

Using a Genetic Algorithm to Generate Alternatives for Multiobjective Corridor Location Problems

Xingdong Zhang and Marc P. Armstrong

Department of Geography, The University of Iowa,
316 Jessup Hall, Iowa City, IA 52242
Tel: +1 319-335-0150
Fax: +1 319-335-2725

Email: [xingdong-zhang | marc-armstrong] @uiowa.edu

Abstract

Corridor planning often falls into the domain of multiobjective evaluation. In this paper, we design and implement a multiobjective genetic algorithm for corridor location problems (MOGADOR). The new method produces a wide range of feasible corridor alternatives to help decision makers understand trade-offs among competing objectives. The results suggest that the MOGADOR approach will outperform traditional shortest-path algorithm methods.

1. Introduction

Specifying an optimal corridor that connects an origin and destination is analogous to identifying a shortest-path (SP) or least-cost-path through a varying space. Many existing corridor location methods in geographic information systems (GIS) and spatial decision support systems (SDSS) are derived from Dijkstra's shortest-path algorithm (1959), which has polynomial computational complexity and is intended to generate a single global optimal solution. Good corridor planning, however, often requires the participation of multiple stakeholders with different interests and emphases (e.g. engineering and environmental) and it is difficult to find a single solution that is best for all.

The purpose of this paper is to elucidate important characteristics of multiobjective corridor location problems and to present a multiobjective genetic algorithm (MOGA) approach to their solution on raster surfaces. A focus is placed on the geographical representation of corridor selection problems and the design of the crossover and mutation operators used by this type of genetic algorithm (GA).

The rest of the paper is organized as follows. In the next section, we review existing techniques for corridor alternative generation and principles of GAs. The design of the MOGADOR model, including the formulation of the problem, the development of a genetic representation and evolutionary operators, and the evaluation of solution fitness, is described in section 3. In section 4, we discuss the effectiveness and efficiency of MOGADOR based on the results of computational experiments. The paper concludes with some summary remarks and recommendations for further work on MOGA-based corridor location models.

2. Background

In this section, existing techniques for corridor alternative generation are described and compared. After an evaluation of those methods, we discuss some basic concepts and characteristics of MOGA, and argue that MOGA is appropriate to address multiobjective corridor location problems.

2.1. Multiobjective Corridor Location Problems

Multiobjective corridor location problems seek to find an optimal route, or a set of satisfactory routes, that link origins and destinations according to certain criteria. Such multiobjective decision-making problems have stimulated much research activity (Hopkins, 1984; Jankowski, 1995; Cheng and Li, 1997; Coello, 2000). Instead of looking for a global optimal solution, ill-structured corridor location problems often require the identification of reasonable alternatives for consideration in the decision making process, because of inaccuracies and errors in data collection and network representation, and subjectivity in land suitability score assignment. Therefore, it is important for corridor planners to identify a collection of possible alternatives and examine their suitability.

The two existing general approaches to alternative generation are based on classical SP algorithms. The first approach, a shortest path analysis method (SPAM), reruns a SP algorithm on a set of predefined weighted combinations of multiple objectives that reflect the relative importance of each different criterion. The second approach is to modify a SP algorithm and generate alternatives iteratively. Techniques using the latter approach include the k -shortest path algorithm (Yen, 1971; Shier, 1979), Iterative-Penalty Method (Turner, 1968), Difference Maximization (Huber, 1980), and Gateway Shortest Path (Lombard and Church, 1993; Scaparra and Church, 2004). The main idea of those techniques is to penalize (or exclude) those arcs that have been used in previously found solutions, or to restrict shortest paths to traverse certain gateway nodes. Such techniques aim to generate alternatives that are different in spatial arrangement.

Despite these efforts, five significant deficiencies exist in conventional SP-based corridor analysis and alternative generation methods:

- Most existing methods depend on a calibration and weighting mechanism. Though this approach may be effective in some circumstances and many techniques, such as the Delphi and Nominal Group methods (Hobbs and Voelker, 1978) are available, the process of assigning suitability values to each cell on a cost surface, and determining an appropriate weighting scheme, is subjective and controversial.
- For a complex geographical problem, the number of possible solutions is large and it is not easy for decision makers to discover compromises between conflicting objectives.
- While the SPAM approach is able to generate a multiobjective trade-off curve to identify alternatives, it does not widely explore the solution space because the search is restricted by the weighting scheme used.

