Reducing the Infeasibility and Oversuppression for m-LP Cell Suppression Process

Bei Wang*
U.S. Census Bureau†
Washington, DC 20233

Abstract
The 2012 Economic census uses cell suppression to protect sensitive information. We general use 1-LP sequential approach for small table and m-LP partial simultaneous approach for large table. This research is about limiting oversuppression and infeasibility caused by grouping particular cells together in m-LP. - We have some examples illustrating why and how m-lP causes infeasibility and oversuppression. - We establish a baseline to evaluate oversuppression. The baseline uses 1-LP, but given a particular n, there n! outcomes needed to run 1-LP multiple times to get an average and variance. We use a 3-d table from the 2012 Economic Census. - We develop some algorithms to reduce infeasibility. The general idea is to set m cells wide apart in terms of relationships such that each targeted cell finds its own protection without interacting each other. We identify a class of cells which should be done first and fit this into our algorithm for forming m-groups. This research should benefit the 2017 Economic Census.

1. Introduction to Linear Programming Cell Suppression

The current Economic Census uses cell suppression to protect sensitive data. The cell suppression software uses a linear programming (LP) model. It sufficiently protects sensitive cells, but it isn't efficient enough to handle the largest data sets. We remodeled into an m-LP, see reference [1], to accommodate the larger data; these usually have a detailed geographic level. m-LP model based cell suppression not only works, but also works efficiently.

A typical LP sets constraints in a multi dimensional grid with one targeted entry (the primary cell) and finds an "optimal" suppression pattern to protect that target. The cell suppression process completes after solving an LP for each of the primary cells. This is a simple sequential approach. The number of constraints is determined by the size and complexity of the tables being published. The performance depends on both the number of constraints and variables and the number of primary cells. There are two problems with the simple sequential approach. The first, even if the execution time for each target is fast enough, the time spent on the whole process, which depends directly on the number of sensitive cells, can be unsupportable. The second, while each LP is optimal it is not optimal globally. In reference [1], we introduced a partial simultaneous approach to address these issues. A partial simultaneous approach is when several targets are formulated in one LP; the computing time remains the same while the overall processing time is reduced. It is clear that the larger the targeted m is in m-LP, the less the overall processing time. The processing time is generally reduced to a fraction of m, provided it is successful . However, the problem may be infeasible as some of these

*bei.wang@census.gov

†This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. Any views expressed on statistical, methodological, technical, or operational issues are those of the author(s) and not necessarily those of the U.S. Census Bureau.

m cells could conflict with each other. It is important to find m cells that aren't mutually conflicted , or even better, are complement each other in a way reduces oversuppression.

We have completely processed the most challenging part of 2012 Economic Census. We are doing further research to find an m-selection algorithm that reduces infeasible, and also oversuppression, to improve the process for 2017 Economic Census. Those are the two complementary objectives. Section (2.1) illustrates m-LP model. Section (2.3) shows some examples illustrating why/how m-LP causes infeasible and oversuppression.

Section (3) establish a baseline to evaluate oversuppression. The baseline consists of a variety of runs using 1-LP under different processing orders. We take the average suppression pattern of the multiple runs as our baseline. We use one of 3-d economic census' manufacture geographical area series (GAS) data which is a large data set and provides sufficient data structure for the research.

In section (4), We develop some algorithms to show how we can reduce infeasible and oversuppression.

2. m-LP - a Simultaneous LP Model

Definition 1 *an m-LP is a simultaneous linear programming model in Economic Census Cell Suppression Program to protect multiple Ps, say m Ps where $m > 1$, in one LP formulation. In particular, a 1-LP or LP is where m=1*

m-LP has the same complexity as one P LP (1-LP) because its constraints are still linear without adding additional variables and constraints. However, it reduces the number of times the solver is called, and processes m Ps at the same cost as it does one P. Therefore, it dramatically reduces the running time, usually to a factor of m. The only changes are on the bounds of the protection needed variables (Ps), ie: $x_+ = prot\_required$, and $x_- = 0$ instead of $x_+ \in [0, value]$, and $x_- \in [0, value]$, as described in section (2.1). There can be two side affects with this approach. One is oversuppression. The other is that it leads to conflicting constraint causing an infeasibility. We'll talk about in Section (2.3) and (5.1).

