Controlling and investigating Cellular Automata behavior via interactive inversion and visualization of search space

F. Boschetti, CSIRO, Exploration and Mining.

Abstract

We use an Interactive Genetic Algorithm to optimise the input parameters controlling the behaviour of a Cellular Automata. We aim to deduce which combination of parameters allow the CA to reproduce patters seen in geological scenarios as a result of fluid flow and chemical reaction in fractured media.

Via the Interactive Genetic Algorithm the user can provide subjective feedback on the quality of the CA results, which would otherwise be difficult to express numerically. A simple modification to the IGA ranking process, combined with a Self Organised Map, enables the rapid on-line visualisation of the high dimensional parameter space and consequent control over the inversion itself. The insights into the topology of the parameter space offer a rough understanding of which parameters control different Cellular Automata behaviours.

Introduction

Interactive Genetic Algorithms (IGA) have been used in the optimization of problems for which it is impossible, or very difficult, to define a proper numerical cost function. The fitness, or quality, of a solution is defined subjectively by the user and provided to the algorithm, often in the form of a ranking.

IGA were initially proposed in artistic applications, for which numerical evaluation of quality are not yet established. Subsequently, they have been extended to other engineering and scientific problems. An exhaustive review of academic and industrial applications, as well as details of different implementations, can be found in Takagi (2001).

We have used IGA in the optimization of geo-dynamical problems related to mineral exploration. Our interest lay not only in the reconstruction of the initial parameters which can generate a certain geological behavior (Boschetti and Moresi, 2001), but also in a rough description of the parameter space of the problem, in order to achieve an approximate understanding of different mechanical behaviors found in nature (Wijns et al, 2002). To this purpose we found particularly useful to employ methods for visualization of high dimensional spaces (Boschetti et al, 2002). By plotting the entire population of an IGA run on a Self Organized Map (SOM) (Kohonen, 2001) an approximate understating of sub-domains of different geo-mechanical behavior can be obtained.

In generating such plot the user is faced with a relatively simple, but time consuming problem. The subjective evaluation of the solutions in the IGA run is often provided in the form of a ranking. This ranking is performed only within individual generations, not between individuals from different generations. Consequently, at the end of an IGA run no 'global' ranking of the overall population is available. In a nutshell, we are not able to evaluate whether, say, the 7^{th} individual of the 4^{th} generation is better than the 5^{th} individual of the 2^{nd} generation, unless we ask the user again. The only 'link' between generations is given by the best individual(s) been carried forward from generation to generation, when elitism (a standard GA module, see Davis, 1991) is applied. However, this is not enough to generate a reliable global ranking.

In our previous work we have asked the user to re-rank the entire population, on a global scale, at the end of the IGA run. This is time consuming, as well as very tedious. In this paper we propose a very simple method to overcome this problem.

An interesting immediate application of the idea is the possibility of using the SOM (or similar high dimensional visualization methods) to monitor the IGA search on-line. This gives the user a second level of interactivity in the possibility to alter some GA parameters (like population size, mutation rate ect) as well as to modify the search space by changing the range or resolution of certain parameters (dimensions) or even altogether removing certain parameters deemed already 'optimized'.

The test application

For the sake of clarity, we present our method by describing a test case. The purpose of the test case is to deduce, via global optimization, the set of input parameters which allows a Cellular Automata (CA) to reproduce patterns seen in real geological scenarios. Previous work on the use of Evolutionary Computation to evolve CA rules can be found in Mitchell et al. (1997).

The CA used in this work models fracturing, fluid flow and chemical reaction in a geological medium. Figure 1 shows a CA run at different stages of its evolution. The lattice represents a 2D vertical section through the earth crust. Two fluids (red and blue) are injected into the system from the bottom with a certain pressure. Each cell in the CA is characterized by a (random) material strength value. When the fluid pressure reaches the maximum strength of a cell a fracture occurs. The presence of the fluid in the fracture further reduces the material strength, which increases the probability of the fluid to accumulate in the cell and the fracture to propagate from it. Consequently, the input material properties control the probability of the fracture to propagate in one direction, to bifurcate or to diffuse. After a while the two fluids spread in the lattice and may get in contact with one another. When this happens, the two fluids mix, a chemical reaction occurs and gold precipitates (the fluid turns yellow).

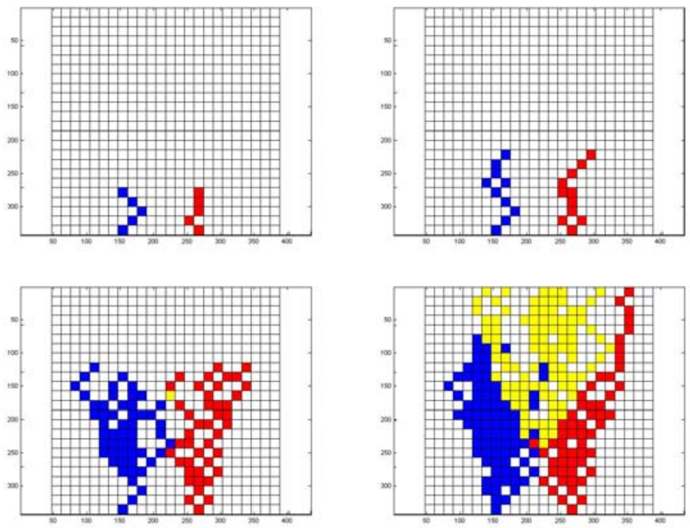

Figure 1. Four snapshots of the CA run at different stages. The calculation domain represents a 2D vertical section through the earth crust. Two fluids of different composition (red and blue) are injected into the system under pressure from the bottom. They fracture and move in the medium. When they mix, they react and deposit gold (yellow).

