

A simple model for the earthquake cycle combining self-organized complexity with critical point behavior

W. I. Newman¹ and D. L. Turcotte²

¹Departments of Earth and Space Sciences, Physics and Astronomy, and Mathematics, University of California, Los Angeles, CA 90095, USA

²Department of Earth and Atmospheric Sciences, Snee Hall, Cornell University, Ithaca, NY 14853, USA

Received: 2 April 2002 – Accepted: 22 May 2002

Abstract. We have studied a hybrid model combining the forest-fire model with the site-percolation model in order to better understand the earthquake cycle. We consider a square array of sites. At each time step, a “tree” is dropped on a randomly chosen site and is planted if the site is unoccupied. When a cluster of “trees” spans the site (a percolating cluster), all the trees in the cluster are removed (“burned”) in a “fire.” The removal of the cluster is analogous to a characteristic earthquake and planting “trees” is analogous to increasing the regional stress. The clusters are analogous to the metastable regions of a fault over which an earthquake rupture can propagate once triggered. We find that the frequency-area statistics of the metastable regions are power-law with a negative exponent of two (as in the forest-fire model). This is analogous to the Gutenberg-Richter distribution of seismicity. This “self-organized critical behavior” can be explained in terms of an inverse cascade of clusters. Small clusters of “trees” coalesce to form larger clusters. Individual trees move from small to larger clusters until they are destroyed. This inverse cascade of clusters is self-similar and the power-law distribution of cluster sizes has been shown to have an exponent of two. We have quantified the forecasting of the spanning fires using error diagrams. The assumption that “fires” (earthquakes) are quasi-periodic has moderate predictability. The density of trees gives an improved degree of predictability, while the size of the largest cluster of trees provides a substantial improvement in forecasting a “fire.”

1 Introduction

Complexity commonly describes many geophysical systems, although complexity itself is not simple to define. Complexity describes the nonlinear interactions in systems with a large or an infinite number of degrees of freedom. Complexity manifests itself in linkages between space and time,

generally producing patterns on many scales, and often including a hierarchy of interactions, cascades—both direct and inverse—and the emergence of fractal structures. Complex systems are generally sensitive to initial conditions, and exhibit a transition from order to chaos. Overall, complexity describes situations where the nature of the nonlinear interactions cause the whole to be greater than the sum of its parts.

Complexity in the earth often possesses other characteristic features. In many situations, we are dealing with systems that are driven by an external mechanism, e.g., plate tectonics in the case of earthquakes, and that the process is irreversible. In many such problems, temperature does not play an explicit role, so temperature cannot be directly regarded as a parameter. Commonly, these geophysical problems are scale-invariant over a large range of scales. Power-law scaling is endemic in such systems. While scale-invariance is a necessary condition for power-law scaling, it is not sufficient and a profound question that emerges is to explain *why* do power-laws emerge, *what* is the cause of the self-organization that is manifest. We will refer to such problems as manifesting self-organized complexity.

Bak et al. (1988) introduced a concept that they termed “self-organized criticality,” in order to explain the behavior of their “sandpile” model. Two other models that were found to exhibit this type of behavior were the slider-block model (Burridge and Knopoff, 1967; Carlson and Langer, 1989), and the “forest-fire” model (Bak et al., 1992; Drossel and Schwabl, 1992). A feature of these models is that they involve “avalanches” with a power-law (fractal) frequency-size distribution. There is a steady-state “input” and the “output” occurs in the “avalanches.”

The use of the word “criticality” to describe the behavior of these models has led to considerable controversy. The formal definition of a critical phenomena when introduced by physicists concerns Hamiltonian systems where the temperature plays a central role in the calculation of the partition function in statistical mechanics (from which all thermodynamic quantities can be calculated), a feature that is absent from the formulation of the problems which we wish to ad-

dress. Also, defining a critical point requires the “tuning” of a control parameter. In the forest-fire model, which had been cited by some as a paradigm for so-called “self-organized criticality,” Gabrielov et al. (1999) have shown analytically that the behavior of the model is the asymptotic outcome of the evolution of a hierarchical set of ordinary differential equations. The behavior is that of an inverse cascade of clusters of trees. Small clusters coalesce to form larger clusters. Fires terminate the cascade, and there is no tuning parameter. Accordingly, what has come to be called “self-organized criticality” by some, we will call without prejudice “self-organized complexity” to avoid any conflict with physicists’ use of these words. The discovery by Bak et al. (1988) that simple, discrete models could produce power-law scaling represents an important milestone in scientific inquiry, but their use of the word “criticality” has caused unnecessary controversy which we seek to avoid. However, it is noteworthy that certain probabilistic distribution functions (with no thermodynamic basis) may asymptotically become step functions, e.g. site percolation on a lattice; we will confine our use of the words “criticality” and “critical point” to this mathematical setting.

Self-organized complexity has been associated with several natural hazards (Malamud and Turcotte, 1999). For earthquakes, the steady-state input is the tectonic stress accumulation. Thus stress is relieved in earthquakes (i.e., avalanches) which satisfy a power-law (Gutenberg-Richter) frequency-magnitude distribution. For landslides, slope instabilities develop slowly and are relieved in the landslides. There is accumulating evidence that landslides may be described by power-law distributions (Guzzetti et al., 2002). For forest- and wild-fires, combustible material grows on long time scales, and is destroyed in fires on short time scales. The frequency-area distribution of forest and wild fires is well-approximated by power-law distributions (Malamud et al., 1998).

For the three natural hazards described above, hazard assessment and event forecasting are of great importance. The applicability of power-law frequency-size distributions can be directly applied to hazard assessment. The rate of occurrence of smaller events can be extrapolated to give the expectation of larger events. This is routinely done for earthquakes (Kossobokov et al., 2000). The forecasting of individual events, however, remains extremely controversial.

