WP. 32
ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE
EUROPEAN COMMUNITIES (EUROSTAT)

Joint UNECE/Eurostat work session on statistical data confidentiality
(Geneva, Switzerland, 9-11 November 2005)

Topic (v): Confidentiality aspects of tabular data, frequency tables, etc.

CONFIDENTIALITY PROTECTION BY CONTROLLED TABULAR ADJUSTMENT USING METAHEURISTIC METHODS

Invited Paper

Submitted by the U.S. National Center for Health Statistics, United States of America[1]

[1] Prepared by Lawrence H. Cox.

Confidentiality Protection by Controlled Tabular Adjustment Using Metaheuristic Methods

Lawrence H. Cox[1]

[1] National Center for Health Statistics, Hyattsville, MD 20782, USA (lcox@cdc.gov)

Abstract. Controlled tabular adjustment is an SDL methodology based on a mixed integer linear programming model. We develop new hybrid heuristics and new meta-heuristic learning approaches for solving this model, and examine their performance. Our new approaches are based on partitioning the problem into its discrete and continuous components, and first creating a hybrid that reduces the number of binary variables through a grouping procedure that combines an exact mathematical programming model with constructive heuristics. We then replace the MILP with an evolutionary scatter search approach that extends the method to large problems with over 9000 entries. Finally, we introduce a new metaheuristic learning method that significantly improves the quality of solutions obtained.

Keywords. statistical disclosure limitation, mixed integer linear program, scatter search, adaptive learning

1 Introduction

The need to safeguard the confidentiality of survey data presents a monumental task, and government agencies must wrestle with this problem on a continuing basis. The continuing challenge is to maximize data quality and usability while preserving confidentiality. The focus of this paper is on *controlled tabular adjustment*, a method for confidentiality protection for tabular data, which recently has been extended to preserve data quality as well (Cox et al. 2004).

The importance of the confidentiality protection problem is confounded by its computational complexity. The primary mechanisms, cell suppression, controlled data rounding and perturbation, and controlled tabular adjustment, are expressed as decision problems subject to linear constraints involving potentially many binary variables. Moreover, NSOs must solve such problems on an ongoing basis and in many (survey) settings. Thus, the confidentiality problem for tabular data in most cases is not solvable optimally or even feasibly by standard algorithmic approaches.

In this paper, we analyze and augment the mixed integer linear programming formulation for controlled tabular adjustment. We conduct an empirical investigation of alternative methods for handling the underlying mixed integer/continuous optimization formulation that derives from this model. Our study creates new methods: a hybrid approach, a combined hybrid scatter search approach, and a metaheuristic learning approach. Our computational investigations disclose the difficulty of solving the problem due to the inherent combinatorial complexity of effective confidentiality protection, and illustrate how the new procedures can provide advances. Most significantly, we show that metaheuristic learning succeeds in improving the solutions to a degree that establishes these models as both a theoretical contribution and a truly practical advance in safeguarding sensitive data.

Controlled tabular adjustment (CTA) affords an opportunity to overcome many of the problems associated with traditional cell suppression and perturbation methods. CTA introduces controlled perturbations (*adjustments*) into tabular data that satisfy the protection ranges and tabular constraints (*additivity*) while minimizing data loss as measured by one of several linear measures of overall data distortion, such as the sum of the absolute values of the individual cell value adjustments. CTA typically replaces each sensitive cell by either of the two endpoints of its protection range, referred to as the *minimally safe values*. Selected nonsensitive cell values are then adjusted from their true values to restore additivity. Subject to assuring feasibility, nonsensitive cell perturbations are constrained to be small, such as within sampling variability, and cell values deemed undesirable for adjustment can be held fixed. Cox (2000) provides an early MILP formulation for CTA. The end result of CTA is a tabular system without suppressions meeting the disclosure rule, and close to original data with respect to a distortion measure.

