Guaranteed Controlled Rounding for Many Totals in Multi-way and Hierarchical Tables

Gordon Sande

Sande & Associates, 10 Regency Park Drive #604
Halifax, Nova Scotia B3S 1P2
g.sande@worldnet.att.net (902) 443-2528

Abstract: Adjusting a column of rounded percentages to sum to 100 percent is often done. Rounding of entries in statistical tables of counts has been a standard tool for preserving confidentiality of respondent data. Rounding has the disadvantage that the tables may no longer add up as expected. Controlled rounding addresses this problem but in turn introduces new problems. The rounding is no longer to the nearest value which is a benefit for the confidentiality application. However controlled rounding may not be possible for hierarchical tables or for tables with three or more dimensions. Many heuristics have addressed these problems. A careful analysis of the structure of tables shows how to identify totals for which controlled rounding is always possible. The analysis shows that some choice is possible in the identified totals. The guaranteed rounding may often be extended to other totals depending upon the data in any example.

Introduction

The problem of publishing a column of percentages so that its sum is 100 percent is an old problem. Sometimes a column of rounded entries will sum to the rounded total but often it will not. The problem is present for the common deterministic rounding rules of rounding down, rounding to nearest or rounding up. Various practical schemes for solving the problem of controlling the rounding of entries to match the rounded total by using a mix of rounding up and rounding down can be constructed. The rounding mode would no longer be the same for all entries. These schemes can be extended to include columns with internal subtotals, also called hierarchical structures.

Statistical tables of counts are pervasive. Their release requires that the collecting agency take care to ensure that they limit statistical disclosures of the data of the respondents. A common technique for addressing identity disclosure in tabulations of counts is rounding of the entries. Rounding is typically done to multiples of five with two, three or ten sometimes used. Five seems to balance the competing requirements of being readily observed as the rounding base while not introducing excessive, but still adequate, ambiguity into small counts even while having little effect on large counts. The use of randomized rounding rather than rounding to the nearest multiple of the rounding base provides slightly more ambiguity with little other change beyond requiring attention to the problem of repeatedly randomly rounding the same count for differing publications. Rounding in all its various forms has the disadvantage that tables may no longer add up to their reported totals.

Rounding has many other applications, some of which may require either rounding down or rounding up rather than rounding to the nearest or random rounding. A table of integer values may be subjected to various adjustments yielding noninteger values although integer values are desired, so a final adjustment may be to round the values to be integers. Many applications of operations research to allocation problems can be viewed as the rounding of allocation fractions to be either zero or one. This note will use the methods of operations research at various points even while operations research uses both randomization and rounding as important techniques in finding solutions.

The problem of rounded tables not adding up to their reported totals has attracted considerable attention over time. The first solution proposed was for single rows or columns of data. A solution is to progressively accumulate the dropped fractional amounts and use that to increase the proposed valued before rounding. The rounding is no longer to the nearest value but rather to an adjacent value as is done with randomized rounding. This is an illustration of the need to carefully balance the form of the competing required properties of the rounding methods. The accumulation of fractional parts can be formulated in several different ways. Such a simple problem with a simple solution seems to make careful analysis an example of overkill in formalized formulation. The same problem arises in sampling with probability proportional to size where the size is the fractional part of the value (Fellegi 1975). Another formulation is to form a new sequence of partial sums of values. The partial sums are rounded to form a new sequence of whole values. Then a new sequence of whole values with a partial sums sequence matching the rounded partial sums is found by differencing (Sande 1977). This can be described as the sum, round and difference method.

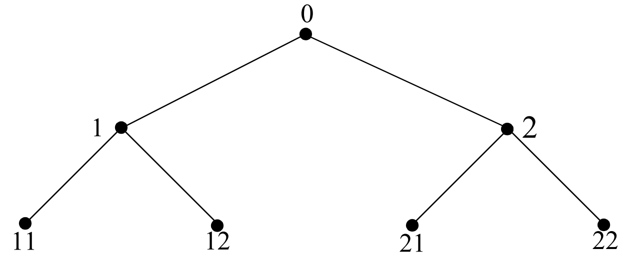
If each row of a two-way table is rounded to match its row total the new column totals may not be the rounded values of the original column totals. We need an algorithm for controlling the rounding in two-way tables. The sum, round and difference method can be generalized to two dimensions. The constructed values will have the property of correctly adding up to the row and column totals but they will not be rounded values as they may have excessive departures from the original values. For higher dimensions the deviations can be even greater. Cox (1987) proposed a solution to the rounding of simple two-way tables by use of a transportation problem from operations research. Transportation problems have a rich combinatorial structure which has drawn much attention over time. The immediate generalization of a transportation problem is a transshipment problem with the full generalization being known as a network problem. These generalizations to not extend to simple three-way tables or to two-way tables with hierarchical classifications both ways. A two-way table with one hierarchical classification corresponds to a transshipment problem. Examples showing that controlled rounding in general is not possible except for these particular cases, or equivalents, have been given. Notwithstanding the negative mathematical results based on very special tables many ordinary tables permit controlled rounding. There have been repeated development of heuristics to provide controlled rounding for the more general settings (Fagan 1988, others - see Bakker 1997 or Fischetti 1998) when they are present. None of these heuristics seem to have asked whether it might be possible to guarantee to control the rounding for an interesting, and hopefully large, subset of the totals present.

Structure of Tables

Hierarchies of Codes: We will need appropriate terminology to discuss the structure of multi-way and hierarchical tables. The difficulty is that there is little common terminology for the relevant notions so we need to list and define the terminology which will be used. The simple two-way table is often illustrated by a table of counts classified by age and sex. The entries in this table are denoted by the symbols $x_{i,j}$, the row totals by $x_{+,j}$, the column totals by $x_{i,+}$ and the grand total by $x_{+,+}$. This notation is natural if the total is regarded as separate from the other classification values. For many purposes it is natural to regard the total as just another classification value with zero as a common choice. This leads to the various entries of $x_{i,j}$, $x_{i,0}$, $x_{0,j}$ and $x_{0,0}$.

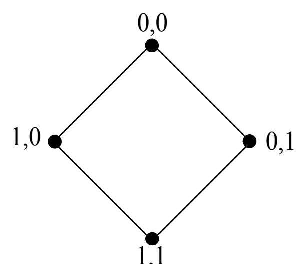
The distinction between dimension and hierarchy is often confusing and made worse by a lack of good, or even reasonable, notation. An example might be revenue sources for a car dealership. There would be revenue from car sales or from car servicing. The classification codes would be 0 for total revenue, 1 for car sales revenue and 2 for car servicing revenue. A more refined view would allow for revenue from new car sales and from used car sales as subcategories of car sales as well as vendor, such as warranty, car serving and owner car servicing as subcategories of car servicing. The classification codes would now be 11 for new car sales, 12 for used car sales, 21 for vendor car servicing and 22 for owner car servicing. There would be no need to combine the respective first sub-classifications as new car sales and vendor car servicing are not naturally combined, except by those seeking to demonstrate contorted linguistic meanings. These codes can be readily arranged into a hierarchical, or tree, structure. If one starts from the most refined code values it might appear natural to use 10 for car sales and 20 for car servicing. The convenience of describing 1 and 2 as first level codes and 11, 12, 21 and 22 as second level codes, with the level matching the number of digits in the code, quickly becomes apparent in practice. The minor extension to describe the code 0 as being at level zero is natural. Many illustrations of hierarchy tend to be of car or truck sales where the cars or trucks might be either red or blue. The distinction between dimension and hierarchy depth is poorly made and might even not make sense if there were some reason why red and blue vehicles required different painting procedures without regard to their being cars or trucks. A common statistical terminology is of nested or crossed variables for hierarchical or multidimensional variables with the nested variables examples often being of the red or blue colored cars or trucks variety.

