1 Computational Complexity

(ϵ, δ) differentially private *partition selection* is considerably more efficient and easier to implement than current state-of-the-art DP prefix-based query methods. To support large scale applications of our *TraVaS* framework, we will briefly discuss the most important complexity considerations of all associated algorithms (see our paper).

In the preprocessing phase of both the *TraVaS* core and the optimizer shell *uTraVaS*, an event log L needs to be transformed into its aggregated simple form, holding only variant-frequency pairs $(\sigma, L(\sigma))$. This operation can be described as a Group-By-Count query that consumes $\mathcal{O}(n)$ steps with n being the number of cases in L. Although large input logs may lead to runtime bottlenecks, the aggregation must only be executed once independent of further privatization processes. The *TraVaS* core then requires i.i.d *k-TSGD* random samples and variant deletion checks for all m unique variants of L. Hence, we face a separate maximum complexity of $\mathcal{O}(m)$ with each *TraVaS* run. Eventually, our extension *uTraVaS* calls u *TraVaS* instances, implicating a worst-case complexity of $\mathcal{O}(u^2 \cdot m)$ according to the *Gaussian* sum formula. In addition, the same number of steps are needed for all merging and mean operations to obtain the respective candidate logs. Together with u utility computations and the final argmax function, Algorithm 2 (*uTraVaS*) therefore runs with a cumulative complexity of $\mathcal{O}(u^2 \cdot m + n)$. Note, that the cost function might possess a larger complexity than the *TraVaS* body.

One compelling advantage of our algorithm design is that only the initial event log transformation represents a sequential task. All subsequent routines such as the sublog construction, the merging or the cost evaluation can be perfectly parallelized on multiple computing units. As a result, industrial applications may partially compensate large inputs by stronger technology backbones.

1.1 Algorithms

In the following, we list both *TraVaS* algorithms (*TraVaS* and *uTraVaS*) for the sake of completeness.

Algorithm 1: Differentially Private Trace Variant Selection (TraVaS)

```
Input: Event log $L$, DP-Parameters $(\epsilon, \delta)$
Output: $(\epsilon, \delta)$-DP log $L'$
function travas $(L, \epsilon, \delta)$
    $p = 1 - e^{-\epsilon}$                                                    // compute probability
    $k = \lceil 1/\epsilon \times \ln((e^{\epsilon} + 2\delta - 1)/(\delta(e^{\epsilon} + 1))) \rceil$    // compute threshold
    forall $(\sigma, L(\sigma)) \in L$ do
        $x_\sigma =$ rTSGD $(p, k)$                                            // generate i.i.d k-TSGD noise
        if $L(\sigma) + x_\sigma > k$ then
            add $(\sigma, L(\sigma) + x_\sigma)$ to $L'$
    return $L'$
```

Algorithm 2: Utility-Aware TraVaS (uTraVaS)

```
Input: Event log $L$, DP-Parameters $(\epsilon, \delta)$, Subquery bound $u$
Output: $(\epsilon, \delta)$-DP log $L'$
function optimize $(L, \epsilon, \delta, u)$
    forall $i \in \{1, 2, \ldots, u\}$ do
        create $i$ empty sublogs: $\{sL_1, sL_2, \ldots, sL_i\}$                    // initialize
        forall $j \in \{1, \ldots, i-1, i\}$ do
            add travas($L$, $\epsilon/i$, $\delta/i$) to $sL_j$                        // run TraVaS
        add all unique variants $\sigma$ from sublogs to a simple log $pL'$
        forall $\sigma \in pL'$ do
            forall $sL \in sublogs$ do
                if $\sigma \in sL$ then
                    $f_\sigma = sL(\sigma)$                              // get frequency from sublog
                else
                    $f_\sigma$ = freqEstimate $(\epsilon, \delta)$                        // estimate frequency
                update $pL'(\sigma)$ to $pL'(\sigma) + f_\sigma$ in $pL'$
        update $pL'(\sigma)$ to $pL'(\sigma)$ / $i$ in $pL'$                            // compute mean
        add $pL'$ to the set pool.                                         // optimization domain
    $L' = \underset{pL' \in pool}{\text{argmax}}\ \text{utility}(L, pL')$                  // optimize data utility
    return $L'$
```

1.2 Note of Authorship

This document is part of the supplementary material created for the paper ”TraVaS: Differentially Private Trace Variant Selection for Process Mining”, written by Majid Rafiei, Frederik Wangelik and Wil M.P. Van der Aalst. Please contact *frederik.wangelik@rwth-aachen.de* for further information.