INVITED

Invasion Percolation Automata in Sessile Colony Simulation

David Russell and **Rob Day**, Zoology Department, University of Melbourne, Parkville, Vic. 3052.

Abstract. Colonies of sessile organisms such as corals recruit, grow, and compete in patches of space. While a study of sessile bryozoans on a coral reef revealed local rules of recruitment, growth and interaction, the implications of these local rules for a global system of competing species could not be determined. We therefore simulated colonies using percolation automata, represented by hexagonal cells that grow and interact on a flat plane. Percolation automata can be viewed as the opposite of diffusion limited aggregation automata. The recruitment heuristic allows colonies to search for good positions to begin growth. Recruitment, growth and interaction rules differ between species. All interactions occur at edges: colonies grow into empty space, over another colony, or stop at a common boundary. Zooids of a colony cooperate in defence, so that smaller colonies are usually overgrown. Thus size determines the relative 'fitness' of simulated colonies. Colonies of the same species may compete, stop at a common boundary (cooperation), or fuse, to form a colony with a combined level of 'fitness'. The invasion percolation rules led to many emergent properties. At the levels of single colonies and communities of neighbouring colonies, emergent properties mimicked observations of real colonies. At the global level, we have found thus far that cooperation is often counterproductive, rapid recruitment of a species offsets slower growth, and larval searching is effective when recruitment is high. Attention has focussed on non-trivial replicating systems in cellular automata, while the uses of trivial systems of replication have been almost ignored. SESSIM demonstrates that trivial replicating systems for modelling growth processes deserve more investigation. Our invasion percolation automata could model many space limited organisms, such as trees, grasses, moulds, and lichens.

EXTRA KEYWORDS: Competition for space, Emergence, Finite State Machines, Growth Rules, Systolic Arrays.

1. INTRODUCTION

There have been many studies of non-trivial replicating systems in cellular automata (Toffoli and Margolus 1987 p.6), (Poundstone 1987), while the use of trivial systems of replication in automata have been almost ignored. In this paper we use a trivial replicating system to model growth rules in sessile animals or plants. We suggest that trivial replicating systems deserve more investigation as a modelling tool for systems with complex interactions. Even simple systems of replication display emergent properties that appear to be worthy of study.

The type of invasion percolation automata (Chandler *et al.* 1982) we have developed could be used to model interactions between grasses and other spreading plants, moulds, lichens, or marine animals such as corals, bryozoans, and ascidians; or any other organisms that become attached, grow, and engage in essentially two-dimensional competition for space. The simulated organisms show demarcation zones (Medvinsky *et al.* 1993), and take over effects at colony edges. They can be viewed as the opposite of 'diffusion limited aggregation automata' (Toffoli and Margolus 1987 pp.167-168).

1.1 The Real World

SESSIM was designed more specifically to simulate competition between populations of sessile marine organisms, in which species use different strategies to gain space. Our simulation thus differs fundamentally from those of Karlson and Jackson (1981) and Karlson and Buss (1984), who investigated the community consequences of fixed competitive relationships.

Colonies of sessile animals settle and become established (recruitment) on a hard surface such as a rock or a pier pylon, then grow and so limit the space available for the recruitment of other colonies (Day and Osman 1981, Harris 1990). The colonies eventually meet and interact. They may aggressively take over the space of neighbours or, when the neighbouring colony is of the same species, they often cease growing at the mutual border, or fuse (Day 1977, Ryland 1977, Buss 1986).

Recruitment for many sessile marine organisms involves a search for suitable sites where the larva can settle and begin growing, without, for example, the danger of a nearby established colony growing over it (Grosberg 1981). Growth is often a process of roughly circular expansion over the surface (Jackson 1979). Reproduction occurs to form larvae that can settle at new sites, and the number of larvae produced will depend on the size of the colony (Jackson and Wertheimer 1985). These processes all involve zooids (or parts) of sessile colonies cooperating with each other in food gathering, defence, growth and reproduction (Harris 1990), and this led to the use of invasion percolation automata for the simulation.

