A New Methodology for Mitigation of Lava Flow Invasion Hazard
Morphological Evolution of Protective Works by Parallel Genetic Algorithms

G. Filippone[1], W. Spataro[1], D. D'Ambrosio[1] and D. Marocco[2]

[1]*Department of Mathematics and High-Performance Computing Center, University of Calabria, Arcavacata di Rende, Italy*

[2]*Centre of Robotics and Neural Systems, School of Computing and Mathematics, Plymouth University, Plymouth, U.K.*

Keywords: Evolutionary Computation, Genetic Algorithms, Parallel Computing, Cellular Automata, Lava Flow Simulation, Lava Flow Control.

Abstract: The determination of areas exposed to be interested by new eruptive events in volcanic regions is crucial for diminishing consequences in terms of human causalities and damages of material properties. Nevertheless, urbanized areas, cultural heritage sites or even important infrastructures, such as power plants, hospitals and schools can be protected by diverting the flow towards lower interest regions. This paper describes 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 (Sicily, Italy). In particular, the application regards the optimization of the position, orientation and extension of an earth barrier built to protect Rifugio Sapienza, a touristic facility located near the summit of the volcano. The study has produced extremely positive results and represents, to our knowledge, the first application of morphological evolution for lava flow mitigation. Among different alternatives generated by the Genetic Algorithm, an interesting scenario consists of an earthen barrier solution (with a length of 225 m, average height of 25 m, base width of 10 m and volume of 56180 m^3) which completely deviates the flow avoiding that the lava reaches the inhabited area.

1 INTRODUCTION

Among several approaches for modeling natural complex phenomena proposed in the literature, Cellular Automata (CA) represent a possible solution when the phenomena to be simulated evolve on the basis of local interactions of their constituent parts. CA are dynamical systems, discrete in space and time. They can be thought of as a lattice of cells, each one embedding an identical finite automaton, interacting only with a small set of neighbouring cells. The state of each finite automaton is changed by applying the transition function, which defines local rules of evolution for the cell. The CA overall global dynamic emerges from the simultaneous application of the local rules to each cell.

In the specific case of simulating macroscopic natural events, Macroscopic Cellular Automata (MCA) can represent a valid methodology to model numerous complex non-linear phenomena (Di Gregorio and Serra, 1999), such as lava and debris flows. MCA are an extension of classical CA, developed for overcoming some of the limitations affecting conventional CA frames such as the modeling of large scale complex phenomena. Due to its particulate nature and local dynamics, MCA are very powerful in dealing with complex boundaries, incorporating microscopic interactions and parallelization of the algorithm.

In the risk assessment of lava flow context, 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 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 occured in the last four centuries report damage to human properties in numerous towns on the volcano flanks (Behncke and Neri, 2003). Notwithstanding, the susceptibility of the Etnean area to lava invasion has increased in last decades due to continued urbanization (Dibben, 2008), the inevitable consequence being that new eruptions may involve even greater risks. Current efforts for hazard evaluation and con-

Filippone G., Spataro W., D'Ambrosio D. and Marocco D..
A New Methodology for Mitigation of Lava Flow Invasion Hazard - Morphological Evolution of Protective Works by Parallel Genetic Algorithms.
DOI: 10.5220/0004540400130024
In *Proceedings of the 5th International Joint Conference on Computational Intelligence* (ECTA-2013), pages 13-24
ISBN: 978-989-8565-77-8

tingency planning in volcanic areas draw heavily on hazard maps and numerical simulations for the purpose of individuating affected areas in advance. Although many computational modeling methods (Ishihara et al., 1990; Miyamoto and Sasaki, 1997; Avolio et al., 2006; Del Negro et al., 2008) 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 (Barberi et al., 2003; Colombrita, 1984; MacDonald, 1962) or dams (Barberi et al., 1993), 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 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 is a multidisciplinary endeavour that integrates concepts from evolutionary algorithms, engineering and complex systems to solve engineering design problems (Bentley, 1999). Morphological evolution has been largely explored in evolutionary robotics, both for the design of imaginary 3D robotics bodies (Sims, 1994) and for the efficient and autonomous design of adaptive moving robots (Bongard, 2011; Lipson and Pollack, 2000). 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 (Kicinger et al., 2005).

This paper describes the application of Parallel Genetic Algorithms (PGAs), for the first time to our knowledge, for optimizing earth barriers construction by morphological evolution, to divert a case study lava flow that is simulated by a CA model. The GA fitness evaluation has implied a massive use of the numerical simulator running thousands of concurrent simulations for every generation computation. Therefore, a GPGPU (General Purpose computation with Graphic Processor Units) library was developed to accelerate the GA execution. A visualization system (Filippone et al., 2013) was also implemented, thereby allowing interactive analysis of the results. A study of GA dynamics, with reference to emergent behaviors, is also discussed later. The paper is organized as follows: after a brief description of the SCIARA-fv2 CA model (Section 2) that was used for lava flow simulation of the case study adopted for the experiments (Section 3), the main characteristics of the implemented algorithm, framework and results are presented (Section 4). Section 5 concludes the paper with final comments and future works.