A more extensive discussion of results reported here, and additionally comparison of simple heuristic procedures to exact solutions computing using the *ILOG CPLEX* ™ solver and limitations on computing exact solutions, are reported in Cox et al. (2006). Here we summarize development and analysis of new methods based on strategies of grouping and evolutionary scatter search. Scatter search offers particular advantages by running far more efficiently than CPLEX, and significantly extending the size of problems that can be addressed, yet still encounters limitations shared with its predecessors in generating solutions of high quality. For that reason, we develop a new metaheuristic learning algorithm that performs far more effectively than all other methods and provides a reliable and efficient approach for producing high quality solutions for problems of practical size.

2 Mixed Integer Linear Programming Model for CTA

The underlying concept of CTA is simple: the value of each sensitive cell is replaced by an *adjusted value* selected to be at a safe distance from the original value. Often, adjustment is to either of the sensitive cell's minimal safe values. Some or all nonsensitive cell values are then adjusted from their true values by small amounts to restore additivity to totals within the tabular system. Tabular data systems with marginal entries can be represented by their system of linear equations in matrix form: **MX = 0**. Column vector **X** represents the tabulation cells of the system; **x*** represents the original data. Matrix **M** is the *aggregation matrix* representing the tabular structure among the cells. The entries of **M** are –1, 0 or +1; each row of the **M** corresponds to one *aggregation* (tabular equation) in which “+1” denotes a contributing internal cell and “–1” a total (*marginal*) cell. With this notation, the mathematical structure of optimal synthetic tabular data is specified below by a mixed integer linear programming (MILP) formulation, containing binary and continuous variables, analogous to that introduced in Cox (2000). Our notation:

$i = 1,\ldots, p$: denotes the p sensitive cells

$i = p+1,\ldots, n$: denotes the (n-p) nonsensitive cells

B_i = binary (zero/one) variable denoting selection of the lower/upper limit for sensitive cell $i = 1,\ldots,p$

L_i = lower adjustment required to protect sensitive cell $i = 1,\ldots,p$

U_i = upper adjustment required to protect sensitive cell $i = 1,\ldots,p$

y_i^+ = nonnegative continuous variable identifying a positive adjustment to cell value i

y_i^- = nonnegative continuous variable identifying a negative adjustment to cell value i

UB_i, LB_i = upper/lower cell capacities on change to cell i

c_i = cost per unit change in cell i

MILP for Optimal Controlled Tabular Adjustment (to Minimal Safe Values)

$$\text{Min } \overset{n}{\underset{i=1}{?}}\ c_i\,(\,y_i^+ + y_i^-\,) \qquad (1)$$

Subject to:

For $i = 1,\ldots, n$:

$$\mathbf{M\,(\,y^+ - \,y^-\,) = 0} \qquad (2)$$

$$0 \;=\; y_i^+ \;=\; UB_i \qquad (3)$$

$$0 \;=\; y_i^- \;=\; LB_i \qquad (4)$$

For $i = 1,\ldots, p$:

$$y_i^+ = U_i\,B_i \qquad (5)$$

$$y_i^- = \;L_i\,(\,1 - B_i\,) \qquad (6)$$

After solving the MILP, the adjusted tabular data **t** = (t_i) are: $t_i = x^*_i + y_i^+ - y_i^-$. The objective function (1) minimizes the cost due to cell deviations. Linear costs are typically defined over the net adjustment $y_i^+ + y_i^-$. Two cost functions commonly used are: all $\boldsymbol{C_i}$ = 1, to minimize total absolute adjustment, and $\boldsymbol{C_i} = 1/\,x^*_i$ for nonzero cells, to minimize total percent absolute adjustment.

It is possible that (2) – (6) gives rise to an infeasible problem. Relaxing the sensitive cell constraints eliminates a large number of these types of problems:

$$y_i^+ \geq U_i B_i \tag{7}$$

$$y_i^- \geq L_i (1 - B_i) \tag{8}$$

3 Hybrid Heuristic

Because computation for the MILP roughly doubles with the addition of each binary variable, a sensible approach towards a computationally efficient, near-optimal algorithm is to group the sensitive cells, assign a unique binary variable to the group, and adjust all cells in a group in the same direction. We first tried random grouping, which performed poorly. We suggest ordering sensitive cells from largest to smallest, and assigning variables to different groups successively. This encourages between-group homogeneity, so large cells are less likely to be adjusted predominantly in one direction, expected to improve the solution. An exception: if a sensitive cell value equals one of its totals, both are assigned to the same group.