The purpose of this paper is to examine the forecasting statistics of a hybrid model in which the events to be forecast are well defined, and the model contains the basic aspects of self-organized complexity. The model is a hybrid between the forest-fire model and the site-percolation model. Before introducing the details of our model, we briefly review the relevant version of the forest-fire model and the site-percolation model.

The forest-fire model we consider consists of a square grid of sites. At each time step, we either attempt to plant a tree on a randomly chosen site, or we drop a match on the site. A tree is planted if a site is not occupied by a tree. If a match is dropped on a site occupied by a tree, that tree and all adjacent

trees are destroyed in a model fire. The sparking frequency is the inverse of the number of attempted tree drops before a model match is dropped. It is found that the frequency-area distribution of fires over a range of sizes is power-law with a slope near unity. The frequency-area distribution of trees clusters is also power-law with a slope near two. This difference in slope follows since the probability that a match will ignite a cluster is proportional to the area of the cluster. The site-percolation model can be regarded as the forest-fire model without fires. This is a transient problem as the number of trees on the grid increases monotonically until every site is occupied by a tree. However, there is a well-defined critical point during this evolution. This critical point occurs when a cluster of trees spans the grid. Monte Carlo simulations with large grids show that a spanning cluster is formed when 59.275% of the grid sites are occupied by trees (Stauffer and Aharony, 1992). At this critical point, the frequency-area distribution of tree clusters is power-law with a slope of 2.055. This similarity to the forest-fire model only occurs at the critical point. At other times during the transient planting of trees, the distribution is not power-law (Turcotte, 1999). The tuning parameter in this critical-point problem is the fraction of sites occupied ρ . The critical value of this tuning parameter is $\rho_c = 0.59275$.

2 The model

In this paper, we introduce a model that combines aspects of both the forest-fire and site-percolation models. We impose the condition that a model fire occurs when a cluster spans the grid from one side to the other (either horizontally or vertically). An attempt is made to plant a tree at each time step. If the randomly chosen site is unoccupied, the tree is planted. When a tree cluster spans the grid, it then ignites and all the trees in that cluster are removed from the grid. This is a quasi steady-state model like the forest-fire model, but it has only large (characteristic) fires. In the past, substantial progress in percolation theory was made as a consequence of the development of special-purpose computational algorithms. Similarly, in order to perform the large number of Monte Carlo simulations required in our model, we found it necessary to develop special purpose methods for this problem based on principles widely used in the computer science literature, but not well-known in physics-based disciplines. These algorithms are described in Appendix A.

Six typical grid-spanning fires are shown in Fig. 1. These simulations were run on a 128×128 grid. The red grid sites represent the fires that destroy the spanning clusters, the green grid sites are occupied by trees that are not part of the spanning cluster, and the white sites are unoccupied. The fire in Fig. 1a spans the grid horizontally, while the other five examples in the figure span the grid vertically. The spanning cluster can be formed in two ways. A tree is planted that connects a cluster to the boundary, as in the example in Fig. 1a. Usually, however, the spanning cluster is formed

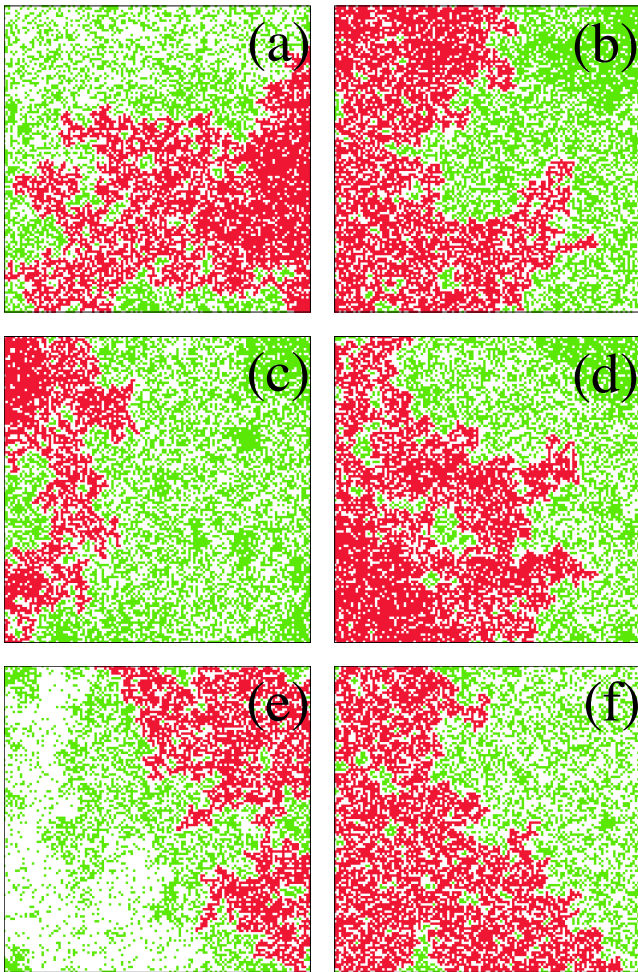


Fig. 1. Six examples of typical grid-spanning (percolating) fires on a 128×128 grid. The red sites represent trees burned in the fire, green sites unburned trees, and white sites are unoccupied.

when the gap between two relatively large clusters is bridged by a planted tree, the other examples in Fig. 1.

The evolution of the model between two typical fires is illustrated in Fig. 2. A typical grid-spanning fire is illustrated in Fig. 2a. The distribution of unburned trees immediately after the fire is given in Fig. 2b. A few isolated small clusters within the fire zone have survived the fire. The distribution of trees at approximately 25% of the time until the next spanning fire is shown in Fig. 2c. The burned area is being filled by small clusters of trees. The distribution of trees at approximately 50% of the time until the next spanning fire is shown in Fig. 2d and the distribution at approximately 75% is shown in Fig. 2e. The burned area is gradually filled with clusters of trees. The next grid-spanning fire is shown in Fig. 2f. This fire burns principally in the area that was not burned during the previous fire.

