief summary of a preliminary empirical study of the speedup and efficiency achieved by the parallel algorithm is also presented. Our study is motivated by the linear-time sequential algorithm [8] and a logarithmic-time and optimal-work parallel algorithm on the parallel random access machine (PRAM) [3] for this computation problem.

For computing a single (minimal) maximum subsequence of a length-n real-valued sequence of X, a simple sequential algorithm solves this problem in $O(n)$ optimal time. A parallel algorithm [1] on the PRAM model solves the single maximum subsequence problem in $O(\log n)$ parallel time using a total of $O(n)$ operations (work-optimal). A generalization of the problem and the selection problem is the sum-selection that, for given input length-n sequence X, range-bound $[l, u]$, and rank k, finds a subsequence of X such that the rank of its cumulative sum is k among all subsequences with cumulative sum in $[l, u]$. A randomized algorithm [7] solves the sum-selection problem in expected $O(n \log(u - l))$ time.

For the problem of finding the sequence of all successive minimal maximum subsequences of a length-n real-valued sequence X, a recursive divide-and-conquer strategy can apply the linear-time sequential algorithm above to compute a minimal maximum subsequence of X whose deletion results in a prefix and a suffix for recursion. The algorithm has a (worst-case) time

complexity of $\Theta(n^2)$. Empirical analyses of the algorithm [8] on synthetic data sets (sequences of independent and identically distributed uniform random terms with negative mean) and score sequences of genomic data indicate that the running time grows at $\Theta(n \log n)$.

In order to circumvent the iterative dependency in computing the sequence of all successive minimal maximum subsequences, Ruzzo and Tompa [8] prove a structural characterization of the sequence as follows. Denote by $\mathrm{Max}(X)$ the set of all successive minimal maximum subsequences or their corresponding index subranges (when the context is clear) of a real-valued sequence X.

Theorem 1. [8] *For a non-empty real-valued sequence X, a non-empty subsequence S of X is in* $\mathrm{Max}(X)$ *if and only if: (1)* [*monotonicity*] *the subsequence S is monotone: every proper subsequence of S has its cumulative sum less than that of S, and (2)* [*maximality of monotonicity*] *the subsequence S is maximal in X with respect to monotonicity, that is, every proper supersequence of S contained in X is not monotone.*

Hence, we also term $\mathrm{Max}(X)$ as the set of all maximal monotone subsequences of X. This gives a structural decomposition of X into $\mathrm{Max}(X)$: (1) every non-empty monotone subsequence of X is contained in a maximal monotone subsequence in $\mathrm{Max}(X)$; in particular, every positive term of X is contained in a maximal monotone subsequence in $\mathrm{Max}(X)$, and (2) the set $\mathrm{Max}(X)$ is a pairwise disjoint collection of all maximal monotone subsequences of X.

Based on the structural characterization of $\mathrm{Max}(X)$, Ruzzo and Tompa present a sequential algorithm that computes $\mathrm{Max}(X)$ in $O(n)$ optimal sequential time and $O(n)$ space (worst case). Alves, Cáceres, and Song [2] develop a parallel algorithm for computing $\mathrm{Max}(X)$ of a length-n sequence X on the bulk synchronous parallel/coarse grained multicomputer model of p processors in $O(\frac{n}{p})$ computation time and $O(1)$ communication rounds.

In the following section, we introduce other structural decompositions of a sequence X that lead to computing $\mathrm{Max}(X)$ with: (1) a parallel algorithm on the PRAM model [3] in logarithmic parallel time and optimal linear work, and (2) a domain-decomposed parallel algorithm implemented on cluster systems with Message Passing Interface. This paper presents the skeletons for the main results without lengthy derivations and proofs, which are detailed in the full version.

2 Structural Decompositions of X Leading to $\mathrm{Max}(X)$

For a real-valued sequence $X = (x_\eta)_{\eta=1}^n$, denote by $s_i(X)$ the i-th prefix sum $\sum_{\eta=1}^i x_\eta$ of X for $i \in [1, n]$, and $s_0(X) = 0$. We abbreviate the prefix sums $s_i(X)$ to s_i for all $i \in [1, n]$ when the context is clear. For a subsequence Y of X, denote by $\alpha(Y; X)$, $\beta(Y; X)$, and $\gamma(Y; X)$ its starting index, ending index, and index subrange $[\alpha(Y; X), \beta(Y; X)]$ ($\gamma(Y; X) = \emptyset$ if Y is empty) in the context of X, respectively, and by $\gamma_+(Y; X)$ the set of all indices in $\gamma(Y; X)$ yielding positive terms of Y. When considering the subsequence Y as a sequence in its

own context we abbreviate $\alpha(Y;Y)$, $\beta(Y;Y)$, $\gamma(Y;Y)$, and $\gamma_+(Y;Y)$ to $\alpha(Y)$, $\beta(Y)$, $\gamma(Y)$, and $\gamma_+(Y)$, respectively.

The following characterization of monotonicity [3] yields an effective computation of the index subrange of a non-trivial monotone subsequence containing a given term of X.

