

# Emergency management using geographic information systems: application to the first Romanian traveling salesman problem instance

Gloria Cerasela Crişan<sup>1</sup> · Camelia-M. Pinteă<sup>2</sup> · Vasile Palade<sup>3</sup>

Received: 27 November 2014 / Revised: 11 February 2016 / Accepted: 22 March 2016 /  
Published online: 8 April 2016  
© Springer-Verlag London 2016

**Abstract** The strategic design of logistic networks, such as roads, railways or mobile phone networks, is essential for efficiently managing emergency situations. The geographic coordinate systems could be used to produce new *traveling salesman problem (TSP)* instances with *geographic information systems (GIS)* features. The current paper introduces a recurrent framework designed for building a sequence of instances in a systematic way. The framework intends to model real-life random adverse events manifested on large areas, as massive rainfalls or the arrival of a polar front, or targeted relief supply in early stages of the response. As a proof of concept for this framework, we use the first Romanian *TSP* instance with the main human settlements, in order to derive several sequences of instances. Nowadays state-of-the-art algorithms for TSP are used to solve these instances. A branch-and-cut algorithm delivers the integer exact solutions, using substantial computing resources. An implementation of the *Lin–Kernighan* heuristic is used to rapidly find very good near-optimal integer solutions to the same instances. The *Lin–Kernighan* heuristic shows stability on the tested instances. Further work could be done to better exploit GIS features in order to efficiently tackle operations on large areas and to test the solutions delivered by other algorithms on new instances, derived using the introduced framework.

**Keywords** Emergency management · Traveling salesman problem · GIS · Geographic coordinates

---

✉ Vasile Palade  
vasile.palade@coventry.ac.uk

<sup>1</sup> Faculty of Sciences, Vasile Alecsandri University of Bacău, Bacău, Romania

<sup>2</sup> Faculty of Sciences, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

<sup>3</sup> Faculty of Engineering and Computing, Coventry University, Coventry, UK

## 1 Introduction

Emergency management is a continuous process, aiming at preventing anthropogenic and natural disturbances, designing plans to mitigate their impact, efficiently responding when disasters strike, and recovering to normal state.

Communication procedures are essential in case of disaster. Their proper design could reduce the natural effects and the social costs of an adverse event: clear rules and responsibilities in emergency plans lead to correct assessment and rapid response. For example, the Final Report on the Accident at the Fukushima Nuclear Power Stations describes that the government authorities "...fell into a state of confusion at various stages and consequently failed to respond to problems timely or adequately" [38].

Communication infrastructure is equally important in hazards. A cooperative emergency communication system must be quickly activated and should effectively work until normal communications are re-established. The same report specifies that "...the Off-site Center, which should have taken a primary role in accident responses, could not function fully due to failure of communication means and other reasons" [38]. This paper is a contribution toward the proper design of communication infrastructure, in order to efficiently connect specific locations, distributed on a large territory.

When designing communication networks in a specific area, many geographical, ecological and socioeconomic characteristics need to be considered. For example, mobile communications require cost-effective ubiquitous coverage for addressing challenges toward 5G standards. The mobile network infrastructure has to be economically designed: the deployment, usage and development costs have equal decision weights as the quality of the services. Some of the main phases of a communication network design are to model in an accurate way the in situ characteristics and to simulate the normal and abnormal working conditions.

Efficient emergency response is crucial during the first hours. If the communication network is operational, people can signal their presence and an accurate intervention map can be quickly drawn by the rescuers. In dramatic cases, when some persons are physically unable to use their working phones, the mobile network providers could cooperate and offer the exact locations. When the GPS service is not available, it is possible to use Wireless Application Protocol with the Google Maps API [84]. When the cell phone signals are unreachable, it is possible to use the active ad hoc or multihop networks [6].

If the roads are not operational, then unmanned aerial vehicles (UAVs) can discover people in need and helicopters can be used for assisting and transporting them. Such tours for observation drones, and rescue helicopters or boats must be quickly and efficiently designed. They do not need roads, and they directly reach the points where they are sent. Researchers constructed platforms that can control groups of commercially available UAVs able to independently and dynamically adjust their on-site behavior [64].

In order to efficiently use the software platforms, it is suitable to choose techniques that build syntactically correct finite automata models of the component system [16,24]. Recent methodologies as in [56] could be used to evaluate the quality of the software.

An academic problem that successfully addresses such real-life situations is the *traveling salesman problem (TSP)*. In its standard formulation, it seeks for the shortest tour connecting all the cities on a map [46,65]. Currently, there are many *TSP* variants, modeling real-life situations with high economic and social costs: bottleneck *TSP*, prize collecting *TSP*, multiple *TSP*, generalized *TSP*, etc. [35,61].

Approaching emergency management through *TSP* is our choice for this paper, motivated by both the above-mentioned reasons and the previous. Other approaches focus on facility location problems (i.e., where to put the rescue center to optimally cover a region) [8,76], or

on resource allocation problems (how to optimally dispatch the aids from the rescue centers) [26], or both types [49].

The current work introduces a *Romanian TSP* instance based on *geographic information systems (GIS)* as the first national *GIS-TSP* instance. Two types of situations when several cities are excluded are modeled starting from this instance. A novel framework, designed to model a dynamic situation using a recursive sequence of instances, is also introduced. It offers a general and flexible method for systematically derive scenarios starting from a given situation. It generalizes the methods presented in many previous works, but here we focus on its importance for the emergency preparedness: multiple scenarios could be treated and adequate responses could be set up. The framework is illustrated starting from the novel instance; several best solutions for the derived instances are represented as maps based on *GIS* services.

The idea of bridging academic optimization problems with crisis management is approached by many researchers, for example in [43,71]. The adaptive governance intends to mitigate the effects of climate change through low-regret adaptive strategies in planning. For example, the European Commission promotes strategic environmental assessment for large-scale trans-boundary projects, as long as changing hazard patterns and changing societal circumstances are new challenges [23]. Our paper is a step toward this goal, as it connects academic Operational Research with modern features of *GIS* services and environmental sciences.

The paper is structured as follows. In Sect. 2, the algorithms used for the introduced instances are described; several ways of using *GIS* services in *Operational Research* domain are also shown. Section 3 introduces the first national *GIS-TSP* instance and the novel framework for recursive modification of an instance. Numerical experiments and discussions about their results are detailed in Sect. 4. Several conclusions and research directions are presented in Sect. 5 of the paper.

## 2 Preliminaries

### 2.1 Literature preview

To optimally solve a difficult problem is not an easy task. For example, for the NP-hard class of problems it is unlikely that a solving algorithm with polynomial time for worst cases will ever be found. This is why these difficult optimization problems, including traveling salesman problem, are approached using exact, approximate or heuristic methods.

The branch-and-bound algorithm is used for global optimization for non-convex problems. It was first proposed in [45]. It is interesting that the name of the method first occurred in connection with the TSP [51]. Its basic idea is to maintain a lower and upper bound for the optimal solution and to exit when the distance between them becomes insignificant. At each step, an efficient and problem-depending method chooses how the solution space is split, decomposes the current problem into several (usually two) problems, computes the corresponding bounds for them and updates the global bounds [47]. As the solution space for each subproblem becomes smaller, the local bounds are expected to become sharper, and so the gap between the global bounds narrows. Sometimes, the current subproblem can be eliminated, with consequences in improving the execution time. If its lower bound is greater than the current upper bound, obviously its solution space does not contain an optimal solution.

The cutting plane technique is dedicated to solve mixed integer programming problems using a repeated sequence of relaxations [30]. Once such fast methods for linear programming (as simplex) exist, the first step of the cutting plane approach is to consider a relaxation of the initial problem, by removing the constraint that the variables have to take integer values. If the solution to this problem has integer values for the relaxed variables, then the algorithm stops. If not, a non-integer component of the solution is chosen; a hyperplane (a Gomory cut) is defined and added to the previous relaxation. The cut detaches the solution (as it is not a solution to the initial problem), but maintains all feasible integer solutions. The algorithm iterates by solving the new tightened relaxed problem.

