

Constrained optimisation for tabular suppression in the Australian Bureau of Statistics (ABS)

Chris Mann* and Bernadette Fox**

* Australian Bureau of Statistics, chris.mann@abs.gov.au

** Australian Bureau of Statistics, bernadette.fox@abs.gov.au

Abstract: The ABS is implementing a transformation program which includes developing standardised services for data confidentiality. Traditionally employed methods within the bureau for secondary cell suppression are not entirely future proofed in terms of effectiveness, efficiency and adaptability. A constrained optimisation (CO) approach underpinned by the Fischetti-Salazar (2001) method is currently being explored. Consideration is being given to validation of the suppression results, suppression quality measures and dealing with complex tables. The current recommendation for implementation includes the CO method and modular method as approaches to determine secondary suppressions, and a tabular audit system as part of a larger suite of quality measures. Initial attempts to implement the CO method using the commercial solver Gurobi and *A Mathematical Programming Language* (AMPL) have demonstrated that the method is feasible. Additional work is required to look into extensions of the method as proposed by Fischetti and Salazar, particularly in identifying a stronger set of constraints to improve the efficacy of the final model.

1 Confidentiality and tabular suppression in the ABS

For the majority of its economic collections, the ABS uses suppression as its primary method of confidentiality for published tables. After identifying and suppressing cells that may breach disclosure policy (primary suppressions), additional suppressions must be found to prevent the primary cell's value being calculated (secondary or consequential suppressions).

Currently, suppression is either applied manually via a program such as SAS or Excel, or using the Disclosure Avoidance Analysis System (DAAS).

A glossary of terms used is available in Appendix 1.

1.1 Transformation agenda

The ABS is undergoing a major transformation. The volume of data and the way these data are acquired and disseminated are changing more rapidly than ever before.

Technology and innovation have advanced what is possible with official statistics. Meanwhile social, economic and environmental policy-making has become much more complex and intertwined. The ABS has taken stock of its systems, processes and

work practices to be in the best possible position to continue to provide trusted official statistics that will help enable a brighter future for the Australian community.

The ABS is implementing a wide reaching transformation program, supported by a new structure and governance arrangements, and substantial Government investment in modernising our infrastructure. The Statistical Business Transformation Program (SBTP) has been established and is well underway to implement this important body of work and ensure continued delivery of trusted, world class statistics.

1.2 Transforming confidentiality

One of the initiatives of the SBT Program is to develop standardised services for data confidentiality. The goal for the service, as for other transformation infrastructure, is that it is effective, efficient and adaptable.

1.2.1 Current state

Tabular suppression in the ABS is predominantly reliant on the program DAAS. The method for selecting secondary suppressions in DAAS is based on graph theory, and specifically network flow through directed graphs (Gusfield, 1988).

This method is only computationally feasible for 2-dimensional tables with marginal and grand totals, and so additional procedures were developed in DAAS to handle higher dimensions, and non-linear relationships. While the method guarantees that cell values cannot be determined exactly, it does not guarantee against disclosing the value to within an arbitrary interval around the true value.

The DAAS program will not be supported as transformation statistical infrastructure due to high redevelopment and on-boarding costs.

1.2.2 Alternatives considered

In mid-2015 a project was initiated to replace DAAS with Tau-Argus. Tau-Argus is an off-the-shelf data confidentialisation package for cell suppression. Originally, developed by Statistics Netherlands, the Tau-Argus program software is now open source (as a result of a project partly funded by Eurostat). The program utilises statistical best practice methodologies in the identification of both primary cells requiring suppression and secondary suppression patterns. Tau-Argus uses the method by Fischetti and Salazar (2001), among others.

As a result of a number of methodological and systems issues the project was prevented from moving forward and delivering the planned deliverables by the due date. The primary issue was integration into the ABS IT environment and issues relating to Tau-Argus's ability to provide solutions for high dimensional tables.

The project was subsequently closed based on the determination that the outstanding issues were unlikely to be resolved in a reasonable timeframe.

2 Constrained optimisation (CO) suppression method

The method being considered as part of SBTP, as outlined by Fischetti and Salazar (2001), is a two-step process involving CO models. Many National Statistical Offices (NSOs) either use the model directly, or use it as the core of other models, including in the Tau-Argus software (Castro, 2012).