Lemma 1. *Let X be a non-empty real-valued sequence and Y be a non-empty subsequence of X (with index subrange $[\alpha(Y;X), \beta(Y;X)]$). The following statements are equivalent:*

1. *Y is monotone in X.*
2. *The starting prefix sum $s_{\alpha(Y;X)-1}(X)$ of Y is the unique minimum and the ending prefix sum $s_{\beta(Y;X)}(X)$ of Y is the unique maximum of all $s_i(X)$ for all $i \in [\alpha(Y;X) - 1, \beta(Y;X)]$.*
3. *All non-empty prefixes and non-empty suffixes of Y have positive cumulative sums.*

The key to the parallel implementation [3] of finding $\mathrm{Max}(X)$ for a length-n sequence $X = (x_\eta)_{\eta=1}^n$ lies in the concurrent computation of the ending index of the maximal monotone subsequence constrained with the starting index $i \in \gamma(X)$. Lemma 1 suggests to consider only positive terms x_i of X for the desired computation. Let $\epsilon : \gamma_+(X) \to \gamma(X)$ be the function that $\epsilon(i)$ denotes the ending index of the maximal monotone subsequence of X constrained with the starting index i. The concurrent computation of ϵ via the computations of all-nearest-smaller-values and range-minima, when applied to all the positive terms x_i in X, generates the statistics $\mathrm{Mon}(X) = \{[i, \epsilon(i)] \mid i \in \gamma_+(X)\}$ for the set of all index subranges of all maximal monotone subsequences of X constrained with given positive starting terms. The following theorem [3] reveals the structural decomposition of X into $\mathrm{Mon}(X)$, which refines $\mathrm{Max}(X)$ and provides a basis for a parallel computation of $\mathrm{Max}(X)$ from $\mathrm{Mon}(X)$.

Theorem 2. *For a real-valued sequence X, $\mathrm{Mon}(X)$ enjoys the following parenthesis structure:*

1. *Every positive term of X has its index as the starting index of a unique index subrange in $\mathrm{Mon}(X)$,*
2. *For every pair of index subranges in $\mathrm{Mon}(X)$, either they are disjoint or one is a subrange of another, and*
3. *For every maximal monotone subsequence of X in $\mathrm{Max}(X)$, its index subrange is in $\mathrm{Mon}(X)$.*

Our current work on Max-computation includes adapting the logarithmic-time optimal-work parallel algorithm on practical parallel systems. However, in view of the efficient linear-time sequential algorithm [8], we devise and implement a domain-decomposed parallel algorithm computing Max that employs the optimal sequential algorithm in subsequence-hosting processors.

An ideal domain decomposition of a sequence X is a partition of X into a pairwise disjoint family $\mathcal{X}$ of non-empty subsequences of X that are length-balanced and Max-independent: $\mathrm{Max}(X) = \cup_{Y \in \mathcal{X}} \mathrm{Max}(Y)$ (Y as a sequence in

its own right). We first finds a sufficient condition for the Max-independence that can be computed locally in subsequence-hosting processors. The characterization of monotonicity in Lemma 1 suggests to consider the following two functions on indices of positive terms of X with index range $\gamma(X)$ $(= [1, n])$. Let $\mathrm{rm}_X : \gamma_+(X) \to [\alpha(X) + 1, \beta(X)] \cup \{\beta(X) + 1\}$ $(= [2, n + 1])$ denote the nearest-smaller-or-equal right-match of the prefix sum s_{i-1} of X:

$$\mathrm{rm}_X(i) = \begin{cases} \min\{\eta \in [i+1, \beta(X)] \mid s_{i-1} \geq s_\eta\} & \text{if the minimum exists,} \\ \beta(X) + 1 \,(= n + 1) & \text{otherwise.} \end{cases}$$

A symmetric analogue of rm_X is the nearest-smaller left-match function lm_X. Note that the families $\{[\mathrm{lm}_X(i), i] \mid i \in \gamma_+(X)\}$ and $\{[i, \mathrm{rm}_X(i)] \mid i \in \gamma_+(X)\}$ satisfy the parenthesis structure similar to that of Mon – but permitting abutting index subranges (at subrange ends) in the lm_X-family. Both lm_X and rm_X help locate the (minimum) starting and (maximum) ending indices, respectively, of a maximal monotone subsequence of X containing the positive term x_i: determine if a merge of multiple maximal monotone subsequences covering the index subrange $[\mathrm{lm}_X(i), i]$ may occur.

Lemmas 2 and 3 give a sufficient condition for the Max-independence of a partition of X based on a local computation of rm_{X_i} and its intuitive equivalence by the X_i-hosting processor for each $i \in \{1, 2, \ldots, n\}$.

