Predicting Mining Activity with Parallel Genetic Algorithms

Sam Talaie, Ryan Leigh, Sushil J. Louis
Evolutionary Computing Systems Lab
Dept. of Computer Science and Engineering
University of Nevada, Reno, NV 89557
[talaie,leigh,sushil]@cse.unr.edu

Gary L. Raines
U.S. Geological Survey
MS 176 c/o Mackay School of Mines
University of Nevada, Reno, NV 89557
graines@usgs.gov

ABSTRACT

We explore several different techniques in our quest to improve the overall model performance of a genetic algorithm calibrated probabilistic cellular automata. We use the Kappa statistic to measure correlation between ground truth data and data predicted by the model. Within the genetic algorithm, we introduce a new evaluation function sensitive to spatial correctness and we explore the idea of evolving different rule parameters for different subregions of the land. We reduce the time required to run a simulation from 6 hours to 10 minutes by parallelizing the code and employing a 10-node cluster. Our empirical results suggest that using the spatially sensitive evaluation function does indeed improve the performance of the model and our preliminary results also show that evolving different rule parameters for different regions tends to improve overall model performance.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Global Optimization; I.6.8 [**Types of Simulation**]: Parallel; J.2 [**Physical Sciences and Engineering**]: Earth and Atmospheric Sciences

General Terms

Algorithms, Design, Performance

Keywords

Parallel Genetic Algorithms, Cellular Automata

1. INTRODUCTION

A genetic algorithm was used by Louis and Raines to calibrate a cellular automata that was utilized to model land activity as a result of mining in Idaho and western Montana [13]. Their genetic algorithm (GA) was able to tune the cellular automata (CA) as well as an expert geologist could, with the benefit of performing the tuning in a fraction of the time. The models were created to aide the US Geological Survey (USGS) in providing forecasts of surface disturbances to the US Forest Service (USFS). It is crucial that these forecasts be accurate in part because they will dictate government planning and resource management for the region. We have recognized this need and have explored several techniques to extend previous research and provide more accurate predictions. In this paper, we extend the earlier work in two ways. First we parallelized the code using the LAM-MPI implementation of the Message Passing Interface thus obtaining results in minutes versus hours. This allowed us to more easily explore four different fitness metrics and reasonable combinations of these metrics to improve qualitative performance. Preliminary results show that our best evaluation metric leads to a significant increase in performance of about 155% and that our parallelization efforts have reduced the time needed to perform a simulation by approximately 92%.

The next section discusses the significance of using cellular automata for modeling geological activities and explains the rationale for using genetic algorithms to tune cellular automata parameters. In section 3 we introduce the Kappa statistic for evaluating the performance of the model, because it provides a more robust means of distinguishing between chance agreement and true agreement. Section 4 describes the four different evaluation metrics used by our genetic algorithm and explains our reasoning behind dividing the land into small regions and using the genetic algorithm to evolve different parameters for each region. Section 5 introduces the user interface and system architecture. We compare our results with the results of research done by Louis and Raines in Section 6. The results of research done by Louis and Raines is presented and compared with our results in Section 6. The last section provides conclusions drawn from this research and suggests directions for future work.

2. CELLULAR AUTOMATA AND GENETIC ALGORITHMS

Dadson [4] defines cellular automaton as "a dynamical system, wherein space, time and the states of the system are all represented discretely." He continues to say that the space is defined by a lattice and that each cell in the lattice can assume a finite number of states. The temporal aspect of a CA is exhibited by the transformation of a cell due to a set of rules on its neighboring cells. Cellular automata can be used to model complex behaviors with a set of very simple rules [18]. Due to their ability to capture a subset of reality in both a temporal and a spatial sense, they have

GECCO'05, June 25–29, 2005, Washington, DC, USA.

