A Real-Time Emergency Inspection Scheduling Tool Following a Seismic Event
Nikos Ath. Kallioras and Nikos D. Lagaros

Abstract Emergency infrastructure inspections are of the essence after a seismic event as a carefully planned inspection in the first and most critical hours can reduce the effects of such an event. Metaheuristics and more specifically nature inspired algorithms have been used in many hard combinatorial engineering problems with significant success. The success of such algorithms has attracted the interest of many researchers leading to an increased interest regarding metaheuristics. In the present literature many new and sophisticated algorithms have been proposed with interesting performance characteristics. On the other hand, up to date developments in the field of computer hardware have also had a significant influence on algorithm design. The increased computational abilities that are available to researchers through parallel programming have opened new horizons in architecture of algorithms. In this work, a methodology for real-time planning of emergency inspections of urban areas is presented. This methodology is based on two nature inspired algorithms, Harmony Search Algorithm (HS) and Ant Colony Optimization (ACO). HS is used for dividing the area into smaller blocks while ACO is used for defining optimal routes inside each created block. The proposed approach is evaluated in an actual city in Greece, Thessaloniki.

Keywords Nature inspired algorithms, parallel programming, GPU programming, harmony search, ant colony optimization, emergency inspections, districting problems, routing problems

N.D. Lagaros (Corresponding Author)
Institute of Structural Analysis & Seismic Research, National Technical University of Athens, 9, Iroon Polytechniou Str., Zografou Campus, GR-15780, Athens, Greece
e-mail: nlagaros@central.ntua.gr

N.Ath. Kallioras
Institute of Structural Analysis & Seismic Research, National Technical University of Athens, 9, Iroon Polytechniou Str., Zografou Campus, GR-15780, Athens, Greece
e-mail: nkallioras@yahoo.com
1 Introduction

Following a catastrophic seismic event, the first hours are critical as far as rescue operations, evacuation procedures and infrastructure repair procedures are considered. As such events can lead to severe short and long-term economic losses, the efficiency of the civil services has to be of a great level. In case of a hazardous seismic events civil services are challenged by damaged infrastructure networks, communities in a panicking state-of-mind and system vulnerabilities. Post-disaster management is a multi-level procedure which includes a thorough planning of system response, as well as tactical and operational management. Designing of an optimized inspection plan is definitely a very complex NP-hard combinatorial problem. Metaheuristics present a very successful behavior when dealing with such problems as the one previously described.

Latest advances in metaheuristic algorithms and specifically nature inspired algorithms present automated trial-and-error techniques mostly based on the survival-of-the-fittest principal as species evolution does. It also important to point out that the advances in the available hardware and parallel programming have also offered a significant assist in the implementation of metaheuristics. As performance features of single cores of Central Processing Units (CPUs) have not improved significantly in the last years, CPU manufactures turned to producing multi-core CPUs to increase performance through parallel processing. Graphic processing units (GPUs) also consist of processing cores which in the recent past have been used for mathematical programming. Modern hi-end CPUs consist of 18 cores with clock rates up to 3.6 GHz. On the other hand, high-end GPUs have more than 2,500 cores with clock rates up to 875 MHz. These two different types of processor, each with advantages and disadvantages, offer the ability to efficiently handle problems that in the past were too demanding to be solved.

In this work, two algorithms are used for designing the emergency inspection plan of urban areas. The first algorithm, Improved Harmony Search (IHS), was presented by Kallioras et al. [1] in 2014 while the original version of the algorithm was introduced by Geem et al. [2] in 2001. The second algorithm used in this work is Ant Colony Optimization (ACO) which was presented by Dorigo and Stützle [3] in 2004. The problem of finding the optimal inspection plan of urban structures following a seismic event can be divided into two sub-problems. Firstly, the urban area must be divided into smaller areas, one for each available inspection crew (districting sub-problem), while for each defined area, an optimal route is designed inside its premises (routing sub-problem). The districting sub-problem is dealt with the use of IHS as presented later on in this work while the routing sub-problem is solved with the use of ACO. The biggest challenge in designing this formulation apart from the quality of proposed results, is the necessary computational work-time for the algorithms to locate the solution. In order for this implementation to be able to act as a real-time solution, executional time is rather critical. With the use of GPU programming, computational times are reduced to minimum as can be seen in this work.
2 Post-Earthquake Response Mechanism