- The computation time of SPAM depends not only on the number of different weighting schemes, but also on the number of objectives to be examined. Running a SP algorithm for each possible alternative is expensive and problems become intractable when datasets are large and many objectives need to be evaluated simultaneously.
- SPAM is not able to produce near-optimal solutions. The ill-structured nature of many spatial problems like corridor location, however, suggests the need for techniques that produce a wide range of potential (optimal and near optimal) solutions for good decision making (Brill *et al.*, 1990; Lai and Hopkins, 1989).

Multiobjective genetic algorithms can be used to overcome each of the deficiencies listed above.

2.2. Multiobjective Genetic Algorithms

Genetic algorithms are a particular type of evolutionary algorithm initially developed by Holland (1975) in the early 1970s. Theoretically, they are search algorithms that aim to maximize (or minimize) an objective function by emulating the mechanics of natural selection and genetics. GAs are computationally simple, yet robust and powerful, ways to search for potential solutions to optimization problems (Goldberg, 1989). They are able to produce optimal and near optimal solutions with a limited amount of computation and prior knowledge of problems. This characteristic makes them especially useful for optimization problems that have a large and complex solution space, in which conventional search methods normally lack either robustness or efficiency.

In GAs, each possible solution to a problem is encoded as an individual (chromosome) that when assembled collectively form a population. Each individual has a fitness value, which is usually derived from one or more objective functions and indicates how well the solution matches objectives. After creating an initial population, GAs run iteratively until a termination condition is satisfied. In each iteration (or generation), individuals with higher fitness values have a greater likelihood to be selected and reproduced in the population of the next iteration. There are three basic operators in a GA procedure: reproduction, crossover, and mutation. Reproduction is a process in which individuals are copied directly to the next generation according to their fitness values. Reproduction ensures that the genes of the fittest individuals are preserved across generations. Crossover is an exploitative method that exchanges partial genes from both parents and allows the combination of partial solutions to produce high-quality offspring. Mutation, on the other hand, is an explorative method that reintroduces genetic diversity and prevents premature convergence, thereby avoiding local sub-optimal traps in a solution space. A typical GA procedure can be described as follows:

```

/* GA Procedure( ) */
  i := 0
  initialize population  $p(i)$ 
  evaluate fitness of  $p(i)$ 
  while (not termination condition) do
     $i = i + 1$ ;
    generate  $p(i)$  from  $p(i-1)$ 
    evaluate fitness  $p(i)$ 
  od

```

The power of GAs for ill-structured problem solving can go beyond traditional single-objective optimization approaches because more than one objective can be incorporated into their fitness function. Since many real-world problems have several competing objectives, multiobjective genetic algorithms have become a popular research area and various MOGA methods have been developed for decision problems when multiple criteria are involved. A comprehensive technical summary of MOGA techniques and related criticism can be found in Coello's (2000) survey paper and Deb's (2001) book. More practical information about MOGA can be found in a number of independent implementations (e.g. Fonseca and Fleming, 1993; 1995; Srinivas and Deb, 1994; Horn, 1997).

Recent work in MOGA, motivated by Goldberg's (1989) suggestion of non-dominated solutions, has concentrated on the generation of a set of promising solutions that are Pareto-optimal (Pareto, 1971). A solution S is said to be Pareto optimal if no other solution dominates it. Deb (2001) showed a simple procedure to rank all solutions in a population by iteratively finding the non-dominated set of solutions after omitting any non-dominated solutions found in previous iterations. In doing so, a population of solutions can be classified into groups of different levels of non-domination (Goldberg, 1989; Deb, 1999, 2001); this procedure can replace the original objective function values and be applied in the fitness evaluation function of GAs. With the introduction of the Pareto-optimality concept and MOGA technology, decision-makers are now able to generate feasible alternatives that are on, or close to, the Pareto optimal front.

