Distr.
GENERAL

ECE/CES/SEM.54/4
10 May 2006

ENGLISH ONLY

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

**EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE
EUROPEAN COMMUNITIES (EUROSTAT)**

**ORGANISATION FOR ECONOMIC COOPERATION
AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE**

Joint UNECE/Eurostat/OECD Seminar on the Management of Statistical Information Systems (MSIS)
Sofia, Bulgaria, 21-23 June 2006

Topic (i): Changes in statistical processes

APPLYING EVOLUTIONARY ALGORITHMS TO DISCLOSURE CONTROL PROBLEM

Invited Paper prepared by Andrea Toniolo Staggemeier and Vic Duoba,
Office for National Statistics, United Kingdom

I. Introduction

1. International and National Statistical Institutes face a difficult task in protecting their sources of information from disclosure of individuals whilst increasing the amount of publicly available and informative data. One of the widely used methodologies to address this problem is called Cell Suppression. This consists of trying to find an ideally optimal pattern of secondary suppressed cells that minimizes the total information loss subject to preserving the table additivity and the requirements for protection levels. Another methodology used in this area is called Controlled Rounding which rounds the nominal values of a table up or down, to their nearest integer of a base number, also preserving the protection level requirements for each cell and the additivity of the table. Both of these methods are present in the Tau-Argus software tool, which has been funded through the EU CASC project. Statistics Netherlands (Anco Hundepool), and University of La Laguna, Tenerife (Professor Salazar), have been major contributors to the dissemination of Mathematical Programming in the EU context, with additional software contributions from Germany.

2. Although the current implementations of these two available methodologies have their own merit, practical problems remain due to the large size of some tables that ONS and its external data providers need to deal with. The computational resource demands of mathematical programming algorithms that seek ideal solutions mean that often confidentialising of large tables either exceeds the maximum allocated time or exceeds resources in some other way. Hence there is a requirement to invest in research to try to overcome the numerical and computational difficulties experienced by the current available methods in Tau-Argus.

II. Meta-heuristics and Evolutionary Algorithms: A general Overview

3. Hybrid approaches, that use a less formal mathematical definition to a problem, are generally called heuristic methods. These methods have the greatest ability to generate fast near-optimal solutions to large problems, but on the other hand, they also may get trapped on local solution space and not be capable of improving the quality of their results through longer runtimes. This is a classical problem for most optimisation algorithms.

4. To meet the challenge of finding adequate solutions to large table confidentialising using heuristic methods, a "meta" phase is added to the heuristic process. This additional phase enables solutions of large problems within a reasonable limited amount of computational effort. Often this approach to solving complex systems is linked with mathematical programming, i.e. linear programming formulations, which undertake the evaluation process (feasibility check) of each solution.

5. Meta-heuristic approaches are well known in the Operational Research field and have been successfully adapted to a variety of optimization, and combinatorial problems. However, it is a relatively new approach to Statistical Disclosure Control (SDC). Cox et al., 2006, were pioneers in the introduction of the combined approach of exact methods, i.e. Linear Programming, and heuristic/meta-heuristic. For example, Taboo Search and Scatter Search procedures were applied to disclosure control, with a different methodology (to that in the Tau-Argus "Optimal" methods) called Controlled Tabular Adjustment (for the US Bureau of the Census).

6. This research has served to motivate the idea of using another metaphoric approach in ONS called the Evolutionary Algorithm (EA). The EA mimics the natural evolution process by representing a problem solution as genetic material, and through processes of selection, reproduction, and mutation evolves a population of solutions tending towards best results. The learning process is implemented by offspring inheriting good bits of each parent, and later a degree of *meme* can be added to each individual solution. In the context of disclosure control we are talking about evolving a population of suppressed patterns of a table such that at the end of the generating process we have selected the best individual solution to represent the algorithmic solution to the problem.

7. Other metaphoric approaches are also under investigation at ONS, such as the use of Ant Colony principles, originally introduced by [Dorigo 1996], and Greedy Randomised Adaptive Procedure, [Resende and Velarde 2003], to guide the search for the best suppression pattern in a table.