Bryozoan colonies for example, feed by filtering the water flowing over the colony. Their zooids cooperate to direct the flow of water over the colony, providing each other with food and directing filtered water and excreta towards neighbouring enemy colonies. This would limit the enemy's food gathering potential (Buss 1979). The degree of control of water flow appears to be directly related to the size of a colony (Harris 1990), so that the relative fitness of colonies of marine sessile organisms in interactions with neighbours is often related to colony area, although colonies of some species can grow over larger neighbours (Day 1977, 1985, Buss 1980, Russ 1982). To simulate this process we use a 'fitness engine' based on the number of cells (tiles) in a simulated colony.

1.2 Invasion Percolation Automata

Invasion percolation automata are formed by a space filling growth rule, and grow to occupy neighbouring space by copying themselves from tile to tile in a matrix of tiles, by means of a trivial replicating system (Chandler *et al.* 1982). A trivial replicating system is like a biscuit cutter in that it continually stamps out biscuits, without knowledge of the processes involved in the making of the stamp or the stamping. What is interesting is the biscuits and their interactions.

Our invasion percolation automata continually attempt to place copies of themselves at the edges of already 'owned' tiles, subject to a program that interprets rules of interaction between different colonies. They grow and behave in an efferent way. The colonies compete for space on the plane with colonies of other species, and this process depends on characteristics of the colonies that interact, rather than a set of global rules for the entire matrix (Poundstone 1987, Waldrop 1992 p.17).

In cellular automata, the state of a tile changes because of the states of neighbouring sets of tiles (nucleation), whether the tile is occupied or unoccupied (Toffoli and Margolus 1987, Poundstone 1987). In contrast, changes in states are propagated to neighbouring tiles in our system only by occupied *activated* tiles, using rules specific to that species. This corresponds to the colonisation of space by real organisms such as corals or bryozoans, which have a growing edge that expands over empty space or interacts with neighbours.

In this way we can model competition between populations of colonies belonging to different species, where each species uses a different takeover strategy. The strategy and attributes of each species are kept in a look up table (LUT), which the system interprets to determine which of two neighbouring colonies will occupy an adjacent tile. Thus the outcome of competition between species is based on the LUT of each species.

Because each automaton (tile of a simulated colony) has species specific rules of propagation to its local neighbouring tiles, the interacting strategies of species may cascade and effect a suppression or enhancement of the rules applying to a particular automaton. This results in emergent behaviour on a local basis (a neighbouring set of tiles), community basis (interacting neighbour colonies) and global basis (all colonies on the surface). Thus, unique dynamical structures emerge via the interactions.

2. SYSTEM CONSIDERATIONS

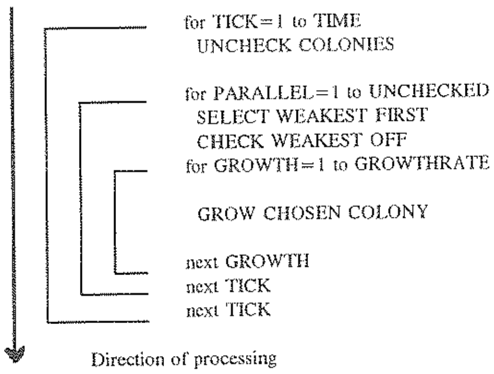
2.1 The Geometry

We have chosen bryozoans, such as *Flustrellida hispida* (Harris 1990, p. 160), as the simplest organisms to model, in terms of the packing of cells (tiles). To simulate the hexagonal tile like growth of the colony we must define a local neighbourhood geometry that fits the growth pattern required (Simon 1992). An hexagonal neighbourhood allows for the tightest packing of tiles, and allows a simulation of approximately circular colonies made up of small zooids. These hexagons act as a systolic network called a beehive (Harel 1992, pp. 274-277), with the colony propagating from tile space to tile space.

The growth of colonies occurs by means of rings of tiles being added to the outside of a colony, a tile width at a time (Penrose 1994, p. 69). To produce different growth rates in different species, [1 to N] rings are added to the growing edge of a colony at each tick of the clock.

