IP Cell Suppression

Model Reduction and a Top Down Design[1]

Bei Wang [23](CENSUS/ESMD FED)

Abstract

Most Economic Census (EC) products use cell suppression to avoid disclosure in publication. We have a sophisticated sequential linear programming (Seq-LP) cell suppression program that is as efficient and robust as that model can provide. The sequential approach is a process restricted by the linear programming (LP) model. Therefore, it doesn't provide a global optimal solution and sometimes oversuppresses. The ideal model is a simultaneous integer programing (IP) algorithm. The simultaneous IP (Sim-IP) provides an ideal solution without oversuppression.

Sim-IP cell suppression is well known for providing a global optimal solution. However, the complexity of the model makes it impractical to use in a production environment. The complexity of the model, that is largely determined by the number of primaries (targets) and solution space (available cells to complement primaries), grows exponentially as the amount of data being processed increases. In other words, Sim-IP scales poorly, and can only process small datasets. Over the years, researchers have been trying to solve the cell suppression problem with IP, but only found it to have a limited application.

Due to the characteristics of Economic Census data, the cell suppression model is usually sparse. In addition, the data are widely duplicated and there are many small, sensitive cells with small protection requirements. Our first step in this research is to do a limit test: explore what data size the Sim-IP can endure in terms of targets and solution space. Then, we will offer remedies to reduce the IP model size and achieve solutions for larger tables. With our understanding of Census data characteristics, we can reduce model size. We employ a top-down approach that decomposes the data into a geographic hierarchical space. Within this framework we examine three remedies: 1) restriction of the solution space, 2) restricting the number of targets, and 3) a combination of 1 and 2. With Sim-IP cell suppression, we aim for solving a medium-large Economic Census problem.

Key words: simultaneous/sequential IP (Sim-IP/Seq-IP), simultaneous/sequential LP (Sim-LP/Seq-LP), top down, model reduction, Cell Suppression

---

[1] This paper is disclosure approval under approval ID: CBDRB-FY23-ESMD005-004

[2] *Any opinions and conclusions expressed herein are those of the author(s) and do not reflect the views of the U.S. Census Bureau.*

[3] Many thanks to Phil Steel for his constructive suggestions toward this research, and spending numerous hours editing and reviewing this paper.

Section 1. Introduction

Most Economic Census (EC) products use cell suppression to avoid disclosure in publication. We have a sophisticated sequential LP (Seq-LP) cell suppression program that is as efficient and robust as possible for the given model. However, it sometimes oversuppresses. The ideal model is a simultaneous integer programing (Sim-IP) algorithm. It is quite tempting to ask if we can use a simultaneous LP (Sim-LP) to achieve what Sim-IP does. We have built-in a partial simultaneous algorithm (m-LP) in the cell suppression system where we can select the m, the number of primary (P) cells to be processed simultaneously. Theoretically, we can set m as the total number of P cells, making the process a Sim-LP. However when processing the Sim-LP we almost always find that the problem is infeasible, even when m is short of the full set of primaries. In addition, even if it successfully produced a solution, it would oversuppress (Wang, 2015). For Seq-LP to produce minimal oversuppression, it requires a sophisticated arrangement of P cells' flow directions in the model, see details at the end of Section 2.02(d).

Sim-IP provides a globally optimized solution. It builds a comprehensive model, oversees all the cells and relations, and coordinates among available cells. As a result, it suppresses the minimal amount in terms of value or number of cells necessary to protect all the primaries, depending on the users' objective. However, the model has serious limitations. The model has linear and Boolean variables, and the computation time grows exponentially as the amount of data being processed increases. This computational complexity results in very poor scalability, and only small datasets can be processed. Sim-IP best represents the cell suppression problem, but solution times become unmanageable. Section 2 discuss the Sim-IP model. In this research, we will first explore the Sim-IP model and test the size of data it can handle. We employ a top-down approach that decomposes the data into a geographical hierarchical space. Within this framework we examine three remedies to reduce IP model size, when data size has reached to an intolerable level. One is to restrict solution space, another is to restrict target number, and the other, in the case of first two remedies failed, is a combination of the two approaches which reduces model on both dependencies. Finally, a top down design is discussed in Section 5. In Section 4.01(a), we discuss the first remedy reduce targets; in Section 4.02,Section 4.02 the second remedy reduce solution space . Test result and analysis are in Section 3, Section 5.01 and Section 5.02.

Section 2. Sim-IP model

Sim-IP was discussed in Dulá et al. It had good theoretical basis, but proved unrealistic because computing power was not adequate at the time. Research was focused on network minimum cost flow model. The Economic Census used that model for disclosure avoidance from 1992 to 2007, then the LP model for 2012, 2017 and 2022 Economic Census. As LP becomes more mature this is a good time to devote some time on IP; relying on the growing efficiency of computing will not put the IP problem within our reach in the foreseeable future. We need to find an efficient algorithm to overcome the problem.

