Evolving Protection Measures for Lava Risk Management Decision Making

Giuseppe Filippone, Donato D'Ambrosio, Davide Marocco and William Spataro

Abstract Many volcanic areas around the World are densely populated and urbanized. For instance, Mount Etna (Italy) is home to approximately one million people, despite being the most active volcano in Europe. Mapping both the physical threat and the exposure and vulnerability of people and material properties to volcanic hazards can help local authorities to guide decisions about where to locate a priori critical infrastructures (e.g. hospitals, power plants, railroads, etc.) and human settlements and to devise for existing locations and facilities appropriate mitigation measures. We here present the application of Parallel Genetic Algorithms for optimizing earth barriers construction by morphological evolution, to divert a case study lava flow that is simulated by the numerical Cellular Automata model Sciara-fv2 at Mt Etna volcano (Sicily, Italy). The devised area regards Rifugio Sapienza, a touristic facility located near the summit of the volcano, where the methodology was applied for the optimization of the position, orientation and extension of an earth barrier built to protect the zone. The study has produced extremely positive results, providing insights and scenarios for the area representing, to our knowledge, the first application of morphological evolution for lava flow mitigation.

Keywords Evolutionary computation · Genetic algorithms · Parallel computing · Decision support system · Cellular automata · Morphological evolution

1 Introduction

When dealing with lava flow risk assessment, the use of thematic maps of volcanic hazard is of fundamental relevance to support policy managers and administrators in effective land use planning and taking proper actions that are required during an

G. Filippone (✉) · D. D'Ambrosio · W. Spataro
Department of Mathematics and Computer Science, University of Calabria,
Rende, Italy
e-mail: G.Filippone@ed.ac.uk

D. Marocco
Centre of Robotics and Neural Systems, School of Computing and Mathematics,
Plymouth University, Plymouth, UK

K. Madani et al. (eds.), *Computational Intelligence*,
Studies in Computational Intelligence 613,
DOI 10.1007/978-3-319-23392-5_4

emergency phase. In particular, hazard maps are a key tool for emergency management by describing the threat that can be expected at a certain location for future eruptions. At Mt. Etna (Italy), the most active volcano in Europe, the majority of events that have occurred in the last four centuries report damage to human properties in numerous towns on the volcano flanks [1]. Current efforts for hazard evaluation and contingency planning in volcanic areas depend heavily on hazard maps and numerical simulations for the purpose of individuating affected areas in advance.

Although many computational modeling methods [2–4] for lava flow simulation and related techniques for the compilation of susceptibility maps are already known to the international scientific community, the problem of defining a standard methodology for the construction of protection works, in order to mitigate volcanic risk, remains open. Techniques to slow down and divert lava flows, caused by collisions with protective measures such as artificial barriers [5, 6] or dams [7], are now to be considered empirical, exclusively based on past experiences. The proper positioning of protective measures in the considered area may depend on many factors (viscosity of the magma, output rates, volume erupted, steepness of the slope, topography, economic costs). As a consequence, in this context one of the major scientific challenges for volcanologists is to provide efficient and effective solutions.

Morphological Evolution (ME) is a recent development within the field of engineering design, by which evolutionary computation techniques are used to tackle complex design projects. This branch of evolutionary computation is also known as evolutionary design and it is a multidisciplinary endeavour that integrates concepts from evolutionary algorithms, engineering, and complex systems to solve engineering design problems [8]. Morphological evolution has been largely explored in evolutionary robotics, both for the design of imaginary 3D robotics bodies [9] and for the efficient and autonomous design of adaptive moving robots [10]. Principles of evolutionary design have been also applied in structural engineering at different level of the design process, from the structural design itself to the logistic involved in the construction [11].

This paper describes the application of ME by Parallel Genetic Algorithms (PGAs), for the first time to our knowledge, for optimizing earth barriers construction to divert a case study lava flow that is simulated by the latest release fv2 of the SCIARA Cellular Automata lava flow model [12]. Cellular Automata (CA) were introduced in 1947 by John von Neumann [13], quickly gaining the attention of the Scientific Community both as powerful parallel computational models and as a convenient apparatus for modeling and simulating several types of complex physical phenomena. CA have been applied to a variety of fields and their major interest regard their pratical use in Complex Systems modelling in Physics, Biology, Earth Sciences and Engineering (e.g., see [14–17]).The GA fitness evaluation, which was adopted for evaluating the "goodness" of the protective works of the CA model generated lava flow scenarios, has implied a massive use of the numerical simulator that runs thousands of concurrent simulations for every GA generation computation. Therefore, a GPGPU (General Purpose computation with Graphic Processor Units) library was developed to accelerate the GA execution. A visualization system [18] was also implemented, thereby allowing interactive analysis of the results.

Eventually, a study of GA dynamics, with reference to emergent behaviors, is also discussed later. In the following, after the description of the case study adopted for the experiments (Sect. 2), the main characteristics of the implemented algorithm, framework and results are presented (Sect. 3). Section 4 concludes the paper with final comments and future works.

