International Journal of Geographical Information Science
Vol. 20, No. 4, April 2006, 401–423

Research Article

An evolutionary algorithm for multicriteria path optimization problems

PETER MOONEY* and ADAM WINSTANLEY
National Center for Geocomputation, National University of Ireland Maynooth, Co. Kildare, Ireland

(*Received 24 March 2005; in final form 27 October 2005*)

For many years researchers and decision makers (DMs) faced with multicriteria shortest path problems (MSPPs) have resorted to reductions to the classical shortest path problem (SPP) by means of weighted linear combinations of the criteria. Algorithmic and approximation schemes are available to solve MSPPs but these approaches often display complexities prohibitive to their implementation on real-world applications. This paper describes the development of an Evolutionary Algorithm (EA) approach to MSPPs on networks with multiple independent criteria. The EA approach is shown to sufficiently explore the underlying network space, generate large candidate path sets, and evolve high quality approximations to the optimal MSPP solution(s). Opportunities for early termination of the EA in time-critical applications are also offered. Among the issues for further work is the integration of the EA as a tool within a GIS for path optimization.

1. Introduction

Real-world optimization problems can rarely be expressed with just one criteria and real life 'is rarely characterized by a position on the real number line' (Corne *et al.*, 2003). Examples of these problems abound: car parking; choosing from a restaurant menu; land-use suitability planning; and public transport journey planning (PTJP). In PTJP one is usually seeking the optimal path of travel between two locations with respect to some costs (criteria) and other addition constraints. Depending on the problem environment the decision maker (DM) may consider many criteria. In PTJP criteria include geographical distance, financial cost of a journey, number of route changes, and overall journey time. Conflicts will occur between criteria: to minimize route changes may require longer journey time but to minimize overall journey time, several changes may be required. As additional criteria are considered, in problems such as this, the decision process becomes much more difficult. For large spatial analysis problems such as route planning and facility location GIS are used given their powerful abilities in acquisition, management, visualization, and analysis of spatial data. Delavar *et al.* (2004) and Chakhar and Martel (2003) remark that typical GIS usually lack 'more powerful analytical tools' enabling DMs to explore the solution spaces for multicriteria problems particularly.

In this paper we consider an extension to a fundamental problem in spatial data handling. When the classical shortest path problem (SPP) is extended to incorporate two or more independent criteria the SPP is transformed into the multicriteria

*Corresponding author. Email: peter.mooney@nuim.ie

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810600607766

shortest path problem (MSPP). For MSPPs involving multiple independent criteria (where all criteria are considered equally important) a unique solution optimizing all the criteria simultaneously will rarely, if ever, exist in reality (Zitzler *et al*., 2003). Given this situation one must be content with solutions that are 'something a little less than optimal', 'compromise solutions', 'near optimal solutions' or 'solutions displaying the best trade-offs amongst the criteria considered'. The MSPP is classified as NP Hard (Gandibleux *et al*., 2004). Historically many MSPPs are *reduced* to an SPP by using a weighted linear combination of all criteria as the cost function. The MSPP does not respond satisfactorily to this reduction in reality. In all but a handful of pathological cases this type of reduction is a radical simplication of a complex problem (Corne *et al*., 2003). It may be very difficult to compute an appropriate set of weightings for the criteria involved. Pereira (2004) states that in most cases 'weightings must be performed by expert analysts'. In the absence of such experts a single criteria is chosen for optimization with the other criteria used as additional constraints. The latter approach may lead to a situation where some optimal solutions are overlooked or left undiscovered (Pereira, 2004). To compound these issues many MSPP approximation schemes (described later in section 2) become unworkable in practice and exhibit complexities prohibitive to their implementation on large graphs or networks.

Recently EAs have been applied to a number of difficult multicriteria optimization problems. Yet only sporadic work has appeared applying EAs to directly to MSPPs. As will be discussed in section 2 there have been many applications of EAs to various path planning type problems. Many involve the linear combination of criteria while others are embedded in spatial coordinate systems. The diversity of application in robot/automated vehicle navigation is certainly a yardstick to the success of EAs as path optimizers. Our EA requirements for MSPP are as follows. The EA will be developed without reduction to optimization of a linear weighted cost function. The EA will be developed without incorporation of SPP algorithms. SPP algorithms will not be integrated into the EA as the EA is designed (using random walking) to generate the solutions to the SPP for the D criteria in the MSPP. The EA should evolve *high quality solution sets* approximating the global optimal set of paths (between a source s and destination node t). Using an effective candidate path generator and genetic operators the MSPP will be approximated without heuristic information. Before EAs for MSPP can be applied to larger spatial network optimization problems (and integrated with a GIS) the suitability of the EA must be quantified on network structures without spatial attributes (also stated by Xiao *et al.* (2002)).

We organize the paper as follows. Section 2 provides an overview of other approaches to MSPP-type problems. Section 3 outlines the formulation of MSPP, the data structures implemented, and an overview of the Evolutionary Algorithm and relevant notation. Details of the experimental analysis are provided in section 4 highlighting the key strengths of the EA approach to the MSPP. The paper closes in section 5 where our conclusions are discussed and a brief discussion of further research directions is provided.

2. Literature review

The application of EAs and GAs to routing problems is not new. The literature review below is divided into two sections. The first section overviews the classical approaches to MSPP and some of the drawbacks of these classical methods. The second section discusses the application of evolutionary computational methods to various types of routing problems.

Skriver and Andersen (2000) employ a label correcting approach to a bicriterion SPP. Their algorithm stores only label-sets and Pareto paths are recovered at the termination of the algorithm but appears to be limited to relatively small networks. In Martins *et al.* (1999) a tree-based algorithm for ranking optimal paths in MSPP is outlined for combinations of two criteria. In Martins and dos Santos (1999) a labelling algorithm for MSPP is outlined. This labelling search tree approach outlined works well in theory but in practice the memory costs of this approach are prohibitive to its implementation. Hallam *et al.* (2001) outline an approximation algorithm for MSPP supplying heuristic information (like that for A^*) to the algorithm. Pareto paths are selected on the basis of their *selection-function* value which contains heuristic and constraint information. Nepal and Park (2003) combine heuristic labelling and exhaustive search algorithms but again their approach is limited to small network specifications.

A GA for the vehicle dispatching routing problem is developed in Baker and Ayechew (2003). The problem is a single criteria problem under a one source to many destinations specification. The initial population is generated using randomly generated path candidates. The authors argue that this randomly generated population (over a more structured approach) 'provides a more diverse population that converges to a near optimal solution quickly.' The problem of path optimization when network information is changing over time is addresed in Davies and Lingras (2003). A GA is used to find the shortest path and alternative backup paths. Roulette wheel selection for crossover with an elitist strategy is employed. A novel form of mutation is used where two points are chosen in the path whereupon a random path is inserted into this subpath. Random walks are used to find some alternative paths. In Shad *et al.* (2003) three shortest path algorithms and a genetic algorithm are analysed on Iranian road networks to assess the suitability of particular algorithms to path finding on different sized networks. The GA is found to perform best on small networks (less than 1000 nodes) but performing poorly on larger sized networks. The fitness function employed is a linear combination of the sum of products of route lengths and associated costs.

The shortest path routing problem for the transportation of hazardous materials is approached using a GA in Huang *et al.* (2004). The GA is used not in the determination of the routes but in the determination of the weights of the factors involved in route choice. A generalized route cost function is used in fitness evaluation with elitism set at a value of 10—allowing only the 10 best solutions from each generation to proceed to the next generation. A GA developed for the adaptive navigation of a robot-like simulation vehicle is considered by Nearchou (1999). The search space is a two-dimensional gridded space. The fitness function is a linear combination of the positional error of the path, the length of the path, the cost of the path, and the possible collisions on the path. Binary tournament selection with elitism is used. Ahn and Ramakrishna (2002) develop a GA for shortest path routing with a single criteria. The authors choose to allow looped paths to enter the population. They estimate that the fitness function (comprised of a linear combination of route characteristics) will 'weed out these bad candidates later in the generational process'. The population is initialized randomly without the use of any classical algorithms. Mutation is the standard gene flip mutation. Crossover is the conventional one-point crossover and candidates for crossover are chosen from pairwise tournament selection without replacement. Quality of the solutions generated by the GA is assessed by analysing how many times the GA generates the Dijsktra path. In the work of Delavar *et al.* (2004) a Pareto-based approach is considered on Iranian road networks. Three criteria are considered—length of route, time to drive the route, and the ease of driving the route. Population initialization for the GA is constrained within a rectangular area on the road network map. The GA does not use any mutation operator. The crossover operator uses a roulette wheel selection incorporating the best individuals and random ones. The authors conclude that routing problems dictate a one-point crossover.

In the work presented by Roy *et al.* (2002) the authors provide a clear and very useful application of EAs to a problem that is multicriteria by its definition. The primary goal of Quality of Service (QoS) routing is to efficiently allocate wireless resources to satisfy these QoS requirements. Conflicts between these individual QoS parameters makes 'the QoS challenge even more difficult'. The authors comment that 'more recently research work in determining QoS multicast routes clearly demonstrate the power of GAs in finding near optimal solutions satisfying the QoS requirements in computationally feasible time'.

Many examples appear in the literature on application of evolutionary computation (EC) approaches to problems in GIS. We do not explicitly consider the advantages and disadvantages of any particular approach but discuss these applications as a strong indication of the suitability of EC to multicriteria problems in GIS. The work of Bennett *et al.* (2003) uses an evolutionary approach to help cartographers create optimal shapes for the geographical and statistical characteristics of choropleth maps. For resolution of conflicts between the location of objects resulting from scale reduction on maps Wilson *et al.* (2003) use GA to search for optimal generalization. GAs are also used by Chemin *et al.* (2004) to estimate pixel-based water/plan parameters in the study of crop productivity indicators from remote sensing data. The *P*-Median problem in Correa *et al.* (2000) is also studied using several different types of GAs. The multicriteria nature of environmental land-use planning requires the generation of many alternative candidate solutions that optimize criteria such as spatial allocation, operations costs and environmental impacts. Several approaches combining evolutionary computation approaches and a GIS have been documented recently with Bennett *et al.* (2004), Matthews *et al.* (2000) and Bjornsson and Strange (2000) prominent amongst these.