Table 1: The probabilistic annealed voting rule

No. of active neighbors	Current state	
	Active	**Inactive**
N > top	V. Likely	Likely
bottom < N < top	Likely	S. Likely
N < bottom	V. S. Likely	Unlikely

Table 2: Encoding of the parameters for the GA

Top	Bottom	Likely Inactive	Likely Active	Very Likely	Somewhat Likely	Very Somewhat Likely	Unlikely Probability	RT
4	4	7	7	7	7	7	7	7

been used for modeling in many areas such as urban planning [2, 13, 17], ecological systems, population dynamics [4], and in our case, to model mining permit activity [13, 17]. The behavior of a cellular automata is directly governed by its set of transition rules. In the modeling of mining permit activity, a modified annealed voting rule was employed [13, 17]. The rules of this model are not deterministic, in that they only specify the probability of the cell being in a given state instead of deterministically specifying the state of the cell in the next time step. Overall, there are nine parameters that specify the transition rule space. To achieve reasonable results with the CA for modeling mining permit activity, it may take an expert up to two weeks of tedious work involving a great deal of trial and error. A genetic algorithm can perform the same task with equivalent results within a fraction of the time, approximately 6 hours [17].

Our CA uses a modified annealed voting rule as laid out in Table 1 taken from Louis and Raines [13]. To interpret the table as a set of rules, consider the first row. This defines two rules: (1) **If** N, the number of active neighbors, is above *top* **and** the current state of the center cell is **active**; **Then** set the next state of the center cell to active with a probability corresponding to Very Likely. (2) **If** N, the number of active neighbors, is above *top* **and** the current state of the center cell is **Inactive**; **Then** set the next state of the center cell to active with a probability corresponding to Likely" [13]. Our GA searches the space of possible values for these probability parameters as well as another parameter, the resource threshold (RT). The USGS has provided mineral resource data which can be used to indicate the likelihood of activity on a piece of land. If the resource value of a cell is below what we call the resource threshold, then that cell may be considered as inactive. Table 2 shows the number of bits used to encode these parameters for the GA.

Genetic algorithms are search techniques inspired by biological evolution. They search a given space by examining a population of solutions and then recombining and mutating partial solutions from the given population to create subsequent populations [8, 9]. Though canonical genetic algorithms may not be good function optimizers [5], modified versions such as those with elitist selection have been shown to perform relatively well on function optimization problems [6, 16]. The tuning of the rule set of a CA is an optimization problem which has been shown to benefit from evolution via the use of a GA [2, 10, 13, 14].

Table 3: Agreement measures for categorical data

Kappa Statistic	Strength of Agreement
< 0.00	Poor
0.00−0.20	Slight
0.21−0.40	Fair
0.41−0.60	Moderate
0.61−0.80	Substantial
0.81−1.00	Almost Perfect

Louis and Raines successfully employed a CHC flavored GA to tune their CA model [6, 13, 14].

In a GA, the survival of an individual in the population is proportional to its fitness. Fitness is defined by Goldberg [8] as the "measure of profit, utility, or goodness that we want to maximize." The fitness of an individual is generally determined by an evaluation function. In order to boost the performance of our model, we have explored several different evaluation functions which use linear combinations of different fitness metrics and Kappa statistics as defined in sections 3 and 4. The overall performance of the GA is taken as the highest Kappa value achieved by any member of the population through all generations.

3. KAPPA STATISTIC

In [13], the overall performance of the GA tuned CA and the hand tuned CA were compared based on the total numbers of cells in different states. The number of cells in various states provides a measure of agreement but doesn't account for chance. For example, simply predicting a dead state for all cells will result in a 97% agreement by chance alone. We chose to use the Kappa statistic in our experiments because it provides a means of distinguishing between chance agreement and true agreement. The Kappa statistic is a measurement of agreement between two observers with respect to chance agreement [15]. It is defined by the formula [11, 12, 15]:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{1}$$

where $P(A)$ is the probability of actual agreement and $P(E)$ is the probability of chance agreement.