The CA algorithm can be summarized in the following way:

1) the lattice represent the material strength of the medium. The lattice points are initialized with random values between a minimum (*min_stre*) and a maximum material strength allowed (*min_stre+stre_var*);
2) fluid is injected into the system at a certain pressure from the bottom;
3) when the fluid pressure is greater than the material strength at a site the site cracks. In this case a fraction (*%_fluid_flow*) of fluid contained at the site flows to an adjacent site. The adjacent cell is chosen randomly. However the lower the material strength, the more likely a site is chosen.
4) Once the fluid flows through a site, the material strength on that site decreases of the amount *decr_strength* (as it happens in real systems). However, the material strength at the sites where no fluid flows increases on the amount *incr_strength*. This is supposed to mimic the natural strengthening of rock bonds with time. As a result the more fluid flows in a site, the more likely it is that more fluid will flow. This condition allows the creation of fluid channels as seen in nature.

Consequently, the values of *decr_strength* and *incr_strength* affect the pattern of fractures which can arise in a CA run.

5) the parameter (*press_for_crack*) determines how likely a fracture can change direction of propagation. This parameter further affects the process at step 3 to determine the final direction of fracture propagation at each step. The physical interpretation of this parameter is to mimic the presence of local anisotropy in the medium.
6) Finally, a parameter (*time steps*) determine the the number of cycles the CA is allowed to run, which mimic the duration of the geological process

This is clearly an oversimplification of a real mineralization process. The purpose of this experiment is merely to see if patterns similar to the ones seen in nature can be obtained by tuning a CA simply by providing subjective judgments on the CA performance to an IGA.

In this particular experiment we aim to obtain a final geological scenarios characterized by gold mineralization occurring only along long, narrow, isolated and possibly inclined thin bodies (dykes in geological jargon). A simple sketch of the target geological scenario can be seen in Figure 2.

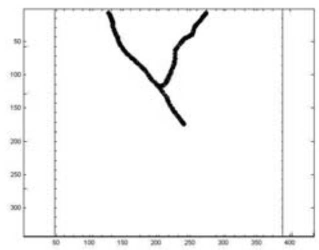

Figure 2. Target pattern of gold mineralization we aim to reconstruct via the CA. Because of the stochastic CA behavior, we are not interested in the exact spatial location of the mineralization, rather on its 'statistical appearance', which resembles pattern seen in nature.

'Global' fitness evaluation in IGA

The only fundamental difference between a standard GA and an IGA lies in the fitness evaluation, which is performed by the user (IGA) rather than calculated via a numerical cost function (GA). This does not require any algorithmic modification between the codes. It does however generate some implementation issues, mainly:

1) the requirement of the subjective evaluation of the solutions imposes limitations on the population size. This occurs because such evaluation may be both time consuming and tiring (what Takagi (2001) defines as 'human fatigue');
2) the requirement of a small population size may affect the choice of crossover and mutation rates;

3) a proper interface is beneficial to speed up the user evaluation. This is very important in order to make routine real world applications efficient.

Our proposed modification concentrates on point 3. In Figure 3 we can see a screen capture of the user interface we employed in our previous work (Boschetti and Moresi, 2001). This is not identical, but conceptually similar, to other implantations described in the literature. This user interface employs a 'drop down' menu to input the rank position of the solution. Other user interface proposed in the literature allow to choose a class ('very good', 'good', 'bad', and so on). Repeatedly pressing the + or – key in the keyboard has also been proposed to rank in real time the quality of short music passages (??). These methods are effective when a global ranking is not required.

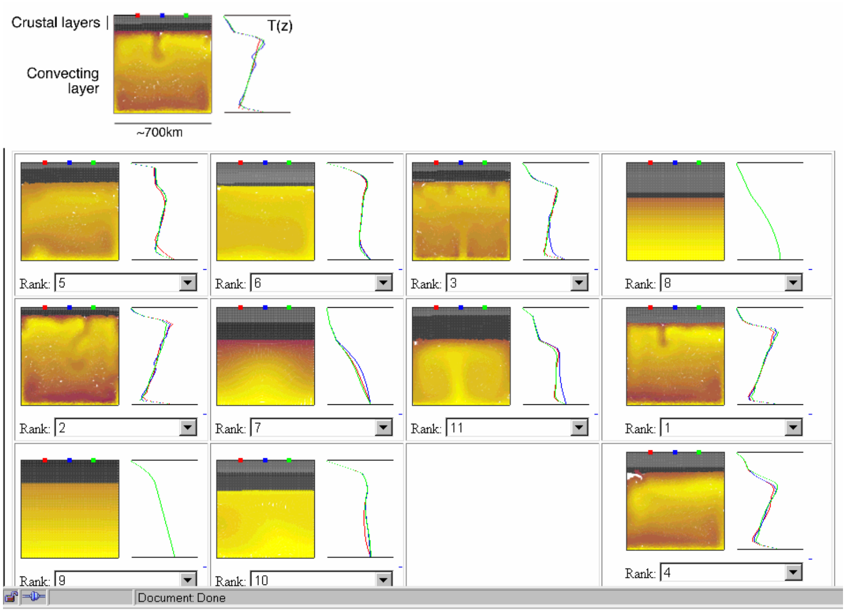

Figure 3. Screen capture of an interface for subjective ranking of solution quality. On the top left we can see the target image. Underneath the GA current generation is displayed. At the bottom right the best individual from the previous generation (elitism) is also shown. Ranking is performed by using the 'drop-down' menu underneath each individual. This sort of interface allows for 'intra generation' ranking.

When the optimization problem involves images or computer animations, and a global ranking is required, we propose to use an interface as in Figure 4. This is nothing but an image display software, which allows easy organization, view and sharing of digital pictures and videos. In this example we have used the free-ware software FotoAlbum (??). Other packages can be used in similar fashion.

