(CENSUS) Guide to the Census 2018 End-to-End Test Disclosure Avoidance Algorithm and Implementation

Philip Leclerc, on behalf of the 2020 Census DAS Development Team

June 2019

Disclaimer

This technical paper is preliminary and has not undergone full Census Bureau internal peer-review. It contains no sensitive data. Comments are welcome. Please address comments to philip.leclerc@census.gov.

1 Motivation: Formal Description and Companion Guide to the TopDown Algorithm in the 2018 Census End-to-End Test

This document describes the variant of the *TopDown* algorithm implemented in the *2018 End-to-End Test Disclosure Avoidance System*, which is the test version of the not-yet-finalized *2020 Decennial Census Disclosure Avoidance System* (2020 DAS). We occasionally use the short-hand TopDown$_{\text{PL94}}$ to refer to this algorithm; note that TopDown$_{\text{PL94}}$ refers specifically to the instance of the 2020 DAS employed in the 2018 End-to-End Census Test (2018 E2E Test), and made publicly available in its official, public Census GitHub repository.[3] For the 2020 Census production run, a later version of the 2020 DAS will be used, and TopDown$_{\text{PL94}}$ will be deprecated.

This document is designed to serve as a draft guide to the algorithm used in the 2018 E2E test release. Hyperlinks are used liberally throughout, primarily to indicate relevant lines in the code release where ideas described here are implemented, but also occasionally to reference relevant supplementary papers or other material that can support the reader in understanding the design of TopDown$_{\text{PL94}}$. Reader feedback on mistakes, ambiguities, or language awkwardness is welcome. Please send `philip.leclerc@census.gov` questions or comments.

1.1 Form of Inputs and Outputs

Both inputs and outputs to TopDown$_{\text{PL94}}$ are specified by a *schema* $\mathcal{S}$, describing a set of variables and their levels and a *histogram* of counts $H : \times_{v \in \mathcal{S}} v \to \mathcal{N}$ for each record type that is possible under $\mathcal{S}$ (where $\mathcal{N} := \{0, 1, 2, \dots\}$). $H(r)$ can then be used to denote the count of records of type r in histogram H. As an example, we describe in the next section the TopDown$_{\text{PL94}}$ schema that was used for the 2018 E2E test.

Using the histogram representation is without loss of generality, since they convey information that is equivalent to micro-data: for any table of micro-data, there is an equivalent histogram, and vice versa.[1] Histograms of counts are a more convenient representation for engineering differentially private subroutines and mathematical programming models used in TopDown$_{\text{PL94}}$.

Note that in the pseudo-code below we often to suppress reference to the schema $\mathcal{S}$ to save space. We specifically mention the schema only when necessary (i.e., when discussing the detailed, internal structure of particular classes of queries).

[1] In statistics, the histogram representation is called the fully-saturated contingency table with all structural zeros removed.

1.2 Primary Target Schema: PL94-171

TopDown$_{\text{PL94}}$ was designed primarily to execute on a schema suitable for supporting the Public Law 94-171 ("redistricting") data product, although it can also be run on other data products with different schemas (e.g., the public-use, person-level 1940 Census data packaged with the TopDown$_{\text{PL94}}$ public code release).

The primary schema used to support Public Law 94-171 is:

$$\text{Block} \times \text{HHGQ} \times \text{VA} \times \text{HISP} \times \text{RACE}$$

where:

- *Block* is a dimension with $\approx 12,000$ levels (one per Census block in Providence County, RI)
- *HHGQ* has 8 levels (for the purposes of PL94-171, a person resides in a Household or in one of 7 major Group Quarters types)
- *VA* has 2 levels (Voting Age or not)
- *HISP* has 2 levels (Hispanic ethnicity, yes or no)
- *RACE* has 63 levels (one for each non-empty combination of the 6 major OMB race categories

Although it is not critical to refer to the PL94-171 schema to understand TopDown$_{\text{PL94}}$, but it is useful to consider as an example of how this algorithm would work on a similar data product from a single micro-data input table.

1.3 The Geographic Hierarchy

TopDown$_{\text{PL94}}$ operates over geographic units or "geounits" that correspond to nodes of a directed rooted tree (or "arboresence"). For the 2018 E2E Census Test, conducted in Providence, Rhode Island, this rooted tree has nodes arranged into 4 geographic levels: County, Tract, Block Group, Block, with a single County (serving as the tree's unique root node, γ_0) and about $12,000$ Census blocks (which, in the arborescence, are leaves).

Other problems may, of course, have other geographic hierarchies (as is true, for example, of the 1940 Census data provided for public testing of TopDown$_{\text{PL94}}$). We denote by Γ the generic geographic-hierarchy arboresence over which TopDown$_{\text{PL94}}$ is assumed to operate, and typically use γ or τ to refer to a single node in Γ. We use the semantically named functions children(γ) and parent(γ) to refer to the set of nodes that are children or the unique parent of γ, respectively. We also use the short-hand $c(\gamma)$ and $p(\gamma)$ where the complete labels would fail to typeset properly due to length.

When reading, using, or modifying the 2018 E2E test code, note that the geounit node class is our primary data structure for organizing and referencing data associated with specific geounits, with the attributes of this class at various times in the program's operation storing, for example, raw (unprotected/sensitive), DP (differentially private), and "syn(thetic)" (final microdata; formally private but not differentially private, as explained below) versions of the reference geounit's histogram/micro-data. These geounit nodes are the primary objects manipulated by the 2018 E2E test code: at a very high level, TopDown$_{\text{PL94}}$ can be thought of as a mechanism for taking block-level geounit nodes containing raw, sensitive-input-data, and turning them into equivalent node objects with the raw histogram/micro-data object in each node removed and replaced by a formally private equivalent.

1.4 Motivation and Notable Features

The form of TopDown$_{\text{PL94}}$ is motivated by three primary considerations:

- it must scale to accommodate tens of billions of differentially private measurements/queries on which low accuracy is desired
- it must produce an output easily transformable into micro-data (e.g., a nonnegative count for each $t \in \times_{v \in \mathcal{S}} v$)
- a set of *invariants* $\mathcal{I}$ will be provided as an additional input. Invariants are queries on which the final, protected histogram generated by TopDown$_{\text{PL94}}$ and the sensitive input histogram $\tilde{H}$ must agree.

To contend with the scale of the problem, the 2020 DAS team elected to design TopDown$_{\text{PL94}}$ to proceed top-down (as its name suggests), decomposing the problem of generating nation-wide micro-data into micro-data-generation sub-problems from "parent" geounits to "child" geounits, beginning with the National geounit γ_0. This design choice also improves the statistical efficiency of TopDown$_{\text{PL94}}$ by allowing estimation of micro-data in geounits at lower geolevels to "borrow strength" from the micro-data generated in higher geolevels; this borrowed statistical signal is especially relevant to "sampling zeros" (record types $t \in \times_{v \in \mathcal{S}} v$ but with observed count $H_\gamma(t) = 0$), which–for a given scale of noise–are easier to estimate when the underlying count of records is larger, as with geounits in higher geolevels. In the 2018 E2E test, moving top-down involved generating a histogram at the County level, then expanding this histogram to assign micro-data records to each of the Census tracts in the County, then expanding each tract-level histogram to assign records to each block group in the tract, and so on down to census blocks.