Let M = 2 be the number of groups. Add these constraints to the mathematical program: For i=1 to M, $B_i = B_{i+M} = B_{i+2M} = \ldots B_{i+kM}$, for (i+kM) = p. This reduces the number of binary variable to M. If M = p then the solution is optimal and if M < p then the solution may or may not be optimal. The mathematical program can be enhanced with additional constraints to improve the statistical characteristics of the solution (Cox et al. 2004). The Hybrid may be run multiple times and the best solution selected: we used groups of size = M, M-1, M-2, …., to produce a range of results and chose a superior solution. The Hybrid is more sophisticated than simply ordering cells by size and assigning directions alternately, as it does not predefine directions and evaluates M^2, not just one, assignments.

To evaluate the effectiveness of the Hybrid, sets of 2- and 3-dimensional test tables were randomly generated using the following specifications:

- 2-dimensional tables ranging in size from 4x4 to 25x25.
- 3-dimensional tables having sizes: nxnxn for n = 5,6,…11,12…20
- 3-dimensional tables having sizes: 10x10xn for n= 3,4,…,19,20
- Data values for internal tabular entries range from 0 to 1000 and are selected from a uniform distribution.
- 10% of the internal entries are selected randomly (uniformly distributed) and are assigned a value of 0.
- For all tables, 30% of the internal entries are defined as sensitive. The sensitive cells are distributed randomly (uniform) throughout the table. Marginal cells are not defined as sensitive.
- Sensitive entries must be assigned a value 20% greater than the original value or 20% smaller than the original value. All nonsensitive cells can be modified to values within 20% of their original values.
- In all tables, the sum of absolute changes is minimized.

Figure 1 shows the performance of heuristics compared to the optimal solution for moderately sized 2-dimensional tables. The heuristics are: Hybrid with M = 16, Ordering-With-Alternate-Assignment, and Best-Among-Random-Assignment over 100 and 1000 repetitions. The optimal solution curve is not displayed because its information is embodied in the report of the percent error of heuristic solutions with respect to optimal. M=16 was chosen to provide solutions in approximately the same time as required by Random-1000. The results indicate that the Hybrid is superior.

Figure 2 shows results for 3-dimensional tables. Optimal solutions could not be obtained for the larger tables, so Percentages are those relative to the Best-Heuristic solution, which, in almost every case, is achieved by the Hybrid heuristic. These results indicate that creating groupings of sensitive cells can significantly extend the applicability of the integer-programming model.

Finally, we explore an advanced approach for building groups. The principle is to minimize the number of potential within-group conflicts so that assignments do not produce large perturbations to totals. First, M groups are formed using the previous approach. For each group, we calculate the number of totals that are in common with each pair of cells, called the *group score*. We then *swap* cells between groups to decrease the grand total of all group scores. Swaps are continued until no further score reduction is possible. The resulting groups are then used to populate the mixed integer program. This procedure is referred to as Hybrid-With-Swaps. This strategy improved solutions approximately 10% on average.