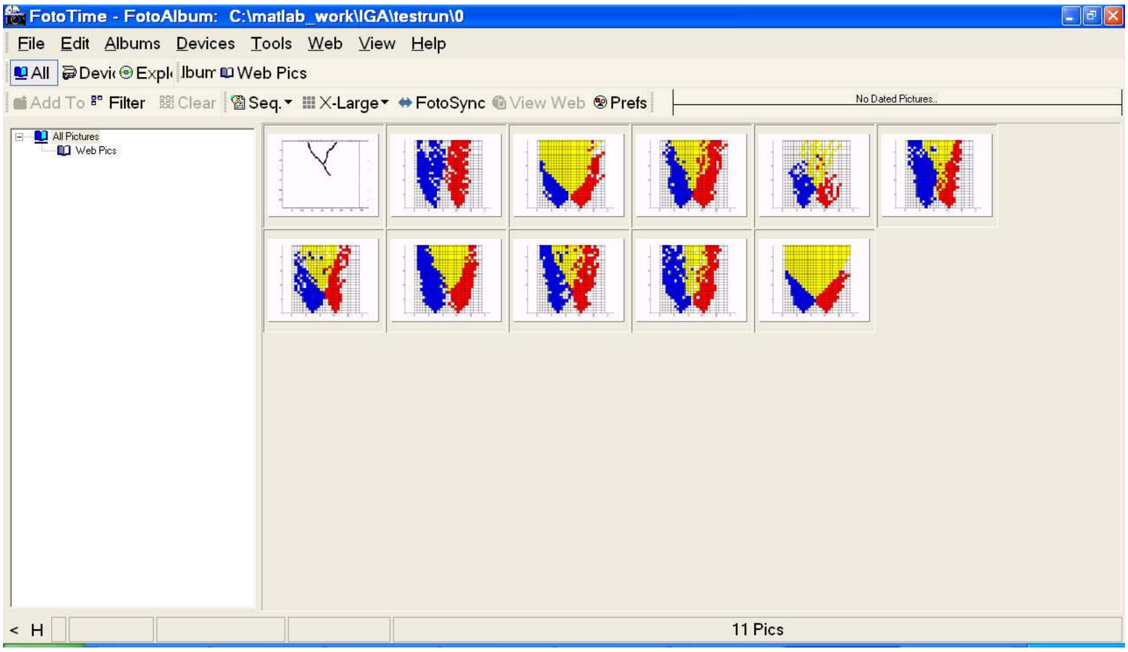

Figure 4. Screen capture of a proposed interface for ‘global ranking’. The image on the top left shows the target of the inversion. The remaining images are the IGA population at the first generation. The user can re-arrange the images in terms of quality by swapping their position via point and drag with the mouse.

Figure 4 shows the output at the first generation of the IGA run (before the first ranking). The images are listed in order of individual number (1-10). The quality ranking at this stage is expected to be random. The user can re-arrange the image in terms of quality by swapping their position. This is achieved simply by dragging the images to the correct rank position. The result can be seen in Figure 5.

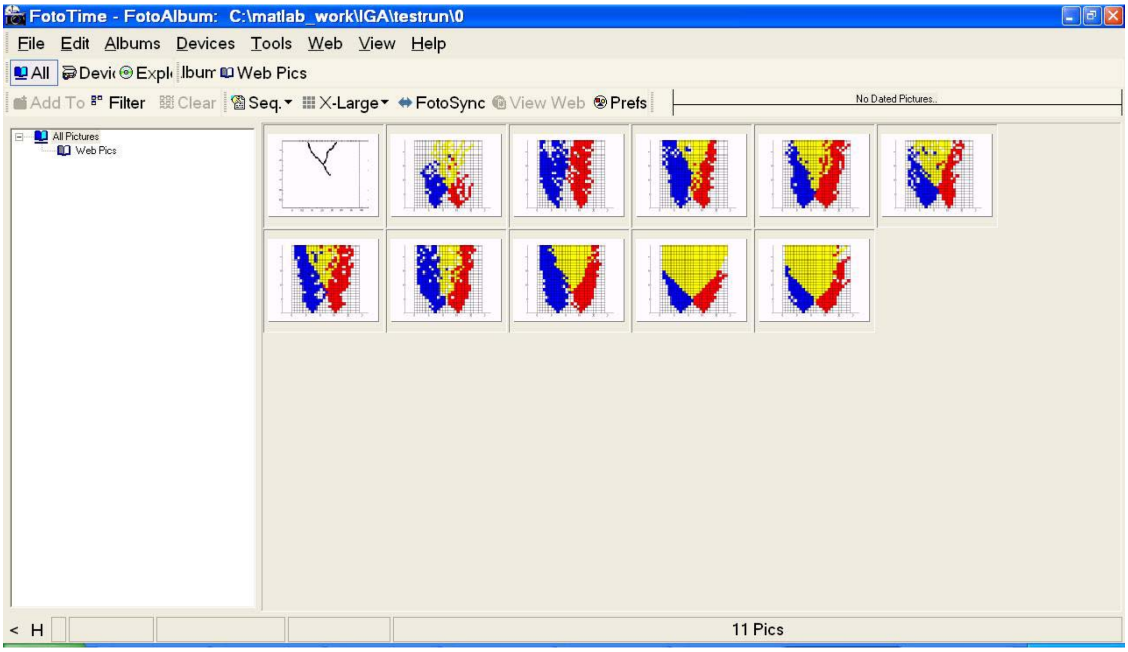

Figure 5. Screen capture of the proposed interface after first ranking. The image positions have been re-ordered according to the similarity with the target.

So far no great advantage is obtained compared to standard drop-down menu ranking. The benefit becomes apparent at the second generation (Figure 6). The first 10 images are the individuals from the first generation, with their previous ranking. The remaining 10 individuals are from the second generation, still to be ranked. Their ranking is performed simply, as for the first generation, by dragging them into their correct position. By taking into account the quality of the individuals from the first generation as well, we implicitly obtain a global ranking among the entire IGA population. This can be seen in Figure 7. The process can be repeated at each generation.