Inspection time required for assessment of possible damages in structure and infra-structure systems is critical for minimizing cost of effects of a possible hazardous event. Every scheduled event such as support, recovery and rehabilitation can only be properly scheduled after the inspection of infrastructures is completed. This proves the importance of a carefully designed inspection plan towards dealing with a seismic event in an urban area. Generally speaking, we can divide the disaster response procedure [4] into four basic steps:

- Mitigation [5], which includes assessment of seismic hazards, probabilistic damage projection [6,7], and integration of emergency processes through decision support systems [8,9].
- Preparedness, which includes preparation for dealing with seismic events and evacuation procedures [10-15].
- Response, which basically deals with creating a plan for response-relief operations [16-19] and post-disaster infrastructure performance evaluation [20-23].
- Recovery which deals with relief performance assessment [24], protection of infrastructure element [25], and repair-fund allocation [26,27].

Emergency response has attracted the interest of many researchers and significant work has been done in this field in the past years. Despite that, after reviewing up-to-date literature, it can be seen that very little work is done considering the proper planning of inspection crews following a seismic event though it is a very important procedure. This can be explained due to the fact that necessary data is difficult to be collected, solving the associated mathematical problem is very demanding and also, formulating an algorithm that can handle such data and calculations in real-time procedure is exceptionally difficult. In this work, we present the mathematical formulation and solution of such a problem while we also present the parallel formulation of a software that solves such problems in real-time.

3 IHS and ACO Metaheuristic Algorithms

In this work, two metaheuristic algorithms are used for real-time inspection scheduling. As discussed previously, IHS is applied on the districting problem while ACO is used for solving the routing problem. IHS algorithm consists of four basic actions similar to HS. These actions are [1]:

- Parameter Initialization
  In this step, the parameters of the algorithm and the problem are defined: 
  \( n \) is the number of decision variables, \( s_i^l \leq s_i \leq s_i^u \), \( i=1,2,...,n \) are the lower and upper bounds of each decision variable, \( HMS \) is the size of the harmony memory (HM) of solution vectors and \( HMCR \) is the rate for considering the harmony memory.
• **Harmony Memory Initialization**
  In this step, the initialization of the algorithm’s memory is executed

\[
HM = \begin{bmatrix}
s_1^1 & s_2^1 & s_3^1 & \cdots & s_n^1 \\
s_1^2 & s_2^2 & s_3^2 & \cdots & s_n^2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
s_1^{m} & s_2^{m} & s_3^{m} & \cdots & s_n^{m}
\end{bmatrix}
\]  

(1)

where \(s_i^j\) is the \(i_{th}\) element of the \(j_{th}\) solution vector \(s\)

• **New Harmony Improvisation**
  In this step, a new harmony-solution vector is generated with the use of the random selection procedure or memory consideration procedure. In random selection, each member of the solution vector is randomly generated with probability \(1 - \text{HMCR} (0 \leq \text{HMCR} \leq 1)\). In memory consideration, each variable of the solution vector is randomly chosen from HM with probability \(\text{HMCR}\).

\[
s_{i}^{\text{new}} = \begin{cases} 
  s_i \in [s_i^j, s_i^j'] \text{ with probability } 1 - \text{HMCR} \\
  s_i \in \text{HM} = \{s_i^1, s_i^2, ..., s_i^{\text{max}}\} \text{ with probability } \text{HMCR}
\end{cases}
\]  

(2)

where \(s_{i}^{\text{new}}\) is the new generated solution vector

• **Harmony Memory Update**
  In this step, the generated solution vector is compared to the worst one stored in HM according to the objective function of each problem. If it is better, it replaces it. If not, it is dropped.

This procedure is repeated as many times as the population of the maximum function evaluations set by the user. Once the maximum function evaluations are completed, the best solution stored in HM is the solution of the problem.

ACO algorithm [3] was inspired by the food searching pattern of ants in nature. As ants search inside an area for possible food resources and the shortest route between the nest and the food position, artificial ants search inside a weighted graph for the optimal path. The search space consists of nodes which represent the places that need to be visited by the ants. At first, a colony of \(m\) ants are positioned randomly inside the search space. At each repetitive step of the algorithm, ant \(k\) decides which of the nodes it will visit next with the use of a random proportional rule. In order to avoid visiting nodes which are already visited by the same ant, ant \(k\) currently positioned at node \(i\), maintains a memory \(M^k\) containing all nodes previously visited. This memory defines the feasible neighbourhood \(N_i^k\) which contains the nodes that have not been visited by ant \(k\). Ant \(k\), positioned at node \(i\), chooses to move to node \(j\) with a probability defined as follows:

\[
p^{i,j}_{k} = \frac{(\tau_{i,j})^\alpha \cdot (\eta_{i,j})^\beta}{\sum_{j \in N_i^k} (\tau_{i,j})^\alpha \cdot (\eta_{i,j})^\beta}, \quad \text{if } j \in N_i^k
\]  