3. A MOGA approach to corridor selection problems

In this section, we present a MOGA approach to corridor alternative generation problems (MOGADOR) on raster surfaces. The discussion of the new approach covers the topics of land suitability data generation, the design of a network, the genetic representation of corridor selection problems, the development of genetic operators, and the evaluation of fitness functions. A MOGA-based corridor location model will typically include these functionalities.

3.1. Data Collection

A fundamental step in a corridor location procedure is to translate landscapes in a study region into a set of suitability maps; each represents a certain criterion that must be evaluated by decision makers. A suitability map is usually represented using grid cells. Each cell is assigned a numerical score to represent its suitability for inclusion in a

possible route based on a particular criterion. The suitability score is often calculated using a weighted combination of data from different categories, including elevation, slope, soil type, ownership, flood plains, and land cover type.

For simplicity, we rely on a set of simulated datasets to illustrate the functionality and validity of the MOGADOR model. Three suitability surfaces are used; each surface is a grid (50 x 50), representing the suitability of corridor alignment with respect to engineering cost (ENG), environmental impact (ENV), and economic and social benefit (SEO) (see Figure 1). Following previous work on corridor planning (Huber and Church, 1985; Lombard and Church, 1993), the suitability score assigned to each cell is scaled to a value between 0 and 10; low scores depict high suitability and high scores depict low suitability with respect to each criterion [value “0” indicates complete compatibility and “10” indicates the lowest compatibility]. Complete exclusion of a cell is possible by assigning it an arbitrary high value, say “9999”. In this way, the multiobjective corridor selection problem could be stated as the determination of a route that simultaneously optimizes all three objectives (described below) using a Pareto optimal solution concept.

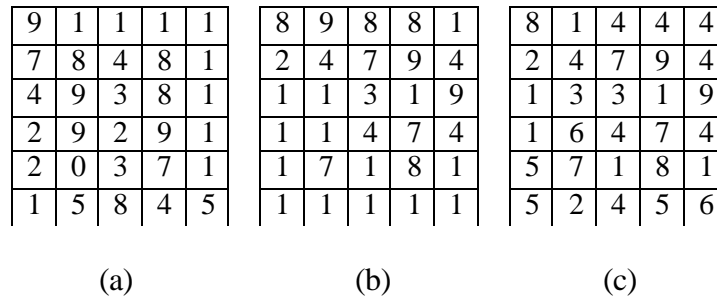


Figure 1. A 5 x 5 sample dataset; (a) ENG surface; (b) ENV surface; (c) SEO surface.

3.2. Design of networks

Unlike a classic routing problem, which is based on a predefined network and can be solved using a SP algorithm, there are an infinite number of possible routes between an origin and a destination on a continuous surface. Though grid representation of a continuous surface simplifies the problem, its use also introduces error. Because continuous data is made discrete and represented as a rectangular grid network, the effects of cell size, the number of allowed inter-cell links, and the method of assigning costs to links introduce inaccuracies. Such errors and inaccuracies have been formalized and analyzed by several researchers (Hopkins, 1973; Goodchild, 1977; Huber and Church, 1985; Xu and Lathrop, 1995). Huber and Church (1985) provide a thorough review of the geometric distortions that occur as a consequence of grid orientation, proximity, elongation and deviation.

A commonly used method to reduce errors between an ideal path across a continuous landscape and a best path on a restricted network is to increase cell connectivity at the expense of computational intensity. Compared with the Rook’s case (orthogonal) links alone, allowing Queen’s links (horizontal, vertical, and diagonal) will improve accuracy of distance measurement by 30% and allowing Knight’s moves (diagonal moves with a

1:2 or 2:1 ratio) will additionally improve accuracy by 7%. Huber and Church (1985) found that allowing the knight’s move reduces the representation errors to a reasonable level with an acceptable network size. In this study, all three types of movement (rook, queen, knight) are allowed. Each cell is allowed to link to its 16 nearest neighbors, which approximates a continuous space with an acceptable amount of computational effort.