The model uses the following input data:

- The cell values, written as a single vector;
- Information on the linear relationships between cells expressed as a matrix;
- Information on publically known upper and lower bounds on each cell value (e.g. the default lower bound on a cell in many tables will be 0);
- The information cost (or utility loss) of suppressing each cell;
- A list of primary suppressions; and
- For each primary suppression, a set of parameters called protection levels representing the minimum requirement to consider the cell to be sufficiently protected.

The input values may be defined algebraically (e.g. set the cost of suppression equal to the cell's value or a function thereof) or clerically (e.g. set a cell's cost arbitrarily high to reduce the chance that it will be suppressed). Some of the values may be informed by policy or external information – for example, while unit record information is not directly used by the model, the protection limits could be based on the contribution of the largest unit in a cell.

The choice of methods used to identify primary suppressions and choose appropriate protection levels is outside the scope of this paper, but the primary method to identify unsafe cells is likely to be similar to either the current application of (n,k) dominance rules in the ABS, or future application of the p% rule.

The aim of the process is to find a suppression pattern that satisfies the protection levels, while maximising the statistical utility of the resulting table, which is represented by minimising the sum of the cost of the suppressed cells.

- If the cost is the same for every cell, the objective is to minimise the total number of cells suppressed; if the cost is related to the cell values, the objective is to minimise the total value of the suppressed data.
- There is scope to allow adjustment of the cost values to meet particular user requirements (e.g. choose to suppress cells with higher RSEs, or whose values are less “interesting” to key stakeholders), although in practice the aim would be to minimise the use of this capability to maintain consistency across collections and prevent users of the system getting overwhelmed with options.

In theory, any collection of statistics with linear or nearly linear relationships (i.e. where the sum of particular cell values gives another cell value either exactly or approximately) can be protected with this method.

- One of the simplest cases is a 2-dimensional table, e.g. an Australian grand total, plus marginal breakdowns by either broad industry or state, plus the full cross-classification of the two.
- A more complicated example could be where two tables are strongly correlated, such as turnover and gross operating profits, such that an attacker could borrow strength across the two tables, so the relationship between the two would be encoded into the initial data and the tables would be protected simultaneously.
- For periodical collections (e.g. quarterly business surveys), it may be possible to process a pair of level estimates at two time points and the movement between them simultaneously, by adding a suitable set of additional constraints to take into account the fact that any values that were previously published need to be considered “publically available” and hence not suitable for suppression. This extension of the method has not yet been tested, but it may be a useful research direction.

This method cannot solve the suppression problem for data with non-linear relationships such as ratios.

2.1 Method details

Given the input data, the method begins with a simplified (linear) version of the problem (called the *Master LP* or MLP), that initially contains a bare minimum of constraints, and lets a cell be “fractionally suppressed”, which allows for solutions to be found relatively quickly.

The solution to the MLP is tested against the protection limits using a second CO, called the *attacker sub-problem* or ASP. Each protection level *not* met defines a constraint to add to the MLP – these constraints represent a combination of cells that an attacker could leverage to calculate a protected value, and hence identify sets of cells that are likely to require suppressions to protect that value. In Fischetti and Salazar (2001), these are called *capacity constraints*.

After the capacity constraints are added to the MLP, it is re-solved and the new solution is again tested against the ASP. This process repeats until, eventually, the solution satisfies all of the protection levels. This means that a complete set of capacity constraints has been identified. The capacity constraints are then added to the “true” (integer) suppression model (in which cells are either fully suppressed or not suppressed at all), and this model is then solved to find the final suppression pattern.

A flowchart representation of the process is shown in Appendix 2, and a small example of the algorithm is outlined in Appendix 3.

2.2 Validating the suppression results

A complementary problem to finding a suitable suppression pattern is determining whether a given suppression pattern protects against disclosure. This would be particularly pertinent in cases where suppression patterns have been clerically modified in order to force the publication of key values.

For a table in the same format as the input data described previously, and a suppression pattern for that table, finding the upper and lower bounds of feasible values for a suppressed cell can be expressed as pairs of linear programming constrained optimisation problems.

Comparing the interval of feasible values against the required protection levels for a primary cell identifies whether or not the cell has been sufficiently protected, and serves as an audit of the suppression pattern.

This model is in fact the same one used in the first step of the CO suppression method to identify additional model constraints, and so minimal additional coding is required to develop the audit system.