2.1 Setting Up a m-LP - constraints and cost objective function

We are solving a m-LP for m Ps. The goal is to

$$\begin{array}{lll} \text{minimize} & \sum_{i,j,k} v_{ijk}^r (x_{ijk}^+ + x_{ijk}^-) & \\ \text{subject to} & \sum_{j,k} (x_{ijk}^+ - x_{ijk}^-) = x_{i11}^+ - x_{i11}^- & \forall i \\ & \sum_{i,k} (x_{ijk}^+ - x_{ijk}^-) = x_{1j1}^+ - x_{1j1}^- & \forall j \\ & \sum_{i,j} (x_{ijk}^+ - x_{ijk}^-) = x_{11k}^+ - x_{11k}^- & \forall k \end{array} \tag{1}$$

where $x_{ijk}^{\pm}$ are flows through each cell, v_{ijk} generally are the cell value and r is the real exponents.

Additionally, for these selected m primary cells, we have

$$\begin{array}{l} x_{ijk}^+ + x_{ijk}^- = \text{prot}_{ijk} \\ x_{ijk}^+ = 0 \ \text{ or } \ x_{ijk}^- = 0 \end{array} \tag{2}$$

Notices that constraints (2) add $2m$ binary variables, and makes it a mixed integer programming which is a much harder(complex) problem than LP. To keep

Sales				
	1	2	3	4
1	10	1	9P(1)	21
2	5P(1)	5	16	26
3	5	10	25	40
4	20	16	50	86

Sales				
	1	2	3	4
1	D	D	9P(1)	21
2	5P(1)	D	D	26
3	5	10	25	40
4	20	16	50	86
1-LP				

Sales				
	1	2	3	4
1	D	1	9P(1)	21
2	5P(1)	5	D	26
3	5	10	25	40
4	20	16	50	86
2-LP				

Table 1: an Reduced Oversuppression Example

it a true linear programming, we simplify constraints (2) to

$$\begin{aligned} x^{+}_{ijk} &= \text{prot}_{ijk} \\ x^{-}_{ijk} &= 0 \end{aligned} \tag{3}$$

which force flows in one direction for all m Ps. However, replacing contraint (2) with constraint (3) has pros and cons which we discuss next.

To solve a cell suppression problem, m-LP solvers are expected to be called no more than $\#Total\text{P}s/m$ time. Ideally, using m-LP will reduce computation time to a fraction of m in a cell suppression process.

Although the simplified constraints (3) keep the complexity of m-LP the same as 1-LP, it should create more oversuppression than constraints (2) does because the former creates more needed protection. For example, in a relationship, where there are multiple Ps, constraints (3) ask for the amount of protection needed as $\sum_i \text{prot}_i$, while constraints (2) will negotiate among the Ps and may settle the needed protection with the amount diff$|x_i - x_j|$.

2.2 Example of reduced oversuppression

m-LP may have less oversuppression than 1-LP has. For example (Table (1)), when another P, which is able to provide protection to the targeted P, is not in the optimal path. To protect both Ps using 1-LP, 4 cells, (1,2), (2,2), (2,3) and (1,1), are selected . While using 2-LP, only 2 cells, (2,3) and (1,1), are selected.

2.3 Example of oversuppression and infeasible

Both oversuppression and infeasibility could occur when there are several Ps in a constraint on the mPs list. For example, in a constraint $a = b + c$, where both cell b and c are Ps. In 1-LP, P cells are encouraged to protect target cell, i.e., cell b maybe used to protect cell c, and vice verse. However, in m-LP, cell a has to be invoked to protect targets both b and c, because the given constraint, aflow = bprot+cprot, determines. When this happens oversuppression occurs, see Table (2) , a two-dimension example. Furthermore, if cell a is also a P then it is very

Sale				
	1	2	3	4
1	700	400	375P(25)	1475
2	1000	600	450P(30)	2050
3	100	375	650	1125
4	1800	1375	1475	4650

Sale				
	1	2	3	4
1	700	D	375P(25)	1475
2	1000	D	450P(30)	2050
3	100	375	650	1125
4	1800	1375	1475	4650

1-LP

	1	2	3	4
1	700	D	375P(25)	1475
2	1000	D	450P(30)	2050
3	100	D	D	1125
4	1800	1375	1475	4650

2-LP

Table 2: an Oversuppression Example

unlikely the equality (aprot = bprot + cprot) holds - a conflict constraint, which causes infeasible. Generally, a feasible flow that protects the Ps jointly also is a feasible flow that protects them separately, but the union of two feasible flows that protect the Ps separately may not be feasible to protect the Ps jointly (the capacity constraints may be violated) . Thus the set of feasible flows for the joint problem is smaller than that of the separate problem - causing oversuppression.