Lemma 2. *Let $(X_\eta)_{\eta=1}^m$ be a sequential partition of a real-valued sequence X with X_η, for $\eta = 1, 2 \ldots, m$, represented as a sequence in its own right over its index range $\gamma(X_\eta)$. If the partition satisfies the* rm*-closure condition: for all $i \in \{1, 2, \ldots, m-1\}$ and all $j \in \gamma_+(X_i)$, $\mathrm{rm}_{X_i}(j) \in [j+1, \beta(X_i)]$, then the partition is* Max*-independent: $\mathrm{Max}(X) = \cup_{\eta=1}^m \mathrm{Max}(X_\eta)$.*

Lemma 3. *For a non-empty real-valued sequence Y, the right-match function $\mathrm{rm}_Y : \gamma_+(Y) \to [\alpha(Y)+1, \beta(Y)] \cup \{\beta(Y)+1\}$ satisfies the* rm*-closure condition stated in Lemma 2 (for all $j \in \gamma_+(Y), \mathrm{rm}_Y(j) \in [j+1, \beta(Y)]$) if and only if the sequence Y satisfies the minimum prefix-sum condition: the ending prefix sum of Y, $s_{\beta(Y)}(Y)$, is a global minimum of all $s_i(Y)$ for all $i \in [\alpha(Y) - 1, \beta(Y)]$.*

The minimum prefix-sum condition, equivalent to the rm-closure condition as shown in Lemma 3, exposes a stringent sufficiency for Max-independence of a priori sequential partition of a sequence X: for all $i \in \{1, 2, \ldots, m-1\}$, the ending prefix sum is a global minimum of all prefix sums of X_i. We incorporate the minimum prefix-sum condition into constructing a posteriori sequential partition of X that forms the basis in designing a domain-decomposed parallel algorithm in computing $\mathrm{Max}(X)$.

For two sequences X and Y, denote the concatenation of X and Y by the juxtaposition XY. Let X be a non-empty real-valued sequence with a sequential partition $\mathcal{P}(X) = (X_1, X_{1,2}, X_2, X_{2,3}, X_3, \ldots, X_{m-1}, X_{m-1,m}, X_m)$. For notational simplicity, let $X_{0,1} = \emptyset$ and $X_{m,m+1} = \emptyset$.

For every $i \in \{1, 2, \ldots, m-1\}$, denote by β_i^* the maximum/right-most index $\eta \in \gamma_+(X_{i-1,i}X_i)$, if non-empty, such that $s_{\eta-1}(X_{i-1,i}X_i)$ is the minimum prefix sum of those of $X_{i-1,i}X_i$ over $\gamma_+(X_{i-1,i}X_i)$; that is,

$$\beta_i^* = \max \arg\min\{s_{\eta-1}(X_{i-1,i}X_i) \mid \eta \in \gamma_+(X_{i-1,i}X_i)\ (\neq \emptyset)\}.$$

The sequential partition $\mathcal{P}(X)$ satisfies the rm-locality condition if for every $i \in \{1, 2, \ldots, m-1\}$ with non-empty $\gamma_+(X_{i-1,i}X_i)$, $\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) \in [\beta_i^* + 1, \beta(X_{i-1,i}X_iX_{i,i+1})]$.

The rm-localized sequential partition $\mathcal{P}(X)$ derives a Max-independent partition $\tilde{\mathcal{P}}(X) = (X''_{i-1,i}X_iX'_{i,i+1})_{i=1}^m$ where $X''_{i-1,i}$ and $X'_{i,i+1}$ are respectively the suffix of $X_{i-1,i}$ and prefix of $X_{i,i+1}$ that are determined by rm-computation as follows. Recall that $X_{0,1} = \emptyset$ and $X_{m,m+1} = \emptyset$, let $X''_{0,1} = \emptyset$ and $X'_{m,m+1} = \emptyset$ accordingly. For every $i \in \{1, 2, \ldots, m-1\}$, define $X'_{i,i+1}$ as:

$$\begin{cases} \emptyset \text{ if } \gamma_+(X_{i-1,i}X_i) = \emptyset \vee \\ \quad \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) \in [\beta_i^* + 1, \beta(X_{i-1,i}X_i; X_{i-1,i}X_iX_{i,i+1})], \\ \text{the prefix of } X_{i,i+1} \text{ with} \\ \text{index subrange } [\alpha(X_{i,i+1}; X_{i-1,i}X_iX_{i,i+1}), \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*)] \\ \quad \text{if } \gamma_+(X_{i-1,i}X_i) \neq \emptyset \wedge \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) \in \gamma(X_{i,i+1}; X_{i-1,i}X_iX_{i,i+1}), \end{cases}$$

and $X''_{i,i+1}$ to be the (remaining) suffix of $X_{i,i+1}$ such that $X'_{i,i+1}X''_{i,i+1} = X_{i,i+1}$. Note that the first case in defining $X'_{i,i+1}$ may be absorbed into the second case.