2 The Case Study: The 2001 Mt Etna Eruption

The 2001 eruption of Mt. Etna began on July 17, characterized by lava emission from several vents on the southern flank of the volcano, at elevations of 2100, 2550, 2600, 2700, 2950, 3050 m, the latter four being directly connected to the conduit of the SE crater [19] (see Fig. 1). Lava flows emitted from the lowermost vents (2100, 2550, 2600–2700 m) caused damage and threatened some important facilities and infrastructure, which were protected by earthen barriers. Effusion rates at the main eruptive vents were estimated daily by [1] from the volume/time ratio and were obtained by careful mapping of the flow area and estimating its mean thickness. The facilities of the Sapienza zone were undoubtedly at risk because of their short distance from the 2700 and 2550 m effusive vents (respectively 3 and 2.5 km). The most probable path for the lava flow emitted from the 2100 m fissure was simulated (Crisci et al. 2001 and M.T. Pareschi, unpublished reports to Civil Protection) and was considered for the carried out experiments presented in the next sections. Thirteen artificial barriers were built during the July August 2001 Mt. Etna eruption. Their locations, together with investigated area here considered, are shown in the map of Fig. 1. The flow emitted from the lower vent, the 2100 m fissure, immediately interrupted the road SP92 and invaded a part of the adjacent wide parking area located between Mts. Silvestri and the Sapienza zone (1900 m a.s.l.). Starting on 21 July, a large barrier was progressively built on the eastern flank of the flow to protect two tourist facilities. This barrier worked properly and the two buildings were saved. The lava flow emitted from the 2100 m fracture descended about 6 km southwards (Fig. 1) and after the SP92 road near Mts. Silvestri it cut some other minor rural roads and destroyed a few isolated country houses. Had the lava advanced further, it would have re-crossed the SP92 road at a lower elevation, causing the complete isolation of the upper part of Mt. Etna. Workers and machines were moved to a possible critical point on the western front ready to build a diversion barrier to protect the road. An intervention plan was also set up for the protection of the Nicolosi and Belpasso villages, located on the most probable path of the lava, at only 4 km distance from its lowermost front. Eventually, the rate effusion decrease beginning in the last days of July prevented any further advance of the flow and thus the planned interventions were not necessary.

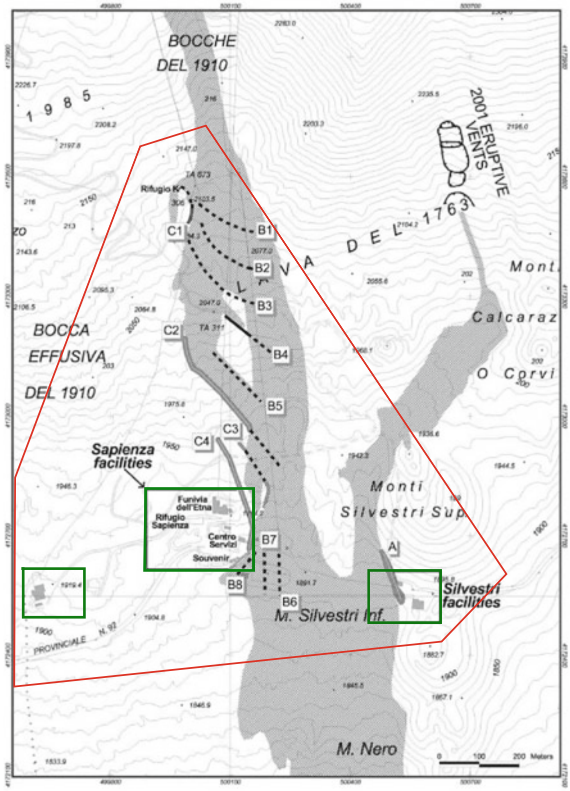


Fig. 1 Set of interventions carried out during the 2001 eruption event to divert the lava flow away from the facilities. The *green* perimeters represent the Rifugio Sapienza and other facilities (security area), which delimitates the area that has to be protected by the flow for the study. The *red* perimeter (work area), specifies the area in which the earth barrier can be located (Base figure taken from [5])

3 Morphological Evolution of Protective Works Through Parallel Genetic Algorithms

Genetic Algorithms (GAs) [20] are general-purpose iterative search algorithms inspired by natural selection and genetics. Among other applications, GA have been applied to combinatorial problems [21] in the study of the interaction between evolution and learning [22], evolutionary robotics [23, 24], for improving the performance of CA in resolving difficult computational tasks (e.g. [25]). GAs based methods have also been applied to CA for modelling bioremediation of contaminated soils [26] and for the optimisation of lava and debris flow simulation models (e.g., [27–30]).

GAs simulate the evolution of a population of candidate solutions, called phenotypes, to a specific problem by favouring the reproduction of the best individuals. Phenotypes are codified by genotypes, typically using strings, whose elements are called genes. In order to determine the best possible solution of a given problem, the GA must explore the so-called search (or solution) space, defined as the set of all possible values that the genotype can assume. The members of the initial population evaluated by means of a “fitness function”, determining the individuals “adaptivity”

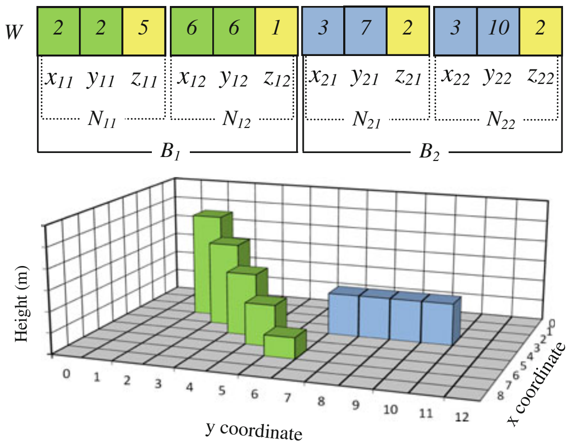


Fig. 2 Example of barriers encoding into a GA genotype. The height of the intermediate points of each barrier is obtained by connecting the work protections extremes through a linear function

value (also called fitness value), i.e. a measure of its goodness in resolving the problem. Best individuals are chosen by means of a "selection" operator and reproduced by applying random "genetic" operators to form a new population of offspring. Typical genetic operators are "crossover" and "mutation": they represent a metaphor of sexual reproduction and of genetic mutation, respectively. The overall sequence of fitness assignment, selection, crossover, and mutation is repeated over many generations (i.e. the GA iterations) producing new populations of individuals. According to the individual's probability of selection, any change that actually increases the individual's fitness will be more likely to be preserved over the selection process, thus obtaining better generationsas stated by the fundamental theorem of genetic algorithms [20]. For a complete overview of GAs, see [31, 32].