The percentage of fires with areas smaller than a specified value is given in Fig. 3. The mean size of the spanning fires is 4180 or 25.5% of the 16384 grid sites. This compares with the 59.275% or 9712 grid sites that would be

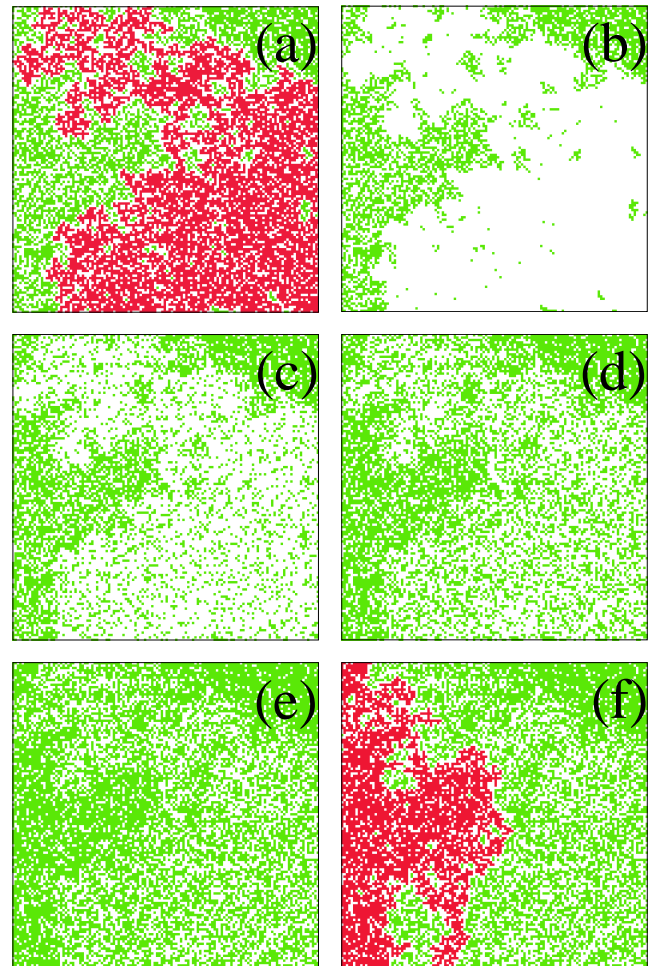


Fig. 2. Evolution of the model between two grid-spanning fires. (a) The initial spanning fire is illustrated; (b) Distribution of unburned trees just after the fire; (c) Distribution of trees $\approx 25\%$ of the time until the next spanning fire; (d) Distribution of trees $\approx 50\%$ of the time until the next fire; (e) Distribution of trees $\approx 75\%$ of the time until the next fire; and (f) The next spanning fire is illustrated.

found for a spanning cluster in the site percolation model. Put another way, a grid-spanning fire is initiated when the overall number of occupied sites is near 59.275% of the total available but only 25.5% of the total available are in the particular cluster that is consumed by the fire. The spanning fires also have considerable variability of size ranging from about 2000 to 8000 burned trees whereas in the site percolation model the spanning clusters have little variability about the critical value of 9712 grid sites. The smaller mean spanning cluster and the variability of cluster sizes in our model relative to the site percolation model is easily explained. In the site percolation model, the density of occupied sites is nearly uniform across the grid, while in our model there are considerable variations in density as illustrated in Fig. 2. The reason for this variability is the residual memory of previous fires.

The noncumulative distribution of cluster sizes is given in

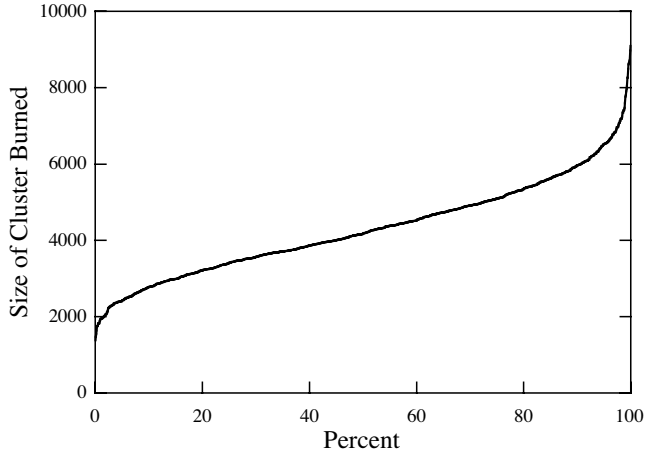


Fig. 3. The percentage of fires smaller than a specified value is shown. The mean size of all fires is 4180 grid sites.

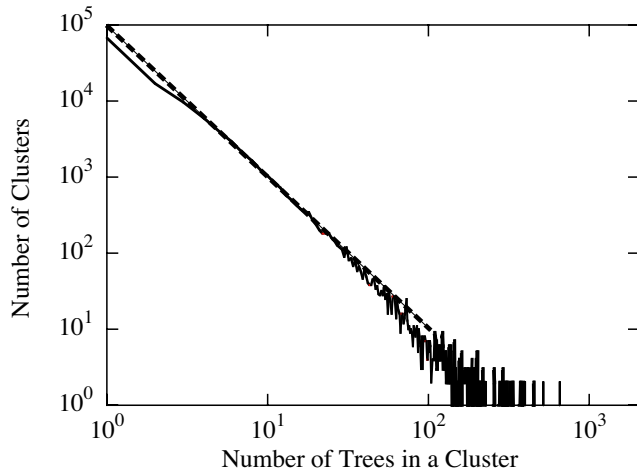


Fig. 4. The noncumulative distribution of cluster sizes. The mean number of clusters of integer sizes is given as a function of cluster size. A time average has been taken. Dashed line has slope of -2 .