The cutting plane method was considered impractical, so it was integrated into branch-and-bound implementations with very good practical results [60]. The new algorithm, named branch and cut, is currently one of the best exact methods for solving many COPs [54]. If the problem is large or hard and cannot be solved to the optimality, the branch-and-cut approach can supply very useful lower and upper bounds on the optimal value. The current computing systems and the parallel programming languages allow successful branch-and-cut implementations for a broad range of complex problems. An example of such demanding problem is treated in [1]. Very effective branch-and-cut approaches are designed for specific problems; for example, symmetric TSP is successfully tackled in [59] by using strong cutting planes from the polyhedral theory. Since 2002 IBM ILOG CPLEX is implementing the concurrent optimization, which allows simultaneously solving a problem by means of multiple algorithms. The version CLPEX 11 includes parallel optimization as a general feature [37].

*Traveling salesman problem (TSP)* can be formulated in terms of graph theory as a minimization problem on a complete weighted graph as in [59]: given a complete graph  $G$  with vertex set  $V = \{1, 2, \dots, n\}$ , and  $\{d_{ij}\}$ ,  $1 \leq i, j \leq n$  its weights on edges, find a least weighted cycle that visits each vertex exactly once.

As exact methods are sometimes computationally expensive for difficult problems, the research community continuously designs and tests heuristic solving methods [13,62,73]. Among constructive heuristics for TSP, we mention successful applications of genetic algorithms [63], tabu search [25], and ant colony optimization [21]. Several local search heuristics for TSP involve sub-tour swaps: *2-opt* or *3-opt* work on a solution and try to improve it by deleting two or three edges and reconstructing the tour in the best way.

One of the best local search heuristics for symmetric TSP is *Lin-Kernighan*; it generalizes the classic *2-opt* and *3-opt* moves as the  $\lambda$ -*opt* move, by considering  $\lambda$  edges from a tour and replacing them with other  $\lambda$  edges such that the total length is shorter. The *Lin-Kernighan* method starts with repeating *2-opt* moves until no improvement is possible, and continues with ascending values of  $\lambda$  in a dynamic way: it decides at each step if it should consider a bigger  $\lambda$  or it should stop. Very recent scientific applications of the method are on modeling wireless sensor networks [3] and on multirobot rendezvous for recharging in persistent tasks [53].

Uncertainty of the real systems and the dynamism of life added new characteristics and compounded the difficulty of the TSP. Today there are many demanding versions such as Probabilistic TSP [39], Fuzzy TSP [14], Robust TSP [55], Online TSP [7] and Time-dependent TSP [52].

Comprehensive information on TSP history, datasets and currently one of the best solvers [10] is maintained on a specialized Web site [12]. Very good books including also the state of the art of TSP computation are [4,11]. The related software available online at Czyzyk et al. [19] is used in Sect. 4 to illustrate the proposed framework on the new GIS-TSP national data.

## 2.2 GIS services in operational research

The new features of GIS tend nowadays to merge with the academic Operational Research domain. Several GIS commercial packages use heuristic approaches for TSP. For example, Route solver from ArcGIS is proprietary software that has been researched and developed extensively to quickly yield good results [5]. The documentation of Esri ArcGIS 10.1 shows that the solver uses Dijkstra's algorithm to find shortest paths. Several heuristics for managing spatial data are discussed in [83]. Comparisons between available applications for route planning are made in [18].

A successful real application for traffic management in case of disaster is the contraflow evacuation on freeways during Hurricane Katrina, which was completed in half of the planned time. Integration of spatiotemporal data into *database management systems (DBMSs)* is at its beginnings. *TerraLib* is an open-source package [75] for large-scale environmental and economic applications. The *National Atlas of the USA* includes GIS and geo-statistic datasets under a Public Domain license [57]. Other spatial DBMSs are: *Oracle Spatial*, *MySQL*, *PostGIS*.

Early studies show the efficiency of logistic approaches in emergency transportation for disaster relief in Somalia [41]. Current works explicitly model the disaster relief tour planning as TSP [42], or show connections with TSP variants, for example with *multiple TSP* in [82]. From the opposite side, there are many attempts to include GIS services into the academic research. Modern *global navigation satellite systems (GNSSs)* like GPS, Galileo, GLONASS or BeiDou offer accurate location tracking. Broad integration of GNSS into consumer products allows collecting cheap and accessible data for academic use.

For example, in [58] it is presented a Bayesian approach for the assessment of digital risk probability in Romania. The authors consider in [67] a benchmark of similar instances from central Spain and study the influence of the asymmetry on the quality of the solutions. Several bus routes from Zagazig, Sharqia Governorate, Egypt, were approached in [22].

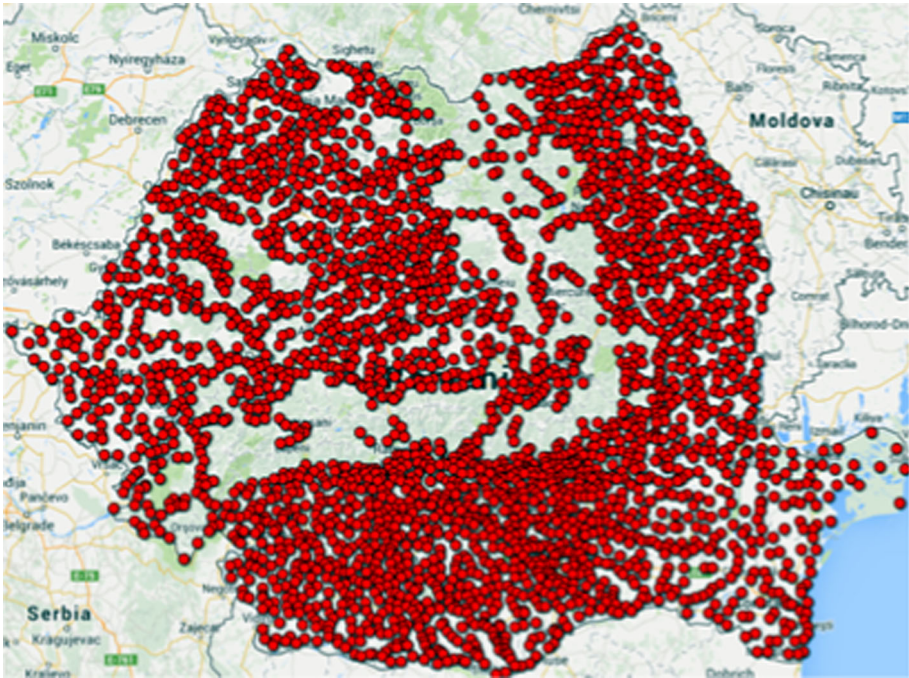
## 3 Proposed instances and a new framework for TSP

The current section includes the main novelties of the paper. A new TSP instance based on the major settlements of Romania is introduced. It follows a novel framework to create a sequence of successive instances used to illustrate an emergency management process.

### 3.1 The first national GIS-TSP instance

As today the GPS coordinates allow intensive computations, accurate positioning and adjustable maps, the data from Crăciunescu [15] were used to build the first *Romanian TSP instance* with orthodromic (great circle) distances. We meant to model one large territory TSP instance in the most correct way, and we decided to use the geodesic distances instead of the Euclidean distances. Realistic data lead to relevant characteristics and complex attributes; for example, car engine emissions could be considered as in [80].

To our knowledge, this is the first national TSP instance with nodes defined by the geographic coordinates of localities, and great circle distances. All the national instances from traveling salesman problem [78] specified through geographic coordinates use the EUC\_2D metric. This is the main contribution of this instance: spreading on a large territory, having more than 238,000 km<sup>2</sup>, it is the first that accurately reflects the real distances between its nodes. The need for this specific instances is driven by commercial or open-source pack-



**Fig. 1** The new TSP *romania2950.tsp* Romanian instance based on geographic coordinates

ages [28,31,70]. This instance is used to illustrate the framework introduced in the next sub-section.

From more than 13,000 Romanian human localities, we selected 2950 with administrative autonomy [68]. Their geographic coordinates (pairs of decimal values for latitude and longitude) were transformed to comply with *TSPLIB* format, *GEO* type [79]. The first *GIS-TSP* national instance *romania2950.tsp* was set up and is represented in Fig. 1.

This new *TSP* instance integrates the geographic coordinates and can be used in all modern applications based on *GIS* services. For example, it is possible to study the temporal evolution of a large territory using geoinformatics systems and packages. Satellites data could be compared with data collected from the territorial sensors. Ecological, geographical, meteorological or environmental characteristics could be integrated in such modern *GIS-TSP* instances, allowing complex models and accurate predictions at national scale.