Section 2.01 Review of Seq-LP, m-LP, 1-LP cell suppression model and process

Suppose the data set has n_p primaries (Ps), then we need to run a simple LP model n_p times, in theory. Henceforth we refer to this process as Sequential 1-LP (Seq-1LP). However, in practice, we are able to reduce the number of optimizations substantially using a procedure we call skip P, When using the skip P procedure, we generally find that a typical Seq-LP process only runs approximately 10% of n_p. We can also set model to protect multiple primaries at one time, we refer it as m-LP. If we set m=n_p, it becomes one simultaneous LP model to solve all primaries. However, such a model is impossible to solve without running into infeasible. This is because that the $m(= n_p)$-LP will have stacking flow which sometimes violates the model constraints.

"Skip P", is a mechanism used in our Seq-LP process. We maintain a queue of primaries, each requiring optimization. At the end of each optimization, we update any primaries on the queue that had enough flow to cover the primary in the solution from current optimization. If it does, we mark it as "skipped". Then at the beginning of the next optimization, we check if the primary on top of the queue is marked "skipped". The process will continue until it finds next unmarked primary in the queue for the subsequent optimization. Steel et al.notes that solutions are re-usable, provided the flow can be scaled downward, thus insuring that the bounding conditions are maintained.

Section 2.02 Sim-IP process

To understand the Sim-IP process, let's first look at the Seq-mLP process. Below is the diagram of a sequential m-LP process:

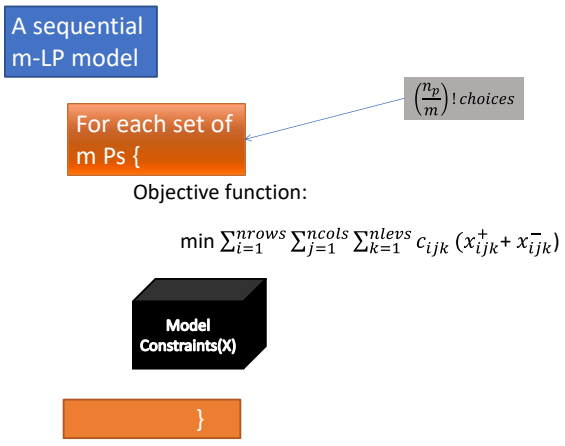


Figure 1: Seq-LP process, where $x_{ijk}^{\pm}$ is the cell variable, c_{ijk} is the cost

Notice that in a Seq-mLP process, in theory, it will have $(\mathbf{n_p}/\boldsymbol{m})$ optimizations (less because of skipPs) , and it will have $C_{n_p}^{m}$ combinatorial ways to select m Ps. There are at most $[\frac{\mathrm{n_p}}{m}]+1$ sets of mPs. The number of orderings for a mLP process is $[\frac{\mathrm{n_p}}{m}]!$ in theory.

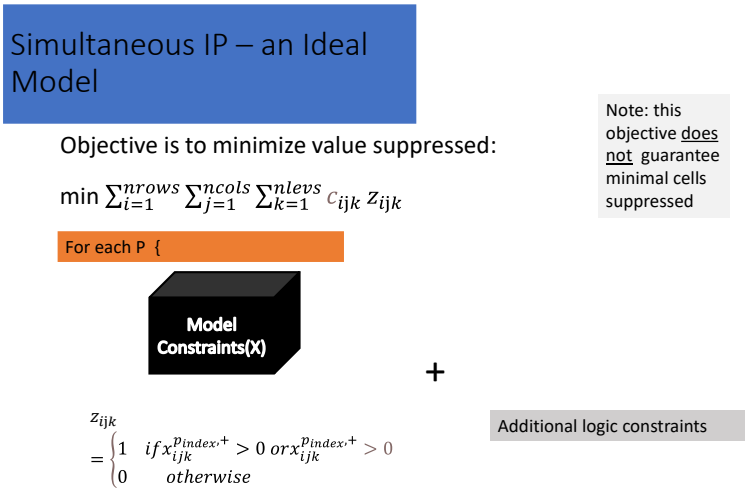


Figure 2: Sim_IP process, where z_{ijk} replaces $x_{ijk}^{\pm}$ in the objective, bounded by logic constraints.

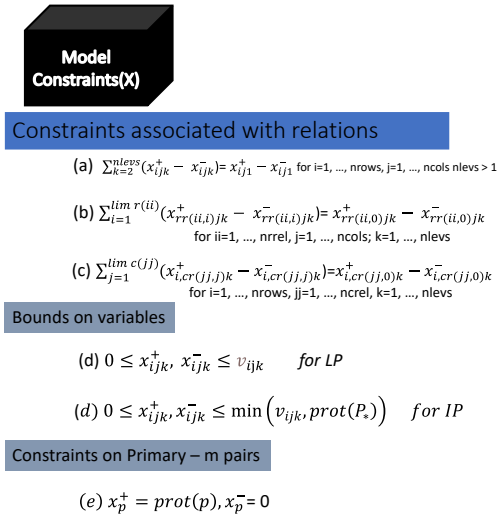


Figure :3 Black box: equality constraints and bounds

There are some things that are worth noting in IP:

1. IP is one model, one optimization.
2. A tighter bound can be used to increase numerical stability without losing any performance efficiency, see bound constraint d) in Figure :3.
3. The sequence of the models in Figure 1 is replaced by a single model in a product space referred to in orange on both of the diagrams.
4. One Boolean for each candidate variable (instead of plus or minus 2 variables), where candidate variables are the variables available for complementary suppression.
5. n_p sets of linear (up/down) variables
6. The objective for this model optimizes the value. It does not necessary guarantee the minimal number of cells are suppressed true for both LP and IP.