2.2 Parallel Processing

Real colonial organisms grow simultaneously and continuously in competition with other colonies. Time is not a set of discrete intervals. In our simulation time moves in discrete 'ticks' of the clock, and this allows us to introduce a form of pseudo - parallelism into the system, to make the model comparatively fair to all colonies in allocating process time in which to grow and compete. Our system is asymmetrically parallel as opposed to symmetrically parallel, as this reduced mutual exclusion problems in programming. The algorithm in Figure 1 displays the main loop of the simulation process, showing how pseudo - parallelism is induced.



```
for TICK=1 to TIME
  UNCHECK COLONIES

  for PARALLEL=1 to UNCHECKED
    SELECT WEAKEST FIRST
    CHECK WEAKEST OFF
    for GROWTH=1 to GROWTHRATE

      GROW CHOSEN COLONY

    next GROWTH
  next TICK
next TICK
```

Figure 1: The algorithm producing asymmetric parallelism. The pseudo code calculates the growth priority of colonies, choosing the weakest to grow first.

The priority rule: WEAKEST GROWS FIRST was used to produce the pseudo - parallelism, so that the growth of all the colonies in the universe is ordered at each tick of the clock. We could have chosen the rule: FITTEST GROWS FIRST, as our system is not symmetric. When colonies are equally fit the program selects one to grow first. This set of rules allows us to step around the mutual exclusion problem that is inherent in parallel methodologies (Harel 1992 pp. 283-294) by ignoring it until one colony is fitter than the other. The situation of two equally fit colonies does not occur often or for long, because differing growth rates and priority to grow affect fitness.

Note that the rule chosen attempts to even out fitness ratings between colonies at each tick of the clock, so that the weakest colonies are given the best opportunity possible to grow and compete: the earlier growth of weak colonies increases their relative fitness in interactions. The growth priority rule acts as an interactive governor between many colonies of differing fitness and directly affects keystone events in colony interactions (which determine their future interactions), and also the timing of such events, as they may be delayed by the governing process. The keystone events tend to occur earlier when the reverse growth rule is used.

3. THE SIMULATION

The simulation consists of three sub processes.
1.Recruitment of colonies onto the surface.
2.The fitness engine, which simulates the interactions between colonies at their edges.
3.Reproduction, which determines the number of larvae produced by each species.

3.1. Recruitment

Colonies arrive on a single tile on a finite surface, and start growing, so that the surface area available for future recruitment is reduced by the area occupied by established colonies. This reduction in available space affects not only the rate at which new recruits become established, but also which species can recruit, as species have different recruitment strategies. Recruitment is governed by a placement strategy algorithm, where each arriving larva will search the surface in ever tightening random walks, until it decides the area chosen is clear of other colonies and future growth can occur. This mimics the larval searching described by Crisp (1974).

The search areas are scaled for each species, so that some species hunt around more, some less, and some do not hunt at all, and simply recruit on any blank surface area. This effects the frequency of recruitment and time of arrival and spacing of colonies, which again directly affects future recruitment. Spacing between colonies and recruits is important to the outcome, as are other characteristics of species, such as recruitment rate and growth rate. This is illustrated in the example below.

Consider a species that recruits rapidly, with small recruitment search areas and a fast growth rate. This leads to 'buffering' between the many, closely spaced sibling colonies as they stop growing at their mutual borders, which reduces the fitness of each colony. This will make it easy for a slow recruiting, slow growing species with a large recruitment search area to take over space from the small sibling colonies, providing one or more colonies recruits early, while sufficient open space is available for some growth. The usual result is that this second species overwhelms all the fast-growing but less fit 'buffered' colonies. This is an example of a keystone process that emerges from simple rules, and is counter-intuitive: rapidly recruiting and fast-growing species were expected to dominate the space available.

3.2. Growth and Fitness

As each activated tile attempts to propagate, the program runs through a decision tree that decides whether a neighbouring tile is occupied by another colony, and if so which species occupies it and what to do about it (Figure 2). All growth effects will be generated by this rule set.

