Constraint Databases and Geographic Information Systems

Peter Revesz

Department of Computer Science and Engineering
University of Nebraska-Lincoln,
Lincoln, NE 68588, USA
{revesz}@cse.unl.edu

Abstract. Constraint databases and geographic information systems share many applications. However, constraint databases can go beyond geographic information systems in efficient spatial and spatiotemporal data handling methods and in advanced applications. This survey mainly describes ways that constraint databases go beyond geographic information systems. However, the survey points out that in some areas constraint databases can learn also from geographic information systems.

1 Introduction

Relational databases [13] and geographic information systems [44, 46] developed separately. Relational databases used high-level, logic-based query languages like SQL and Datalog but were concerned only with table-oriented business data. Geographic information systems (GISs) used more operational query languages and dealt with geographic-oriented data.

Constraint databases [24, 35] were motivated in part to combine these two separate areas. In particular, the original constraint database paper [22, 21] insisted on using constraint query languages that are high-level, logic-based and have the following additional features:

1. The query evaluation must terminate and be computationally efficient.
2. The query evaluation needs to find all answers to a query.
3. The output of the query evaluation should be represented as another constraint relation.

These requirements are generalizations of the most important features of relational databases, which are just special cases of constraint databases. At the same time, constraint databases can also describe geographic data. Hence in theory constraint databases seem ideal to combine relational data and geographic data, which companies in practice still keep separately within a relational database and a geographic information system, causing various problems.

The insistence of meeting the above three requirements is novel with respect to constraint logic programming [19]. In fact, many constraint logic programmers expressed skepticism whether the requirements can be satisfied beyond trivial

Dagstuhl Seminar Proceedings 07212
Constraint Databases, Geometric Elimination and Geographic Information Systems
http://drops.dagstuhl.de/opus/volltexte/2007/1281

cases [20]. However, over the years many cases of constraint query languages were found. These cases can be grouped into the following three broad categories:

Queries that are restricted: Queries can be restricted to be nonrecursive, for example, relational calculus or nonrecursive SQL. Relational calculus queries can be evaluated with real polynomial constraints [22]. Nonrecursive SQL queries with rational linear constraints can be evaluated such that the maximum and minimum aggregate operators are generalized very naturally to a linear optimization problem. That has been implemented using the simplex method from linear programming in the MLPQ system [41] and the CCUBE system [1].

Queries with restricted types of constraints: Datalog queries with gap-order constraints [30, 29], half-addition constraints [33], positive linear constraints [33], and integer periodicity constraints [45].

Queries with restricted variable dependence: A pair of variables is *dependent* on each other if they occur together in the same constraint in either the query or the input database. Variables which are not dependent are called independent. Stratified Datalog queries with gap-order constraints and the restriction that in each negated relation all the variables are independent are also evaluable [34, 31]. Variable independence in the case of linear constraints is investigated by Chomicki et al. [7].

It is easy to check that a query satisfies one of the above three restrictions. In fact the checking can be automated, hence there is no burden on the users to prove query termination like in constraint logic programming.

Whereas most extensions of relational databases significantly complicate the SQL query language, constraint databases essentially keep the SQL query language unchanged. The greater expressive power of constraint databases is achieved by extending the database. As a result, writing constraint database queries can be done with the same ease as in relational databases.

For example, let us find the intersection of Cities(x,y) and Clouds(x,y). We write this in SQL as follows:

```
SELECT x, y
FROM Cities, Clouds
WHERE Cities.x = Clouds.x AND Cities.y = Clouds.y
```

The above query finds all (x,y) locations that belong to both the Cities and the Clouds relations. The same query applies to a relational database if the relations are represented as a finite number of grid elements or to a constraint database if the relations are represented using linear constraints or polynomial constraints.

Users who are familiar only with relational databases are sometimes afraid of using constraint databases because they mistakenly assume that they need to create the constraint databases. However, users need not create themselves the constraint databases. For example, users may simply download from websites some constraint databases that represent maps. Further, the users need not look at the constraints but only at the visualization of the map on the computer screen, just like in GISs.

