WP. 32 Confidentiality Protection by Controlled Tabular Adjustment Using Metaheuristic Methods

Lawrence H. Cox
U.S. National Center for Health Statistics

UNECE-Eurostat 2005 Work session on statistical data confidentiality
Geneva
9-11 November 2005

What Is CTA?

(Near) Actual Magnitude Table with Disclosures

167	317	1284	587	4490	3981	2442	1150	70(21)
57(1	1487	172	667	1006	327	1683	1138	46(7)
616	202	1899	1098	2172	3825	4372	300(40)	787
0	36(10)	0	16(4)	0	0	65	0	140(40)
840	2042	3355	2368	7668	8133	8562	2588	1043

Example 1: 4x9 Table of Magnitude Data & Protection Limits for the 7 Disclosure Cells (red)

D	317	1284	D	4490	3981	2442	1150	D
D	1487	172	667	1006	327	1679	D	D
616	D	1899	1098	2172	3825	4371	D	787
0	D	0	D	0	0	70	0	D
840	**2042**	**3355**	**2368**	**7668**	**8133**	**8562**	**2588**	**1043**

Example 1a: After Optimal Suppression: 11 Cells (*30%*) & 2759 Units (*7.5%*) Suppressed

167	317	1276	587	4490	3981	2442	1150	91
56	1487	172	667	1006	327	1683	1138	39
617	196	1899	1095	2172	3825	4372	260	797
0	26	0	12	0	0	65	0	180
840	**2026**	**3347**	**2361**	**7668**	**8133**	**8562**	**2548**	**1107**

Example 1b: After Controlled Tabular Adjustment

Collaborators

Quality-preserving CTA & CTA computation

James P. Kelly, OptTek Systems, Inc.
Rahul Patil, OptTek and University of Colorado
Fred Glover, OptTek and University of Colorado

Statistical Disclosure Limitation (SDL) for Tabular Data

Tabular data

* frequency (*count*) data organized in *contingency tables*
* *magnitude* data (income, sales, tonnage, # employees, ..) organized in sets of tables

Tables

* there can be *many*, many, many tables (national censuses)
* tables can be 1-, 2-, 3-, ...up to many *dimensions*
* tables can be *linked*
* table entries: *cells* (industry = retail shoe stores & location = Washington DC)
* data to be published: *cell values* (first quarter sales for shoe stores in Washington DC = $17M)

What is disclosure?

Count data: disclosure = small counts (1, 2, ...)
Magnitude data: disclosure = dominated cell value

Example: Shoe company # 1: $10M
Shoe company # 2: $ 6M
Other companies (total): $ 1M
Cell value: $17M

2 can subtract its contribution from cell value and infer contribution of #1 to within 10% of its true value = *DISCLOSURE*

Cells containing disclosure are called *sensitive cells*

How is disclosure in tabular data *limited* by statistical agencies?

* identify cell values representing disclosure
* determine *safe values* for these cells

Example: If estimation of any contribution to within 20% is deemed safe (policy decision), then a safe value is $18M viz., $18M - 6M = 12M $\geq$ (120%) $10M

Traditional method for SDL in magnitude data--*cell suppression*

* replace each disclosure-cell value by a symbol (*variable*)
* replace selected other cell values by a symbol (*variable*) to prevent narrow estimates of disclosure-cell values
* process is complete when resulting system of equations divulges no *unsafe estimates* of disclosure-cell values

Some properties of cell suppression:

* based on mathematical programming
* very complex theoretically, computationally, practically viz., NP-hard even for 1-dimensional tables
* destroys useful information
* thwarts many analyses; favors sophisticated users

How does cell suppression addresses *data quality?*

Cell suppression employs a linear objective function to control *oversuppression*
Namely, the mathematical program minimizes:

* total value suppressed
* total percent value suppressed
* number of cells suppressed
* logarithmic function related to cell values (*Berg entropy*)
* etc.

