Evolutionary Applications to Cellular Automata Models for Volcano Risk Mitigation

Giuseppe Filippone[1], Roberto Parise[1], Davide Spataro[1],
Donato D'Ambrosio[1,2], Rocco Rongo[1,2], and William Spataro[1,2](✉)

[1] Department of Mathematics and Computer Science, University of Calabria,
Via Pietro Bucci, 87036 Rende, Italy
spataro@unical.it
[2] CNR-IRPI, Sezione di Cosenza, Rende, Italy

Abstract. A GPGPU accelerated evolutionary computation-based decision support system for defining and optimizing volcanic hazard mitigation interventions is proposed. Specifically, the new Cellular Automata numerical model SCIARA-fv3 for simulating lava flows at Mt Etna (Italy) and Parallel Genetic Algorithms (PGA) have been applied for optimizing protective measures construction by morphological evolution. A case study is considered, where PGA are applied for the optimization of the position, orientation and extension of earth barriers built to protect a touristic facility located near the summit of Mt. Etna (Italy) volcano which was interested by the 2001 lava eruption. The methodology has produced extremely positive results and, in our opinion, can be applied within a broader risk assessment framework, having immediate and far reaching implications both in land use and civil defense planning.

Keywords: Evolutionary computation · Parallel genetic algorithms · Decision support system · Cellular automata · Morphological evolution

1 Introduction

In the modelling and simulation field, Complex Cellular Automata (CCA) can represent a valid methodology to model numerous complex non-linear phenomena [1], such as lava and debris flows. CCA are an extension of classical Cellular Automata (CA) [2], developed for overcoming some of the limitations affecting conventional CA frames such as the modelling of large scale complex phenomena. Due to their particular nature and local dynamics, CCA are very powerful in dealing with complex boundaries, incorporating microscopic interactions and easy parallelization of algorithms.

In lava risk mitigation, the building of artificial barriers [3,4] is fundamental for controlling and slowing down the destructive effects of flows in volcanic areas. Nevertheless, the proper positioning of protective measures in a specific 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.

C. Pizzuti and G. Spezzano (Eds.): WIVACE 2014, CCIS 445, pp. 99–112, 2014.
DOI: 10.1007/978-3-319-12745-3_9

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 integrates concepts from evolutionary algorithms, engineering and complex systems to solve engineering design problems [5]. Morphological evolution has also been largely explored in evolutionary robotics, for instance in the design of imaginary 3D robotics bodies [6].

Genetic Algorithms (GAs) [7] are general-purpose iterative search algorithms inspired by natural selection and genetics and have been applied several times in the past for optimizing CCA models (e.g., [8,9]). This work describes the application of morphological evolution by Parallel Genetic Algorithms (PGAs) for optimizing earth barriers construction to divert a case study lava flow, occurred in Mt. Etna in 2001. The GA fitness function, adopted for evaluating the "goodness" of the protective works deviating lava flow scenarios generated by the new SCIARA-fv3 CA lava flow model [10], has implied a massive use of the numerical simulator, consisting in thousands of concurrent simulations for every GA generation. Therefore, a GPGPU (General Purpose computation with Graphic Processor Units) library was developed to accelerate the GA execution.

After a brief description of the new SCIARA-fv3 CA model adopted in experiments (Sect. 2), the main characteristics of the implemented evolutionary algorithm and carried out experiments, together with reference to emergent behaviors, are presented in Sect. 3. The developed Web user interface for interactive visualization of results are described in Sect. 4. Eventually, Sect. 5 concludes the paper with final comments and future works.