A hierarchical classification variable need not have a uniform depth. Addition of notional entries to the hierarchy allows this practical problem to be dealt with easily. Another common practical problem are classifications that appear to be, and are largely, hierarchical but have alternate hierarchies present. There is a single level zero code, or root of the code tree, and a set of lowest level codes, or leave nodes of the code tree, but with a mixture of intermediate codes. This often arises in geographical codes where the codes involving counties will be a complete well formed hierarchical classification. The problem is that there may be codes for metropolitan areas which recombine the urban and rural municipalities without regard to the county combinations. The metropolitan codes may not be well formed as the non-metropolitan combination may not be explicitly defined although it will be implicitly defined by exclusion. This complication will not be explored or illustrated here.



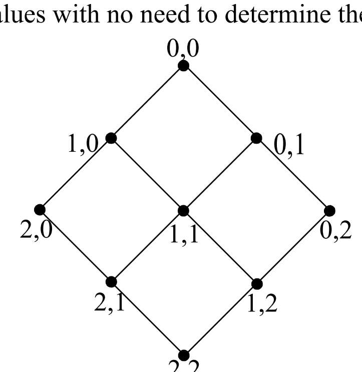
Simple Hierarchy of Depth 2

Lattices of Totals: Totals and subtotals come in many types. An orderly classification scheme for subtotals when there is both multiple dimensions and hierarchy will be helpful. In a simple two-way table there is a grand total, row totals, column totals and internal entries. The row classification will have a level 0 code and several level 1 codes when we use the hierarchy terminology in this simple situation. The column classification has a similar structure. From this we would say that the grand total is a total for the level 0 row and the level 0 column. The row totals would be for level 1 rows and the level 0 column. The columns totals would be for level 1 columns and the level 0 row. The internal entries would be the total, of only a single value, for the level 1 rows and level 1 columns. There is a single total of type 0 and 0, several of each of the types 0 and 1 or 1 and 0 and many of type 1 and 1. Treating the lowest level entries as being totals of themselves removes an irregularity in the definitions. This is typically structured as a lattice with a diamond shaped diagram.



Simple Two-way Lattice of Total Types

The 0 and 0 type is at the top, the 0 and 1 or 1 and 0 types are on the sides and the 1 and 1 type is at the bottom. The distance from the top indicates the degree of disaggregation from the grand total. The lattice nomenclature allows us to say that the types 0 and 1 or 1 and 0 are more disaggregated than the type 0 and 0 but not to say which of the two is more disaggregated as such a comparison is not meaningful for the lattice. There are two paths from the bottom to the top of this simple lattice. A path is a sequence of aggregations that tells us how to form the grand total starting from the internal entries. The path using the row subtotals is a formalized statement that we can determine the grand total by using the row totals as intermediate values with no need to determine the column subtotals.

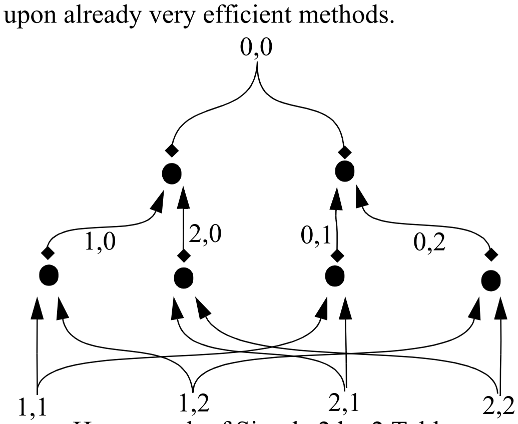


Hierarchical Two-way Lattice of Total Types

A more elaborate example is provided when each of two dimensions has a depth 2 hierarchical classification. We had an example above of seven codes with a depth two hierarchy. If both dimensions had such a structure there would be 49 values with only 16 values for the lowest level internal entries. There would be more than twice as many totals as lowest level internal entries. There will be nine types of totals of which one will be the lowest level internal entries.

The lattice diagram will be a three by three diamond of points on five levels, of distances zero through four. The points at distances zero and one would be like the simpler example, as would be the points at distances four and five. The three points at distance three are more interesting. These would be of types 0 and 2, 1 and 1 or 2 and 0. The totals of types 0 and 2 or 2 and 0 are either complete row or columns totals. The total of type 1 and 1 is a mixed total in which the classification variables are at their intermediate, or only partially disaggregated, level. This example also requires more elaborate table presentations. One presentation would be with seven rows and seven columns using embedded subtotals. For larger examples this would become impractical. From the lattice we see that there are four types of sub-lattices for simple tables. There would be nine simple tables of four types, as different types may have a different number of examples. The four totals of type 1 and 1 would each be displayed four times, as either an internal entry, a row subtotal, a column subtotal or a grand total of the four different types of simple tables. Other total types may be displayed a different number of times. There are six paths from the bottom to the top of this lattice as there are even more ways to construct the grand total than in the simpler example.

Hypergraphs of Totals in Tables: The representation of the additive structure of tables that has been used in many of the approaches to the controlled rounding problem is a network. A network has two components. One component is a set of nodes. The other is a set of arcs where each arc is a pair of nodes, one conventionally called the head and the other the tail of the arc. The arc is said to connect the two nodes. At each node there is an operation defined of combining the arc values by adding, with coefficient plus one if the node is the head of the arc and coefficient minus one if the node is the tail of the arc. If the combining equation has a nonzero constant the node would be called a source or sink node. There are many variations on network definitions as arcs may be of restricted sign or not, for example. One definition of generalized networks even allows for coefficients other than plus and minus one. There is much interest in network theory with considerable attention to algorithm efficiency in determining network flows. Networks are of great practical concern as they model physical flows with conservation laws, such as water, electricity or goods being transported, and have strikingly efficient algorithms. Rather we will use a related, but different, combinatorial structure which seems better suited to modeling information flows such as multi-way and hierarchical tables. The structure is the hypergraph. A hypergraph has two components. One is a set of nodes. The other is a set of hyperarcs where each hyperarc is a set of nodes with each node in the hyperarc labeled as either a head or a tail. The hyperarc is said to connect the nodes in its definition. A hyperarc may have only heads, a mix of heads and tails or only tails. At each node there is an operation defined of combining the hyperarc values by adding, with coefficient plus one if the node is the head of the hyperarc and coefficient minus one if the node is the tail of the hyperarc. Source and sink nodes are not usually defined as a hyperarc with a single head or tail has the same effect. There are many variations on hypergraph definitions with a common variation being a restriction of a hyperarc having either zero or one heads or a similar restriction for tails. One definition of generalized hypergraphs even allows for coefficients other than plus and minus one, so every equation system is this type of generalized hypergraph. Hypergraphs model the assembly process in which the same number of differing components are combined. Some applications of hypergraphs do not use the node operation as they model the connectivity that the hypergraph represents. Research in algorithms for hypergraphs addresses rather different concerns than research in networks which seeks to improve upon already very efficient methods.