There is a growing number of constraint database system prototypes. We developed a constraint database system called MLPQ (Management of Linear Programming Queries) [39, 23, 41], which runs both SQL and Datalog queries on linear constraint databases, and another constraint database system called DISCO (Datalog with Integer and Set Constraints) [2, 32], which allows set constraints on variables that range over subsets of the integers. Other research groups implemented several additional constraint database systems. For example, CCUBE [1] was implemented at George Mason University and DEDALE [16] at INRIA.

2 Geographic Data Handling in Constraint Databases

Constraint databases with rational linear inequality constraints can easily extend GISs. GISs usually deal with static two-dimensional objects, but constraint databases can easily add an extra space dimension or a time dimension and describe three dimensional objects or moving objects and other spatiotemporal phenomena. Below I review some research areas within constraint databases that are closely related to GISs.

2.1 Data models

New data models that are between linear constraint databases and the data models of geographic information systems are interesting to look at because the new data models can be more expressive that GISs but can be solved by efficient query evaluation methods.

Examples of new data models include the *parametric triangles* data model [8, 11, 10], the *parametric rectangles* data model [38, 3], and the *parametric geometric transformations* data model [12]. A common feature of these parametric data models is the introduction of a parameter t to represent time. Hence all of these data models are extending static spatial data models to *spatiotemporal* data models.

With each new data model, several important issues need to be investigated, for example the suitable query languages for the new data model and *interoperability*, that is, the relationship with and transformability into other data models.

An interesting new data model is GML, a spatial extension of XML. The interoperability of GML and constraint databases is explored in [5].

2.2 Indexing

Indexing is an organization of the data for efficient retrieval. In general the more complex the data are the harder it is to find an efficient indexing method for them. There is still no known efficient data structure to index general linear constraint databases. However, for the new data models discussed in Section 2.1, some efficient indexing methods can be developed.

The *PR-tree* indexing method is suitable for parametric rectangles [4]. A new problem on moving object databases is the *max-count problem*, which asks what is the maximum number of moving points in a moving rectangle window during a given period of time. It is important to answer max-count queries for example when scheduling airplane flights to limit the maximum congestion of airspace above an airport. Several new indexing methods that answer efficiently the max-count and related problems were developed recently [36, 6, 40].

Another geographic application is finding a shortest path in a network of roads. Geerts et al. [15] describe an indexing method that can efficiently answer shortest path queries in weighted graphs. When a landscape is seen from an airplane or a satellite from different angles, the different images generated are affine-invariant transformations of each other. Haesevoets et al. [18] describe an indexing method that can efficiently retrieve images that are similar to any affine transformation of a query image.

2.3 Interpolation of Spatial and Spatiotemporal Data

Spatial interpolation is the problem of estimating the unknown values of a spatially variable function based on known neighboring values. Spatial interpolation data can be represented in constraint databases [17].

While the interpolation of spatial data is a well-studied problem, the interpolation of spatiotemporal data is a more recent problem. Spatiotemporal interpolation arises in many applications. For example, the sale price of sold homes in a town is a spatiotemporal data. The interpolation in this case requires one to estimate the price of unsold homes at any given instance of time.

Most researchers considered spatiotemporal interpolation easily solvable by just using 3-dimensional Krieging or Inverse Distance Weighting (IDW) method from the spatial interpolation area. However, in fact a generalization of the shape functions method performs better than either Krieging or IDW [27, 26]. Since the shape functions method yields linear constraints, the interpolation data can be represented in a linear constraint database and queried in a constraint database system. Spatiotemporal interpolation algorithms can be fine-tuned depending on whether the spatiotemporal data is spatial-oriented or temporal-oriented [14].