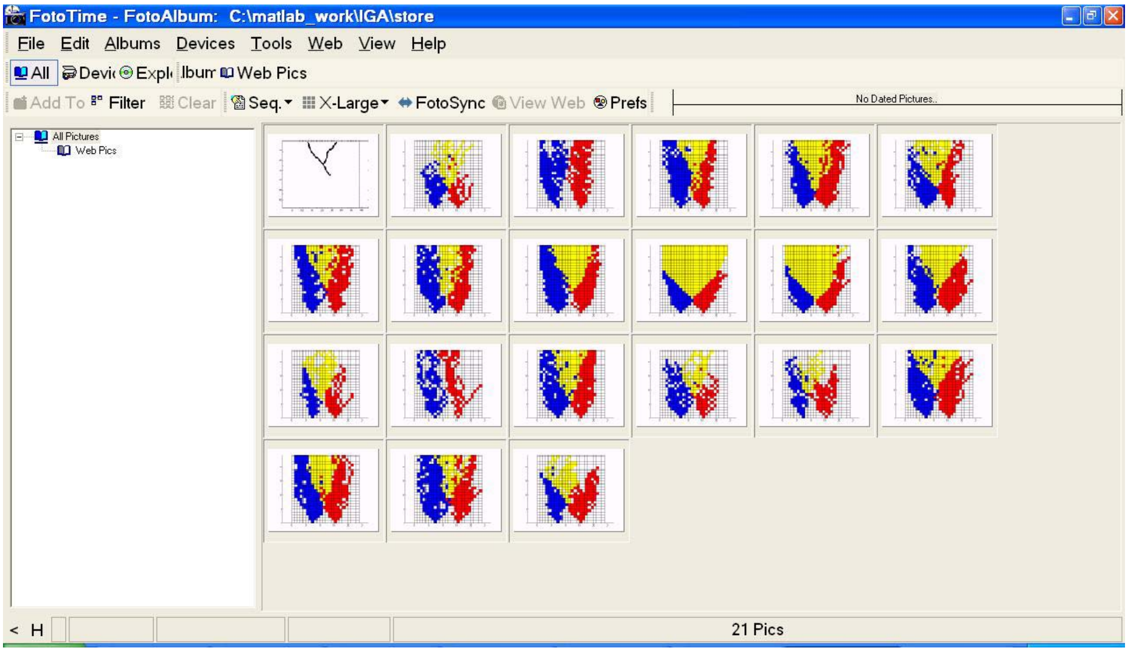

Figure 6. Screen capture of the proposed interface after the second generation. At this stage only the individuals of the 1st generation (first 10 after the target) have been ranked. The individuals from the second 2nd (remaining 10) have not been ranked yet.

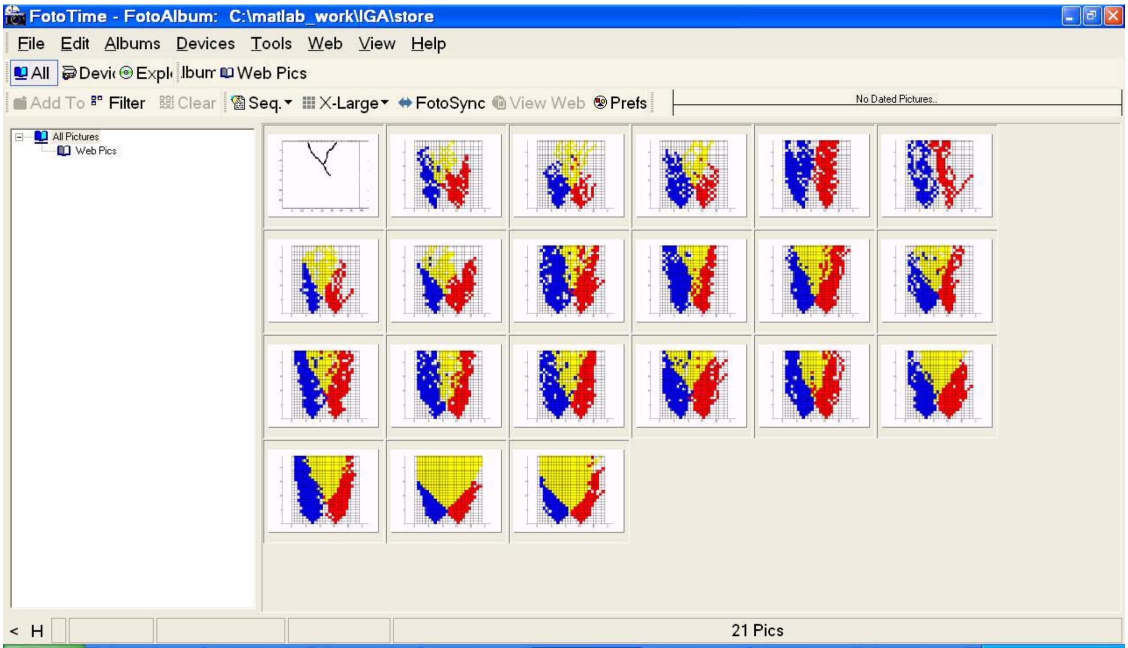

Figure 7. Screen capture of the proposed interface after the second generation. The individuals of the 2nd generation have been ranked, but accounting also for the individuals form the 1st generation. The result is a 'global' ranking among all the individuals.

The overall process is particularly simple. At each generation the user has to rank the same number of individuals as in the standard IGA implementation. All this involves is to find an image in the previous population set which is similar in appearance and insert the new image close to that ranking position (judging whether it is slightly better or slightly worse). The effort required is approximately the same at each generation, almost independent on the number of individuals accumulated in the IGA run, since the previous generations are already ranked. On the contrary, with the standard ranking method, at the last generation we would be faced with the challenging and time consuming task to have to re-rank several images (easily in the order of hundreds).

Visualization of high dimensional parameter space

In a previous work (Boschetti et al, 2002) we have shown how it is possible to obtain insights into an optimization problem via the use of high dimensional visualization tools. In that work we have used a Self Organised Map (SOM). SOM is a transformation of high-dimensional (nD) data into a lower-dimensional (usually 2D) plot. It is a classification algorithm which separates all the input data into clusters according to similarity and preserves topology, *i.e.* two points lying close to one another in the higher dimensional space also do so in the 2D space. SOM has been extensively employed in recent years in both scientific and engineering applications in order to visualise high dimensional data and highlight data structure and clustering. The SOM plots presented in this work have been obtained with the use of the Matlab$_{TM}$ SOM Toolbox, written by Juha Vesanto. More details about SOM, as well as the specific SOM implementation used in this work, can be obtained at http://www.cis.hut.fi/projects/somtoolbox.