The audit model should be expected to perform more efficiently than the suppression model, and hence should be able to scale to larger tables, although there may still be practical limitations; the model also does not directly extend to data items with non-linear relationships. As such, it may be necessary to investigate a method analogous to the modular suppression algorithm to extend the audit model.

2.3 Suppression quality measures

Currently, there are few or no quality measures used to validate tabular suppression results in the ABS, with most collection areas relying on visual inspection and/or lengthy manual validation of tables.

Several measures are being considered as part of the implementation of the CO method. Examples include:

- Total cost of suppressed cells:
 - As an absolute value
 - Relative to some theoretical optimal value
 - Relative to the cost of suppressing the entire table
- Number and total value of suppressed cells:
 - As an absolute value
 - Relative to the size and total value of the table
- Information on whether any of the primary suppressions is not sufficiently protected (based on the output of the audit method); and

- Information on whether any of the protection intervals around a primary suppression is particularly large (based on the output of the audit method) – this helps measure how efficient the suppression pattern is, and identifies whether cells are being over protected (possibly due to another cell requiring particularly large amounts of protection). A particularly large protection interval may not be due to the performance of the algorithm but the structure of the table.

Additional measures may be identified during investigation and implementation, and some may emerge when developing the modular approach (below).

2.3.1 Dealing with more complex tables

Where the CO method does not scale well, it may be necessary to develop an alternative approach. One method to consider is based on the modular top-down approach (de Wolf, 1999).

In this method, a table (or collection of tables) is broken down into a set of covering sub-tables. Each sub-table should be small and/or simple enough that a suitable suppression pattern can be found quickly through some other method (e.g. Fishetti-Salazar (2001)). A cumulative list of secondary suppressions is built from the results of protecting each sub-table. Then, the cells identified on that list are suppressed on each sub-table they appear in, and the sub-tables are re-run through the suppression model. This repeats until no new suppressions are added to the list.

This method is similar to the approach used to protect multi-dimensional tables in DAAS, but it can use any method at the lowest level to determine the suppression pattern. It may also be modifiable to handle relationships such as ratios.

It is not guaranteed that the suppression pattern produced by this method is either optimal or provides complete protection against disclosure, but best practice guidelines can be developed to ensure that the resulting suppression pattern is fit for purpose in line with ABS policy, including making use of the audit model and other quality measures discussed in previous sections.

This method will be tested against true optimal solutions where possible to determine its robustness and level of protection.

3 Method implementation

The current recommendation for implementation includes the CO method and modular method as systems to determine secondary suppressions, and a tabular audit system as part of a larger suite of quality measures.

3.1 Benefits

3.1.1 Input data flexibility

The optimal and modular approaches can be combined to provide confidentiality across a wide range of statistical outputs, including data across multiple time points and various relationships between data items.

3.1.2 Use of common infrastructure

The CO method can be implemented using AMPL and Gurobi, which have been identified as core components of delivering constrained optimisation capability within the ABS. Corporate data and metadata repositories will also help realise input data preparation.

3.1.3 Model flexibility

The model can accommodate different cost values and protection levels, resulting in a variety of suppression outcomes. Empirical evaluations will be conducted to produce guidelines across survey areas on how cost values and protection levels should be set.

3.2 Issues for future consideration

3.2.1 Vulnerability to repeated attacks

Consequential suppressions are designed to protect specific cells in a specific table. If a table is changed slightly then the suppression pattern is also likely to change. In an externally-facing environment this means that a user could defeat suppression very quickly by requesting a small number of tables with common cells. One such environment is TableBuilder (Thompson, Broadfoot, Elazar, 2013), the ABS developed remote server system with automated confidentiality routines that allow users to build their own custom tables. Suppression cannot be implemented for such environments and care needs to be taken in dealing with custom data requests where suppression is used.

3.2.2 Unknown scalability

Fischetti and Salazar (2001) claimed that their method produced an optimal suppression pattern in short time for tables of up to three dimensions and tens of thousands of cells but began to show performance issues with one additional dimension. While their results were from 2001, running on a modest desktop system, and there is an expectation that a server-based implementation will perform better, it is still likely that there is a practical limit on the size and/or complexity of tables that can be feasibly solved with the CO method. The modular method represents one means of circumventing this issue.