The new *GEOM* type introduced for the *World TSP* instance [77] offers a computational shortcut as it directly handles the geographic coordinates. It allowed us to define the equivalent Romanian TSP instance with geographic coordinates in *GEOM* norm. The *GEOM* norm is specifically designed to provide the distances in meters, while the *GEO* norm uses integer kilometers.

As there are large regions with high node density, we chose to further work with the *GEOM* version, for higher accuracy.

Both the Romanian instances have the same map (as in Fig. 1), only their solutions differ and, of course, their visual representations as well. The *Romanian TSP* link from the *TSP* main page [78] provides the detailed page with instances and results; both instances are available online for download at [69]. There are other attempts in defining TSP instances with

geographic coordinates, but none has the real settlements from a country and the orthodromic distance.

For example, the national instances from traveling salesman problem [78] use the EUC\_2D norm. In [67] the instances are produced by randomly plotting points on the map of Spain. In [74] random instances on a sphere are used. The researchers' broad interest on such instances led to a new TSP variant: *Spherical TSP* or *TSP with spherical coordinates*.

### 3.2 Introducing a new framework for TSP

The main TSP data collections currently available [77,79] consist of sets of independent instances, used to assess the overall performance of a specific solver. But in real life, there are situations when the behavior of a solver on close or dependent instances is important. An example is the need for stable solvers: if two problems are close (in a specific way), then their solutions are expected to be close.

In an industrial process based on stable applications, the technological flow and the production activities for several initial products are designed by computing their exact optimum solutions.

Later on, slightly modified products can be designed using the same technological solutions: the results are expected to have (almost) the same quality. Emergency management considers also interrelated situations: one geographic area with rapid deteriorating conditions can be modeled by a sequence of pairs of close instances.

To construct a sequence of test instances, a new framework called *ALTER-Framework for TSP (ALTER-FTSP)* is proposed. It is further exemplified from the first national *Romanian GIS-TSP* instance already introduced.

The framework *ALTER-FTSP* uses a method *ALTER* to construct instances based on the recurrence relation (1) where *instance(0)* is an initial instance, *parameters* designates a set of (variable or not) parameters, and  $i > 0$ .

$$instance(i) = ALTER(instance(i - 1), parameters(i - 1)). \quad (1)$$

The researchers appropriately choose the function *ALTER* and the parameters. The latter could be dynamically adjusted to tailor the characteristics of specific instances. This framework is general and flexible. It builds a sequence of interrelated instances, controlled by the function *ALTER*, the initial instance and some dynamic parameters. The sequence could be used for diverse investigations. A solver could be tested for behavioral patterns, continuity for example, using these sequences. At theoretical level, Eq. (1) allows mathematical investigations, for example convergence or structural stability theorems. From the viewpoint of algorithmic theory, smoothed analysis [72] can be done.

The hazard vulnerability and the effectiveness of the response are highly influenced by representative and broad simulations. An example of strategic application of the framework is to include it as a module for generating training situations for spatial decision support systems. A complex control module could replace the basic control parameter set *parameters(i)*. Another example is designing stress tests for the software packages: a battery of complex scenarios can be obtained using this framework.

In operational situations, certified behavioral patterns could allow solution re-usage in case of small modifications. This includes the case when local conditions change due to unexpected events. For example, the 2003 Northeast power outage from USA and Canada was triggered by successive power plants shutdown, dropping in minutes the load with more than 80 %. The lack of efficient communication and instant network balance overloaded the energy providers, forcing their automatic disconnection from the grid. Shortly afterward, the

North American power grid lost many nodes. This type of cascade events is suitable for our framework.

The general construction we introduce here brings together many experiments from previous works and correlates with many other methodologies. It is general enough to merge the methods of producing TSP instances starting from an initial set. In [27], new instances are produced only in one step, using two operators: *reduction* and *shake*. In [2], a *Lindenmayer system* is used to construct in one step a new instance by moving a fixed number of nodes in their neighborhood. The same idea of perturbation is presented in [9].

In general, this framework can deliver a sequence of dependent instances for modeling dynamic TSP. In [40], a dynamic TSP with continuous functions for the distances is modeled through a sequence of classic TSP instances with constant distances on all edges.

In [48], a dynamic elastic operator is introduced for three expressions of dynamism: a node appears, a node disappears or a node moves. An alternative to our approach is to consider evolving algorithms that constantly reflect the changes. An example is the ant colony algorithms used in [34].

Other methods for generating instances either produce independent ones, or deliver instances depending on more than one instance. For example, in [66] the authors use a modification of the `portmgen` generator from [29] for producing random independent instances with 300, 500 and 700 nodes. In [81], the traditional genetic operators *crossover* and *mutation* are used to produce new instances starting from a pool of random instances with 100 nodes. So, the new instance depends on two old ones. The framework we introduce here uses just one instance to derive the next one.

The idea of slight modification is extensively used in optimization. For example, *simulated annealing* and *tabu search* exceptionally accept worse solutions in order to escape from local optimum. Local search is a related paradigm, as sometimes good solutions are close in the search space, forming promising regions. The same idea applied to a different problem is presented in [17]. The authors evaluate the impact of deleting nodes on the network vulnerability. The extreme case is treated in [44]: the most effective way to disrupt a network is investigated.

## 4 Numerical experiments and discussion

To illustrate the *ALTER-FTSP* framework defined in Sect. 3.2, we present here two cases. The former models the uncertainty by randomly isolating some localities and dropping them from the instance *romania2950.tsp*. The latter case deterministically excludes the least populous localities. The first case simulates the effect of an adverse event that randomly hits, or multiple errors appeared in a technological network; the second is appropriate in the early recovery phase, when relief dispatching is concentrated in most populous places, or is suited for modeling coordinated, concurrent attacks on networks.<sup>1</sup>

### 4.1 Uncertainty modeling: case 1

The function *ALTER* randomly selects and deletes nodes from the current instance and returns a new instance. The process is repeated nine times, so a sequence of nine new *TSP* instances with fewer nodes is generated. The general Eq. (1) becomes Eq. (2) with  $i > 0$ .

$$instance(i) = Delete\_at\_Random(instance(i - 1), p), \quad (2)$$

<sup>1</sup> The instances are available by request to the authors.



**Table 1** The solutions of the *romania2950* instance with the geographical norm in meters (GEOM); the results are for the branch-and-cut algorithm and the Lin–Kernighan heuristic

GEOM	Branch and cut	Lin–Kernighan	
TSP	Optimal	AVG	MIN
Solution (m)	20,338,171	20,375,727.5	20,362,640
RPD (%)	–	0.1847	0.1203
Time (s)	2852	3.25	2.53

where the function *Delete\_at\_Random* takes a TSP instance and just one constant parameter  $p$  that specifies, as a percentage value, how many nodes are randomly deleted from the input instance and outputs the new instance. We run three such simulations with constant  $p \in \{1\%, 3\%, 5\%\}$ .

*ALTER-FTSP* can produce a more complex control on the sequence of instances. For example, the unique parameter  $p$  could be a variable, modeling a powerful event that decreases in time: seismic sea waves, floods or nuclear accidents. An increasing series of values for  $p$  can model cascade power outages or consecutive earthquakes hitting the same region. The corresponding equation in this case is Eq. (3) with  $i > 0$ .

$$instance(i) = Delete\_at\_Random(instance(i - 1), p(i - 1)). \tag{3}$$

A more general illustration of *ALTER-FTSP* is based on multiple parameters. Classifications can serve as a supplementary selection criterion. For example, Eq. (4) with  $i > 0$ .

$$instance(i) = Delete\_at\_Random(instance(i - 1), p(i - 1), q(i - 1)), \tag{4}$$

where  $p(i)$  is the percentage value of the excluded nodes and  $q(i)$  is the minimum distance between them. This could be a request for a balanced territorial relief supply.

In our case, driven by Eq. (2), the exact solutions are obtained using Concorde, a branch-and-cut solver [10], described in [33]. It uses a cutting plan method, iteratively solving linear programming relaxations of the initial instance. We chose the online run, accessible through NEOS server [19,20]. The Concorde solver ran for at most four hours on two Intel Xeon processors at 2.2GHz and 64 GB RAM; implicit attributes are used there: CPLEX and fixed seed. The exact solutions were used to assess the results provided by the heuristic method. The *Lin–Kernighan* [50] method was chosen as it is one of the most effective heuristic techniques designed for symmetric TSP. The heuristic application was run ten times, and we report here the average and the best values.