In reality, when c_b and c_c can protect each other, no additional cell will be suppressed. But m-LP asks for aggregate protection of both cells, c_b and c_a, which is $prot_a + prot_b$ and will take c_c for complimentary suppression. In this case, oversuppression is unavoidable, even we alternate the constraint for targeted Ps in opposite direction, i.e.,

$x_b^+ = 0$, and $x_b^- = prot\_required$, and $x_c^+ = prot\_required$, and $x_c^- = 0$

2.4 Dealing with Infeasibility

There are three different causes of infeasibility. One is caused by m-LP where some of m targeted Ps conflict each other, as explained in Section (2.2). Another is caused by freeze cell which, by its nature, has attribute of zero capacity[1]. The other is caused by a targeted P who associates with multiple tables, some of which have no other cells presented in the table relationship. For illustration purpose of the third cause, I have included a table, see Table (3).

This research is about m-LP, and m-LP induced infeasibility. The other two causes are not in this scope.

Can one avoid infeasible during the whole process? Probably not. We developed a way to cope with infeasibility for 2012 Economic Census by switching to m 1-LPs. The problem with this approach is it turns m-LP into m 1-LP, therefore it loses some of m-LP's time saving advantage. The proposal is to move the infeasible m-set to the end of list first. Then, at the end of process, reduce the infeasible m-set to certain number of smaller sets. Finally, if the smaller set is infeasible, switch to 1-LP.

[1]We adopt capacity as it is in network flow. A zero capacity cell has no contribution to other cells

Sale				
	1	2	3	4
0				1P(1)
1	10	1	9P(1)	21
2	5P(1)	5	16	26
3	5	10	25	40
4	20	16	50	86

Table 3: an Infeasible Example: row 0 has no other data

BASICL PAYANN				
NAICS	Total Cells	Total Ps	NAICS rels	NAICS
31	3524659	2677265	154	518
311	330366	264471	21	64
312	67580	54743	3	10
313	46767	38529	3	10
314	77735	59800	4	9
315	48402	38686	3	10
316	23820	19989	2	6
321	190961	146340	6	20
322	82458	65499	6	17
323	135991	84328	2	6
324	45118	37836	3	8
325	227568	182968	13	42
326	171501	131618	8	24
327	227752	179870	8	26
331	104999	88282	9	27
332	487476	348282	15	51
333	331171	258530	13	53
334	165052	127730	6	30
335	110574	91441	11	32
336	198445	163163	8	35
337	165140	119807	5	17
339	229942	163704	4	20
Total Geo Relation 5222				

Table 4: BASICL PAYANN Distribution

Do we know which are the conflicting cells? Yes we can find out , but is it worth the efforts? By moving the infeasible m-set to the end of list, we avoid the problem 'temporary'. But I expect the infeasible problem disappear by itself eventually. Because by ignoring it temporary, the process continues to next m-set. You may ask what happen at the end of list, remember we moved all the infeasible down the list. But the m-set is no longer the same set of m, for two reasons. First, some of the Ps no longer need protection (Skipped Ps), see reference [3]. Second, due to the nature of selection of m-set, m-queue is not evenly distributed. However, this proposal has not implemented yet.

3. Baseline Statistics

3.1 Data

We need to select test data. Table (4) gives a detailed breakdown of cell makeups, such as total cells, Ps, NAICS, and NAICS relations, for each NAICS catergory with the grand total on NAICS = 31. Cell total and number of NAICS define the size of model, and NAICS relation defines the complexity of the model. These information

Baseline Suppression		
SortBy	Cells	Value
random	16060	100808500
protAcen	15914	96762191
protDcen	16584	104096383
geo	16142	99530059
naics	16135	103594375
valDcen	16473	103594375
valAcen	15939	96794686
depth	15976	103594375

Table 5: Data Source: BASICL PAYANN NAICS-327; 1-LP Used

along with total Ps generally tell the processing time.