The overall performance of our model is taken as the highest Kappa value achieved by any member of the population in any generation. Note that the Kappa statistic has two roles, the first of which is the assessment of the overall model. The Kappa statistic can also be used in evaluation functions for assessing the value of individual members of the population. We use the categorical table of Kappa values (Table 3) set forth by Landis and Koch [12] when evaluating the overall performance of the model.

4. EVALUATION HEURISTICS

In the CA model of the land, at any given time, each land cell can be in one of the four following states:

- Alive
- Dead
- Just Born
- Just Died

At any given time, there are significantly more dead (inactive) cells than those with mining activity. This means that if the CA simply adopted a rule that every cell should always be dead, it would be surprisingly accurate. In order to prevent such scenarios, different weights need to be assigned for each state. As in [13], we have also used different weights to constitute equal influence from the four possible states.

The original GA used what we call the Total Cell State Count (TCSC), as the metric for its evaluation function. TCSC minimizes the error between model predicted number of cells in a state and USGS observations [13]. It achieves this by comparing the total number of cells for each state as predicted by the model, and comparing it to the total number of cells for each state based on the observed data. The difference between these two values signifies the error between the model and the observed data. TCSC is described by the following function:

$$\text{Minimize } g = \sum_{j=0}^{nyears} \sum_{i=0}^{nstates} 100 \;\; w_i \frac{\mid M_{ij} - O_{ij} \mid}{M_{ij} + O_{ij}} \tag{2}$$

where O_{ij} is the observed number of cells in state i in year j and w_i is the weight of state i

After close inspection of the TCSC, we realized that we were only considering the total distribution of cell activity while ignoring the spatial distribution. TCSC was simply summing the total numbers of each state and comparing it to values from the observed data. We theorized that better overall results may be achieved if the spatial accuracy of individuals directly impacted their fitness. This is achieved by comparing the state of each cell predicted by the model, to the state of the same cell in the observed data set. If two predictions match, we simply increment the total correct predictions for that particular state. The process is performed for all cells, and then the total correct predictions are offset by the weight of that state and summed up to provide the fitness for the individual. This method is referred to as the NSCP (Number of Spacially Correct Predictions) and is described by the following function:

$$Fitness = \sum_{j=0}^{nyears} \sum_{i=0}^{nstates} w_i M_{ij} \tag{3}$$

where M_{ij} is the number of spatially correct predictions of state i in year j and w_i is the weight of state i

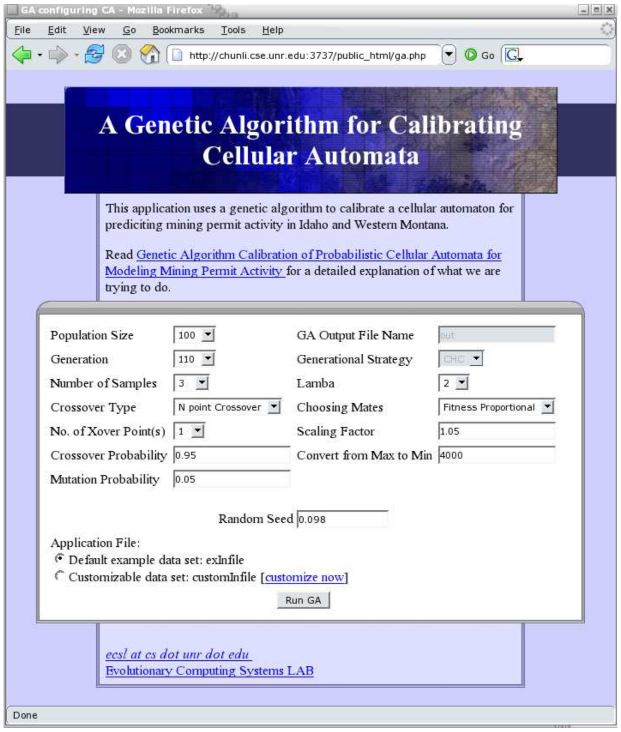
As mentioned in section 3, the Kappa statistic was utilized in some experiments as the sole fitness value for individuals. We assumed that since the overall performance of the model is being evaluated by the maximum Kappa value achieved by any individual throughout all generations, it would be possible to improve performance by using the Kappa value as the sole fitness metric. We also experimented with evaluation functions that utilized linear combinations of the Kappa statistic with TCSC and NSCP values. By combining these metrics we would give the GA more feedback regarding the impact of individuals on the overall performance of the system, thus providing a more accurate measure of fitness for the process of natural selection. The various combinations of these metrics are described below and summarized in Table 4.