Fig. 4. The mean number of clusters of integer sizes N_c is given as a function of the cluster area A , number of trees in a cluster. A time average over many fire cycles has been taken. The distribution of cluster sizes given in Fig. 4 correlates very well with the power-law distribution

$$N_c = CA^{-\alpha} \quad (1)$$

taking $C = 10^5$ and $\alpha = 2$. This distribution of cluster sizes is the same as that found for the forest-fire model (Turcotte et al., 1999).

This power-law distribution of cluster sizes can be explained by the inverse cascade model introduced by Turcotte et al. (1999), Turcotte (1999), and Gabrielov et al. (1999), as well as Newman and Knopoff (1982, 1983, 1990), and Knopoff and Newman (1983). In this cascade model, smaller tree clusters coalesce to form larger clusters. On average, individual trees found in small clusters which evolve into larger

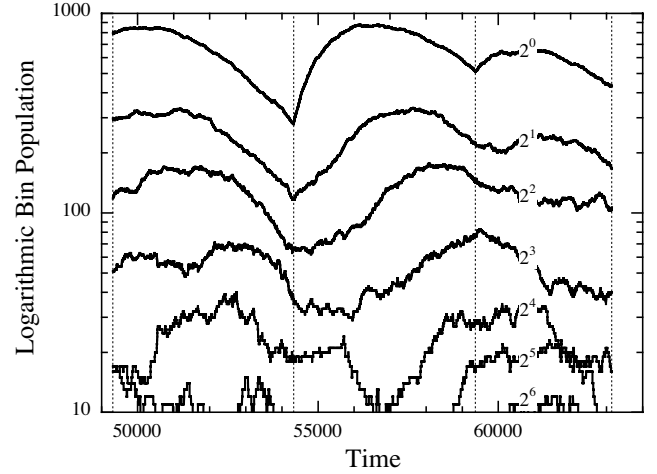


Fig. 5. Dependence of the number of clusters of a specified size upon time for the time-intervals between four fires. The vertical dashed lines are when the fires occurred. The distribution of cluster areas has been logarithmically binned in bins of size 2^n .

and larger clusters until they burn and are destroyed in the large spanning clusters. The inverse cascade model predicts the inverse slope of 2 shown in Fig. 4.

The populations of clusters of various sizes as a function of time are given in Fig. 5. We have used a logarithmic binning of the distribution of cluster sizes. We consider bins of size 2^n , where n is an integer. The bin $n = 0$ contains clusters with $2^0 = 1$ tree, $n = 1$ contains clusters with 2 and 3 trees, and $n = 2$ contains clusters with 4, 5, 6, and 7 trees. In general, the n^{th} cluster in the hierarchy can have $[2^n]$, $[2^n + 1]$, ..., $[2^{n+1} - 1]$ trees. The binned populations for $n = 0$ through 6 are given in Fig. 5 for the time intervals between four fires. Consider the time dependence after the second fire. There is a rapid increase in single tree clusters, as planted trees fill in the burned areas. As the burned area fills with tree clusters, the number of single tree clusters decreases. A similar behavior is seen for the larger clusters but with a time delay. This behavior is explained by the inverse cascade described above. As the burned area fills in, the tree clusters coalesce to form large clusters, and individual trees migrate from small to large clusters. The behavior illustrated in Fig. 5 is also found (Turcotte, 1999) for both the forest-fire model and the transient forest-fire model without fires (i.e. site-percolation model).

3 Forecasting events

The spanning fires in this model occur at quasi-periodic time intervals. There is certainly a stochastic component since the selection of sites is random. However, repeated runs of the site-percolation model leads to a sequence of percolating clusters that are, to a good approximation, periodic. Because of memory, however, there is considerable variability in the intervals between events in our model.

If the forest-fire events were periodic, then the events could be forecast exactly based on the time when the previous event occurred. This is not the case when there is a stochastic component in the site-selection process. We will consider a number of ways of forecasting the occurrence of spanning fires in our model. It is also necessary to evaluate the value or the effectiveness of the forecasts. At a specified time, a forecast of a future event will be made. If the event occurs prior to the forecast, it is deemed a failure to predict. If the event occurs after the forecast time, the prediction is deemed a success. One measure of the value of a forecast is the fraction of events that are successfully predicted. If 90% of the events can be successfully forecast, this approach would be quite good for many applications, including the forecasting of an earthquake, assuming that the “cost” associated with the forecast is not excessive.

However, there is an important second measure of success. This is the fraction of time during which an “alarm” is in effect prior to the occurrence of a forecast event; we refer to this as the fraction of the total time that the forecast is in effect or, simply, the alarm time. If a forecast is made immediately after an event, each and every subsequent event is forecast, but the forecasts have no value since the alarm is activated at all times, i.e. the fraction of alarm time would be 1. Successful forecasts would be made very shortly before the occurrence of an event, and would be made before essentially all events. For earthquakes a challenging goal would be to forecast 90% of events with the fraction of alarm time 10% or less.

A quantitative measure of the success of a forecasting algorithm is the “error diagram.” This is a plot of the fraction of alarm time versus the fraction of failures to forecast. The error diagram was introduced to earthquake forecasting by Molchan (1991, 1997). We illustrate the use of the error diagram by considering a specific example.