2 THE CELLULAR AUTOMATA MODEL SCIARA - $fv2$

For the morphological evolution of protective works by PGAs, the latest release fv2 of the SCIARA CA model for simulating lava flows was adopted. SCIARA is a family of bi-dimensional MCA lava flow models, successfully applied to the simulation of many real cases such as the 2001 Mt. Etna (Italy) Nicolosi lava flow (Crisci et al., 2004) and the 1991 Valle del Bove (Italy) lava event (Barca et al., 1994), which occurred on the same volcano and was employed for risk mitigation. In formal terms, the SCIARA-fv2 model (Spataro et al., 2010) is defined as:

$$SCIARA - fv2 = < R, L, X, Q, P, \tau, \gamma > \quad (1)$$

where:

- R is the set of square cells covering the bi-dimensional finite region where the phenomenon evolves;
- $L \subset R$ specifies the lava source cells (i.e. craters);
- $X = \{(0,0),(0,1),(-1,0),(1,0),(0,-1),(-1,1),(-1,-1),(1,-1),(1,1)\}$ is the set that identifies the pattern of 8 cells influencing the cell state change (i.e., the Moore neighborhood);
- $Q = Q_z \times Q_h \times Q_T \times Q_f^8$ is the finite set of states, considered as a Cartesian product of substates. SCIARA-fv2 substates, which describe relevant quantities representing a particular feature of the phenomenon to be modeled are: cell elevation a.s.l., cell lava thickness, cell lava temperature, and lava thickness outflows from the central cell toward neighbours, respectively;
- $P = \{w, t, T_{sol}, T_{vent}, r_{Tsol}, r_{Tvent}, hc_{Tsol}, hc_{Tvent}, \delta, \rho, \varepsilon, \sigma, c_v\}$ is the finite set of parameters (invariant in time and space) which affect the transition function (refer to (Spataro et al., 2010) for their specifications);

- $\tau : Q^9 \rightarrow Q$ is the cell deterministic transition function, applied to each cell at each time step, which describes the dynamics of lava flows, such as cooling, solidification and lava outflows from the central cell towards neighbouring ones.
- $\gamma : Q_h \times N \rightarrow Q_h$ specifies the emitted lava thickness, h, from the source cells at each step $k \in N$ (N is the set of natural numbers).

3 THE CASE STUDY: THE 2001 Mt Etna ERUPTION

As reported in (Barberi and Carapezza, 2004), 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 m, 2550 m, 2600 m, 2700 m, 2950 m, 3050 m, the latter four being directly connected to the conduit of the SE crater (see Figure 1). Only lava flows emitted from the lowermost vents (2100 m, 2550 m, 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 (Behncke and Neri, 2003)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-m 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 Figure 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 (Figure 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. The plan included: (a) the building of lateral barriers to drive the flow toward large depressions of old quarries, located near the lava trajectory, (b) the erection of other barriers orthogonal to the flow direction to increase the retaining capacity of the quarries and (c) the establishment of a detailed preparedness plan for the Nicolosi and Belpasso populations, mobile goods and cultural heritage. Diversion of the flow from its natural channel near the 2100-m vent, using the technique employed in 1992 in Valle del Bove (Barberi et al., 1993), was also considered, as a possible last emergency measure to be adopted should the lava have approached the villages. The decrease in the rate of effusion beginning in the last days of July prevented any further advance of the flow and so, fortunately, the planned interventions were not necessary.

4 MORPHOLOGICAL EVOLUTION OF PROTECTIVE WORKS THROUGH PARALLEL GENETIC ALGORITHMS

Genetic Algorithms (GAs) (Holland, 1992) are general-purpose iterative search algorithms inspired by natural selection and genetics. They have been applied, with good results, in many different fields: for the solution of difficult combinatorial problems (Goncalves and Resende, 2011) in the study of the interaction between evolution and learning (Hinton and Nowlan, 1987); in evolutionary robotics (Nolfi and Marocco, 2001; ElSayed et al., 2012). GAs have also been used for improving the performance of CA in resolving difficult computational tasks: (Piwonska et al., 2013) applied GAs to a cellular automata-based solution of a binary classification problem; (Tomassini and Venzi, 2002) evolved asynchronous CA to face similar problems. GAs based methods have also been applied to CA for modelling bioremediation of contaminated soils (Di Gregorio et al., 1999) and for the optimisation of lava and debris flow simulation models (e.g., (Spataro et al., 2004; Iovine et al., 2005; Rongo et al., 2008; D'Ambrosio et al., 2012c; D'Ambrosio et al., 2012b)).