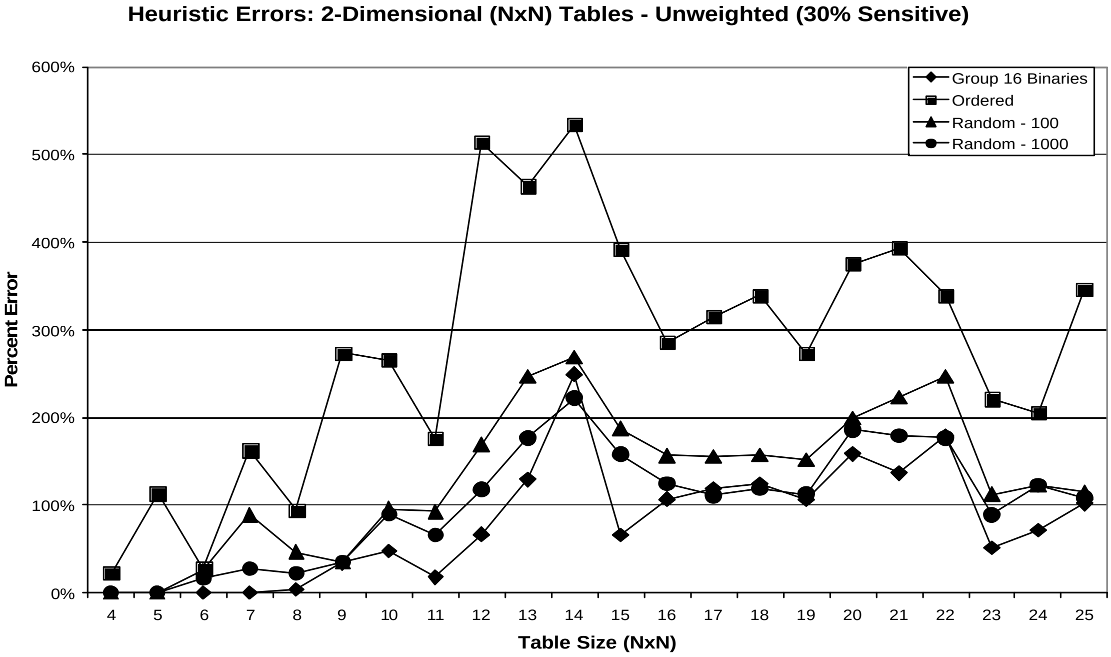

Figure 1. Performance of Hybrid on 2-dim tables based on percent error; 30% sensitive cells

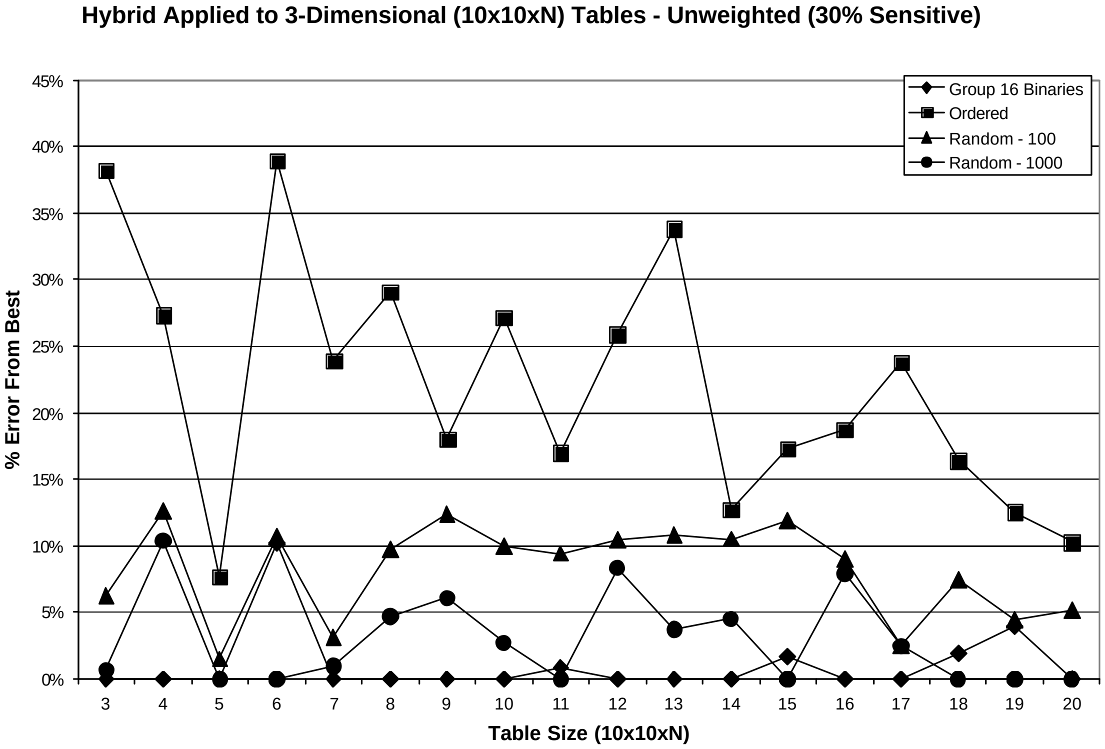

Figure 2. Performance of Hybrid on 3-dim tables based on percent error; 30% sensitive cells