Table 1 shows the solutions for the initial instance, with integer distances in meters (GEOM norm). The statistic relative percentage difference (RPD) is computed using the formula:

$$RPD = \frac{Solution - Best}{Best} \%$$

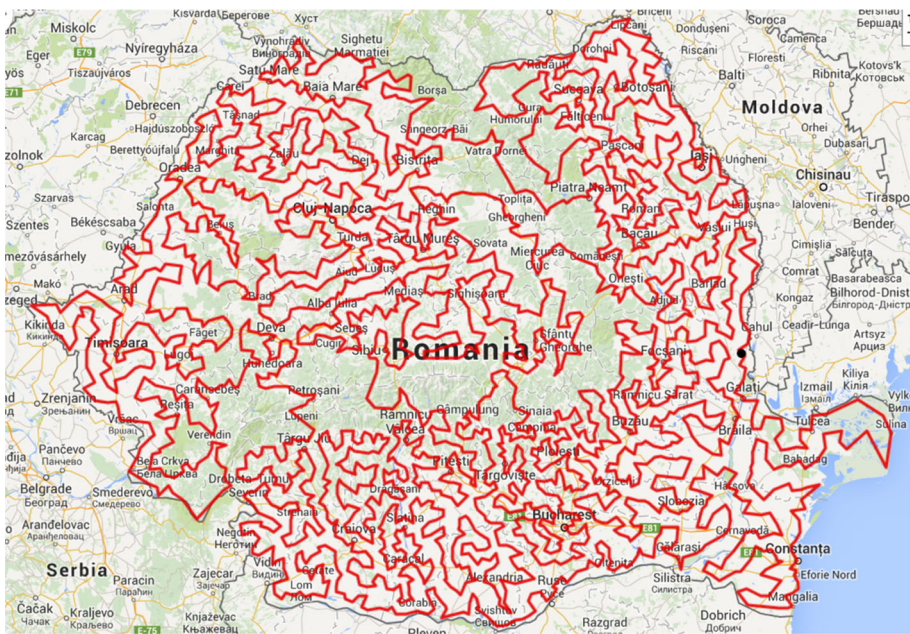
where *Solution* is solution provided by the heuristic method and *Best* is the exact solution. For representing the instances and the solutions, GPS Visualizer, a popular, highly customized and free online tool [32] is used.

Table 2 shows the exact solutions and running time in seconds for first nine iterations based on Eq. (2), using the function *Delete\_at\_Random* and constant  $p \in \{1\%, 3\%, 5\%\}$ . The dimension of the current instance, the optimum value (in m) and the running time (in s) are mentioned.

Figure 2 shows an exact solution to *romania2950.tsp* with GEOM norm. Figure 3 illustrates the repeated transformations of the initial instance, showing the nine sets of excluded nodes

**Table 2** Optimum solutions (in m) and running time (in s) for the introduced instances using branch and cut

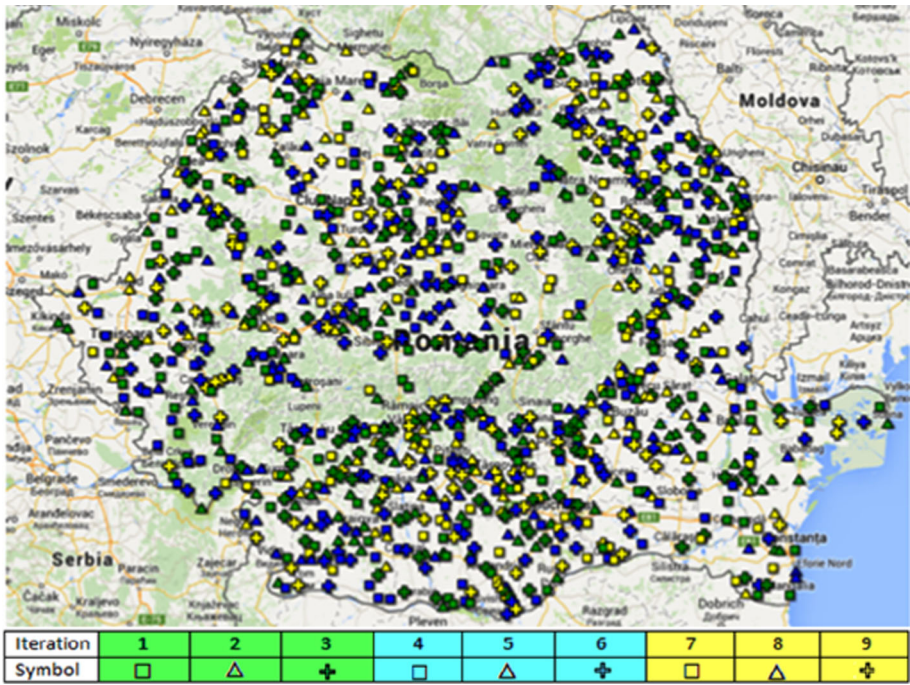
Iter	1 %			3 %			5 %		
	Inst	Solution	Time	Inst	Solution	Time	Inst	Solution	Time
1	2921	20,205,928	1305	2862	19,966,083	3619	2803	19,730,203	9115
2	2892	20,095,059	2350	2777	19,627,256	2169	2663	19,188,050	991
3	2864	19,966,026	10,322	2694	19,236,971	3487	2530	18,661,985	321
4	2836	19,865,204	834	2614	18,885,027	2944	2404	18,163,059	2034
5	2808	19,742,128	326	2536	18,594,278	7196	2284	17,623,115	1686
6	2780	19,606,514	1162	2460	18,255,359	5933	2170	17,096,230	9303
7	2753	19,498,877	1022	2387	17,957,873	1152	2062	16,629,320	3909
8	2726	19,399,992	2751	2316	17,626,444	4710	1959	16,153,456	5358
9	2699	19,276,005	820	2247	17,338,055	526	1862	15,708,029	2904



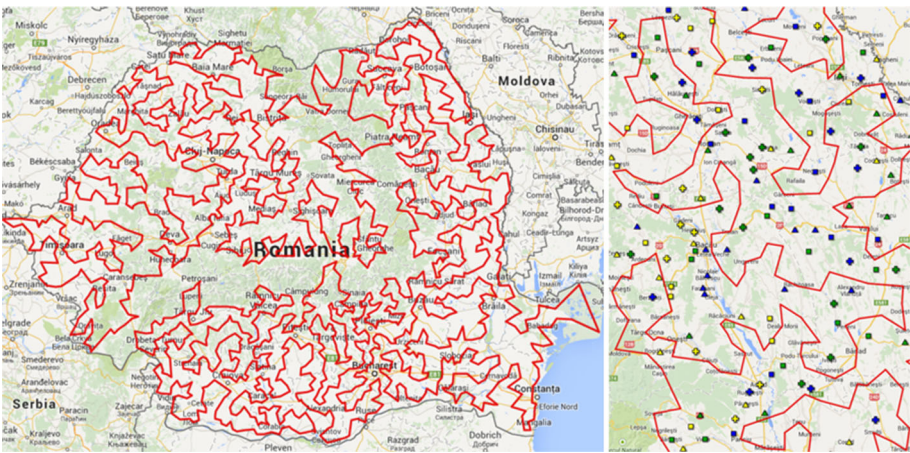
**Fig. 2** An optimum solution to *romania2950.tsp*

from the instance *romania2950.tsp* and their visual characteristics when  $p = 5\%$ . For example, the excluded nodes at the first iteration are represented by green squares. The instance resulted at the ninth iteration, with only 1862 nodes, has the optimum path depicted in Fig. 4 (left). When compared with Fig. 2, the path clearly connects fewer nodes, uniformly distributed on the territory.