Given the decision to proceed top-down, the requirement to produce protected micro-data under a set of complex invariants then led the 2020 DAS development team to incorporate both "implied constraints" ($\mathcal{C}_\gamma$) and a "failsafe" mechanism. Implied constraints are inequalities used to constrain the micro-data-generation in each geounit, and are needed to ensure any solution derived at a non-block geographic unit (e.g., the histogram generated at the county level) can be extended to a valid representation as micro-data that satisfies the invariants at the block level. If some implied constraints are missing, then the optimization problem solved at some geographic unit γ in order to generate an estimated histogram for each element of children(γ) may be infeasible, i.e., have no solution. We expand on this idea below, providing a simple example of how this phenomenon can occur, and carefully describe the histogram/micro-data-generation optimization sub-problems.

At the time of the 2018 E2E test, the DAS development team did not have a proof that the set of implied constraints was sufficient to avoid infeasible histogram/micro-data-generation sub-problems, and so it was necessary to define a "failsafe" sub-problem as a fallback in the event that the algorithm encounters an infeasible sub-problem. The failsafe is defined forming they Lagrange-multiplier representation of the constrained optimization, converting previously hard/exact constraints into soft objective function penalty terms. The constraints ensure that child histograms sum to the previously derived parent histogram: $\sum \tau \in \text{children}(\gamma) H_\tau = H_\gamma$. This constraint relaxation was sufficient to admit a proof of feasibility and that integer-valued solutions (corresponding to a valid histogram/micro-data) would be generated reliably and rapidly. However, relaxing this constraint can harm accuracy, and so further developments were incorporated to limit the negative effects of the failsafe on accuracy:

- marginals over variables that participate only in the total population invariant, but not any of the more complex invariants, were added as constraints to the failsafe problem
- on encountering an infeasible sub-problem, a new L1 problem was created and solved to determine the minimum violation of the infeasible problem's hierarchical-summation constraints that could be achieved. The failsafe problem then incorporated a constraint requiring that solutions imply violation of hierarchical-summation no greater than this value

1.4.1 End-to-End Invariants

The declared set of invariants for the production run of the 2018 DAS include:

- The Total Population of Providence RI
- The number of housing units in each block of Providence RI
- The number of occupied housing units in each block of Providence RI
- The number of Group Quarters facility by major type in each block of Providence RI

There are a number of additional invariant properties that result logically from the declared set of invariants. For example, the number of vacant housing units in each block will be invariant. Also, Group Quarters facilities are by definition occupied so whether a block is populated or not also remains invariant. Implementation details for the invariants are discussed in Section 2.4.

1.4.2 Computational Considerations: Apache Spark

$\text{TopDown}_{\text{PL94}}$ is organized in a top-down fashion in part to address issues of computational scale. However, this structure is not sufficient in itself: looking forward to the full 2020 DAS run, we also require a framework for rapidly increasing the parallelism with which the DAS runs. To this end, the DAS is implemented using Apache Spark, a popular "bulk-synchronous parallelism" system often used when grappling with "big data" or "big compute". It is beyond the scope of the present document to describe either Spark or the DAS use of Spark in detail.The use of Spark is integral to the current DAS implementation, and the Spark Python API calls are present throughout the 2018 E2E test code. Although it is possible to rewrite the DAS so as not to use Spark, doing so would be a very time-consuming task.

Here we briefly summarize $\text{TopDown}_{\text{PL94}}$'s use of Spark: the DAS spends most of its run-time manipulating Spark Resilient Distributed Dataset (RDD) objects, and most of the DAS mathematical operations are implemented through the use of map functions that are used to produce a new RDD from a previous one. An RDD is a list object where the elements of the list are spread across multiple nodes in the cluster. The first RDD created consists of the block-level micro-data, which are then mapped into block-level histograms, producing the second RDD. Both of these RDDs contain the raw, unprotected data. Spark reduce and reduceByKey functions are used to create the top-level histogram, which is held in its own RDD. The geometric mechanism is applied to the raw data through the use of another map function to create the noisy measurements. Finally, map functions take the top-level histogram and the noisy measurements and apply the Gurobi optimizer, producing a series of RDDs corresponding to each geolevel, starting at the National level and ending at the block level. Finally, the histograms in the block-level RDD are expanded into micro-data and written out.

This concludes our narrative description of notable design features of $\text{TopDown}_{\text{PL94}}$. It is intended to supplement a careful reading of the formal description of $\text{TopDown}_{\text{PL94}}$ given below, and to help fix ideas by illustratiion with concrete examples drawn from the 2018 E2E test.

2 Description of $\text{TopDown}_{\text{PL94}}$

2.1 High-Level Description of $\text{TopDown}_{\text{PL94}}$

2.1.1 Inputs

$\text{TopDown}_{\text{PL94}}$ takes 7 primary inputs:

- a directed root tree, Γ, with root γ_0 corresponding to the uppermost geounit (County, in the 2018 E2E test; Nation, in 2020 and in the 1940 test data)
- an input histogram H representing the sensitive persons-universe input

data. H has shape $12000 \times 8 \times 2 \times 2 \times 63$ in the 2018 E2E test. We often abuse notation and often treat H as a function, where convenient, e.g., for the 2018 E2E test, $H(\gamma_0)$ returns the shape $8 \times 2 \times 2 \times 63$ histogram summarizing Providence, Rhode Island's sensitive micro-data. We use $|H(\gamma_0)|$ to denote the length of the flattened histogram, e.g., $8 \cdot 2 \cdot 2 \cdot 63$

- an input histogram U representing the sensitive units-universe input data (shape 1×8 in the 2018 E2E test)

- a description I of the desired invariants. Below, we specify this as a list of two tuples indicating invariant choices from a finite set of available options (and their lowest geolevel)

- a description C of the desired invariants

- ϵ, a function giving the privacy-loss budget (PLB) allocated to each geolevel, and – within each geounit – the split of the available PLB between the detailed queries (an identity sub-matrix of W_γ) and the distinct, user-specified, sets of marginal queries. In the 2018 E2E test code, this breakdown of the PLB is specified in the budget section of the relevant config file

- the target query workload $W \in \mathcal{R}^{k \times |H|}$, in which each row represents a target query on which low accuracy is desired, and k is a positive user-specified integer representing the number of workload queries (i.e., each row W_{i*} of W gives a workload query/tabulation, and that row's coefficients indicate how to compute W_{i*} by taking a suitable vector product with H). As with H, we use subscripts to denote useful transformations of W, e.g. W_{γ_0} denotes the $(8 \cdot 2 \cdot 2 \cdot 63)$ workload matrix in which each row represents a query for Providence, Rhode Island

These 7 parameters are fundamental for understanding TopDown$_{\text{PL94}}$. In addition, however, a large number of additional parameters are exposed for convenient alteration through configuration files (e.g., 2018 E2E test config file).