Hypergraph of Simple 2 by 2 Table

A hypergraph model of a simple two-way table has hyperarcs for the internal entries. Each of these hyperarcs will have no tail and two heads. One of the heads will connect to a node for a row total and the other will connect to a node for a column total. There will be row total hyperarcs connecting each row total node, using a tail, to a combined row total node, using a head. There will also be column total hyperarcs connecting each column total node, using a tail, to a combined column total node, using a head. There will be a grand total hyperarc with a tail for the combined row total node and a tail for the combined column total node but with no head. There are two nodes yielding the grand total value to reflect the two formulae for the grand total. There is a hyperarc for each value and a node for each formulae. The multiple heads of a hyperarc allow for the same value to be used in different formulae. The multiple tails of a hyperarc allow the same value to be determined by different formulae

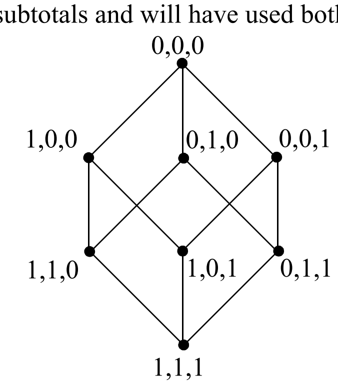
The hypergraph for a simple three-way table will be more complex. The lowest level will have hyperarcs with three heads and no tails. The first level of aggregation has only one set of defining formulae but may be used in two ways. There will be three groups of nodes matching the three heads on the lowest level hyperarcs. The hyperarcs for the first level of aggregation will have one tail and two heads. The second level of aggregation has two sets of defining equations for each of the three groups of intermediate totals for six groups of nodes. The third, or top, level of aggregation has three sets of defining equations. The three groups of hyperarcs leading to these three nodes will have two tails. The hyperarc leading from these three nodes will have three tails and no head. The multiple heads and tails of the hyperarcs allow us to construct this hypergraph but the single head and tail of a graph prevents us from representing the complete structure with a graph.

The lattice of totals indicates both how many alternate formulae there are for a total as well as how many uses there will be of a total in other formulae. These are number of tails for the hyperarcs to account of alternate formulae and the number of heads to account for the multiple use of the values.

Embedded Graphs for Tables: We have observed that a path through the lattice of totals is a specification of a sequence of aggregations to be formed to generate the grand total. This also specifies a subset of the hypergraph for the table which is a tree when we restrict the hypergraph to the specified nodes. This tree, or subhypergraph, will have the lowest level entries as its leaves and the grand total as its root. The internal nodes will include many, but not all except in the simplest of cases, of the subtotals of the underlying table. The leave nodes of a tree can be laid out as if they were a row of data. Controlled rounding of a row of data is a solved problem. The values of subtotals which were not selected as tree nodes can be reconstructed. The resulting table will add up but the reconstructed subtotals may not be correctly rounded. There are alternate paths through the lattice of totals, except for trivial examples, which lead to differing subsets of included, or controlled, and reconstructed, or not controlled, subtotals. This is not very interesting but serves to illustrate part of the notions.

It may be possible to choose a second path through the lattice of totals which does not overlap the first path except for the bottom and top of the lattice. If we label the first path *up* and the second path *down*, and interchange heads and tails in its hyperarcs, we will have a network which leads down from the grand total to the lowest level entries and back up to the grand total. Controlled rounding of a network is a solved problem. The nodes of the two trees will include many, and in simple cases all, of the subtotals of the underlying table.

For the simple two-way table, one of the trees will include the row subtotals and the other tree will include the column subtotals. This is all the possible subtotals and will have used both of the two possible paths.



Simple Three-way Lattice of Total Types

For a simple three-way table there are subtotals not included as well as some control over which subtotals are included. A simple four by four by four table has 64 internal entries and 61 totals and subtotals for 125 values in all. There are three sets of 16 one-way subtotals and three sets of four two-way subtotals and the three-way grand total. A path would include one set of one-way and one set of two-way subtotals. The second path can choose either remaining set of one-way totals and either remaining set of two-way totals. This means 41 totals and subtotals and 64 internal values are included and 20 subtotals are not included. We can choose which set of 16 one-way and which set of 4 two-way subtotals are to be not included and guarantee that the remaining values will be subject to controlled rounding. For a larger example of 10 by 10 by 10 there are 1000 internal entries, three sets of 100 one-way totals, three sets of 10 two-way totals and the three-way grand total for 1331 values in all. There will be 1221 values with guaranteed rounding and 110 not included subtotals.

When the not included subtotals are reconstructed some may be correctly rounded even though this was not guaranteed. The various solutions may also have differing patterns of correctly rounded reconstructed totals.

The two-way table with hierarchical classifications has a slightly different structure. For the simple case above there were 16 internal entries and 49 values in all for 33 totals and subtotals. There are six paths through the lattice of totals. There is no possible second path for two of the paths. For another two of the paths there is a unique second path. For the remaining two paths there a two possible second paths. There are six ordered pairs or three unordered pairs of possible paths. When we look at these we see that we have the choice of not including the subtotals of type 0 and 2, 1 and 1 or 2 and 0, each of which is at distance two from the top of the lattice. In each case we will have 16 lowest level entries and 29 subtotals controlled and 4 subtotals not controlled. We can choose which of the three sets of four subtotals to not include in the guaranteed controlled rounding. For a larger example for branching by two and then 10 for 23 codes along each dimension there will be 400 internal entries and 529 values in all with 129 totals and subtotals. We can choose which subtotals to not include with the size of the not included set being either 20, 4 or 20.

Cox and George (1989) study the example two-way table with hierarchical classifications by starting from a simple four by four two-way table and its network representation. They introduce subtotals into the network in a fashion which corresponds to adding points along the edges of the lattice of totals. Their construction does not lead to the intermediate crossed subtotal, the interior point in the three by three lattice, so they cannot control that subtotal or recognize the choices that are possible. Their construction does not gain flexibility when extended to more general settings. Both Kelly, Assad and Golden (1990) and Ring, George and Kuan (1997) use all nine embedded graphs of a simple three-way table but do not consider more elaborate cases.

Numerical Examples

Some of the problems in finding a controlled rounding can be illustrated by working through numerical examples. We have a simple two-way table with hierarchy in each dimension. Many cells are zero and the rounding base is ten. The example is specially structured to illustrates problems.

Original Table

	0	1	11	12	2	21	22
0	40	20	10	10	20	10	10
1	20	10	5	5	10	5	5
11	10	5	5	0	5	5	0
12	10	5	0	5	5	0	5
2	20	10	5	5	10	5	5
21	10	5	5	0	5	0	5
22	10	5	0	5	5	5	0

An immediate concern is whether a controlled rounding is possible. If we replace the *(11,11)* cell value by an unknown x, and treat the block totals as values to be determined, the row and column totals will allow the remaining nonzero values to completed by formulae involving x. Everything will be determined by the value given to x. Showing that a table is uniquely determined by its marginal totals and specified zeros is a common method of showing that a controlled rounding is not possible. A related 2 x 2 x 2 example in three dimensions has been often used to show that controlled rounding is not always possible in higher dimensions but is less suited for our purposes. Some examples are more readily followed than others.