3.3. Development of Genetic Algorithm in MOGADOR

Problem representation is a critical issue in GA design. Since it affects the performance of a GA significantly, designers must use a representational form that is suitable for a specific problem. In our method, a feasible corridor alignment, represented as a chromosome, is a continuous sequence of positive integers (genes); each integer represents the IDs of nodes through which the corridor passes. Each ID refers to the location of a corresponding cell on a raster surface. For example, in a 5 x 5 grid surface, node 17 refers to the cell located at the fourth row and third column using row ordering. Variable-length chromosomes and their genes are used for encoding the problem. Every valid chromosome starts with an origin cell, ends with a destination cell, and contains a set of connecting links that stretches from the origin to the destination (Figure 2). A valid chromosome must not have duplicated nodes in the sequence, as this would indicate a circular path.

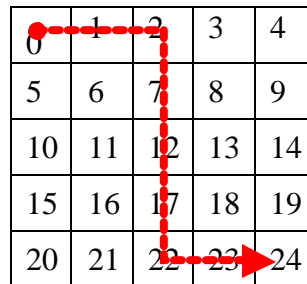


Figure 2. The node indexes in a 5 x 5 grid and a valid route: 0-1-2-7-12-17-22-23-24.

3.3.1. Population initialization

An initial population is generated at the beginning of the MOGADOR procedure. We experimented with two techniques to create a group of individuals in the population: random and heuristic. Traditionally, each individual in a GA is initialized randomly. In our case, the initiation procedure starts with an origin and randomly chooses a valid cell based on the topological (connectivity) information of the network. The encoding process keeps selecting a valid cell that can be connected to the last cell of the current route and has not been included in the route so far, until a destination is reached. A random walk alone often results in poor performance (e.g. extremely long length chromosomes and a large amount of computation time) when working on larger datasets. In response, a heuristic method, called “directionally biased walk”, has also been developed. Cells in a direction leading from the origin towards the destination have a higher probability to be selected as the next gene. The combination of random and biased walk approaches is able to produce a better initial population while maintaining its diversity. In our study, for example, 80% of the genes in each individual corridor are chosen heuristically and 20% are chosen randomly.

3.3.2. Selection

The selection operations in MOGADOR select chromosomes from the current population for reproduction, crossover, and mutation based on a probability that is proportional to the fitness of each individual. Two selection schemes are implemented in our model: roulette wheel and tournament (see Goldberg, 1989).

3.3.3. Crossover

The crossover operation in a conventional GA is based on the exchange of genes between two fixed length chromosomes when a bit string representation is used. In the MOGADOR model, variable length chromosomes are used and two crossover operators have been designed. The first crossover procedure, “shared node” crossover, chooses two parent chromosomes that may have different lengths but must have at least one gene in common (except for the origin and destination). The second crossover method, called “shared neighbor” crossover, is used when two parents have no shared nodes, but have at least a pair of genes (one from each parent) within a predefined proximity (e.g. 8 or 16 neighboring cells). After crossover, the partial-routes of two parents, separated by the pair of genes, are exchanged and the paired “shared neighbor” genes from two parents are connected together and included in both child chromosomes.

3.3.4. Mutation

Mutation introduces random changes to a chromosome and thus maintains population diversity. It also prevents the premature convergence of the population, a phenomenon of convergence to *non*-optimal solutions due to the lack of population diversity. Two mutation methods are designed in MOGADOR. In the first method, called “partial” mutation, a gene in a parent chromosome is randomly selected as a break point. The mutation process will generate (using a similar procedure as for initialization) a partial-route starting from the mutation break point to the destination. This is then combined with the partial-route from the origin to the break point in the parent’s chromosome to form a new child. A second mutation operator, called “add-drop” mutation, randomly selects one or more nodes (determined by parameters) from a chromosome. For each node, the procedure checks its connectivity and eliminates the node if a node before and after it is within the same neighborhood. In the same way, the “add-drop” method selects several nodes in a parent and inserts a valid new node before (or after) each of them.

3.3.5. Fitness evaluation

Different from single-objective GAs, in which fitness is often the objective function of the indicated optimization problem, in a multiobjective scenario, the fitness function of a GA should reflect the optimality of a solution in each of the objectives. Therefore, a Pareto optimum concept is adopted. The fitness value of a solution depends not upon the values from a single objective function, but upon its Pareto rank within the entire population. According to the three suitability surfaces described in the previous section, three objective functions are constructed; each calculates the accumulated suitability of a corridor alignment based on engineering, environmental, or social and economic criteria. In so doing, the multiobjective corridor problem can be stated as the determination of route alternatives that simultaneously optimize all three objectives using a Pareto non-

inferior solution concept. Here, $z(c_i)$, $z(e_i)$ and $z(s_i)$ are the suitability value of site i for engineering, environmental, and socio-economic criteria, respectively.