8. This paper stresses the importance of keeping a close link between ISIs, NSIs and universities so that creative thinking is applied to the challenges of large tables with multiple hierarchies and varying densities of zeros and sensitive cells. All this work is being developed in close partnership with two UK universities, namely the University of the West of England (UWE, Bristol-UK) and Cardiff University. Dr. Alistair R. Clark and Dr. James Smith (both from UWE-Bristol) lead the work on Evolutionary Algorithms [Clark and Smith 2006] and Dr. Jonathan Thompson (Cardiff) leads the work on Ant Colony Optimization and GRASP

algorithms [Thompson 2006]. For the purpose of this paper we will focus on only one meta-heuristic approach namely the Evolutionary Algorithms. For more details on Ant Colony Optimization or GRASP please do not hesitate to contact the authors of this document.

9. Before we go any further in describing the evolutionary algorithm developed by Dr Clark and Dr. Smith in collaboration with the authors of this paper we would like to introduce the readers to some of more general definitions of the meta-heuristic concerned.

10. Genetic Algorithms were first introduced by [Holland 1975] and named as Simple Genetic Algorithm. This method uses a string of binary values to represent the problem solutions and a generational evolutionary process is applied to evolve a population of individual solutions using reproduction and mutation operators as a way of intensifying and diversifying the search for better results. The following pseudo-code gives an idea of how a generic genetic algorithm works in practice.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Figure 1: pseudo-code for the Genetic Algorithm

11. The process from which a candidate solution is considered fit to the problem is guided by a fitness function, so when evaluating each candidate solution this will measure the optimization criteria to the problem. The evolution cycle happens in a loop of several consecutives operators. First a selection process has to be established in terms of what to do and how many solutions will figure as “parents”. Parent solutions are the ones responsible for creating new offspring. This process is called in evolutionary terms Recombination or Crossover operator. After a new population of offspring is created they will go through a mutation process which aims to alter the solution by randomly modifying parts of the encoded solution. This happens in a variety of different ways and we will be mentioning which one was adopted at our first EA implementation in the next section. A new round of evaluation of the offspring mutated is required so that according to the Replacement operator individuals are selected evolve from one generation to another.

12. In practice there are nowadays many varieties of genetic algorithms, and as a result of having been intensively studied in the last few decades, they have been applied to a variety of different problems, and the field has been renamed to Evolutionary Algorithms as it accommodates more and more metaphoric operators introduced so that the process of mimicking the natural evolution is extended to computer sciences widely. The alternative procedure adopted by ONS on this research project is described in the section below.

13. ONS is supporting and participating in pioneering work in the field of the application of Evolutionary Algorithms to the SDC, bringing the state-of-art research and algorithmic developments in this area closer to real world applications. For convenience, we will focus the description of our approach to the Cell Suppression Problem and a description on how the heuristic methodology can be applied. We also believe that after the initial phase of research has been completed it can be extended to the Controlled Rounding Problem.

III. Algorithm Contextual Description

14. A trade-off between optimal solutions obtained by the current optimal (i.e. optimal wrt a specific mathematical model) suppression methodologies in Tau-Argus (Fischetti and Salazar, 2001) and the solutions obtained by the new technology developed have recently been established that show that there is no detriment to safe suppressed patterns (as defined by the "optimal" method). A choice of a standard Incremental Attacker Model (Fischetti and Salazar, 2001), which verifies the solution for safety, is used to check solution feasibility and also serves as our evaluation criteria on how good/bad the solution generated is in relation to the total information loss. Initial results have shown that for small problems this computational effort is relatively insignificant and very effective to guide the search for optimal solutions, whilst for large tables this can take for example 30 seconds to find a solution for a table with 10,000 cells and 700 primary disclosive cells. Further testing on a wide range of data sources will be undertaken in order to verify the initial promising findings.

15. If further testing yields results compatible with the early results, ONS will have determined a more scalable approach to confidentialising (although we emphasize that we acknowledge the need for further confirmatory experimentation and stress-testing). The new research approach relaxes the requirement of full optimality in favour of achieving protection with a suppression pattern, which is likely to be close to optimal. This can be achieved using heuristic, and/or meta-heuristic methods. It is ONS's intention to make available any successful developments in the application of heuristics to protecting large tables through Tau-Argus. Improvements in the way we handled the required number of feasibility checks and the process of guiding the Evolutionary Algorithm, through reproduction and mutation operators, are discussed on the paper. Initial results and further directions to be taken in this research are also presented at the end of the document.

16. This paper highlights the importance of keeping Statistical Offices closely linked to universities and other NSIs in order to encourage creative and innovative thinking for the solution of difficult problems that can be addressed by advanced applied mathematics (OR) and high-powered computing platforms.