Theorem 3. *Let X be a non-empty real-valued sequence with an* rm*-localized sequential partition $\mathcal{P}(X) = (X_1, X_{1,2}, X_2, X_{2,3}, X_3, \ldots, X_{m-1}, X_{m-1,m}, X_m)$ and its derived sequential partition $\tilde{\mathcal{P}}(X) = (X''_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1})_{\eta=1}^m$. Then:*

1. *$\tilde{\mathcal{P}}(X)$ is* Max*-independent:* $\mathrm{Max}(X) = \cup_{\eta=1}^m \mathrm{Max}(X''_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1})$*, and*
2. *For all $i \in \{1, 2, \ldots, m\}$,* $\mathrm{Max}(X''_{i-1,i}X_iX'_{i,i+1}) = \mathrm{Max}(X_{i-1,i}X_iX'_{i,i+1}) - \{Y \in \mathrm{Max}(X_{i-1,i}X_iX'_{i,i+1}) \mid \alpha(Y; X_{i-1,i}X_iX'_{i,i+1}) \in \gamma(X'_{i-1,i}; X_{i-1,i}X_iX'_{i,i+1})\}$*; so* $\mathrm{Max}(X) = \cup_{\eta=1}^m (\mathrm{Max}(X_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1}) - \{Y \in \mathrm{Max}(X_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1}) \mid \alpha(Y; X_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1}) \in \gamma(X'_{\eta-1,\eta}; X_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1})\})$.

3 Probabilistic Analysis of the Locality Condition

The structural decomposition of a non-empty real-valued sequence X in Theorem 3 suggests a basis for an ideal decomposition of X with length-balance and Max-independence – provided the decomposition satisfies the rm-locality condition. While the rm-localized decomposition $\tilde{\mathcal{P}}(X)$ is the (derived) sequential partition $(X''_{\eta-1,\eta}X_\eta X'_{\eta,\eta+1})_{\eta=1}^m$ in m pairwise disjoint subsequences, our domain-decomposed parallel algorithm computing $\mathrm{Max}(X)$ will employ m processors with the i-th processor hosting the subsequence $X_{i-1,i}X_iX_{i,i+1}$ for $i \in \{1, 2, \ldots, m\}$. The subsequences $X_{i-1,i}X_iX_{i,i+1}$ and $X_{i,i+1}X_{i+1}X_{i+1,i+2}$ hosted in successive i-th and $(i+1)$-th processors have the common subsequence $X_{i,i+1}$ that serves as a buffer to capture the rm-locality originated from $X_{i-1,i}X_i$ and a floating separation between successive Max-sets: $\mathrm{Max}(X''_{i-1,i}X_iX'_{i,i+1})$ and $\mathrm{Max}(X''_{i,i+1}X_{i+1}X'_{i+1,i+2})$. A longer common subsequence facilitates the satisfiability of the rm-locality of the preceding subsequence while a shorter one avoids redundant computation among successive processors.

In this section we analyze the length bound of the common subsequences probabilistically for random sequences of normally-distributed terms – via the theory of random walk. Let $X_1, X_2, \ldots$ be a sequence of pairwise independent and identically distributed random variables. Denote by $(S_\eta)_{\eta=0}^{\infty}$ the sequence of prefix-sum random variables with $S_0 = 0$ and $S_i = \sum_{\eta=1}^{i} X_\eta$ for $i \geq 1$, which corresponds to a general random walk for which S_i gives the position at epoch/index i. A record value occurs at (random) epoch $i \geq 1$ corresponds to the probabilistic event "$S_i > S_\eta$ for each $\eta \in [0, i-1]$". For every positive integer j, the j-th strict ascending ladder epoch random variable is the index of the j-th occurrence of the probabilistic event above. We define analogously the notions of: (1) strict descending ladder epochs by reversing the defining inequality from "$>$" to "$<$", and (2) weak ascending and weak descending epochs by replacing the defining inequalities by "$\geq$" and "$\leq$", respectively.

The first strict ascending ladder epoch is the random index of the first entry into $(0, +\infty)$, and the continuation of the random walk beyond this epoch is a probabilistic replica of the entire random walk. Other variants of (strict/weak, ascending/descending) ladder epoch yield similar behavior.

Viewing the sequence X in the Max-computation in an appropriate probabilistic setting studied below and following the above-stated denotations and construction of the Max-independent sequential partition $\tilde{\mathcal{P}}(X)$ from an rm-localized sequential partition $\mathcal{P}(X)$, we: (1) see intuitively that the random index-difference $\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) - \beta_i^* + 1$ behaves like the first weak descending ladder epoch T of the underlying random walk (yielding $\sum_{\eta=1}^{\kappa} y_{\eta+\beta_i^*-1}$ for $\kappa = 0, 1, \ldots$) conditional on the probabilistic event "the positivity of the first term $y_{\beta_i^*}$" – with finite variance (and mean), and (2) develop a probabilistic upper bound on the length of the common subsequences in $\tilde{\mathcal{P}}(X)$ via the mean and variance of a variant of the first ladder epoch.