(3)
where $\tau_{ij}$ is the amount of pheromone between nodes $i$ and $j$, $\alpha$ is a parameter controlling the pheromone’s influence, $\eta_{ij}$ is a heuristic information denoting the desirability of the path between nodes $i$ and $j$ where:

$$\eta_{ij} = \frac{1}{d_{ij}}$$  \hspace{1cm} (4)

while $\beta$ is a parameter controlling the influence of the path’s desirability $\eta_{ij}$. The probability of choosing a particular connection $ij$ increases with the value of the pheromone trail $\tau_{ij}$ and the heuristic information value $\eta_{ij}$. When all ants have completed their routes, the pheromone concentration for each connection between $i$ and $j$ nodes, is updated for the next iteration $t+1$ as follows:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{K} \Delta \tau_{ij}^k(t), \quad \forall (i, j) \in A$$  \hspace{1cm} (5)

where $\rho$ is the rate of pheromone evaporation, $A$ is the set of paths (edges or connections) that fully connects the set of nodes and $\Delta \tau_{ij}^k(t)$ is the amount of pheromone ant $k$ has deposited on connections it has visited during its tour $T^k$ and it is given by:

$$\Delta \tau_{ij}^k(t) = \begin{cases} 0 & \text{if connection } (i,j) \text{ belongs to } T^k \\ \frac{Q}{L(T^k)} & \text{otherwise} \end{cases}$$  \hspace{1cm} (6)

The main concept behind the ACO algorithm is that connections chosen by many ants and have a shorter length, receive bigger amounts of pheromone due to more deposition and less evaporation. This ensures that these paths are more likely to be chosen by other ants in future iterations of the algorithm resulting to the path with the minimum distance.

## 4 Optimal Inspection Problem Formulation

As described previously, the optimal inspection problem is divided into the districting and the routing sub-problem. In the first sub-problem, the urban domain is divided into a number of areas of responsibility equal to the population of the available inspection crews. In the second sub problem, for each area of responsibility, the optimal route is designed. Urban areas consist of building blocks and road structures. Each building block is defined by the coordinates of its edges, its area, use of land, building factor and maximum allowed height of constructions. These characteristics provide us with the demand evaluation of each building block. Neighbouring city blocks with similar characteristics regarding use of land, building factor and allowed height can be joined and considered as one in order to reduce the dimensionality of the problem without degrading the quality of the solution. The problem is formulated as a nonlinear optimization problem and the objective function is:

$$\min \sum_{i=1}^{N_C} \sum_{k=1}^{d[i]} \left[ \frac{D(k)}{U_{in}} + \frac{d(SB_i, C_i)}{U_{tr}} \right] \cdot \sigma_{ik}$$  \hspace{1cm} (7)
where \( N_{IC} \) is the number of the available inspection crews, \( n_{SB} \) is the number of structural (or building) blocks assigned to the \( i \)-th inspection crew, \( d(SB_k, C_i) \) is the distance between the \( SB_k \) building block and \( C_i \) is the starting block of the crew responsible for the \( i \)-th group of structural blocks, \( U_w \) is the inspection speed of the inspection crews, and \( U_r \) is the travelling speed of the inspection crews. \( D(k) \) is inspection “demand” for the \( k \)-th building block defined as the product of the building block total area \( A(k) \) times the building factor \( f_B(k) \) (i.e. the structured percentage of the area) and \( \sigma_{IC} \) is the value of the standard deviation of the working hours of all inspection crews. Thus, the districting problem is formulated as a discrete unconstrained nonlinear optimization problem where the objective is to define which inspection crew is responsible for which building block. The node of each building block is placed in its geometrical centre while all Euclidean distances are calculated with respect to the centre of weight of the building block.

According to the described formulation, every available inspection crew is assigned to a specific district. In the general form of the districting problem, the population of the design variables is equal to the number of building blocks. This leads to a problem with increased complexity. In order to avoid this complexity, the authors suggest a modification to the problem formulation by setting the starting point of the inspection crews as the design variables. This reduces the complexity from \( n_{SB} \) to \( N_{IC} \). This is accomplished with the use of a four-step procedure where at first

- The starting positions of the inspection crews are defined
- Areas of responsibility are defined by assigning each building block to the closest inspection crew
- The centre of gravity of each area of responsibility is found and this becomes the starting point of each inspection crew and finally,
- New areas of responsibility are created around the new starting positions.