Table with Unknowns for Cells and Block Totals

	0	1	11	12	2	21	22
0	40	20	10	10	20	10	10
1	20	**2x**	x	x	**20-2x**	10 - x	10 - x
11	10	x	x	0	10 - x	10 - x	0
12	10	x	0	x	10 - x	0	10 - x
2	20	**20-2x**	10 - x	10 - x	**2x**	x	x
21	10	10 - x	10 - x	0	x	0	x
22	10	10 - x	0	10 - x	x	x	0

The original table is the only possible table which matches the row, column and block totals so controlled rounding is not possible for this example. The rounded values for *x* are either 0 or 10 so the possible rounded values for the block totals are 0 and 20. One of the two possible cases is shown. Each block total will be adjusted by 10 for a total adjustment of 40. The block total adjustments are of alternating signs as the grand total remains unadjusted. Notice that the two possible block total values are separated by more than the rounding base. If the blocks were larger the separation would be larger and would be arbitrarily large with large enough blocks. The structure of the *(2,2)* block would be more evident with values only along the subdiagonal and in the upper right corner. Consistent permuting of the rows and columns would make the structure less evident.

We may also use the algebra to readily explore related examples. If the value for *x* had been 4 the possible rounded outcomes would have been the same. But the block total of 8 would have suggested a range of 0 to 10 as possible roundings. We notice that one of the possible outcomes is consistent with this suggestion so that the block total adjustments are in only one direction.

Table with Block Totals Adjusted for a Controlled Rounding

	0	1	11	12	2	21	22
0	40	20	10	10	20	10	10
1	20	**20**	10	10	**0**	0	0
11	10	10	10	0	0	0	0
12	10	10	0	10	0	0	0
2	20	**0**	0	0	**20**	10	10
21	10	0	0	0	10	0	10
22	10	0	0	0	10	10	0

We may choose to preserve the block totals and adjust either the row or column totals. For this highly symmetrically structured example the results would be similar. Two of the row totals are adjusted by 10 for a total adjustment of 20. One of four possible cases is shown. The row totals that remain unchanged differ in the four possible cases. This illustrates that the choice of which totals are guaranteed may influence the amount of adjustment required.

Table with Row Totals Adjusted for a Controlled Rounding

	0	1	11	12	2	21	22
0	40	20	10	10	20	10	10
1	20	10	10	0	10	0	10
11	10	10	10	0	0	0	0
12	10	0	0	0	10	0	10
2	20	10	0	10	10	10	0
21	**0**	0	0	0	0	0	0
22	**20**	10	0	10	10	10	0

We might choose to preserve all the totals and adjust cells. A single cell adjustment by 10 is adequate for this example. One of two possible cases is shown. This further illustrates that the choice of which totals to adjust and of even whether only totals are to be adjusted may influence the amount of adjustment required.

Table with a Cell Adjusted for a Controlled Rounding

	0	1	11	12	2	21	22
0	40	20	10	10	20	10	10
1	20	10	10	0	10	0	10
11	10	10	10	0	0	0	0
12	10	0	0	0	10	0	10
2	20	10	0	10	10	10	0
21	10	0	0	0	10	**10**	0
22	10	10	0	10	0	0	0

Solution Methods

Solving for a controlled rounding of a table requires a more detailed specification than that implicitly given by examples. The machinery of mathematical programming, or operations research, would start by associating a variable with each cell, whether internal or marginal, of the table. The hypergraph structure provides equations which define the aggregate values in terms of the internal values. We may assume that only one equation is used to define each aggregate by using only one of the alternatives when several are available. This has the effect of making the equations be of full row rank which is often assumed for ease of development in operations research. It is a minor technical complication to deal with the absence of this assumption in practice. Production quality linear programming codes provide artificial variables to deal with redundant equations. We must also specify the properties of the variables.

Controlled rounding leaves cells which are already multiples of the rounding base unadjusted. Other cells are adjusted up or down to the adjacent multiple of the rounding base. The rounded down value is the floor value $\lfloor x \rfloor$ where $\lfloor x \rfloor + f = x$ for $0 \le f < 1$ and the rounded up value is the ceiling value $\lceil x \rceil$ where $\lceil x \rceil = x + r$ for $0 \le r < 1$. $\lfloor x \rfloor \le x \le \lceil x \rceil$ provides a range of width $f + r$ for the variables. We will abuse notation by not using diacritical marks to distinguish between data and variables in the problem. The meaning should be clear from the context as we avoid the complications of a more elaborate notation. The width will be 0 when x is an integer and 1 otherwise.

Operation research has many techniques which use either rounding or variables which assume integer values. To match those practices it is convenient to divide the statistical table values by the rounding base to obtain operations research values which may yield integer valued solutions which are in turn multiplied by the rounding base to provide the desired statistical table answers. To combine the two usages it is convenient to have integer valued mean that the value is an integer multiple of the rounding base and fractional mean that the value is not an integer multiple of the rounding base. This will be a further abuse of notation.

With a set of variables subject to both a system of equations and ranges for the variables we have a bounded polyhedron. The original values will be within the polyhedron's boundaries. This indicates that the polyhedron is not empty. In the terminology of linear programming the original values are a feasible solution. They need not be a basic solution as there is no condition on the variables being at the limits of their ranges. Standard linear programming theory assures us that since there is a feasible solution there will be basic solutions as well. A basic solution will have many components, called basic variables in linear programming, at the limits of their ranges, or rounded. There will also be many components, called nonbasic variables, with values determined by the equation system, or basis, that will be within their permitted range. The nonbasic variables may be at their limits but that is neither prevented nor guaranteed. When we count equations and variables there will be as many equations as aggregates and as many variables as aggregates and internal cells. There are as many nonbasic variables as equations but the nonbasic variables need not be those of aggregate cells. There will be as many basic variables as internal cells but the basic variables need not be those of internal cells. The polyhedral theory only tells use that we can guarantee to round as many cells as there are internal cells.

In the special case that the equation system is that of a network we are guaranteed that the nonbasic values will be integer valued as well. This is the underlying mathematics of the solution of the controlled rounding of two-way tables. It is also why we were interested in identifying the embedded graphs in the tables. With the identification of the embedded graphs we may classify table cells as internal cells, aggregates defined on the identified embedded graph and aggregates not defined on the identified embedded graph. The internal cells and the graph aggregates form a network which has a controlled rounding. Rounded values of the internal cells define rounded, but not controlled, values for the nongraph aggregates. This is the starting point for the various proposed controlled rounding heuristics. A equivalent system could be set up by using the equation system we had used to define the polyhedron above, the variables and their ranges for the internal cells and graph aggregates and the variables, but with no restrictions, for the nongraph aggregates. This uses the technical fact that the unrestricted variables must become nonbasic and act to reduce the effective size of the equation system. The heuristics provide various *ad hoc* processing for the unrestricted variables although for practical reasons they would neither directly represent nor implement their processing in this format.