2 Complex Cellular Automata and the SCIARA-fv3 Lava Flow Model

As previously stated, CCA represent an extension of the classical homogeneous CA, particularly useful for the modeling of spatially extended systems. Formally, a CCA is a 7-tuple:

$$A =< R, X, Q, P, \tau, L, \gamma >$$

where R, X, Q and τ are as in the homogeneous CA definition, respectively defining the CA space, cell neighborhood, set of states and the deterministic transition function applied to each cell, simultaneously and at discrete steps. However, in the CCA frame, the set Q of state of the cell is decomposed in substates, $Q_1, Q_2, \ldots, Q_r$, each one representing a particular feature of the phenomenon to be modelled. The overall state of the cell is thus obtained as the Cartesian product of the considered substates: $Q = Q_1 \times Q_2 \times \ldots \times Q_r$. A set of parameters, $P = p_1, p_2, \ldots, p_p$, is furthermore considered. These allow to calibrate the model for reproducing different dynamics. As the set of states is split in substates, also the transition function, τ, is split in elementary processes, $\tau_1, \tau_2, \ldots, \tau_s$, each one describing a particular aspect that rules the dynamics of the considered phenomenon. Eventually, $L \in R$ is a subset of the cellular space that is subject to external influences, as specified by the supplementary function γ. External influences are introduced in order to model features that are not easy to be described in terms of "local interactions".

2.1 The SCIARA-fv3 Lava Flow Model

SCIARA-fv3 is the latest release of the SCIARA family of Complex Cellular Automata Models for simulating lava flows. As its predecessor, SCIARA-fv2 [11], it is based on a Bingham-like rheology. While more specific details can be found in [10], we briefly describe the model in the following. In formal terms, SCIARA-fv3 is defined as:

$$SCIARA - fv3 =< R, X, Q, P, \tau, L, \gamma >$$

where:

1. R is the cellular space, i.e. the set of square cells covering the bi-dimensional finite region where the phenomenon evolves;
2. X is the pattern of cells belonging to the adjacent eight-cell Moore neighborhood that influence the cell state change;
3. $Q = Q_z \times Q_h \times Q_T \times Q_{\overrightarrow{p}} \times Q_f^9 \times Q_{\overrightarrow{v_f}}^9$ is the finite set of states, considered as Cartesian product of substates. Their meanings are: cell altitude a.s.l., cell lava thickness, cell lava temperature, momentum (both x and y components), lava thickness outflows (from the central cell toward the adjacent cells) and flows velocities (both x and y components), respectively;
4. $P = w, t_0, P_T, P_d, P_{hc}, \delta, \rho, \epsilon, \sigma, c_v$ is the finite set of parameters (invariant in time and space), whose meaning can be found in [10];
5. $\tau : Q^9 \rightarrow Q$ is the cell deterministic transition function; it is split in "elementary processes", which are described in the following sections;
6. $L \subseteq R$ specifies the lava source cells (i.e. craters);
7. $\gamma : Q_h \times \mathbb{N} \rightarrow Q_h$ specifies the emitted lava thickness from the source cells at each step $k \in \mathbb{N}$.

2.2 Elementary Process τ_1: Lava Flows Computation

The elementary process τ_1 computes lava outflows and their velocities, formally defined as:

$$\tau_1 : Q_z^9 \times Q_h^9 \times Q_{\overrightarrow{p}} \rightarrow Q_f^9 \times Q_{\overrightarrow{v_f}}^9$$

Lava flows are computed by a two-step process: the first computes the CA clock, t, i.e. the physical time corresponding to a CA computational step, while the second the effective lava outflows, $h_{(0,i)}$, their velocities $v_{f_{(0,i)}}$ and displacements $s_{(0,i)}$ $(i = 0, 1, \ldots, 8)$. The elementary process τ_1 is thus executed two times, the first one in "time evaluation mode", the second in "flow computing mode". Both modes compute the so called "minimizing outflows", $\phi_{(0,i)}$, i.e. those which minimize the unbalance conditions within the neighborhood, besides their final velocities and displacements. In "time evaluation mode", t is preliminary set to a large value, $t_{\max}$, and the computed displacement, $s_{(0,i)}$, is compared with the maximum allowed value, $d_{(0,i)}$, which is set to the distance between the central cell and the neighbor that receives the flow. In case of over-displacement, the time t must be opportunely reduced in order to avoid the overflow condition (i.e., a flow going beyond the cell neighbourhood). In case no over-displacement are obtained, t remains unchanged. Eventually, in "flow computing mode", effective lava outflows, $h_{(0,i)}$, are computed by adopting the CA clock obtained in "time evaluation mode", by guarantying no overflow condition.