The optimal routing (also called scheduling) problem of the inspection crews is formulated as a travelling salesman problem (TSP). TSP is represented by a weighted graph \( G=(N,A) \), where \( N \) is the set of nodes and \( A \) is the set of connections/paths that connect all \( N \) nodes. A cost function is assigned to paths between two nodes \( (i \neq j) \), represented by the distance between the two nodes \( d_{ij} \). A solution of the TSP is a permutation \( p=\{p(1), \ldots, p(N)\}^T \) of the node indices \( \{1, \ldots, N\} \), as every node must not appear more than once in a solution. The solution that minimizes the total length \( L(p) \) given by:

\[
\mathcal{L}(p) = \sum_{i=1}^{N-1} \left( d_{p(i),p(i+1)} + d_{p(N),p(1)} \right)
\]

is noted as the optimal one. The corresponding scheduling problem is defined as follows:

\[
\min \left[ \sum_{k=1}^{n_{SB}^{(i)}} d(SB_i, SB_{k+1}) + d(SB_i, SB_j) \right], i = 1, \ldots, N_{KG}
\]
where \(d(SB_k, SB_{k+1})\) is the distance between the \(k\)th and \(k+1\)th building blocks. The desired result is to locate the shortest route between all structural blocks assigned to each inspection crew.

### 5 Numerical Tests

The development and testing of the proposed real-time application is executed on pragmatic data of a city. The chosen city Thessaloniki is the second biggest city in Greece and has in the past suffered from a seismic event. On the 20th of June of 1978 an earthquake with a magnitude of 6.5 R took place in Thessaloniki and caused the death of 45 people while 9,480 buildings suffered non-repairable damages. The city’s area is equal to 154,205.128 m\(^2\) while it consists of 471 joint structural blocks as it can be seen in Figure 1. A hypothetical seismic event is applied in the urban structure of Thessaloniki and an inspection scenario is examined.

According to the scenario, there are 40 inspection crews available for the required task. These crews work in 8-hours shifts per day resulting to having 20 crews working in the city at any moment for 16 hours per day. As we were expecting small differences in the necessary working hours of each crew, we set the time needed for the whole inspection to be completed equal to the maximum inspection time of the various crews. The parameter values for IHS and ACO algorithms are chosen with respect to a previous work of the authors which included a thorough sensitivity analysis [29 - TRC]. In detail, HMS is chosen equal to 12 and HMCR is equal to 0.105693 regarding IHS. Regarding ACO, \(\alpha = 1.0\), \(\beta = 2.0\), \(\rho = 0.10\), \(m = 65\) and \(Q = 0.306\). The average inspection speed of the crews, \(U_{in}\) is set equal to 50 square meters per minute while the average traveling speed of the crews, \(U_{tr}\) is set equal to 10km/hour. For IHS the maximum iterations for are equal to 200,000 while for ACO they are equal to 200.

| Inspection Crew | Inspection Time (hours) | Travel Time (hours) | Total Time (hours) |
|-----------------|-------------------------|---------------------|--------------------|
| Inspection Crew 1 | 5556.778                | 3.150               | 5559.928           |
| Inspection Crew 2 | 5574.094                | 1.722               | 5575.816           |
| Inspection Crew 3 | 5679.853                | 3.152               | 5683.006           |
| Inspection Crew 4 | 4569.000                | 1.707               | 4570.708           |
| Inspection Crew 5 | 5154.745                | 1.764               | 5156.509           |
| Inspection Crew 6 | 4918.512                | 1.595               | 4920.107           |
| Inspection Crew 7 | 4826.740                | 1.676               | 4828.417           |
| Inspection Crew 8 | 5087.935                | 1.800               | 5089.735           |
| Inspection Crew 9 | 5097.223                | 1.554               | 5098.777           |
| Inspection Crew 10 | 4936.829                | 3.022               | 4939.851           |
The hours needed for the inspection crews to complete the task can be seen in Table 1. In particular, the maximum time which also corresponds to the end of the inspection is equal to 5683.006 hours. The average working hour of the ten crews is equal to 5142.285 hours while the maximum variation from the average time needed for the crews to finish the inspection crew is equal to 11.11%. The value of the objective function is equal to 18650882.911. The best and worst value of the objective function stored in IHS memory per each of the 200,000 iterations can be seen in Figure 2.