Remark 1. Ideally in $\tilde{\mathcal{P}}(X)$, we desire that:

$$\begin{aligned} &|X_{i,i+1}| \,(= |[\alpha(X_{i,i+1}; X_{i-1,i}X_iX_{i,i+1}), \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*)]|) \\ &\leq |[\beta_i^*, \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*)]| = \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) - \beta_i^* + 1. \end{aligned}$$

Thus, if we select the common subsequence $X_{i,i+1}$ such that $|X_{i,i+1}| \geq \lceil \mathrm{E}(T) + \delta\sqrt{\mathrm{Var}(T)} \rceil$ for some positive real δ, then the following two probabilistic events satisfy the subset-containment:

$$\begin{aligned} &\text{“}\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) - \beta_i^* + 1 \geq |X_{i-1,i}|\text{”} \\ &\subseteq \text{“}(\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) - \beta_i^* + 1) - \mathrm{E}(T) \geq \delta\sqrt{\mathrm{Var}(T)}\text{”}, \end{aligned}$$

and, in accordance with Chebyshev's inequality,

$$\begin{aligned} &\mathrm{pr}\,(\text{random index-difference } \mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) - \beta_i^* + 1 \geq |X_{i-1,i}|) \\ &\leq \mathrm{pr}\,(T - \mathrm{E}(T) \geq \delta\sqrt{\mathrm{Var}(T)}) \leq \mathrm{pr}\,(|T - \mathrm{E}(T)| \geq \delta\sqrt{\mathrm{Var}(T)}) \leq \frac{1}{\delta^2}. \end{aligned}$$

These will be applied to bound the likelihood of (non-)satisfiability of the rm*-locality condition for $\mathcal{P}(X)$.*

We now relate the conditional weak descending ladder epoch T to the unconditional one and then, in an appropriate probabilistic setting, the means and variances of the two random variables.

For a sequence of pairwise independent and identically distributed random variables $X_1, X_2, \ldots$ and its associated random-walk sequence $(S_\eta)_{\eta=0}^{\infty}$ of prefix-sum random variables, denote by T_1 its first weak descending ladder epoch. Assume hereinafter that $(X_\eta)_{\eta=1}^{\infty}$ follows a common random variable X_1 with $\mathrm{pr}\,(X_1 > 0) \geq 0$. For notational simplicity, denote by p and $\bar{p}$ $(= 1 - p)$ the probabilities $\mathrm{pr}\,(X_1 > 0)$ and $\mathrm{pr}\,(X_1 \leq 0)$, respectively.

The unconditional and conditional ladder epochs T_1 and T $(= T_1 \mid X_1 > 0)$ have sample spaces of $\{1, 2, \ldots\}$ and $\{2, 3, \ldots\}$, respectively, and for every $t \in \{2, 3, \ldots\}$,

$$\mathrm{pr}\,(T = t) = \mathrm{pr}\,(T_1 = t \mid X_1 > 0) = \frac{\mathrm{pr}\,(T_1 = t \cap X_1 > 0)}{\mathrm{pr}\,(X_1 > 0)} = \frac{1}{p}\,\mathrm{pr}\,(T_1 = t)$$

due to the subset-containment of the probabilistic events: "$T_1 = t\ (\geq 2)$" $\subseteq$ "$X_1 > 0$".

Lemma 4. *Assume that the variance, hence the mean, of the unconditional weak descending ladder epoch T_1 exist. The means and variances of the unconditional and conditional ladder epochs T_1 and $T = T_1 \mid X_1 > 0$ are related as follows:*

$$(1)\ \mathrm{E}(T) = \frac{1}{p}\,\mathrm{E}(T_1) - \frac{\bar{p}}{p} \quad \textit{and} \quad (2)\ \mathrm{Var}(T) = \frac{1}{p}\,\mathrm{Var}(T_1) - \bar{p}\left(\frac{1}{p}(E(T_1) - 1)\right)^2.$$

Remark 2. Remark 1 and Lemma 4 suggest to seek lower and upper bounds on $\mathrm{E}(T_1)$ *and an upper bound on* $\mathrm{Var}(T_1)$ *for their use with the mean- and variance-relationships – which translate to non-trivial bounds on* $\mathrm{E}(T)$ *and* $\mathrm{Var}(T)$. *Note that, by the assumption of* $\mathrm{pr}\,(X_1 > 0)$, *we have* $\mathrm{E}(T_1) > 1$.

For our Max-computing problem, we assume hereinafter (unless explicitly stated otherwise) that the sequence $X = (x_\eta)_{\eta=1}^{n}$ is a random sample from a normal distribution with mean $-a$ and variance b^2 for some positive reals a and b. That is, a sequence of pairwise independent and identically distributed random variables $X_1, X_2, \ldots$ with a common normal distribution with mean$-a$ and variance b^2 gives rise to the observed values $x_1, x_2, \ldots$. In applications, the knowledge of the mean and variance of the common random variable is known (see a uniformly-distributed case studied in [8]) or can be approximated.

The negativity of the mean $(-a)$ of the underlying normal distribution is desired in order to avoid yielding unrealistically long minimal maximum subsequences for viable applications. Formally for the induced random-walk sequence $(S_\eta)_{\eta=0}^{\infty}$ of $(X_\eta)_{\eta=1}^{\infty}$, since $\mathrm{E}(X_1)$ is finite and negative, the first (weak descending) ladder epoch T_1 has a proper probability distribution with finite mean and the random walk drifts to $-\infty$. For notational simplicity, denote by λ the "mean to standard deviation" ratio $\frac{\mathrm{E}(X_1)}{\sqrt{\mathrm{Var}(X_1)}}$; $\lambda = \frac{-a}{b}$ for a common normal distribution X_1 with mean $-a$ and standard deviation b.