4 Scatter Search to Enhance Hybrid Heuristic

Using the mixed integer programming based approach becomes impractical when the number of tabular entries exceeds 1000, e.g., the 10x10x20 table in Figure 2 required 76 minutes of computational time on 2.8GHz, Pentium 4 , 512 MB PC. To overcome this limitation, we used an evolutionary *scatter search* procedure (Laguna and Marti 2003). Scatter search is designed to operate on a set of points, called *reference points*, which constitute good solutions obtained from previous efforts. The basis for "good" includes criteria, e.g., diversity, that go beyond the objective function value. Scatter search then generates new points as combinations of the reference points. Combinations are generalized forms of linear combinations, accompanied by processes to adaptively enforce feasibility.

Points are considered *diverse* if their elements are "significantly" different from one another. The optimizer uses Euclidean distances to determine how close a potential new point is from those in the reference set, in order to decide whether the point is included or discarded. The number of solutions created depends on the quality of the solutions being combined, viz., combining the best two reference solutions generates up to five new solutions, while combining the worst two generates only one.

Combination may not generate solutions of enough quality to join the reference set, in which case a diversification step is triggered. The reference set is rebuilt to balance solution quality and diversity. Quality is preserved by seeding the reference set with a small subset of *elite solutions*; diversification is used to repopulate the reference set with solutions diverse relative to the elite set.

We used the *OptQuest*™ solver to implement the scatter search method for the CTA problem. Figure 3 shows the results of the scatter search method used in combination with hybrid and swap. Figure 3 provides results from taking the best solution obtained from using M= 9, 10, …, 16 (encompassing cases M = 1, …, 8). This experiment provided the best solutions in all cases and only doubled the computation time required for the M=16 run. It should also be noted that for all tables N= 10 the scatter search heuristic solutions were optimal. For larger tables, CPLEX was unable to run to optimality of the scatter solutions.