There are some challenges in implementing the logic constraint. We discussed the approach in (a)-(d) below. the IP model properties, computational complexity, and applications are discussed in 0-(e)(ii) below.

(a) To simplify the logic constraints in Figure 2, we linearize the constraints. This makes solver more efficient because it eliminates the need of creating auxiliary Boolean variables. For example, the logic constraint can be expressed as either (1), (2) or (3)

$$\left(x_i^{p_{index},+} \geq 1\right) + \left(x_i^{p_{index},-} \geq 1\right) - z_i = 0 \qquad (1)$$

Or

$$x_i^{p_{index},+} + x_i^{p_{index},-} - bigM * z_i = 0 \qquad (2)$$

Or

$$x_i^{p_{index},+} - bigM * z_i \leq 0, \qquad x_i^{p_{index},-} - bigM * z_i \leq 0 \qquad (3)$$

Where $bigM \approx max_i(U_i)$ and U_i is the upper bound for cell i, Among three choices, implementation (2) is preferred. It would seem to that implementation (2) and (3) are similar, however they have large differences in running time. The difference in runtime could be caused by the number of constraints represented in the equation, where equation (2) has one constraint, and equation (3) has two constraints.

(b) The Choice of *bigM*

In Dulá et al. (2004), a cell's upper bound U_i is used in the following manner:

$$x_i^{p_{index},+} - U_i * z_i \leq 0, \qquad x_i^{p_{index},-} - U_i * z_i \leq 0$$

This is not a viable choice because it sometimes leaves a primary suppression without any protection or complementary suppressions otherwise known as "Single D." In other words, it returns a sub-optimal solution.

To further explain this, consider the constraint $x^{pindex} < bigM * z$. When $bigM \gg x^{pindex}$ => z = $\frac{x^{pindex}}{bigM} \sim 0$ ($\neq 0$). In other words, z is close to zero. In fact, due to numerical precision, it rounds to 0 and this causes "Single D". To mitigate this problem, we can interfere with CPLEX by setting z based on x. However, the solution will be sub-optimal.

(c) How to setup *bigM*

In our test for ACES, we set $bigM = 10^7$, because it was a large number that was greater than any observed value in the dataset. When the same value was used for the EC's 2017 detailed geographic employment data, it caused a "Single D". After some examination, we noted that the EC 2017 employment values needed small protection at the state level. For the whole 2017 EC employment? data set, the largest protection values is less than 1000. To avoid "Single D", $bigM$ has to be compatible with protection values. We finally set $bigM$ = 100 for employment. A $bigM$ that is smaller than protection value may cause cells split up the protection value to protect the target. Later we learn that there is a more accurate setup for $bigM$ discussed in (d).
Another case to be cautious is when *bigM* is set too small, i.e., smaller than the required protection. It leads to infeasibility as the constraint contradicts itself: $prot(p_{index}) > bigM$ and constrained to $x^{pindex} < bigM$.

(d) The best setup for *bigM*

$bigM = prot(p_{index})$, the protection requirement of the p_{index}, is the best choice to avoid numerical problems and infeasiblity.

We like $bigM$ to be greater than= x^{pindex}, and close to x^{pindex} so the fraction $\frac{x^{pindex}}{bigM}$ isn't mistakenly set to zero because of numerical error. We know x^{pindex} is the flow in and out a cell through the model. The flow is controlled by protection required for a particular p_{index}. The closest setting is *bigM*= $prot(p_{index})$. This setting is ideal, because it is numerically appropriate, and it is dynamic because it automatically adjusts for each p_{index}.

Note on constraints (e) in Figure :3: in IP model m is always equal to 1 and in LP model, it could be m pairs for mLP. It can be set in alternate fashion to get a better suppression. i.e., for some primaries, the (-) variable sets to protection value and the (+) variable sets to 0 and vice versa. This is because, in the case of multiple Ps in the same relation, the alternating assignment of protection may cancel each other in the additive constraint. Therefore the combined protection requirement is less. On the other hand, if it is not set alternatively, the protection requirement is an augmented sum.

(e) IP Model: Properties, Computational Complexity, and Application

In this section, model properties and their complexity and application will be discussed.

(i) Properties of IP Model

Sim-IP model consists of n_p LP models and a set of Booleans. Each LP model sets the protection for a P. The Booleans define the objective function.

Claim 1: The (optimal) solution of Sim-IP is also a (optimal) solution of Seq-LP, i.e., min(obj(IP)) <= ∑ min(obj(LP)), but not vice versa.

Claim 2: The solution of Seq-mLP model satisfies the Sim-IP model constraints, but is not necessarily an optimized solution for the Sim-IP model.

Claim 3: A Seq-mLP solution can be used as a basis for Sim-IP, i.e., the solution of a Seq-LP can be used as a solution space for Sim-IP to eliminate extraneous complementaries, Cs.