3. Prelimenaries and implementations

Suppose a network $G=(V, E)$ is defined such that $\forall\, e \in E : e \rightarrow Q\left(c^1, c^2, \ldots, c^D\right)$ is a vector of edge costs or criteria of size D. The *path description vector* (*pdv*) for any path $P_{s,\,t}$ is calculated by adding the values of each vector q on each edge $e \in P_{s,\,t}$. So for path $P_{s,\,t}$ the path description vector $pdv(P_{s,\,t})=(q_1, q_2, \ldots, q_D)$ is calculated using equation (4). Then q_i is calculated by adding the values of c^i for every edge in the path $P_{s,\,t}$. If required a vector of costs can also be associated with each $v \in V$ which is included in the calculation of the *pdv* for $P_{s,\,t}$. MSPP requires one to 'find a simple path $P_{s,\,t}$ between two fixed nodes $s,t \in V$ such that $pdv(P_{s,\,t})$ is minimized over all valid paths $S_{s,\ t}$'. A simple path is a path between s and t that does not contain any loops or repeated edges.

$$P_{s,\,t} = argmin\ pdv(P_{s,\,t}) \forall\, P_{s,\,t} \in S_{s,\,t}, \tag{1}$$

$$\sum_{(i,\,j)\in P_{s,\,t}} X_{i,\,j} - \sum_{(i,\,j)\in P_{s\,t}} X_{j,\,i} = \begin{cases} 1, & i = s, \\ 0, & \text{for } i \in V - \{s,\, t\}, \\ -1, & \text{for } i = t, \end{cases} \tag{2}$$

$$X_{i,\,j} = \begin{cases} 1, & \text{if edge } (i,j) \text{ is on the path } P_{s,\,t}, \\ 0, & \text{otherwise,} \end{cases} \quad \forall\ edges \quad (i,j) \in E, \tag{3}$$

$$\begin{cases} q_1 = \sum_{(i,\,j)\in P_{s,\,t}} c^1_{i,\,j} X_{i,\,j}, \\ q_2 = \sum_{(i,\,j)\in P_{s,\,t}} c^2_{i,\,j} X_{i,\,j}, \\ \ldots,\ \ldots \\ q_D = \sum_{(i,\,j)\in P_{s,\,t}} c^D_{i,\,j} X_{i,\,j}. \end{cases} \tag{4}$$

MSPP specifies that no path $P_{s,\ t}$ will contain loops and each path considered must have the same source s and destination t. Each incoming edge of a vertex which is on a path P must be matched by an outgoing edge on that path except for vertices s and t (as in equation (2)). All valid paths $P_{s,\ t}$ have the form $P_{s,\ t}=\{p_0=s, p_1, \ldots, p_{k-1}=t\}$ where $k \geqslant 3$ is the number of nodes in the path $P_{s,\ t}$. As the globally optimal solution to a MSPP (in equation (1)) rarely, if ever, exists, an alternative notion of optimality is required. If criteria are incomparable then a pareto optimal formulation of the problem is required. Given k criteria (minimization), two solutions have been computed to a multicriteria problem: $X=\{x_0, \ldots, x_{k-1}\}$ and $Y=\{y_0, \ldots, y_{k-1}\}$ (which are both called *decision vectors*). Then X **dominates** Y *iff* $\forall$i : $\rightarrow x_i \leq y_i$ and $\forall j : x_j < y_j$. For example, suppose we have generated 3 solutions to a $D=2$ MSPP: $S_1=\{38, 56\}$, $S_2=\{50, 60\}$ and $S_3=\{44, 46\}$. The solutions S_1 and S_3 are non-dominated or *pareto-optimal*. This binary relationship partially orders the space of alternative solutions. For each MSPP we must assume a globally optimal set of path solutions (possibly singleton) exists. We denote this as P_{GLOBAL}. The EA must evolve as close an approximation as possible to P_{GLOBAL}. After each generation g the EA outputs P^g_{approx}, the current approximation. The final output of the EA, whether after early termination or after its final generation, is denoted as P_{approx}.

3.1 *Evolutionary algorithm*

The EA (in figure 1) operates as follows. The chromosomes in this study are represented as paths between s and t. Each chromosome is defined as an ordered list of path nodes. Each edge in the path (that is a *gene* in the chromosome) is represented by an edge object containing the start and end nodes of the edge and the vector of costs defined on the edge. An initial population $X_{i=0}$ of size $|N|$ is generated using random walking (described later in section 4.3). The number of generations G is set as are the application rates of the genetic operators crossover C and mutation M.

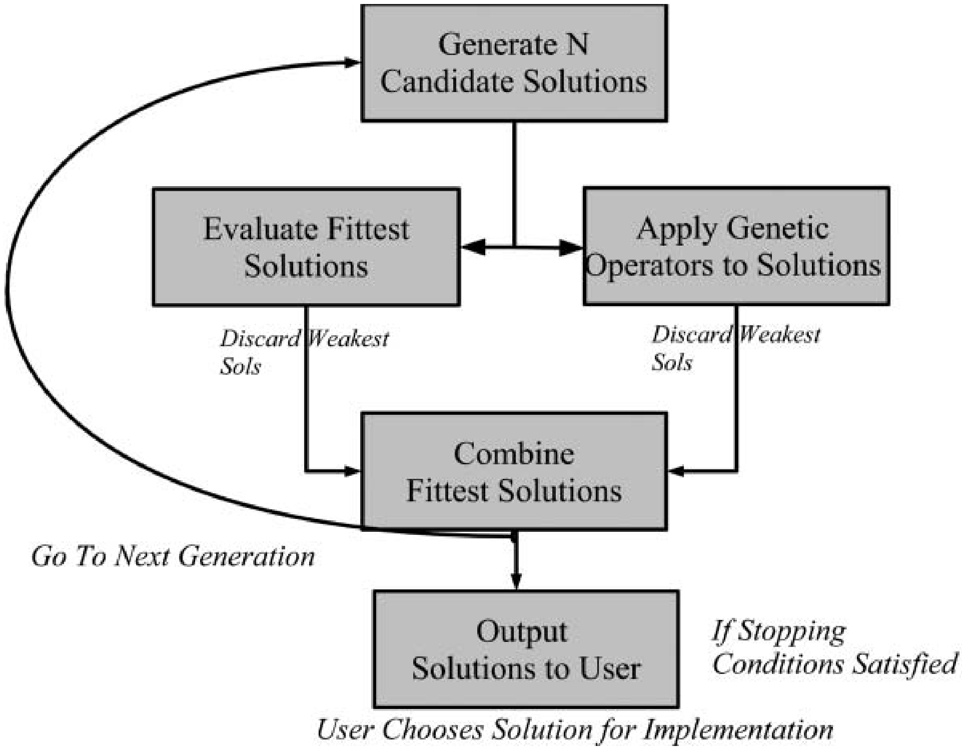

Figure 1. Our EA for MSPPs.

The population X_i is operated on in two ways to determine the *fittest* solutions.

1. Using a *pareto-elitist* approach the set of fittest solutions X_i^{PO} (possibly singleton) in X_i is determined using pareto domination.
2. The original set X_i is copied to X_i^{gen} and the genetic operators are applied to this copy. Crossover (see section 3.3) and mutation (see section 3.2) are applied to X_i^{gen} as determined by the application rates C and M. When genetic modification is complete the pareto optimal set (denoted as X_i^{OS} of X_i^{gen}) is computed.

The genetic operators access a copy of X_i which ensures that *good* solutions are not destroyed by genetic modification. This copy approach is a form of *elitism* and can be seen in evolutionary approaches such as Huang *et al.* (2004) and Knowles and Corne (2000). However, our EA stores the fittest candidates while allowing a copy of them to possibly undergo genetic alterations. This assists *good* solutions to survive longer in the evolutionary process until *better* or *fitter* solutions are discovered or evolved (from genetic modification of an existing solution). Finally for generation i, the pareto optimal set X_i^{F} of the set intersection of X_i^{OS} and X_i^{PO} is computed. This removes any duplication of candidates caused by our elitist approach. This set intersection is denoted as P^i_{approx}. The set X_i^{F} proceeds to the next generation $(i+1)$ and forms the basis of the initial population of the next generation $(i+1)$, where $(|N|-|X_i^{\mathrm{F}}|)$ candidate paths solutions are generated. The EA iterates or evolves in this fashion until the final generation G or until some termination condition(s) (see section 4.4) is satisfied.

3.2 *The path mutation operator*

A valid path $P_{s,\ t}$ is computed between s and t and $pdv(P_{s,\ t})$ is also calculated. For notation sake $P_{s,\ t}$ has the form $P_{s,\ t}=\{p_0=s, p_1, \ldots, p_{k-1}=t\}$, where k is the number of nodes in the path $P_{s,\ t}$. To perform a single node mutation of this path, a node R is chosen at random from the nodes in the path, i.e. $R\in\{\mathrm{Nodes}(P_{s,\ t})-\{p_0, p_{k-1}\}\}$. The index of R is derived from the node indices of the path $P_{s,\ t}$ and is bounded by $r=[1, (k-2)]$. This node R is *mutated* from this path. This involves the removal of R from this path and the insertion of a new *auxiliary* node R^* to replace R. This auxiliary node is chosen such that the path $P_{s,\ t}$ is mutated to a new valid path $P^*_{s,\ t}-\{R\}$. To find the node R^* the set intersection is computed between the node leaving p_{r-1}, denoted by $FS[p_{r-1}]$, and the node on edges incident on p_{r+1} denoted by $BS[p_{r+1}]$. The auxiliary node is extracted from the set $\{FS[p_{r-1}]\cap BS[p_{r+1}]\}$. The default case occurs when the cardinality of $FS[p_{r-1}]\cap BS[p_{r+1}]$ is 1. This means that there is no auxiliary node to replace R in the original path $P_{s,\ t}$ as the intersection is itself the node R. In this case the path $P_{s,\ t}$ cannot be mutated on this particular node. Mutation has the effect of increasing or decreasing the additive costs of the individual elements in $pdv(P_{s,\ t})$ if the mutation operator was successful. The example in figure 2 illustrates this idea with an example of mutation on a path within a component of a larger graph. The mutation operator is concerned with path connectivity relations on the mutation candidate path. It is length preserving on the candidate path $P_{s,\ t}$ but may have the effect of mutating a *good* or *bad* parent path into a *better* or *worse* offspring path. For path optimization problems based in robot configuration space (gridded cells), evolutionary approaches, such as Xiao *et al.* (1997) and Correa *et al.* (2000), use models such as Gaussian mutation to select a neighbouring cell as an auxiliary node.