The simplest forecasting algorithm for quasi-periodic events is to make a forecast of the future event at a specified time interval τ after the previous event. The results of this forecasting algorithm for our model are given in Fig. 6. The curve gives the trade-off between the fraction of alarm time and the fraction of failures-to-predict as a function of the interval time. If a forecast is made immediately after an event, the fraction of alarm time is 1 but the fraction of failures to forecast is zero. The best forecast using this method would have the fraction of alarm time near 0.3 and the fraction of failures to forecast also near 0.3. The diagonal line in this diagram gives the result of this prediction algorithm if the occurrence of events is completely random. This behavior is derived in Appendix B.

The second forecasting algorithm is based on the density of trees on the grid, N_t/N_g , where N_g is the number of sites and N_t is the number of trees. The results of this forecasting algorithm are given in Fig. 7. The best forecast would have the fraction of alarm time near 0.3, and the fraction of failure to forecast near 0.2. The forecasts using this algorithm are somewhat better than the interval forecasts given in Fig. 6. The explanation is that some fires leave more residual trees

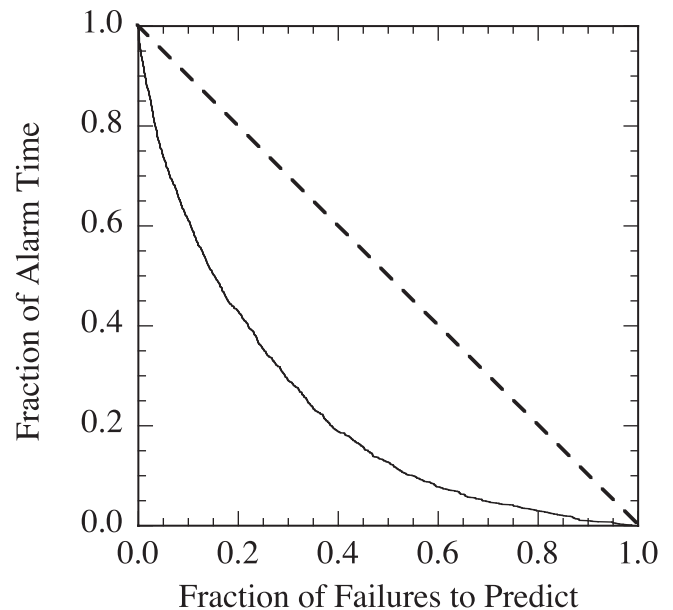


Fig. 6. Error diagram for event forecasts based on the time of previous events. Forecasts are made at various times τ after an event. The fraction of alarm time is given as a function of the fraction of failures to predict. The dashed diagonal line gives the result of this prediction algorithm if the occurrence of events is completely random, i.e. Poissonian.

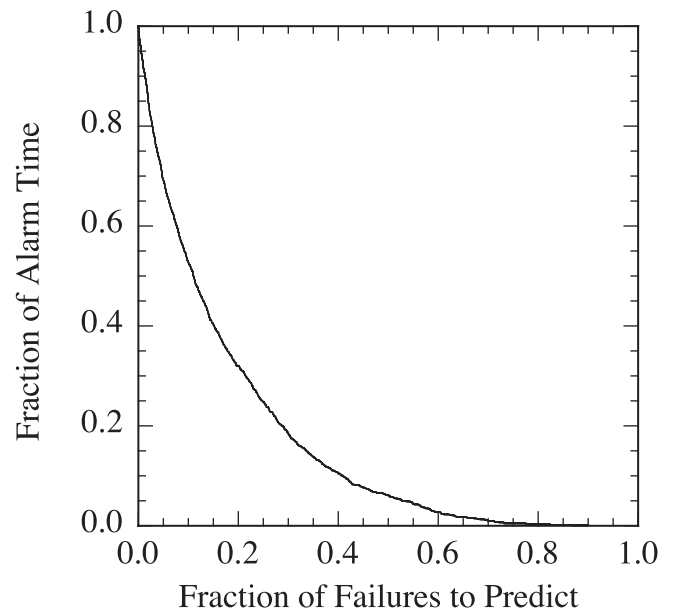


Fig. 7. Error diagram for event forecasts based on the density of trees on the grid. The fraction of alarm time is given as a function of the fraction of failures to predict.

than other fires, on average the time interval times after severe fires are greater than interval times after smaller fires.

The final forecasting algorithm is based on the fractional size of the largest cluster, N_{ct}/N_g where N_{ct} is the number of trees in the largest cluster. This is shown in Fig. 8. The best

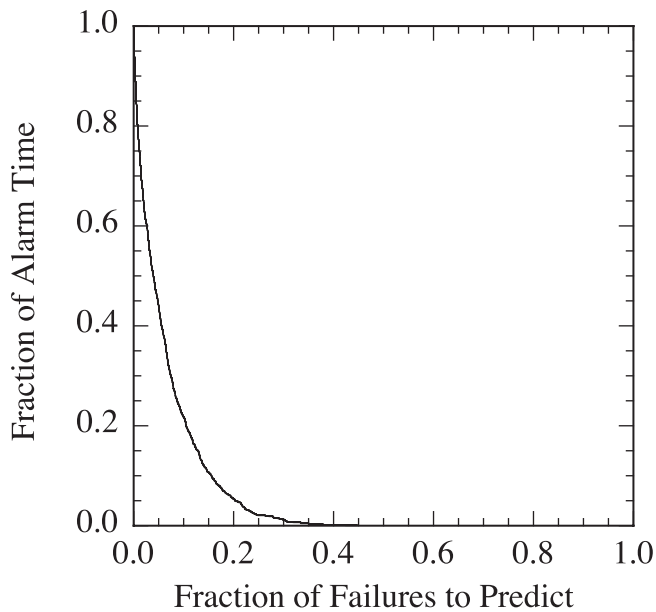


Fig. 8. Error diagram for event forecasts based on the fractional size of the largest cluster. The fraction of alarm time is given as a function of the fraction of failures to predict.

forecast would have the fraction of alarm time near 0.2 and the failures to forecast near 0.1. This is the best forecasting algorithm with quite good forecasts.