Rather than have the nongraph aggregates be unrestricted variables with *ad hoc* processing we could use them with a piecewise linear objective function. We would need three regions in the objective function. If the nongraph aggregate's variable value is less than the lower bound of the polyhedron range there would be a negative cost for the negative deviation, for values in the polyhedron range there would be a zero cost and for values greater than the upper bound of the polyhedron range there would be a positive cost for the positive deviation. We could write $x = x_N + x_Z + x_P$ where $x_N \le 0$, $l \le x_Z \le u$ and $x_P \ge 0$ where the polyhedron limits are l and u. The costs would be c_N, c_Z and c_P for an objective of $c_N x_N + c_Z x_Z + c_P x_P$. For piecewise convexity we require $c_N < c_Z < c_P$ which is realized if the cost are respectively negative, zero and positive. When x is below the lower limit l, x_N will be nonzero, x_Z will be at its lower limit l and x_P will be zero, and similarly in the other two segments. Such details would be handled internally within a piecewise linear code. We would let the cost associated with internal cells and graph aggregates be zero. The original values will yield an objective value of zero as will any feasible point of the polyhedron defined above. This objective value will be highly degenerate with many possible solutions but it may have fractional components as we have seen above. To address the degeneracy we would use the standard technique of small perturbation of the problem. We would let the cost of any variable with a fractional value be positive if it were near its lower limit and negative if it were near its upper limit. These perturbation costs would tend to nudge fractional values towards their rounded limits and might be repeatedly assigned as necessary. If the result is a solution with all internal cells at their limit values we will have found a desired controlled rounding which does not require any adjustment to the nongraph aggregates. In practice this a common outcome. The embedded graph and piecewise linear objective have been used to proceed directly to a controlled rounding when it does not require adjustment of the nongraph aggregates.

Linear programming theory assures us that the network variables can achieve their bounds. If the lower and upper bounds are the same, or the variable has a fixed value, then both bounds are achieved. If the bounds are unequal then the original feasible, but not basic, solution will be a convex combination of achievable basic values lying below and above the original solution. Those achievable values will be integer valued but obviously can not be outside the given bounds which are adjacent integers. So the achievable values must be the given bounds. But for the nonnetwork variable the achievable values may be outside the piecewise linear breakpoints as we have already seen in the numerical example.

When adjustment of the nongraph aggregates is required we would assign a large cost to one of the network variables to push it towards a limit and repeat the degeneracy breaking nudging with possible pushing of other variables if that is required. The cost used to push the variables would be large compared to the costs in the piecewise linear segments. To allow additional control of the solution sought we might temporarily modify the bounds to force particular network variables to achieve their bounds. The nonnetwork variables require a different approach as they are unbounded and their costs are specified by the piecewise linear objective. To check that a nonnetwork variable may achieve its lower bound an inequality constraint would be imposed. When the network variables have achieved their limits the lower bound will have been confirmed or an adjustment made.

To be able to document the additional error introduced into the data by the rounding procedure we would like to identify all the adjustments that might have been required. We have seen an example above in which a controlled rounding was possible with the suggested limits but other controlled roundings would required adjusted limits. The first step would be to ensure that all the network variables, that is the internal cells and graph aggregates, can achieve their bounds. The nudge and push scheme, with possible forcing, could be used. For the nonnetwork variables, that is the nongraph aggregates, forcing by constraint modification would be required. The constraint would force the nonnetwork variable to be at or beyond its limit. This is to allow for the case where adjustment will be necessary as the suggested limit is not possible for a controlled rounding.

The nudge and push scheme with piecewise linear constraints would be described in operations research terminology as a rounding of a continuous relaxation of an integer problem. The manipulation of the costs for the network variables is done to achieve the desired form of the solution as the value of the objective function is of little interest compared to the requirement that the constraints of being at the limits be met. The machinery of linear programming is being used more for constraint satisfaction than optimization of some objective function chosen in advance. The manipulation of costs to force constraint satisfaction is a standard method for Phase 1 of the simplex algorithm. When alternate patterns of adjustments are possible this method will find a pattern. The finding of alternate patterns is generally a hard problem and this is not exception to that rule.

Randomized Controlled Rounding

Cox (1987) gives an algorithm for constructing a randomized controlled rounding of a two-way table. The constructed rounding is unbiased as it is the sum of steps each with an expected value of zero. The algorithm has a familiar structure for students of linear programming as it is essentially the pivotal exchange for the primal simplex algorithm. Two obvious points of difference are in the choice of entering variable for the pivotal exchange and the requirement that the solutions should be both feasible and basic. If we rewrite the variables as $x = x_N + x_0 + x_P$ where $x_N \le 0$, x_0 is the original value and $x_P \ge 0$ then the original values will correspond to the basic feasible point $x_N = 0$ and $x_P = 0$. This is a representation often used with l_1 fitting with its absolute value objective. It is an example of a piecewise linear objective. The pivotal exchange does not depend on the choice rule for the entering variable beyond the technical requirement that the result should be a possible basis. Pivotal exchange has various applications for the simplex algorithm and in other areas. We would follow the common practice of subtracting out the original values to use $x' = x_N + x_P$ in computations. The limits would be similarly revised.

The realized randomized controlled rounding would depend upon the adjusted bounds obtained from the earlier computation. If no bounds were adjusted then the same result would have been obtained for all specifications of guaranteed subtotals. If there were adjustments then the realized controlled rounding will be conditional on both the choice of guaranteed subtotals and the adjustments identified. The earlier numerical examples have shown that the adjustment can depend upon the choice of guaranteed subtotals and that there may be alternate possible solutions for a single choice of guaranteed subtotals. Recall the example where guaranteeing column and block subtotals allowed two solutions for which row subtotals might be subject to adjustment.

We can follow a complete numerical example to show the working of the unbiased randomized controlled rounding. This example is two copies of our earlier example with slightly different numerical values to illustrate the effects of the nonnetwork aggregates. The analysis above finds that the block total of 10 must be adjusted to be a range of 0 to 20. When rounded to base 10 this example has four possible controlled roundings. The total of the first block will be 0, 20 or 40. The rounding with the 40 total is not consistent with the given total of 10 so we would exclude it as a possible result

Original Table for Unbiased Controlled Random Rounding

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	10	2	2	3	3	30	8	8	7	7
11	10	2	2	0	0	0	8	8	0	0	0
12	10	2	0	2	0	0	8	0	8	0	0
13	10	3	0	0	3	0	7	0	0	7	0
14	10	3	0	0	0	3	7	0	0	0	7
2	40	30	8	8	7	7	10	2	2	3	3
21	10	8	8	0	0	0	2	0	2	0	0
22	10	8	0	8	0	0	2	2	0	0	0
23	10	7	0	0	7	0	3	0	0	0	3
24	10	7	0	0	0	7	3	0	0	3	0

The unbiased rounding starts with an entering variable and a corresponding feasible direction. The first entering variable can be chosen as the *(11,11)* cell and a possible feasible direction might be