2.4 Visualization

Visualization is important because users like to see pictures instead of the raw constraint data. Visualization methods include include value-by-area cartogram animation [28], animation of parametric triangles [9], and the visualization of a whole class of recursively defined spatio-temporal data [42, 43].

3 Geographic Applications using Constraint Databases

Section 2 gave a summary of research areas. This section reviews some geographic applications of constraint databases. Each of the applications would be difficult to solve in a traditional GIS system. These applications include the following:

Epidemics: A major problem is the efficient prediction of the spread of epidemics. A correct prediction can help the proper distribution of medical resources and timely warning of the endangered human populations. Revesz and Wu [43] presents a system to predict the spread of epidemics and reason about the consequences of the epidemics.

Voting: Another prediction method is developed by Gao and Revesz for the prediction of presidential voting outcomes. The voting results of the 2000 and 2004 presidential elections are examined and compared to what could have been predicted by the method for these elections based on data that were available before the elections.

House price estimation: The prices of houses were estimated with an average of less than 10% error using spatiotemporal interpolation, where the spatiotemporal data is stored in a constraint database [27].

Drought risk assessment: Many insurance companies provide farmers some insurance in case of crop failure due to drought or other conditions. Hence it is important for the insurance companies to properly assess the risk of drought in a given region at a given year considering also the type of crop planted and its water needs. A drought risk assessment system based on constraint databases was presented in [42].

Forest-fire management: The prediction of forest fires is also possible using parametric rectangles [3]. A method that watches the satellite image of the forest fire and predicts how its approximation as a set of parametric rectangles is likely to grow is described in the last chapter of [35].

Data mining in remote sensing: Li et al. [25] describe the application of object recognition in landscapes using satellite images. For example, a lake can be identified among a number of other lakes in the area based on its shape.

Habitat range of bird species: The habitat range of bird species is approximated using a set of parametric triangles in [9]. The approximation can be visualized and provides an extrapolation for future years.

The `TIME CLOSEST` problem: Given two moving points that move along two different lines with uniform speed, the `TIME CLOSEST` problem asks to find the time when they are closest to each other. Although this problem seems to require polynomial constraints at first sight, it can be expressed using linear constraint database queries [37].

4 Future Developments

Most of the applications in Section 3 were implemented on top of the MLPQ system. Geographic information system customers are looking for exactly those types of applications, but those are hard to implement in existing GIS systems. Hence GIS systems need to expand towards constraint databases, especially in the handling of spatiotemporal data.

In addition, there are an increasing number of companies that need to manage both relational databases and GIS. Relational databases and GISs were developed independently, and it is hard to manage simultaneously both types of data. Constraint databases are a generalization of both relational databases and GIS, hence it is imaginable that in the future some companies may like to combine their disparate databases into one common constraint database.

There are some areas where constraint databases can learn also from GISs. GISs can store not only the maps but some higher-level topological information as well. Constraint database research focused on the map representations. However, recent work in the constraint database area incorporates also special methods for handling topological information. In the future, some advanced and efficient GIS techniques for handling topological information could be adopted by constraint databases. There is a possibility for a fruitful interaction in the future between constraint database and GIS researchers and developers.

References

1. Brodsky, A., Segal, V., Chen, J., and Exarkhopoulo, P. The CCUBE constraint object-oriented database system. In *Proc. ACM SIGMOD International Conference on Management of Data* (1999), pp. 577–9.
2. Byon, J.-H., and Revesz, P. DISCO: A constraint database system with sets. In *Proc. Workshop on Constraint Databases and Applications* (1995), vol. 1034 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 68–83.
3. Cai, M., Keshwani, D., and Revesz, P. Parametric rectangles: A model for querying and animating spatiotemporal databases. In *Proc. 7th International Conference on Extending Database Technology* (2000), vol. 1777 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 430–44.
4. Cai, M., and Revesz, P. Parametric R-tree: An index structure for moving objects. In *Proc. 10th COMAD International Conference on Management of Data* (2000), McGraw-Hill, pp. 57–64.
5. Chen, Y., and Revesz, P. Querying spatiotemporal XML using DataFox. In *Proc. IEEE International Conference on Web Intelligence* (2003), IEEE Press, pp. 301–9.
6. Chen, Y., and Revesz, P. Max-count aggregation estimation for moving points. In *Proc. Eleventh International Symposium on Temporal Representation and Reasoning* (2004), IEEE Press, pp. 103–8.