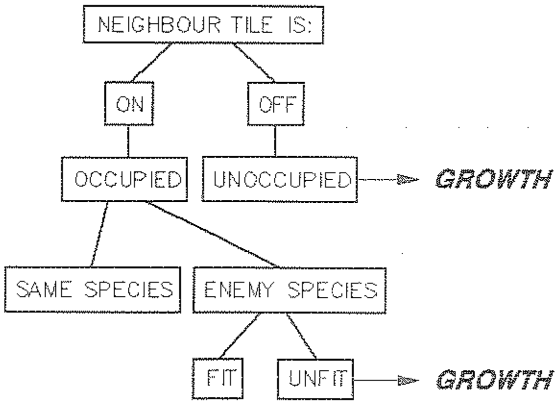


Figure 2: Decision chart for the fitness engine

If the neighbouring tile is occupied by a different species, the outcome is based on a fitness decision process. The fitness of a colony is usually defined by how many tiles a colony has at the last tick of the clock. The rule is: IF FITTER THAN A FOREIGN COLONY THEN INVADE. This drives the system to keystone effects. The result of the rule is that colonies damp each other's growth by invading foreign colonies and buffering the growth of conspecific colonies (Harris 1990).

Although fitness introduces a colony constraint to individual tile replication, it is easier to envision the fitness engine as a finite state automaton, where fitness is a boolean state (STATE = (FIT, UNFIT)). When an activated tile at the edge of a growing colony attempts to propagate to neighbouring sites, the decision tree determines whether propagation is successful, for each tile (vector) surrounding the activated tile in the chosen geometry. Fitness of the growing colony is decided before any growth occurs, to maintain parallelism.

In this way each tile attempts to occupy the tiles around itself and as a result of this local process, each colony will grow into a ring of tiles around itself, except where damped by other colonies. The changed number of tiles in the colony changes its fitness at the next tick of the clock.

3.3 Reproduction

A colony reaches maturity when it is large enough (occupies enough tiles) to start producing larvae. This is a species specific characteristic and is governed by the fitness engine. From this point on it spawns at specified intervals and the numbers of larvae produced depend on the species and size of the colony. As colonies interact a colony may become less fit and start producing less larvae, or even stop reproducing. In this way colonies may have limited reproduction governed by the space it holds over time, and the characteristics of the species. We can measure the success of the strategy of each species in terms of its capacity to reproduce itself, by calculating the total number of larvae produced during a simulation run, and this can be used to set a recruitment rate for the species in another simulation.

4. BEHAVIOUR OF SESSIM

As well as behaviours specified in the design, SESSIM displays many behaviours that were not predicted. In the following descriptions the changes in a two dimensional matrix of hexagons over time are described by one dimensional slices, because as two dimensional colonies expand, at crucial points in time when they meet the action in space is essentially one dimensional.

4.1 Predicted Behaviours

The fitness engine was designed to produce the following behaviours:
(A-->): Unfettered growth, a colony expands into empty space on the plane.

(A-->B): A takes over B when A is fitter because it recruited earlier or grows faster or both. This is called 'invasion'. Invasion involves redefining the edges of a colony, which creates a shortest pathway graph problem (see Sedgewick 1988).

(A1--><--A2): A1 and A2 are similar species and so stop growing at the mutual border. This is called 'buffering'.

((A-->B)...(A<--B)): A invades B initially because it recruited first and is therefore larger when they meet, but B is growing faster and therefore eventually invades A. This is called 'reflux'.

4.2 Unpredicted Behaviours

Unpredicted interactions that emerged from the basic specifications include:
((A1--><--A2)...<--B): Because A1 and A2 buffer each other, B can invade A1 or A2 even if it grows more slowly. Reflux occurs such that A1, A2 or both eventually win, except in very small environments, where B wins.

((A1--><--A2--><--A3--><--A4--><--A5)...<--B): In this interaction many colonies of species A recruit, with only a small search before settling, and grow, buffering each others fitness. This allows a B colony which searches a larger area before settling to invade the A's, even if it grows more slowly, as it is not buffered by siblings and can therefore grow larger (greater fitness). This is a more complex version of the previous interaction where B wins.

'Healing' and 'shadows': When reflux occurs at the corners of expanding hexagonal colonies, the scars of the earlier invasion remain as shadows during further growth. If reflux occurs along a flat edge, the damage is 'healed' because each hexagonal tile along a flat edge has more neighbours of the same colony, so that more activated tiles are available to occupy the tile space. While this is an artefact of the hexagonal shape of simulated colonies, it has its counterpart in a real system, as Palumbi and Jackson (1983) describe slower regeneration after damage of zooids behind the growing edge of bryozoan colonies.