3.3 *Path crossover operator*

'Almost every crossover operator chooses two solutions at random and some portions of the solutions are exchanged between the solutions to create a new solution' (Deb, 2001, p. 89). The chance of creating better solutions is 'far better than random'. In this model two valid distinct paths $P_{s,\ t}$ and $Q_{s,\ t}$ must be available as the parent paths for crossover. Suppose that $P_{s,\ t}$ is defined as $P_{s,\ t}=\{p_0=s, p_1, \ldots, p_{k-1}=t\}$ with k nodes and $Q_{s,\ t}$ is defined as $Q_{s,\ t}=\{q_0=s, q_1, \ldots, q_{j-1}=t\}$ with j nodes. $pdv(P_{s,\ t})$ and $pdv(Q_{s,\ t})$ are also computed. The crossover operator takes two valid paths and splits them at two randomly chosen points. Recombination takes place to concatenate the two split subpaths. The index x of the split point in $P_{s,\ t}$ is in

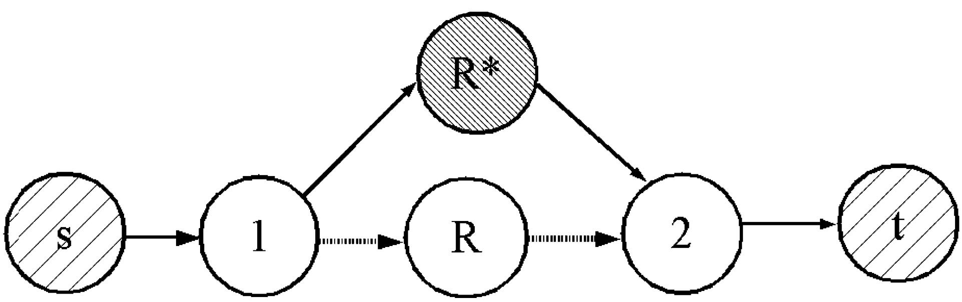

Figure 2. An example of single-point mutation on a path.

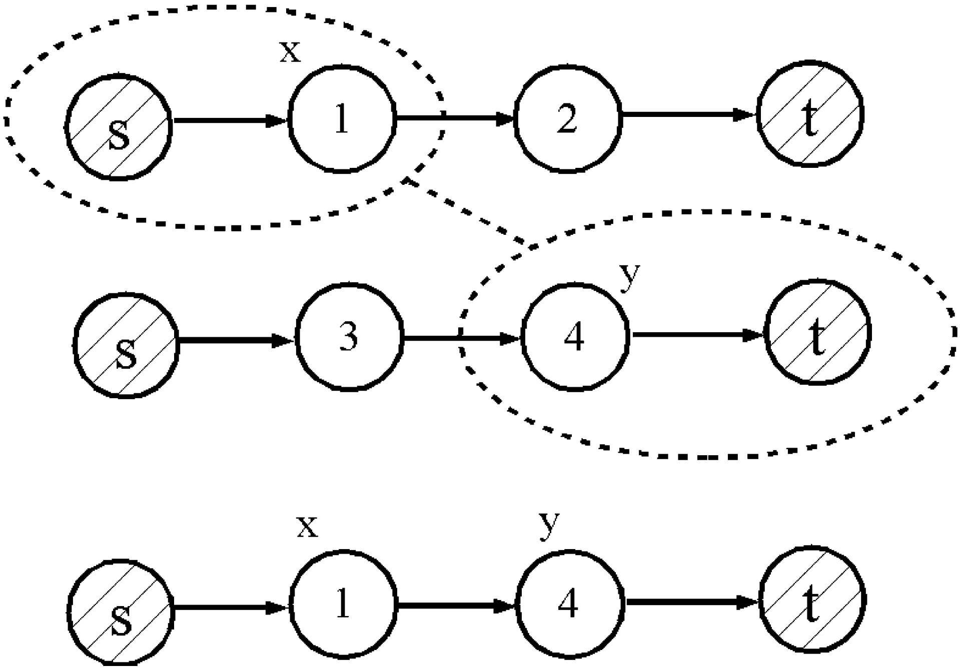

Figure 3. An example of single-point crossover.

the range [1, …, $(k-2)$]. The index y of the split point in $Q_{s,\ t}$ is in the range [1, …, $(j-2)$]. Both x and y are chosen randomly within this range. The nodes s and t cannot be chosen in either path as the (origin, destination) characteristic of the path may be destroyed by the crossover operator and hence introduce instability and illegal paths (on the $(s,\ t)$ pair requirement) into the evolutionary process. The subpaths $P_{s,\ x}$ and $Q_{y,\ t}$ are extracted and are recombined. The child offspring path $P_{s,x}+Q_{y,t}=PQ^{*}_{s,\ t}$ is created where + is the concatenation of the two paths provided $(x,\ y)\in E$. We assume that nodes are indexed in each path from 0, …, $(\alpha-1)$ where α is the total number of nodes in a path (including origin and destination nodes). An example of single-point crossover is shown in Figure 3. Should $(x,\ y)\neq E$ then this particular crossover is not performed and a new crossover is attempted. Crossover randomly chooses its split points in the parent paths. It may therefore evolve *better* or *worse* candidates from a given pair of parents. There are many ways to implement crossover within an EA or Genetic Algorithm (GA). Our approach is a widely used approach called *pairwise tournament selection without replacement*. This means that two candidate solutions are chosen at random. The fitter of the pair is chosen to be one of the parents. The same candidate solution should not be chosen twice as a parent and therefore crossover candidates are not re-introduced into the population. This approach has been successfully implemented in several closely related works including Ahn and Ramakrishna (2002), Huang *et al.* (2004) and Dozier *et al.* (1998).

4. Experimental analysis

As stated above the following experimental analysis is required *before* EAs can be applied to spatial networks as it is important from a graph theoretic viewpoint to quantify the performance and suitability of EAs for MSPP on non-spatial network models. The objectives of the experiments are as follows: (1) to measure the ability of the EA to cover the network search space in MSPP; (2) to assess the effectiveness of early termination of the EA; (3) to assess the quality of the optimal path approximations provided by the EA and (4) to compare the EA against Dijkstra's algorithm on several real world road networks.

4.1 *Experimental setup*

The EA is implemented in the Java programming language. A Pentium 4 with 2.26 GHz processor and 256 Mb of RAM running SUSE Linux 8.1 was used for all experimentation. Without loss of generality in all experiments all D criteria expressed in the vector of costs (or criteria) on edges in the networks are equally important. This assumption is derived from Zitzler *et al.* (2003) where all D criteria are simultaneously optimized when no additional knowledge is available about the problem. Every EA is allowed to run to completion of its designated number of generations. Where applicable the EA highlights the times during evolution where early termination conditions are satisfied. The fitness of path $P_{s,\ t}$ is calculated by testing the pareto optimality of $pdv(P_{s,\ t})$ against the current fittest individuals. The EA does not directly deal with invalid paths. Components controlling random walk generation, crossover and mutation deal with invalid paths. Invalid paths are not permitted to enter candidate sets in the EA. The only mechanisms available for the generation of candidate paths for the EA are random walks combined with mutation and crossover operators. No classical shortest path algorithms are used to generate members of candidate sets. SP and SFN are generated for experimentation with arc-node characteristics of real world networks having arc-node ratios in the range 2.66…3.28 as outlined in Zhan (1998). For all experimentation outlined below this arc-node ratio range is used. Table 1 illustrates the arcnode characteristics of several real world networks.

A significant problem in designing an EA (or GA) is the determination of the correct values for the control parameters (generations, number of candidates, crossover, mutation, termination conditions, etc.). There is no formal theoretical methodology for this problem since different combinations lead to different characteristic behaviour of the EA. Traditionally, parameter determination is achieved through exhaustive experimental work (Eiben *et al.*, 1999). The final settings used were: number of generations 100; candidates per generation 40–80; crossover 0.6; and mutation of 0.25. The experiments detailed in section 4.3 involve testing the random walk component of the EA on SPRAND and SFN. Section 4.4 looks at the feasability of early termination of the EA. The quality of solutions *evolved* by the EA is addressed in experiments in section 4.5. The EA is applied to a number of real world road networks for the experimentation in section 4.6.

Table 1. Arc-node ratios of selected Real World Networks.