17. The Evolutionary Algorithm implemented at the first phase of this research project encoded each solution to the cell suppression problem as an ordered set of sensitive cells. By sensitive cell we understand the set of cells that fail the primary suppression rules in use[1]. The reasoning behind this decision was based on the evidence that the quality of the solution for the cell suppression problem depends on the sequence of which each sensitive cell is taken to assure the feasibility criteria is met. Using [Fischetti and Salazar 2001] heuristic procedure for the incremental attacker model as the way to measure the quality of the sequence evaluated and to ensure the protection levels on each cell this procedure builds up the secondary extra suppressed cells set following definitions of protection levels and cell weights from Tau-Argus. Another important factor that also influences the quality of the solution obtained when using this model is a cleaning-up process to remove redundantly suppressed cells. In fact, as this was not built as a constraint to the original incremental attacker model, so to reduce the complexity and allow quick run times, it is still required as a second step after the secondary suppressed set has been constructed.

18. To give the readers an example of how the representation is encoded to the Clark and Smith Evolutionary Algorithm here is one small and randomly generated frequency type table with 25 rows by 5 columns from which 10% of its values were considered sensitive cells and further 25% of its values were considered empty cells, is as follows:

2325	443	288	479	534	581
100	30	25	-	23	22
87	26	-	-	31	30
90	24	-	36	-	30
80	x	x	29	24	25
68	-	-	30	-	38
88	30	-	28	29	x
116	x	34	25	30	26
86	30	x	x	29	24
90	26	-	-	34	30
114	-	23	33	38	20
103	29	-	20	27	27
120	23	29	38	-	30
139	21	29	31	32	26
54	-	-	27	-	27
97	30	-	-	27	40
48	-	x	-	24	23
79	-	24	27	28	-
84	23	-	-	31	30
115	-	31	27	26	31
109	29	x	25	26	27
73	27	23	-	-	23
124	29	32	34	-	29
60	26	-	x	32	-
50	-	x	30	18	-
151	38	31	35	25	22

Table 1: Unprotected simulated frequency table[2]

[1] For more detail on different primary suppression rules please refer to [Willenborg and de Wall 2000]).
[2] Red crosses indicate sensitive cells after primary suppression rules were applied

19. From table 1, and using the linear representation of it as per the JJ-format file template, the natural order of the cell indexes set can be seen as {26, 27, 42, 44, 51, 52, 99, 129, 142, 147}. In terms of Evolutionary Algorithm this permits a permutation representation which is well studied in the field of optimization and combinatorial problems, such as the Travelling Salesman Problem (TSP). One could think to implement a canonical representation of the evolutionary algorithms by [Holland 1975] instead of the preferred choice of a permutation representation, however as this type of encoding would require an enormous lengthy binary solution string (i.e., 2^n, where n is the number of cells in a table) this has been rejected from the beginning of the project.

20. Before we go any further in defining other elements of the evolutionary algorithm adopted it is important to state that the objective of the cell suppression problem in minimising the total information loss is represented by the minimum sum of weights of those extra secondary cells required as in [Fischetti and Salazar 2001]. In other words, we were not actually interested in the nominal value of each cell but the relative importance the extra secondary suppressed cells have on the overall table so that the protection level requirement is achieved following Tau-Argus definition of weights. Perhaps it is also important to emphasis that this criterion does not guarantee that the total number of suppressed cells will be less than that found when using a different methodology from Tau-Argus. However, for methods that share the same objective function such comparison can be drawn provided the weights are calculated in the same way.

21. In terms of Evolutionary Algorithm design the information lost function is used as a way of measuring the fitness of each candidate solution, i.e. when combining the linear programming formulation for the incremental attacker model this also guarantees the solutions created are feasible so the EA task is to find the best sequence of sensitive cells that minimises the objective function and still satisfies the feasibility criteria.

22. The choice of evolution process implemented was very much nurtured by the fact that the problem had to maintain feasibility of candidate solutions whilst trying to improve the objective function. This meant that a generational evolutionary process, where all the new offspring population replaces the entire old population, was out of question. The preferred approach was a steady state model [Whitley 1989] where only few new, and good, offspring created would replace the worst individuals from the old population in the next generational cycle. This area is well studied in the Artificial Intelligence field and more studies can be found in [Rogers and Prugel-Bennett 1999]. It is important to mention that evolutionary algorithms use a large degree of randomisation. In order to guarantee a minimum variance, between runs on the same table, of the end result every time a user is performing the technique, thus the steady-state model played an important role.