Theorem 4. *For a sequence of pairwise independent and identically distributed random variables* $(X_\eta)_{\eta=1}^{\infty}$ *with a negative (common) finite mean* $\mathrm{E}(X_1)$ *and a positive probability* $p\,(=\ \mathrm{pr}\,(X_1 > 0))$*, the unconditional and conditional first weak descending epochs,* T_1 *and* $T\,(=\ T_1 \mid X_1 > 0)$ *respectively, satisfy the followings:*

1. [*General Case: Means*] *For* T_1*:* $\mathrm{E}(T_1) = \exp(\sum_{\eta=1}^{\infty} \frac{\mathrm{pr}\,(S_\eta > 0)}{\eta})$*; for* T*:* $\mathrm{E}(T) = \frac{1}{p}\exp(\sum_{\eta=1}^{\infty} \frac{\mathrm{pr}\,(S_\eta > 0)}{\eta}) - \frac{\bar{p}}{p}$.
2. [*Normally-Distributed Case: Means*] *For a common normal distribution of* $(X_\eta)_{\eta=1}^{\infty}$ *with mean* $-a$ *and variance* b^2 *for some positive reals* a *and* b *and for every positive integer* l*, denote* $B(\lambda, l, \eta) = 1 - \exp(-\frac{\lambda^2}{2\sin^2(\eta\pi/(2l))})$ *for* $\eta \in \{1, 2, \ldots, l\}$*, then:*

$$\text{for } T_1\text{: } 1 < (\prod_{\eta=1}^{l-1} B(\lambda, l, \eta))^{-\frac{1}{2l}} \le \mathrm{E}(T_1) \le (\prod_{\eta=1}^{l} B(\lambda, l, \eta))^{-\frac{1}{2l}};$$

$$\text{for } T\text{: } \frac{1}{p}(\prod_{\eta=1}^{l-1} B(\lambda, l, \eta))^{-\frac{1}{2l}} - \frac{\bar{p}}{p} \le \mathrm{E}(T) \le \frac{1}{p}(\prod_{\eta=1}^{l} B(\lambda, l, \eta))^{-\frac{1}{2l}} - \frac{\bar{p}}{p}.$$

For our purpose in this study, we consider $l = 6$, and denote by μ' and μ'' the lower and upper bounds on the mean $\mathrm{E}(T_1)$ obtained in Theorem 4.

Remark 3. The range-constraint on $\mathrm{E}(T_1)$*:* $\mathrm{E}(T_1) \in [\mu', \mu'']$ *induces an upper bound on* $\mathrm{Var}(T_1)$ *via some stochastic relationships of the first- and second-order moments of the first weak descending ladder epoch* T_1*, its associate (first weak descending) ladder height* S_{T_1}*, and the common distribution* X_1 *of the underlying random walk.*

The following scenario will appear in upper-bounding $\mathrm{Var}(T_1)$ *and* $\mathrm{Var}(T)$*: a quadratic polynomial* Q *with negative leading coefficient and two distinct real roots* r' *and* r'' $(r' < r'')$ *serves as an upper bound on a nonnegative quantity* v *(such as a variance):* $0 \le v \le Q(s)$ *where* s *is a real-valued statistics – which induces a range-constraint:* $s \in [r', r'']$.

Denote by q_1 and q the two quadratic polynomial forms that represent upper bounds on $\mathrm{Var}(T_1)$ and $\mathrm{Var}(T)$, respectively, in Theorem 5 below:

1. $q_1(t) = 2(-t^2 + (1 + \frac{2}{\lambda^2})t)$ with distinct real roots r_1' and r_1'' $(r_1' < r_1'')$, and
2. $q(t) = -(2 + \frac{\bar{p}}{p^2})t^2 + 2(1 + \frac{2}{\lambda^2} + \frac{\bar{p}}{p^2})t - \frac{\bar{p}}{p^2}$ with distinct real roots r' and r'' $(r' < r'')$.

Theorem 5. *For a sequence of pairwise independent and identically distributed random variables* $(X_\eta)_{\eta=1}^{\infty}$ *with a negative (common) finite mean* $\mathrm{E}(X_1)$*, a finite (common) third-order absolute moment* $\mathrm{E}(|X_1|^3)$*, and a positive probability* $p\,(=\ \mathrm{pr}\,(X_1 > 0))$*, the unconditional and conditional first weak descending epochs* T_1 *and* $T\,(=\ T_1 \mid X_1 > 0)$ *respectively, satisfy the followings:*

1. [*General Case: Means and Variances*] *For* T_1*:* $r' \leq \mathrm{E}(T_1) \leq r''$ *and*

$$\mathrm{Var}(T_1) < q_1(\mathrm{E}(T_1)) = 2(-\mathrm{E}(T_1)^2 + (1 + \frac{2}{\lambda^2})\,\mathrm{E}(T_1));$$

for T*:* $\frac{1}{p}r' - \frac{\bar{p}}{p} \leq \mathrm{E}(T) \leq \frac{1}{p}r'' - \frac{\bar{p}}{p}$ *and*

$$\mathrm{Var}(T) < q(\mathrm{E}(T_1)) = -(2 + \frac{\bar{p}}{p^2})\,\mathrm{E}(T_1)^2 + 2(1 + \frac{2}{\lambda^2} + \frac{\bar{p}}{p^2})\,\mathrm{E}(T_1) - \frac{\bar{p}}{p^2}.$$