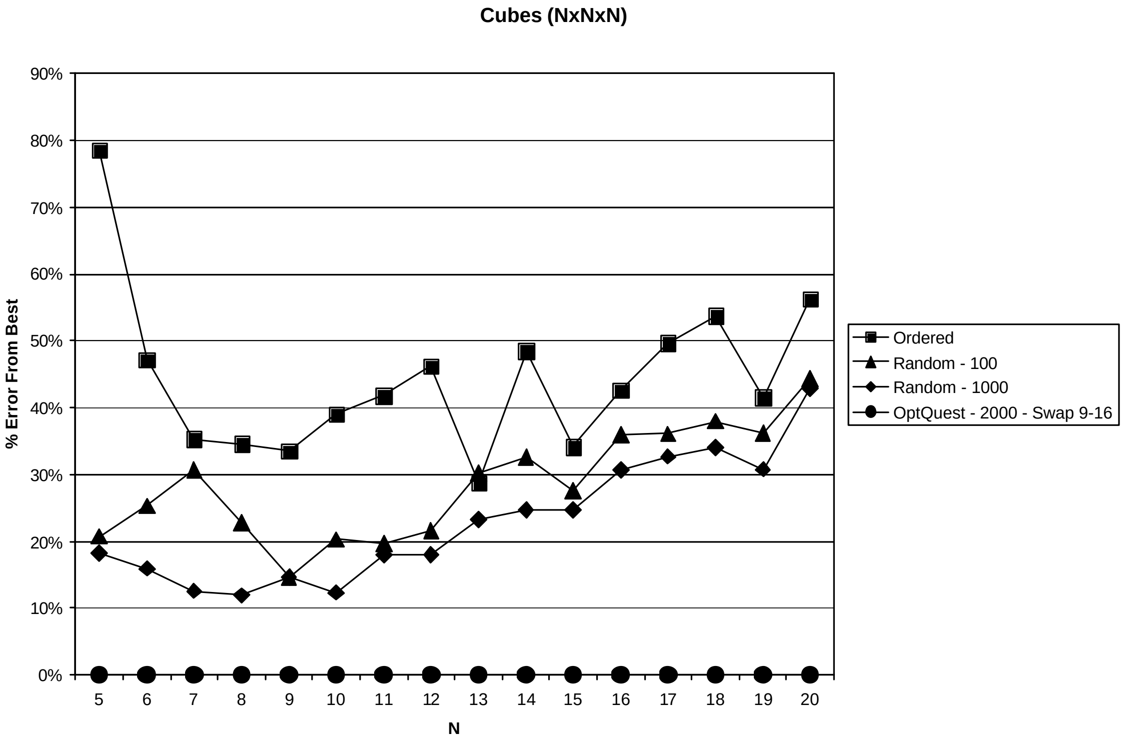

Figure 3. Performance of scatter search in combination with Hybrid-With-Swaps on cubic 3-dim tables based on percent error; 30% sensitive cells

5 Metaheuristic Learning Algorithm for Binary Variables

5.1 Learning algorithm

The grouping heuristics proposed in the previous section significantly reduced the problem size and thereby quickly solved the resulting integer program. However, these methods failed to produce satisfactory solutions for problems beyond a relatively limited size. The Best heuristic solution was at least 50% inferior to the optimal solution for all moderately large 2 dimensional tables. Moreover, heuristics exhibited considerable variation in the solution quality. These experiments demonstrate the importance of reducing the size of the integer programs for gaining computational efficiency. Inferior performance of these methods is attributed to their inability to predict and set appropriate values for a subset of variables. In this section we show that a metaheuristic learning strategy for fixing a subset of variables can be exceedingly useful for generating high quality solutions without consuming vast amounts of computer time to discover such solutions. This is based on the *proximate optimality principle,* which implies that a good solution at one level is likely to lead to good solutions at adjacent levels (Glover and Laguna 1997).

5.2 Parametric image

Our approach creates a strategic image of part of the problem to generate information about problem characteristics. Such processes have been used successfully in the fixed charge context (Glover et al. 2003), and are the basis for a class of metaheuristics procedures for mixed integer programming proposed in Glover (2003). Adapted to the present setting, the basic idea is to introduce parameters that penalize a variable's violation of integer feasibility, and to drive selected subsets of variables in preferred directions, viz., towards 0 or 1.

We are interested in identifying appropriate directions for selected subsets of binary variables, which are then tentatively fixed at their preferred values. The resulting reduced problem is then solved much more readily than the original problem providing an iterative process that results in high quality (optimal or near-optimal) solutions while expending only a small fraction of the computational effort required by a more traditional integer programming solution approach. We utilize this strategy to develop a parametric objective function approach to generate information on behavior of binary variables in the following manner.

We represent the objective in the compact form: minimize $x_o = cx$, where **x** is set of binary variables used to protect sensitive cells. We refer to "1" direction as (UP) and "0" direction as (DN) direction in our framework. These are called goal conditions (denoted as $x_j^{'}$) because we do not seek to enforce (UP) and (DN) directions by imposing them as constraints in the manner of customary branch and bound method but rather indirectly by incorporating them into the objective function of the linear programming relaxation. N^+ and N^- denote selected subsets of N containing UP and DN goal conditions; their union is $N^{'}$. $x^{'}$ denotes the associated goal imposed solution vector and M a very large positive number used as a penalty:

$$(LP^{'})\text{ Minimize } x_o^{'} = \sum_{j\in N^-}\left(c_j + M\right)x_j + \sum_{j\in N^+}\left(c_j - M\right)x_j + \sum_{j\in N/(N^+ + N^-)} c_j\, x_j \qquad (9)$$

$(LP^{'})$ targets goal conditions by incentive driven by penalty M. Binary variables of N^{-} are induced to go DN and those in N^{+} to go UP. Remaining variables are free to select direction. We are solving a linear program with penalty coefficients in the objective to gain insight about good values for binary variables.