In Figure 8 we can see the SOM visualization of the IGA population after the 2nd generation. The first 7 plates show how the input parameter values are spread over the SOM 2D map. The final plate shows the fitness, i.e., the quality of the IGA population according to the user subjective evaluation. The best individuals (lower fitness) lie in the top left corner of the SOM. This seems to correlate with high values of incr_stre, high resistance to cracking (crak_pres), high fluid flow and strong strength variation.

Another SOM based visualization helps to get further insight into the parameter space of the problem at hand. Figure 9 shows another SOM plate, usually called U-matrix. This plate gives us a rough picture of the search space topology. The colors give a measure of the distance between points in the SOM map, i.e., a measure of how stretched the 2D map is. Blue color corresponds to short distance and purple to long distance. Roughly, purple areas can be interpreted as ridges dividing clusters, that is, as areas where points lie very far from one another, subdividing areas where many point lie close to one another. Figure 9 shows two main clusters at the top and bottom left hand side of the image. From the analysis above, we know the top left cluster corresponds to the best fit images.

Further insight can be obtained by displaying the images corresponding to each model and locating them over the SOM. This confirms that high fitness images are located in the top left, with other medium/good fit images distributed in the other clusters. Poor fit images seem to be spread over the rest of the domain.

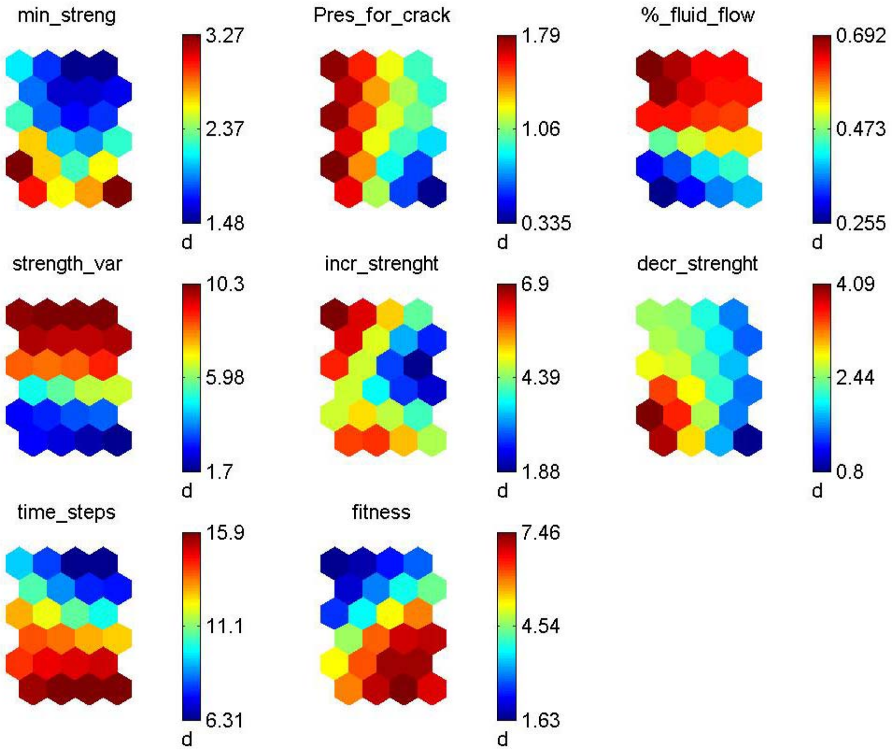

Figure 8. SOM visualization of the IGA population after the 2^{nd} generation.

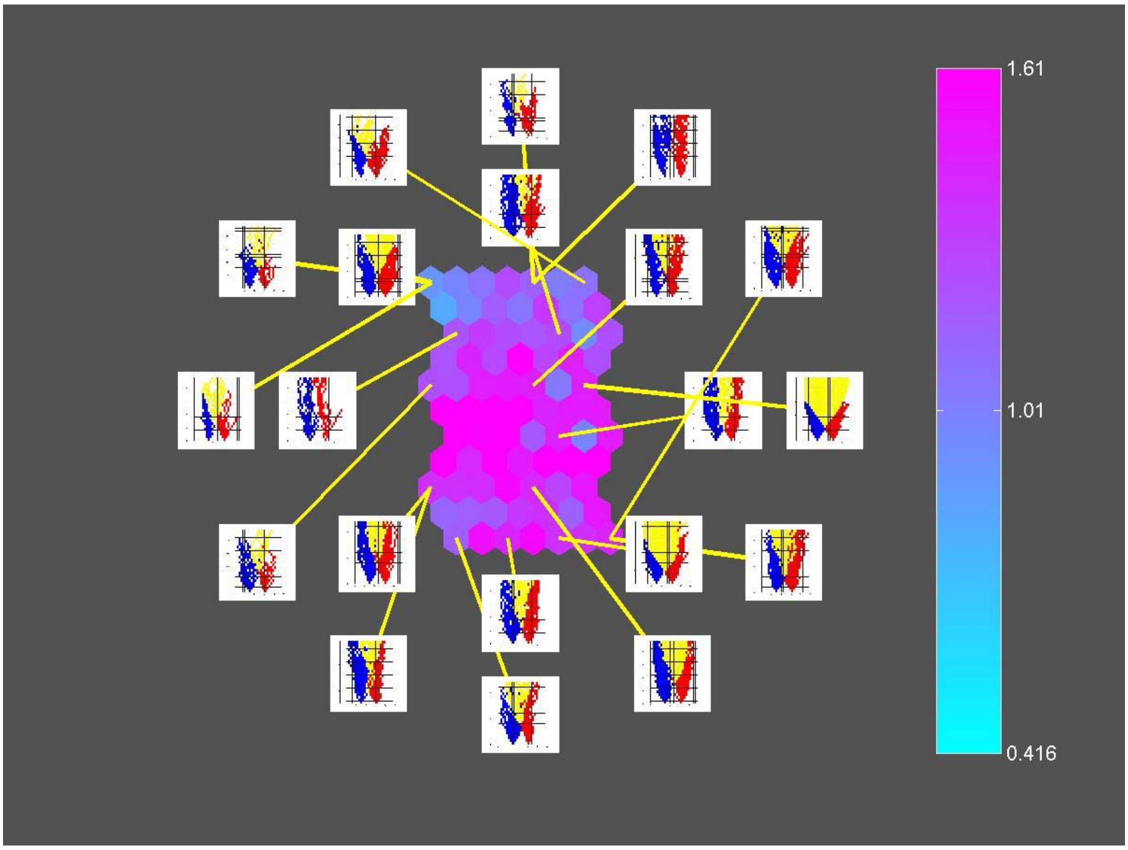

Figure 9. Topology of the SOM map at the end of the 2nd IGA generation (center). The IGA individuals are displayed around the SOM and point to the location on the SOM map. This helps to get a visual picture of the search parameter space.