These are overall (*global*) measures of data distortion
Further, individual cell *costs* or *capacities* can be set to control individual (*local*) distortion

These are all sensible criteria and worth doing

However, they do not preserve statistical properties (*moments*)

Moreover, *suppression destroys data and thwarts analysis*

Controlled Tabular Adjustment (CTA)

* method for SDL in tabular data
* perturbative method–changes, does not eliminate, data
* alternative to complementary cell suppression
* attractive for *magnitude data* & applicable to count data

Original CTA Method

* identify sensitive tabulation cells
* replace each disclosure cell by a *safe value*–namely, move the cell value *down* or *up* until safety is reached
* use linear programming to adjust nonsensitive values in order to restore additivity (*rebalancing*)
* if second and third steps are performed simultaneously, a *mixed integer linear program* (MILP) results. MILP is extremely computationally demanding
* otherwise (most often), the down/up decision is made heuristically, followed by rebalancing via linear programming (LP) which computes efficiently even for large problems

(Near) Actual Magnitude Table with Disclosures

167	317	1284	587	4490	3981	2442	1150	70(21)
57(1	1487	172	667	1006	327	1683	1138	46(7)
616	202	1899	1098	2172	3825	4372	300(40)	787
0	36(10)	0	16(4)	0	0	65	0	140(40)
840	2042	3355	2368	7668	8133	8562	2588	1043

Example 1: 4x9 Table of Magnitude Data & Protection Limits for the 7 Disclosure Cells (red)

D	317	1284	D	4490	3981	2442	1150	D
D	1487	172	667	1006	327	1679	D	D
616	D	1899	1098	2172	3825	4371	D	787
0	D	0	D	0	0	70	0	D
840	**2042**	**3355**	**2368**	**7668**	**8133**	**8562**	**2588**	**1043**

Example 1a: After Optimal Suppression: 11 Cells (*30%*) & 2759 Units (*7.5%*) Suppressed

167	317	1276	587	4490	3981	2442	1150	91
56	1487	172	667	1006	327	1683	1138	39
617	196	1899	1095	2172	3825	4372	260	797
0	26	0	12	0	0	65	0	180
840	**2026**	**3347**	**2361**	**7668**	**8133**	**8562**	**2548**	**1107**

Example 1b: After Controlled Tabular Adjustment

MILP for Controlled Tabular Adjustment

Original data: nx1 vector **a**

Adjusted data: nx1 vector a + y^+ - y^-

T denotes the coefficient matrix for the tabulation equations

Denote $y = y^+ - y^-$

Cells i = 1, ..., s are the *sensitive cells*

Upper (lower) *protection* for sensitive cell i denoted p_i $(-p_i)$

<u>MILP for case of minimizing sum of absolute adjustments</u>

$$\min \sum_{i=1}^{n} (y_i^- + y_i^+)$$

Subject to: T (y) = 0

$$0 \le y_i^-, \; y_i^+ \le e_i \qquad i = 1, \ldots, n$$

$$e_i(1 - I_i) \ge y_i^- \ge p_i(1 - I_i)$$
$$e_i I_i \ge y_i^+ \ge p_i I_i \qquad i = 1, \ldots, s \text{ (sensitive cells)}$$

I_i binary

Capacities e_i on adjustments to nonsensitive cells are typically small, e.g., within measurement error

Quality-Preserving CTA

Based on mathematical programming, CTA can minimize:

* total (or max) of absolute values of adjustments
* total (or max) percent absolute adjustment
* number of cells changed
* logarithmic functions of absolute adjustments
* etc.

In addition, adjustments to nonsensitive cells can be restricted to lie within *measurement error*

Still, this may not ensure good statistical outcomes, namely,

Objective

analyses on original vs adjusted data yield comparable results

Towards Ensuring Comparable Statistical Analyses

Verification of “comparable results” is mostly empirical
Many, many analyses are possible: Which analysis to choose?

First, we focus on preserving key statistics and linear models

In the univariate case, we seek to preserve:

* mean values
* variance
* correlation
* regression slope

between original and adjusted data

Preserve means that adjusted data approximate reasonably well values for these quantities from original data

Can do this ***well in most cases using LP***

Preserving Univariate Statistics

Define: $L(\,y) = (1/(n\mathrm{Var}(a)))\sum_{i=1}^{n}(a_i - \bar{a})y_i = \mathrm{Cov}(a, y)/\mathrm{Var}(a)$