3.2.3 Heuristic methods for sub-optimal solutions

If it turns out that finding the optimal solution to the suppression model is too slow a process, heuristic methods may need to be developed to choose solutions that meet the required protection levels but do not necessarily reach the optimal value.

3.2.4 Robustness of modular algorithm

Further empirical research will be conducted to identify how sensitive the modular algorithm may be to the order in which sub-tables are processed, and if that is the case then guidelines to standardise the algorithm will be developed to improve consistency.

3.2.5 Management and annotation of outputs

The data that come from the suppression and audit methods will feed into clearance and publication processes. In addition to implementing the methods, consideration needs to be given to how to best present the data to inform these processes, including annotations on cells where manual intervention or managerial sign-off has been required.

4 Conclusions

Initial attempts to implement the CO method using AMPL and Gurobi have demonstrated that the method is feasible.

There are still some bugs involved and it is unclear how well the method will scale to large, complicated tables. The current implementation requires the input data be put into a very specific format that is slightly cumbersome to produce from normal tabular data, particularly the matrix of linear constraints. This may be alleviated when the data are extracted from corporate repositories.

Additional work is required to look into extensions of the method as proposed by Fischetti and Salazar, particularly in identifying a stronger set of constraints to improve the efficacy of the final model. The issues outlined in section 3.2 will be considered as the method is realised in full.

References

- Castro, J. (2012). Recent advances in optimization techniques for statistical tabular data protection. *European Journal of Operations Research*, 216(2), 257-269.
- de Wolf, P.-P. (1999). *A heuristic approach to cell-suppression in hierarchial tables*. Voorburg: Statistics Netherlands Department of Statistical Methods.
- Fischetti, M., & Salazar, J. J. (2001). Solving the Cell Suppression Problem on Tabular Data with Linear Constraints. *Management Science*, 47(7), 1008-1027.

- Gusfield, D. (1988). A Graph Theoretic Approach to Statistical Data Security. *SIAM Journal on Computing*, 17(3), 552-571.
- Thompson, G., Broadfoot, S., & Elazar, D. (2013). Methodology for the Automatic Confidentialisation of Statistical Outputs from Remote Servers at the Australian Bureau of Statistics. *Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, Ottawa, Canada, 28-30 October 2013*, Working paper.

Appendix 1 - Glossary of terms

Term	Meaning
Attacker sub-problem (ASP)	A secondary model used in the Fischetti-Salazar method. The goal of the model is to calculate the upper or lower bound that an attacker could assign to a primary suppression, given a particular suppression pattern.
Capacity constraints	The constraints added to the cell suppression model from the attacker sub-problem. Each constraint approximately represents a combination of cell values that might allow an attacker to deduce the value of a primary suppression, and hence identifies a set of cells which is likely to require a number of suppressions.
Constrained optimisation	A field of mathematical science and computing that aims to find the values of a set of variables that minimise or maximise a particular objective function, subject to a set of constraints. For example, find the most valuable combination of objects to fit into a suitcase, given that each object has a particular value and size.
Constraint	In an optimisation model, an expression that defines a valid solution. For example, in a Sudoku puzzle, there is a set of constraints stating that a given row contains exactly one of each digit from 1 to 9.
Feasible value	A cell value that is consistent with the rest of the table. If a cell has only one feasible value, then that cell's value is essentially public.
Master LP (MLP)	The optimisation model built by the Fischetti-Salazar method. It starts out with no meaningful constraints, but is iteratively built up to include a set of capacity constraints that sufficiently define the suppression problem. The goal of the MLP is to minimise the total cost of (approximately) suppressed cells while providing sufficient protection against disclosure, as represented by the capacity constraints.
Protection level	In the cell suppression model, the closest that an intruder should be able to calculate some bound of a cell's true value. For example, if

the true value is 10 and the lower protection level is 3, then an intruder should not be able to calculate a lower bound on the cell's value that is higher than 7.

Primary suppression A cell in a table that has been identified as requiring protection, because it provides sufficient information to identify an individual unit's contribution. Its value is removed, or suppressed, from publication.

Secondary suppression A cell in a table that is suppressed in order to protect indirect calculation of a primary suppression cell's value.

Suppression pattern A set of suppressed cells in a table. The aim of the cell suppression problem is to find a suitable suppression pattern to protect against disclosure.