Recall that we are trying to minimize either the overall value suppressed or the number of cells suppressed in cell suppression. The objectives can be achieved either way. Using IP for cell suppression we get an exact and minimal solution. On the contrary with LP we get an approximate solution that may not be minimal overall. Each step in the process is a minimal solution, but when combined it may not be minimal. Following is the proof that IP is a better model than LP for the cell suppression program.

Considering in a sequential 1-LP for example, there is a P_1 (protection$_1$) which finds its set of Cs denoted by $\boldsymbol{A}$. $\boldsymbol{A}$ is the optimal set for . In the following sequence, P_2 finds its set of Cs denoted by $\boldsymbol{B}$. $\boldsymbol{B}$ is the optimal set for P_2. Suppose that $\boldsymbol{G}$ is another set of Cs that met the constraints for P_1 and P_2 , but is not the optimal set for either. In this case we have:

$$obj(\boldsymbol{A}) \leq obj(\boldsymbol{G})$$

$$obj(\boldsymbol{B}) \leq obj(\boldsymbol{G})$$

$$obj(\boldsymbol{A} \cup \boldsymbol{B}) \geq obj(\boldsymbol{G})$$

In a Sim-IP model, the model knows that $\boldsymbol{A} \cup \boldsymbol{B}$, and $\boldsymbol{G}$ are two feasible solutions. However, $\boldsymbol{G}$ is the optimized solution. In Seq-LP model, the model at a time knows that $\boldsymbol{A}$ and $\boldsymbol{G}$, and $\boldsymbol{B}$ and $\boldsymbol{G}$ are feasible solutions at a sequential step in two different models. The two models choose their own optimal one. $\boldsymbol{A}$ and $\boldsymbol{B}$ are the optimal solution respectively in each sequential step. $\boldsymbol{G}$ is the optimal solution in the simultaneous process.

(ii) Application of IP model

1. It can be used to find a global optimal pattern (least value or least cells) for cell suppression as we discussed above in Section 2.01.
2. It can be used to improve the Seq-LP solution by removing excessive suppression as we discuss below in Section 4.02.

(iii) Sim-IP model complexity

A simultaneous IP (Sim-IP) cell suppression process is one model process, which essentially combines n_p 1-lp models. The number of variables and constraints are

$$nvars(Sim-IP) = 2n_p n + (n - n_p) \quad (4)$$

$$ncons(Sim\text{-}IP) = n_p * ncons(1\text{-}lp) + n_p \quad (5)$$

$$ncons(1lp) = \begin{cases} m^v n_r + n^v m_c & 2D \\ n^v m^v l_l + m^v l^v + n^v l^v & simple3D \end{cases}$$

where $2n_p n$ is the number of linear variables and $n - n_p (= n^{Bool})$ is the number of Booleans. A simple 3D is where both row and column are simple one level, and only the 3rd dimension is complex, as illustrated in the example: geographics relations. $n^v,\ m^v,\ l^v$ *are number of* elements in its row, column and level relations, n_r, m_c, l_l are number of row, column and level relations. In case of simple3D, $n_r = m_c = 1$.

For a given table the number of constraints for the LP model is fixed. The number of constraints of IP is a linear function of n_p, see equation (4) & (5). We are working on a computational test to see how the number of Booleans, n^{Bool} affect the runtime.

In Section 4, we discuss two simpler models, the first that reduces the number of targets (related to n_p) in Section 4.01(a); then, the second that reduces the number of n^{Bool} in Section 4.02. We want to know how these heuristic approaches compare to the Sim-IP discussed in this section.

Section 3. Looking at the computation limits of basic IP

We ran an initial test on three data sets with IP model see Table 3-1. It lists the model size in terms of number of Ps and total cells. It shows the results on the suppression pattern with number of suppressed cell and value of total suppression for both IP and LP, for comparison purpose. The detailed description on the test data is after the table display.

Table 3-1 Small tests on three different data sources comparing IP and LP

Data sources: Tiny is an extraction of sector 71 taxable/tax exempt table from 2007 Economic Census; 2015 Annual Capital Expenditure Survey (ACES), and a subset of Sector 22 annual payroll from the 2017 Economic Census

		Tiny	2015 ACES	2017 EC Annual Payroll from Sector 22
# Cells in the overall data		70	4620	1958
# P's (primaries)		42	71	891*
Number of Complementary Cells Suppressed	LP (Standard for comparison)	14	197	182
	IP	9	146	135
Total Value of Complementaries	LP	90984	4475688188	3021597
	IP	63166	4153993520	2370002
Processing time	LP	2sec	5sec	1min
	IP	<10sec	5min	35min
%reduction in suppression (cells,value)	Complementary Cells	-35%	-26%	-25%
	Total Value of Complementary	-30%	-7.20%	-16%
*This is an unduplicated count				

We see reduction of suppression around 7-35%, at the same time, rapid increasing of processing time as data grows as expected. We use unduplicated count of Ps to reduce model size for 2017 EC as it approaches model compacity.

(a) Tiny is a three-dimensional table, consisted of a 5x10x3 table linked to a 5x3x3 table. It is a small table, but it still has a linked structure that exhibits several different problems and properties for cell suppression. This simple yet structured table is great in testing and identifying code issues. It helps track whether our model is robust. It was also served in batch marking in IP cell suppression research.