One further comment about W is appropriate: in the TopDown$_{\text{PL94}}$ code, W_γ is assumed to be identical across geounits (so the dependence on γ is unnecessary; this will not be the case in the full 202 DAS0), and within each geounit γ is partitioned into two types of rows—a set of *detailed* queries, and a set of *DPQueries*. Without loss of generality, we may treat W_γ as having the form:

$$W_\gamma = \begin{bmatrix} \text{DPQueries}_1 \\ \vdots \\ \text{DPQueries}_{k'} \\ I_{|H_\gamma|} \end{bmatrix}$$

That is, W_γ consists of $k' + 1$ vertically stacked sub-matrices: each of the first k' sub-matrices represents a distinct class of DPQueries with its own portion of this geolevel's PLB assigned to it. In the 2018 E2E test, there were two such matrices, identified by the names *hhgq* and *va_hisp_race*.

The first, "DP queries" sub-matrix has shape $k \times |H|$, and each row represents a query manually selected to target for low error; these queries are specified by names in the configuration file which are converted into corresponding Python objects created in the queries module. The second, "detailed queries" sub-matrix is an $|H| \times |H|$ identity matrix, which is always present to ensure that, as $\epsilon \to \infty$, the output of $\text{TopDown}_{\text{PL94}}$ will correspond to the input H with probability 1.

To express dependence on a specific part of the geograhic hierarchy, or to emphasize that they can be treated as functions with domain over the entirety of Γ, we sometimes refer to $W(\gamma)$ or $W(\Gamma)$, $H(\gamma)$ or $H(\Gamma)$, $\mathcal{M}(\text{children}(\gamma))$ and so on.

2.1.2 Organization into Major Phases

A *Driver* program serves as the main entry point for $\text{TopDown}_{\text{PL94}}$. *Driver* takes as inputs H, ϵ, and W. The DAS then proceeds in 3 major phases:

- Phase 0
  - Phase 0.1: calculate invariants $\mathcal{I}(\gamma)$ for each $\gamma \in \text{leaves}(\Gamma)$
  - Phase 0.2: calculate implied constraints $\mathcal{C}(\gamma)$ for each geounit $\gamma \in \text{leaves}(\Gamma)$
  - Phase 0.3: for each $\gamma \in \Gamma\backslash\text{leaves}(\gamma)$, compute $\mathcal{C}(\gamma) = \sum_{\tau\in\text{leaves}(\gamma)} \mathcal{C}(\tau)$
- Phase 1: for each $\gamma \in \Gamma$ and $w \in \text{rows}(W)$, apply Geometric Mechanism to sample an estimate $\tilde{\mathcal{M}}(w)_\gamma$
- Phase 2: post-process to produce block-level microdata, proceeding top-down

Note that the sensitive underlying data is only accessed in Phases 0–1; Phase 1 is thus the noise barrier. Phase 2 is pure post-processing, as its only connection to the sensitive data is through the DP measurements $\tilde{\mathcal{M}}$ and constraints $\mathcal{C}$, which it receives as inputs. We describe these phases more precisely below (see pseudo-code and formal subroutine / sub-problem descriptions).

Specification of the invariants $\mathcal{I}$ and implied constraints $\mathcal{C}$ is in the configuration file. For an example, see the 2018 E2E test configuration file, line 81. As that line in the configuration file suggests, the constraints and invariants modules then contain the code responsible for calculating specific invariants and building the set of implied constraints requested in the configuration file.

This concludes our narrative overview of $\text{TopDown}_{\text{PL94}}$. We next provide a more formal, detailed description of its operation.

2.2 Privacy Properties and Allocation of the PLB

For the DP sub-routines used to take measurements $\tilde{\mathcal{M}}_\gamma$, we used the common variant of DP known as *bounded differential privacy*[2]: a randomized algorithm M is said to be τ-bounded DP if, for any two databases D, D' that differ in the modification of a single record, and any set of possible outputs $\mathcal{O}$, we have $Pr[M(D) \in \mathcal{O}] \leq e^\tau Pr[M(D') \in \mathcal{O}]$.[2]

Bounded DP is weaker than unbounded DP, the most commonly used form of DP, because bounded DP makes no attempt to protect the grand-total count of persons in the population (or sample, for surveys). For the 2020 Census, we have operated under the assumption that the count of the Resident Population of the United States will be unprotected (not perturbed by the DP system), thus bounded DP is the appropriate theoretical primitive.[3]

Given a total PLB of ϵ, this PLB is allocated in $\text{TopDown}_{\text{PL94}}$ according to calculations specified in the budget section of a configuration file. In the 2018 E2E test, the total PLB of $\epsilon = 0.25$ was first divided equally across each of the county, tract, block group, and block geolevels, so that $\epsilon_g = \frac{0.25}{4}$ was assigned to each geolevel g.[4] This division is justified by invoking *sequential composition*: for any two τ-bounded DP mechanisms, their simultaneous release cannot be worse than 2τ-bounded DP.

Within each geolevel, we further invoke sequential composition to divide the $\frac{0.25}{4}$ PLB available between the detailed queries and each set of DPQueries (considered as queries over all geounits in the geolevel) as specified in the budget section of the config file. In the 2018 E2E test, 10% of the per-geounit PLB was assigned to the detailed queries, 22.5% PLB to the *hhgq* DPQuery, and 67.5% to *va_hisp_race*.

Lastly, we calculated the *global sensitivity* for use with the Geometric Mechanism, which is a primitive DP mechanism used to generate estimated counts for each DPQuery and the Detailed Queries. Global sensitivity is defined for a query q by $\Delta(q) = \sup_{D,D':\text{D is a neighbor of D'}} |q(D) - q(D')|$. It is important

[2] Here we use τ here rather than the more common ϵ to avoid confusing descriptions of generic, non-specific DP mechanisms with the specific PLB ϵ used by $\text{TopDown}_{\text{PL94}}$.

[3] Note, however, that the 2018 E2E test DAS is not—and the 2020 DAS will not be—end-to-end ϵ-bounded DP. The DPMeasurements module of the DAS is ϵ-bounded DP; however, the full system cannot be DP, because of the requirement that invariants be imposed on the final micro-data; the treatment of invariants, for which no noise can be infused, is fundamentally non-DP. Systems with a modest set of precisely specified invariants and an ϵ-bounded DP subroutine for taking noisy measurements—such as the DAS—*can* be shown to provide reasonably general, rigorously provable, strong privacy guarantees, but they do so for a more limited range of possible attacker inferences than are typically protected by DP (though this is a high bar, as DP typically provides protection against *arbitrary* attacker inference). Exploring this issue in detail is beyond the scope of this document, but the reader is encouraged to contact the DAS development team authors if it is of interest, as a separate paper is in preparation explaining the nature of these somewhat truncated but deductively precise, general, DP-like guarantees.

[4] A careful reader of the code will note that the global PLB was actually set to 0.2499, not 0.25. We ignore this in our exposition, but note that the small 0.0001 share of the PLB missing was reserved to implement a simple confidence-interval-based, probabilistic check that non-zero noise had been infused into the final micro-data (a kind of sanity check that neither person nor error had "turned off' noise infusion).

because infusing additive, two-tailed geometric noise with scale parameter $\frac{\Delta(q)}{\tau}$ is sufficient to achieve τ-bounded DP. For each of the detailed queries, *hhgq*, and *va_hisp_race*, considered as a query across all geounits in a fixed geolevel, has global sensitivity of 2: modifying a record can change the output counts by 2 if a record is modified to change one geounit in the geolevel to a different geounit in that geolevel, and this is the worst possible change. We therefore set global sensitivity to 2, and used this sensitivity parameter when taking measurements in the TopDown$_{\text{PL94}}$ engine code.