2. [*Normally-Distributed Case: Means and Variances*] *With a common normal distribution of* $(X_\eta)_{\eta=1}^{\infty}$ *with mean* $-a$ *and variance* b^2 *for some positive reals* a *and* b*:*

$$\text{for } T_1\text{: } \mu' \leq \mathrm{E}(T_1) \leq \mu'' \text{ and } \mathrm{Var}(T_1) < q_1(\mathrm{E}(T_1));$$
$$\text{for } T\text{: } \frac{1}{p}\mu' - \frac{\bar{p}}{p} \leq \mathrm{E}(T) \leq \frac{1}{p}\mu'' - \frac{\bar{p}}{p} \text{ and } \mathrm{Var}(T) < q(\mathrm{E}(T_1)).$$

4 Max-Algorithms, Performance, and Conclusion

We have implemented a Max-computing parallel algorithm on cluster systems in which subsequence-hosting processors employ an optimal linear-time sequential algorithm Max_Sequential (which is detailed in the full version) for local Max-computation. Improvements to the algorithms and work in progress will be addressed in the conclusion. The algorithms implemented with Message Passing Interface (MPI) are available from the authors.

The performance of the parallel algorithm Max_Parallel is assessed in a preliminary empirical study on a cluster with synthetic random data as follows: (1) $N = 100$ trial-sequences, each is a random sample/sequence of length $n = 5 \cdot 10^6$ from a normal distribution with mean -0.25 and variance 1.0, and (2) Performance measures in (absolute) speedup and efficiency of Max_Parallel are collected in two sets of mean-statistics: (2.1) the set of conditional mean-statistics on "success" scenario (satisfiability of the rm-locality condition for the first $(p-1)$ processors) from N trial-sequences and the Max-computing by (local) Max_Sequential in Max_Parallel: Steps 1 – 3, and (2.2) the set of unconditional ones for Max_Parallel: all steps.

Based on the optimal sequential-time algorithm [8], the (mean) optimal sequential time for Max-computation of a length-n sequence, $T^*(n)$, is approximately 0.155881 sec for the synthetic random data prepared in item 1 above (when averaged over $N = 100$ sequences).

Table 1 summarizes the above-stated two sets of mean-statistics of the running time, speedup, and efficiency of Max_Parallel for $\delta = 3$ (in Remark 1 and Max_Parallel: Step 1) and m processors with $m \in \{1, 2, 4, 8, 16, 32, 64\}$: $T_m(n)$ (in seconds), $S_m(n) = \frac{T^*(n)}{T_m(n)}$, and $E_m(n) = \frac{T_1(n)}{mT_m(n)}$, respectively.

Since pr (satisfiability of rm-locality for single processor) $\geq 1 - \frac{1}{\delta^2}$ $(= \frac{8}{9})$, the expected number N_s of "successes" from N trial-sequences is bounded below:

$N_s \geq N(1 - \frac{1}{\delta^2})^{m-1}$. The empirical and statistical results tabulated in the two columns: (expected) N_s and empirical-N_s show that the constraints on $\mathrm{E}(T)$ and $\mathrm{Var}(T)$ (Theorem 5: part 2) in bounding $E(T) + \delta\sqrt{\mathrm{Var}(T)}$ (Max_Parallel: Step 1) serves as a good lower-bound predictor for N_s. For the conditional statistics on "success" scenario, the speedup and efficiency are close to their theoretical bounds of m and 1, respectively. For the unconditional ones, even for a small δ $(= 3)$, the speedup and efficiency exceed $\frac{3}{4}$ of their theoretical bounds, except for $m = 64$. The speedup and efficiency performance of an improved Max_Parallel depends on the extent of resolving violations of rm-locality among neighbor processors and tradeoffs involving δ and m.

Algorithm Max_Parallel**.**

Require: A length-n real-valued sequence X (which is a random sample satisfying the assumptions in Theorem 5: part 2) and a prescribed probability threshold δ (Remark 1: Chebyshev's inequality).

Ensure: The sequence of all successive minimal maximum subsequences (all maximal monotone subsequences) of X.

1: Construct sequential partition $\mathcal{P}(X) = (X_1, X_{1,2}, X_2, X_{2,3}, X_3, \ldots, X_{m-1}, X_{m-1,m}, X_m)$ of X (stated in Section 3) such that: (1) for all $i \in \{1, 2, \ldots, m\}$, processor P_i hosts the subsequence $X_{i,i-1}X_iX_{i,i+1}$ in a length-balanced manner except possibly for the last processor P_m, and (2) for all $i \in \{1, 2, \ldots, m-1\}$, $|X_{i,i+1}|$ is the least upper bound of $\lceil \mathrm{E}(T) + \delta\sqrt{\mathrm{Var}(T)} \rceil$ computed via Theorem 5: part 2;

2: {Decide if $\mathcal{P}(X)$ is an rm-localized partition:}

2.1: **for all** $i \in \{1, 2, \ldots, m\}$

$\{1 \leq i \leq m-1$: processor P_i computes:

$is\_\mathrm{rm}Localized_i := (\gamma_+(X_{i-1,i}X_i) = \emptyset) \vee$

$(\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*) \in [\beta_i^* + 1, \beta(X_{i-1,i}X_iX_{i,i+1})])$;

$i = m$: processor P_m computes:

$is\_\mathrm{rm}Localized_m := \mathrm{true}$;$\}$

2.2: Compute $is\_\mathrm{rm}Localized := \wedge_{\eta=1}^{m-1} is\_\mathrm{rm}Localized_\eta$ using prefix-sum function;

2.3: **for all** $i \in \{1, 2, \ldots, m\}$ processor P_i updates:

$is\_\mathrm{rm}Localized_i := is\_\mathrm{rm}Localized$;

3: {If $\mathcal{P}(X)$ is rm-localized, then compute $\mathrm{Max}(X)$ via Theorem 3: determine $X'_{i,i+1}$ for all $i \in \{1, 2, \ldots, m-1\}$ and compute $\mathrm{Max}(X''_{i-1,i}X_iX'_{i,i+1})$ for all $i \in \{1, 2, \ldots, m\}$:}

for all $i \in \{1, 2, \ldots, m\}$ processor P_i updates:

if $is\_\mathrm{rm}Localized_i$ **then**

$\{1 \leq i \leq m-1$: processor P_i sends $\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*)$ to processor P_{i+1};

processor P_{i+1} receives $\mathrm{rm}_{X_{i-1,i}X_iX_{i,i+1}}(\beta_i^*)$;

$i = m$: **null**;$\}$

Invokes Max_Sequential to compute $\mathrm{Max}(X''_{i-1,i}X_iX'_{i,i+1})$;

else goto Step 4;

4: Invoke a parallel algorithm adapted from the Max-computing PRAM-algorithm [3] in which two embedded problems are solved by parallel algorithms implemented with MPI: "all nearest smaller values" [4] and "range-minima" [5];

Table 1. Preliminary empirical study of speedup and efficiency of Max_Parallel

	mean-statistics over N		conditional on "success" scenario: mean-statistics over observed N_s			unconditional: mean-statistics over N		
m	N_s	observed N_s	$T_m(n)$	$S_m(n)$	$E_m(n)$	$T_m(n)$	$S_m(n)$	$E_m(n)$
1	100.00	100	0.156833	0.9939	1.0000	0.156835	0.9939	1.0000
2	88.89	98	0.078377	1.9889	1.0005	0.078712	1.9804	0.9963
4	70.23	95	0.039663	3.9301	0.9885	0.040095	3.8878	0.9779
8	43.85	81	0.020464	7.6173	0.9580	0.021470	7.2604	0.9131
16	17.09	72	0.010410	14.9742	0.9416	0.011246	13.8610	0.8716
32	2.60	43	0.005312	29.3451	0.9226	0.006318	24.6725	0.7757
64	0.06	21	0.003002	51.9257	0.8163	0.005047	30.8859	0.4855

Our work in progress includes a comparative empirical/probabilistic study based on current implementation and refining the algorithms to detect and resolve violations of rm-locality among near-neighbor processors. There are two directions for general theoretical developments. First, the length bound of the common subsequences (to capture the rm-locality) is achieved via explicit bounds on the mean/variance of the first ladder epoch in the underlying random walk with normal distribution. This leads to a deserving study for general probability distribution. Second, there are other notions of (minimal) maximality for ranking subsequences of a real-valued sequence, developing efficient parallel algorithms for their computation is interesting.

References

1. Akl, S.G., Guenther, G.R.: Applications of Broadcasting with Selective Reduction to the Maximal Sum Subsegment Problem. International Journal of High Speed Computing **3**(2), 107–119 (1991)
2. Alves, C.E.R., Cáceres, E.N., Song, S.W.: Finding All Maximal Contiguous Subsequences of a Sequence of Numbers in $O(1)$ Communication Rounds. IEEE Transactions on Parallel and Distributed Systems **24**(3), 724–733 (2013)
3. Dai, H.-K., Su, H.-C.: A parallel algorithm for finding all successive minimal maximum subsequences. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 337–348. Springer, Heidelberg (2006)
4. He, X., Huang, C.-H.: Communication Efficient BSP Algorithm for All Nearest Smaller Values Problem. Journal of Parallel and Distributed Computing **61**(10), 1425–1438 (2001)
5. JáJá, J.: An Introduction to Parallel Algorithms. Addison-Wesley (1992)
6. Karlin, S., Brendel, V.: Chance and Statistical Significance in Protein and DNA Sequence Analysis. Science **257**(5066), 39–49 (1992)
7. Lin, T.-C., Lee, D.T.: Randomized Algorithm for the Sum Selection Problem. Theoretical Computer Science **377**(1–3), 151–156 (2007)
8. Ruzzo, W.L., Tompa, M.: A linear time algorithm for finding all maximal scoring subsequences. In: The Seventh International Conference on Intelligent Systems for Molecular Biology, pp. 234–241. International Society for Computational Biology (1999)