GAs simulate the evolution of a population of candidate solutions, called phenotypes, to a specific prob-

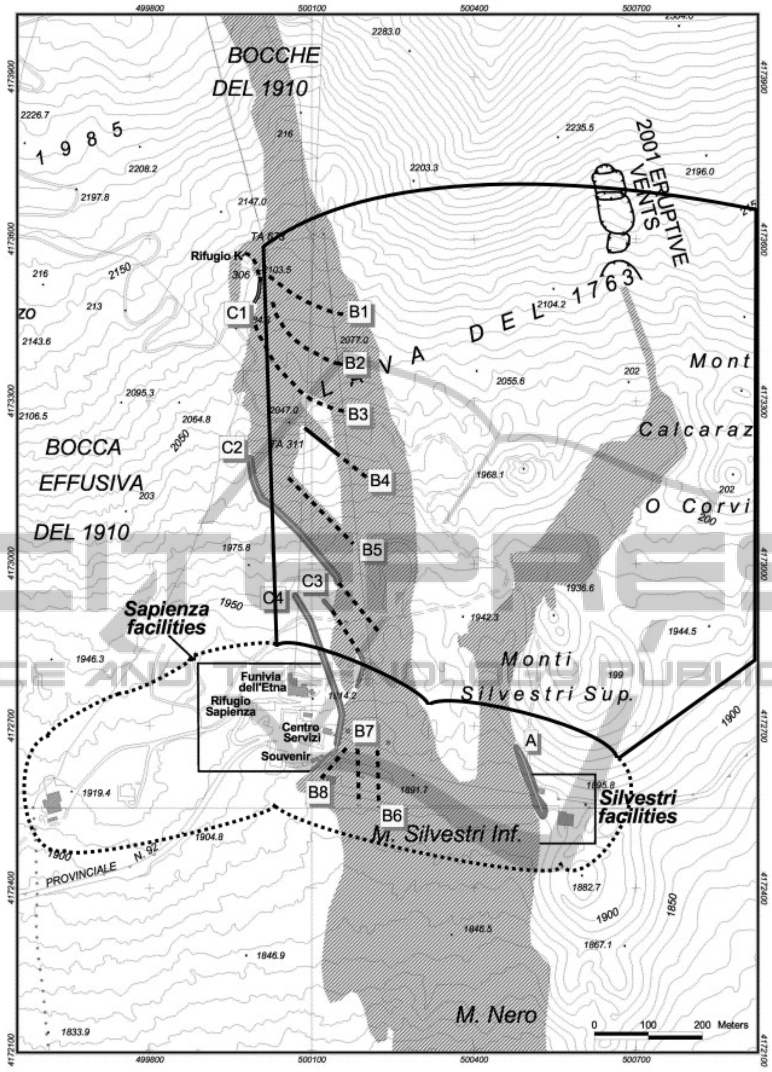


Figure 1: Set of interventions carried out during the 2001 eruption event to divert the lava flow away from the facilities. The dashed perimeter represents the Rifugio (security area), which delimitates the area that has to be protected by the flow for the study. The solid line perimeter (work area), specifies the area in which the earth barrier can be located (Base figure taken from (Barberi et al., 2003)).

lem 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. For example, if the genotype codifies real numbers, the search space is defined as the Cartesian product of the ranges in which they are allowed to vary. The members of the initial population are usually randomly generated and successively evaluated by means of a “fitness function”. This latter determines the individuals “adaptivity” 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. According to a prefixed probability, pairs of genotypes produce offspring via crossover, receiving components from each parent. Then, according to a prefixed and usually small probability, each gene of the offspring is subject to mutation. The mutation operator simply changes the gene value with another one, randomly chosen within those allowed. 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. The basic idea is that better individuals (i.e. characterised by higher fitness) can be obtained over time by combining partial solutions (i.e. portions of the genotypes) and by randomly changing the genes values. 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 (Holland, 1992). For a complete overview of GAs, see (Goldberg, 1989) and (Mitchell, 1996).

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:

$$Z \subseteq R \tag{2}$$

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

$$P \subseteq R, P \cap Z = \oslash \tag{3}$$

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

The Protection work $W = \{B_1, B_2, ..., B_n\}$ 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 (figure 2) and a population of 100 individuals, randomly generated inside the Protection Measures Zone, was considered.