Source	Network type	Ratio
Magoni and Pansiot (2001)	*The Internet*	2.6
Case (2001)	*Electicity Power Grid*	2.67
Zhan (1998)	*Road Networks in USA*	2.66…3.28
Jacob *et al.*(1998)	*Texas Road Network*	2.86

4.2 *Types of experimental data*

Three network types are used in experimentation—two artificial networks and one set of real world networks. The artificial networks contain *scale free networks* (SFN) and the Shortest Path Library (SPLIB) of Cherkassky *et al.* (1996). Recent studies, such as Kim *et al.* (2002), on SFN reveal that a 'small world phenomena' exists revealing that two distinct vertices are usually connected by a remarkably small number of edges with the structure of the WWW conforming to that of SFNs. In SPLIB the SPRAND networks are strongly connected random networks. For experimentation we can generate SFN and SPLIB given a specified number of nodes and edges or number of nodes and network density. The vector of costs on each edge in the network is generated randomly. The third type of network we consider is a set of real-world networks. A reliable and convenient source of such data (available to public users) was obtained from the online data centre for the National Highway Planning Network (NHPN, 2003) of the United States of America. NHPN is a 1:100 000 scale network database containing line features representing current and planned highways in the USA. Some authors, Noon and Zhan (1996) for example, have used the NHPN as a source of real world networks for testing purposes. The TIGER online data centre (US Census Bureau, 2003) provides TIGER/Line files from a digital database of geographic features, such as roads, railroads, rivers, lakes, legal boundaries, census statistical boundaries, etc. that cover the entire United States. For the purposes of this work we downloaded the transportation networks for Florida, Louisiana, Utah and Texas.

All networks are stored in the forward and backward-star data structures (see Ahuja *et al.* (1993)). All genetic operators and components of the EA have access to these data structures. We do not consider any spatial embeddings of the real world networks: that is the spatial information or spatial coordinates associated with nodes and edges in $G=(V, E)$ is ignored. Length and geographic distance information *between* nodes is retained. The SPRAND and SFN are generated with densities resembling those of real-world networks in table 1. To provide adequate testing of the EA we require the origin destination pairs (s, t) be *far apart* in the network. Finding nodes *far-apart* in a geographical network is relatively straightforward. However the Euclidean or Manhattan metrics that could be used in geographical networks of NHPN cannot be applied to SPRANDs or SFNs. These networks are not embedded in any spatial co-ordinate system. To counteract this we use the concept of *geodesics* to compute a set of nodes at *maximal edge distance* from the source. This is implemented as a pre-processing step in experimentation below. Geodesic distances indicate the distances between nodes in an *edge sense*.

4.3 *Random walking and the EA*

All EAs require a diverse set of candidate solutions in order for the evolutionary process to be effective (Zitzler *et al.*, 1999). Our EA (described above in section 3.1) requires a diverse set of candidate paths for (s, t). The EA will require approximately $G*N$ (N paths for each generation $i \in G$). Further valid candidate paths are generated by the crossover and mutation operators. For typical values of $G=50$ and $N=100$ the EA must employ a dedicated candidate path generation component. Costelloe *et al.* (2001a) suggest using *random walks* as an effective and robust means of generating large quantities of candidate paths for EAs. We have chosen random walking (RW) for the EA as it allows the EA to explore the graph space in a manner not available to algorithmic approaches to MSPP. RW will generate a more diverse set of paths than classical approaches (such as Breath-First-Search, A*, etc). These approaches are limited in the number of unique paths $P_{s,\ t} \in S_{s,\ t}$ that they can generate as their tendency is to enumerate only paths extracted from the spanning tree rooted at the source node *s*. Given the candidate path requirements of the EA the RW component enumerates large quantities of paths $P_{s,\ t} \in S_{s,\ t}$ but does not attempt to generate *all* paths $S_{s,\ t}$. Generating all paths $S_{s,\ t}$ is computationally unrealistic and would reduce the EA to brute-force enumeration. A path repair function is built into the RW component to ensure illegal paths (those with loops) are not entered into the EA. The use of the repair function does not require the EA to enforce penalties on the fitness of illegal paths.

This experiment quantifies the effectiveness of RW by measuring 'coverage'—the total number of unique edges and nodes that are members of *any* valid simple path used by the EA. In these experiments the coverage was measured using parameterizations of the EA ultimately yielding approximately 2000 and 3000 candidate paths. Both SP and SFN networks are generated with between 500 and 25 000 nodes. As before the EA evolves P_{approx} for $D=3$ MSPP between geodesic nodes. Figure 4 illustrates the percentage of nodes in a given network that are visited (or covered) by the EA under the parameterizations above. In both cases for networks containing up to 5000 nodes coverage rates over 95% are achieved. For very large networks node coverage rate is above 50%. Similarly figure 5 illustrates

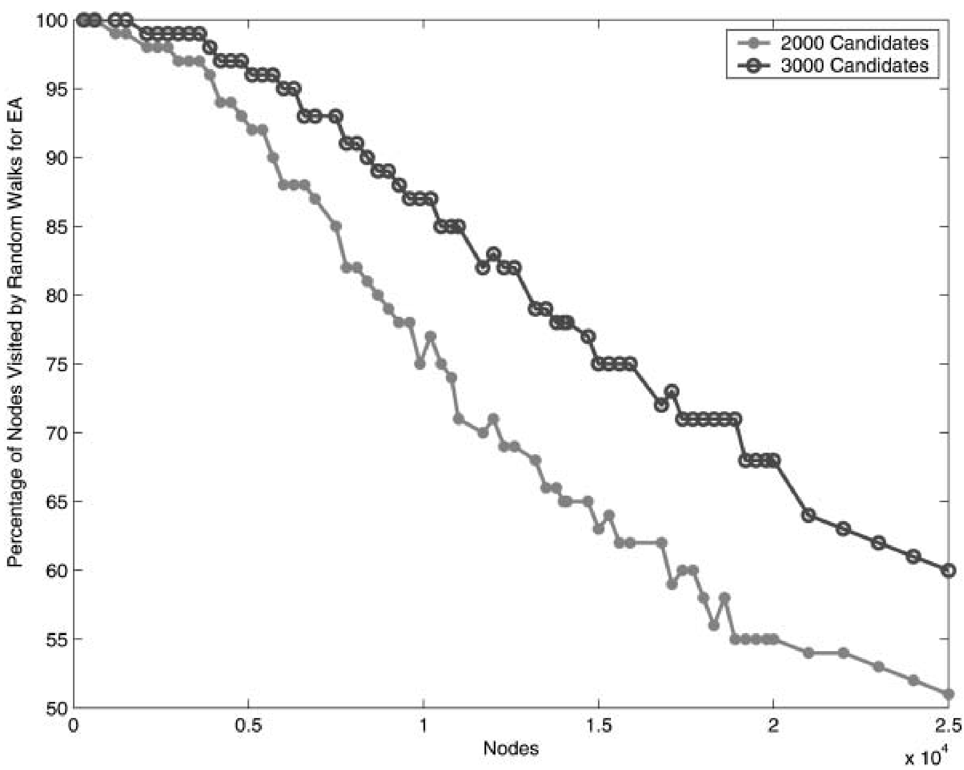

Figure 4. Node coverage by the EA using random walks on SP networks.

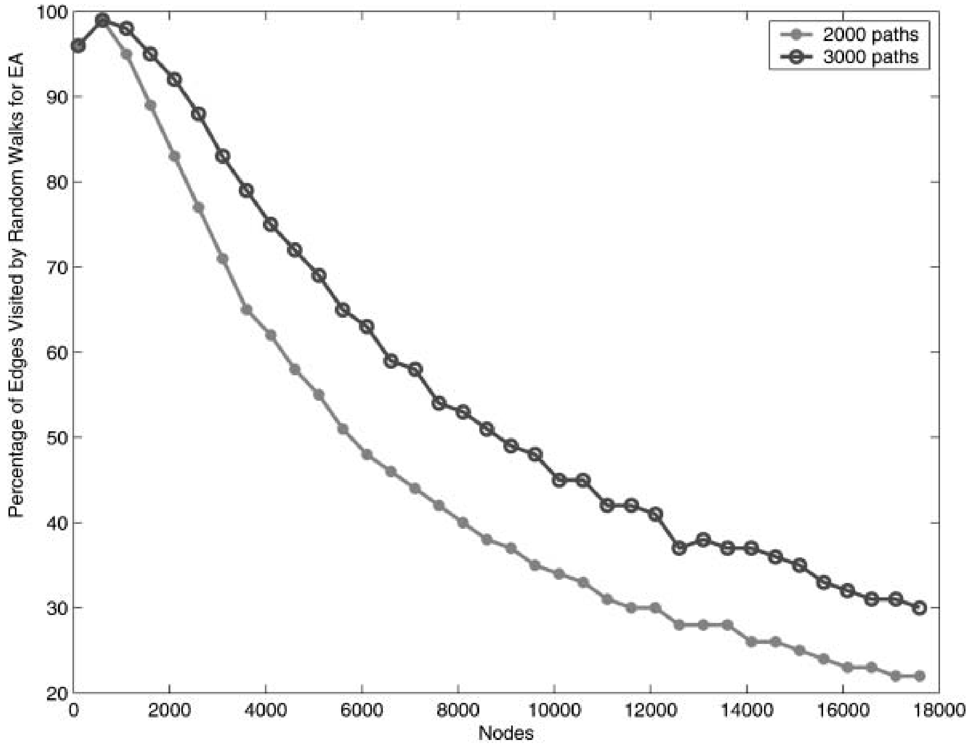

Figure 5. Edge coverage by the EA using random walks on SP networks.

the percentage of edges in the same networks covered by the EA. For networks with up to 3000 nodes over 90% of edges are visited. Edge coverage rates of over 20% for very large networks are also achieved. These coverage rates are very satisfactory, particularly for large instances of MSPP. As outlined in Xiao *et al*. (1997) 'it is unlikely that all optimal paths would require a large number of nodes. Even in quite complex environments an optimal path might be quite simple'. Given that the networks are not spatially embedded there is no way the EA can be constrained to search in specific neighbourhoods or components of the network. Consequently in spatially embedded networks it may not be necessary to achieve very high coverage rates.