Outflows Computation. In "flow computing mode", minimizing outflows, $\phi_{(0,i)}$, are re-computed by considering the new CA clock $\bar{t}$. Subsequently, lava outflows, $h_{(0,i)}$, are computed proportionally to the displacement, by simply multiplying the minimizing outflow by the ratio between the actual displacement, $s_{(0,i)}$, and the maximum allowed, the cell side w:

$$h_{(0,i)} = \phi_{(0,i)} \frac{s_{(0,i)}}{w}$$

2.3 Elementary Process τ_2: Updating of Mass and Momentum

The elementary process updates lava thickness and momentum. It is formally defined as:

$$\tau_2 : Q_f^9 \times Q_{\overrightarrow{v_f}}^9 \rightarrow Q_h \times Q_{\overrightarrow{p}}$$

Once the outflows $h_{(0,i)}$ are known for each cell $c \in R$, the new lava thickness inside the cell can be obtained by considering the mass balance between inflows and outflows:

$$h_{(0)} = \sum_{i=0}^{9} (h_{(i,0)} - h_{(0,i)})$$

Moreover, also the new value for the momentum can be updated by accumulating the contributions given by the inflows:

$$\overrightarrow{p}_{(0)} = \sum_{i=0}^{9} h_{(i,0)} \overrightarrow{v_f}_{(i,0)}$$

2.4 Elementary Process τ_3: Temperature Variation and Lava Solidification

$$\tau_3 : Q_f^9 \times Q_T^9 \rightarrow Q_T \times Q_h$$

As in the elementary process τ_1, a two step process determines the new cell lava temperature. In the first one, a temperature $\overline{T}$ is obtained as weighted average of residual lava inside the cell and lava inflows from neighboring ones. A further step updates the calculated temperature $\overline{T}$ by considering thermal energy loss due to lava surface radiation:

$$T = \frac{\overline{T}}{\sqrt[3]{1 + \frac{3\overline{T}^3 \epsilon\sigma\bar{t}\delta}{\rho c_v w^2 h}}}$$

where ϵ, σ, $\bar{t}$, δ, ρ, c_v, w and h are the lava emissivity, the Stephan-Boltzmann constant, the CA clock, the cooling parameter, the lava density, the specific heat, the cell side and the lava thickness, respectively. When the lava temperature drops below the threshold T_{sol}, lava solidifies. Consequently, the cell altitude increases by an amount equal to lava thickness and new lava thickness is set to zero. For more details on all above models specifications, please refer to [10].

3 Morphological Evolution of Protective Works

Genetic Algorithms (GAs) [7] are general-purpose iterative search algorithms inspired by natural selection and genetics, and have been extensively applied in many scientific fields (e.g., [12–14]). 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 whose elements are called genes. To determine the best possible solution of a given problem, members of the initial population are evaluated by means of a "fitness function", determining the individuals "adaptivity" value. 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.

In this work, GAs were adopted in conjunction with the SCIARA-fv3 CA model for the morphological evolution of protective works to control lava flows in the Rifugio Sapienza area, which was indeed interested by the 2001 Nicolosi case study and object of a set of real mitigation interventions construnction [15].

3.1 Parallel Genetic Algorithm Definition

The CA 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

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

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

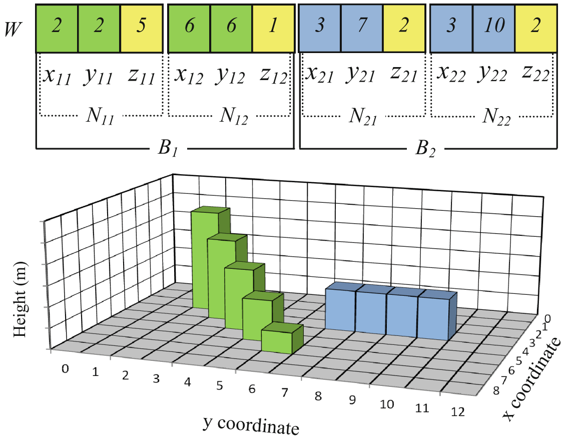


Fig. 1. Example of barriers encoding into a GA genotype. The height of the intermediate points of each barrier is obtained by connecting the work protections extremes through a linear function.