First Feasible Direction for *(11,11)*

	0	1	11	12	13	14	2	21	22	23	24
0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	1	1	0	0	-2	-1	-1	0	0
11	0	1	1	0	0	0	-1	-1	0	0	0
12	0	1	0	1	0	0	-1	0	-1	0	0
13	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
2	0	-2	-1	-1	0	0	2	1	1	0	0
21	0	-1	-1	0	0	0	1	0	1	0	0
22	0	-1	0	-1	0	0	1	1	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0

with the feasible multiplier of either -2 or 5. The probability of choosing -2 would be 5 / (2 + 5) and of choosing 5 would be 2 / (2 + 5). The results would be either

Table after Feasible Step by -2

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	6	**0**	**0**	3	3	34	**10**	**10**	7	7
11	10	**0**	**0**	0	0	0	**10**	**10**	0	0	0
12	10	**0**	0	**0**	0	0	**10**	0	**10**	0	0
13	10	3	0	0	3	0	7	0	0	7	0
14	10	3	0	0	0	3	7	0	0	0	7
2	40	34	**10**	**10**	7	7	6	**0**	**0**	3	3
21	10	**10**	**10**	0	0	0	**0**	0	**0**	0	0
22	10	**10**	0	**10**	0	0	**0**	**0**	0	0	0
23	10	7	0	0	7	0	3	0	0	0	3
24	10	7	0	0	0	7	3	0	0	3	0

or

Table after Feasible Step by 5

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	**20**	7	7	3	3	**20**	3	3	7	7
11	10	7	7	0	0	0	3	3	0	0	0
12	10	7	0	7	0	0	3	0	3	0	0
13	10	3	0	0	3	0	7	0	0	7	0
14	10	3	0	0	0	3	7	0	0	0	7
2	40	**20**	3	3	7	7	**20**	7	7	3	3
21	10	3	3	0	0	0	7	0	7	0	0
22	10	3	0	3	0	0	7	7	0	0	0
23	10	7	0	0	7	0	3	0	0	0	3
24	10	7	0	0	0	7	3	0	0	3	0

After a first step of -2 we would choose a second entering variable of *(22,22)* and a feasible direction of

Second Feasible Direction for *(22,22)*

	0	1	11	12	13	14	2	21	22	23	24
0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	0	0	1	1	-2	0	0	-1	-1
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	1	0	-1	0	0	-1	0
14	0	1	0	0	0	1	-1	0	0	0	-1
2	0	-2	0	0	-1	-1	2	0	0	1	1
21	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	-1	0	0	-1	0	1	0	0	0	1
24	0	-1	0	0	0	-1	1	0	0	1	0

with the feasible multiplier of either -3 or 7. The probability of choosing -3 would be 7 / (3 + 7) and of choosing 7 would be 3 / (3 + 7). The results would be either

Table after Feasible Steps by -2 and -3

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	**0**	**0**	**0**	**0**	**0**	**40**	**10**	**10**	**10**	**10**
11	10	**0**	**0**	0	0	0	**10**	**10**	0	0	0
12	10	**0**	0	**0**	0	0	**10**	0	**10**	0	0
13	10	**0**	0	0	**0**	0	**10**	0	0	**10**	0
14	10	**0**	0	0	0	**0**	**10**	0	0	0	**10**
2	40	**40**	**10**	**10**	**10**	**10**	**0**	**0**	**0**	**0**	**0**
21	10	**10**	**10**	0	0	0	**0**	0	**0**	0	0
22	10	**10**	0	**10**	0	0	**0**	**0**	0	0	0
23	10	**10**	0	0	**10**	0	**0**	0	0	0	**0**
24	10	**10**	0	0	0	**10**	**0**	0	0	**0**	0

with a probability of (5 / (2 + 5)) * (7 / (3 + 7)) = 5 / 10 or

Table after Feasible Steps by -2 and 7

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	**20**	**0**	**0**	**10**	**10**	**20**	**10**	**10**	**0**	**0**
11	10	**0**	**0**	0	0	0	**10**	**10**	0	0	0
12	10	**0**	0	**0**	0	0	**10**	0	**10**	0	0
13	10	**10**	0	0	**10**	0	**0**	0	0	**0**	0
14	10	**10**	0	0	0	**10**	**0**	0	0	0	**0**
2	40	**20**	**10**	**10**	**0**	**0**	**20**	**0**	**0**	**10**	**10**
21	10	**10**	**10**	0	0	0	**0**	0	**0**	0	0
22	10	**10**	0	**10**	0	0	**0**	**0**	0	0	0
23	10	**0**	0	0	**0**	0	**10**	0	0	0	**10**
24	10	**0**	0	0	0	**0**	**10**	0	0	**10**	0

with a probability of (5 / (2 + 5)) * (3 / (3 + 7)) = 15 / 70. After a first step of 5 we would choose a second entering variable of *(11,11)* and a feasible direction of

Second Feasible Direction for *(11,11)*

	0	1	11	12	13	14	2	21	22	23	24
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	-1	-1	0	-1	-1	1	1
11	0	1	1	0	0	0	-1	-1	0	0	0
12	0	1	0	1	0	0	-1	0	-1	0	0
13	0	-1	0	0	-1	0	1	0	0	1	0
14	0	-1	0	0	0	-1	1	0	0	0	1
2	0	0	-1	-1	1	1	0	1	1	-1	-1
21	0	-1	-1	0	0	0	1	0	1	0	0
22	0	-1	0	-1	0	0	1	1	0	0	0
23	0	1	0	0	1	0	-1	0	0	0	-1
24	0	1	0	0	0	1	-1	0	0	-1	0

with the feasible multiplier of either 3 or -7. Notice that this is a repeated use of this variable as an entering variable but the feasible direction is different. The results would be either

Table after Feasible Steps by 5 and 3

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	**20**	**10**	**10**	**0**	**0**	**20**	**0**	**0**	**10**	**10**
11	10	**10**	**10**	0	0	0	**0**	**0**	0	0	0
12	10	**10**	0	**10**	0	0	**0**	0	**0**	0	0
13	10	**0**	0	0	**0**	0	**10**	0	0	**10**	0
14	10	**0**	0	0	0	**0**	**10**	0	0	0	**10**
2	40	**20**	**0**	**0**	**10**	**10**	**20**	**10**	**10**	**0**	**0**
21	10	**0**	**0**	0	0	0	**10**	0	**10**	0	0
22	10	**0**	0	**0**	0	0	**10**	**10**	0	0	0
23	10	**10**	0	0	**10**	0	**0**	0	0	0	**0**
24	10	**10**	0	0	0	**10**	**0**	0	0	**0**	0

with a probability of (2 / (2 + 5)) * (7 / (3 + 7)) = 2 / 10 or

Table after Feasible Steps by 5 and -7

	0	1	11	12	13	14	2	21	22	23	24
0	40	40	10	10	10	10	40	10	10	10	10
1	40	**20**	**0**	**0**	**10**	**10**	**20**	**10**	**10**	**0**	**0**
11	10	**0**	**0**	0	0	0	**10**	**10**	0	0	0
12	10	**0**	0	**0**	0	0	**10**	0	**10**	0	0
13	10	**10**	0	0	**10**	0	**0**	0	0	**0**	0
14	10	**10**	0	0	0	**10**	**0**	0	0	0	**0**
2	40	**20**	**10**	**10**	**0**	**0**	**20**	**0**	**0**	**10**	**10**
21	10	**10**	**10**	0	0	0	**0**	0	**0**	0	0
22	10	**10**	0	**10**	0	0	**0**	**0**	0	0	0
23	10	**0**	0	0	**0**	0	**10**	0	0	0	**10**
24	10	**0**	0	0	0	**0**	**10**	0	0	**10**	0

with a probability of (2 / (2 + 5)) * (3 / (3 + 7)) = 6 / 70. We add the earlier probability of this configuration to have a final probability of 15 / 70 + 6 / 70 = 21 / 70 = 3 / 10. The three configurations have probabilities of 2 / 10, 3 / 10 and 5 / 10 so the controlled rounding will be unbiased. If a different feasible direction had been used in the first step then the intermediate results would have been different but the final results would be the same.