The choice of an appropriate fitness function is essential to evaluate the goodness of a given solution. In the present study, two different fitness functions were considered: 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:

$$f_1 = \frac{\mu(S \cap Z)}{\mu(S \cup Z)} \tag{4}$$

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:

$$f_2 = \frac{\sum_{i=1}^{|W|} p_c \cdot d(B_i) \cdot h(B_i)}{V_{max}} \tag{5}$$

and since the barriers are composed of two nodes, the function can be written as:

$$f_2 = \frac{\sum_{i=1}^{|W|} p_c \cdot d(N_{i1}, N_{i2}) \cdot \bar{h}(N_{i1}, N_{i2})}{V_{max}} \tag{6}$$

where $\bar{h}(N_{i1}, N_{i2}) = \frac{|z_{i1}+z_{i2}|}{2}$ represents 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 its length (in meters). The final fitness function f_2 can be written as:

$$f_2 = \frac{\sum_{i=1}^{|W|} p_c \cdot \sqrt{(x_{i1}-x_{i2})^2 + (y_{i1}-y_{i2})^2} \cdot \frac{|z_{i1}+z_{i2}|}{2}}{V_{max}} \tag{7}$$

where 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. 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:

$$f_3 = f_1 \cdot \omega_1 + f_2 \cdot \omega_2 \tag{8}$$

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 were 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

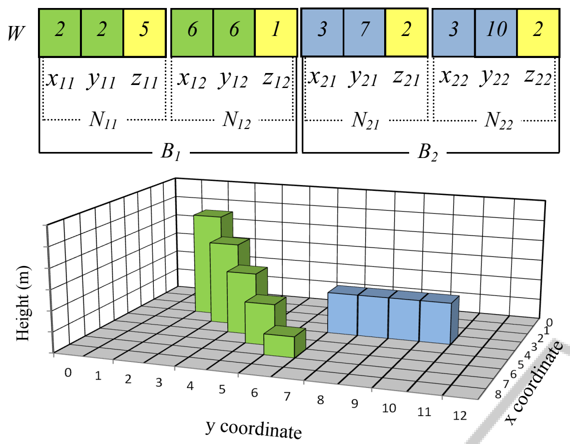


Figure 2: Barriers encoding into a GA genotype.

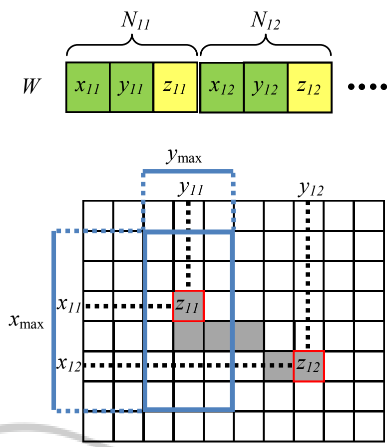


Figure 3: Graphical representation of the genotype mutation phase. Each gene, representing a CA coordinate, can vary within a variation radius (x_{max}, y_{max}).

(i.e, the barrier cells). To complete this step, an extending Bresenham's original algorithm (Bresenham, 1965) 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 (Figure 3). This strategy ensures the possibility for the GA to provide, as output, either protective barriers or dams.

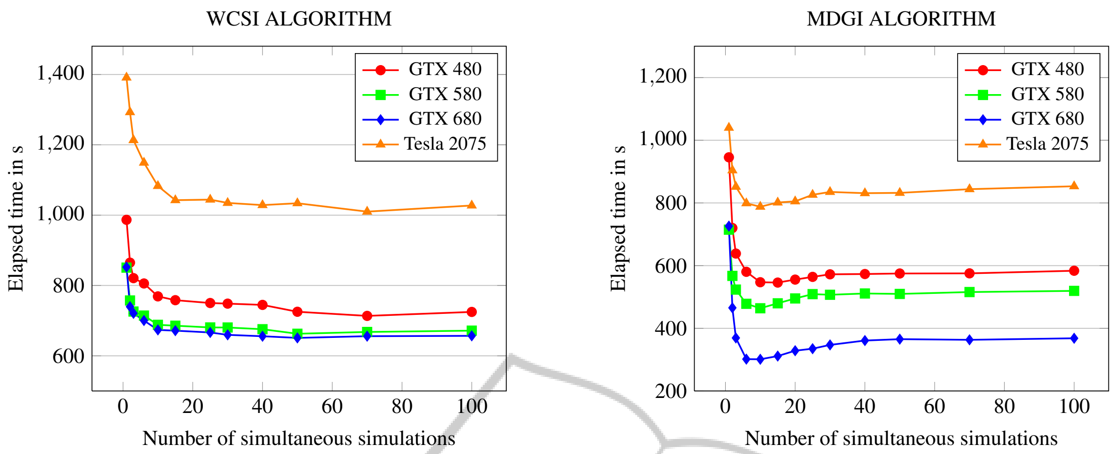
4.1 Parallel Implementation and Performance

To evaluate a given GA individual, an entire CA simulation has to be performed, followed by a comparison of the obtained result with the actual case study. Depending on the adopted computer framework, such an operation 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. Thus, if the GA population is composed of 100 individuals, the time required to run one seed test (100 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.