L(**y**) is a linear function of the adjustments **y**

Preserving means

The mean of any marginal total fixed by the LP is fixed
Any other mean can be fixed by fixing the corresponding total,
viz., the corresponding $\bar{y} = 0$

Preserving variances

Seek: $|Var(\,a + y) - \mathrm{Var}(a)|$ small, assuming $\bar{y} = 0$

$Var(\,a + y) = \mathrm{Var}(a) + 2\mathrm{Cov}(a, y) + \mathrm{Var}(y)$

$Var(\,a + y)/\mathrm{Var}(a) = 2L(y) + (1 + \mathrm{Var}(y)/\mathrm{Var}(a))$

$|Var(\,a + y)/\mathrm{Var}(a) - 1 | = | 2L(y) + (\mathrm{Var}(y)/\mathrm{Var}(a)) |$

Typically, $Var(\mathbf{y})/Var(\mathbf{a})$ *is small*
Thus, variance is approximately preserved by minimizing $|\,L(\,y)|$

* incorporate two new linear constraints in the system:

$w \geq L(\,y)$

$w \geq -L(y)$

* minimize *w*

Assuring high positive correlation

Seek: $Corr($ a, a + y) near 1

Corr (**a**, **a + y**) = $Cov($ a, a + y) $\div \sqrt{\text{Var(a) Var(a + y)}}$

$= (1 + L($ y)) $\div \sqrt{\text{Var(a + y) / Var(a)}}$

* denominator near one

* min $|L($ y)| drives numerator to one

Preserving regression coefficients

Seek: under ordinary least squares regression

$Y = \beta_1 X + \beta_0$

of adjusted data Y = **a + y** on original data X = **a**,

we want: β_1 near 1 and β_0 near 0

$\beta_1 = Cov($ a + y, a) / Var(a) = 1 + L(y),

$\beta_0 = (\bar{a} + \bar{y}) - \beta_1 \bar{a}$

As $\bar{y} = 0$, then β_0 near 0 if β_1 Near 1

This corresponds to $L($ y) near 0

Best result achieved for min $|L($ y)|

Comment: $L($ y) near 0 is motivated statistically because,

as solutions **y** and **-y** are equally good,

data **a** and adjustments **y** should be uncorrelated

Results for 4x9 table (7 sensitive cells)

Summary: 4x9 Table		**Linear**	**Programming**
Statistic	Corr.	Regress. Slope	New Var. / Original Var.
Value	**1.00**	**1.00**	**1.00**

Summary of Results of Numeric Simulations on 4x9 Table Using Linear Programming

Binary variables solved optimally

Results for 13x13x13 table (200 sensitive cells)

Summary: 13x13x13 Table		**Linear**	**Programming**
Statistic	Corr.	Regress. Slope	New Var. / Original Var.
Value	**1.00**	**1.00**	**1.00**

Summary of Results of Numeric Simulations on 13x13x13 Table Using Linear Programming

Binary variables solved by grouping heuristic (Hybrid)

Preserving Multivariate Statistics

Preserving the variance-covariance matrix

Data: **a**, **b**
Adjustments: **y**, **z**

Variances approximately preserved by preserving means and adjoining L (**y**) near 0 to CTA constraints: we call these the *univariate constraints*

$$\text{Cov}(\mathbf{a}+\mathbf{y}, \mathbf{b}+\mathbf{z}) = \text{Cov}(\mathbf{a}, \mathbf{b}) + \text{Cov}(\mathbf{y}, \mathbf{b}) + \text{Cov}(\mathbf{a}, \mathbf{z}) + \text{Cov}(\mathbf{y}, \mathbf{z})$$

Thus, $\text{Cov}(\mathbf{a}+\mathbf{y}, \mathbf{b}+\mathbf{z}) = \text{Cov}(\mathbf{a}, \mathbf{b})$ iff

$$\text{Cov}(\mathbf{a}, \mathbf{z}) + \text{Cov}(\mathbf{y}, \mathbf{b}) + \text{Cov}(\mathbf{y}, \mathbf{z}) = 0$$