4.4 *Early termination of the EA*

Terminating conditions can incorporate elaborate decision making surrounding the termination of the EA. On the other hand many simple terminating conditions are available offering several options to DMs. Terminating conditions may involve but are not limited to characteristics such as:

1. the total CPU time passed since the beginning of the EA;
2. the EA has executed a predefined number of generations;
3. path constraints, extracted from path characteristics, have been satisfied or violated;
4. a predefined number of pareto optimal solutions have been computed;
5. quality measurements of the current P_{approx} have satisfied the DM;

6. P_{approx} has remained unchanged over α consecutive generations.

The first four terminating conditions are easily integrated into the EA. We now investigate the *generational change parameter* α as a strong stopping condition. The idea of this parameter is to terminate the EA early if the solutions in P_{approx} have remained unchanged after approximately $\alpha \times G$. This terminating condition assumes that if P_{approx} is not changing and has remained stable for $\alpha \times G$ generations then the EA can be terminated early. This is based on the assumption that no new pareto optimal solutions will be found which will change P_{approx}. If there are new solutions found there will only be a very small number of *new* solutions. The key characteristic of this metric is the number of solutions *lost* by the EA as a consequence of terminating early based on α. A more detailed discussion of quality is provided in the next section.

To quantify the effectiveness of parameterizing the EA to terminate early based on this terminating condition we used various α parameters. Large values of α (those over 50% of G) proved wasteful in terms of overall computation time. Small values (those less than 20% of G) often terminated too soon and lost a large number of potential solutions that were undiscovered at the cut-off generation. Figure 6 shows the results of 1600 runs of the EA on geodesic $D=3$ MSPP on both sparse and dense SFN and SPRAND respectively with between 500 and 10 000 nodes. The y axis in both figures denotes the number of solutions *lost* by terminating early. This was achieved by comparing $P^{\text{cut-off}}_{\text{approx}}$ and P^{G}_{approx} where *cut-off*$<G$. In the case of sparse networks (figure 6) over 75% of early terminations after α generations resulted in

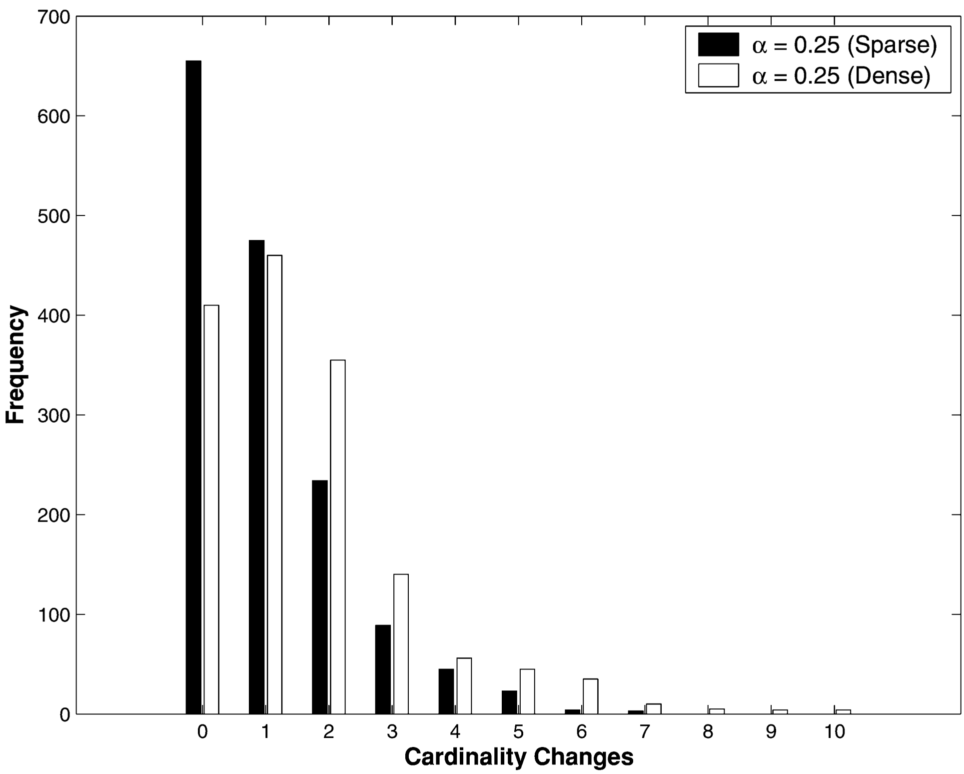

Figure 6. Changes in cardinality for sparse and dense networks.

$P^{\text{cut-off}}_{\text{approx}} = P^{G}_{\text{approx}}$ or $P^{\text{cut-off}}_{\text{approx}}$ discovering all but one solution in P_{approx}. For dense networks losing 1 solution is the most frequent result of this type of early termination. However the results are still very positive. For dense networks 60% of early terminations at α resulted in $P^{\text{cut-off}}_{\text{approx}} = P^{G}_{\text{approx}}$ or $P^{\text{cut-off}}_{\text{approx}}$ omitting 1 solution. We believe the frequency of times $P^{\text{cut-off}}_{\text{approx}}$ lost 2 solutions correlates with the density of the networks.

4.5 *Assessing the quality of the EA approach*

Despite the very encouraging results in the previous section, *generational change* stopping conditions display a major drawback in some instances. While the EA has settled on a P_{approx} there is no quantitative measurement of quality. In any GIS problem a rational DM will require some measure of solution *goodness*. The notion of performance and quality of the EA approach to MSPP includes characteristics of the evolved path solutions in P_{approx} as well as the computational resources needed to generate these solutions. With the latter many of the problems are common in both single and multicriteria optimization—correct selection of data structures, fitness evaluations and disk operations. In SPP quality is defined by means of an objective function. Clearly paths are ranked by this function and the path(s) with the lowest value of this function represent the best solution(s) or the highest quality solution(s). Zitzler *et al.* (2003) state that ‘it is not clear what quality means in the presence of several optimization criteria’. In the case of this work on MSPP several specific aspects of the evolved solutions could be considered as measures of quality. These include the cardinality of the final P_{approx}, the closeness of P_{approx} to P_{GLOBAL}, or the search space coverage of the solutions. Ideally some easily communicated quantitative measurement of quality is desired for most DMs. There are several types of quality metrics to assess multicriteria solutions. Many quality indicators or metrics require knowledge of P_{GLOBAL} for the problem in hand in order to measure how close the current approximation is to the globally optimal solution(s). As it is infeasible to generate P_{GLOBAL} for MSPP we seek the best approximation possible as a respresentation of P_{GLOBAL} which we denote as P_{true}.

To generate P_{true} we implemented an *ad hoc* approach involving several different approaches. A geodesic pair of nodes (s, t) is chosen. Then a combination of Dijkstra’s algorithm (optimizing separately on each of the D criteria), the k-shortest path algorithm, and the EA (executed several times) are implemented to solve MSPP between these nodes. After all algorithms had terminated the unique paths from all runs of the EA and the classical approaches are combined as P_{true}. This ensures that P_{true} contains *all* of the D paths computed by Dijstra’s algorithm on each of the D criteria and other pareto optimal solutions generated by the EA and the k-shortest path algorithm. The approach we use to generating P_{true} is similar to that used in Okabe *et al.* (2003) and Baran *et al.* (2001) For a more indepth coverage of quality metrics for multicriteria optimization the work by Van Veldhuizen and Lamont (2000) provides examples of a wide range of metrics.

Our quality experimentation is divided into two distinct experiments. Generating P_{GLOBAL} is NP-Hard (Gandibleux *et al.*, 2004) and generating P_{GLOBAL} is exponential in the worst-case (Skriver and Andersen, 2000). Given these discouraging aspects of the problem we decided that an interesting measure of quality would be to analyse the frequency which the EA evolved $P_{\text{approx}} = P_{\text{true}}$. Our second set of experimentation is motivated by the lack of support for MSPP type problems in current GIS (as mentioned also by Chakhar and Martel (2003) and Delavar *et al.* (2004)). Current users of most GIS software have a network optimization extension available. These extensions allow users to retrieve (at most) D pareto solutions for a MSPP in a GIS by running the Dijstra algorithm component separately for each criteria D. This will retrieve the pareto solutions at the extrema of the pareto frontier for the given MSPP. To motivate the use of the EA over this approach we analyse how the EA performs in comparison to the combination of multiple runs of the Dijkstra algorithm on the real world networks mentioned in section 4.2.

4.5.1 Evolving $P_{\text{approx}}=P_{\text{true}}$. Experimentation was carried out on the following types of networks. SFN and SPRAND with $D=3$ and $D=4$ criteria were used. Each network had density parameter 2.66…3.28 (as in table 1). Networks ranged from 100 nodes to 3500 nodes. 50 distinct pairs of geodesic (s, t) nodes were computed for each network. The EA then evolved approximations to MSPP for each geodesic pair. The variable $cd(P_{\text{app}}, P_{\text{true}})$ indicates the difference in path solutions between P_{approx} and P_{true}. A value of $D(P_{\text{ap}}, P_{\text{true}})=0$ indicates that the EA evolved $P_{\text{approx}}=P_{\text{true}}$ for this MSPP. If $cd(P_{\text{ap}}, P_{\text{true}})=n$ with $n\geqslant 1$ then P_{approx} did not contain n of the path solutions contained in P_{true}. The results are very impressive. For all networks the EA evolves $P_{\text{approx}}=P_{\text{true}}$ in over 80% of the experiments. Figures 7 and 8 detail a summary of all of these experiments. The x axis shows the number of nodes in the networks while the y axis shows the number of