Due to the high computational complexity of the algorithm, a *CPU/GPU* library was developed to accelerate the GA running. A "Master-Slave" model was adopted in which the Host-CPU (Master) executes the GA steps (selection, population replacement

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
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



(a) (b)

Figure 4: Elapsed time as a function of simultaneus lava events usign the WCSI (a) and MDGI (b) approaches on different considered GPGPU hardware.

and mutation), while GPU cores (slaves) evaluate the individuals fitness.

In order to test and evaluate the different implementation strategies, 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. 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. Two parallelisation strategies and different graphic hardware were adopted in all experiments reported below. In particular, four CUDA devices were used in the experiments: one 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.

Inspired by previous works in CA modelling with GPGPU (D'Ambrosio et al., 2012a; Filippone et al., 2011), a first straightforward parallel implementation, labeled as WCSI (Whole Cellular Space Implementation) was considered where the CUDA kernels operate on the whole automaton. Since during a lava flow simulation only the transition function of the currently active cells do significant computation, simulating only one simulation at a time would imply a high percentage of uselessly scheduled threads. In addition, given the small size 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. For these reasons, 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 propagated 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 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. Using the adopted GPU devices, the algorithm was solved with the WCSI approach and a variable number of simultaneous lava simulations. According to the results shown in Figure 4.a, the GTX 680 achieved the lowest elapsed time of 650,96s, 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 40 over the used CPU.

A critical aspect of CA implementations that can improve performance, which is also related to CA sequential versions, is that the application of the transition function can be restricted to the only active cells where computation is actually taking place. When considering a phenomenon (e.g., a lava flow) that is topologically connected (i.e., a simulation starts from few active cells and evolves by activating neighbour cells), 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. In case of the above WCSI CA GPGPU implementation,

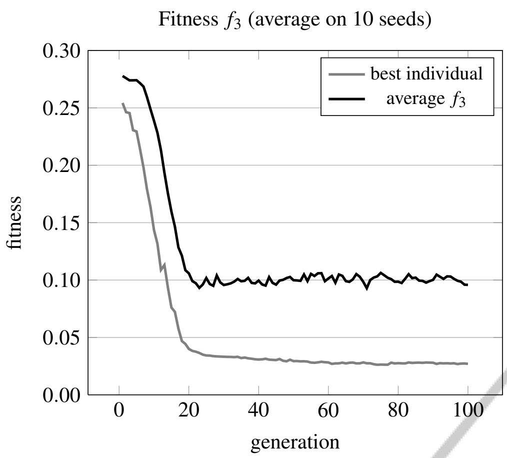


Figure 5: Temporal evolution of composite f_3 fitness of best individual (in black) and of average fitness of whole population (in gray). Fitness values were obtained as an average of 10 GA runs, carried out by adopting different seeds for generation of random numbers.

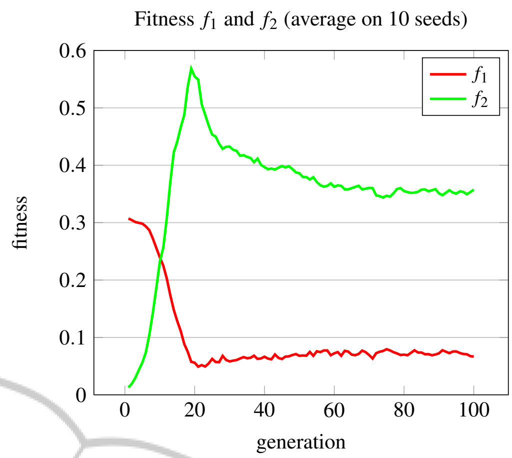


Figure 6: Temporal evolution of average fitness f_1 (in red) and f_2 (in green) of whole population. Fitness values were obtained as an average of 10 GA runs, carried out by adopting different seeds for generation of random numbers.

this weakness is even more evident due to the difficulty of having a high percentage of computationally active threads in the CUDA grid. For these reasons, an alternative 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 order to perform a fair comparison, an analogous strategy based on the bounding box has been developed for the sequential version of the program. Using the reference CPU, such sequential procedure required 20180 s for the case study adopted. Figure 4.b shows the corresponding 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 301,18s, corresponding to a parallel speedup of 67.