$N_{ij} = x_{ij}, y_{ij}, z_{ij}$, where x_{ij}, y_{ij} are the CA coordinates for the generic node j of the barrier i, and z_{ij} the height (in m). The solutions were encoded into a GA genotype as integer values (Fig. 1) 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:

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

where S and Z respectively identify the areal extent of the simulated lava event and the Safety Zone area, with $\mu(S \cap Z)$ being the measure of their intersection. 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 i-th barrier, respectively. The parameter p_c is the CA 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, the determination of the cells inside the segment between the work protection extremes and f_2 subsequently are computed.
2. A SCIARA-fv3 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. 2).

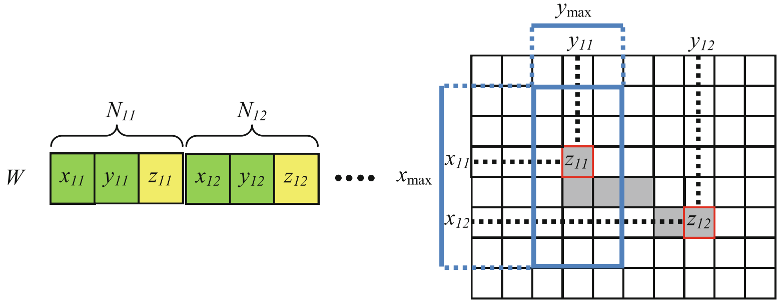


Fig. 2. Graphical representation of the genotype mutation phase. Each gene, representing a CA coordinate, can vary within a variation radius $[x_{max}, y_{max}]$.

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 also introduced. Two parent individuals are randomly chosen from the mating pool and two different cutting points for each parents are selected. After the selection portions 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.

3.2 Experiments and Results

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 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-fv3 simulation). Please refer to [16–18] for more details on the different GPGPU parallel implementations which are adopted for speeding up the GA running.

By considering the Nicolosi lava flow event (barriers uphill from Sapienza Zone), ten parallel GA runs (based on different random seeds) of 100 generation steps 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

Table 1. List of parameters of the adopted GA

GA parameters	Specification	Value
g_l	Genotype's length	6
p_s	Population size	100
n_g	Number of generations	100
p_{mc}	Coord. gene mutation probability	0.4
x_{max}	Gene x position variation radius	10
y_{max}	Gene y position variation radius	10
p_{mh}	Height gene mutation probability	0.45
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	0.95
ω_{f2}	f_2 weight parameter	0.05

Table 2. Properties of the best barrier evolved by GA run

Barrier	Length	Height	Base width	Volume	Inclination
	(m)	(m)	(m)	(m^3)	(degrees)
$[210, 94, 3]$ $[236, 124, 12]$	397	7.5	10	24750	131

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 new 3D WebGL visualization system (discussed in Sect. 4) was developed, making the model fully portable and allowing the interactive visualization and analysis phases of the results.

For this experiment, only solutions with two nodes were considered ($|W| = 1$), while Z and P were chosen as in Fig. 5. 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:

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

The temporal evolution of the f_3 fitness is graphically reported in Fig. 3(a), 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. 5.