Last term is nonlinear
Could use quadratic programming
We prefer to solve

$$\min |\text{Cov}(\mathbf{a}, \mathbf{z}) + \text{Cov}(\mathbf{y}, \mathbf{b}) + \text{Cov}(\mathbf{y}, \mathbf{z})|$$

subject to univariate constraints

as a sequence of LPs: for $\mathbf{y} = \mathbf{y_0}$ (constant), solve optimal $\mathbf{z} = \mathbf{z_0}$
fix $\mathbf{z} = \mathbf{z_0}$ (constant), solve optimal $\mathbf{y} = \mathbf{y_1}$
Continue
STOP when sufficiently close to 0

the *variance-covariance constraints*

Summary of Quality-Preserving CTA

Controlled tabular adjustment (**CTA**) can

- provide disclosure-protected tabular data
- preserve additive tabular structure
- be implemented using linear programming (**LP**)

Univariate case

CTA can be extended using LP to preserve

- means and variances
- correlation and regression between original and adjusted data

Multivariate case

Univariate CTA can be extended using LP to preserve

- multivariate variance-covariance matrix
- bivariate correlations
- bivariate simple linear regression coefficient

Metaheuristic Methods for Solving the CTA Binary Decision Problem

Quality-preserving CTA methods concentrate on the continuous portion of the MILP for CTA, viz., solve an LP

Arguably, with all quality constraints present, any solution for the integer variables that yields a feasible quality-preserving CTA is sufficient, viz., there is no need to solve the MILP optimally

Still, the binary and continuous portions are intertwined, and finding even a feasible quality-preserving for a problem with many binary variables can be challenging

So, the pursuit of heuristics for solving the binary CTA variables is well-motivated

Test Data

2-D and 3-D tables

Randomly generated internal entries

* 10%: randomly selected and set to 0
* 90%: uniform distribution 1 to 1000
* 30% randomly selected as sensitive
* sensitive cell protection limits +/- 20% cell value
* no sensitive marginals

Tables

* 2-D tables (NxN): N = 4, ..., 25
* 3-D tables (NxNxN): N = 5, ..., 20
* 3-D tables (10x10XN) N = 3, ..., 20

Objective function

* sum of absolute adjustments

We quickly found that simple heuristics, e.g.,

* alternate assignment of binaries based on size
* random assignment of binaries

performed poorly

Best-random (over 100 or 1000 tries) performed considerably better but we could not judge how well in absolute terms when the optimal solution was unknown

We investigated *metaheuristic algorithms* of 3 types

* grouping of binary variables (*Hybrid heuristic*) followed by incremental greedy improvement of groups (*Hybrid-with-swaps*)
* Hybrid-with-swaps followed by *scatter search*
* Tabu search *parametric learning algorithm*

Hybrid heuristic

Assume p binary variables
Choose M so that a binary problem with m variables can be solved optimally, e.g., M = 16
Order binary variables by size of cell value
Alternatively assign p binary variables to the M groups, except when a cell equals its marginal, in which case both go in the same group
Binary variables in each group are replaced by a single, common variable
Solve the resulting M-binary variable MILP optimally
This can be done for multiple M (e.g., M = 9, ..., 16)