(b) ACES 2015 has 5390 cells and 85 Ps; 2 table groups, one large and one small. We focused on large one: 4620 cells, 71 Ps. This program was managed by Richard Moore. We followed his parameters.

ACES is our second goto data set other than Tiny. We often use it to research oversuppression by LP.

(c) The excerpt from the EC has 1958 cells and 1553 Ps with 891 unduplicated Ps. Note: Among all the Ps, unduplicated are amount to 43% (=891). Only unduplicated are the necessary targets built in the model.

(d) Employment data from 2017 EC

The challenge with EC employment data is the large amount of data and the small protection requirement for primaries. There were 518,683 cells with 305,062 Ps. It is clearly inaccessible by IP. To address this, we consider both heuristic approaches and a top down design approach to reducing the model, which are detailed in Section 4 and Section 5, respectively.

Section 4. Heuristic approaches – model reduction

Serpell et al. has discussed model reduction for IP cell suppression model.

In this section, we discuss model reduction. That is, how to reduce the size of an IP model with information learned from LP or some intelligent guess. An IP model's complexity is determined by the size of variables, the Boolean variables especially, and the feasible region. We explore two basic ideas; reduce the number of Ps processed (smaller target list) and reduced solution space. The goal here is to reduce the model without compromising Sim-IP's optimality.

We look at four methods to reduce targets:

1. Reduce to $Px(lp)$ from LP discussed in Section 4.01(a)

2. Exclude self-sufficient skips from overall $\boldsymbol{Pset}$ discussed in Section 4.01(b)

3. Reduce to single Ps and super cell discussed in Section 4.01(c)

4. Use data information that is to exclude primaries that require small protection(=1) discussed in the example at the end of Section 5.01 Claim 4.

Another model reduction method is to reduce the solution space, this is discussed in Section 4.02.

These reduction methods could be applied either alone or in conjuncture with a top down approach discussed in Section 5.

Section 4.01 Reduce Targets

For this section when discussing the LP setting we will use the following notation:

$$Ps(lp) = \{SkippedPs\}$$

$$input: \boldsymbol{Pset}(lp) = \{all\ sensitive\ cells\ \},$$

$$ouput: Cs(lp) = \{Cs: complementary\ suppression\ from\ LP\},$$

$$derived: Px(lp) = \{processedPs\ from\ LP, as\ targets\ in\ model\},$$

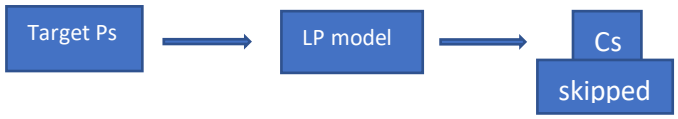
We have: $$\boldsymbol{Pset}(lp) = Px(lp) \cup Ps(lp)$$

$$P\&C(lp) \triangleq \boldsymbol{Pset}(lp) \cup Cs(lp)$$

The **complexity** of a Sim-IP model is dependent on the size of $\boldsymbol{Pset}(lp)$**,** the number of linear and integer variables, and the number of constraints. One question we consider, when using the IP approach, can we ignore the Ps that are skipped in LP? While all the additivity constraints are necessary to guarantee that the cell suppression pattern prevents the suppressed value from being estimated too closely, it may be possible to reduce the size of $\boldsymbol{Pset}(lp)$ and/or number of variables in IP. To understand these ideas, we need to discuss how LP cell suppression works with respect to two universes: targets and solution space, which are discussed in this section (Section 4.01) and Section 4.02 respectively.

Figure 4 1-LP illustration

LP (1-LP) optimization flow chart



This process will run sequentially until all Ps in $\boldsymbol{Pset}$ are exhausted. *Final products*:

- Definition: What are the targets ($Px(lp)$) in LP?

LP cell suppression **produces** a solution. only 10% of Ps are targeted (processed) in a model, the others are skipped because they are protected either initially by other suppressions of sensitive cells or have become protected by complements already selected in the process. We call the processed Ps Px(LP), which is all the Ps minus skipped Ps.

- LP cell suppression solution:

Each model produces a solution; a set of complementary (Cs). The cumulative Cs are the final result $\boldsymbol{Cs}(\boldsymbol{lp})$.

How would an LP output information useful in IP?

- For this discussion we assume that the collection and order of the $\boldsymbol{Pset}$ is random[4]. The initial $\boldsymbol{Pset}$ is a queue. The target p is the first non skipped primary on the $\boldsymbol{Pset}$ from previous optimization. There is no prior knowledge of how many items in the $\boldsymbol{Pset}$ will be skipped.
- In an LP model for each target, we see two parts of skips. One part is connected to the target. Another bonus part is mutually protected in its own circuit (self-sufficient skips). The first part of skips and the target are connected by a traceable path following some additive relations in the model. We say the target p carries the skips. We are interested in the self-sufficient bonus part. Can we sort it out from the two parts? See discussion in (b).

(a) Reduce targets using $Px(lp)$ from LP cell suppression

As we discussed above, we will be only interested in the Ps in $Px(lp)$

In this process, we restrict the target Ps to these in the $Px(lp)$. In our experience, $Px(lp) \leq 10\%$ $\boldsymbol{Pset}$

In number. This reduces size of the Sim-IP model by 90%!