Appendix 2 – Constrained optimisation process flow chart

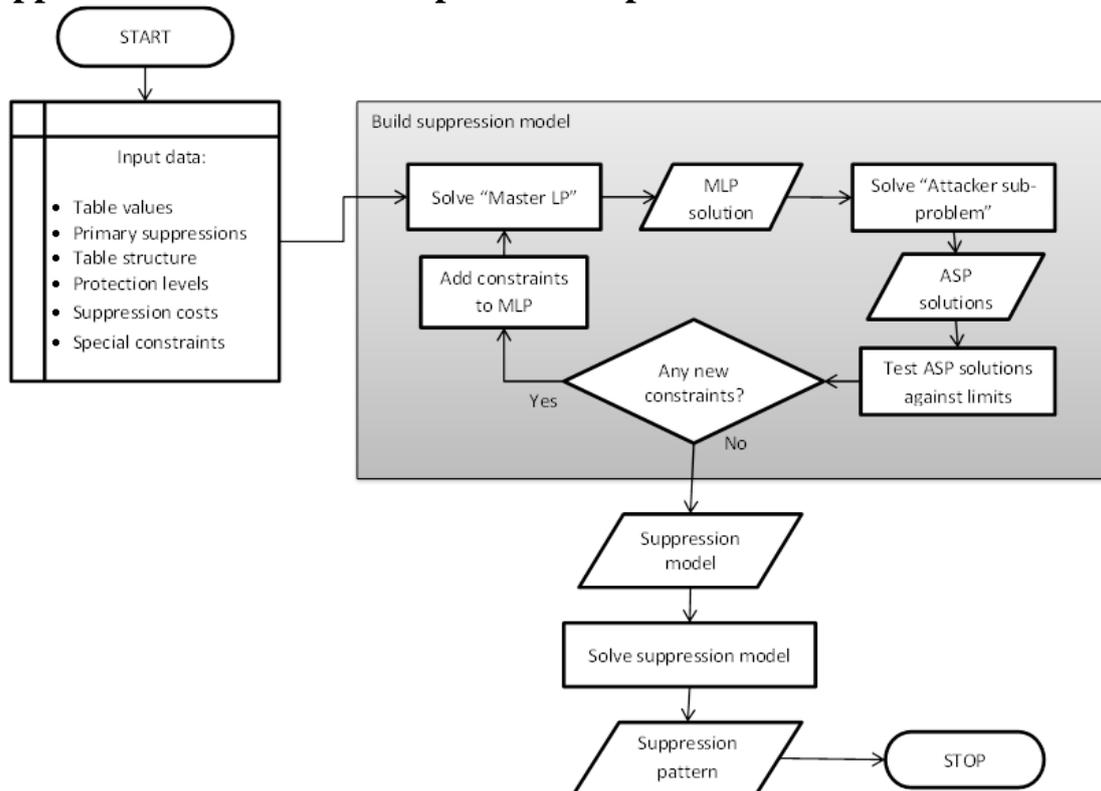


Fig 2.1 Model of the Fischetti-Salazar process.

Appendix 3 – A partial worked example of the Fischetti-Salazar method

The following is a description of how the Fischetti-Salazar method would work to identify secondary suppressions in a given table. Note that this is using the simplest form of the method, which requires a large number of iterations to complete. When implementing the method in practice, a number of improvements can be made – several of them suggested in Fischetti and Salazar’s paper – that introduce stronger constraints, or reduce the number of models to solve, or otherwise make the method more efficient.

Let the initial table be:

	Region A	Region B	Region C	Total
Industry 1	20	50	10	80
Industry 2	8	19	22	49
Industry 3	17	32	12	61
Total	45	101	44	190

Table 3.1 Initial table.

And label the cell indexes as follows:

	Region A	Region B	Region C	Total
Industry 1	A1	B1	C1	D1
Industry 2	A2	B2	C2	D2
Industry 3	A3	B3	C3	D3
Total	A4	B4	C4	D4

Table 3.2 Initial table indexed.

Suppose that the value in cell C2 (22) is sensitive, and needs to be protected within the interval [5, 30] – i.e. an attacker should be unable to put a lower bound on the cell’s value no higher than 5, and an upper bound no lower than 30. Additionally, by setting the “cost” of suppressing a cell as the cell value itself 22 “points” of information is lost

by suppressing C2 (for example). The objective therefore is to minimise the total value of all suppressed cells.