5.3 Goal infeasibility and resistance

If a variable favors a particular direction, then it will achieve its targeted goal; otherwise, it will show some resistance to its imposed goal. We say that an optimal LP solution **x** = $x^{''}$ is *goal infeasible* if: for some $j \in N^{+}, x^{''}_{j} < x^{'}_{j}$ (**V-UP**), or, for some $j \in N^{-}, x^{''}_{j} > x^{'}_{j}$ (**V-DN**)

We call a variable x_j associated with violation (V-UP) or (V-DN) a *goal infeasible variable*. We create a measure of *overt resistance* (b UP, b DN), based on goal conditions, to learn about variable predilection for either direction:

For (V-UP), $bUP_j = x^{'}_{j} - x^{''}_{j}$ (10)

For (V-DN) $bDN_j = x^{''}_{j} - x^{'}_{j}$ (11)

No goal violation means zero overt resistance. If a variable does not violate its goal condition, it may *potentially resist* it: potential resistance = (dUP, dDN):

$$dUP_j = M + c_j + RC_j \tag{12}$$

$$dDN_j = -(-M + c_j + RC_j) \tag{13}$$

where RC_j is a suitably reduced cost for variable x_j .

The trial solution vector may contain variables without penalties. We use their solution values for the problem (LP) to create free resistances (aUP, aDN):

$$aUP_j = 1 - x_j \tag{14}$$

$$aDN_j = x_j \tag{15}$$

The parametric image of the objective is generated using a goal vector. A diversified sample of goal vectors is generated and resistance measures recorded to estimate directional effects. See Cox et al. (2006) for specification of the *parametric image learning algorithm*. The parametric image of the objective is:

$$x^{'}_{o} = \sum_{j \in N^{-}} \left(c_j + M\right) x_j + \sum_{j \in N^{+}} \left(c_j - M\right) x_j + \sum_{j \in N \big/ (N^{+} + N^{-})} c_j \, x_j \tag{16}$$

5.4 Performance of the learning algorithm for 2-dimensional tables

We implemented the learning algorithm using C++, ILOG- Concert Technology 1.2, and ILOG-CPLEX 8.1. Figure 4 shows the performance of our proposed learning method compared to other variable fixing heuristics.