2.3 TopDown$_{\text{PL94}}$ Pseudo-code

TopDown$_{\text{PL94}}$ is described by the following pseudo-code, with specific subroutines and optimization problems described precisely in the following subsections.

```
function Driver($\Gamma, H, U, I, C, \epsilon, Workload$):
    $\mathcal{I} \leftarrow$ Invariants($I, H, U$)
    $\mathcal{C} \leftarrow$ Constraints($C, \Gamma, \mathcal{I}$)
    $\tilde{\mathcal{M}} \leftarrow$ DPMeasurements($\Gamma, H, \epsilon, W$)
    $\widetilde{H} \leftarrow$ TopDownPostprocess($\Gamma, \mathcal{C}, \tilde{\mathcal{M}}, W$)
    return $\widetilde{H}$
function TopDownPostprocess($\Gamma, \mathcal{C}, \tilde{\mathcal{M}}, W$):
    node-queue $\leftarrow \emptyset$                    // List of processed nodes
    /* Generate root-node histogram $\widetilde{H}^0$                         */
    $H^* \leftarrow$ $NationalL2(\mathcal{C}(\gamma_0), \mathcal{M}(\tilde{\gamma_0}), W(\gamma_0))$
    $\widetilde{H}^0 \leftarrow$ $NationalL1(\mathcal{C}(\gamma_0), \mathcal{M}(\tilde{\gamma_0}), W(\gamma_0), H^*)$
    /* Recurse down the hierarchy                                 */
    node-queue.append(root node $\gamma_0$)
    while node-queue is not empty do
        $\gamma \leftarrow$node-queue.pop()
        $i \leftarrow$level($\gamma$) $m \leftarrow |c(\gamma)|$
        $\gamma_1, \ldots, \gamma_m \leftarrow c(\gamma)$
        /* Generate child histograms $\widetilde{H}_\ell^{i+1}$                    */
        $H^*_{c(\gamma)} \leftarrow$ $PrimaryL2(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W(c(\gamma)), \tilde{H}_\gamma)$
        if $H^*_\alpha \neq \emptyset \forall \alpha \in c(\gamma)$ then
            $\widetilde{H}^{i+1}_{c(\gamma)} \leftarrow$ $PrimaryL1(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W(c(\gamma)), H^*_{c(\gamma)})$
        else
            $D \leftarrow$ $FeasL2(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W, \tilde{H}_\gamma)$
            $H^*_1, \ldots, H^*_m \leftarrow$ $FailsafeL2(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W, \tilde{H}_\gamma, D)$
            $D \leftarrow$ $FeasL1(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W, H^*_1, \ldots, H^*_m)$
            $\widetilde{H}^{i+1}_{\gamma_1}, \ldots, \widetilde{H}^{i+1}_{\gamma_m} \leftarrow$
              $FailsafeL1(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W, H^*_1, \ldots, H^*_m, D)$
        end
        for each $\gamma_j$ do
            if $\gamma_j$ has children then
                node-queue.append($\gamma_j$)
            end
        end
    end
    Concatenate the leaf histograms ($\widetilde{H}^k_\gamma$ for $\gamma \in$ leaves($\Gamma$)) into the single
      histogram $\widetilde{H}$
    return $\widetilde{H}$
```

Algorithm 1: Census TopDown Algorithm

Within the TopDown pseudocode, 3 notable subroutines and 8 optimization sub-problems are used. We describe these below.

2.4 Invariants

2.4.1 Informal Description

Invariants(I, H, U) takes a description I of the desired invariants, the sensitive-input-data persons-universe histogram H, the sensitive-input-data units-universe histogram U, and outputs an invariant function $\mathcal{I} : \Gamma \times I[0] \to \cup_{k \in \{1,2,\ldots\}} \mathcal{P}(\mathcal{R}^k)$.

4 invariants are available in the publicly released 2018 E2E test code:

- `tot`: total population
- `gqhh_vect` $\to$ (#Occupied Households, Group Quarters Major Type $1, \ldots,$ Group Quarters Major Type 7)
- `gqhh_tot` : total number of Occupied Households & (by-definition-occupied) Group Quarters facilities by major type
- `gq_vect` : total number of (by-definition-occupied) Group Quarters facilities by major type

Invariants calculates and returns the requested invariants only at the geographic levels specified in the configuration file and above. The E2E config file specifies, for example,

```
theInvariants.Block: gqhh_vect,gqhh_tot
theInvariants.County: tot
```

indicating that in the 2018 E2E test

$$\begin{aligned} \text{gqhh\_vect} = U = & (\text{\#Occupied Households, Group Quarters Major Type } 1, \ldots, \\ & \text{Group Quarters Major Type } 7) \end{aligned}$$

and $\sum_{i=0}^{7} u_i$ (Units *gqhh_tot*) were calculated for every geographic unit in Providence, Rhode Island (i.e., from the block level up), while total population (persons *tot*) was calculated only at the County level (i.e., for the single county, Providence, in the 2018 E2E test).

Note that in the 2018 E2E test code, outside of the primary engine and reader, U was actually of length 9. The 9th element contained a count of vacant households. This count was held invariant in the 2018 E2E test, and was not needed for calculation of primary tables or measurements, so it was removed from U and "passed through" unaltered outside of the primary engine code. For the purposes of this document, we ignore this complication, and assume U is of length 8 when concrete examples are required.

2.4.2 Formal Description

I is assumed to be of the form

$[(\text{invariant type string1}, \ldots, \text{invariant type string}k), (\text{geoevel string1}, \ldots, \text{geolevel string}k)]$

for $k \in \mathcal{N}_+$.

We further assume the availability of a function

$$\text{getGeolevel} : \text{nodes}(\Gamma) \rightarrow \{\text{Nation}, \text{State}, \text{County}, \text{Tract}, \text{Blockgroup}, \text{Block}\}$$

mapping nodes to geolevel strings (e.g., getGeolevel(γ_0) = County).

U_γ **for** $\gamma \in$ **nodes**(Γ) **is assumed to be of the form:**

$$(\#\text{Occupied Households}, \text{Group Quarters Major Type 1}, \ldots, \text{Group Quarters Major Type 7})$$

Invariants(I, H, U) **then implements the pseudocode:**

```
function Invariants($I, H, U$):
    $\mathcal{I} \leftarrow \{\}$                    // Fxn:  $\Gamma \times I[0] \rightarrow \cup_{k \in \{1,2,\dots\}} \mathcal{P}(\mathcal{R}^k)$
    for $invariant, level \in zip(I[0], I[1])$ do
        $Node\text{-}queue \leftarrow \text{getGeolevel}^{-1}(\text{level})$
        while $Node\text{-}queue$ is not empty do
            $\gamma \leftarrow$Node-queue.pop()
            if $\gamma \in getGeolevel^{-1}(level)$              // $\gamma$ in starting level
             then
                $\text{tot} \in I[0] \Rightarrow \mathcal{I}(\gamma, \text{tot}) \leftarrow \sum_{s \in \mathcal{S}(H)} H_\gamma(s)$
                $\text{gqhh\_vect} \in I[0] \Rightarrow \mathcal{I}(\gamma, \text{gqhh\_vect}) \leftarrow U_\gamma$
                $\text{gqhh\_tot} \in I[0] \Rightarrow \mathcal{I}(\gamma, \text{gqhh\_tot}) \leftarrow \sum_{s \in \mathcal{S}(U)} U_\gamma(s)$
                $\text{gq\_vect} \in I[0] \Rightarrow \mathcal{I}(\gamma, \text{gq\_vect}) \leftarrow [U_\gamma(s) : \text{s is a gq}]$
            end
            else
                $\mathcal{I}(\gamma, \text{invariant}) \leftarrow \sum_{c \in \text{children}(\gamma)} \mathcal{I}(c, \text{invariant})$
            end
            if $parent(\gamma) \notin Node\text{-}queue$ then
                Node-queue.push(parent($\gamma$))
            end
        end
    end
    return $\mathcal{I}$
```