While GAs have been applied several times in the past for optimizing CA models, as the ones previously reported, by considering the 2001 Nicolosi case study, in this work GAs were adopted in conjunction with the SCIARA-fv2 CA model for the morphological evolution of protective works to control lava flows. The numerical model finite set of states was extended by introducing two substates defined as:

$$Z \subseteq R \tag{1}$$

where Z is the set of cells of the cellular automaton that specifies the Safety Zone, which delimitates the area that has to be protected by the lava flow and

$$P \subseteq R, P \cap Z = \emptyset \tag{2}$$

where P is the set of CA cells that identifies the Protection Measures Zone identifying the area in which the protection works are to be located.

The Protection work $W = B_1, B_2, \ldots, B_n$ was represented as a set of barriers, where every barrier $B_i = N_{i1}, N_{i2}$ is composed by a pair of nodes $N_{ij} = x_{ij}, y_{ij}, z_{ij}$, where x_{ij}, y_{ij} represent CA coordinates for the generic node j of the barrier i, and z_{ij} the height (expressed in m). The solutions were encoded into a GA genotype, directly as integer values (Fig. 2) and a population of 100 individuals, randomly generated inside the Protection Measures Zone, was considered.

Two different fitness functions were considered to suitably evaluate the goodness of a given solution: f_1, based on the areal comparison between the simulated event and the Safety Zone (in terms of affected area) and f_2, which considers the total volume of the protection works in order to reduce intervention costs and environmental impact.More formally, the f_1 objective function is defined as:

$$f_1 = \frac{\mu(S \cap Z)}{\mu(S \cup Z)} \tag{3}$$

where S and Z respectively identify the areal extent of the simulated lava event and the Safety Zone area, with $\mu(S \cap Z)$ e $\mu(S \cup Z)$ being the measures of their intersection and union. The function f_1, assumes values within the range [0, 1] where 0 occurs when the simulated event and Safety Zone Area are completely disjointed (best possible simulation) and 1 occurs when simulated event and Safety Zone Area perfectly overlap (worst possible simulation).

The f_2 objective function is defined as:

$$f_2 = \frac{\sum_{i=1}^{|W|} p_c \cdot d(B_i) \cdot h(B_i)}{V_{max}} \tag{4}$$

where $d(B_i)$ and $h(B_i)$ represent the length (in meters) and the average height of the ith barrier, respectively. The parameter p_c is the cell side and $V_{max} \in R$ is a threshold parameter (i.e., the maximum building volume) given by experts, for the function normalization. Since the barriers are composed of two nodes, the function can be written as:

$$f_2 = \frac{\sum_{i=1}^{|W|} p_c \cdot d(N_{i1}, N_{i2}) \cdot \bar{h}(N_{i1}, N_{i2})}{V_{max}} \tag{5}$$

where $\bar{h}(N_{i1}, N_{i2}) = \frac{|z_{i1}+z_{i2}|}{2}$ is considered as the average height value between two different nodes and $d(N_{i1}, N_{i2}) = \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2}$ identifies the Euclidean distance between them. The final fitness function f_2 is thus:

$$f_2 = \frac{\sum_{i=1}^{|W|} p_c \cdot \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2} \cdot \frac{|z_{i1}+z_{i2}|}{2}}{V_{max}} \tag{6}$$

The function f_2, assumes values within the range [0, 1]: it is nearly 0 when the work protection is the cheapest possible, 1 otherwise.

For the genotype fitness evaluation, a composite (aggregate) function f_3 was also introduced as follows:

$$f_3 = f_1 \cdot \omega_1 + f_2 \cdot \omega_2 \tag{7}$$

where $\omega_1, \omega_2 \in R$ and $(\omega_1 + \omega_2) = 1$, represent weight parameters associated to f_1 and f_2. Several different values where tested and the considered ones in this work chosen on the basis of trial and error techniques.The goal for the GA is to find a solution that minimizes the considered objective function $f_3 \in [0, 1]$.

In order to classify each genotype in the population, at every generation run, the algorithm executes the following steps:

1. CA cells elevation a.s.l. are increased/decreased in height on the basis of the genotype decoding (i.e., the barrier cells). In addition, an extending Bresenham‘s original algorithm [33] is applied to determine the cells inside the segment between the work protection extremes and f_2 subsequently computed.
2. A SCIARA-fv2 simulation is performed (about 40000 calculation steps) and the impact of the lava thickness on Z area (f_1 computation) is evaluated.
3. f_3 is computed and individuals are sorted according to their fitness.

The adopted GA is a rank based and elitist model, as at each step only the best genotypes generate off-spring. The 20 individuals which have the highest fitness generate five off-spring each and the $20 \times 5 = 100$ offspring constitute the next generation. After the rank based selection, the mutation operator is applied with the exception of the first 5 individuals.

The complete list of GA characteristics and parameters is reported in Table 1. Each gene mutation probability depends on its representation: p_{mc} for genes corresponding to coordinates value and p_{mh} viceversa. Therefore, if during the mutation process, a coordinate gene is chosen to be modified, the new value will depend on the parameters x_{max} and y_{max} which represent the cell radius within the node, the position of which can vary. The interval $[h_{min}, h_{max}]$ is the range within which the values of height nodes are allowed to vary (Fig. 3). This strategy ensures the possibility for the GA to provide, as output, either protective barriers or ditches.