4.2 Experiment and Results

By considering the Nicolosi lava flow event, 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 (Filippone et al., 2013), 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 Figure 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:

$$S_r = \{[P_{x_{min}}, P_{x_{max}}] \times [P_{y_{min}}, P_{y_{max}}] \times [(h_{min} \cdot n_g), (h_{max} \cdot n_g)]\}^{2|W|} \tag{9}$$

The temporal evolution of the f_3 fitness is graphically reported in Figure 5, in terms of average results over the ten considered experiments. GA experiment pa-

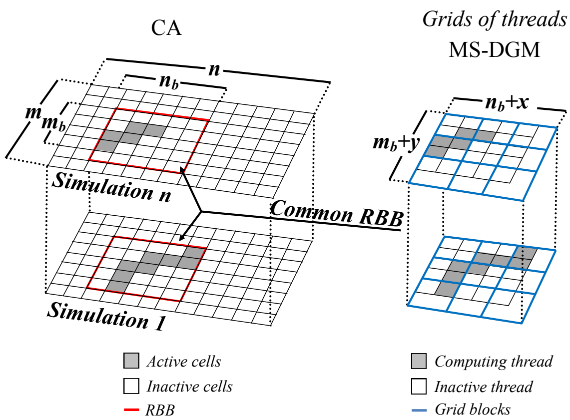


Figure 7: Mapping of the CA transition function into a CUDA grid of threads (right) in case of the simultaneous lava flows (left).

Table 2: Dimensions of the ten best barriers carried out by GA run.

Seed test	Barrier	Length (m)	Height (m)	Base Width (m)	Volume (m^3)	Inclination (degrees)
1234	[134,173,18] [114,158,35]	250	26,5	10	66250	143
2345	[135,178,8] [114,157,43]	297	25,5	10	75731	135
3456	[133,172,19] [115,155,37]	247	28	10	69324	137
4567	[113,158,45] [132,177,9]	269	27	10	72549	135
5678	[115,154,44] [133,171,14]	248	29	10	71800	137
6789	[133,172,12] [115,155,42]	248	27	10	66848	137
7890	[114,159,40] [133,171,10]	225	25	10	56180	148
8901	[115,156,48] [134,173,9]	255	28,5	10	72661	138
9101	[134,173,8] [114,157,41]	256	24,5	10	62750	141
1011	[115,152,38] [134,172,18]	276	28	10	77241	134

rameters values are also listed in table 1. The related CA simulation, obtained by adopting the best individual is shown in Figure 8.

The study, though preliminary, has produced quite satisfying results. Among different best individuals generated by the GA for each seed test (table 2), the best one consists of a barrier with an average height of 25 m and 220 m in length with an inclination angle of 130° with respect to the direction of the lava flow. The barrier (seed 7890 in table 2) completely deviates the flow avoiding that the lava reaches the inhabited area. The relative elevated height of the barrier is due to the adopted GA (only solutions characterized by single walls) and problem constraints, like the crater location too close to the area to be protected or the morphology characteristics.

4.3 Considerations on the GA Dynamics and Emergent Behaviors

In the GA experiments that have been performed, individuals with high fitness evolved rapidly, even if

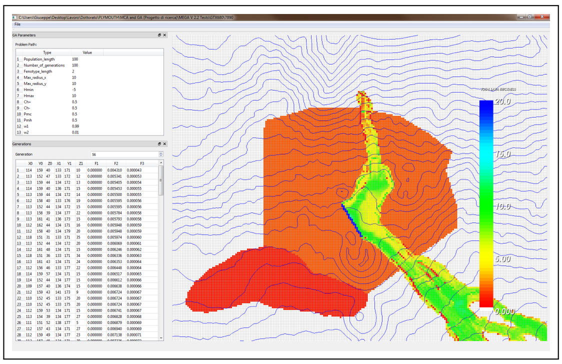


Figure 8: SCIARA - fv2 simulation visualization adopting the GA best solution. As seen, the devised barrier (in blue) completely diverts the lava flow from the Safety Area (in red).

the initial population was randomly generated and the search space was quite large (Equation 9). 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. Figure 6).

In any case, the evolutionary process has shown, in accordance with the opinion of the scientific community (Barberi et al., 2003; Fujita et al., 2009), the ineffectiveness of barriers placed perpendicular to the lava flow direction despite diagonally oriented solutions ($130° - 160°$) (see Table 2).

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 (Figure 9). 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 (radius 10) and the central cell. More formally, the function that assigns to each generic node j a *slope proximity* value is defined as:

$$sp_j = \frac{\sum_{i=1}^{|X|} \bar{z}_i - \bar{z}_0}{|X|} \tag{10}$$

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 Figure 10, starting from the tenth GA generation, the evolutionary process has shown an increase in slope proximity values. Therefore, after the f_1 optimization (cf. Figure 6), 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.

5 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.

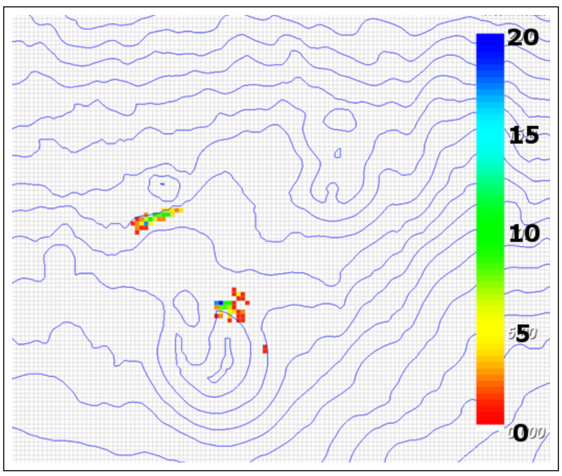


Figure 9: Nodes distribution of the best 100 solutions generated by the GA. Scale values indicates occurrence of nodes.