We are using one of the 3-d tables from economic census manufacture geographic area series (GAS), also called "basicl", which is a large data set and provides sufficient data structure for the research. The data consists of total eleven content variables, two of which are independent, three are in a relationship, and the rest are in one larger relationship, along with detailed NAICS and geographic location. We choose the three variable group, which is still too large with all detailed NAICS. We further reduced the size by choosing a subset of total NAICS - 327. The resulting test data provide the desired data structure, and large enough size, but computational suitable for research, see Table (4).

3.2 Baseline Statistics for 1-LP

We establish a baseline to evaluate oversuppression. The baseline uses 1-LP, but given a particular number n - the number of Ps, there are n! outcomes needed to run 1-LP multiple times to get an average and variance. However, we came up with 8 cases of different processing orders, see Table (5), with the 3-d table from the 2012 Economic Census as described in Section (3). The eight cases consist of one from random order, two from protection requirement sorted by descending or ascending, two from cell value sorted by descending or ascending, one by geographical code, one by NAICS, and one by depth[2]. These cases are a good representation of some of the best and worst scenarios which generally are used for testing. A more convenient approach is to establish baseline by all random cases. But we are unable to provide that due to software limitation.

The statistics are in Table (5) and (6). It shows suppression statistics from 8 cases both number of cells and total value. The means is compared with the suppression produced by some m-LP in the same order of processing.

4. m Ps Selection Algorithm

4.1 Cell Suppression Process

A cell suppression process, using m-LP solver, processes all Ps sequentially m number of Ps at a time until all protection needed Ps are fed into a m-LP solver once.

[2]This is a term used in production software. It captures the depth of a node from its root in a graph

Baseline Supression Statistics				
	Mean	std Dev	Minimum	Maximum
cells	16153	248.2	15914	16584
value	101,096,868	3,106,352	96,762,191	104,096,383

Table 6: Data Source:Table (5)

Suppression for m = 1, 5, 10, 20									
	m = 5			m = 10			m =20		
SortBy	Cells	Value	T(hrs)	Cells	Value	T(hrs)	Cells	Value	T(hrs)
random	16159	100936	2:20	16160	100894	1:18	16166	100939	1:07
protAcen	16115	99148	2:22	16129	99120	1:29	16137	98940	1:21
protDcen	16122	99997	2:18	16158	100127	1:29	16100	100623	1:25
geo	16018	99233	2:27	16020	98959	1:24	16026	99783	1:18
naics	16172	100308	2:25	16141	100101	1:28	16136	100159	1:30
valDcen	16084	99217	2:20	16137	101586	1:25	16114	99455	1:15
valAcen	16150	101445	2:23	16108	99442	1:22	16193	101473	1:46
depth	16081	99385	2:14	16114	99963	1:18	16148	99830	1:05

Table 7: Data Source: BASICL PAYANN NAICS:327 m-LP used, value are in thousand.

For this paper, m-LP is processed with the same order that produces baseline 1-LP.

4.2 Baseline Statistics for m-LP

We provide means for eight different processing order where m is 5, 10, and 20. Table (8) shows the suppression statistic with 1-LP result from earlier runs. Table (7) is the data source for the means. Comparing with 1-LP, each of m-LP results are improving in suppression. However we see huge improvement in processing time as predicated. Because of the number of infeasible encountered in the process, we are not surprised with 20-LP's run-time which supposed to be twice faster than 10-LP. When infeasible occurs, the production software sets back m=1 until all m cells are protected.