4 Conclusions

We have studied the behavior of a model that combines aspects of the forest-fire model and the site-percolation model. In our model, trees are randomly planted until a cluster of trees “percolates,” that is the cluster spans the grid from one side to the other. The trees in the spanning cluster are immediately destroyed in a model “fire” and the planting sequence is continued until the next fire occurs.

If trees are planted randomly on a large empty grid, the site-percolation model is recovered. If all trees were removed in a model fire when a cluster spans the grid, the fires would be nearly periodic and the number of trees “burned” in each fire would be nearly equal. In our model, the variations in the number of residual trees after a fire results in a memory that influences the occurrence of future fires. There is considerable variability in fire sizes and the intervals between fires.

We believe that the fires in our model are somewhat analogous to characteristic earthquakes on a major fault or fault segment. The planting of trees corresponds to the tectonic increase in stress. The variations in cluster structure correspond to stress variations in the crust. The variability of the sizes is analogous to the variability in characteristic earthquakes and the associated variability in the intervals between characteristic earthquakes. In our model, there is a memory of the structure of previous fires in any given fire, each char-

acteristic earthquake is influenced by the structure of previous characteristic earthquakes.

Spanning clusters are created by the coalescence of smaller clusters. This coalescence process is a self-similar inverse cascade in which the power-law distribution of cluster sizes has been shown to have an exponent of two. The clusters of trees are analogous to the metastable regions on a fault over which our earthquake will propagate once triggered.

In our analogy with earthquakes, we are considering characteristic earthquakes on a single fault or fault segment, *not* regional seismicity. The Gutenberg-Richter frequency-magnitude distribution of earthquakes is associated with earthquakes on many faults. We are considering the behavior of a single fault or fault segment that experiences characteristic earthquakes. Examples would be the northern and southern segments of the San Andreas Fault in California. There are very few earthquakes occurring on these fault segments but future characteristic earthquakes similar to the 1906 and 1857 events are expected.

We have introduced several algorithms for forecasting when a model fire will occur. We have quantified the forecasting success of an algorithm in terms of an error diagram. The fraction of alarm time is given as a function of the fraction of failures to forecast. A successful forecast should have a small fraction of alarm time – ideally less than 0.1 – and a small fraction of failures to forecast – again ideally less than 0.1.

Our first forecasting algorithm issues an alarm at a specified time after the previous event, say at the mean interval time for events. If the events were periodic, this forecasting algorithm would give perfect results. This algorithm is only modestly successful on our model. It gives 30% failures to forecast, with 30% alarm time. If the events were randomly spaced, then the method would have 50% failure to forecast, with 50% alarm time.

This algorithm is analogous to the seismic gap approach to earthquake forecasting. The essential question is how one characteristic earthquake is related to previous characteristic earthquakes. If they are random (i.e. Poissonian), then there can be no forecasting. If they are periodic, then there can be perfect forecasting. Actual earthquakes fall somewhere in between, just as our model does.

Our second forecasting algorithm issues an alarm based on the density of trees on the grid. This is an improved algorithm, because the residual number of fires varies from fire to fire. The final forecasting algorithm issues an alarm based on the size of the largest cluster on the grid. This algorithm is quite successful because the size of the largest cluster is closely related to the development of a percolating cluster. This algorithm gives 10% failure to predict with a 20% alarm time.

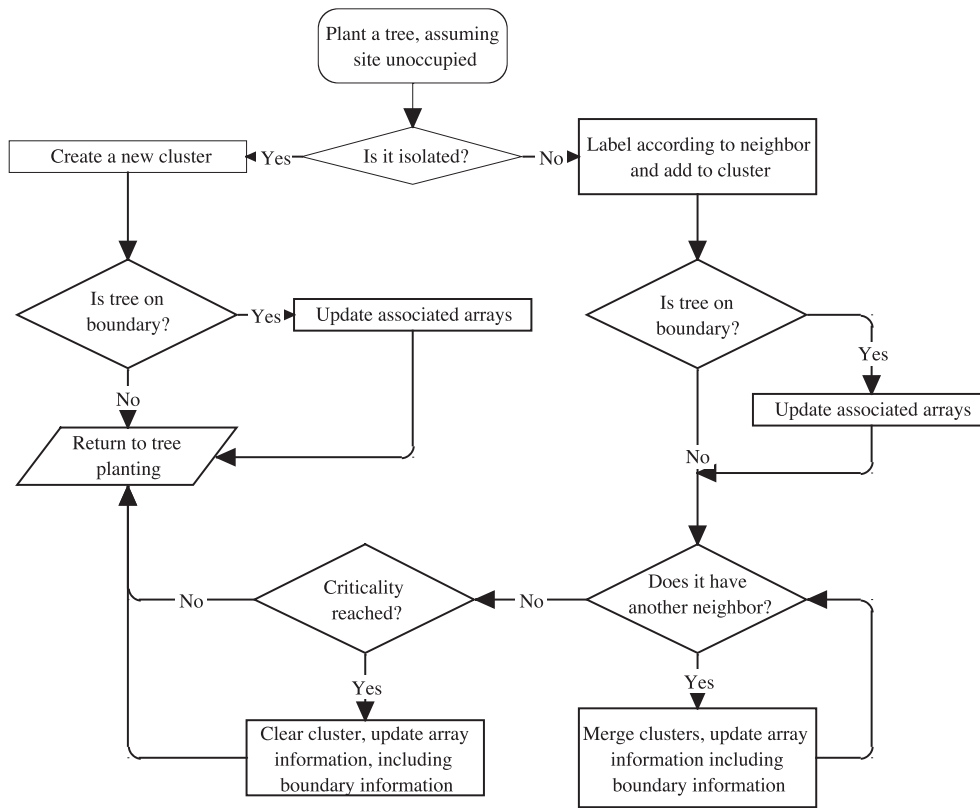


Fig. A1. Flowchart for the algorithm employed in the dynamic simulation of our modified forest-fire model.