7. Chomicki, J., Goldin, D. Q., Kuper, G. M., and Toman, D. Variable independence in constraint databases. *IEEE Transactions on Knowledge and Data Engineering 15*, 6 (2003), 1422–36.
8. Chomicki, J., Haesevoets, S., Kuijpers, B., and Revesz, P. Classes of spatiotemporal objects and their closure properties. *Annals of Mathematics and Artificial Intelligence 39*, 4 (2003), 431–461.
9. Chomicki, J., Liu, Y., and Revesz, P. Animating spatiotemporal constraint databases. In *Proc. Workshop on Spatio-Temporal Database Management* (1999), vol. 1678 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 224–41.
10. Chomicki, J., and Revesz, P. Constraint-based interoperability of spatiotemporal databases. In *Proc. 5th International Symposium on Spatial Databases* (1997), vol. 1262 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 142–61.
11. Chomicki, J., and Revesz, P. Constraint-based interoperability of spatiotemporal databases. *Geoinformatica 3*, 3 (1999), 211–43.
12. Chomicki, J., and Revesz, P. A geometric framework for specifying spatiotemporal objects. In *Proc. International Workshop on Time Representation and Reasoning* (1999), IEEE Press, pp. 41–6.
13. Codd, E. F. A relational model for large shared data banks. *Communications of the ACM 13*, 6 (1970), 377–87.
14. Gao, J., and Revesz, P. Visualization of temporal-oriented datasets. In *Proc. First International Conference on Geometric Modeling and Imaging* (2006), IEEE Press, pp. 57–62.
15. Geerts, F., Revesz, P., and Van den Bussche, J. On-line maintenance of simplified weighted graphs for efficient distance queries. In *Proc. 14th ACM International Symposium on Advances in Geographic Information Systems* (2006), IEEE Press, pp. 203–10.
16. Grumbach, S., Rigaux, P., and Segoufin, L. The DEDALE system for complex spatial queries. In *Proc. ACM SIGMOD International Conference on Management of Data* (1998), pp. 213–24.
17. Grumbach, S., Rigaux, P., and Segoufin, L. Spatio-temporal data handling with constraints. In *ACM Symposium on Geographic Information Systems* (1998), pp. 106–11.
18. Haesevoets, S., Kuijpers, B., and Revesz, P. Efficient affine-invariant similarity retrieval. In *Proc. Second International Conference on Geometric Modeling and Imaging* (2007), IEEE Press, pp. 99–108.
19. Jaffar, J., and Lassez, J. L. Constraint logic programming. In *Proc. 14th ACM Symposium on Principles of Programming Languages* (1987), ACM Press, pp. 111–9.
20. Jaffar, J., and Maher, M. Constraint logic programming: A survey. *J. Logic Programming 19/20* (1994), 503–581.
21. Kanellakis, P. C., Kuper, G. M., and Revesz, P. Constraint query languages. In *Proc. ACM Symposium on Principles of Database Systems* (1990), pp. 299–313.
22. Kanellakis, P. C., Kuper, G. M., and Revesz, P. Constraint query languages. *Journal of Computer and System Sciences 51*, 1 (1995), 26–52.
23. Kanjamala, P., Revesz, P., and Wang, Y. MLPQ/GIS: A GIS using linear constraint databases. In *Proc. 9th COMAD International Conference on Management of Data* (1998), C. S. R. Prabhu, Ed., McGraw Hill, pp. 389–93.
24. Kuper, G. M., Libkin, L., and Paredaens, J., Eds. *Constraint Databases.* Springer-Verlag, 2000.