TCSC is a measurement of error and a minimization problem, and before it can be used by a GA, it needs to be converted to a maximization problem. This can be achieved by simply subtracting TCSC from a large enough constant [13]. **NSCP** and **Kappa** are both to be maximized so we can use them for measuring fitness directly. In the case of **TCSC and Kappa**, again we subtract from a constant to compute fitness, and in order to properly combine Kappa which is a positive measure of fitness, we take its additive inverse $(1-Kappa)$. For **NSCP and Kappa** (the last row), since both NSCP and Kappa are positive measurements of performance and NSCP is significantly larger than Kappa, we take the product of NSCP and Kappa as the metric for evaluation. Aside from these evaluation functions, we also experimented with subregion modeling.

The motivation behind subregion modeling arose from the divide and conquer concept, whereby it is presumably easier to solve a large complex problem by breaking it into smaller simpler problems. We hypothesized that it may be possible to achieve higher performance by allowing our GA to calibrate CA rules for small subregions of land. To test this hypothesis, we initially experimented with a few sample subregions. The results were quite promising. The high performance on the test subregions led us to believe that perhaps by subdividing the whole region into many smaller regions, and evolving a unique set of parameters (one for each subregion), we could achieve significantly more satisfying results for the entire region. The entire region consists of a 496 x 503 grid of cells each of which is one square mile. We performed three trials, dividing the land into 36, 64, and 100 subregions. For each trial we computed the mean of all the subregions and compared it to the highest attained performance without subregions. Our results are presented in section 6.

5. SYSTEM INTERFACE

We have done substantial work in preparing this project for the end-user. We have built a web interface that allows the user to run simulations with specific parameters for the cellular automata, as well run the GA to find desirable parameters. Figure 1 shows the web interface that allows users to specify GA parameters, upload GIS data, and run the GA. Default values allow users to not have to deal with GA parameters unless necessary. We chose the web interface because it serves as an excellent front-end to the cluster and allows the user to easily access the program from different platforms. We utilized various Perl and Shell scripts to run our application on the cluster. Then we created a PHP driven website to interface with these scripts. All in all, the architecture of our system consists of three parts: the GA and CA simulation which runs on the cluster, a set of scripts to initialize the proper files and launch the simulation, and a web interface to the scripts which allows the users to easily run simulations with complete transparency to the inner-workings of the system. We have also provided the user several tools to monitor the progress of the GA; including, a dynamically updated graph that shows the maximum, average, and minimum Kappa values (see Figure 2). The user can also see dynamic graphs that display the total number of cells in different states.


GA configuring CA - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
http://chunli.cse.unr.edu:3737/public_html/ga.php
Go
Run GA
Done

A Genetic Algorithm for Calibrating Cellular Automata

This application uses a genetic algorithm to calibrate a cellular automaton for prediciting mining permit activity in Idaho and Western Montana.

Read Genetic Algorithm Calibration of Probabilistic Cellular Automata for Modeling Mining Permit Activity for a detailed explanation of what we are trying to do.