Algorithm 2: Invariants Subroutine

where we use the short-hand predicate ⇒ implication to indicate that if predicate holds, then we carry out implication (i.e., a one-line if-then).

2.5 Constraints

2.5.1 Informal Description

`Constraints`$(\Gamma, \mathcal{I})$ takes the geolattice Γ, calculated invariants $\mathcal{I}$, and generates constraints that will be directly enforced in the optimization problems solved during $\text{TopDown}_{\text{PL94}}$. These constraints accomplish two goals:

- Ensuring the invariants $\mathcal{I}(\gamma, *)$ hold for each $\gamma \in \Gamma$
- Ensuring that combinations of variables that occur according to the schema but are known *a priori* to be impossible—so-called "structural zeros"—do not occur in the output histograms/micro-data produced by $\text{TopDown}_{\text{PL94}}$
- Reducing the invocation rate of the failsafe, which is called when an infeasible optimization problem is encountered during $\text{TopDown}_{\text{PL94}}$

Directly enforcing the invariants in γ is straightforward and obviously necessary to ensure that micro-data can be successfully generated down to leaves(Γ). In addition, for PL94, a single "structural zero" was present: a record type $s \in \mathcal{S}(H)$ for which we know *a priori* that we must have $\tilde{H}_\gamma(s) = H_\gamma(s) = 0 \forall \gamma \in \Gamma$.

Less obviously, enforcing $\mathcal{I}(\gamma, *)$ in each γ is not sufficient to ensure that TopDown$_{\mathrm{PL94}}$ will terminate successfully in the absence of the failsafe. Because of this, additional "implied constraints" that are not simple sums of invariants computed at the lowest required level are also useful, as they help to ensure that solutions generated intermediately in TopDown$_{\mathrm{PL94}}$ can be extended to the block level without invoking the failsafe.[5]

6 classes of constraints are available in the publicly released 2018 E2E test code:

- `total`: total population equality constraint
- `hhgq_total_lb`: requirement that at least as many persons be in GQs of each type as there are facilities of that type in each geounit (i.e., there must be enough people in each GQ type to assign at least one person to each facility of that type)
- `hhgq_total_ub`: requirement that no more persons be in GQs of each type than (total population − #persons in other GQ types)
- `union_hhgq_lb`: for each subset S of `gqhh_vect`, the number of persons residing in units of a type in S must be at least `hhgq_total_lb`$_\gamma$, or must be `total`$_\gamma$ if S is a superset of the unit types with non-zero counts in γ
- `union_hhgq_ub`: for each subset S of `gqhh_vect`, the number of persons residing in units of a type in S must be no more than `total` less the number of units in gqhh_vect $\setminus S$
- `nurse_nva_0`: structural zero (by definition, no persons under 18 years of age may reside in "Nursing facilities/Skilled-nursing facilities", GQs of type 301)

Note that `union_hhgq_lb` and `union_hhgq_ub` encode a kind of non-linearity, and it is from this non-linearity that the need for "implied constraints" stems. Specifically, the number of persons required to be assigned to a unit type in a subset of unit types S increases linearly with the number of such units observed with a type in S, unless S contains all unit types in a geounit, in which case S must contain everyone in the geounit. We provide a brief example to illustrate this point.

"Implied Constraints" Example

[5] Avoiding the failsafe is, in turn, important because the failsafe tends to be less statistically efficient than the primary TopDown$_{\mathrm{PL94}}$ solves. This property is a result of the failsafe relaxing the requirement that the DP histogram $\tilde{H}$ representing micro-data for $c \in$ children(γ) satisfy $\tilde{H}(\gamma) = \sum_{c \in \mathrm{children}(\gamma)} \tilde{H}(c)$.

Suppose we have a block group composed of two blocks, with the first block containing 100 persons all living in 1 GQ of type A and 1 GQ of type B (and no other units), and the second block containing 100 persons all living in 1 GQ of type C. The naïve block group-level invariants would then require that all 200 persons reside in a GQ of type A, B, or C, with at least 1 person in each type.

`union_hhgq_lb` then identifies an extra implication the naïve approach misses: since only GQs of type A, B are in block 1, and only GQs of type C are in block 2, it follows that at least 100 persons must reside in GQs of types A or B in the block group, and at least 100 persons must reside in GQs of type C in the block group. Without the "implied constraints" `union_hhgq_lb`, this information would be missed, and the failsafe would as a result be invoked if a block-group-level solution were estimated by TopDown$_{\text{PL94}}$ with fewer than 100 persons in the block group to A, B, or fewer than 100 persons in the block group to C.

Note that, at the time of the End-to-End test, the 2020 DAS development team believed but did not have a formal proof that the "implied constraints" `union_hhgq_lb` and `union_hhgq_ub` together were sufficient to ensure successful termination of TopDown$_{\text{PL94}}$ without the failsafe (and, even were this not the case, the all-possible-subsets character of these two sets of "implied constraints" can become a binding computational constraint on larger problem instances than PL94). Given the computational cost, we did not use `union_hhgq_lb` and `union_hhgq_ub` in the 2018 E2E run. We retained the failsafe as part of the 2018 E2E run & code release.[6]

[6] Of course, there is little reason to remove the failsafe even if a proof is available that the primary mechanism will never fail. It may still in that case be preserved as a kind of defense against human error in checking of mathematical proofs, or that implementations correspond as expected to the abstract idealizations used in proofs.

2.5.2 Formal Description

C is assumed to be of the form

$$[(\text{constraint type string1}, \ldots, \text{constraint type string}k), (\text{geoevel string1}, \ldots, \text{geolevel string}k)]$$

for $k \in \mathcal{N}_+$.

For each $(\text{constraint type string}, \text{geolevel string}) \in \text{zip}(C[0], C[1])$ and each $\gamma \in \Gamma$ in geolevel string or higher, $\text{Constraints}(C, \Gamma, \mathcal{I})$ is responsible for constructing a constraint object of the form $(A, \text{rhs}, \text{sign})$. This constraint is intended to express that

$$A\tilde{\mathcal{H}}_\gamma \text{ sign rhs}$$

should hold for the histogram/microdata $\tilde{\mathcal{H}}_\gamma$ generated for geounit γ, where H_γ is assumed to be flattened in a canonical fashion, $\text{sign} \in \{\leq, \geq, =\}$, and $A\tilde{\mathcal{H}}_\gamma$ denotes left-matrix-multiplication by A.[7] We assume the availability of a function buildConstraint which maps pairs like $(\text{constraint type string}, \text{geolevel string}) \in \text{zip}(C[0], C[1])$ to suitable triplets $(A, \text{rhs}, \text{sign})$. In the public 2018 E2E test code release, the methods of the *ConstraintsCreatorPL94* class are responsible for constructing the constraint objects, i.e. objects that behave like the triplets $(A, \text{rhs}, \text{sign})$.