Obviously, this sort of analysis becomes more accurate the further we proceed in the IGA run, that is, the more sampling of the search space we obtain. Figure 10 shows the SOM visualization at the end of the IGA run, that is after 6 generations. The main correlations seen after the 2nd generation are confirmed and the mapping of the search space is now more reliable.

Figure 11 shows the search space topology and the distribution of the models at the end of the run. As expected the cluster containing the best fit images has been further sampled by the IGA and now it is larger and contains more information. The rest of the search domain has been sampled considerably less.

The SOM map suggests the existence of 3 main clusters (see Figure 12). The images most resembling the target are all located on the top part of the SOM (cluster 1). Less fit images are located into a main cluster at the bottom right of the SOM map (cluster 2). Between the two main clusters a large ridge (purple) cuts the SOM map approximately diagonally. Another minor cluster can be seen at the bottom left (cluster 3). This also is isolated from cluster 2 by a wide ridge. Finally, the main cluster at the top may be subdivided into 2 minor clusters, due to the presence of a minor ridge running almost vertically (Clusters 1_a and 1_b).

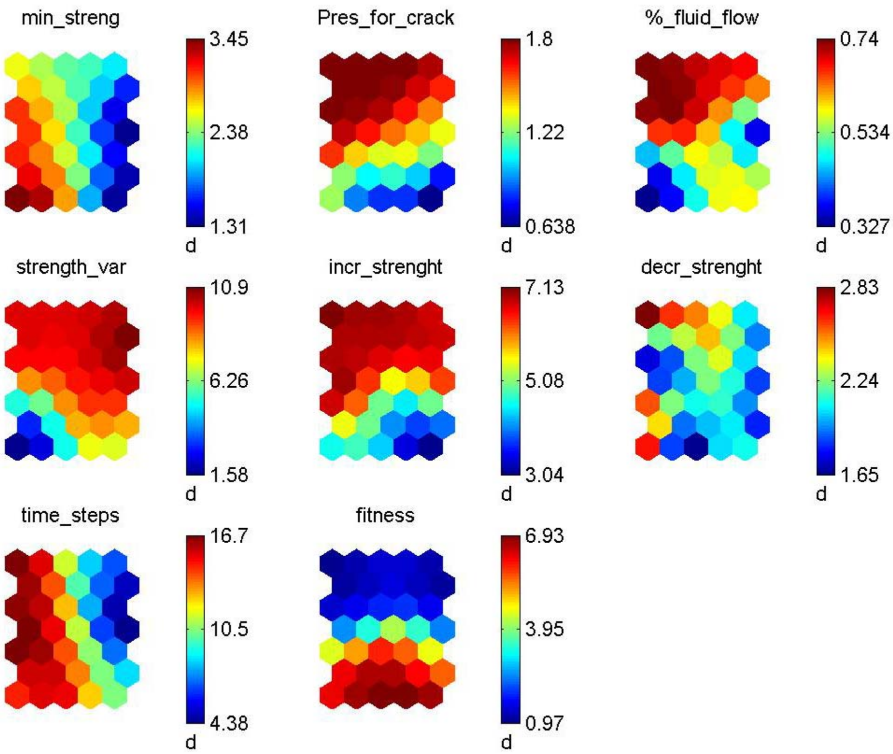

Figure 10. SOM visualization of the IGA population after the last generation.