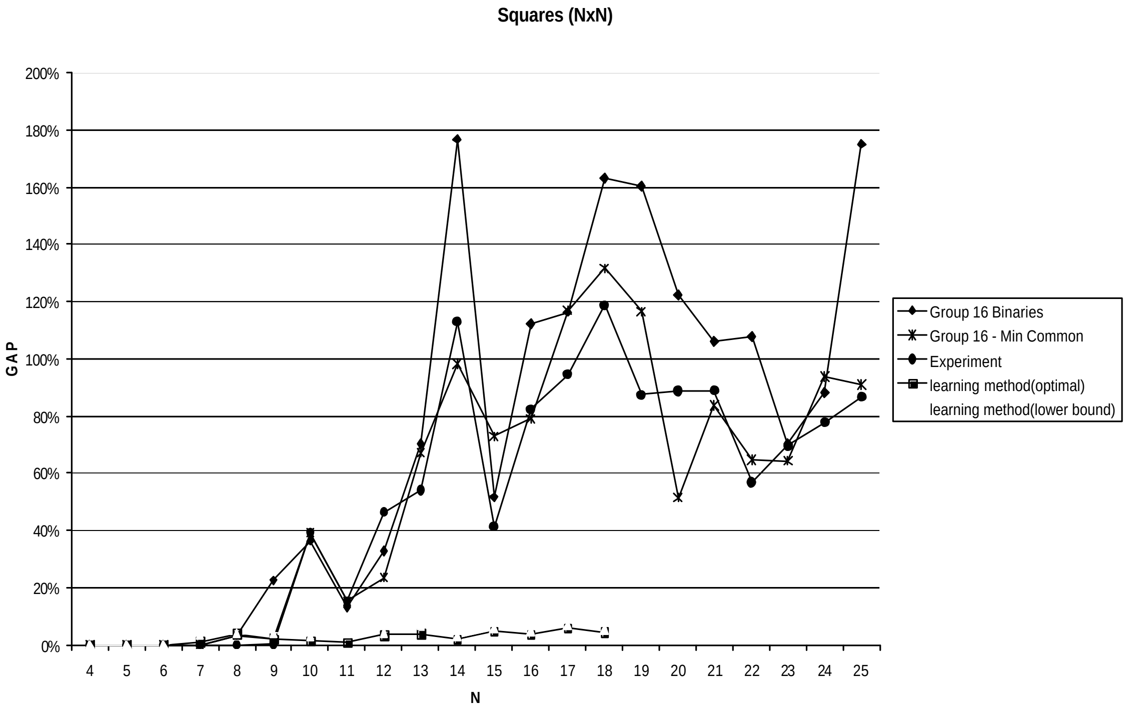

Figure 4: Performance of metaheuristic learning algorithm on optimality gap

The 25x25 problem exhibited an optimality gap of 9.6 %, but direct verification of the optimum was prohibitive. We needed a computationally efficient good lower bound on the optimum to measure the gap. (Cox et al. 2005) proposed a set partitioning based method, which we used as a proxy optimum for computing the gap in larger problems. These lower bounds were consistently very close to the optimum, e.g., for 2-dimensional tables restricted in size to no more than 18 rows and columns, the optimality gap was approximately 1%. In Figure 4, the “learning method (optimal)” curve identifies the optimality gap with respect to the known optimal value, and the “learning method (lower bound)” curve identifies the optimality gap with respect to the lower bound.

References

Cox, L.H. (2000), “Discussion (on Session 49: Statistical Disclosure Control for Establishment Data),” in: **ICES II: The Second International Conference on Establishment Surveys-Survey methods for businesses, farms and institutions**, Invited Papers, Alexandria, VA: American Statistical Association, 904-907.

Cox, L.H., Glover, F., Kelly, J.P. & Patil, R.J. (2006), “Confidentiality Protection By Controlled Tabular Adjustment: An Analytical and Empirical Investigation of Exact, Heuristic and Metaheuristic Methods,” *Decision Sciences Institute,* to appear.

Cox, L.H. & Kelly, J. P. (2004), "Balancing Data Quality and Confidentiality for Tabular Data," Proceedings of the UNECE/EUROSTAT Work Session on Statistical Data Confidentiality, Luxembourg, 7-9 April, 2003, **Monographs of Official Statistics**, Luxembourg: EUROSTAT., 2004, 11-23.

Cox, L.H., Kelly, J.P. & Patil, R.J. (2004), "Preserving Quality and Confidentiality for Multivariate Tabular Data," in: **Privacy in Statistical Databases 2004 (PSD 2004), Lecture Notes in Computer Science 3050** (J. Domingo-Ferrer and V. Torra, eds.) New York: Springer-Verlag, 87-98.

Cox, L.H., J.P. Kelly & Patil, R.J. (2005), "Computational Aspects of Controlled Tabular Adjustment: Algorithm and Analysis," in: **The Next Wave in Computer, Optimization and Decision Technologies** (B. Golden, S. Raghavan and E. Wasil, eds.), Boston: Kluwer, 45-59.

Glover, F. (1977), "Heuristics for Integer Programming Using Surrogate Constraints," *Decision Sciences* **8**, 156-166.

Glover, F. (2004), "Parametric Tabu Search Methods for Mixed Integer Programming," Leeds School of Business, University of Colorado, Boulder.

Glover F. & Laguna M.(1997), **Tabu Search**, Kluwer, Boston.

Karger, D.R. (1999), " Random Sampling in Cut, Flow, and Network Design Problems," *Mathematics of Operations Research* **24(2),** 383-413.

Laguna, M. & Marti, R. (2003), **Scatter Search: Methodology and Implementations in C**, Kluwer, Boston.

Lewis, M.W. (2004),"Solving Fixed Charge Multi-Commodity Network Design Problems using Guided Design Search," University of Mississippi, Hearin Center Technical Report , HCES-01-04.

Montgomery, D.C.(1984), **Design and Analysis of Experiment***s*, John Wiley and Sons, New York.