To ensure a better exploration of the search space and to avoid a fast convergence of solutions to local optima a n point crossover operator has been introduced. Two parent individuals are randomly chosen from the mating pool and two different cutting points for each parents are selected. Cut points always coincide with the first gene of a sub-solution and after the selection portions of the sub-solution chosen in the genotype, they are exchanged. The crossover operator is applied according to a prefixed probability, p_c, for each sub-solution encoded in the genotype.

3.1 Parallel Implementation and Performance

The fitness evaluation of a GA individual consists in an entire CA simulation, followed by a comparison of the obtained result with the actual case study. This phase

Table 1 List of parameters of the adopted GA

GA parameters	Specification	Value
g_l	Genotypes length	6
p_s	Population size	100
n_g	Number of generations	100
p_{mc}	Coord. gene mutation probability	0.5
x_{max}	Gene x position variation radius	10
y_{max}	Gene y position variation radius	10
p_{mh}	Height gene mutation probability	0.5
h_{min}	height min variation range	−5
h_{max}	height max variation range	10
p_c	crossover probability	0.05
c_{h+}	Cost to build	1
c_{h-}	Cost to dig	1
ω_{f1}	f_1 weight parameter	0.90
ω_{f2}	f_2 weight parameter	0.10

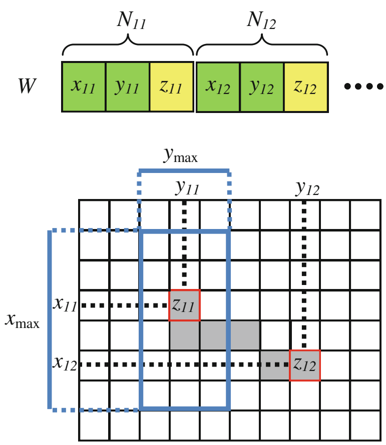


Fig. 3 Graphical representation of the genotype mutation phase. Each gene, representing a CA coordinate, can vary within a variation radius $[x_{max}, y_{max}]$

may require several seconds, or even several hours: for example, on a 2-Quadcore Intel Xeon E5472, 3.00 GHz CPU such evaluation requires approximately 10 min, as at least 40,000 CA steps are required for a simulation. For instance, if the GA population is composed of 100 individuals, the time required to run one seed test (100 generation steps) exceeds 69 days. Moreover, the GA execution can grow, depending on both the extent of the considered area and the number of different tests to run.

As a consequence, a *CPU/GPU* library was developed to accelerate the GA running. Specifically, a "Master-Slave" model was adopted in which the Host-CPU (Master) executes the GA steps (selection, population replacement, mutation and crossover), while GPU cores (slaves) evaluate the individuals fitness (i.e., a complete SCIARA-fv2 simulation).

Since the most intensively computational work is needed for this latter phase, a *multi-simulator* was devised to efficiently exploit the considered GPGPU hardware, in order to permit the execution of more simulations in parallel. For this purpose, two different implementation strategies were implemented and a landscape benchmark case study was considered, modeled through a Digital Elevation Model composed of 200×318 square cells with a side of 10 m. In addition, a set of 50 hypothetical barriers placed with 2 different inclinations (135, 225°) to the lava flow direction was considered leading to a total of 100 simulations to be performed. Four CUDA devices were used in the experiments: a nVidia Tesla C2075 and three nVidia Geforce graphic cards, namely the GTX480, GTX 580 and the GTX 680. Also, in order to quantify the achieved parallel speedup, sequential versions of the same GPU strategies were run on a workstation equipped with a 2-Quadcore Intel Xeon E5472 (3.00 GHz) CPU.

Starting from previous research in CA modelling by means of GPGPU (e.g. [34–37]), a first straightforward parallel implementation, labeled as WCSI (Whole Cellular Space Implementation) was considered where the CUDA kernels operate on the whole cellular space. However, since the transition function of the currently active cells (i.e., cells containing lava) is invoked, simulating only one simulation at a time would imply a high percentage of uselessly scheduled threads. In addition, given the limited extension of most simulations (on average, 20 % of cells of the entire automaton are active during a single simulation), the number of active threads would be too low to allow the GPU to effectively activate the latency-hiding mechanism [38] of CUDA. To increase thread occupancy, in the WCSI approach more than a single lava episode are simultaneously executed. This means that the main CUDA kernel is executed over a number of simulations which are simultaneously executed at the same CA step. In particular, each simulation performed is mapped on a different value of z and on a grid of threads composed of 16×16 blocks. That is, the grid of threads used for the CA transition function is three-dimensional, with the base representing the considered CA space and the vertical dimension corresponding to the different launched simulations.

Using the adopted GPU devices, the algorithm was implemented with the WCSI approach and execution times evaluated for a variable number of simultaneous lava simulations. For a fair comparison, the sequential version of the same algorithm was used and the elapsed time achieved by the CPU was 26039 s. According to the results shown in Fig. 4a, the GTX 680 achieved the lowest elapsed time of 650,96 s, concurrently simulating 50 lava events. The gain provided by the parallelisation in terms of computing time was significant and corresponded to a parallel speedup of over 40 for the used CPU (Fig. 4).