Appendix A Computational algorithms

The physics community has appreciated the need for special-purpose algorithms for percolation problems for many years. In the case of site-percolation, Hoshen and Kopelman (1976) developed a scheme that made it possible to identify the size of the largest cluster in a square array where the sites had been occupied according to a random number generator, while Leath (1976) employed a rather different methodology. Stauffer and Aharony (1992) present a summary of many of the ideas employed in such schemes. Our problem, however, is distinctly different from those treated above in that our lattice is undergoing continuous updating, through the planting of trees. While the ideas described by Hoshen and Kopelman (1976) and Stauffer and Aharony (1992) were helpful, we found it necessary to develop special-purpose algorithms to meet our computational needs and to reduce the cost of a single Monte Carlo simulation from nearly a day to less than a minute.

Subsequent to the work of Hoshen and Kopelman (1976), computer scientists have introduced a number of concepts that are central to our algorithm, including the use of data structures, pointers, flags, and linked lists (Cormen et al., 1990). What we required was a methodology that, with every planting event, would allow us to update our knowledge of the clustering present on our array. Following the recommendation of Stauffer and Aharony (1992), we label all sites

consecutively, rather than by a square array, utilizing a single index i to reduce computation time. The overall logic employed is displayed in the accompanying Fig. A1. With the successful planting of a tree, we ask whether the tree is in isolation. If so, we establish a new cluster, check whether it is on the boundary, update relevant information stored in arrays, and return to tree planting. Otherwise, we identify the cluster(s) with which the new tree is associated, and merge the tree into the cluster. If the tree is on a boundary, we update certain array information, and ask whether the tree has another neighbor, i.e. does it bridge two clusters. If so, we merge the relevant clusters, update all array information, including that pertinent to the boundaries. If criticality (i.e. site percolation) has been achieved, we clear away the new cluster, update array and boundary information, and return to tree planting.

In order to perform these tasks most efficiently, we establish data structures through a set of three arrays. The first array α has a one-to-one correspondence with the sites in the lattice. If a site location i is unoccupied, we set $\alpha(i)$ to zero; if it is occupied, then we set $\alpha(i) = j$ where j is the identification number for that cluster. We number clusters consecutively, beginning with the first cluster – consisting of only one tree – that forms. When a cluster is eliminated through a grid-spanning forest-fire or when it merges with a larger cluster, its identification number j is returned to the pool of available cluster identification numbers. The second

array β is associated with the clusters; each of its columns has seven elements and corresponds to a cluster identification number $j = 1, \dots, N \times N$. (The number of columns required is far fewer than this in practice.) The first row of the β array identifies the number of members in each of the clusters; if the cluster with identification number j is not in use, that $\beta(1, j) = 0$. If $\beta(1, j) \neq 0$, then we set $\beta(2, j)$ to the first site in that cluster. Similarly, if $\beta(1, j) \neq 0$, then we set $\beta(3, j)$ to the last site location in that cluster. Finally, the fourth through seventh rows of β correspond to whether a given cluster extends to the “north,” “east,” “west,” or “south” boundary. The relevant β array element is set to 0 if that cluster does not extend to that boundary, and to a positive integer indicating the number of sites on that boundary occupied by that cluster. The third array γ is associated with cluster pointers. There are two ways that a cluster can grow. In the most common, a cluster grows by the addition of a new tree. Hence, when a new cluster is created at a site i , we set $\gamma(i)$ to zero, since there are no other trees associated with that cluster. When a new tree is added to that cluster at a site k , we set $\gamma(i) = k$ and $\gamma(k) = 0$. In other words, the γ array allows us to go from each member of a cluster to the next (in order of their addition) until we have run out of cluster members. If two clusters merge via a bridging tree and, assuming that we have merged the bridging tree with the larger of the two clusters, then we re-assign the γ pointer of the bridging tree with the first site of the smaller cluster. Loosely speaking, this exploits an important idea in Hoshen and Kopelman (1976). This has the immediate advantage of eliminating the need for doing substantial renumbering of indices and pointers. Renumbering is only significant for the α array when clusters merge, and only the smaller cluster is renumbered. The updating of the different rows of β is straightforward and, in the context of the flowchart presented, this algorithm permits 10^7 planting events – including all relevant book-keeping, the source of almost all of the computational cost – on a 128×128 lattice in a matter of minutes.

Appendix B Forecasting evaluation diagram for random events

Consider N events that occur randomly during a time interval NT . In particular, we assume that the probability density distribution at time t for an event $f(t)$ is given by

$$f(t) = \frac{1}{T} \exp\left(-\frac{t}{T}\right) \quad (\text{B1})$$

where T is the mean interval between events. Assume that an event alarm is issued at a time τ after the previous event. We first determine the cumulative distribution function $F(t)$ for the probability that the event will occur *after* the alarm has been issued

$$F(\tau) = \int_{\tau}^{\infty} f(t) dt \quad (\text{B2})$$

Substituting Eq. (B1) into the latter, we have

$$F(\tau) = \frac{1}{T} \int_{\tau}^{\infty} \exp\left(-\frac{t}{T}\right) dt \quad (\text{B3})$$

Making the substitution $t' = t - \tau$, we obtain

$$\begin{aligned} F(\tau) &= \exp\left(-\frac{\tau}{T}\right) \int_0^{\infty} \exp\left(-\frac{t'}{T}\right) d\left(\frac{t'}{T}\right) \\ &= \exp\left(-\frac{\tau}{T}\right) \quad (\text{B4}) \end{aligned}$$

Moreover, the fraction of events that occur prior to the alarm being issued $G(\tau)$ is given by

$$G(\tau) = 1 - F(\tau) = 1 - \exp\left(-\frac{\tau}{T}\right) \quad (\text{B5})$$

This is the fraction of failures to predict.