- minimize $Z(c) = \sum z(c_i)$, total ENG of a valid route.
- minimize $Z(e) = \sum z(e_i)$, total ENV of a valid route.
- maximize $Z(s) = \sum z(s_i)$, total SEO of a valid route.

The initial fitness of an individual is defined as an inverse function of its Pareto rank ($1/\text{rank}$), as suggested by Goldberg (1989). While this method effectively drives the solution process toward the front, it does not explicitly ensure a diverse solution set that expands across all portions of the trade-off curve. As a result, the GA will often converge to subregions along the Pareto front (Bennett *et al.*, 2004). Two techniques, “population gap” (De Jong, 1975) and “niche count” (Beasley *et al.*, 1993; Srinivas and Deb 1994), are used to address this problem. In population gap, a proportion of the old population is directly copied to the next generation without any modification. The niche count technique boosts the fitness of those solutions that lie sparsely along the Pareto front by adjusting their fitness according to the number of solutions within their neighborhood.

4. Experimental Results and Discussion

The MOGADOR model described above was coded in Microsoft Visual C++ 6.0 and tested on a Windows XP PC (Pentium 4/3.2-GHz processor with 1 GB memory). Three 50 x 50 grids are used to test the validity and performance of the model.

MOGADOR was run for 100 generations with 100 individuals. For each generation, all feasible solutions were reported and displayed. The results and associated parameters of the MOGADOR model on the sample dataset are showed in Figure 4. Black dots in the figure represent the solutions produced in the GA procedure. MOGADOR generated 10000 possible corridor alignments on the 50 x 50 grid dataset in 20 seconds.

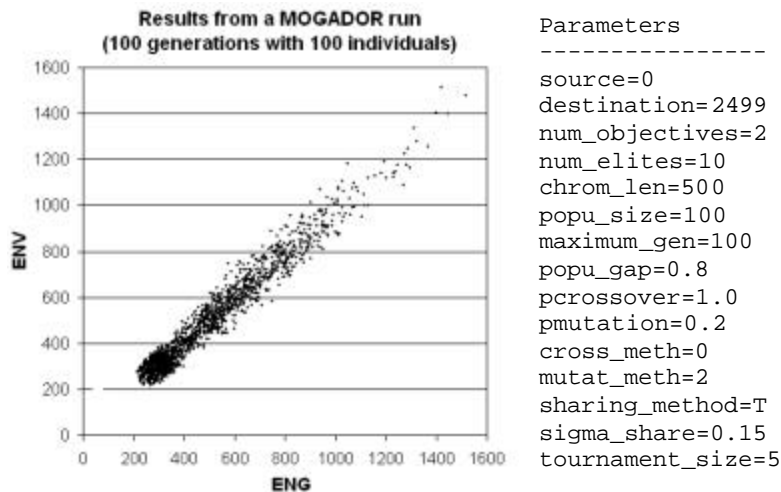


Figure 4. The results and parameters of MOGADOR on the 50 x 50 grid dataset.

Parameters and strategies used in the MOGADOR location model play a significant role with respect to its performance and the quality of the solutions obtained. Through a set of experiments on the sample dataset, their effects are noted as follows:

- The random initialization procedure is fast when the dataset is small. However, the computation time of the random initialization process increases dramatically as the data size grows. In our experiment, MOGADOR with a random initiation procedure took 340 seconds, compared with 20 seconds for biased walk, to generate an initial population for the 50 x 50 grid dataset.
- The niche count technique increases the diversity of the population. According to our experiment, its effect, however, is limited. The population will expand only to a limited portion of the Pareto front (Figure 5 (b)). Using niche count alone, a well formed Pareto front is difficult to achieve.
- Also from Figure 5, we found that the population generated by MOGADOR did not approach the grey circles (a sample of the Pareto optimal front created using a traditional SP algorithm) very well. We also tested the model with a large population size and a large number of generations (*popu_size* = 5000 and *maximum_gen* = 500) that requires a considerable amount of computation time. MOGADOR approaches but is not guaranteed to reach the Pareto optimal front. So additional knowledge is needed to guide the evolution of the population more efficiently.