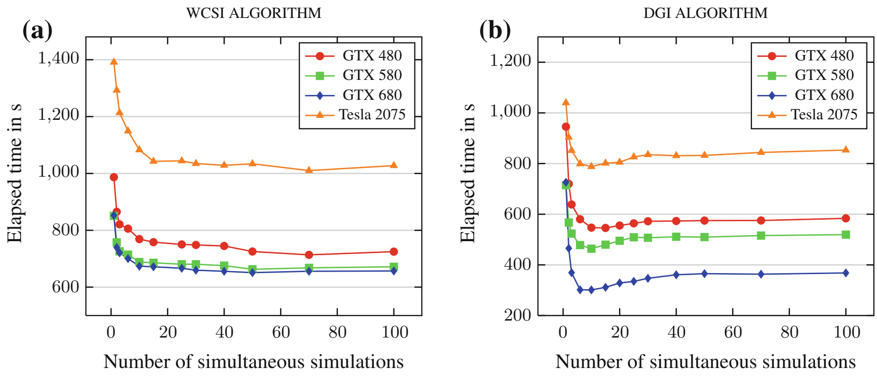
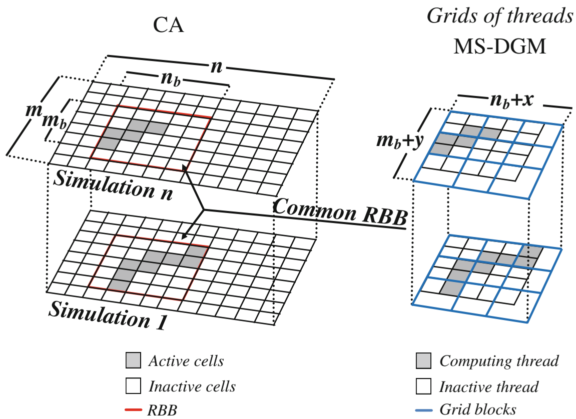


Fig. 4 Elapsed time as a function of simultaneous lava events usign the WCSI (**a**) and DGI (**b**) approaches on different considered GPGPU hardware

For CA lava flow models, the application of the transition function can be restricted to the only active cells where computation is actually taking place. Thus, the CA space can be confined within a rectangular bounding box (RBB). This optimization drastically reduces execution times, since the sub-rectangle is usually quite smaller than the original CA space. This may result in having a high percentage of computationally inactive threads in the CUDA grid, as in the case of the WCSI CA implementation. For these reasons, a second approach was developed in which the grid of threads is dynamically computed during the simulation in order to keep low the number of computationally irrelevant threads. In such an approach, labelled as DGI (Dynamic Grid Implementation), a number of lava flow simulations are simultaneously executed as in the WCSI procedure.

In addition, at each CA step the procedure involves the computation of the smallest *common* rectangular bounding box (CRBB) that includes any active cells in every concurrent simulation. Figure 5 shows all kernels required by the CA step that are

Fig. 5 Mapping of the CA transition function into a CUDA grid of threads (*right*) in case of the simultaneous lava flows (*left*)



mapped on such CRBB, thus reducing the number of useless threads and significantly improving the computational performance.

An analogous strategy based on the bounding box has been developed for the sequential version of the program. for a fair comparison. Using the reference CPU, such sequential procedure required 20180 s on the same case study. Figure 4b shows times taken by the parallel DGI approach as a function of the number of concurrent simulations. As seen, the GTX 680 achieved the lowest elapsed time of 301,18 s, giving rise to a parallel speedup of 67.

3.2 Experiment and Results

By considering the Nicolosi lava flow event (barriers uphill from Sapienza Zone) and by adopting the parallel multi-simulator described in the previous section, ten GA runs (based on different random seeds) of 100 generation steps each were carried out, each one with a different initial population. The elapsed time achieved for the ten GA runs was less than nine hours of computation on a 10 multi-GPU GTX 680 GPU Kepler Devices Cluster (note that the same experiment, on a sequential machine, would had lasted more than seven months). Furthermore, during the running, a Visualization System Software [18], based on OpenGL and C++ and integrated into Qt interface, allowed the interactive visualization and analysis phases of the results.

For this preliminary experiment, only solutions with two nodes were considered ($|W| = 1$), while Z and P were chosen as in Fig. 1. The cardinality of W (Protection work) and the gene values in which they are allowed to vary (depending of Z area), define the search space S_r for the GA:

$$S_r = \{\left[P_{x_{min}}, P_{x_{max}}\right] \times \left[P_{y_{min}}, P_{y_{max}}\right] \times \left[\left(h_{min} \cdot n_g\right), \left(h_{max} \cdot n_g\right)\right]\}^{2|W|} \quad (8)$$

The temporal evolution of the f_3 fitness is graphically reported in Fig. 7a, in terms of average results over the ten considered experiments. GA experiment parameters values are also listed in Table 1. The related CA simulation, obtained by adopting the best individual is shown in Fig. 6.

The study, though preliminary, has produced quite satisfying results. Among different best individuals generated by the GA for each seed test, the best one (Table 2) consists of a barrier with an average height of 7,5 and 410 m in length with an inclination angle of 141° with respect to the direction of the lava flow. The barrier (its properties are shown in Table 2) completely deviates the flow avoiding that the lava reaches the inhabited and building facilities areas. It is worth to note that, the best solution provided by GA (Fig. 6) in this work is approximately five times more efficient (in term of total m^3 volume used to keep safe tha safety areas) respect to the one applied in the real case (Fig. 1), consisting of thirteen earthen barriers.