***Constraints*$(C, \Gamma, \mathcal{I})$ implements the pseudocode:**

[7] As we discuss later, the mathematical optimization sub-problem solved at parent(γ) will then be responsible for incorporating this constraint, which—due to the need to achieve integer-valued histograms—is not quite as straightforward as just directly adding the requested constraint to the model.

```
function Constraints($C, \Gamma, \mathcal{I}$):
    $\mathcal{C} \leftarrow \{\}$
    for $constraint, level \in zip(I[0], I[1])$ do
        $Node\text{-}queue \leftarrow \text{getGeolevel}^{-1}(\text{level})$
        while $Node\text{-}queue$ is not empty do
            $\gamma \leftarrow$node-queue.pop()
            if $\gamma \in getGeolevel^{-1}(level)$            // $\gamma$ in starting level
             then
                $\mathcal{C}(\gamma, \text{constraint}) \leftarrow \text{buildConstraint(constraint, level)}$
            end
            else
                $\mathcal{C}(\gamma, \text{constraint}) \leftarrow \sum_{c \in \text{children}(\gamma)} \mathcal{I}(\gamma, \text{constraint})$
            end
            if $parent(\gamma) \notin Node\text{-}queue$ then
                Node-queue.push(parent($\gamma$))
            end
        end
    end
    return $\mathcal{C}$
```

Algorithm 3: Invariants Subroutine

2.6 DPMeasurements

DPMeasurements(Γ, H, ϵ, W) takes as inputs the geolattice arborescence γ, the persons-universe sensitive-input-data histogram H, a description ϵ of the total PLB and its share allocated for the different geolevels, detail queries, and each class of DPQueries, and the workload, W. In the case of the 2018 E2E test, the total PLB was set to $\epsilon = 0.25$, and evenly split to allocate $\frac{0.25}{4}$ PLB to each of the county, tract, block group, and block levels. The $\frac{0.25}{4}$ PLB available per level was then split into three parts, with $0.1 \cdot \frac{0.25}{4}, 0.225 \cdot \frac{0.25}{4}, 0.675 \cdot \frac{0.25}{4}$ allocated to the Detail Queries, the *hhgq* marginal, and the *va_hisp_race* marginal, respectively.

With these budget settings, DPMeasurements applied the standard Geometric Mechanism to each user/config-specified query separately, in each geounit, with scale parameter $\frac{\text{Sensitivity}}{\text{shareOfPLB}}$, with Sensitivity = 2 throughout the 2018 E2E test code. Mathematically, to achieve τ-DP with the Geometric Mechanism[8] and get an estimate for the query corresponding to each row of a matrix Q, we do

$$\tilde{\mathcal{M}_\gamma}(H_\gamma) = Q_\gamma H_\gamma + \text{TwoGeo}(\frac{\Delta(Q)}{\tau})$$

where $\Delta(Q)$ denotes the global sensitivity of Q, and TwoGeo$(\frac{\Delta(Q)}{\tau})$ is a length-$|\text{rows}(Q)|$ vector of i.i.d. random draws from the two-tailed geometric distribu-

[8] For bounded or unbounded DP; the only difference between the cases is the manner in which global sensitivity is calculated.

tion, which has probability mass function

$$Pr[K = k] = \frac{1-\alpha}{1+\alpha}\alpha^{-|k|}$$

where $\alpha = e^{-\frac{\Delta(Q)}{\tau}}$. In applying the Geometric Mechanism in practice, we apply it separately to the detailed queries and each of the classes of DPQueries specified, in each geounit.

In the 2018 E2E test code, the implementation of the Geometric Mechanism can be found in the primitives module, and its use to take DPMeasurements in the primary TopDown$_{\mathrm{PL94}}$ "engine" can be found in the noisyAnswers method of TopDown_Engine, which in turn invokes the make_dp_node function to create a differentially private version of each geounit "node" object, with the noise infusion occurring specifically in line 669 for the DPqueries (the marginal queries) and in line 678 for the detail queries.

The object $\tilde{\mathcal{M}}$ returned by *Measurements* can primarily be thought of as a function from Γ to vectors with length $|rows(W)|$, each component representing the estimated measurement value a single row (query) in the workload matrix W.

Looking forward to future improvements in the 2020 Disclosure Avoidance System, note that current internal builds of the DAS code-base do not use the Geometric Mechanism to directly estimate the values of user-specified queries. Instead, current internal code takes the user-specified workload matrix W of all queries on which low error is desired and uses the high-dimensional matrix mechanism to try to find an alternative set of queries to answer with the Geometric Mechanism, from which the original queries' answers can be derived and which will have optimal mean-square-error among all sets of queries that can be used to estimate the original queries' answers.[1] Although HDMM is not present in the 2018 E2E test code base, the user should anticipate that HDMM will likely be used in the 2020 production run of the DAS.

2.7 National L2

"National L2"[9], written

$$NationalL2(\mathcal{C}(\gamma_0), \tilde{\mathcal{M}(\gamma_0)}, W(\gamma_0))$$

[9] The naming convention here may be somewhat confusing, since the 2018 E2E test begins from the county level, while the "national" designation is a reflection of the intention to use an improved version of TopDown$_{\mathrm{PL94}}$ for the full production run in the 2020 Census, at which point the resident United States population (nation) will be the geolattice arborescence's root node. If it is useful to the reader, the "national" L2 and L1 problems may alternatively be thought of as "standalone" or "node-to-node" problems, in that their distinguishing feature is that they work over a single geounit node, not over a geounit node and its children. We retain the "national" naming convention here because the 2018 E2E test code also uses this naming approach, with the suffix "nat" often distinguishing methods or classes used for standalone problems.

to explicitly indicate the inputs needed to construct it, is the very first optimization problem solved. It is the linearly constrained non-negative least squares problem:

$$\begin{aligned}
&\arg\min_{H^*} \sum_{i\in|\text{rows}(W(\gamma_0))|} ||W_{i,\star}(\gamma_0)(H^*) - \tilde{\mathcal{M}}(\gamma_0)_i||_2^2 && (1)\\
&\quad \text{s.t.} && (2)\\
&\qquad H^* \geq 0 &&\\
&\qquad AH^* \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\gamma_0) && (3)\\
&\qquad H^*(s) \in \mathbb{R} \forall s \in \times_{v\in\mathcal{S}} v && (4)
\end{aligned}$$

where we interpret $\mathcal{C}(\gamma_0)$ as $\{(A, \text{sign}, \text{rhs}) : \exists \text{ constraint s.t. domain}(\mathcal{C})(\gamma_0, \text{constraint}) = (A, \text{sign}, \text{rhs})\}$.