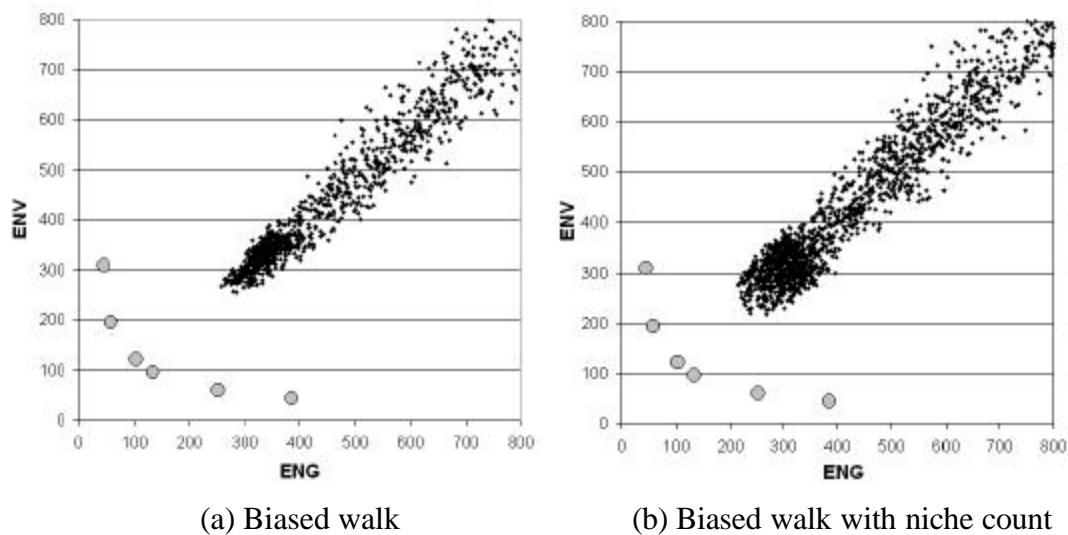


Figure 5. The results of MOGADOR on the 50 x 50 grid with and without “niche count”. Solutions with objective function values less than 800 are reported. Grey circles indicate the Pareto optimal front defined by a shortest path algorithm.

- The “partial” mutation drives a population progressively toward a trade-off curve on the objective space. The use of the “add-drop” mutation method, however, can speed up the procedure and find better solutions. Figure 6 shows the results of

MOGADOR with and without the “add-drop” mutation. The black line represents the evolution of minimum ENG values over generations when only the “partial” mutation is used. The grey line is the corresponding ENG values when both mutations are used. The latter, though, started with a relatively larger ENG value, decreased quickly and achieved a smaller value after the second generation.

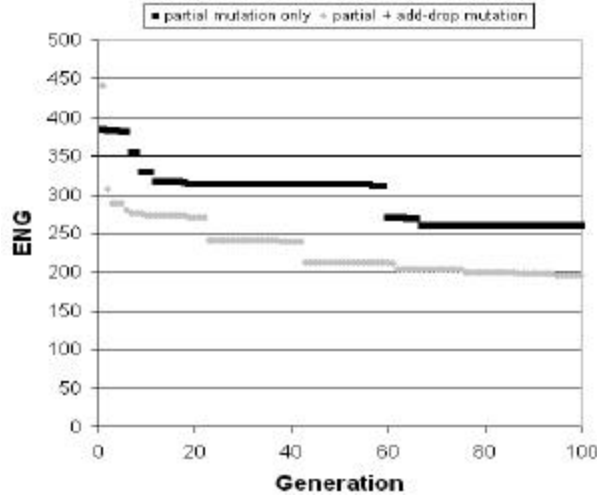


Figure 6. MOGADOR with and without the “add-drop” mutation method.