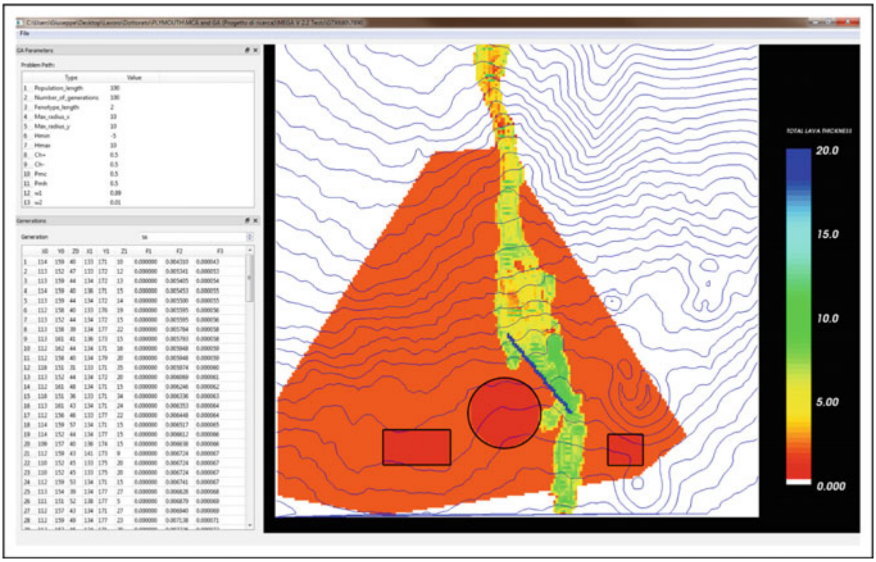


Fig. 6 SCIARA—fv2 simulation visualization adopting the GA best solution. As seen, the devised barrier (*blue*) completely diverts the lava flow from the Safety Areas (*red*)

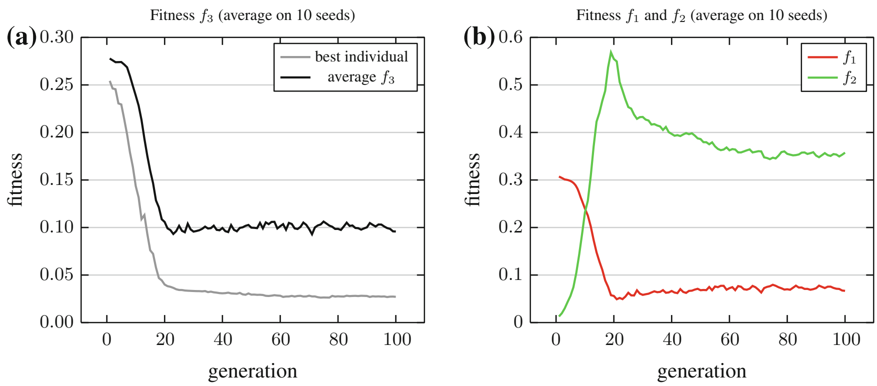


Fig. 7 Temporal evolution of composite f_3 fitness of best individual (*in black*) and of average fitness of whole population (*in gray*) (**a**). Temporal evolution of average fitness f_1 (*in red*) and f_2 (*in green*) of whole population (**b**). Fitness values were obtained as an average of 10 GA runs, carried out by adopting different seeds for generation of random numbers

3.3 Considerations on the GA Dynamics and Emergent Behaviors

In the GA experiments that have been performed, individuals with high fitness evolved rapidly, even if the initial population was randomly generated and the search

Table 2 Properties of the best barrier evolved by GA run

Barrier Propertiers	Length (m)	Height (m)	Base Width (m)	Volume (m^3)	Inclination (°)
[206,96,2] [238,122,13]	410	7,5	10	24750	141

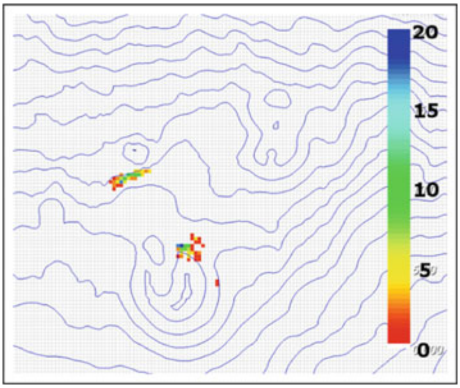


Fig. 8 Nodes distribution of the best 100 solutions generated by the GA. Scale values indicates occurrence of nodes

space was quite large (Eq. 8). By analyzing several individuals evolved in ten different GA executions, similar solutions were observed. This behavior is due to the presence of problem constraints (e.g. morphology, lava vent, emission rate, Z and P areas) that lead the GA to search in a "region" of the solution space characterized by a so called "local optimum". In particular, f_1 reaches the minimum value (0) around the twentieth GA generation and the remaining 80 runs are used by GA for the f_2 optimization (cf. Fig. 7b).

In any case, the evolutionary process has shown, in accordance with the opinion of the scientific community [5, 39], the ineffectiveness of barriers placed perpendicular to the lava flow direction despite diagonally oriented solutions (130–160°).

Furthermore, a systematic exploitation of morphological characteristics by GA, during the evolutionary process, has emerged. To better investigate such GA emergence behaviour, a study of nodes distribution was conducted (Fig. 8). By considering the best 100 solutions provided by GA, each node was classified on the basis of the *slope proximity* calculation, as an average of altitude differences between node neighborhood cells (with radius 10) and the central cell. More formally, the function that assigns to each generic node j a *slope proximity* value is defined as:

$$sp_j = \frac{\sum_{i=1}^{|X|} \bar{z}_i - \bar{z}_0}{|X|} \tag{9}$$

where X is the set of cells that identifies the neighborhood of j and $\bar{z}_i \in Q_z$ is the topographics altitude (index 0 represents the central cell). As shown in Fig. 9, starting from the tenth GA generation, the evolutionary process has shown an increase