23. The fundaments on how two “parent” solutions would recombine to form new individual solutions, i.e. crossover operator, were chosen based on the definition of how we represented the problem. An ordered base crossover mechanism [Michalewicz 1996] was selected to be the first one to be tested, particularly because this is a well-known procedure employed on other problems which also shared the same combinatorial aspects from the cell suppression problem.

24. Crossover mechanisms are used, most of the time, to intensify the search for good solutions as they work by trying to transfer the "good" features parent' solutions have to the new offspring created. Another mechanism used in the process of searching for better objective function values is called mutation operator. Mutation is an element of diversification and it works by randomly selecting a proportion of the offspring solution to change. The ones chosen to be the first tested were the insert and swap operators.

25. Next section will present the computational tests we performed up to the present moment, describing how instances to the problem were generated, specially the distinction between frequency and magnitude type of data, and some analysis of the results found.

IV. Computational Tests, Initial Results, Further Work

26. Implementation of the EA using C++ and the open source optimization library called COIN-OR, allowed ONS to retain solver independence feature when building the mathematical model. This means that ONS is not tied to any commercial optimization packages in order to perform the confidentialising routines. However, especially since open source code is under continuous development by a community of users that support the software, there is no guarantee that the code implemented is the most computationally efficient on its reusable parts. In fact, when using commercial available optimization solvers, such as Dash Optimization Xpress-MP, or Ilog Cplex, both outperformed in terms of run times the built-in CLP solver from COIN-OR package. The computational tests were run two machines one with 2GHz processor and running Windows XP with 2GB RAM and other a twin-core 64-bit Athlon processor using Linux.

27. Computational experiments were performed using frequency (80 different data files) and magnitude (80 different data files) type of data that were randomly generated using Poisson and Log Normal distribution functions, respectively. The choice of the distribution function came after extensive discussion between Information Management and SDC ONS research staff as it was desired that the simulated data would present key features found in real data. These data sets have a range of different sizes which can be describe as small, medium and large tables with very few considered very large. The size of the table is determined by the number of cells and the number of constraining equations. It is perhaps worth noting that hierarchical data have large degree of complexity (i.e. more equations are required to describe the table structure, for example to express sub-totals for sub-levels of the hierarchy) than none hierarchical one. All simulated data omitted any hierarchical explanatory variable however more tests will be performed for the hierarchical type of table structure. We have also considered other factors such as table density and the percentage of sensitive data desired. A choice of low sensitivity and low density and also of high sparsely and high sensitivity were considered. Table 2 presents a summary of the data type and features described related to the average computational time and the total weight of the secondary suppressed cells for the case of frequency data.

Computing Time

Average of time	Bvar				
Avar	5	10	20	50	Grand Total
25	9				9
50	26				26
100	83	301	1137	1804	825
200	232			1874	1053
Grand Total	90	301	1137	1839	695

Total Weight of Secondarily Supressed Cells

Average of mincost	Bvar				
Avar	5	10	20	50	Grand Total
25	1447				1447
50	1059				1059
100	972	806	539	236	640
200	957			184	571
Grand Total	1098	806	539	210	762

Table 2: Average computing times and solution quality when maximum run time is set to 1800 seconds for frequency type of table

28. The analysis of scalability of the approach can be described in terms of the time taken to solve the instances on different optimization solvers and the different types of instance data. These can be resumed as follows:

- There is no significant difference when compiling and running the code using Visual Studio .NET or other C compiler such as cygwin on Windows or gcc on Linux in terms of run times.

- As mentioned the choice of a commercially available solver package and an open source code results in speedups of a factor of 5. ONS has long adopted Dash Optimization Xpress-MP solver, which is also used by Tau-Argus optimization routines, and it was reassuring to have now the evidence on the efficiency of the commercial package. However, if time taken to find solution is not really an issue for some other NSI then CLP is an accurate and reliable solver in terms of the quality of the solution obtained. The other factor of consideration one could argue in favour of the open source code is the cost, free under gnu license agreement, whereas commercial optimization packages can cost in an excess of several thousands of sterling pounds.

- More on the process of determining the scalability of the EA was observed by Dr Clark and Dr. Smith that in fact the time taken to find the best solution increases as the square of the table size. Although this was not a complete surprise it is vital to define precisely which factors do play a significant role when trying to solve the cell suppression problem. Other factors were added to the statistical analyses and we uncovered that in fact the relation between the number of sensitive cells and the total number of cells in a table also plays a significant role. Due to the pressures to finish the current stage of research little can be said in terms of what is the exact link between these two factors (more likely to be the product between the two factors rather than the sum of them) and this will be followed up on the next stages of testing which need to be carried out by ONS.