Population Size 100
Generation 110
Number of Samples 3
Crossover Type N point Crossover
No. of Xover Point(s) 1
Crossover Probability 0.95
Mutation Probability 0.05
GA Output File Name out
Generational Strategy CHC
Lamba 2
Choosing Mates Fitness Proportional
Scaling Factor 1.05
Convert from Max to Min 4000

Random Seed 0.098

Application File:
● Default example data set: exInfile
○ Customizable data set: customInfile [customize now]

ecsl at cs dot unr dot edu
Evolutionary Computing Systems LAB

Figure 1: Web interface for running the GA to calibrate the CA

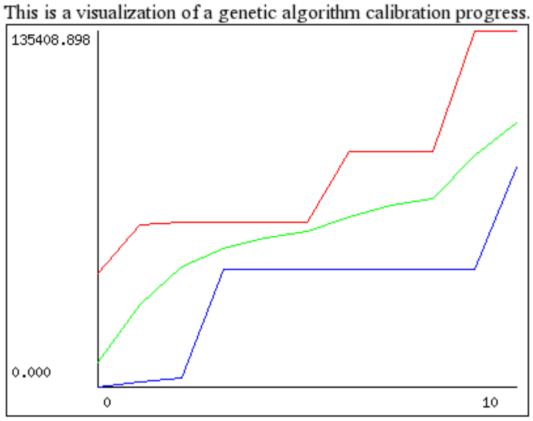

This is a visualization of a genetic algorithm calibration progress.

Figure 2: Interface that displays the maximum, average, and minimum Kappa values per generation

Table 4: Various fitness functions used by the genetic algorithm

Function Name	Fitness
TCSC	$Constant - TCSC$
NSCP	$NSCP$
Kappa	$Kappa$
TCSC and Kappa	$Constant - (TCSC \times (1 - Kappa))$
NSCP and Kappa	$NSCP \times Kappa$

Table 5: Performance of different evaluation functions in terms of mean Kappa value

Evaluation Functions	Mean Kappa Value
TCSC	0.2814
Kappa	0.4362
NSCP	0.3154
TCSC and Kappa	0.4356
NSCP and Kappa	0.4366

This allows users to compare the GA predicted number with the ground truth data enabling them to visually monitor the performance of the GA in real-time.

6. RESULTS AND ANALYSIS

We tested all the evaluation functions presented in Table 4. For these experiments we performed 10 runs with different random seeds and for the GA, we used a population of 60 individuals over 60 generations with a crossover rate of 0.99 and a mutation rate of 0.05.

Table 5 shows the mean of the highest achieved Kappa values and Figure 3 shows the results of the 10 runs with 95 percent confidence intervals for each evaluation function. Based on these observations it is clear that the use of Kappa values in the fitness evaluation does improve overall fitness. It is also clear that NSCP performs significantly better than the TCSC. Figure 4 is a visualization of these runs where highest Kappa value of each generation is averaged over the 10 runs and graphed as a function of the generation. In the 30 runs where maximum Kappa was larger than 0.4, it appears that a Kappa value of 0.437 is the absolute highest value attainable by our GA (without the use of subregions). Because of the large number of dead cells, this Kappa value represents fairly good performance - equaling or bettering human expert performance.

6.1 Subregion Modeling Results

Table 6 shows the results of the three different subregion trials. Due to the significant time constraints of these simulations, each simulation was only performed once. The results for these runs may not be statistically significant, however they are quite promising. It is definitely clear that higher Kappa values may be achieved for certain subregions defying the 0.437 barrier. Also, we can show from these runs that dividing the problem into smaller pieces can indeed increase performance. The average Kappa value for the three subregion runs was 0.4766 while the previously highest attained value was 0.4366. More investigation is necessary to find the ideal size of a subregion for our GA, and more data is needed to bolster these preliminary findings.