Suppression Means for m = 5, 10, 20			
m	Cells	Value	Time (hrs)
1	16153	101096868	8
5	16113	99959131	2.1
10	16114	99963959	1.17
20	16127	100150758	1.19

Table 8: Data Source: Table (7)

4.3 Results

Table (7) and (8) are some of the results that m-LP comparison with 1-LP. For all selection m, m-LP outperforms 1-LP in all aspects. For total value suppressed, there is more variability going down (order) than by m (across). There is reduced time as m increases, as expected. However we didn't see the time reduced by a fraction of m. There are two reasons. First, even it is clear that the m-LP model has the same complexity as 1-LP, we observed some increase of time for each solver. A complexity of a problem is generally defined in computation time by the problem size. Two problem having the same complexity generally refers the time used solve one is proportional to the time for the other. The second reason why the processing time isn't reduced as much as expected is because the filter imposed on the algorithm in our production software. When m targeted Ps are selected, the filter does a "feasible" check. The purpose is to avoid infeasible among the m cells. However, by filtering, it reduced the size of m Ps in the targeted set. As the result, the m-LP is a much reduced m model.

5. Future Research

There are several ideas for improving m-LP model. One is to better selection of m Ps to get better suppression pattern. Another is to handle infeasible by a more sufficient mechanism.

5.1 Selection of m Ps

Ideally, we like to select m unrelated Ps, so each P is locally isolated. In reality, data cells are all somehow related. We arrange the data based on its relationship in a data structure, picking m Ps from different "depth". We hope each P find protection from its own neighborhood with minimum interference with other Ps in the mPs list, therefore avoiding infeasibility and reducing oversuppression.

We feel that a far apart algorithm would minimizes both infeasible and over-suppression because Ps are loosely related, and each P may find its own protection with minimum interference from other Ps. It is much like protecting each P in its own neighborhood, but finding solution for m Ps at the same time. Therefore choice of Ps may reduce oversuppression.

Cell association makes LP a very big model which takes the program much longer time to process. A cell may associate with multiple tables. But it is very unlikely that more than one cell is associated with the same tables. We create a list of P cells and sort it by number of linked table in descending order. Within the structure, we further sort it by protection required and NAICS level combination of descending and ascending. It leads to 8 sorting possibilities. For example, AAA denotes all three fields are sorted in ascending order.

5.2 Handle Infeasible by Divide and Conquer

As we have explained at the end of Section (2.4). It would be more effective to move the infeasible set to end of its queue. If it comes to near the end of process, there are still some inevitable infeasible sets. For these last several sets of problem, we use divide and conquer. Divide each problem by its nth portion, for some n ¡ m/2. If the smaller set is still infeasible, then solve individually, by setting m = 1.

6. Conclusions

In General, m-LP outperforms 1-LP. m-LP is the solution to our Economic census' large data problem, BASICL in particularly. BASICL proved a challenge in 2012 Econ Census because of the size of the model and Ps. In 2012, We manipulated the model size by split the problem into several pieces, which created some other complications. But, we can't avoid the whole model completely. In the end, after the initial splitting process, we processed the whole to provide protection for remaining 87 Ps. The time took is 10 days! m-LP maintains the model size but hugely reduces the number of solver-calls. For an efficient m-LP, we need to be strategic in selection of m cells that are not conflicting each other. Because conflicting cells create infeasible and m-LP is set back to 1-LP. Therefore improving the m cells selection is a key to further improve running time.

References

[1] Wang B. Improve LP Process in Cell Suppression, Proceedings of the Government Statistics Section, American Statistical Association, Alexandria, VA (2013) CD-ROM

[2] Wang B. Disclosure Protection A New Approach to Cell Suppression. Proceeding of JSM 2008

[3] Steel P. et al Re-development of the Cell Suppression Methodology at the US Census Bureau, UNECE Ottawa, Canada, 28-30 October 2013

[4] Fischetti M., and Salazar-Gonzalez J-J. Combining Complete and Partial Cell Suppression Methodologies in Statistical Disclosure Control. Statistical Journal of the United Nations ECE 18 (2001) 355-361

[5] Fischetti M., and Salazar-Gonzalez J-J. Models and Algorithms for the 2-dimensional cell suppression problem in statistical disclosure control. Math. Program. 84:283-312(1999)

[6] Filipa D. C., Nico P. D., Margarida S. O. Statistical Disclosure in Two-Dimensional Tables Journal of the American Statistical Association. Vol. 89 (1994), No. 428, 1547-1557

[7] Massel P. Using Linear Programming for Cell Suppression in Statistical Tables: Theory and Practice. Proceeding of JASA, 8/2001

[8] Cox L.H. Suppression Methodology and Statistical Disclosure Control. Journal of the American Statistical Association 75: 377-385