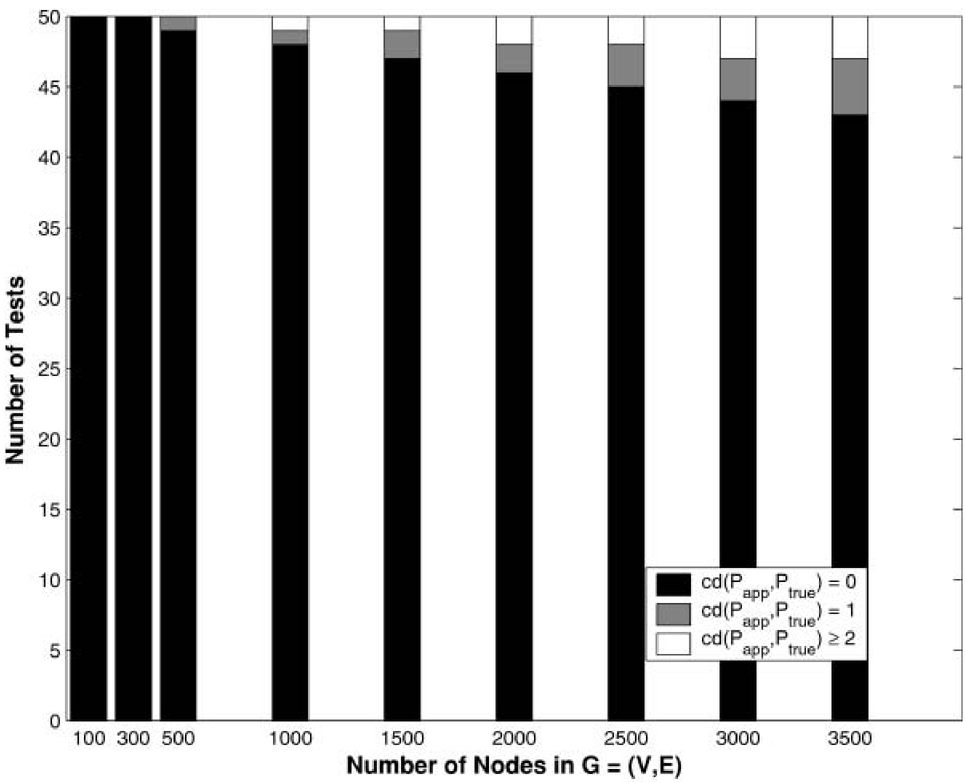

Figure 7. Quality test results from the EA on 3D networks.

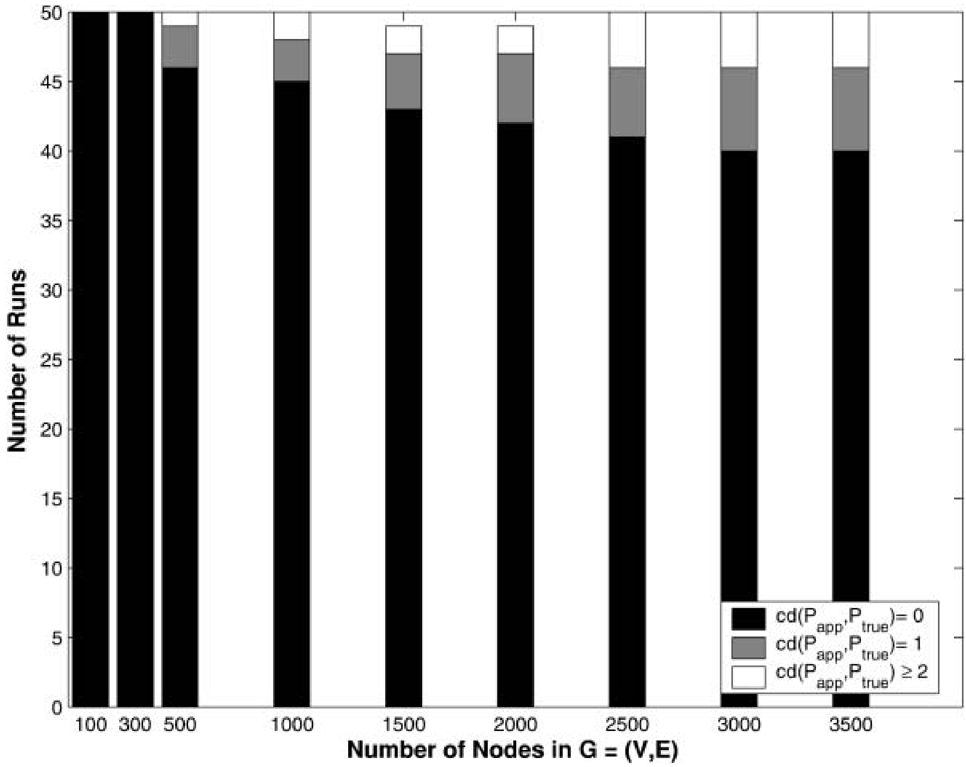

Figure 8. Quality test results from the EA on 4D networks.

experiments for which the three different classifications of $cd(P_{\text{app}}, P_{\text{true}})$ occurred. We believe that results with $cd(P_{\text{ap}}, P_{\text{true}})=1$ are also very valuable as in the case of large P_{true} this is a significant result. If the EA evolves $cd(P_{\text{ap}}, P_{\text{true}})=1$ for large P_{true} then this is considered a very good approximation to the P_{true} of the MSPP in hand, due to the presence of almost all the solutions from P_{true}. Establishing the correct number of solutions (in P_{approx}) to present to any DM is a topic of debate in the area of multicriteria optimization. In fact it may not be necessary to always generate every solution in P_{true} nor to display all P_{true} to a DM. Messac and Mattson (2002) state that 'requiring a DM to manually compare more than ten solutions is never desirable'. In other cases, Skriver and Andersen (2000) remark, the DM might be satisfied with only the set of extreme efficient paths (D paths for any $D \geqslant 2$). The experimentation does not look at very large networks because our approach to generation of consistent P_{true} becomes burdensome and time-consuming. Consequently we feel that as the networks get very large, this approach may make it difficult to rely on P_{true} as an accurate approximation to P_{GLOBAL}.

4.6 *Real-world road networks*

To place the runtime performance of the EA into context for GIS users the EA was compared against Dijkstra's algorithm on real-world road networks. To accurately compare the two approaches Dijkstra's algorithm was configured to terminate upon finding the shortest path from s to t while optimizing on criteria $i \in D$ while the EA

Table 2. Real-world network datasets used in experimentation.

Dataset	Nodes	Edges	Dijkstra time (s)
Utah	1124	3623	4.03
Florida	2155	6370	6.74
Louisianna	2437	6876	6.87
Texas	2103	6027	6.67

terminates upon finding the Dijkstra solutions. Dijkstra's algorithm also used the same data structures for storage and manipulation of the network data structure as the EA. The road network specifications are outlined in table 2. Figure 9 shows boxplot results of the EA running on real-world road networks.

Each box represents the run-times of 100 separate runs of the EA onVfour networks where the EA terminated when all D Dijkstra solution paths were generated. This analysis was carried out on road networks from the states of Texas, Louisiana, Utah and Florida. The runtimes for the EA and Dijkstra's algorithm exclude the time both approaches require to load the input network. The mean cumulative runtimes for Dijkstra's algorithm are also tabulated in table 2. Using this information the performance of the EA against Dijkstra's algorithm on these networks is very encouraging with the mean runtimes almost the same as the Dijkstra combination. The outliers and elongated shape of the box plots can be attributed to the RW mechanism. Without any spatial guidance (and without using Dijkstra's algorithm's greedy approach) the runtime is wholly dependent upon the

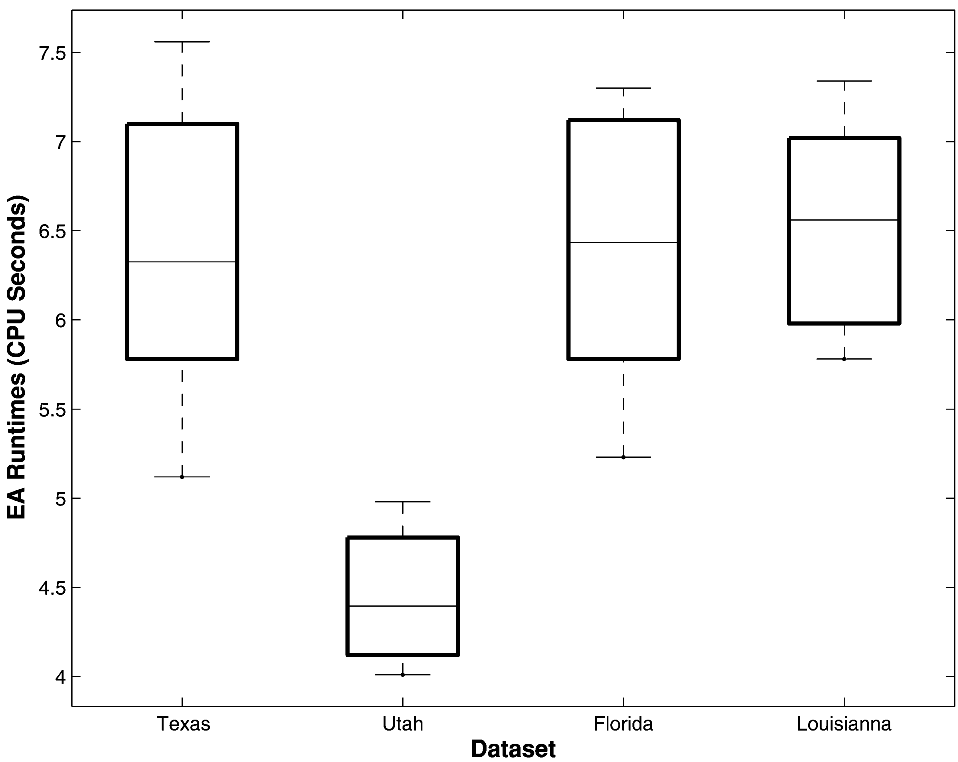

Figure 9. EA running times on real-world road networks.

random nature of the evolutionary search. It is also very encouraging that the overall spread of runtimes of the EA are constrained within an acceptable timespan.

Figure 10 shows the mean running times (in CPU seconds) of Dijkstra's algorithm and the EA on SPRAND and SFN of 100 to 10 000 nodes (with densities as described in table 1). The same terminating condition was used as before with the EA terminating when all of the *D* Dijkstra paths were generated. The results are impressive and highlight the ability of the EA approach to compete with Dijkstra's algorithm on the same MSPP. For networks without up to 1500 nodes there is little to choose between the two approaches. The performance of the EA becomes more pronounced as the networks get larger.