The mean length of time between an alarm and an event that occurs after an alarm has been issued ($t - \tau$) is given by

$$\langle t - \tau \rangle = \int_{\tau}^{\infty} (t - \tau) f(t) dt \quad .$$

But this is also the fraction of time that alarms are in effect, which we will call A , thus yielding

$$A = \frac{1}{T} \int_{\tau}^{\infty} (t - \tau) \exp\left(-\frac{t}{T}\right) dt \quad , \quad (\text{B6})$$

having substituted Eq. (B1). Again making the substitution $t' = t - \tau$, we have

$$\begin{aligned} A &= \exp\left(-\frac{\tau}{T}\right) \int_0^{\infty} \frac{t'}{T} \exp\left(-\frac{t'}{T}\right) d\left(\frac{t'}{T}\right) \\ &= \exp\left(-\frac{\tau}{T}\right) \quad . \quad (\text{B7}) \end{aligned}$$

Eliminating $\exp(-\tau/T)$ from Eq. (B5) and (B7), we obtain

$$G = 1 - A \quad . \quad (\text{B8})$$

In the forecasting evaluation diagrams, this is the straight line from $G = 1, A = 0$ to $G = 0, A = 1$ shown in Fig. 6.

Acknowledgements. Much of this work was performed when W. I. N. was on sabbatical as the Belkin Visiting Professor in the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science in Rehovot, Israel. This work was completed while both authors visited the Institute for Mathematics and its Applications in Minneapolis, Minnesota. We wish to thank both Institutes for their hospitality. We also wish to thank an anonymous referee and David Bowman for their helpful comments.

References

- Bak, P., Tang, C., and Wiesenfeld, K.: Self-organized criticality, *Phys. Rev. A*, 3, 364–374, 1988.
- Bak, P., Chen, K., and Tang, C.: A forest-fire model and some thoughts on turbulence, *Phys. Let. A*, 14, 297–300, 1992.
- Burridge, R. and Knopoff, L.: Model and theoretical seismicity, *Bull. Seismol. Soc. Amer.*, 57, 341, 1967.

- Carlson, J. M. and Langer, J. S.: Mechanical model of an earthquake fault, *Phys. Rev. A*, 40, 6470–6484, 1989.
- Cormen, T. H., Leiserson, C. El., and Rivest, R. L.: *Introduction to Algorithms*, Cambridge: MIT Press, 1990.
- Drossel, B. and Schwabl, F.: Self-organized critical forest-fire mode, *Phys. Rev. Lett.*, 69, 1629–1632, 1992.
- Gabrielov, A., Newman, W. I., and Turcotte, D. L.: Exactly soluble hierarchical clustering model: Inverse cascades, self-similarity, and scaling, *Phys. Rev. E*, 60, 5293–5300, 1999.
- Guzzetti, F., Malamud, B. D., Turcotte, D. L., and Reichenbach, P.: Power-law correlations of landslide areas in central Italy, *Earth Planet. Sci. Lett.*, 195, 169–183, 2002.
- Hoshen, J. and Kopelman, R.: Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm, *Phys. Rev. B*, 14, 3438–3445, 1976.
- Knopoff, L. and Newman, W. I.: Crack fusion as a model for repetitive seismicity, *Pure Ap. Geophys.*, 121, 495–510, 1983.
- Kossobokov, V. G., Keilis-Borok, V. I., Turcotte, D. L., and Malamud, B. D.: Implications of a statistical physics approach for earthquake hazard assessment and forecasting, *Pure Ap. Geophys.*, 157, 2323–2349, 2000.
- Leath, P. L.: Cluster size and boundary distribution near percolation threshold, *Phys. Rev. B*, 14, 5046–5055, 1976.
- Malamud, B. D., Morein, G., and Turcotte, D. L.: Forest fires: An example of self-organized critical behavior, *Science*, 281, 1840–1842, 1998.
- Malamud, B. D. and Turcotte, D. L.: Self-organized criticality applied to natural hazards, *Natural Hazards*, 20, 93–116, 1999.
- Molchan, G. M.: Structure of optimal strategies in earthquake prediction, *Tectonophys.*, 193, 267–276, 1991.
- Molchan, G. M.: Earthquake prediction as a decision-making problem, *Pure Ap. Geophys.*, 149, 233–247, 1997.
- Newman, W. I. and Knopoff, L.: Crack fusion dynamics: a model for large earthquakes, *Geophys. Res. Lett.*, 9, 735–738, 1982.
- Newman, W. I. and Knopoff, L.: A model for repetitive cycles of large earthquakes, *Geophys. Res. Lett.*, 10, 305–308, 1983.
- Newman, W. I. and Knopoff, L.: Scale invariance in brittle fracture and the dynamics of crack fusion, *Int. J. Fracture*, 43, 19–24, 1990.
- Stauffer, D. and Aharony, A.: *Introduction to Percolation Theory*, 2nd ed., London: Taylor and Francis, 1992.
- Turcotte, D. L.: Self-organized criticality, *Rep. Prog. Phys.*, 62, 1377–1429, 1999.
- Turcotte, D. L., Malamud, B. D., Morein, G., and Newman, W. I.: An inverse-cascade model for self-organized critical behavior, *Physica A*, 268, 629–643, 1999.