25. Li, J., Narayanan, R., and Revesz, P. A shape-based approach to change detection. In *Frontiers in Remote Sensing Information Processing*, C. H. Chen, Ed. World Scientific Publishing, 2003, pp. 63–86.
26. Li, L., and Revesz, P. A comparison of spatio-temporal interpolation methods. In *Proc. Second International Conference on Geographic Information Science* (2002), vol. 2478 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 145–160.
27. Li, L., and Revesz, P. Interpolation methods for spatiotemporal geographic data. *Journal of Computers, Environment, and Urban Systems 28*, 3 (2004), 201–27.
28. Ouyang, M., and Revesz, P. Algorithms for cartogram animation. In *Proc. 4th International Database Engineering and Applications Symposium* (2000), IEEE Press, pp. 231–5.
29. Revesz, P. A closed form for Datalog queries with integer order. In *International Conference on Database Theory* (1990), vol. 470 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 187–201.
30. Revesz, P. A closed-form evaluation for Datalog queries with integer (gap)-order constraints. *Theoretical Computer Science 116*, 1 (1993), 117–49.
31. Revesz, P. Safe stratified Datalog with integer order programs. In *Proc. 1st International Conference on Principles and Practice of Constraint Programming* (1995), vol. 976 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 154–69.
32. Revesz, P. Problem solving in the DISCO constraint database system. In *Proc. Constraint Databases and Applications* (1997), vol. 1191 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 302–15.
33. Revesz, P. Safe Datalog queries with linear constraints. In *Proc. 4th International Conference on Principles and Practice of Constraint Programming* (1998), vol. 1520 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 355–69.
34. Revesz, P. Safe query languages for constraint databases. *ACM Transactions on Database Systems 23*, 1 (1998), 58–99.
35. Revesz, P. *Introduction to Constraint Databases*. Springer-Verlag, 2002.
36. Revesz, P. Efficient rectangle indexing algorithms based on point dominance. In *Proc. Twelfth International Symposium on Temporal Representation and Reasoning* (2005), IEEE Press, pp. 210–212.
37. Revesz, P. The expressivity of constraint query languages with Boolean algebra linear cardinality constraints. In *Proc. 9th Conference on Advances in Databases and Information Systems* (2005), vol. 3631 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 167–182.
38. Revesz, P., and Cai, M. Efficient querying of periodic spatio-temporal databases. *Annals of Mathematics and Artificial Intelligence 36*, 4 (2002), 437–457.
39. Revesz, P., Chen, R., Kanjamala, P., Li, Y., Liu, Y., and Wang, Y. The MLPQ/GIS constraint database system. In *Proc. ACM SIGMOD International Conference on Management of Data* (2000).
40. Revesz, P., and Chen, Y. Efficient aggregation over moving objects. In *Proc. Tenth International Symposium on Temporal Representation and Reasoning* (2003), IEEE Press, pp. 118–127.
41. Revesz, P., and Li, Y. MLPQ: A linear constraint database system with aggregate operators. In *Proc. 1st International Database Engineering and Applications Symposium* (1997), IEEE Press, pp. 132–7.
42. Revesz, P., and Wu, S. Visualization of recursively defined concepts. In *Proc. Eight International Conference on Information Visualization* (2004), IEEE Press, pp. 613–21.
43. Revesz, P., and Wu, S. Spatiotemporal reasoning about epidemiological data. *Artificial Intelligence in Medicine 38*, 2 (2006), 157–70.

44. Star, J., and Estes, J. *Geographic Information Systems: An Introduction*. Prentice-Hall, 1989.
45. Toman, D., and Chomicki, J. Datalog with integer periodicity constraints. *Journal of Logic Programming 35*, 3 (1998), 263–90.
46. Worboys, M. F. *GIS: A Computing Perspective*. Taylor & Francis, 1995.