NationalL2 is the simplest of the problems we solve:

- it is necessarily integer-feasible (summing over all census blocks for the sensitive data yields a histogram that satisfies all of *NationalL2*'s constraints)
- its solution is only required to be real-valued
- it generates data only for the single node γ_0, which has no parent node, so hierarchical-summation constraints are unnecessary, and variables/constraints are only necessary for representing a single geounit's histogram/microdata and associated constraints

In the public 2018 E2E test code, *NationalL2* is constructed and solved by the l2geoimp function, invoked with *None* parent argument and *backup_feas=False*.

2.8 National L1

"National L1" is the second optimization problem solved. Emphasizing its arguments, we write it as:

$$\textit{NationalL1}(\mathcal{C}(\gamma_0), \tilde{\mathcal{M}(\gamma_0)}, W(\gamma_0), H^*)$$

The purpose of *NationalL1* is to convert the floating-point/fractionally-valued histogram H^* generated by *NationalL2* into a nearby integer-valued solution by systematically rounding fractional parts of the H^* solution to either 0 or 1, while still maintaining the required constraints and still attempting to approximately match the noisy measurements. *NationalL1* is the mixed-integer linear programming problem:

$$\widetilde{H}^0 = \arg\min_{H^\dagger} -(H^\dagger - \lfloor H^* \rfloor) \cdot (H^* - \lfloor H^* \rfloor) \tag{5}$$

$$\begin{aligned}
\text{s.t. } & H^\dagger \geq 0 \text{ (nonnegativity)} \\
& \sum_x H^\dagger[x] = \sum_x H^*[x] \text{ (total sum constraint)} \\
& |H^\dagger[x] - H^*[x]| \leq 1 \text{ for all cells } x \\
& AH^\dagger \text{ sign rhs } \forall(A, \text{sign}, \text{rhs}) \in \mathcal{C}(\gamma_0) && (6) \\
& \forall x : \quad H^\dagger[x] \in \{0, 1, 2, \dots\} && (7)
\end{aligned}$$

In the 2018 E2E test code, *NationalL1* is constructed and solved by the geoimp_round function, invoked with *None* parent argument and *backup_feas=False*. There, it is expressed somewhat differently from the mathematical formulation above, with the optimization performed over "rounded fractional part" variables constrained to $\{0, 1\}$ rather than over "count" variables constrained to $\{0, 1, \dots\}$. It should be relatively easy to see that this is equivalent to the formulation in 7, since $|H^\dagger[x] - H^*[x]| \leq 1$ implies that $H^\dagger[x]$ must result from rounding $H^*[x]$ either towards or away from zero.[10]

2.9 Primary L2

In each sub-national but super-block geographic unit, "Primary L2" is the first optimization problem solved. Emphasizing its arguments, we write it as

$$\mathit{PrimaryL2}(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W(c(\gamma)), \tilde{H}_\gamma)$$

where $c(\gamma)$ is short-hand for the set of geounit nodes that are children of γ, $\mathcal{C}(c(\gamma))$ is an abuse of notation used to express that we require the set of constraints for each geounit node that is a child of γ, and so on.

PrimaryL2 is the linearly constrained non-negative least squares problem:

$$\begin{aligned}
\arg \min_{H^*_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} \sum_{i \in |\text{rows}(W(\alpha))|} & ||W_{i,\star}(\alpha)(H^*_\alpha) - \tilde{\mathcal{M}}(\alpha)_i||_2^2 && (8) \\
\text{s.t.} & && (9) \\
& H^*_\alpha \geq 0 \quad \forall \alpha \in c(\gamma) \\
& AH^*_\alpha \text{ sign rhs } \forall(A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha), \forall \alpha \in c(\gamma) && (10) \\
& \sum_{\alpha \in c(\gamma)} H^*_\alpha = \tilde{H}_\gamma && (11) \\
& H^*_\alpha(s) \in \mathbb{R} \forall s \in \times_{v \in \mathcal{S}} v, \forall \alpha \in c(\gamma) && (12)
\end{aligned}$$

[10] As a technical aside, the explicit requirement that *gurobi* return binary values (rather than real values in $[0, 1]$) in the 2018 E2E test code is somewhat dangerous, as it could lead to solving large, genuine mixed-integer linear programming models, which in the worst-case may solve in super-polynomial time. In practice this was not an issue for the 2018 E2E test development or production runs of $\text{TopDown}_{\text{PL94}}$, but it is likely that the requirement for binary variables will be relaxed to requiring continuous variables in the unit interval, as the system is refined.

When *gurobi* has completed solving *PrimaryL2*, either an infeasibility exception will be raised if the problem has no solution consistent with the constraints, or the returned histogram objects $H^*_\alpha, \alpha \in c(\gamma)$ will satisfy all invariants and be interpretable as micro-data records assigned from geounit γ to each of its child nodes $\alpha \in c(\gamma)$. In our pseudo-code, we assume that, if infeasibility is encountered, then *PrimaryL2* returns $|c(\gamma)|$ copies of $\emptyset$.

In the public 2018 E2E test code, *PrimaryL2* is constructed and solved by the l2geoimp function, invoked with non-*None* parent argument and *backup_feas=False*.

2.10 Primary L1

In each sub-national, super-block geographic unit, when "Primary L2" does not encounter an infeasibility (and so does not return $\emptyset$), "Primary L1" is the second optimization problem solved. Emphasizing its arguments, we write it as

$$\mathit{PrimaryL1}(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W(c(\gamma)), H^*_{c(\gamma)})$$

It is the mixed-integer linear programming problem:

$$\begin{aligned}
\widetilde{H}^0 = \arg \min_{H^\dagger_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} -(H^\dagger_\alpha - \lfloor H^*_\alpha \rfloor) \cdot (H^*_\alpha - \lfloor H^*_\alpha \rfloor) \qquad (13) \\
\text{s.t. } H^\dagger_\alpha \geq 0 \forall \alpha \in c(\gamma) \text{ (nonnegativity)} \\
\sum_x H^\dagger_\alpha[x] = \sum_x H^*_\alpha[x] \forall \alpha \in c(\gamma) \text{ (total sum constraint)} \\
|H^\dagger_\alpha[x] - H^*_{\alpha|}[x]| \leq 1 \forall \alpha \in c(\gamma), \text{ for all cells } x \\
AH^\dagger_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha) \forall \alpha \in c(\gamma) \qquad (14) \\
\forall x : \quad H^\dagger_\alpha[x] \in \{0, 1, 2, \dots\} \forall \alpha \in c(\gamma) \qquad (15)
\end{aligned}$$

In the 2018 E2E test code, *PrimaryL1* is constructed and solved by the geoimp_round function, invoked with non-*None* parent argument and *backup_feas=False*.