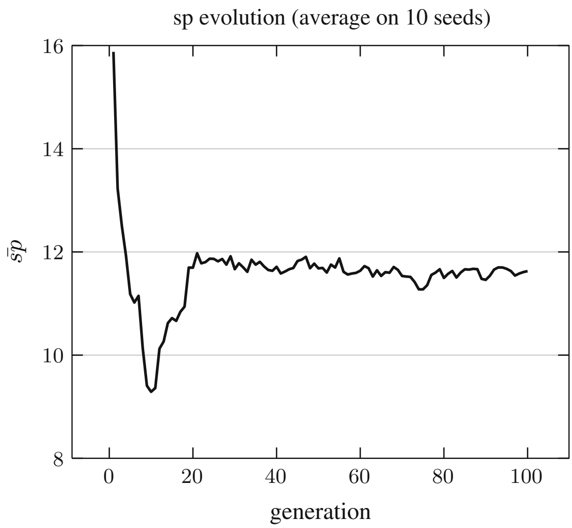


Fig. 9 Temporal evolution of average slope proximity values for the best individuals

in slope proximity values. Therefore, after the f_1 optimization (cf. Fig. 7b), in order to minimize f_2, there is a specific evolutionary temporal phase (i.e., up to the 25th generation) where the algorithm generates solutions that are located in the proximity of elevated slopes.

4 Conclusions and Future Works

This paper has presented a novel approach for devising protective measures to divert lava flows. Starting from the problem of the high computational complexity of the GA algorithm, a library was developed for executing a large number of concurrent lava simulations using GPGPU. The parallel speedups attained through the proposed approaches and by considering GPGPU hardware, were indeed significant. In fact, the adoption of PGAs permitted to perform, in reasonable times, a greater number of tests shortening the execution by a factor of 67. In addition to the GA algorithm acceleration implementation, an interaction visualization system was also developed for the analysis phases of the results.

In this preliminary release of the algorithm only two nodes based solutions were considered and evaluated on the basis of two fitness functions. The first fitness function guarantees the goodness of the solution in terms of security; the second one minimizes the environmental impact.

First observations of the GA results permitted to conjecture the presence of a local optima in the search space, probably due to problem constraints. To better investigate GA dynamic characteristics, a study of nodes distribution was also conducted and a systematic exploitation of morphological characteristics by GA during the evolutionary process emerged.

PGAs experiments, carried out by considering the Nicolosi case-study, demonstrated that artificial barriers can successfully change the direction of lava flow in order to protect predefined point of interests.

In particular, by performing extensive experiments, simulations demonstrated that protective works are more effective when placed nearly parallel to the flow direction, while a barrier placed perpendicular to the flow direction can only stop the flux temporarily, ultimately allowing the solidified crust to accumulate and cause the following mass to go over the barrier.

Though preliminary, the study has produced extremely positive results and simulations have demonstrated that GAs can represent a valid tool to determine protection works construction in order to mitigate the lava flows risk. However, considering two-nodes barriers is a strong limitation and a critical aspect of GA implementations that can significantly improve the efficiency of the final solution is the the possibility to provide multi-barrier protection measures. For this reason, future work will firstly consider the investigation of solutions consisting of multiple protective interventions and the introduction of lava cooling by water jets as parameter of the methodology. By considering the hazard evaluation context, an important application of the methodology could be to take into account as an event to be mitigated a grid of hypothetical vents defined as the source for the simulations to be carried out. In this case, protection measures provided by the GA can represent a preventive solution to assess the effect of possible human interventions. Furthermore, it could be very important to evaluate the extension of this method to other different complex natural phenomena such as a debris flow models.

Acknowledgments Authors gratefully acknowledge the support of NVIDIA Corporation for this research. The work was partially funded by the European Commission European Social Fund (ESF) and by the Regione Calabria (Italy).

References

1. Behncke, B., Neri, M.: The July-August 2001 eruption of Mt. Etna (Sicily). Bull. Volcanol. **65**(7), 461–476 (2003)
2. Miyamoto, H., Sasaki, S.: Simulating lava flows by an improved cellular automata method. Comput. Geosci. **23**, 283–292 (1997)
3. Avolio, M.V., Crisci, G.M., Di Gregorio, S., Rongo, R., Spataro, W., D'Ambrosio, D.: Pyroclastic flows modelling using Cellular Automata. Comput. Geosci. **32**, 897–911 (2006)
4. Del Negro, C., Fortuna, L., Herault, A., Vicari, A.: Simulations of the 2004 lava flow at Etna volcano using the magflow cellular automata model. Bull. Volcanol. **70**(7), 805–812 (2008)
5. Barberi, F., Brondi, F., Carapezza, M., Cavarra, L., Murgia, C.: Earthen barriers to control lava flows in the 2001 eruption of Mt. Etna. J. Volcanol. Geoth. Res. **123**, 231–243 (2003)
6. Colombrita, R.: Methodology for the construction of earth barriers to divert lava flows: the Mt. Etna 1983 eruption. Bull. Volcanol. **47**(4), 1009–1038 (1984)
7. Barberi, F., Carapezza, M., Valenza, M., Villari, L.: The control of lava flow during the 1991–1992 eruption of Mt. Etna. J. Volcanol. Geoth. Res. **56**, 1–34 (1993)