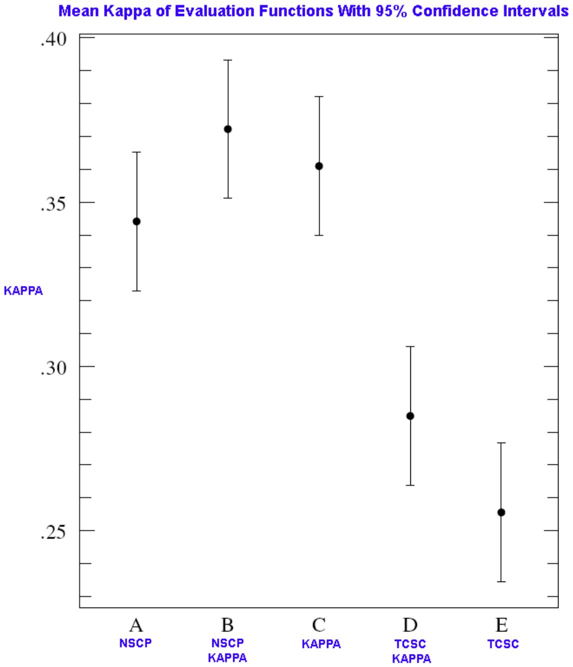


Figure 3: 10 Run Average of different evaluation functions with 95 percent confidence intervals

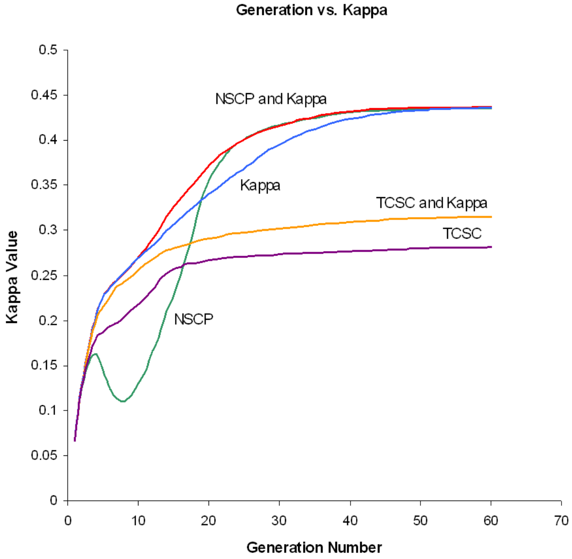


Figure 4: Performance of different evaluation functions in terms of mean Kappa value, per generation of the GA.

Table 6: Average and maximum Kappa values attained by different number subregions. Subregions with Kappa values of 0.000 (due to inactivity over the entire run of the simulation) were excluded from these computations.

Subregions	Mean Kappa	Max. Kappa
36	0.48125	0.997
64	0.48163	0.988
100	0.46696	1.000

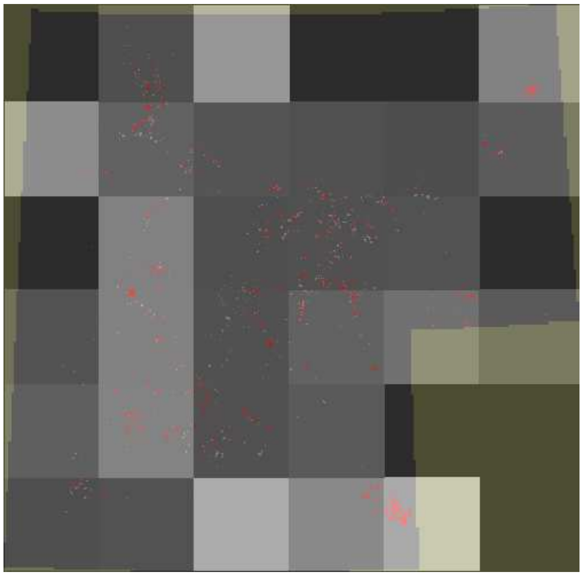


Figure 5: Visualization of a run consisting of 36 subregions

For these runs we used the NSCP and Kappa evaluation function and ran the GA once with a population of 50 individuals over 50 generations with a crossover rate of 0.95 and a mutation rate of 0.05.