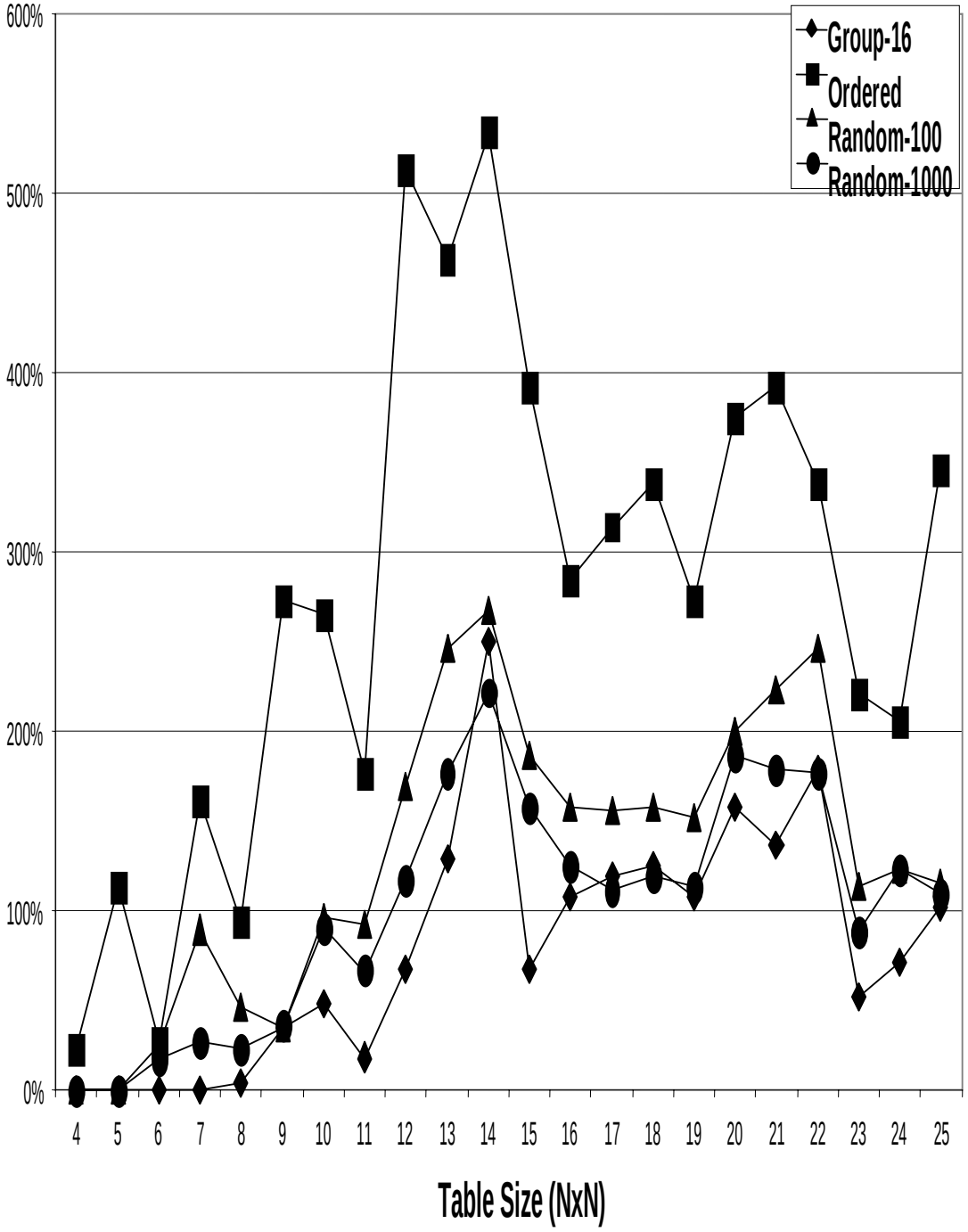
Hybrid-with-swaps

Swap a pair of variables between a pair of groups based on reducing the total of M individual *group scores* which count how many pairs of variables in the group share a common tabular equation
Repeat until no further reduction in total group score is possible
Solve the resulting M-binary variable MILP optimally