8. Bentley, P.: An introduction to evolutionary design by computers. In: Bentley, P.J. (ed.) Evolutionary Design by Computers, ch. 1, pp. 1–73. Morgan Kaufman, San Francisco (1999)
9. Sims, K.: Evolving 3d morphology and behavior by competition. In: Proceedings of Artificial Life IV, pp. 28–39. MIT Press (1994)
10. Bongard, J.: Morphological change in machines accelerates the evolution of robust behavior. In: Proceedings of the National Academy of Sciences, vol. 108, pp. 1234–1239 (2011)
11. Kicinger, R., Arciszewski, T., Jong, K.D.: Evolutionary computation and structural design: a survey of the state-of-the-art. Comput. Struct. **83**, 1943–1978 (2005)
12. Spataro, W., Avolio, M.V., Lupiano, V., Trunfio, G.A., Rongo, R., D'Ambrosio, D.: The latest release of the lava flows simulation model SCIARA: First application to Mt Etna (Italy) and solution of the anisotropic flow direction problem on an ideal surface. In: Proceedings of International Conference on Computational Science, vol. 1, pp. 17–26. Procedia Computer Science (2010)
13. Neumann, J.V.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966)
14. Chopard, B., Droz, M.: Cellular Automata Modeling of Physical Systems. Cambridge University Press, Cambridge (2000)
15. Trunfio, G.A., D'Ambrosio, D., Rongo, R., Spataro, W., Di Gregorio, S.: A new algorithm for simulating wildfire spread through cellular automata. ACM Trans. Model. Comput. Simul. **22**, 6:1–6:26 (2011)
16. Succi, S.: The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Clarendon Press, Oxford (2001)
17. Crisci, G.M., Gregorio, S.D., Rongo, R., Spataro, W.: Pyr: a cellular automata model for pyroclastic flows and application to the 1991 mt. pinatubo eruption. Future Gen. Comput. Syst. **21**(7), 1019–1032 (2005)
18. Filippone, G., D'Ambrosio Spataro, D., Marocco, D.: An interactive visualization system for lava flows cellular automata simulations using CUDA. In: Poster Presented at GPU Technology Conference. San Jose, California (2013)
19. Barberi, F., Carapezza, M.L.: Mt. Etna: Volcano Laboratory, ch. The Control of Lava Flows at Mt. Etna, pp. 357–369. American Geophysical Union, Washington (2004)
20. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The MIT Press, Cambridge (1992)
21. Goncalves, J.F., Resende, M.G.: Biased random-key genetic algorithms forcombinatorial optimization. J. Heuristics **17**(5), 487–525 (2011)
22. Hinton, G.E., Nowlan, S.J.: How learning can guide evolution. Complex Syst. pp. 495–502 (1987)
23. Nolfi, S., Marocco, D.: Evolving robots able to integrate sensory-motor information over time. Theory Biosci. **120**, 287–310 (2001)
24. ElSayed, A., Kongar, E., Gupta, S., Sobh, T.: A robotic-driven disassembly sequence generator for end-of-life electronic products. J. Intell. Rob. Syst. **68**(1), 43–52 (2012)
25. Piwonska, A., Seredynski, F., Szaban, M.: Learning cellular automata rules for binary classification problem. J. Supercomput. **63**(3), 800–815 (2013)
26. Di Gregorio, S., Serra, R., Villani, M.: Applying cellular automata to complex environmental problems: the simulation of the bioremediation of contaminated soils. Theoret. Comput. Sci. **217**(1), 131–156 (1999)
27. Iovine, G., D'Ambrosio, D., Di Gregorio, S.: Applying genetic algorithms for calibrating a hexagonal cellular automata model for the simulation of debris flows characterised by strong inertial effects, Geomorphology, vol. 66, no.14, pp. 287–303 (2005)
28. Rongo, R., Spataro, W., D'Ambrosio, D., Avolio, M.V., Trunfio, G.A., Di Gregorio, S.: Lava flow hazard evaluation through cellular automata and genetic algorithms: an application to Mt Etna volcano. Fundam. Inf. **87**, 247–267 (2008)
29. D'Ambrosio, D., Rongo, R., Spataro, W., Trunfio, G.A.: Meta-model assisted evolutionary optimization of cellular automata: an application to the sciara model. In: Proceedings of the 9th International Conference on Parallel Processing and Applied Mathematics - Volume Part II, PPAM'11, pp. 533–542. Springer, Berlin (2012)

30. D'Ambrosio, D., Rongo, R., Spataro, W., Trunfio, G.: Optimizing Cellular Automata through a Meta-model Assisted Memetic Algorithm. In: Proceedings of Parallel Problem Solving from Nature - PPSN XII, Lecture Notes in Computer Science, vol. 7492, pp. 317–326. Springer, Berlin (2012)
31. Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co. Inc., Boston (1989)
32. Mitchell, M.: An introduction to Genetic Algorithms. MIT Press, Cambridge (1996)
33. Bresenham, J.: Algorithm for computer control of a digital plotter. IBM Syst. J. **4**(1), 25–30 (1965)
34. Filippone, G., Spataro, W., Spingola, G., D'Ambrosio, D., Rongo, R., Perna, G., Di Gregorio, S.: GPGPU programming and cellular automata: Implementation of the SCIARA lava flow simulation code. In: 23rd European Modeling and simulation Symposium (WMSS), pp. 12–14. Rome, September 2011
35. Di Gregorio, S., Filippone, G., Spataro, W., Trunfio, G.A.: Accelerating wildfire susceptibility mapping through GPGPU. J. Parallel Distrib. Comput. **73**(8), 1183–1194 (2013)
36. D'Ambrosio, D., Filippone, G., Marocco, D., Rongo, R., Spataro, W.: Efficient application of gpgpu for lava flow hazard mapping. J. Supercomput. **65**(2), 630–644 (2013)
37. D'Ambrosio, D., Filippone, G., Rongo, R., Spataro, W., Trunfio, G.A.: Cellular automata and GPGPU: an application to lava flow modeling. Int. J. Grid High Perform. Comput. **4**, 30–47 (2012)
38. NVIDIA Corporation, CUDA C Best Practices Guide. NVIDIA Corporation, 2701 San Tomas Expressway, Santa Clara 95050, USA, 5.0 ed. (2012)
39. Fujita, E., Hidaka, M., Goto, A., Umino, S.: Simulations of measures to control lava flows. Bulletin of Volcanology **71**, 401–408 (2009)