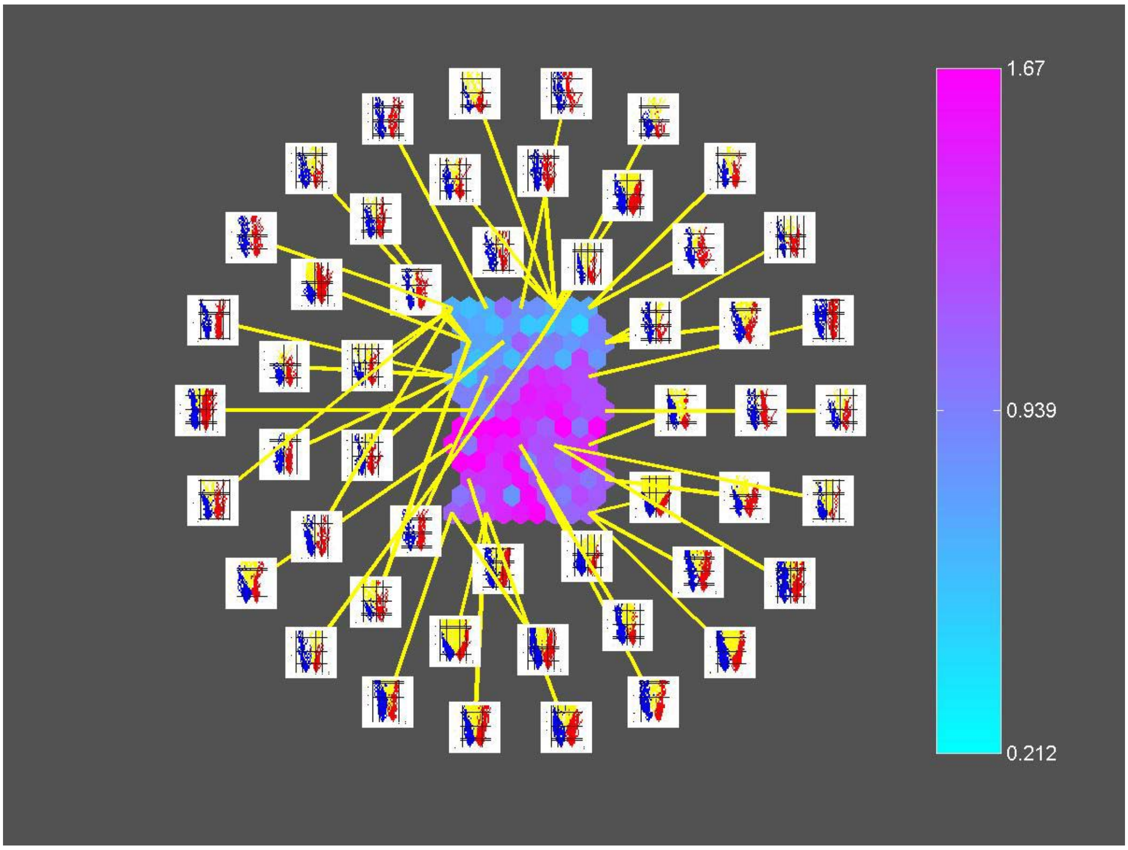

Figure 11. Topology of the SOM map at the end of the IGA run.

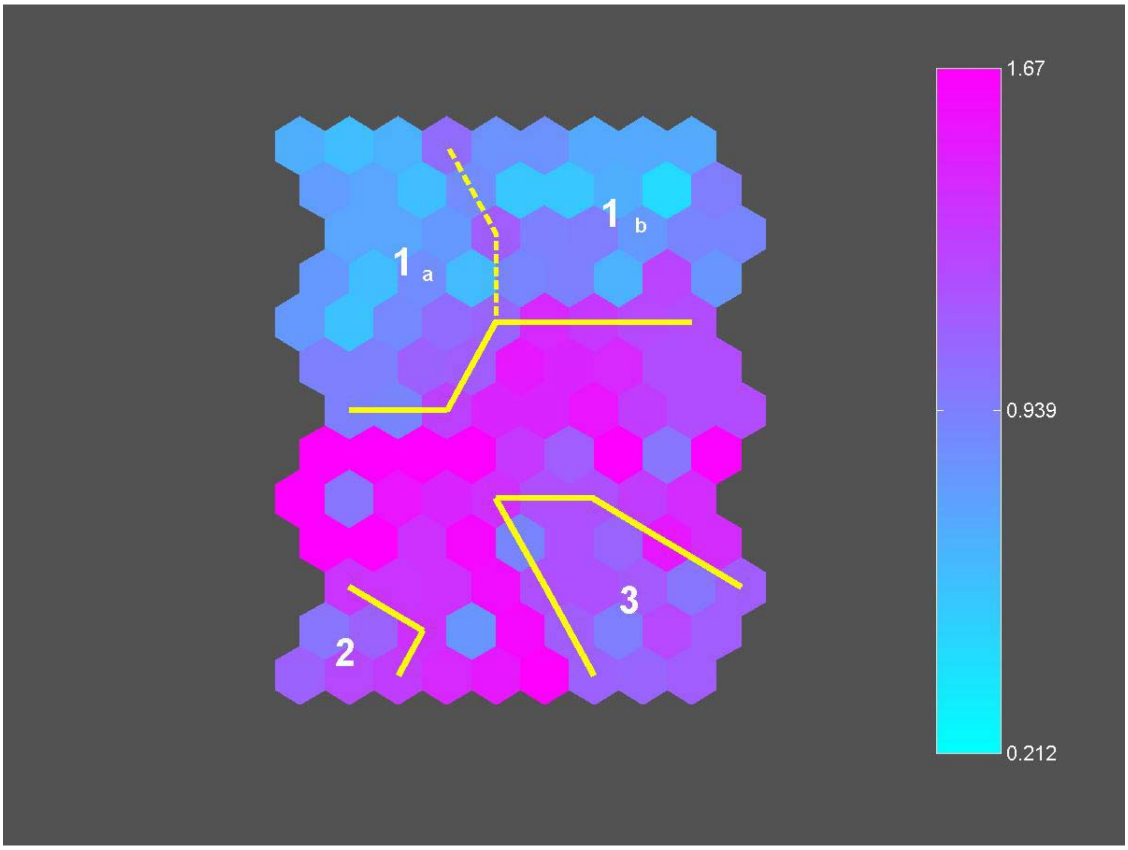

Figure 12. Main clusters arising for the IGA space sampling.

Results

An approximate understanding of the values of the CA input parameters corresponding to the 3 main clusters can be obtained via the use of the plates in Figure 10. It suggests that matches to the target geological scenario (cluster 1) are obtained for:

- strong fluid flows (high value of *%_fluid_flow*),
- high values of the variation of material strength with time (*incr_strenght*),
- high value of the parameter which controls local anisotropy, (*press_for_crack*), which favors cracks propagating along the same direction it comes from
- a large variability in initial material strength (*strength_var*)
- and medium value for the minimum material strength allowed in the system (*min_streng*)

On the contrary, the length of the simulation (*time_steps*) and the decrease of material strength after fluid flow through a site (*decr_stre*) seem to have little effect on the results.

The result makes intuitive sense. Strong variability in material strength allows fluids to concentrate in certain sites. A high amount of fluid flow increases pressure. The strong bias towards anisotropy allows for fluid to flow along preferred paths, and strengthening of material properties with time at sites where no fluid flow occurs reduces the probability of such sites to be fractured in the future. All this results in highly focused fluid flows.

The only result of no immediate intuitive interpretation is the high variability in the *decr_stre* parameter, which controls the decrease in material strength after fluid flows. We would have expected a strong bias towards a high value for this parameter as well. It could be that its effect is compensated by a high value of *incr_stre* making it almost irrelevant.

A similar analysis suggests that the values of the parameters *incr_stre, strength_var* and *%_fluid_flow* are responsible for the differentiation between cluster 2 and 3.