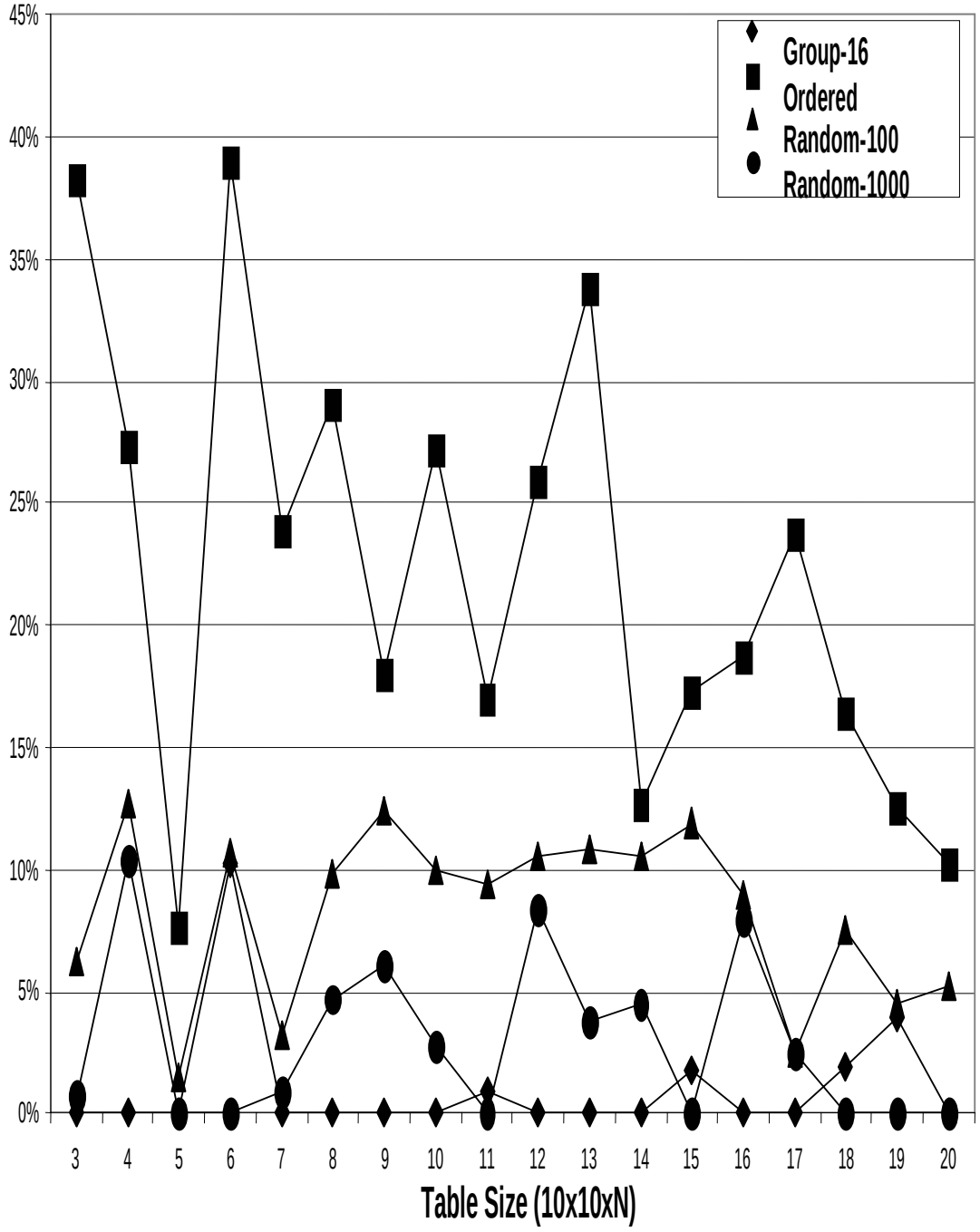
Hybrid-with-swaps improved the objective an average 10% over Hybrid

One could investigate assigning complementary values to selected pairs of binaries within a group

Performance of Hybrid on 2-D (NxN) Tables
Based on Percent Error (30% sensitive)



Performance of Hybrid on 3-D (10x10xN) Tables
Based on Percent Error From Best (30% sensitive)