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 m and 397 m in

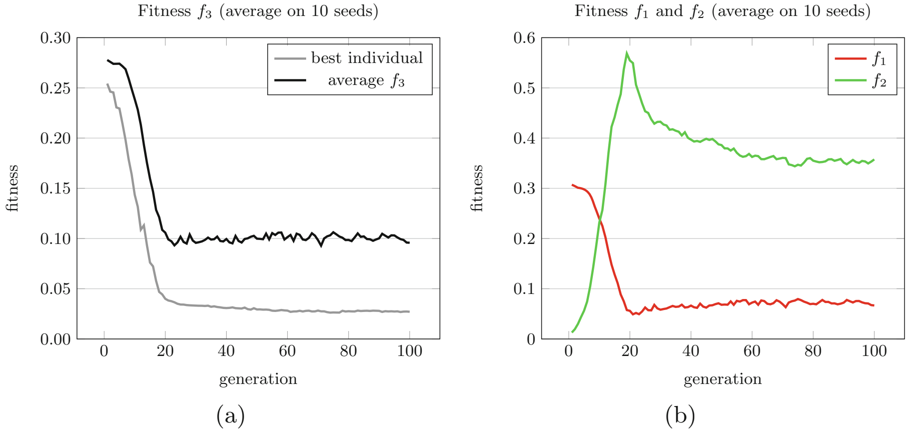


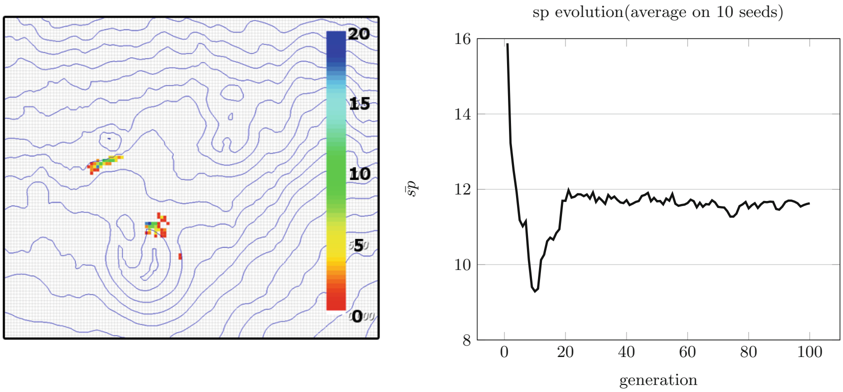
Fig. 3. Temporal evolution of composite f_3 fitness of best individual (in black) and of average fitness of whole population (in gray) (a). Temporal evolution of average fitness f_1 (in red) and f_2 (in green) of whole population (b). Fitness values were obtained as an average of 10 GA runs, adopting different seeds for generation of random numbers (Color figure online).

length with an inclination angle of 131° with respect to the direction of the lava flow. The barrier (cf. Table 2) completely deviates the flow avoiding that the lava reaches the inhabited and building facilities areas. The best solution provided by GA (Fig. 5) 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 [15], consisting of thirteen earthen barriers.

3.3 Considerations on the GA Dynamics and Emergent Behaviors

In the executed GA experiments, individuals with high fitness evolved rapidly, even if the initial population was randomly generated and the search 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. 3 (b)). In any case, the evolutionary process has shown, in accordance with the opinion of the scientific community [3,19], 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. 4(a)). By considering the solutions provided by GA, each node was classified on the



(a) (b)

Fig. 4. (a) Nodes distribution of the best 100 solutions generated by the GA. Scale values indicates occurrence of nodes. (b) Temporal evolution of average slope proximity values for the best individuals.

slope proximity calculation, as an average of altitude differences between node neighborhood cells (with radius 10 cells) and the central cell. The function that assigns to each generic node j a *slope proximity* value is defined as:

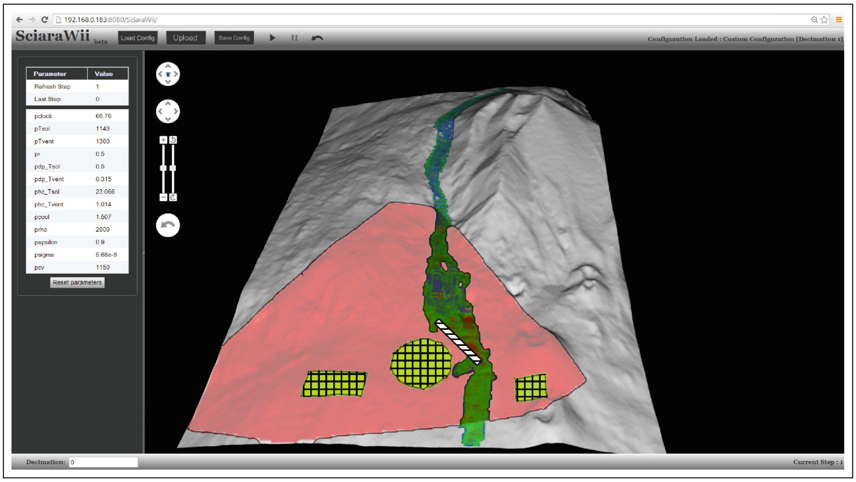
$$sp_j = \frac{\sum_{i=1}^{|X|} \bar{z}_i - \bar{z}_0}{|X|} \tag{9}$$

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. 4(b), starting from the tenth GA generation, the evolutionary process has shown an increase in slope proximity values. Therefore, after the f_1 optimization (cf. Fig. 3(b)), in order to minimize f_2, there is a specific evolutionary temporal phase (i.e., up to the 25^{th} generation) where the algorithm generates solutions that are located in the proximity of elevated slopes.

4 SciaraWii: The SCIARA-fv3 Web User Interface

The simulation results were visualized in real-time by means of a 3D interactive visualization system based on WebGL, a cross-platform application program interface used to create 3D computer graphics in Web browsers. SCIARAWii, a Web 2.0 application, controls the simulation while the SCIARA-fv3 model runs server-side. The application is based on HTML5 and JavaScript, which permits its full portability. The client is able to control the basic SCIARA-fv3 simulation functionalities thanks to asynchronous callbacks to the server.

SCIARAWii was implemented by means GWT where the interaction between the user interface and the SCIARA-fv3 computational model is performed by a set of client-side services which are implemented on the server. Multi-client



Configuration Loaded : Custom Configuration [Decimation 1]

Parameter	Value
Refresh Step	1
Last Step	0
pclock	68.76
pTsol	1143
pTvent	1360
pr	0.5
pdp_Tsol	0.5
pdp_Tvent	0.315
phc_Tsol	23.066
phc_Tvent	1.014
pcool	1.507
prho	2600
pepsilon	0.9
psigma	5.68e-8
pcv	1150

Current Step : 1

Fig. 5. A screenshot of the Web user interface for SCIARA-fv3 showing 3D simulation of 2001 Nicolosi lava flow adopting the GA best solution of the best solution. As seen, the devised barrier (dashed line area) completely diverts the lava flow from the considered Safety Areas (gridded areas).

connections are also possible: whenever a user logs in, an asynchronous request is sent to the server in order to establish a connection. Here, a servlet binds the client to an individual connection-handler, which allows multiple unambiguous communications through HTTP requests and responses.

The SCIARAWii system architecture is the same as [20]. The computational model, SCIARA-fv3, is implemented on the server in C++ (for efficiency reasons) as a static library. A dynamic-link library (DLL) receives requests by Java Native Interface (JNI) methods and provides simulation data to the application server. Data is therefore sent to the client via HTTP and stored into the Web browser cache memory. SCIARA-fv3 parameters are displayed in GUI controls (in which they can also be modified), while simulation data such as the topographic surface or the simulated lava flow, are visualized by means of the 3D WebGL rendering engine, which runs on a HTML5 `<Canvas>`. Whenever the simulation produces a lava flow, it is displayed over the surface and its dynamical behavior can be observed. All the client-server communications are managed by means of asynchronous JavaScript calls, which are able to provide the same usability level of desktop applications. Figure 5 shows a screenshot of SCIARAWii.