5. Conclusions and future work

Researchers and practitioners acknowledge the importance of MSPP on network spaces yet network-based MSPP have received only sporadic and isolated attention from researchers over the years. The reality remains that a truly multicriteria approach to MSPP is often abandoned in favour of a simpler single criteria approach under pressures of time and management (Zeleny, 1982). Given these real-world pressures an *ad hoc* development of an EA (given their recent popularity for multicriteria problems (Goldberg, 1999)) for a given MSPP may not be an immediately profitable approach. The MSPP must be completely transformed to the EA domain. The transformed MSPP must allow the generation of successive

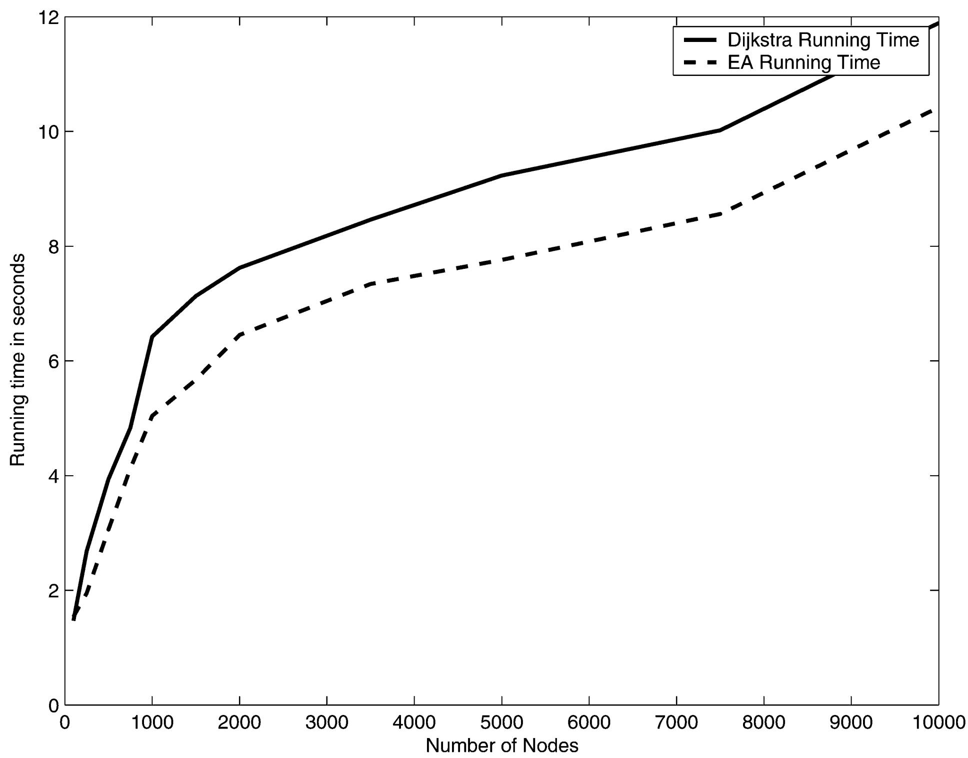

Figure 10. Running times of Dijkstra's algorithm and EA on SPRAND and SFN.

generations of candidate solution populations, genetic operators, and the use of evolutionary stopping conditions.

This paper has described an EA for MSPPs and operates on a classical graph model of the network in the given MSPP. Spatial information can be used for the vector of criteria on edges and nodes in the network. However the EA does not use the spatial coordinates of nodes and edges to guide the evolutionary search. This EA was tested on a suite of artificially generated networks and several real-world road networks. The artificially generated networks allowed the EA to be examined on very large network structures while real-world networks were used to examine the behaviour of EA on networks exhibiting real-world connectivity. Geodesics are used to choose experimental source and destination node pairs. Random walking is shown to be an effective approach in generating a diverse set of candidate path solutions for the EA. The EA uses these candidates for initialization of the population and addition to the population throughout the generational process. For large networks random walking is shown to visit 50% of nodes and over 30% of edges. The networks considered in section 3, both artificial and real-world networks, retained real world characteristics (in terms of connectivity and density). The EA was shown to process networks of up to 20 000 nodes effectively in acceptable run-times. Other approaches, such as Delavar *et al.* (2004) and Shad *et al.* (2003), perform poorly as the sizes of underlying networks grow. All of the D criteria are explicitly considered as independent rather than using a fitness function comprised of linear combinations of the path characteristics. This is an advantage over approaches such as Ahn and Ramakrishna (2002).

Early termination of the EA is possible under a number of suitable termination conditions. Early termination of the EA based on a generational-change parameter (no change in P_{approx} over $\alpha \times G$ generations) is shown to produce useful results. This simplistic approach of *generational change* is acceptable once users are content with the possibility that by saving overall CPU time they may lose out on potentially better solutions that the EA has yet to *evolve*. However, *generational change* allows no quantification of the quality of the evolved P_{approx}. To deal with this situation experimental analysis is provided regarding the generation of a good approximation to P_{GLOBAL} for various MSPPs and then analysing the P_{approx} evolved by the EA. An *ad hoc* approach using a combination of the EA, k-shortest paths algorithm, and Dijkstra's algorithm was used to generate P_{true} (an approximation to P_{GLOBAL}). The measure of quality extracted from this experimentation was the number of solutions in the P_{approx} also included in P_{GLOBAL}.

To evaluate the effectiveness of the EA in a GIS the EA was tested on a set of networks with which many GIS users are familiar. Several real world networks were chosen for this experimentation. The run-time (in CPU seconds) required by the EA to evolve all D Dijkstra solutions to a D-criteria MSPP on these real-world networks was compared to that of combining the D individual runs of Dijkstra's algorithm on the same network sharing the same data structures. The results clearly demonstrate the EA competing very well with the combined Dijkstra approach. The clear advantage held by the EA in this situation is that in many cases the EA evolves $>D$ solutions—that is solutions inside the hypervolume created by the Dijkstra solutions at the extreme points of the pareto frontier. This is clearly a significant advantage of this EA approach in situations where alternative *backup* paths are required in addition to the Dijkstra solutions (Gandibleux *et al.*, 2004). This also explains the spread of the boxplots in figure 9 where extra CPU time is spent evolving both the

Dijkstra solutions and other pareto optimal solutions. The performance of the EA on the real-world networks is particularly important as it competes with Dijkstra's algorithm but without incorporating any spatial attributes of the networks. This is a good indication that research on EAs incorporating this spatial information will be beneficial. This graph theoretic approach to applying an EA to a MSPP is essential to future developments in this area. Several studies (such as Delavar *et al*. (2004) and Costelloe *et al*. (2001b)) have looked at EAs on real-world road networks but the more fundamental study (as provided by this paper) was missing from the literature.

Based upon our observations from the experimentation, the behaviour of the EA is summarized into three distinct categories.

- **Type 1:** The EA almost immediately computes a high quality (measured by some predefined set of quality metrics) approximation P_{approx} to P_{true}. After the first few generations, few if any changes are observed.
- **Type 2:** The EA quickly settles on P_{approx}. Several generations pass without any changes. Another burst of evolutionary activity occurs where new solutions are generated and enter P_{approx} or old solutions are dominated by new solutions and are deleted from P_{approx}.
- **Type 3:** The EA never settles on a particular P_{approx}. In this case the evolutionary search steadily finds new improvements to P_{approx}. In this case P_{approx} receives additions and removals until a few generations before the termination of the EA.

The results (particularly in section 4.4) indicate that the behaviour of the EA when approximating the optimal solution to a geodesic MSPP is classified as Type 1 or Type 2 in most instances. Type 3 behaviour does not occur frequently and our experimentation reveals that this behaviour only occurs about 5% of the time. The Type 3 behaviour of the EA is unpredictable and cannot be forecast *a priori* thus effectively evading most terminating conditions. Work is underway to investigate factors influencing the behaviour *type* classification of the EA *a priori* for certain MSPPs.

From a practical viewpoint the integration of the EA into a GIS for emergency response routing could be explored. The EA operates on the current real-time network configuration with travel time, traffic density, etc. used as conflicting criteria. In our case the EA could be employed in the GIS as a specialist spatial analysis tool to generate approximations to spatial MSPPs. The GIS would provide the spatial network data, user interface (for journey/route selection), and visualization of routing alternatives from P_{approx}. Large quantities of attribute data could be handled by the GIS. Integration with a GIS will require the exploration of ways to allow users to set parameters (such as crossover and mutation) without needing a deep understanding of Evolutionary Computation. Instead the EA would be used extension-plugin type fashion.

References

AHN, C.W. and RAMAKRISHNA, R., 2002, A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, **6**(6), pp. 566–579.

AHUJA, R.K., MAGNANTI, T.L. and ORLIN, J.B., 1993, *Network Flows: Theory, Algorithms, and Applications* (Englewood Cliffs, NJ: Prentice Hall).

BAKER, B. and AYECHEW, M., 2003, A genetic algorithm for the vehicle routing problem. *Computers and Operations Research*, **30**(5), pp. 787–800.

BARAN, B., VALLEJOS, J., RAMOS, R. and FERNANDEZ, U., 2001, Reactive power compensation using a multi-objective evolutionary algorithm. In *IEEE Porto Power Tech Conference*, 10–13 September 2001, Porto, Portugal (Piscataway, NJ: IEEE Press).

BENNETT, D., XIAO, N. and ARMSTRONG, M., 2003, Using genetic algorithms to create multicriteria class intervals for choropleth maps. *Annals of the Association of American Geographers*, **93**(3), pp. 595–623.

BENNETT, D., XIAO, N. and ARMSTRONG, M., 2004, Exploring the geographic consequences of public policies using evolutionary algorithms. *Annals of the Association of American Geographers*, **94**(4), pp. 827–847.

BJORNSSON, C. and STRANGE, N., 2000, Heuristic allocation of wetlands in GIS. In *ESRI GIS 20th Annual User Conference*, 26–30 June 2000, San Diego, California, USA, pp. 1–10 (Palm Springs, CA: ESRI Press).

CASE, J., 2001, The continuing appeal of small-world networks. *SIAM News—Community Lecture 2001*, **34**(9), pp. 1–4.