The input file for IP is taken from the output of LP cell suppression, then set as following

$$input\colon P^{ip} = Px(lp), C^{ip} = \boldsymbol{Pset} - Px(lp)$$

We hope by protecting Ps in $Px(lp)$, the IP model also protect all Ps. The statement is true in a Seq-1LP process under the deterministic solver performed on $Px(1lp)$. However, LP oversuppress and that oversuppression can subsequently be used to skip a primary. Then the skipped primary might be unprotected if it is not included in IP model. For that reason, the protection of $Px(lp)$ in a Sim-IP may not be enough for a Sim-IP model to cover all P in $\boldsymbol{Pset}$, but certainly a good starting set.

(b) Reduce targets by excluding self-sufficient primaries

There are primaries that protect each other among themselves. I call it self-sufficient P. We asked if we can obtain a list of self-sufficient P at the beginning of this section. The answer is yes, we can identify them quite easily.

We know that the model produces a set of self-sufficient skips along with the the skips involving in protecting the current target. Then, what happens if we run an LP model where no target is set? This optimization will return a zero objective for infinitely many solutions. One of them is $\{0\}$ where all available cells have flow 0. Another solution is where cells are allowed to have maximized flow as long as they form a circuit. The later solution is what we obtain from an actual run. We want to believe that these are all the self-sufficient skips (not quite sure whether this is model or solver behavior). To test our hypothesis, we can setup a test by running 1LP one at a time as if it is a only P on remaining P. We want to check if any optimizations return a zero objective. If it does not return any zero objective, then our hypothesis is true.

We then obtain a reduce targets for IP by excluding these self-sufficient skips from $\boldsymbol{Pset}$. Contrary to $Px(lp)$ discussed in (a), the protection of this obtained reducde targets in a Sim-IP is sufficient to cover all P in $\boldsymbol{Pset}$.

[4] We do order in 1-LP but not m-LP. In m-LP, the design is to separate the m Ps. We thought a random ordering is better than any ordering we could think of.

We did a test on employment from 2017 Economic Census. It has 207,015 self-sufficient skips out of 305,062 making the reduce targets to, 98,047. That is less than 33% of total P nationwide. For one of the states the reduction is even larger with only 20% of total P.

(c) reduce targets to single Ps and "super cell"

In Serpell et al., it reduces targets to "initial exposed" (E) cells to include in the IP model. We use a similar idea to reduce targets to "single Ps" and "super cell".

A single P is a primary that is the only suppression in a relation. It is clearly idenfiable and need protection. A super cell is a aggregated sensitive cell consisting of two or more primaries. It is easy to identify single Ps and "super cell".

We tested this approach to St23, a state level subset of the 2017 Economic Census' main employment table. it has 2636 primaries and 4036 total cells. singlePs (163) and Ps from super cell (146) add up to 309 Ps. It was still hard to solve for IP in a one step IP approach. We manage to start with 163 singlePs plus selected super cell. We choose one P from each superCells. The initial total P for IP is around 200, it proves not enough to protect the rest of Ps. We then keep adding Ps to the model selected randomly from superCells, the final total number of P used in IP model is 241 (163 single Ps and 78 from super cell). The audit LP return with no more Cs. The result is shown in Table 5-4.

Section 4.02 Reduce solution space

Seq-LP produces a solution $\boldsymbol{Cs(lp)}$ which is not globally optimal. However, it is also a solution for Sim-IP model. In a Seq-LP process, later optimization call, although it uses information from prior optimizations, doesn't have the ability to cancel any Cs from a prior optimization. However, a Sim-IP model would recognize any excessive cells and retrieve a subset of $\boldsymbol{Cs(lp)}$. This model also can be used to identify excessive Cs and remove oversuppression.

Two possibilities $\boldsymbol{Cs(lp)} \supset \boldsymbol{Cs(ip)}$ or part of $\boldsymbol{Cs(lp)} \subset \boldsymbol{Cs(ip)}$.

If it's the first case, IP model on reduced solution space would achieve its true optimal. If it's later, IP model would result a better solution by removing extraneous suppressions.

We reduce the model by simply restricting its solution space:

- Freeze the cells not in $P \cup C(lp)$ in d_file (using the same input input file as LP).
  Freeze is to remove the cell by making it 0
- Remove any skipped

By freezing in first step, we limited candidate variables to $\boldsymbol{Cs(lp)}$**.** We then run the reduce with IP cell suppression.

Reduce targets and solution space are necessary to deal with most Economic Census' data. We can also have a top down process to work with reduced model. The top down is discussed in Section 5.

Section 5. IP with a Top Down Design

Basicl2017_emp with more than 500,000 cells and more than half of them are Ps, see Table 5-1, It was large but not difficult for LP. It is very difficult and practically impossible for IP. The top down procedure is to decompose the large problem into its "natural" subsets, in geography, which can be processed by IP. Depending on data size, this could be a multi-level process.

Although top down processing has not been done in cell suppression in Economic Census, similar processing has been used, such as, in disclosure group where disclosure process is divided into several groups. In disclosure group procedure, one disclosure group is processed after another. The pattern from the previous is preserved for the later disclosure group, therefore frozen in the later run. So goes on with next disclosure group, etc. Top down generally means to process the data group high in the hierarchy first. Statistics Netherland has a software for cell suppression, τ-ARGUS. Its modular approach is a top down design.