Figure 4 (right) is a close capture of Eastern Romania, with the excluded nodes and the optimum path connecting the remaining ones. All the sequences manifest irregular solving time for *branch and cut*: there is no decreasing tendency, and no continuity is manifested for consecutive instances. For example, the hardest instance with 2864 nodes is solved in 10,322 s. The previous instance is solved in 3250 s, and the next instance is very quickly



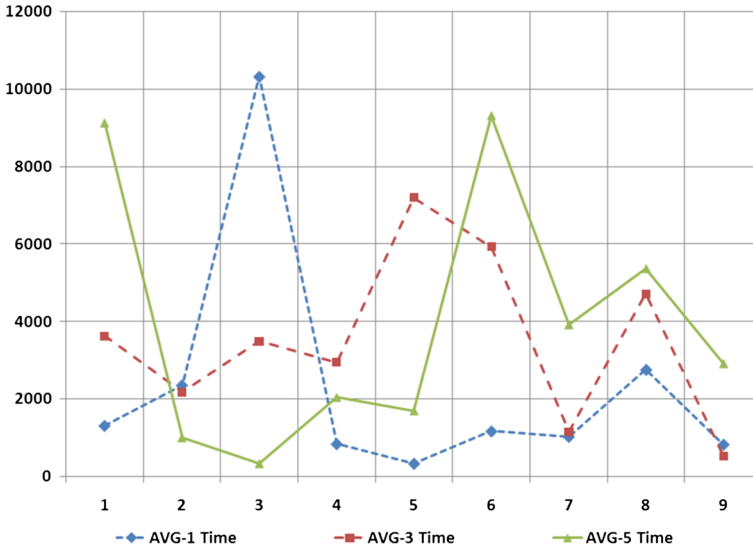
**Fig. 3** The excluded nodes from *romania2950.tsp*, in all iterations, with their symbolic representation, for  $p = 5\%$



**Fig. 4** One optimum solution to *romania1862.tsp* (left) and a close capture with solution and excluded nodes for  $p = 5\%$  (right)

solved in 834 seconds. We can state that the running time for a slightly modified instance cannot be predicted (Fig. 5).

The results provided by the chained Lin–Kernighan heuristic method for the three sequences of instances are given in Tables 3, 4 and 5. The running time values are less than four seconds (Fig. 6), proving that the chosen heuristic method is constantly extremely fast.



**Fig. 5** The average execution time with the *branch-and-cut* algorithm on the nine corresponding instances

**Table 3** The results of Lin–Kernighan heuristic after ten runs for the nine considered instances; the percentage of extracted nodes is 1 %

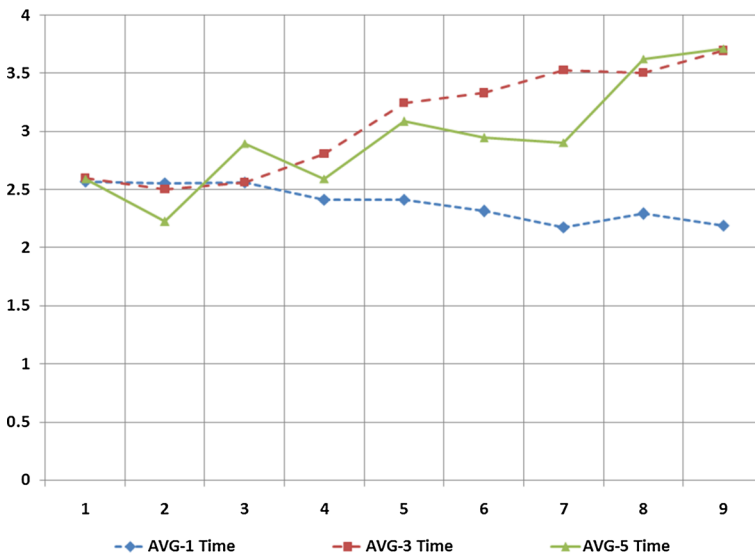
Iter	Inst	AVG		MIN	
		Solution (m)	RPD (%)	Solution (m)	RPD (%)
1	2921	20,246,706.7	0.2018	20,230,043	0.1193
2	2892	20,128,245.7	0.1651	20,110,750	0.0781
3	2864	20,007,995.9	0.2102	19,990,449	0.1223
4	2836	19,909,053.9	0.2207	19,891,244	0.1311
5	2808	19,788,311.4	0.2339	19,774,075	0.1618
6	2780	19,644,172.8	0.1921	19,635,096	0.1458
7	2753	19,527,522.4	0.1469	19,514,930	0.0823
8	2726	19,438,145.2	0.1967	19,420,511	0.1058
9	2699	19,308,946.7	0.1709	19,293,860	0.0926

**Table 4** The results of Lin–Kernighan heuristic after ten runs for the nine considered instances; the percentage of extracted nodes is 3 %

Iter	Inst	AVG		MIN	
		Solution (m)	RPD (%)	Solution (m)	RPD (%)
1	2862	19,992,830.0	0.1340	19,981,062	0.0750
2	2777	19,664,766.8	0.1911	19,648,316	0.1073
3	2694	19,263,934.7	0.1402	19,252,645	0.0815
4	2614	18,915,476.1	0.1612	18,899,831	0.0784
5	2536	18,620,313.8	0.1400	18,606,727	0.0670
6	2460	18,284,351.9	0.1588	18,278,235	0.1253
7	2387	18,002,802.1	0.2502	17,989,492	0.1761
8	2316	17,655,558.4	0.1652	17,637,597	0.0633
9	2247	17,365,411.0	0.1578	17,353,093	0.0867

**Table 5** The results of Lin–Kernighan heuristic after ten runs for the nine considered instances; the percentage of extracted nodes is 5 %

Iter	Inst	AVG		MIN	
		Solution (m)	RPD (%)	Solution (m)	RPD (%)
1	2803	19,730,203	0.1529	19,749,199	0.0963
2	2663	19,188,050	0.1726	19,210,950	0.1193
3	2530	18,661,985	0.1406	18,670,752	0.0470
4	2404	18,163,059	0.1545	18,182,716	0.1082
5	2284	17,623,115	0.2081	17,645,467	0.1268
6	2170	17,096,230	0.1299	17,100,416	0.0245
7	2062	16,629,320	0.2505	16,657,541	0.1697
8	1959	16,153,456	0.1376	16,166,862	0.0830
9	1862	15,708,029	0.1693	15,725,481	0.1111

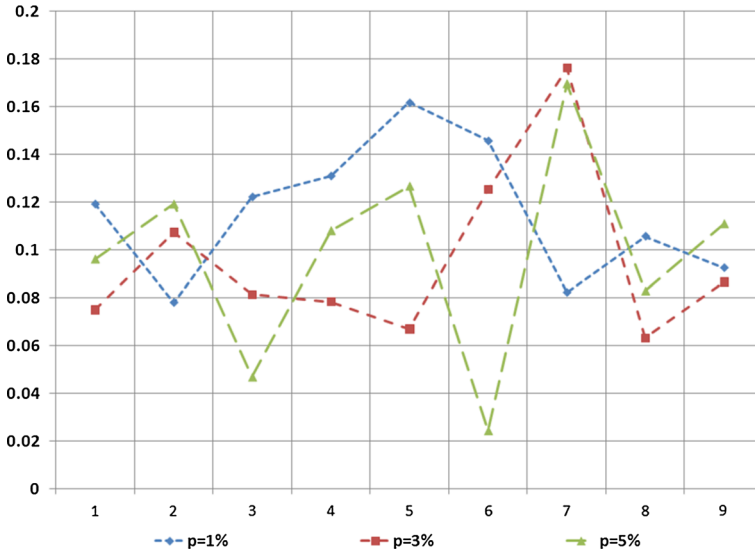


**Fig. 6** The average execution time with the Lin–Kernighan heuristic on the nine corresponding instances

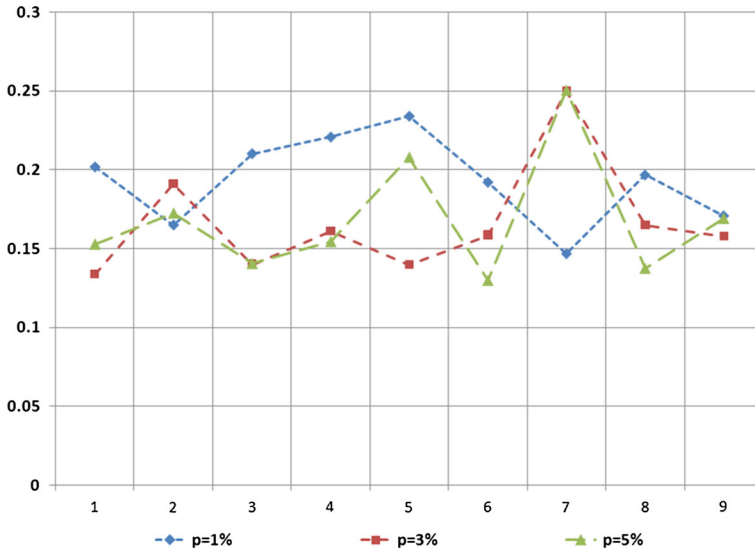
Figures 7 and 8 illustrate graphical comparisons between statistic RPD (%) minimal solutions and the average solutions of Lin–Kernighan heuristic for the three considered instances derived from *romania2950.tsp*. These results show that the average values correlate with the minimum values, and thus the Lin–Kernighan method proves to be stable on the studied dependent instances.