For larger or more complex examples we may not readily find directions that allow this form of the method to succeed. The failure is analogous to a sequence of pivotal exchanges finding fractional values. In the network case this failure can not happen. Heuristics, or guidance from other solution techniques, may provide fewer failures than an unguided pivotal sequence.

A variation on this method is to use a direction that provides small changes to many variables. An illustration of the notion can be provided by a structured one-way example.

Original Table

10	2	2	2	2	2

We want to have a full modification of the first cell and small modifications of the other cells. A possible direction is

Direction - Many small changes

0	1	-0.25	-0.25	-0.25	-0.25

which permits steps of either a positive step of 8 or a negative step of 2. We would randomly choose one of these to have either

Modified Table - Step of 8

10	10	0	0	0	0

or.

Modified Table - Step of -2

10	0	2.5	2.5	2.5	2.5

The rounding will either be completed at the first step or we will be left with a revised table with a structure similar to the original table. The cell which will have its value fixed by the rounding step is the one with the full modification. This allows a more direct control over which cells are to have their values determined at each stage of the rounding process. This is an improved heuristic based on an improved rule for determining the modification direction. Even this may fail.

In the presence of degeneracy even finding a search direction can be difficult. It may be necessary to express the direction finding problem as a related mathematical programming problem to be solved at each stage of the unbiased controlled rounding algorithm.

When the heuristics or backtracking and retries prove ineffective the search for an unbiased controlled rounding may be cut short. A random, but not known to be unbiased, controlled rounding can be found by the nudge and push scheme with the initial nudge direction assigned randomly.

Relaxations and Extensions

Relaxations: Several generalizations have been suggested. One type of generalization deals with cells that are already rounded. These generalizations relax the conditions and permit additional values. One suggestion, called relaxation up, of permitted values is the existing floor value and a possible adjustment up by the rounding base. The upper limit is the floor value $\lfloor x \rfloor$ plus 1. $\lfloor x \rfloor \leq x \leq \lfloor x \rfloor + 1$ provides a range of width 1. To show that the network variables can obtain their limits we introduce a small positive perturbation to the internal values that are already rounded. The perturbation would be small enough that no aggregate variable would have to exceed it bound. In the best mathematical tradition this reduces to the previous case. The previous analysis of the required adjustments to the nonnetwork aggregates requires no modification. The relaxation is suggested to make it easier to find controlled rounded values. An solution of our previous example with some of the totals relaxed up is

Table with Totals Relaxed Up for a Controlled Rounding

	0	1	11	12	2	21	22
0	**50**	20	10	10	**30**	10	**20**
1	20	10	10	0	10	0	10
11	10	10	10	0	0	0	0
12	10	0	0	0	10	0	10
2	**30**	10	0	10	**20**	10	10
21	10	0	0	0	10	0	10
22	**20**	10	0	10	10	10	0

These values are all within the variable bounds so no adjustment of nonnetwork aggregates is suggested. When the analysis is carried further we notice that the *(0,0)* total does not take on its lower limit of 40 unless there are adjustments made to some of the nonnetwork totals. Relaxation up has been successful in permitting more controlled roundings to be readily found but it does not eliminate the need for adjustments. The perturbation is of the same size for all controlled cells independently of whether they are initially rounded or not. Unbiased controlled roundings are not meaningful for relaxation up so we would only seek to produce a random controlled rounding after the analysis for adjustments.

Another suggestion, called relaxation down, of permitted values is the existing ceiling value and a possible adjustment down by the rounding base. The lower limit is the ceiling value $\lceil x \rceil$ minus 1. $\lceil x \rceil - 1 \le x \le \lceil x \rceil$ provides a range of width 1. This does not seemed to have been previously suggested although it is a symmetric analogue of relaxation up. The analysis is as for relaxation up with the obvious changes of plus to minus.

Another suggestion, called relaxation up and down, is to permit the values from either relaxation up or relaxation down. The upper limit is the floor value $\lfloor x \rfloor$ plus 1. The lower limit is the ceiling value $\lceil x \rceil$ minus 1. $\lceil x \rceil - 1 \le x \le \lfloor x \rfloor + 1$ provides a range of width 1 for values not already rounded and a width of 2 for already rounded values. The presence of the intermediate, or central, value can be important. The example

Original Table for Relaxation Up and Down

20	10	10

with relaxation up and down of

Bounds for Relaxation Up and Down

10 - 30	0 - 20	0 - 20

requires the use of the central value for many of its controlled roundings. The separate analyses of relaxation up and relaxation down apply to show that the network variables can achieve their limits but this is incomplete for the analysis of the nonnetwork aggregates. The analysis can be achieved by using separate variables for the relaxation up and relaxation down in one analysis. The technique of using additional variables for partial ranges of an otherwise single variable is used for the piecewise linear objective functions above and is a standard operations representation of variables which may take both positive and negative values. Relaxation up and down permits even more controlled roundings to be readily found.

All the relaxation forms may have a variation to recognize that a value of zero may be present for structural reasons and should not be subject to possible modification. The variation is called zero restricted. Notice that the standard form without relaxation is already zero restricted.

An alternate notation that is sometimes used is $[x]$ for truncation towards zero. A common definition is $[x] + f = x$ where $f = 0$ when $x = 0$, $0 \le f < 1$ when $x > 0$ and $0 \ge f > -1$ when $x < 0$. The symmetry properties of truncation towards zero are more suited for some mathematical applications than are those of either the floor or ceiling functions. $[x] \le x \le [x] + 1$ provides the same range of values as relaxation up when x is zero or positive. For negative values there should be a close examination of definitions and problem statements as this range does not match the usual notions of rounding. Sometimes there is a further technical restriction that the absolute difference between the original and rounded values should be strictly less than one to remove the relaxation. It is easy to overlook this subtle technical condition and it may lead to errors in citations from original sources.

Extensions: Most extensions would be based on no relaxation but combined forms of extension with relaxation would be possible. We have seen that the adjustments to the uncontrolled aggregates will be of both signs as the grand total is controlled. For some allocation problems it may not be acceptable to reduce a subtotal. This requires an alternate formulation for another variant on, or extension to, the rounding problem. The new requirement would be that any adjustment to an aggregate can only increase the value. A symmetric alternative would be to only permit any adjustment to an aggregate to only decrease the value. We have shown one possible set of one sided adjustments, or only increase subtotal values, above as one possible solution under relaxation up.