2.11 Failsafe-Feasibility L2

In each sub-national, super-block geographic unit, "Failsafe-Feasibility L2" is solved after Primary L2, if the Primary L2 solve failed (i.e., an infeasibility exception was raised). Emphasizing its arguments, we write it as

$$\mathit{FeasL2}(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W(c(\gamma)), \tilde{H}_\gamma)$$

FeasL2's role is to determine how closely the children-sum-to-parent constraints can be satisfied while maintaining feasibility, and is equivalent to the mixed-integer linear programming problem:

$$\min_{H^*_\alpha, \alpha \in c(\gamma)} ||\tilde{H}_\gamma - \sum_{\alpha \in c(\gamma)} H^*_\alpha||_1 \qquad (16)$$

$$\text{s.t.} \tag{17}$$

$$H^*_\alpha \geq 0 \quad \forall \alpha \in c(\gamma)$$

$$AH^*_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha), \forall \alpha \in c(\gamma) \tag{18}$$

$$H^*_\alpha(s) \in \mathbb{R} \forall s \in \times_{v \in \mathcal{S}} v, \forall \alpha \in c(\gamma) \tag{19}$$

The output of this sub-problem is a minimum achievable distance D from which hierarchical consistency (the requirement that child nodes sum to their parent node) can be achieved while maintaining feasibility of *PrimaryL2*. This D value is then re-used to build a constraint requiring that *FailsafeL2* not deviate from hierarchical consistency more than is necessary.[11]

In the 2018 E2E test code, *FeasL2* is constructed by the *L2geoimp* function, invoked with non-*None* parent argument and *backup_feas=True*, and is solved in the same function just prior to when the "main" microdata-generating solve ordinarily occurs. *FeasL2* is constructed and solved in the same invocation of *L2geoimp* that builds and solves *FailsafeL2*, which we describe momentarily.

2.12 Failsafe L2

In each sub-national, super-block geographic unit, "Failsafe L2" is solved if the Primary L2 solve fails. Emphasizing its arguments, we write it as

$$\mathit{FailsafeL2}(\mathcal{C}(c(\gamma)), \tilde{\mathcal{M}}(c(\gamma)), W(c(\gamma)), \tilde{H}_\gamma, D)$$

It is the linearly constrained non-negative least squares problem:

$$\arg \min_{H^*_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} \sum_{i \in |\text{rows}(W(\alpha))|} ||W_{i,\star}(\alpha)(H^*_\alpha) - \tilde{\mathcal{M}}(\alpha)_i||_2^2 \tag{20}$$

$$\text{s.t.} \tag{21}$$

$$\left|\left|\tilde{H}_\gamma - \sum_{\alpha \in c(\gamma)} H^*_\alpha\right|\right|_1 \leq D + 1 \tag{22}$$

$$H^*_\alpha \geq 0 \quad \forall \alpha \in c(\gamma)$$

$$AH^*_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha), \forall \alpha \in c(\gamma) \tag{23}$$

$$H^*_\alpha(s) \in \mathbb{R} \forall s \in \times_{v \in \mathcal{S}} v, \forall \alpha \in c(\gamma) \tag{24}$$

In the 2018 E2E test code, *FailsafeL2* is constructed by the *L2geoimp* function, invoked with non-*None* parent argument and *backup_feas=True*, and is solved in the same function.

We have excluded description of one subtlety in the E2E test code's failsafe implementation: in the configuration file, a minimal schema parameter can be specified, and the failsafe uses this parameter to build an extra set of constraints.

[11] In the implementation, a small slack is allowed, to avoid floating-point issues with inexact representation of distance-from-hierarchical-consistency.

These constraints are used to model the requirement that hierarchical consistency still hold, even in the failsafe, for sums taken over variables that do not participate in the non-total-population invariants. Thus, for example, the 2018 E2E test configuration file specifies that the only non-total-population invariants are on the *hhgq* dimension; the failsafe as actually implemented then retains an additional constraint requiring $\sum_{\alpha \in c(\gamma)} \sum_{g \in hhgq} H^*_\alpha(x) = \tilde{H}_\gamma \forall x$, where the cell x is understood to have multiple indices, one of which is g. A later revision of this document will describe this process more precisely; for now, we just comment that the purpose of this additional constraint is to maintain hierarchical consistency as closely as is possible while provably ensuring feasibility.

2.13 Failsafe-Feasibility L1

In each sub-national, super-block geographic unit, "Failsafe-Feasibility L1" is solved after Failsafe L2, if the Primary L2 solve previously failed; we emphasize that "L2" here is not a typo—the *PrimaryL1* solve is never entered if *PrimaryL2* reported infeasibility. Its role is to determine how closely the hierarchical consistency children-sum-to-parent constraints can be satisfied while maintaining feasibility and producing integer output. It is the mixed-integer linear programming problem:

$$\widetilde{H}^0 = \min_{H^\dagger_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} \sum_{x \in \times_{v \in \mathcal{S}} v} || \sum_x H^*_\alpha[x] - H^\dagger_\alpha[x] ||_1 \tag{25}$$

$$\text{s.t. } H^\dagger_\alpha \geq 0 \forall \alpha \in c(\gamma) \text{ (nonnegativity)}$$

$$|H^\dagger_\alpha[x] - H^*_{\alpha|}[x]| \leq 1 \forall \alpha \in c(\gamma), \text{ for all cells } x$$

$$A H^\dagger_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha) \forall \alpha \in c(\gamma) \tag{26}$$

$$\forall x: \quad H^\dagger_\alpha[x] \in \{0, 1, 2, \ldots\} \forall \alpha \in c(\gamma) \tag{27}$$

The output of this problem is a minimum achievable distance D from which hierarchical consistency can be achieved, which is then re-used to build an extra approximate-hierarchical-consistency constraint into *FailsafeL1*, in a manner analogous to the relationship between *FeasL2* and *FailsafeL2*.

2.14 Failsafe L1

In each sub-national, super-block geographic unit, "Failsafe L1" is solved after Failsafe-Feasibility L1, if the Primary L2 solve previously failed; we emphasize (again) that "L2" here is not a typo—the *PrimaryL1* solve is never entered if *PrimaryL2* reported infeasibility. It is the (equivalent-to-a) mixed-integer linear programming problem:

$$\widetilde{H}^0 = \arg \min_{H^\dagger_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} -(H^\dagger_\alpha - \lfloor H^*_\alpha \rfloor) \cdot (H^*_\alpha - \lfloor H^*_\alpha \rfloor) \tag{28}$$

$$\text{s.t. } H^\dagger_\alpha \geq 0 \forall \alpha \in c(\gamma) \text{ (nonnegativity)}$$

$$\begin{aligned}
&\sum_{\alpha\in c(\gamma)} ||\sum_x H^\dagger_\alpha[x] - \sum_x H^*_\alpha[x]||_1 \leq D+1 \text{ (total sum constraint)}\\
&|H^\dagger_\alpha[x] - H^*_{\alpha|}[x]| \leq 1 \forall\alpha \in c(\gamma),\ \text{for all cells } x\\
&AH^\dagger_\alpha \text{ sign rhs } \forall(A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha)\forall\alpha \in c(\gamma) && (29)\\
&\forall x: \quad H^\dagger_\alpha[x] \in \{0,1,2,\dots\}\forall\alpha \in c(\gamma) && (30)
\end{aligned}$$

References

[1] Hay M. Gerome M. McKenna R. Machanavajjhala, A. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment*, 11(10), 2018.

[2] Kifer D. Machanavajjhala, A. No free lunch in data privacy. *SIGMOD Conference*, 2011.

[3] 2020 Census DAS Development Team. End-to-end census github repository. https://github.com/uscensusbureau/census2020-das-e2e/tree/93296dbd4d1e86f6b0e75b6e0b04fe7af973dbff, 2019.