### 4.2 Deterministic intervention: case 2

To illustrate the deterministic control, we imagine below a situation that leads to a complete data-driven node exclusion. We model a targeted network emergency situation on a region with four neighbor counties from Eastern Romania.



**Fig. 7** A graphic comparison between statistic RPD (%) minimal solutions for the *Lin-Kernighan* heuristic on all nine corresponding instances



**Fig. 8** A graphic comparison between statistic RPD (%) average solutions for the *Lin-Kernighan* heuristic on all nine corresponding instances

The considered region are Bacău, Iaşi, Neamţ and Vaslui, marked in Fig. 9. To derive the first set of instances is used Eq. (5) with  $i > 0$ .

$$instance(i) = Delete(instance(i - 1), p, pop, asc, area), \tag{5}$$



**Fig. 9** The area of targeted network emergency situation. Map from Wikipedia

**Table 6** The exact solutions and the time for the nine considered instances in *targeted*, *regional* and *national* exclusion; the percentage of extracted nodes is 1 %

Iter	Inst	Targeted		Regional		National	
		Solution	Time	Solution	Time	Solution	Time
1	2921	20,208,799	3156	20,192,113	1901	20,192,113	1901
2	2892	20,086,682	5559	20,074,178	2410	20,013,775	2632
3	2864	19,971,516	11,629	19,944,305	2374	19,893,760	1671
4	2836	19,833,281	4220	19,836,523	11405	19,720,933	1307
5	2808	19,720,866	6029	19,702,054	3828	19,589,767	707
6	2780	19,562,158	1700	19,568,888	2762	19,484,719	728
7	2753	19,400,909	7944	19,457,320	10081	19,295,968	2187
8	2726	19,194,571	3878	19,310,201	1868	19,185,688	10,645
9	2699	19,054,069	3525	19,201,192	3945	19,056,895	10,736

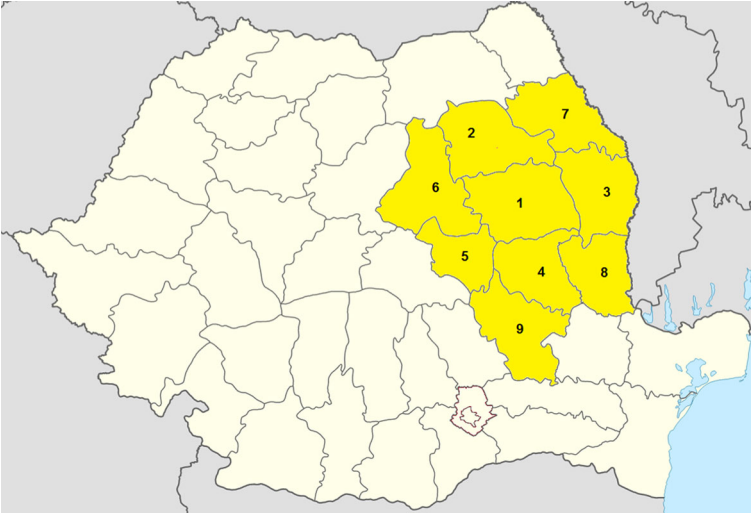
where  $p$  is 1 %, and the function *Delete* excludes the  $p$  % least populous localities from the *area* (it uses the values from the column *population* and deletes the nodes in *ascending* order).

We observe that *ALTER-TSP* uses a function with four constant parameters. The exact solution and the Concorde running time are illustrated in the column *Targeted* from Table 6.

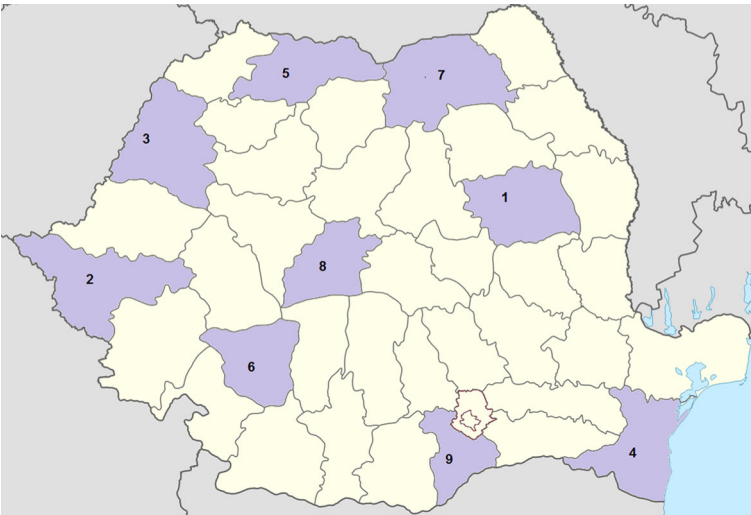
In order to introduce a variable parameter, it is used Eq. (6) with  $i > 0$ .

$$instance(i) = Delete(instance(i - 1), p, pop, asc, area(i - 1)). \quad (6)$$

We made also two more instantiations for our framework: It is designed a regional network attack (from each county shown in Fig. 10, in the specified order, the least populous localities were excluded) and a national network attack (the counties and their order for excluding nodes are presented in Fig. 11).



**Fig. 10** The area of regional network emergency situation. Map from Wikipedia

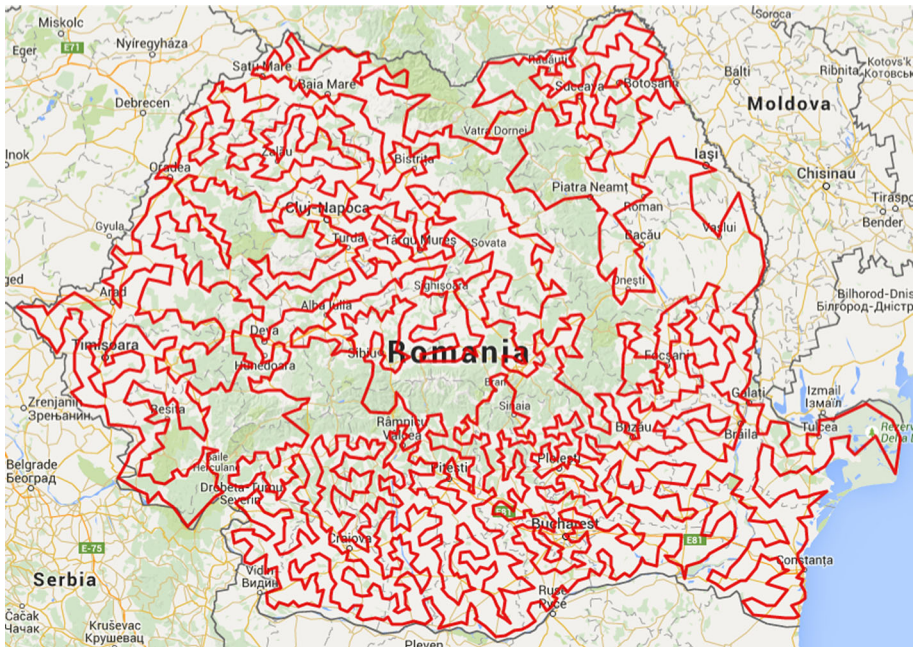


**Fig. 11** The area of national network emergency situation. Map from Wikipedia

The exact lengths and the Concorde running time are presented in the columns *Regional* and *National* from Table 6.

The exact solution to the instance with 2699 nodes from the targeted simulation is presented in Fig. 12. The results from Table 6 show again that the exact method we chose does not manifest a predictable execution time. Although we have derived a small number of instances, this result seems to show that the behavior of Concorde on random uniform instances presented in [36] is unlikely to be generalized.