A more quantitative analysis is also possible. The SOM consists of an invertible mapping. To each point in the SOM map we can assign a vector in the original nD space. By back-inverting into the original nD space the boundaries of the clusters identified in Figure 12 we can thus have an approximate understanding of the input parameter ranges characterizing the clusters themselves. These are displayed in Table 1.

Table 1. range of parameter variability within the 3 clusters identified in Figure 12.

	Min_streng	*Press_for_crack*	*%_fluid_flow*	*Strenght_var*	*Incr_stre*	*Decr_stre*	*Time_steps*
Cluster 1	1.37–2.6	1.0–1.8	0.49-0.7	8.52-10.9	3.02–7.0	1.52-2.4	3.90-10.7
Cluster 2	2.63-2.74	1.67-1.75	0.7-0.72	9.72-10.0	6.87-6.9	2.18-2.5	16.3-17.0
Cluster 3	2.53-2.8	0.92-1.67	0.4-0.65	1.9-9.0	4.43-6.0	2.12-2.8	13.47-16.6

There are a few obvious applications of such analysis:

1) if we see that different clusters correspond to different CA behaviors (in this case to different gold mineralization styles), then we can identify which input parameters control the transition between behaviors;
2) if we are not satisfied with the final IGA inversion, and we seek to generate a model with better match to our target image, then it would be reasonable to restrict the further parameter search to the most favorable area(s) of the solution space. In this case we would limit the search to cluster 1 (possibly even only cluster 1_a). This would involve limiting the ranges of the input parameters (which would also allow to make the search resolution finer, should we wish so, without considerably affecting the search computational time);
3) if either the SOM or the user experience and problem specific knowledge suggest points in the parameter space which it is worthwhile sampling, these can be incorporated into the IGA population. Simple mouse clicks in the appropriate location of the SOM map can generate such points via back-inverting the SOM

mapping. This was proposed by Takagi (2000) and applied in geophysical studies in Boschetti and Takagi (2001)

4) if the SOM suggests what the parameter space is under sampled or over-sampled the user could alter the IGA population size on-line. This would help either a poor run to search the parameter space more exhaustively by increasing the population size or would help a very good run to speed up by reducing the population size. Poor runs (especially if they seem characterized by premature convergence) may also be helped by temporarily boosting the mutation rate. A successful example of this approach is shown in Boschetti and Moresi (2001).

Discussion

Interactive inversion was first proposed merely as a substitute for traditional numerical inversion in problems affected by the limitation of lacking a numerical measure of solution misfit. Its not relying on 'hard' data and numerical estimates gives it the image of a 'toy' inversion option, when nothing else is available. Several years experience has shown us that the global optimization of complex, highly non linear, high dimensional geoscientific problems requires considerable computational effort, usually involving GA runs with hundreds of individuals running for hundreds of generations, resulting in tens of thousands function evaluations. In the light of this, the encouraging results obtained in this paper, as well as in other highly non linear mechanical problems (Boschetti and Moresi, 2001, Wijns et al, 2002), with just and handful of individuals running for less that 10 generation (resulting in less than one hundred function evaluations) is, at least, intriguing.

We believe this is mostly due to the ability of the human brain to evaluate several aspects of the IGA solutions (images or animations) at the same time. This, together with the user expertise on the problem at end, provides the IGA search with far more information than it is contained in the single number used as measure of misfit in traditional inversion. Very broadly speaking, it appears the user is able to turn a standard optimization problem into a sort of Multi-Objective Optimization, in which more factors affecting the quality of a solution may be taken into account and evaluated by the human brain. If this interpretation is roughly correct, it may suggest a interesting area for further investigation.

The use of Evolutionary Computation to evolve CA rules was pioneered by Mitchell et al. (1997). The use of Interactive GA to train a CA should also be considered as a first attempt and far more work is necessary before properly evaluating its usefulness. The complexity inherent in certain CAs, as well as their stochastic behavior, provide considerably challenge to any inverse approach. Nevertheless, the difficulty in describing numerically their behavior and the patters they generate (see Wolfram, 1983), not to mention their defiance of most analytical tools, may suggest that an interactive approach can be useful, at least to quickly reconstructs patters which can be identified visually by expert users.

Conclusion

An Interactive Genetic Algorithm can be used to train a Cellular Automata to reconstruct patterns seen in natural geological scenarios. This has been applied to mineralization processes due to fluid inclusions, fracturing and chemical reactions in the earth crust. The information obtained by the IGA sampling, with the help the visualization of the high dimensional search space obtained via a Self Organised Map, can help in segmenting the parameter space and in obtaining a rough understating of which parameters control the different mechanical behavior expressed by the CA.

Reference

Boschetti, F., and Moresi, L., 2001, Interactive inversion in geosciences: Geophysics, v. 66 (4), p. 1226–1234.

F. Boschetti and H. Takagi, Visualisation of EC landscape to accelerate EC conversion and evaluation of its effect, *CEC2001*, Seoul, May 2001.

F. Boschetti, Wijns, C. and L. Moresi, 2002, Effective exploration and visualisation of geological parameter space, *G-cubed*, (*submitted*).

Kohonen, T., 2001, Self-organizing maps: 3rd Edition, New York, Springer, Series in Information Sciences, v. 30, 501 p.

Davis, L., 1991, Handbook on genetic algorithms: Van Nostrand Reinhold.

Mitchell, M., Crutchfield, J., and Das, R., Evolving Cellular Automata to Perform Computations, In T. Bäck, D. Fogel, and Z. Michalewicz (eds.), *Handbook of Evolutionary Computation.* Oxford: Oxford University Press, 1997

Takagi, H., 2000, Active user intervention in a EC search', Int'l Conf. on Information Sciences (JCIS2000), Atlantic City, NJ, USA, p. 995-998.

Takagi, H., 2001, Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation: Proceedings of the IEEE, v. 89 (9), p. 1275–1296.

Wijns, C., Boschetti, F., Moresi, L., 2002, Inversion in geology by interactive evolutionary computation: Journal of Structural Geology, *in press*.

Wolfram, S., 1983, Statistical mechanics of cellular automata, Rev. Mod. Phys., 55, p 1601-644.