CHAKHAR, S. and MARTEL, J.-M., 2003, Enhancing geographical information systems capabilities with multi-criteria evaluation functions. *Journal of Geographic Information and Decision Analysis*, **7**(2), pp. 47–71.

CHEMIN, Y., HONDA, K. and INES, A., 2004, Genetic algorithm for assimilating remotely sensed evapotranspiration data using a soil–water–atmosphere–plant model. In *FOSS/GRASS User Conference*, 12–14 September 2004, Bangkok, Thailand, pp. 88–92 (Canada: University of British Columbia).

CHERKASSKY, B., GOLDBERG, A. and RADZIK, T., 1996, Shortest paths algorithms: theory and experimental evaluation. *Math Programming*, **73**, pp. 129–174.

CORNE, D.W., DEB, K., FLEMING, P.J. and KNOWLES, J.D., 2003, The good of the many outweights the good of the one: evolutionary multi-objective optimization. Connections. *The Newsletter of the IEEE Neural Networks Society*, **1**(1), pp. 9–13.

CORREA, E.S., STEINER, M., FREITAS, A. and CARNIERI, C., 2000, A genetic algorithm for the P-Median problem. In *Genetic and Evolutionary Computation Conference (GECCO)*, 7–11 July 2000, San Francisco, CA, pp. 523–520 (San Francisco, CA: Morgan Kauffman).

COSTELLOE, D., MOONEY, P. and WINSTANLEY, A., 2001a, From random walks to pareto optimal paths. In *Proceedings of the 4th Irish Artificial Intelligence and Cognitive Science Conference*, 5–7 September 2001, NUI Maynooth, Ireland, pp. 309–318 (Maynooth, Ireland: NUI Maynooth).

COSTELLOE, D., MOONEY, P. and WINSTANLEY, A.C., 2001b, Multi-objective optimisation on transportion networks. In *Proceedings of 4th AGILE Conference on GIS*, 19–21 April 2001, Masaryk University Brno, Brno, Czech Republic, M. Konecny, L. Friedmannova, J. Golan and M. Kolar (Eds), pp. 523–530 (Brno, Czech Republic: Masaryk University).

DAVIES, C. and LINGRAS, P., 2003, Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks. *European Journal of Operational Research*, **144**(1), pp. 27–38.

DEB, K., 2001, *Multi-Objective Optimization using Evolutionary Algorithms* (Chichester, UK: John Wiley and Sons).

DELAVAR, M.R., SAMADZADEGAN, F. and PAHLAVANI, P., 2004, A GIS-assisted optimal urban route finding approach based on genetic algorithms. In *XXth ISPRS Congress: Geo-Imagery Bridging Continents. Commission II, Working Group II/5*, 12–23 July 2004, Istanbul, Turkey, pp. 305–308 (Enchede, The Netherlands: ITC/ISPRS).

DOZIER, G.V., MCCULLOUGH, S., HOMAIFAR, A. and MOORE, L., 1998, Multiobjective Evolutionary path planning via fuzzy tournament selection. In *IEEE International Conference on Evolutionary Computation (ICEC'98)*, 4–9 May 1998, Anchorage, Alaska, pp. 684–689 (Piscataway, NJ: IEEE Press).

EIBEN, A.E., HINTERING, R. and MICHALEWICZ, Z., 1999, Parameter control in evolutionary algorithms. *IEEE Transations on Evolutionary Computation*, **3**(2), pp. 124–141.

GOLDBERG, D., 1999, Genetic and evolutionary algorithms in the real world. *Technical Report 99013*, Department of General Engineering, University of Illnois at Urbana-Champaign, Urbana, Illnois 61801, USA.

HALLAM, C., HARRISON, K. and WARD, J., 2001, A multiobjective optimal path algorithm. *Digital Signal Processing*, **11**(2), pp. 133–143.

HUANG, B., CHEU, R. and LIEW, Y., 2004, GIS and genetic algorithms for HAZMAT route planning with security considerations. *International Journal of Geographical Information Science*, **18**, pp. 1–19.

JACOB, R., MARATHE, M.V. and NAGEL, K., 1999, A computational study of routing algorithms for realistic transportation networks. *Journal of Experimental Algorithmics*, **4**, pp. 1084–1096.

KIM, B.J., YOON, C.N., HAN, S.K. and JEONG, H., 2002, Path finding strategies in scale-free networks. *Physical Review E*, **65**, pp. 027103-1–027103-4.

KNOWLES, J. and CORNE, D., 2000, M-PAES: a memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, 16–19 July 2000, La Jolla Marriott Hotel, La Jolla, CA, USA, pp. 325–332 (Piscataway, NJ: IEEE Press).

MAGONI, D. and PANSIOT, J., 2001, Comparative study of internet-like topology generators. *Technical Report*, LSIIT Laboratory, Université Louis Pasteur, France.

MARTINS, E., PASCOAL, M.M.B. and SANTOS, J.L.E.D., 1999, Deviation algorithms for ranking shortest paths. *International Journal of Foundations of Computer Science (IJFCS)*, **10**(3), pp. 247–262.

MARTINS, E.D.Q. and DOS SANTOS, J.L.E., 1999, The labelling algorithm for multiobjective shortest paths. *Technical Report*, Departmento de Matematica, Universidade de Coimbra, Portugal.

MATTHEWS, K., CRAW, S., ELDER, S., SIBBALD, A. and MACKENZIE, I., 2000, Applying genetic algorithms to multiobjective land use planning. In *Genetic and Evolutionary Computation Conference (GECCO)*, 10–12 July 2000, Las Vegas, Nevada, USA, pp. 613–620 (San Francisco, CA: Morgan Kaufmann).

MESSAC, A. and MATTSON, C.A., 2002, Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization and Engineering*, **3**(4), pp. 431–450.

NEARCHOU, A.C., 1999, Adaptive navigation of autonomous vehicles using evolutionary algorithms. *Artificial Intelligence in Engineering*, **13**(2), pp. 159–173.

NEPAL, K. and PARK, D., 2003, Routing algorithms for transportation systems and service improvement projects in urban transportation networks. *Technical Report*, Department of Civil Engineering, Tokyo Institute of Technology, 2-12-1, Ookayama, Meguro-ku, Tokyo 152-8552, Japan.

NHPN 2003, *National Highway Planning Authority—USA, Technical Report*, Internet Document: http://www.fhwa.dot.gov/planning/nhpn/, Federal Highway Administration, United States Department of Transportation (accessed March 2006).

NOON, C. and ZHAN, F., 1996, Shortest path algorithms: An evaluation using real road networks. *Transportation Science*, **32**(1), pp. 65–73.

OKABE, T., JIN, Y. and SENDHOFF, B., 2003, A Critical survey of performance indices for multi-objective optimization. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, 8–12 December 2003, Canberra, Australia, Vol. 2, pp. 878–885 (Canberra, Australia: IEEE Press).

PEREIRA, C.M., 2004, Evolutionary multicriteria optimization in core designs: basic investigations and case study. *Annals of Nuclear Energy*, **31**, pp. 1251–1264.

PRZYBYLSKI, A., GANDIBLEUX, X. and EHRGOTT, M., 2006, The biobjective integer minimum cost flow problem-incorrectness of Sedeno–Noda and Gonzalez-Martin's algorithm, *Computers and OR*, **33**, pp. 1459–1463.

ROY, A., BANERJEE, N. and DAS, S.K., 2002, An efficient multi-objective QoS routing algorithm for real-time wireless multicasting. *Technical Report*, Center for Research in

Wireless Mobility and Networking, Department of Computer Science, University of Texas at Arlington, TX 76019-0015, USA.

SHAD, R., EBADI, H. and GHODS, M., 2003, Evaluation of route finding methods in GIS application. In *Map Asia 2003*, 13–15 October 2003, Bejing, China, pp. 12–20 (www.gisdevelopment.net/proceedings/mapasia/2003, accessed March 2006).

SKRIVER, A. and ANDERSEN, K., 2000, A label correcting approach for solving bicriterion shortest-path problems. *Computers and Operations Research*, **27**, pp. 507–524.

US CENSUS BUREAU (2003) US census 2000 tiger/line files. Internet Data Center: http://www.census.gov/geo/www/tiger/tiger2k/tgr2000.html (accessed March 2006).

VAN VELDHUIZEN, D.A. and LAMONT, G.B., 2000, On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, 16–19 July 2000, La Jolla Marriott Hotel, La Jolla, CA, USA (Piscataway, NJ: IEEE Press), pp. 204–211.

WILSON, I., WARE, M. and WARE, A., 2003, A genetic algorithm approach to cartographic map generalisation. *Computers in Industry: Special Issue: Soft Computing in Industrial Applications*, **52**(3), pp. 291–304.

XIAO, J., MICHALEWICZ, Z., ZHANG, L. and TROJANOWSKI, K., 1997, Adaptive evolutionary planner/navigator for mobile robots. *IEEE Transactions on Evolutionary Computation*, **1**(1), pp. 18–28.

XIAO, N., BENNETT, D. and ARMSTRONG, M., 2002, Using evolutionary algorithms to generate alternatives for multiobjective site-search problems. *Environment and Planning A*, **34**, pp. 639–656.

ZELENY, M., 1982, *Multiple Criteria Decision Making* (New York: McGraw-Hill).

ZHAN, F., 1998, Three fastest shortest path algorithms on real road networks: data structures and procedures. *Journal of Geographic Information and Decision Analysis*, **1**(1), pp. 70–82.

ZITZLER, E., DEB, K. and THIELE, L., 1999, Comparison of multiobjective evolutionary algorithms on test functions of different difficulty. In *Multi-criterion Optimization Using Evolutionary Methods*, K. Deb, (Ed.), pp. 121–122 (Orlando, FL: Morgan Kaufmann).

ZITZLER, E., THIELE, L., LAUMANNS, M., FONSECA, C.M. and DA FONSECA, V.G., 2003, Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, **7**(2), pp. 117–131.