In order to stress the system's reliability, SCIARAWii was tested on a Local Area Network by only considering laptops, with one acting as a Web server and a maximum of 4 as remote clients accessing simultaneously the former. The level of usability of the GUI resulted more than satisfactory, mainly thanks to the asynchronous communications between client and server. Also the 3D visualization system resulted to be surprisingly efficient, especially if compared with that of the first release of SCIARAWii, by making it comparable to standard desktop applications in terms of both efficiency and usability.

5 Conclusions and Future Works

This paper has presented an evolutionary approach for devising protective measures to divert lava flows. Starting from the adoption of the latest release of the SCIARA-fv3 Cellular Automata lava model and adoption of Parallel Genetic Algorithms, a library was developed for executing a large number of concurrent lava simulations using GPGPU and a new WebGL-based 3D visualization system implemented for the real-time result analysis.

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

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. Future work will consider the investigation of solutions consisting of multiple protective interventions and the SCIARAWii visualization system graphical enrichment to fully exploit the capability of the underlying computational model.

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

References

1. Di Gregorio, S., Serra, R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. Future Gener. Comput. Syst. **16**(2–3), 259–271 (1999)
2. Von Neumann, J.: Theory Self-reproducing Automata. University of Illinois Press, Champaign (1966)
3. 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)
4. 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)
5. Bentley, P.: An introduction to evolutionary design by computers (chap. 1). In: Bentley, P.J. (ed.) Evolutionary Design by Computers, pp. 1–73. Morgan Kaufman, San Francisco (1999)

6. Sims, K.: Evolving 3D morphology and behavior by competition. In: Proceedings of Artificial Life IV, pp. 28–39. MIT Press (1994)
7. 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)
8. D'Ambrosio, D., Spataro, W.: Parallel evolutionary modelling of geological processes. J. Parallel Comput. **33**(3), 186–212 (2007)
9. D'Ambrosio, D., Rongo, R., Spataro, W., Trunfio, G.A.: Optimizing cellular automata through a meta-model assisted memetic algorithm. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part II. LNCS, vol. 7492, pp. 317–326. Springer, Heidelberg (2012)
10. D'Ambrosio, D., Spataro, W., Parise, R., Rongo, R., Filippone, G., Spataro, D., Iovine, G., Marocco, D.: Lava flow modeling by the sciara-fv3 parallel numerical code. In: 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 330–338 (2014)
11. Radu, V.: Application. In: Radu, V. (ed.) Stochastic Modeling of Thermal Fatigue Crack Growth. ACM, vol. 1, pp. 63–70. Springer, Heidelberg (2015)
12. Hinton, G.E., Nowlan, S.J.: How learning can guide evolution. Complex Syst. **1**, 495–502 (1987)
13. Peters, J.F.: Topology of digital images: basic ingredients. In: Peters, J.F. (ed.) Topology of Digital Images. ISRL, vol. 63, pp. 1–76. Springer, Heidelberg (2014)
14. 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: Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J. (eds.) PPAM 2011, Part II. LNCS, vol. 7204, pp. 533–542. Springer, Heidelberg (2012)
15. Barberi, F., Carapezza, M.L.: The control of lava flows at Mt. Etna. In: Bonaccorso, A., Calvari, S., Coltelli, M., Del Negro, C., Falsaperla, S. (eds.) Mt. Etna: Volcano Laboratory, 357th edn, p. 369. American Geophysical Union, Washington, D.C. (2004)
16. Blecic, I., Cecchini, A., Trunfio, G.: Cellular automata simulation of urban dynamics through GPGPU. J. Supercomput. **65**, 614–629 (2013)
17. 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)
18. 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)
19. Fujita, E., Hidaka, M., Goto, A., Umino, S.: Simulations of measures to control lava flows. Bull. Volcanol. **71**, 401–408 (2009)
20. Parise, R., D'Ambrosio, D., Spingola, G., Filippone, G., Rongo, R., Trunfio, G.A., Spataro, W.: Swii2, a HTML5/WebGL application for cellular automata debris flows simulation. In: Sirakoulis, G.C., Bandini, S. (eds.) ACRI 2012. LNCS, vol. 7495, pp. 444–453. Springer, Heidelberg (2012)