Section 5.01 An example of top down practice

Basicl is a detailed geographic where the hierarchy are US, states, counties, …, cities, and places. Our production program automatically identifies independent pieces of the cell suppression problem it is given, ie those without a linking relation. We can easily create a 2-level process. One is the US-State. The others are generated by removing the US_State relation in geo relation file. When we run the problem with the omission we obtain the table groups information. The fifty two state or pseudo states are divided into 9 disjoint groups. Some are single states; some are groups of states. We classify the 9 groups by their size: small (st02, st15), medium (st23, st30), large (st06, st08, st0432), huge (grp01, St16414956). The 2 digit numbers indicate the states where multiple states are indicated with a serial number except for grp01. Grp01 is the largest consisting of the rest of states that are not in other eight groups. See

Table 5-2. We will outline the procedure, with the first two levels needed for a complete solution. Level 3 is very data dependent, and we illustrate what's needed for st23 only.

1st level: The first IP is on US_state, just one relation with few primary suppressions. After the 1st IP, we save the suppression pattern and freeze the pattern if there is an overlap within next geo level, state. we solve IP for next level groups that small enough to fit into IP model. In basicl emp, the state cells pattern is frozen for state_cty level IP.

2nd level: If the first 9 groups are small enough for IP, then we are done decomposing . Otherwise, it will be recursively decomposed into small subsets as described above. The employment, in the detailed geographic series, the two smallest, st02 and st15, are fitted for IP. Other than these two, no other groups seem to fit into current computing power. The IP runs out of memories quickly. St23 needs to be further decompose into 4 groups see

Table 5-3 Basicl_emp_st23 Table_Groups. This becomes the third level processing.

3rd Level: st23 has 4 groups: two are small and two are large. The small two are fitted for IP. The large two G0US23 and US438, both are similar size, failed completely on IP: out of memor[5]. (We could decompose into a 4th level, another approach) the decision was made to focus on US438 with the following approach. The approach that worked with US438 uses method of reduce targets in Section 4.01(a). The detail is following:

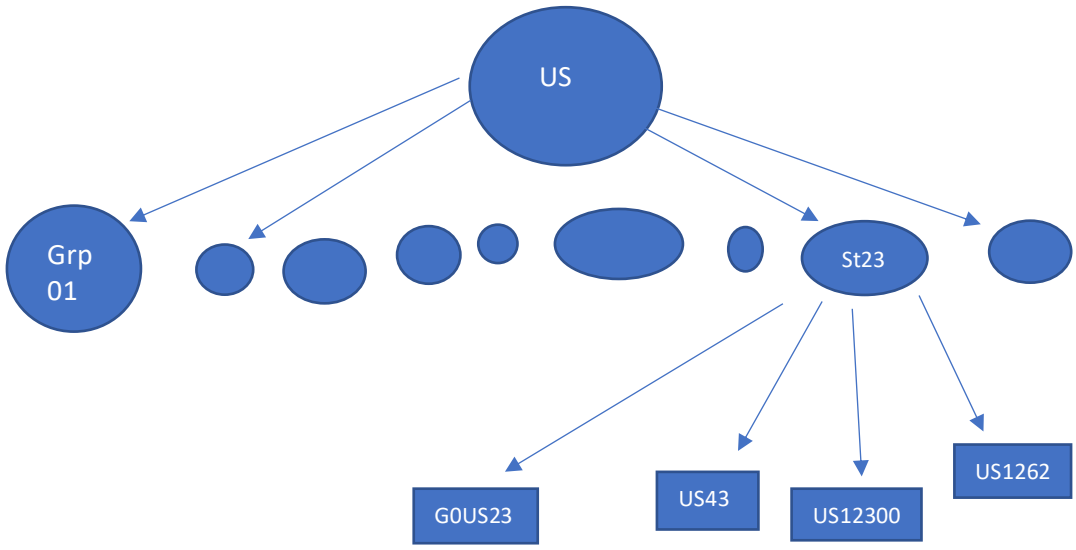
The c_stxcty has 13 Cs associated to US438. The 1st step is to change 13Cs to Ps, making the total Ps 1052 (1039+13). The 2nd step is to change Ps with prot=1 cells to Cs. This reduces the number of Ps to no more than 300 which is a good fit for the model. Then it runs on IP to locate any SingleDs. If SingleDs, 3rd step is to change some or all Cs in SingleDs to Ps, then run IP again. (set p_inclusion to 2: P only)

This approach is quite specific to emp data since there are many Ps with prot=1. That means if they are not SingleDs then we can safely claim they are protected. That is a nice property with prot=1.

Claim 4: a P with a protection requirement 1 is protected if it is not a single D given that all other Ps are protected.