The traditional SPAM approach is also able to generate a multiobjective trade-off curve. SPAM, however, is a NP Hard problem solving method and requires a large amount of computation to produce alternative solutions. The computational complexity of SPAM is dependent not only on the number of objectives investigated but also on the resolution of the weighting scheme on multiple objectives. An n -objective corridor location problem with a 1% objective weighting interval has 100^{n-1} possible weighting schemes and must execute a SP algorithm 100^{n-1} times. Consequently, the computation time of SPAM increases exponentially and the corridor location problem becomes intractable when the number of objectives exceeds five or the grid size exceeds 250 x 250. In contrast, the computational complexity of the MOGADOR model is much less. The entire population is generated by the reproduction, crossover, and mutation operation from the MOGA procedure. The comparison of computation time for a two and three objective corridor alternative generation problem is shown in Table 1. The SPAM procedure is based on a 1% objective weighting interval.

	Two objectives		Three objectives	
	Time	# Solutions	Time	# Solutions
SPAM				
50 x 50	11	100	672	10000
MOGADOR				
50 x 50	20	10000	26	10000

Table 1. The computation time and solution of SPAM and MOGADOR. Computational time is measured in seconds.

5. Conclusion and Further Research

While corridor planning often falls into the domain of multiobjective optimization, existing corridor location models are generally based on shortest-path algorithms and often search for a single global optimal solution. Though different approaches have been developed to generate alternative routes for decision making, those techniques still depend on a weighting mechanism to collapse multiple objectives into a single objective function and, as a consequence, they are not suitable for ill-structured corridor planning problems, in which trade-offs among competing objectives need to be investigated. In this paper, we presented and implemented a novel MOGADOR algorithm to generate multi-objective alternatives for corridor location problems. Using a new set of genetic operators, the approach captured a range of optimal and near optimal corridor solution alternatives. The results from our experiments demonstrate that the MOGADOR model significantly outperforms conventional SP-based methods both in computational complexity and the number of generated alternatives.

The findings in this research make a contribution to GIScience-based spatial analysis in general, and to intelligent corridor planning models in particular. As we conducted this research, however, we also identified several places in which further work is needed. One particular problem is that while MOGADOR is able to generate a variety of multiobjective corridor alternatives, it is difficult for a population to reach the Pareto optimal front. To solve the problem, additional methods need to be investigated to guide evolution. One possible solution is to seed a GA with results of a SP algorithm on several pre-specified weight combinations.

MOGADOR is also dependent on appropriate representation and problem-specific genetic operations. To develop a more general and problem-independent approach, genetic programming (GP) seems promising since it requires no constraints on representation and each solution is represented as a parse tree. Given a small set of fundamental corridor design elements (e.g. move left, move right, move up, and move down) and functions (e.g. get(), put(), insert(), follow()) to manipulate them, GP is able to create a working computer program from a high-level problem statement.

Finally, natural evolution is a massively parallel search process, working on different species at different locations simultaneously. This concept has been recognized by researchers for years and many GAs implement an island model in parallel computing environments (see Xiao and Armstrong 2003). A distributed and parallel version of MOGADOR can not only take the parallel nature of evolutionary processes into account but also has the potential to reduce computation times significantly.

6. References

- Beasley D., Bull, D. R., and Martin, R. R., 1993, A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1, 101-125.
- Bennett, D. A., Xiao, N., and Armstrong, M. P., 2004, Exploring the geographic consequences of public policies using evolutionary algorithms. *Annals of the Association of American Geographers*, 94 (4), 827-847.