For the Master LP (MLP), the aim is to find the fractional suppression values s_{ij} for $i \in \{A, B, C, D\}$ and $j \in \{1, 2, 3, 4\}$. Initially, the MLP contains only the following:

$$\begin{aligned} &\text{Find } s_{ij} \in [0, 1] \text{ to minimise } \sum c_i s_i \\ &\text{Subject to:} \\ &(0): s_{C2} = 1 \end{aligned}$$

The solution to the initial model sets $s_{C2} = 1$, and $s_{ij} = 0$ for all other values (i.e. suppress cell C2 fully, and don't suppress anything else). In the attacker sub-problem (ASP), this gives us:

The lower bound on cell C2 is 22, higher than the target of 5: FAIL.

- Add constraint (1): $158s_{B3} + 20s_{A1} + 50s_{B1} + 49s_{D2} + 61s_{D3} \geq 17$ to the model.

The upper bound on cell C2 is 22, lower than the target of 30: FAIL.

- Add constraint (2): $146s_{C4} + 10s_{C1} + 12s_{C3} \geq 8$ to the model.

These constraints come from the dual coefficients of the ASP and approximately represent a combination of cells which currently provide enough information to allow cell C2 to be calculated.

With these new constraints, the solution to the MLP (a fractional suppression pattern) is as follows:

	Region A	Region B	Region C	Total
Industry 1	0	0	0	0
Industry 2	0	0	1	0
Industry 3	0	0	0.0837	0
Total	0	0	0.0479	0

Table 3.3 First solution to the Master LP.

Testing this new solution against the CSP gives the following results:

The lower bound on cell C2 is 22, higher than the target of 5: FAIL.

- Add constraint (3): $145s_{A4} + 89s_{B4} + 20s_{A1} + 50s_{B1} + 49s_{D2} + 17s_{A3} + 45s_{A4} \geq 17$ to the model.

The upper bound on cell C2 is 22, lower than the target of 30: FAIL.

- Add constraint (4): $173s_{A3} + 158s_{B3} + 10s_{C1} + 61s_{D3} + 45s_{A4} + 101s_{B4} \geq 8$ to the model.

After a number of iterations, a total of 19 capacity constraints are added to the model. The solution to the MLP looks like this:

	Region A	Region B	Region C	Total
Industry 1	0	0	0	0
Industry 2	0	0.4211	1	0
Industry 3	0	0.3306	0.0594	0
Total	0	0.0721	0.1459	0

Table 3.4 Revised solution to the Master LP.

When this solution is tested against the ASP, the lower bound on C2 is 5, and the upper bound as 30, which are the required limits. Hence, there are no new constraints to add to the model, and it has the following form:

Find $s_{ij} \in [0,1]$ to minimise $\sum c_i s_i$

Subject to:

(0): $s_{C2} = 1$

(1): $158s_{B3} + 20s_{A1} + 50s_{B1} + 49s_{D2} + 61s_{D3} \geq 17$

(2): $146s_{C4} + 10s_{C1} + 12s_{C3} \geq 8$

...

(19): $173s_{A3} + 50s_{B1} + 10s_{C1} + 19s_{B2} + 61s_{D3} + 45s_{A4} \geq 8$

Then taking the MLP with its full set of constraints, and add the requirement that s_{ij} be an integer (i.e. replace the restriction $s_{ij} \in [0,1]$ with $s_{ij} \in \{0,1\}$, requiring that each value be either 0 or 1). Solving this model then gives the final suppression pattern:

	Region A	Region B	Region C	Total
Industry 1	0	0	0	0
Industry 2	1	0	1	0
Industry 3	1	0	1	0
Total	0	0	0	0

Table 3.5 Final suppression pattern.

The optimal suppression pattern that satisfies all the capacity constraints is to suppress cells A2, C2, A3 and C3, giving the final publication table:

	Region A	Region B	Region C	Total
Industry 1	20	50	10	80
Industry 2	n.p.	19	n.p.	49
Industry 3	n.p.	32	n.p.	61
Total	45	101	44	190

Table 3.6 Final protected publication table.

And the value in cell C2 is sufficiently protected at a total cost of $8+22+17+12=59$.