Fig. 1. The city-plan of Thessaloniki and the 471 joint city blocks

In Figure 3, the results of IHS applied on the districting problem can be seen. Each colour represents an inspection crew and an area of responsibility. In Figure 4 the routing results of ACO are presented while in Figure 5, a closer look on 5 of the designed routes are visible.
The minimization of the computational effort and time needed to solve this problem is accomplished with the use of GPU programming applied on the ACO algorithm and parallelization of the IHS algorithm. In order to evaluate the gain of such an implementation, the same problem is solved twice, first with the suggested techniques and then without them. Both implementations are solved ten times and the average execution times are presented. The GPU version is completed after $T_1 = 225.65$ sec while the unmodified version demands $T_2 = 1242.87$ sec. The acceleration factor is equal to $T_2 / T_1 = 5.508$ which is rather significant. The acceleration factor increases as the dimensionality of the problem increases as presented by Kallioras et al [29] in a previous work. It is also important to point out that a more up to date GPU processing unit than the one used here would have given even better acceleration results. None the less, solving such a problem in less than 4 minutes does satisfy the criterion as a real-time solver. The experiments of the current study were performed using the Intel i7 3610QM 2.30Ghz processor with 4 cores paired with 12GB memory and the NVIDIA GeForce GTX 660M with 384 stream processors (CUDA cores) with 2GB of memory and supports CUDA compute capability 3.0.
Fig. 3. Districting results for the city of Thessaloniki. Each color represents an area of responsibility for each of the 10 inspection crews.
6 Conclusions

In this work, a real-time emergency inspection scheduling application following a seismic event is presented. This application is based on two nature-inspired metaheuristic algorithms, IHS and ACO. The first is a variation of the well-known and established Harmony Search algorithm while the second one is the also established Ant Colony Optimization algorithm. Both algorithms have proven their ability to handle hard combinatorial problems in literature. Both algorithms have also proven their robustness when applied on such problems. Through this work, a parallel GPU implementation of the algorithm is presented in order to minimize the computational time needed to solve such problems. Computational time is very important.
when we are dealing with situations where solutions need to be presented in real-time. Dealing with a seismic event is such a situation as a delay in scheduling support and relief efforts can cost dramatically.

Fig. 5. Closer look on the routes for 5 of the inspection crews inside their area of responsibility.

Our approach is implemented on the urban environment of the second largest city in Greece, Thessaloniki, who has suffered from a seismic event in the recent past. With the assumption that 20 inspection crews will be available and working on 8-hour shifts, we have 10 crews inspecting the urban structures for 16 hours per day. Based on these estimations, along with the inspection and traveling speed of crews, the necessary time for the inspection of the whole area was equal to 5683.006 hours while the variation between crews is also very small.

By implementing the algorithms with the use of GPU programming, a significant acceleration factor equal to 5.508 is achieved which is very important in real-time applications. It is also important to point out that the acceleration factor increases with the increase of the dimensionality of the problem. The proposed formulation has the ability to robustly solve the described problem in less than four minutes. In the future, it might be useful to apply modifications on the algorithm which will
offer the end-user the ability to choose before executing the algorithm a series of joint city blocks that need to be inspected first. It can also be modified to resolve the problem while city blocks with increased level of damage are set by the user according to emergency calls. Another modification that is worth investigating is giving the algorithm the ability to un-assign city blocks from crews with large inspection crews and assigning them to those who finish first.

Acknowledgments

The research study in the current chapter has been conducted in the frame of the Research Funding Program: Marie Sklodowska-Curie Research and Innovation Staff Exchanges (RISE) “OPTARCH – Optimization Driven Architectural Design of Structures”, H2020-MSCA-RISE-2015.