In order to better understand the performance of the different subregions, we superimposed a visualization of all the active cells with an image representing the performance of all the subregions. The higher the Kappa value of the subregion, the brighter that subregion is depicted. The cell activity which is highlighted in these images by red pixels, is of less importance. Figure 5, Figure 6, and Figure 7 represent the images for the 36-region run, 64-region run, and 100-region run, respectively. The regions that are completely black are regions with Kappa values of 0.000 due to inactivity over the entire run of the simulation. These figures are helpful in that they allow us to locate and better understand the regions of the land where the performance of our model is less satisfactory.

7. CONCLUSION AND FUTURE WORK

We have built a system for the USGS that allows for quicker and more accurate predictions of public land use. We have made improvements to the genetic algorithm that provides adequate modeling parameters to the cellular automata which is the model for the simulation of the mining activity. By parallelizing the code and employing a 10-node

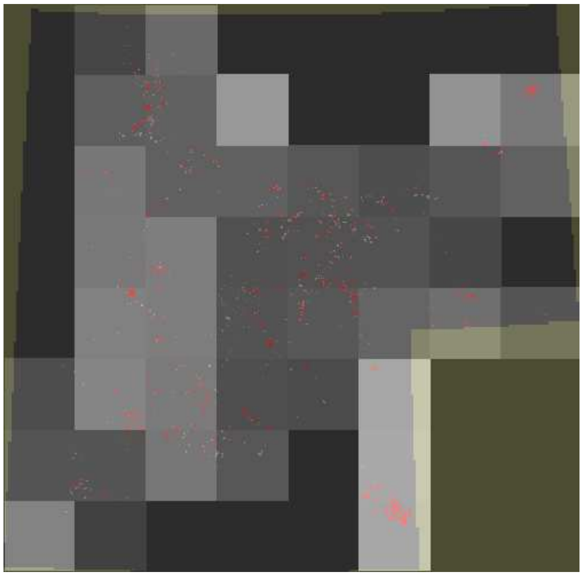


Figure 6: Visualization of a run consisting of 64 subregions.

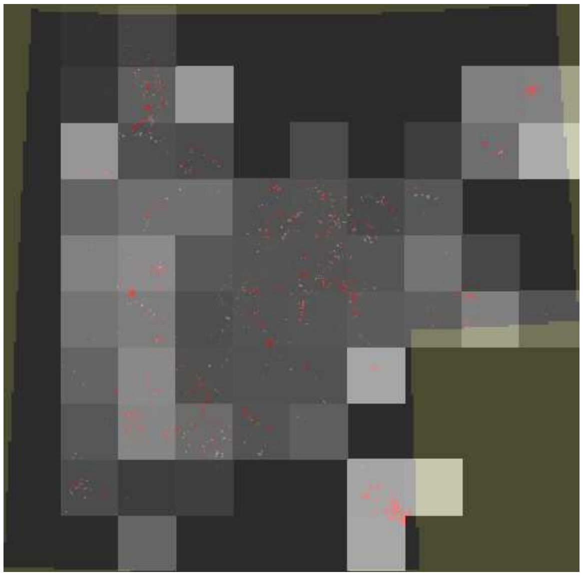


Figure 7: Visualization of a run consisting of 100 subregions.

NPACI Rocks Cluster, we have drastically reduced the time needed to perform the tuning of the CA. We have experimented with several new evaluation functions and, by allowing the GA to examine specific subregions in parallel, we have surpassed previous ceilings on performance.