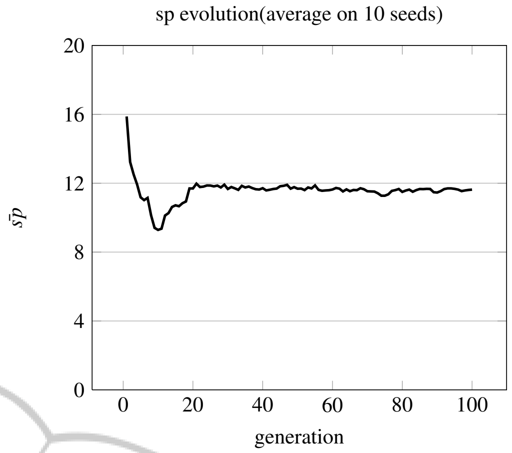


Figure 10: Temporal evolution of average slope proximity values for the best individuals.

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.

Future work will consider the investigation of solutions consisting of multiple protective interventions and the introduction, within the methodology, of lava cooling by water jets.

ACKNOWLEDGEMENTS

This work was partially funded by the European Commission European Social Fund (ESF) and by the Regione Calabria (Italy). Authors are also grateful to Plymouth University (UK) and the CUDA Teaching Center of the same University for supplying computing facilities.

REFERENCES

Avolio, M. V., Crisci, G. M., Di Gregorio, S., Rongo, R., Spataro, W., and D'Ambrosio, D. (2006). Pyroclastic flows modelling using Cellular Automata. *Computers & Geosciences*, 32:897–911.

Barberi, F., Brondi, F., Carapezza, M., Cavarra, L., and Murgia, C. (2003). Earthen barriers to control lava flows in the 2001 eruption of Mt. Etna. *Journal of Volcanology and Geothermal Research*, 123:231–243.

Barberi, F., Carapezza, M., Valenza, M., and Villari, L. (1993). The control of lava flow during the 1991-1992 eruption of Mt. Etna. *Journal of Volcanology and Geothermal Research*, 56:1–34.

Barberi, F. and Carapezza, M. L. (2004). *Mt. Etna: Volcano Laboratory*, chapter The Control of Lava Flows at Mt. Etna, pages 357–369. American Geophysical Union, Washington, D. C.

Barca, D., Crisci, G. M., Di Gregorio, S., and Nicoletta, F. (1994). Cellular Automata for simulating lava flows: a method and examples of the etnean eruptions. *Transport Theory and Statistical Physics*, 23(1-3):195–232.

Behncke, B. and Neri, M. (2003). The July-August 2001 eruption of Mt. Etna (Sicily). *Bulletin of Volcanology*, 65(7):461–476.

Bentley, P. (1999). An introduction to evolutionary design by computers. In Bentley, P. J., editor, *Evolutionary Design by Computers*, chapter 1, pages 1–73. Morgan Kaufman, San Francisco, USA.

Bongard, J. (2011). Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108(4):1234–1239.

Bresenham, J. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30.

Colombrita, R. (1984). Methodology for the construction of earth barriers to divert lava flows: The Mt. Etna 1983 eruption. *Bulletin Volcanologique*, 47(4):1009–1038.

Crisci, G. M., Rongo, R., Gregorio, S. D., and Spataro, W. (2004). The simulation model SCIARA: the 1991 and 2001 lava flows at Mount Etna. *Journal of Volcanology and Geothermal Research*, 132(23):253 – 267.

D'Ambrosio, D., Filippone, G., Rongo, R., Spataro, W., and Trunfio, G. A. (2012a). Cellular automata and GPGPU: an application to lava flow modeling. *International Journal of Grid and High Performance Computing*, 4(3):30–47.

D'Ambrosio, D., Rongo, R., Spataro, W., and Trunfio, G. (2012b). Optimizing Cellular Automata through a Meta-model Assisted Memetic Algorithm. In *Proceedings of Parallel Problem Solving from Nature - PPSN XII*, volume 7492 of *Lecture Notes in Computer Science*, pages 317–326. Springer Berlin Heidelberg.

D'Ambrosio, D., Rongo, R., Spataro, W., and Trunfio, G. A. (2012c). 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, pages 533 – 542, Berlin, Heidelberg. Springer-Verlag.

Del Negro, C., Fortuna, L., Herault, A., and Vicari, A. (2008). Simulations of the 2004 lava flow at Etna volcano using the magflow cellular automata model. *Bulletin of Volcanology*, 70(7):805–812.