We see that one of the cross totals has been increased along with one row total, one column total and the grand total. The adjustments are not just in the noncontrolled totals but in all types of totals. Other solutions for this table will have different patterns of adjustments but the adjustments will still be to all the types of totals. The equation structure needed to implement the one sided adjustment conditions is different than the earlier equation structure. The noncontrolled aggregates will now have a lower bound equation to impose the condition that the adjustment can only be an increase in value. The piecewise linear objective will still be present although the negative adjustment segment will be inactive. The controlled aggregates will have a lower bound as they had before but their upper bound will have become only a break point in a piecewise linear objective as otherwise a strict bound may lead to no solution being possible. There is no difference in the equation structure of the noncontrolled and controlled aggregates. The important consequence of this is that we no longer have the network structure for the equations. The notion of controlled totals does not work with one sided adjustments. This would be a highly structured allocation problem more typical of operations research applications.

The changed equation structure will require adaptations to the solution technique. For the adjustment up case we are no longer assured that the lower bound is achievable. A very large cost would be used in lieu of a constraint in the manner we have seen above. The lower bound will be adjusted up when it can not be achieved. The option of modifying the constraints is not available for the potentially nonachievable bound. The initially specified lower bound would be adjusted up if it could not be achieved with all aggregates possibly subject to adjustment. In the example above the *(0,0)* total is adjusted up to fixed value of 50.

Discussion

A common response to examples of impossible cases in mathematics is to ask if there is a modification, hopefully minor, which will render the example possible. The names of irrational and complex numbers are the historical legacies of problems which were once thought impossible. Controlled rounding is a question which arises repeatedly for which a well formed mathematical response to what is possible is a reasonable expectation. The presence of additional totals makes some controlled rounding examples impossible so we can view the absence of a controlled rounding as having defective additional subtotals. The fix is to cure the defect by adjusting the defective subtotals. Part of the fix is to determine which totals are extra and potentially defective. Another part of the fix is to document the adjustments introduced by the condition of requiring a controlled rounding. The adjustment will be the additional noise introduced by controlling the rounding.

Hypergraphs provide a model for multi-way and hierarchical tables that is very natural and deals with all cases of multi-way and hierarchical structure. The hypergraphs can be seen to be equivalent to the graphs used in the case of simple two-way tables.

Algorithms for determining hypergraph flows rely in linear programming methods. One class of methods is to view the problem as a linear programming problem which may include the recognition of embedded networks. The use of a piecewise linear objective function allows the embedded networks to be easily used as we may choose network costs. The result is that controlled roundings are found directly when they naturally exist. This is to be contrasted with the various heuristics which start with a network solution and try to revise it to be a controlled rounding. When a controlled rounding does not naturally exist the piecewise linear objective function acts both as a soft constraint and as a measure of the amount of adjustment required.

The need for rounding in statistical tables is often based on the need to provide disclosure limitation. Rounding addresses identity disclosure by making the size of identifiable groups inexact. The need for controlled rounding is often based on the failure of rounded tables to add up as expected. Small failures of percentages to add up are widely observed and generally tolerated. Rounding in statistical tables produces relatively small errors when the counts are not small so are generally tolerated. For small counts the rounding is a visible indication that some form of confidentiality protection has been applied so the rounding is again tolerated. Identity disclosure in the lowest level cells of a publication pattern is a mainly a cosmetic problem that data providers would prefer to avoid. Lowest level cells can become marginal cells if there is further disaggregations and they are no longer the lowest level cells. Identity disclosure in marginal entries leads inevitably to attribute disclosures which are to be avoided. Unfortunately attribute disclosures can be present even if there are no identity disclosures. The technical deficiencies of rounding in dealing with statistical disclosure make it a rather incomplete solution to the problem. Rounding does serve of a role in providing a noise source to render small counts unreliable and as an indicator that some disclosure limitation has been applied. The presence of rounding is obvious from the numbers and requires no further documentation, which has a tendency to lost when the contents of the tables are repeatedly transferred to further users.

Statistical tables also have technical uses where disclosure control may not be concern. Repeated adjustments of tables that are expected to be whole values may lead to fractional values. Rounding will restore the whole values. In the technical applications the need for controlled rounding may also arise. The technical applications may be as diverse as sampling factors after matching control totals or economic statistics adjustments after benchmark revisions. Controlled rounding has diverse applications in allocation where there are nominally fractional units to be converted into practical whole units. Some allocation applications must allow for conditions which translate into additional dimensions, hierarchical classifications or one sided adjustments.

The original lower and upper limits for the variables may permit no controlled roundings and require adjustments, may permit some controlled roundings and require some adjustments or may permit controlled roundings and require no adjustments. The middle case can also arise if not all the adjustments have been applied for some reason. The actual limits that operate for the variables are tighter than the specified limits when adjustments are required. The controlled roundings permitted are only a subset of the controlled roundings possible with all the adjustments applied. Not using all the possible controlled roundings means that a selection of one controlled rounding by either a deterministic procedure, such as fitting procedure, or a procedure with a random element will have some form of bias. Restricting the range of possible solutions in fitting may lead to lower quality solution.

The failures of the unbiased controlled rounding procedures to terminate with a controlled rounding are analogous to the failures of some integer programming resulting in fractional values. We would expect improved methods in integer programming to suggest better methods for unbiased controlled rounding.

Conclusions

Controlled rounding of statistical tables is of interest but the mathematics shows that it is not possible in general for other than quite simple table configurations. An immediate question is whether a usefully large subset of a general table configuration can have controlled rounding. This note shows that network based methods can be extended to extensive identifiable subsets of all totals in all cases with some degree of control of the contents of the subset with guaranteed controlled rounding. The lack of strict control is limited to subtotals outside this subset. Often the strict control is possible but in highly structured cases the departure from strict control can be very large. The extent of the lack of control can be determined for each example and would be part of the statistical documentation of any dataset.

References

Bakker, T. (1997) *Rounding in Tables*, student Project Report, Mathematics, Leiden University, Netherlands.

Berge, C. (1970) Graphes et Hypergraphes (French), (1973) Graph and Hypergraphs (English).

Cox, L. H. (1987). *A constructive procedure for unbiased controlled rounding*. Journal of the American Statistical Association 82, 520-524.

Cox, L. H. and J. A. George (1989) *Controlled rounding for tables with subtotals*. Annals of Operations Research 20, 141-157.

Fagan, J. T. and B. V. Greenberg (1988) *Controlled Rounding of Three Dimensional Tables*, Technical Report, Statistical Research Division, USBC.

Fellegi, I. P. (1975) *Controlled Random Rounding*, Survey Methodology 1, pp 123-133.

Fischetti, M. and J-J. Salazar-Gonzalez (1998) *Experiments with Controlled Rounding for Statistical Disclosure Control in Tabular Data with Linear Constraints*, Journal of Official Statistics 14, 553-565.

Kelly, J. P., B. L. Golden and A. A. Assad (1990) *Large-scale Controlled Rounding using TABU Search with Strategic Oscillation*, Annals of Operations Research 41, 69-84.

Ring, B. J., J. A. George and C. J. Kuan (1997) *A Fast Algorithm for Large-scale Controlled Rounding of 3-Dimensional Census Tables*, Socio-Economic Planning Science 31, 41-55.

Sande, G. (1977) *Alternate Formulation of Controlled Random Rounding with Generalizations*, internal note, Statistics Canada.