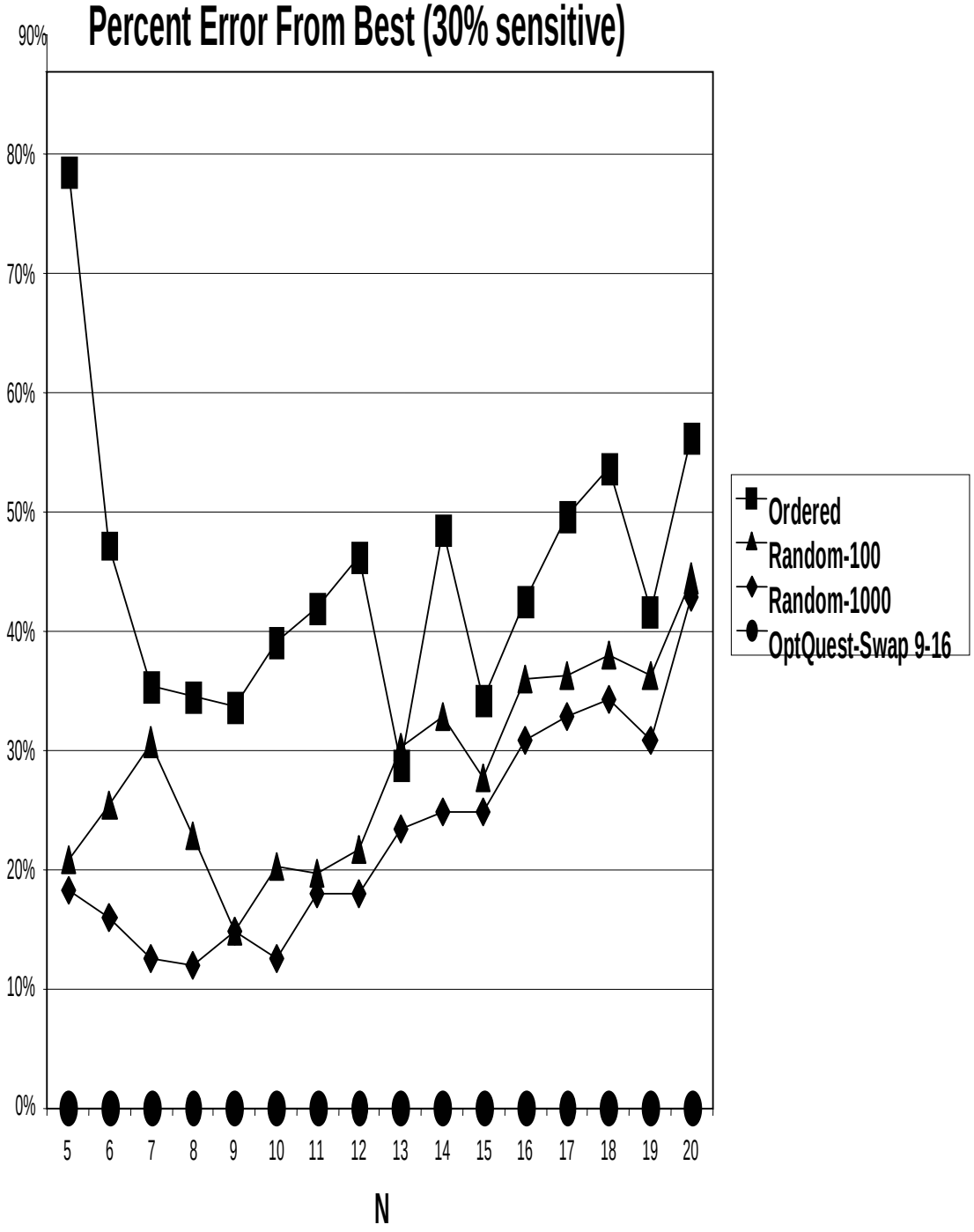


Scatter search (Laguna & Marti 2003)

A *reference set* of feasible solutions is formed
Membership combines
    * quality of objective value
    * diversity within the set (e.g., Euclidean distance)
Feasible non-convex combinations of reference solutions are formed (good solutions with good solutions, good solutions with bad solutions, ...)
Higher-quality combinations replace lower-quality solutions, subject to diversity

In CTA setting, reference set drawn from M = 9, ..., 16

In our experiments, for $N \leq 10$, scatter search achieved the optimal solution for all tables


Performance of Scatter Search Enhancing Hybrid-With-Swaps On 3-D (NxNxN) Tables Based on Percent Error From Best (30% sensitive)
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
N
Ordered
Random-100
Random-1000
OptQuest-Swap 9-16

Parametric learning algorithm (Glover 2004)

For large tables, grouping methods tended to
* exhibit considerable variability in solution quality
* be far from the optimum, viz., 50%

Given a selected subset S of binary variables, a more refined approach would introduce parameters that
* penalize a variable's resistance to achieve a binary value
* drive S-variables towards *preferred binary values*