- Berry, J. K., 2004. Map Analysis: Procedures and Applications in GIS modeling, available online: www.innovativegis.com/basis/.
- Brill, E. D. Jr., Flach, J. M., Hopkins, L. D., and Ranjithan, S., 1990, MGA: a decision support system for complex, incompletely defined problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 20, 745-757.
- Cheng, F. Y. and Li, D., 1997, Multiobjective optimization design with Pareto genetic algorithm. *Journal of Structural Engineering*, 123 (9), 1252-1261.
- Coello, C. A. C., 2000, An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32, 109-143.
- Deb, K., 1999, Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evolutionary Computation*, 7, 205-230.
- Deb, K., 2001, *Multi-objective Optimization Using Evolutionary Algorithms*. (Chichester: Wiley).
- De Jong, K. A., 1975, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, doctoral dissertation, University of Michigan.
- Dijkstra, E. W., 1959, A note on two problems in Connection with graphs. *Numerische Mathematik*, 1, 269-271.
- Douglas, D. H., 1994, Least-cost path in GIS using an accumulated cost surface and slopelines. *Cartographica*, 31, 37-51.
- Eastman, J. R., 1989, Pushbroom algorithms for calculating distances in raster grids, In *Proceedings of AUTOCARTO 9*, pp. 288-297.
- Fonseca, C. M. and Fleming, P. J., 1993, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, In *Proceedings of the Fifth International Conference on Genetic Algorithms* (San Mateo: Morgan Kaufmann), pp. 416-423.
- Fonseca, C. M. and Fleming, P. J., 1995, An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3, 1-16.
- Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning* (Reading, MA: Addison-Wesley).
- Goodchild, M. F., 1977, An evaluation of lattice solutions to the problem of corridor location. *Environment and Planning A*, 9 (7), 727-738.
- Hobbs, B. F. and Voelker, A. H., 1978, Analytic multi-objective decision making techniques and power plant siting: a survey and critique, *ORNL-5288*, (Oak Ridge, TN: Oak Ridge National Laboratory).
- Holland, J. H., 1975, *Adaptations in Natural and Artificial Systems*. (Ann Arbor: University of Michigan Press).
- Hopkins, L. D., 1984, Evaluation of methods for exploring ill-defined problems. *Environment and Planning B: Planning and Design*, 11, 339-348
- Hopkins, L. D., 1973, Design method evaluation: an experiment with corridor selection. *Socioeconomic Planning Sciences*, 7, 423-430
- Horn, J., 1997, The Nature of Niching: Genetic Algorithms and the Evolution of Optimal, Cooperative Populations, Ph.D. thesis (University of Illinois at Urbana Champaign).
- Huber, D. L., Church, R.L., 1985, Transmission corridor location modeling. *Journal of Transportation Engineering*, 111 (2), 114-130.

- Huber, D. L., 1980, Alternative methods in corridor routing, MA Thesis (University of Tennessee).
- Jankowski, P., 1995, Integrating geographical information systems and multiple criteria decision-making methods. *International Journal of Geographical Information Systems*, 5, 251-273
- Lai, S-K. and Hopkins, L. D., 1989, The meanings of trade-offs in multiattribute evaluation methods: a comparison. *Environment and Planning B: Planning and Design*, 16, 155-170.
- Lombard, K. and Church, R. L., 1993, The gateway shortest path problem: generating alternative routes for a corridor routing problem. *Geographical Systems*, 1, 25-45.
- Miettinen K. M., 1999, *Nonlinear Multiobjective Optimization* (Boston: Kluwer Academic).
- Pareto, V., 1971, *Manual of Political Economy* (Augustus M. Kelley, New York) translated from French edition of 1927 by Ann S. Schwier, first published in 1896.
- Shier, D. R., 1979, On algorithms for finding the k shortest paths in a network. *Networks*, 9, 195-214.
- Srinivas, N. and Deb, K., 1994, Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2, 221- 248.
- Tomlin, C. D., 1986, *The IBM Personal Computer Version of the Map Analysis Package* (Cambridge: The Laboratory for Computer Graphics and Spatial Analysis, Harvard University).
- Tomlin, C. D., 1990, *Geographic Information Systems and Cartographic Modeling* (New Jersey: Prentice Hall).
- Turner, A. K., 1968, Computer-assisted procedures to generate and evaluate regional highway alternatives. *Joint Highway Research Project, C-36-72A* (Lafayette: Purdue University).
- Xiao, N. and Armstrong, M. P., 2003, A specialized island model and its application in multiobjective optimization. *Genetic and Evolutionary Computation – GECCO 2003* (Volume 2) Lecture Notes in Computer Science 2724 (Springer Verlag: Heidelberg), pp. 1530-1540
- Xu, J., and Lathrop, R.G, 1995, Improving simulation of spread phenomena in raster-based geographic information system. *International Journal of Geographic Information Systems*, 9, 154-168.
- Yen, J. Y., 1971, Finding the k shortest loopless paths in a network. *Management Science*, 17 (11), 712-716