Di Gregorio, S. and Serra, R. (1999). An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. *Future Generation Computer Systems*, 16(2-3):259–271.

Di Gregorio, S., Serra, R., and Villani, M. (1999). Applying cellular automata to complex environmental problems: The simulation of the bioremediation of contaminated soils. *Theoretical Computer Science*, 217(1):131 – 156.

Dibben, C. (2008). Leaving the city for the suburbs - The domincance of 'ordinary' decision making over volkanik risk perception in the production of volcanic risk on Mt. Etna, sicily. *Journal of Volcanology and Geothermal Research*, 172(7):288–299.

ElSayed, A., Kongar, E., Gupta, S., and Sobh, T. (2012). A Robotic-Driven Disassembly Sequence Generator for End-Of-Life Electronic Products. *Journal of Intelligent & Robotic Systems*, 68(1):43–52.

Filippone, G., Spataro, D'Ambrosio, D., and Marocco, D. (2013). An interactive Visualization System for Lava Flows Cellular Automata Simulations using CUDA. Poster presented at GPU Technology Conference, San Jose, California.

Filippone, G., Spataro, W., Spingola, G., D'Ambrosio, D., Rongo, R., Perna, G., and Di Gregorio, S. (2011). GPGPU programming and cellular automata: Implementation of the SCIARA lava flow simulation code. In *23rd European Modeling and simulation Symposium (WMSS)*, Rome, Italy.

Fujita, E., Hidaka, M., Goto, A., and Umino, S. (2009). Simulations of measures to control lava flows. *Bulletin of Volcanology*, 71:401–408.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Goncalves, J. F. and Resende, M. G. (2011). Biased random-key genetic algorithms forcombinatorial optimization. *Journal of Heuristics*, 17(5):487–525.

Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, pages 495–502.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press.

Iovine, G., D'Ambrosio, D., and Di Gregorio, S. (2005). Applying genetic algorithms for calibrating a hexagonal cellular automata model for the simulation of debris flows characterised by strong inertial effects. *Geomorphology*, 66(14):287 – 303.

Ishihara, K., Iguchi, M., and Kamo, K. (1990). Numerical simulation of lava flows on some volcanoes in Japan. In Fink, J. H., editor, *Lava Flows and Domes*, volume 2 of *IAVCEI Proceedings in Volcanology*, pages 174–207. Springer Berlin Heidelberg.

Kicinger, R., Arciszewski, T., and Jong, K. D. (2005). Evolutionary computation and structural design: A survey of the state-of-the-art. *Comput. Struct.*, 83(23-24):1943–1978.

Lipson, H. and Pollack, J. B. (2000). Automatic Design and Manufacture of Artificial Lifeforms. *Nature*, 406:974–978.

MacDonald, G. A. (1962). The 1959 and 1960 eruptions of Kilauea volcano, Hawaii, and the construction of walls to restrict the spread of the lava flows. *Bulletin Volcanologique*, 24(1):249–294.

Mitchell, M. (1996). *An introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.

Miyamoto, H. and Sasaki, S. (1997). Simulating lava flows by an improved cellular automata method. *Computers & Geosciences*, 23(3):283–292.

Nolfi, S. and Marocco, D. (2001). Evolving robots able to integrate sensory-motor information over time. *Theory in Biosciences*, 120:287–310.

Piwonska, A., Seredynski, F., and Szaban, M. (2013). Learning cellular automata rules for binary classification problem. *The Journal of Supercomputing*, 63(3):800–815.

Rongo, R., Spataro, W., D'Ambrosio, D., Avolio, M. V., Trunfio, G. A., and Di Gregorio, S. (2008). Lava flow hazard evaluation through cellular automata and genetic algorithms: an application to Mt Etna volcano. *Fundam. Inf.*, 87:247–267.

Sims, K. (1994). Evolving 3d morphology and behavior by competition. In *Proceedings of Artificial Life IV*, pages 28–39. MIT Press.

Spataro, W., Avolio, M. V., Lupiano, V., Trunfio, G. A., Rongo, R., and D'Ambrosio, D. (2010). 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 *International Conference on Computational Science*, pages 17–26.

Spataro, W., D'Ambrosio, D., Rongo, R., and Trunfio, G. (2004). An Evolutionary Approach for Modelling Lava Flows through Cellular Automata. In *ACRI 2004*, volume 3305 of *Lecture Notes in Computer Science*, pages 725–734. Springer Berlin Heidelberg.

Tomassini, M. and Venzi, M. (2002). Artificially evolved asynchronous cellular automata for the density task. In *Procedings of the fifth International Conference on Cellular Automata for Research and Industry.*, volume 2493 of *Lecture Notes in Computer Science*, pages 44–55. Springer Berlin Heidelberg.