**Fig. 12** Exact solution for the instance with 2699 nodes; *targeted* network emergency situation

## 5 Conclusion

This paper has two main contributions. Firstly, it introduces *romania2950.tsp*, the first national GIS-TSP instance with 2950 nodes. The distance between cities is computed using their geographic coordinates, as orthodromic distance. The new instance was recursively modified in order to support an emergency management process, and the solutions were compared with the optimum solutions.

Secondly, it introduces a framework to sequentially build instances, starting from an initial one, in order to evaluate a user-defined software metric; here we consider the continuity of the quality of the results delivered by the *Lin–Kernighan* method, when compared to the exact solutions. Our empirical investigation shows that the *Lin–Kernighan* heuristic is stable on the studied instances. The framework is general and flexible, as it can model many real situations.

The aim of this paper is to develop new methods for properly assessing the risks in large territory, complex networks. For reaching this goal, the two main contributions of this paper (the Romanian GIS-TSP instance and the ALTER-FTSP structure) interfere: the former is the real-word support for the latter. As the real geographic world works in *Space + Time* dimensions, the instance is a composite network of spatial points, and the framework is a flexible and general description of the network evolution. Thus, they complement each other.

A scenario including uncertainty is presented. This model could be used in transportation safety: the randomly broken sensors from a large territory road network have to be inspected after an earthquake by a swarm of drones. Or, during preparedness phase, lots of virtual, randomly generated critical situations could be imagined, and the responses to them could be assessed.

The deterministic scenario presented includes a priority system. It is useful for modeling; for example, top-priority helicopter visits with doctors and supplies to schools and hospitals.

Satellites like *Suomi NPP* transmit global, accurate data on atmospheric, terrestrial and oceanic conditions, helping the forecasters to make early incident announcements. The environmental hazards management systems need to be prepared for cascade severe conditions, and therefore running systematic, theoretic sound simulations and tests would definitely improve the quality of the preparedness. For example, Giovanni Bisignani, IATAs Director General and CEO said in a press release after the 2010 eruptions of Eyjafjallajökull Icelandic volcano: “Governments must place greater urgency and focus on how and when we can safely re-open Europe’s skies. This means decisions based on risk-management, facts and utilizing operational procedures that maintain safety.”

More work need to be done in order to test the method on other longer sequences and also to test other solving methods. For example, a stable *TSP* solver could be successfully integrated in touristic GIS applications, as people tend to visit almost the same points of interest. As a generalization, the third dimension could be considered in order to better reflect the Earth conditions. The undersea or the airspace navigation offers many applications needing tridimensional coordinates. One example is the US-European *Jason* series of satellites for ocean altimetry, which can observe now, from an altitude of 1336 km, water level modifications in the range of few centimeters. Other examples are the celestial missions, or the exploration of the hemispheric craters.

These show that the presented investigation could be used in a broad range of optimization applications.

**Acknowledgements** The authors would like to thank Professor William Cook and Dr. Vasile Crăciunescu for their support. The authors would also like to thank the reviewers for their suggestions and insightful comments to improve this work.

## References

1. Adulyasak Y, Cordeau J-F, Jans R (2014) Formulations and Branch-and-Cut algorithms for multivehicle production and inventory routing problems. *INFORMS J Comput* 26(1):103–120
2. Ahammed F, Moscato P (2011) Evolving L-systems as an intelligent design approach to find classes of difficult-to-solve traveling salesman problem instances. *Lecture notes in computer science*, vol 6624, pp 1–11
3. Apiletti D, Baralis E, Cerquitelli T (2011) Energy-saving models for wireless sensor networks. *Knowl Inf Syst* 28(3):615–644
4. Applegate DL, Bixby RE, Chvátal V, Cook WJ (2006) *The traveling salesman problem: a computational study*. Princeton University Press, Princeton
5. ArcGIS online help (2014) <http://resources.arcgis.com/>
6. Asakura K, Fukaya K, Watanabe T (2013) A map construction system for disaster areas based on ant colony systems. *Procedia Comput Sci* 22:494–501
7. Ausiello G et al (2001) Algorithms for the on-line travelling salesman. *Algorithmica* 29(4):560–581
8. Church R, ReVelle C (1974) The maximal covering location problem. *Pap Reg Sci* 32(1):101–118
9. Cohoon J et al (1998) Perturbation method for probabilistic search for the traveling salesperson problem. In: Bosacchi B et al (eds) *Applications and science of neural networks, fuzzy systems, and evolutionary computation*. SPIE Press, vol 3455, p 118
10. Concorde solver (2011) <http://www.math.uwaterloo.ca/tsp/concorde.html>
11. Cook WJ (2011) *In pursuit of the travelling salesman: mathematics at the limits of computation*. Princeton University Press, Princeton
12. Cook WJ (2005) *The traveling salesman problem web server* <http://www.math.uwaterloo.ca/tsp/>
13. Crainic TG, Crisan GC, Gendreau M, Lahrichi N, Rei W (2009) Multi-thread cooperative optimization for rich combinatorial problems. In: *IEEE international parallel & distributed processing symposium*, Rome, Italy, pp 2284–2291

14. Crisan GC, Nechita E (2008) Solving fuzzy TSP with ant algorithms. *Int J Comput Commun Control* 3S(3):228–231
15. Czáríunescu V (2007) Romania: general vector datasets. <http://earth.unibuc.ro/download/romania-set-uri-vectoriale>
16. Crnkovic I, Larsson M (2002) Building reliable component-based software systems. Artech House Publisher, Norwood
17. Crucitti P, Latora V, Marchiori M, Rapisarda A (2004) Error and attack tolerance of complex networks. *Phys A* 340:388–394
18. Curtin KM, Voicu G, Matthew TR, Stefanidis A (2014) A comparative analysis of traveling salesman solutions from geographic information systems. *Trans GIS* 18(2):286–301
19. Czyzyk J, Mesnier MP, Morao JJ (1998) The NEOS server. *IEEE J Comput Sci Eng* 5(3):68–75
20. Dolan E (2001) The NEOS server 4.0 administrative guide. In: Technical memorandum ANL/MCS-TM-250 Mathematics and Computer Science Division Argonne National Laboratory
21. Dorigo M, Gambardella LM (1997) Ant colonies for traveling salesman problem. *BioSystems* 43:73–81
22. Eldrandaly KA, Abdallah AF (2012) A novel GIS-based decision-making framework for the school bus routing problem. *Geo-spat Inf Sci* 15(1):51–59
23. European Commission (2013) Guidance on integrating climate change and biodiversity into strategic environmental assessment. <http://ec.europa.eu/environment/eia/pdf/SEA%20Guidance>
24. Fanea A, Motogna S, Diosan L (2006) Automata-based component composition analysis. *Studia Universitatis Babeş-Bolyai Seria Informatica* 50(1):13–20
25. Fiechter C-N (1994) A parallel tabu search algorithm for large traveling salesman problems. *Discrete Appl Math* 51(3):243–267
26. Fiedrich F, Gehbauer F, Rickers U (2000) Optimized resource allocation for emergency response after earthquake disasters. *Saf Sci* 35(1–3):41–57
27. Fischer T, Stützle T, Hoos H, Merz P (2005) An analysis of the hardness of TSP instances for two high-performance algorithms. In: Doerner KF et al (ed) Proceedings of the 6th metaheuristics international conference 2005, pp 361–367
28. Geosphere R package. <https://cran.r-project.org/web/packages/geosphere/>
29. Goldwasser M, Johnson DS, McGeoch CC (eds) (2002) In: Proceedings of the fifth and sixth DIMACS implementation challenges, American Mathematical Society
30. Gomory RE (1960) An algorithm for the mixed integer problem. Technical Report RM 2597, RAND Corporation
31. Google Maps Android and JavaScript APIs (2015) <https://developers.google.com/maps/>
32. GPS Visualizer (2013) <http://www.gpsvisualizer.com>
33. Gropp W, Moré JJ (1997) Optimization environments and the NEOS server. Approximation theory and optimization. Cambridge University Press, Cambridge
34. Guntsch M, Middendorf M (2001) Pheromone Modification strategies for ant algorithms applied to dynamic TSP. Lecture notes in computer science, vol 2037, pp 213–220
35. Gutin G, Punnen AP (eds) (2002) The traveling salesman problem and its variations. Combinatorial optimization, vol 12. Springer, New York
36. Hoos H, Stützle T (2014) On the empirical scaling of run-time for finding optimal solutions to the travelling salesman problem. *Eur J Oper Res* 238(1):87–94
37. IBM ILOG (2012) User's manual for CPLEX. [ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps\\_usrmanplex](ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmanplex)
38. Investigation Committee on the Accident at the Fukushima Nuclear Power Stations of Tokyo Electric Power Company (2011) Final report. <http://www.cas.go.jp/jp/seisaku/icanps/eng/final-report.html>
39. Jaillet P (1985) Probabilistic travelling salesman problems. Ph.D. thesis, MIT
40. Kang L, Zhou A, McKay B, Li Y, Kang Z (2004) Benchmarking algorithms for dynamic travelling salesman problems. *Congr Evol Comput* 2:1286–1292
41. Kemball-Cook D, Stephenson R (1984) Lessons in logistics from Somalia. *Disasters* 8:57–66
42. Kirac E et al (2015) The traveling salesman problem with imperfect information with application in disaster relief tour planning. *IIE Trans* 47(8):783–799
43. Konecny M, Zlatanova S, Bandrova TL (eds) (2010) Geographic information and cartography for risk and crisis management. Lecture notes in geoinformation and cartography, Springer
44. Kovács I, Barabási AL (2015) Destruction perfected. *Nature* 524:38–39
45. Land AH, Doig AG (1960) An automatic method of solving discrete programming problems. *Econometrica* 28(3):497–520
46. Lawler EL et al (1985) The traveling salesman problem: a guided tour of combinatorial optimization. Wiley, New York
47. Lawler EL, Wood DE (1966) Branch-and-bound methods: a survey. *Oper Res* 14:699–719