29. The solution quality aspects of the EA were rather good for small and medium size tables and rather disappointing for large tables. However, on the very large data provided (over 40,000 cells with more than 3,000 sensitive cells a solution was obtained within the max time set). It is worth notice that on in this precise case full EA did not play any role on the solution built as no evolution process was performed (i.e. generation cycle). The solution obtained was however produced by one of the initialisation heuristics for the population, underlining the potentially important role that an initialization stage can play in time-limited problem-solving.

30. It is important to remind the readers that the definition of solution quality and feasibility is determined by solving a linear programming model for each sensitive cell. It was thought at the beginning of the project that this would be adequate, however for large and very large tables this is a time consuming process. An alternative method of defining the feasibility criteria has been proposed by Dr Thompson (from Cardiff University) and ONS will pursue this in the next stages of this research process.

31. Another very important factor that was identified by this initial research was the role played by the cleaning up process first mentioned by [Fischetti and Salazar 2001]. This procedure aims to remove redundantly suppressed cells added to the suppression pattern when using the incremental attacker model. This procedure is considered very time consuming but in some cases if not performed it can lead to unprotected redundantly suppressed cells.

32. Another possible way of improving the performance and quality of the solutions has been identified by Dr Clark and Dr Smith, which involves replacing consideration of each of the sensitive cells in a sequence individually by creating groups of these cells to protect jointly and simultaneously. This minimises the number of linear programs to be solved.

33. In an intuit of determining modern approaches to solve large tables other considerations ONS are examining are the use of parallel processing, such as grid computing, to speed up time-to-solution. This is only possible if we can determine a lighter evaluation function or it will be restricted by the number of solver licenses available for the case where commercial solver packages is required.

V. Conclusion

34. As an overall statement to the work presented on this paper one could say that the Evolutionary Algorithm implemented can perform as well as traditional techniques on small and medium data sets, and has the potential to manage large data sets (where traditional techniques fail), but more work remains to prove the quality for large data sets.

35. As we have described, the Evolutionary Algorithm presented is capable of solving large and very large tables, which it was not possible to confidentialise before. However the existing method used to evaluate the feasibility of solutions found is time consuming and requires further investigations.

36. Throughout these 4-6 months of very intensive research work in close collaboration with University of the West of England and University of Cardiff ONS has proven to be keen to explore the field of disclose control to a more modern and state-of-art use of the optimization techniques commonly used in Operational Research area of expertise.

37. This accomplishment of this project was only possible through the dedication of ONS staff involved and the external contractors. The authors would also like to thank the anonymous external referee of the research project as he provided very important points that need, and will be addressed on the next stage of research and also some helpful and interesting suggestions on further research in terms of improving the lower bound algorithm calculation proposed by Dr Thompson.

References

Clark A.R. and Smith J. (2006), “Improvements to Cell Suppression in Statistical Disclosure Control”, internal ONS report.

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, to appear.

Dorigo M., Maniezzo V. and Colorni A. (1996), “The ant system: Optimization by a colony of co-operating agents”, IEEE Transactions on Systems, Man and Cybernetics - Part B 26: 29-41.

Fischetti M. and Salazar J.J. (2001). "Solving the Cell Suppression Problem on Tabular Data with Linear Constraints", Management Science, vol. 47, no. 7, pp 1008-1026.

Holland J. H. (1992), Adaption in Natural and Artificial Systems. MIT Press, Cambridge, MA, 1992. 1st edition: 1975, The University of Michigan Press, Ann Arbor.

Michalewicz Z. (1996), *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, Heidelberg, New York, 3rd edn.

Resende M.G.C., Valarde J.L.G. (2003) “GRASP: Greedy Randomised Adaptive Search Procedures”, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.19 (2003), pp. 61-76.

Rogers A. and Prugel-Bennett A. (1999), “Modelling the dynamics of a steady-state genetic algorithm”. In: Banzhaf, Reeves (eds) *Foundations of Genetic Algorithms V*, Morgan Kauffman, San Francisco, CA, pp. 57-68.

Thompson J.M. (2006), “Heuristics and Metaheuristics for the Cell Suppression Problem”, internal ONS report.

Whitley L. D. and Kauth J. (1988), “Genitor: A different genetic algorithm”. In: *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, pp. 118-130.

Willenborg L., de Waal, T. (2000), “*Elements of Statistical Disclosure Control*”, Springer, Lecture Notes in Statistics.