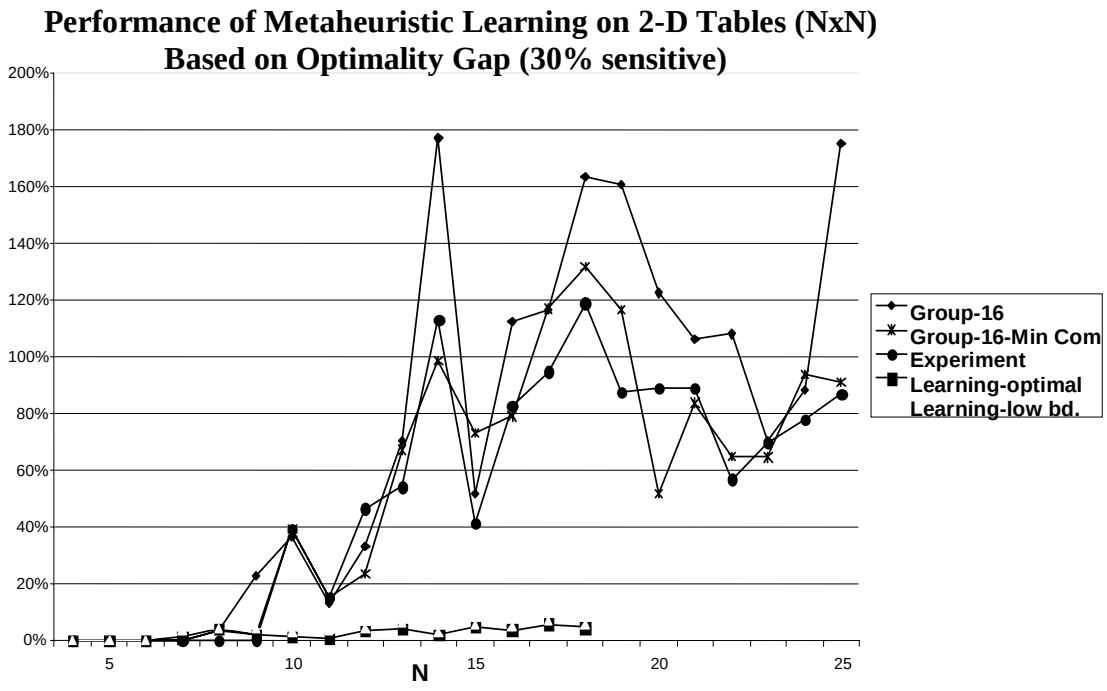
S-variables are tentatively fixed at their preferred binary values
Another subset is selected
Etc.

The key is a *parametric image of the objective function*

given Q >> 0, set S and exhaustive subsets S^0 and S^1

$$\min x = \sum_{j \in S^0} (c_j + Q) x_j + \sum_{j \in S^1} (c_j - Q) x_j + \sum_{other\ j} c_j x_j$$

* cost fluctuations RC_j are generated for the S-variables
* c_j is replaced by $(RC_j + c_j)$
* repeated runs over different parametric images enable measurement of resistances to binary values within S
* these are used to determine *preferred binary values*


Performance of Metaheuristic Learning on 2-D Tables (NxN)
Based on Optimality Gap (30% sensitive)
200%
180%
160%
140%
120%
100%
80%
60%
40%
20%
0%
5
10
15
20
25
N
Group-16
Group-16-Min Com
Experiment
Learning-optimal
Learning-low bd.

References

Cox, L.H., 1995, Network Models for Complementary Cell Suppression, *Journal of the American Statistical Association* **90**: 1453-1462.

Cox, L.H., 2000, Discussion (on Session 49: Statistical Disclosure Control for Establishment Data), **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., and Dandekar, R.A., 2004, A New Disclosure Limitation Method for Tabular Data That Preserves Accuracy and Ease of Use, **Proceedings of the 2002 FCSM Statistical Policy Seminar**, Washington, DC: U.S. Office of Management and Budget, 15-30.

Cox, L.H., Glover, F., Kelly, J.P. and Patil, R.J., 2006, Confidentiality Protection by Controlled Tabular Adjustment: An Analytical and Empirical Investigation of Exact, Heuristic and Metaheuristic Methods, *Decision Sciences Institute*, in press.

Cox, L.H., and Kelly, J.P., 2004, Balancing Quality and Confidentiality for Tabular Data, **Work Session on Data Confidentiality, 7-9 April 2003, Luxembourg, Monographs of Official Statistics**, Luxembourg: Eurostat, 11-23.

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

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

Glover, F. and Laguna, M., 1997, **Tabu Search**, Boston: Kluwer.

Laguna, M. and Marti, R., 2003, **Scatter Search: Methodology and Implementation in C**, Boston: Kluwer.