Our empirical results suggest that incorporating spatial information into the evaluation function does improve the overall performance of the model. Also, it is clear that using the Kappa statistic within the two NSCP and TCSC evaluation functions boosts performance.The evaluation functions with NSCP, NSCP and Kappa, as well as Kappa statistic as the sole evaluation measure, were all able to achieve a "moderate" strength of agreement, as set forth by Landis and Koch [12], while the older TCSC and TCSC with Kappa statistic retained their "fair" strength of agreement. We refrain from making any conclusions about the subregion runs in regards to overall performance due to lack of data needed for statistically sound conclusions. However, from these preliminary runs, we can conclude that the 0.437 Kappa value may be surpassed for individual subregions. At this point we believe that we have exhausted the possibilities of the current GA, and therefore, we need to look in other directions in order to improve the overall performance of our model. Possible areas include exploring rules other than the annealed voting rule for the CA, using the genetic algorithm to evolve the actual CA rules instead of just the rule parameters, and investigating the possible causes of error in the model. Our current findings may be beneficial in other domains such as urban growth modeling [1], disease simulation [7], invasive species modeling [3], and any other spatial-temporal, cellular automata based model.

8. ACKNOWLEDGMENTS

This material is based in part upon work supported by the USGS and in part supported by contract number N00014-03-1-0104 from the Office of Naval Research.

9. REFERENCES

[1] T. Bäck, H. Dörnemann, U. Hammel, and P. Frankhauser. Modeling urban growth by cellular automata. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 636–645, Berlin, 1996. Springer.

[2] K. C. Clarke, S. Hoppen, and L. Gaydon. A self-modifying cellular autmaton model of historical urbanizaion in the san francisco bay area. *Environment and Planning B: Planning and Design*, 24:247–261, 1997.

[3] V. Cole and Albrecht. Modelling the spread of invasive species: parameter estimation using cellular automata. In *Proceedings Second International Workshop on Dynamic and Multi-Dimentional GIS (DMGIS '99')*, 1999.

[4] S. Datson. Cellular automata and gis - geog516 presentation feb 1999, 2 1999. http://www.geog.ubc.ca/courses/geog516/notes/catalk.html.

[5] K. A. DeJong. Genetic algorithms are NOT function optimizers. In L. D. Whitley, editor, *Proceedings of the Second Workshop on Foundations of Genetic Algorithms*, pages 5–18, San Mateo, July 26–29 1993. Morgan Kaufmann.

[6] L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. J. Rawlins, editor, *Foundations of genetic algorithms*, pages 2665–283. Morgan Kaufmann, San Mateo, CA, 1991.

[7] S. C. Fu and G. Milne. A flexible automata model for disease simulation. In *ACRI*, pages 642–649, 2004.

[8] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[9] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, 1992.

[10] W. Hordijk, J. P. Crutchfield, and M. Mitchell. Mechanisms of emergent computation in cellular automata. Working Papers 98-04-034, Santa Fe Institute, Apr. 1998. available at http://ideas.repec.org/p/wop/safiwp/98-04-034.html.

[11] N. Komagata. Chance agreement and the significance of the kappa statistic. http://www.tcnj.edu/ komagata/pub/Kappa.pdf.

[12] R. J. Landis and G. G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33:159–174, 1977.

[13] S. J. Louis and G. L. Raines. Genetic algorithm calibration of probabilistic cellular automata for modeling mining permit activity. In *ICTAI '03: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, page 515. IEEE Computer Society, 2003.

[14] M. Mitchell, J. Crutchfield, and R. Das. Evolving cellular automata with genetic algorithms: A review of recent work, 1996.

[15] M. U. of South Carolina. What is kappa, 11 2000. http://www.musc.edu/dc/icrebm/kappa.html.

[16] D. Parekh, J. Freund, and P. Koumoutsakos. Evolution strategies for parameter optimization in jet flow control, Feb. 22 1999.

[17] G. L. Raines, M. L. Zientek, J. D. Causey, and D. E. Boleneus. Preliminary cellular-automata forecast of permit activity from 1998 - 2010, idaho and western montana. *Natural Resources Research*, to appear.

[18] S. Wolfram. *A New Kind of Science*. Wolfram Media Inc., 2002.