This is to say that we only need to build other single Ds as targets in the IP model.

graph 1 Geography hierarchy composition of groups



[5] On a Red Hat Linux machine E5-2697A V4 @2.60GHz processor, memory 16 GB, 8 thread

Table 5-1 Top down process BASICL_EMP – 1st level: running IP on US and State

Data sourse: Economic Census' employment in 2017

		All Geo	US_STATE
#Cells in Overall Data		518,683	3,410
#Ps (Primaries)		305,062	104
IP	Number of Complementary Cells Suppressed	N/A	40
	Total Value of Complementary		3,144
LP m=1	Number of Complementary Cells Suppressed	39,340	51
	Total Value of Complementary	14,667,479	3,300
	Time	620393s(7d)	15min

Table 5-2 BASICL_EMP data Statistic-2nd level: 9 groups from State and below.

Data sourse: Economic Census' employment in 2017

Groups		Grp01	St02	St0432	St06	St08	St15	St1641495356	St23	St30
#Geo relations		2838	12	38	70	59	5	190	20	25
Rank:ByGeoSize		1	8	5	3	4	8	2	6	7
#cells in overall data			1126		24654	6964		26117	4036	
#Ps (Primaries)			625		12490	3863		14303	2636	
IP	Number of Complementary Cells Suppressed		95				✓	2106	Out of Memory	
	Total value of complementary		13070					633843		
	Run Time		7m							
LP	Number of Complementary Cells Suppressed							2376	236	
LP Audit	Number of Complementary Cells Suppressed							2143		
9 Table Groups are identified by removing US_State relation from geo file: US = ST01 + ST02 + …										

Table 5-3 Basicl_emp_st23 Table_Groups- 3rd level on break down data statistics

Data sourse: Economic Census' employment from 2017

St23		Stxcty only	G0US23	US12300	US12620	US438
#Cells in overall data	4036	415	1642	314	320	1694
#Ps (Primaries)	2636	62	1122	250 + 2freezeCs	225	1039
#Relations in geographic relation		1	11	1	1	6

Section 5.02 Reduced and top down comparison

We test reduced and top down on st23 a subset of Economic Census' employment from 2017. The suppression pattern on these tests is audited successfully using LP cell suppression program.

In reduce targets, we were able to reduce primaries to 241 vs 2636 originally, discussed in Section 4.01(c).

Both approaches have similar improvement on suppression with 18% reduction comparing to LP.

Table 5-4 IP Suppression Statistic from top down and reduce target of IP and LP

Data sourse: Economic Census' employment from 2017

St23	#cells in Data	4036	LP	IP	
	#Ps (Primaries)	2636		Reduce Targets (241 primaries)	Top Down
Number of Complementary Cells Suppressed			236	193	199
runtime				41min	69min
Total Value of Complementary			65672	54091	54329
%reduction in Complementary	#Cells			-18%	-16%
	Total Value			-18%	-17%

Conclusion and Future Research

In this research, we focused on IP model building, reducing model, and top down design.

We improved the bounds and constraints of the model. The tighter bounds and tighter constraints helps to avoid numerical instability caused by huge range of data and leads to a more stable algorithm.

We illustrated both reduce model and top down process in details for st23 a subset of Employment from 2017 EC. Top down procedure working with reduced targets enable us to process more with less steps. This size is so far the largest data IP cell suppression has processed. It produced better if not globally optimal cell suppression pattern than any other considered model produced.

Some ideas come out from this research

The IP model built in this research is focus on table level protection because I was more interested in improving IP model to work for large data. It can be easily extended to company level protection by setting each variable's upper bound to its "capacity" in the Sim-IP model.

Capacity measures how much a cell can give to a target. Therefore, it is an attribute toward a particular target. To provide company level protection, each cell in solution space needs to calculate its capacity toward a target. In a Seq-mLP cell suppression, the table level protection and company level protection are taken care of in two passes. The reason of two passes' mechanism is because it is impossible to calculate cells' capacity, when m the number of targets is great than 1. However, the Sim-IP model made it possible.

We always know it will be very helpful if we could identify these self-sufficient cells. During the review of this paper we unexpectedly found a solution to identify self-sufficient skips. This will not only expand data limit that IP model can process, discussed in Table 3-1, but also provide a necessary reduction. The discovery of a model that produces what appears to be a complete list of self-sufficient P raises several interesting avenues for research. In particular, why the model produces that solution rather than the simple 0 solution.

Bibliography

José H. Dulá, James T. Fagan, Paul B. Massell. (2004). *Tabular Statistical Disclosure Control:Optimization Techniques in Suppression and Controlled Tabular Adjustment.* Suitland: US Census Bureau http://frwebgate2.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107 cong public laws&docid=f:publ347.107.pdf.

Martin Serpell, Alistair Clark, Jim Smith and Andrea Staggemeier. (n.d.). Pre-processing Optimisation applied to the Classical Integer Programming Model for Statistical Disclosure Control.

Peter-Paul de Wolf (Modular), A. H.-J. (2014). *τ-ARGUS User's Manual.* Statistics Netherland.

Philip Steel, James Fagan, Paul Massell, Richard Moore Jr., John Slanta, Bei Wang. (2013). Re-development of the Cell Suppression Methodology at the US. *Joint UNECE/Eurostat work session on statistical data confidentiality*. Ottawa, Canada.

Wang, B. (2015). Reducing the Infeasibility and Oversuppression for m-LP Cell. *Proceedings of JSM 2015*. Alexandria VA: ASM.