References

1. Kallioras, N.A., Lagaros, N.D., Karlaftis, M.G. (2014) An improved harmony search algorithm for emergency inspection scheduling. *Engineering Optimization*, 46: 1570-1592.
2. Geem, Z.W., Kim, J.H., Loganathan, G.V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*; 76: 60-68.
3. Dorigo, M., Stützle, T. (2004). Ant Colony Optimization, *The MIT Press*.
4. Altay, N., Greene, W.G. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*; 175: 475-493.
5. Dong, W.M., Chiang, W.L., Shah, H.C. (1987). Fuzzy information processing in seismic hazard analysis and decision making. *Soil Dynamics and Earthquake Engineering*; 6(4): 2202-2226.
6. Peizhuangm W., Xihui, L., Sanchez, E. (1986). Set-valued statistics and its application to earthquake engineering. *Fuzzy Sets and Systems*; 18(3): 347-356.
7. Tamura, H., Yamamoto, K., Tomiyama, S., Hatono, I. (2000). Modelling and analysis of decision making problem for mitigating natural disaster risks. *European Journal of Operational Research*; 122(2): 461-468.
8. Mendonca, D., Beroggi, G.E.G., Wallace, W.A. (2001). Decision support for improvisation during emergency response operations. *International Journal of Emergency Management*; 1(1): 30-38.
9. Mendonca, D., Beroggi, G.E.G., Van Gent, D. (2006). Designing gaming simulations for the assessment of group decision support systems in emergency response. *Safety Science*; 44: 523-535.
10. Viswanath, K., Peeta, S. (2003). Multicommodity maximal covering network design problem for planning critical routes for earthquake response. *Transportation Research Record*; 1857: 1-10.
11. Nicholson, A., Du, Z-P. (1997). Degradable transportation networks systems: An integrated equilibrium model. *Transportation Research Part B;* 31(3): 209-223.

12. Sakakibara, H., Kajitani, Y., Okada, N. (2004). Road network robustness for avoiding functional isolation in disasters. *Journal of Transportation Engineering;* 130(5): 560-567.

13. Sohn, J. (2006). Evaluating the significance of highway network links under the flood damage: An accessibility approach. *Transportation Research part A;* 40(6): 491-506.

14. Song, J., Kim, T.J., Hewings, G.J.D., Lee, J.S., Jang, S.-G. (2003). Retrofit priority of transport network links under an earthquake. *Journal of Urban Planning and Development;* 129(4): 195-210.

15. Verter, V., Lapierre, S. (2002). Location of preventive healthcare facilities. *Annals of Operations Research;* 110: 123-132.

16. Barbarosoglu, G., Arda, Y. (2004). A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society;* 55(1): 43-53.

17. Barbarosoglu, G., Ozdamar, L., Cevik, A. (2002). An Interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research Society;* 140(1): 118-133.

18. Fiedrich, F., Gehbauer, F., Rickers, U. (2000). Optimized resource allocation for emergency response after earthquake disasters. *Safety Science;* 35(1-3): 41-57.

19. Ozdamar, L., Ekinci, E., Kucukyazici, B. (2004). Emergency logistics planning in natural disasters. *Annals of Operations Research;* 129(1-4): 217-245.

20. Bell, M.G.H. (2000). A game theory approach to measuring the performance reliability of transportation networks. *Transportation Research Part B;* 34(6): 533-545.

21. Chang, S.E., Nojima, N. (2001). Measuring post-disaster transportation system performance: The 1995 Kobe earthquake in comparative perspective. *Transportation Research part A;* 35(6): 475-494.

22. Karaouchi, F., Lida, Y., Shimada, H. (2001). “Evaluation of road network reliability considering traffic regulation after a disaster. The Network Reliability of Transport: Proceedings of the 1st International Symposium on Transportation Network Reliability (INSTR), Bell M.G.H., Lida, Y., (Eds.), Elsevier, Oxford, United Kingdom.

23. Li, Y., Tsukaguchi, H. (2001). “Improving the reliability of street networks in highly densely populated urban areas. The Network Reliability of Transport: Proceedings of the 1st International Symposium on Transportation Network Reliability (INSTR)”, (Bell M.G.H. and Lida, Y., Eds.), Elsevier, Oxford, United Kingdom.

24. Song, B., Hao, S., Murakami, S., Sadohara, S. (1996). Comprehensive evaluation method on earthquake damage using fuzzy theory. *Journal of Urban Planning and Development;* 122(1): 1-17.
25. Cret, L., Yamakazi, F., Nagata, S., Katayama, T. (1993). Earthquake damage estimation and decision-analysis for emergency shutoff of city gas networks using fuzzy set theory. *Structural Safety;* 12(1): 1-19.

26. Karlaftis, M.G., Kepaptsoglou, K.L., Lampropoulos, S. (2007). Fund allocation for transportation network recovery following natural disasters. *Journal of Urban Planning and Development;* 133(1): 82-89.

27. Lagaros, N.D., Karlaftis, M.G. (2011). A critical assessment of metaheuristics for scheduling emergency infrastructure inspections, *Swarm and Evolutionary Computation;* 1(3): 147-163.