48. Li C, Yang M, Kang L (2006) A new approach to solving dynamic traveling salesman problems. *Lecture notes in computer science*, vol 4247, pp 236–243
49. Li X, Zhao Z, Zhu X, Wyatt T (2011) Covering models and optimization techniques for emergency response facility location and planning: a review. *Math Methods Oper Res* 74(3):1–30
50. Lin S, Kernighan BW (1973) An effective heuristic algorithm for the traveling-salesman problem. *Oper Res* 21(2):498–516
51. Little JDC et al (1963) An algorithm for the traveling salesman problem. *Oper Res* 11(6):972–989
52. Malandraki C, Dial RB (1996) A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *Eur J Oper Res* 90(1):45–55
53. Mathew N, Smith SL, Waslander SL (2013) A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In: *IEEE conference on robotics and automation*
54. Mitchell JE (2002) Branch-and-Cut algorithms for combinatorial optimization problems. *Handbook of applied optimization*. Oxford University Press, Oxford GB, pp 65–77
55. Montemanni R, Barta J, Mastrolilli M, Gambardella LM (2007) The robust traveling salesman problem with interval data. *Transp Sci* 41(3):366–381
56. Motogna S, Ciuciu I, Serban C, Vescan A (2015) Improving software quality using an ontology-based approach. *Lecture notes in computer science*, vol 9416, pp 456–465
57. National Atlas of the United States. <http://www.lib.ncsu.edu/gis/natatlas.html>
58. Nechita E, Talmaciu M, Muraru C (2012) A Bayesian approach for the assessment of risk probability. Case study for digital risk probability. *Environ Eng Manag J* 11(12):2249–2256
59. Padberg M, Rinaldi G (1991) A Branch-and-Cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *Siam Rev* 33:60–100
60. Padberg M, Rinaldi G (1987) Optimization of a 532-city symmetric traveling salesman problem by branch-and-cut. *Oper Res Lett* 6:1–7
61. Pintea C-M (2015) A unifying survey of agent-based approaches for equality-generalized traveling salesman problem. *Informatica* 26(3):509–522
62. Pop P, Matei O (2014) An efficient metaheuristic approach for solving a class of matrix optimization problems. In: Toklu YC, Bekdas G (eds) *Proceedings EU/ME workshop*, pp 17–25
63. Potvin J-Y (1996) Genetic algorithms for the traveling salesman problem. *Ann Oper Res* 63(3):337–370
64. Purta R, Dobski M, Jaworski A, Madey G (2013) A testbed for investigating the UAV swarm command and control problem using DDDAS. *Procedia Comput Sci* 18:2018–2027
65. Reinelt G (1994) *The traveling salesman: computational solutions for TSP applications*. Springer, New York
66. Ridge E, Kudenko D (2008) Determining whether a problem characteristic affects heuristic performance. *Stud Comput Intell* 153:21–35
67. Rodríguez A, Ruiz R (2012) The effect of the asymmetry of road transportation networks on the traveling salesman problem. *Comput Oper Res* 39:1566–1576
68. Romanian Parliament Law 351 (2001) [http://www.cdep.ro/pls/legis/legis\\_pck.hp\\_act\\_text?idt=28862](http://www.cdep.ro/pls/legis/legis_pck.hp_act_text?idt=28862)
69. Romania2950.tsp dataset (2014) doi:10.13140/2.1.4706.8165, <http://cadredidactice.ub.ro/ceraselacrisan/cercetare/>
70. Rossant C (2014) *IPython interactive computing and visualization cookbook*. Packt Publishing, Birmingham
71. Showalter PS, Lu Y (eds) (2010) *Geospatial techniques in urban hazard and disaster analysis. Geotechnologies and the environment*, vol 2. Springer, New York
72. Spielman D, Teng S-H (2001) Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. In: *STOC '01: proceedings of ACM*, pp 296–305
73. Stoean C, Stoean R (2014) *Support vector machines and evolutionary algorithms for classification*. Springer, New York
74. Su S, Yu S, Ma Y, Yang Y, Xu H (2011) Routing on a spherical surface using hybrid PSO. *Commun Comput Inf Sci* 237:43–51
75. TerraLib. <http://www.terralib.org/>
76. Toregas C, Swain R, ReVelle C, Bergman L (1971) The location of emergency service facilities. *Oper Res* 19(6):1363–1373
77. TSP data test (2009) <http://www.math.uwaterloo.ca/tsp/data/index.html>
78. Traveling Salesman Problem, TSP Website, Last Updated by William Cook: November 2014. <http://www.math.uwaterloo.ca/tsp/>
79. TSPLIB (2013) <http://www.iwr.uni-heidelberg.de/groups/comopt/software>
80. Urquhart N, Scott C, Hart E (2013) Using graphical information systems to improve vehicle routing problem instances. In: *GECCO'13 companion*, ACM, NY, pp 1097–1102

81. van Hemert JI (2005) Property analysis of symmetric travelling salesman problem instances acquired through evolution. Lecture notes in computer science, vol 3448, pp 122–131
82. Wex F et al (2014) Emergency response in natural disaster management: allocation and scheduling of rescue units. Eur J Oper Res 235(3):697–708
83. Wise S (2013) GIS fundamentals, 2nd edn. CRC Press
84. Zacarias F, Cuapa R, De Ita G, Torres D (2015) Smart tourism in 1-click. Procedia Comput Sci 56:447–452



**Dr. Gloria Cerasela Crișan** is an Associate Professor in the Faculty of Sciences at the Vasile Alecsandri University of Bacău, Romania. Her research is focused on topics that include heuristic and exact solving methods for combinatorial optimization problems, uncertainty modeling, complex real-life problems. She received her Ph.D. in Informatics from the Alexandru Ioan Cuza University of Iași, Romania.



**Dr. Camelia-M. Pinte** is currently working at the TU Cluj-Napoca, Romania. Her Ph.D. and habilitation thesis are in informatics. She is member of scientific committee and reviewer of several journals and conferences. Dr. Habil. Pinte has authored a large number of articles and chapters related to combinatorial optimization, artificial intelligence, metaheuristics, multiagent systems and complex systems. Dr. Habil. Pinte is an IEEE Senior Member, member of the IEEE Computational Intelligence Society and co-chair of the IEEE-CIS Interdisciplinary Emergent Technologies.



**Dr. Vasile Palade** joined the Department of Computing at Coventry University in September 2013 after working for many years with the Department of Computer Science of the University of Oxford. His research interests lie in the area of machine learning with various applications. Dr. Palade is author of more than 120 papers in journals and conference proceedings as well as books on machine learning/computational intelligence and applications. He is an Associate Editor for several reputed journals. He has delivered keynotes and chaired international conferences on machine learning and applications. Dr. Vasile Palade is an IEEE Senior Member and a member of the IEEE Computational Intelligence Society.