'Calving': a colony may be split into two or more fragment colonies by the actions of other colonies growing together. Suppose there are two colonies A1 and A2 (the same species), growing some distance apart. Then colony B, which grows at a faster rate, arrives between A1 and A2. (A1-->B<--A2) occurs because the A colonies are larger. B may not be able to catch up to the size of A1 or A2 because of invasion by the A colonies, and is split into two colonies when A1 and A2 meet and buffer each other. The result is (A1-->Ba<--A2) and (A1-->Bb<--A2).

'Injection': this is one of the more startling results of running multiple simulations of SESSIM. Imagine colony A1 recruits and starts to grow and later colony B recruits some distance away. Species A grows faster than B. Later another colony of species A (A2) recruits between the A1 and B colonies, close to A1. As A2 buffers the growth of A1, and B is larger (fitter) than A2, B invades A2 until it meets A1, which is larger than B. A1 then injects itself into B, and A2 is calved into two separate colonies (A2a and A2b). Note that once A2 is calved the calved portions are protected from B1 as A1 injects itself into B1 but buffers A2a and A2b.

4.3 Summary

A simple fitness rule set governing local, community and global behaviour leads to interesting and complex outcomes. By specifying species with different rule sets, we can explore the strategies that species may use to compete for space. For example the simulations show that increased growth rate is far more effective in gaining space than increased recruitment. Changing the rule sets at the local and colony level leads to changes in behaviours at all levels. The rule sets capture the characteristics that Silander and Pacala (1990) suggest are useful in modelling competition between plant populations, and should be applicable to most sessile organisms. The parallelism could be made to be a more granular process, and our geometry of neighbourhoods could be changed to build differing neighbourhoods.

As all the changes take place at colony edges, by adding or subtracting tiles, the system only needs to keep track of edge tiles, minimising computation in the tessellation, and the memory and parallelism problems that result from expanding colonies of tiles.

5. PROBLEMS AND LIMITATIONS

Most problems and limitations of the system relate to time and memory availability. As the program attempts to solve an (NP-incomplete) system of interactions, it runs into barriers of processor speed and memory size as colonies expand. So we can only model a subset of the possible communities on a surface. Other examples include the calving and parallelism problems discussed below. These problems limit the time, speed and size of the simulation. The problems that need to be faced highlight the fact that the biscuit cutter attitude does not solve all interactive problems at the grainy level, although it makes programming the growth of automata simpler.

5.1 The Calving Problem

Our initial algorithms considered the two calved parts of a colony A, say Aa and Ab, to be one colony. In order to make the fitness engine treat these calved colonies as separate, we needed an algorithm to carry out calving and assign a fitness to each calf. For example, if colony B were to invade Aa, the combined fitness of the two calves Aa and Ab would distort the (B / Aa) interaction. Furthermore the pseudo-parallelism rule would choose the smaller of two colonies to grow first irrespective of any calving. The case of calves of the same colony re-joining into one colony so that they would not have separate fitness values would then also require a new algorithm.

These problems were solved by building a tail recursion algorithm to calculate any calves connected to an activated tile attempting to replicate, and then the tail recursion was designed out of the algorithm (Sedgewick 1988), so as to make it memory efficient and faster than the recursive algorithm.

The calving algorithm works out the fitness for a calved sub colony, and checks whether the neighbouring colony is a calf or not, but the growth priority no longer depends on the size of the neighbouring colony. The growth priority decision is made at the gross level where both interacting colonies are considered on their whole fitness. So our asymmetrical parallelism will choose the weakest colony to grow first no matter how many calves it has broken into. That is, parallelism is not scaled to all actions of growth.

5.2 Parallelism

The parallelism problem is a very real threat to the model, because in real life situations there is no such thing as parallelism. Colonies arrive, feed, grow, breed, compete and die at their own rates. This is because time flows independently in recruitment rates, fecundity, food availability, metabolism and other variables, even between siblings when viewed at the fine grain level.

Because time is not discrete ticks of a global clock the simulation does not reflect real life but is a version of it, focussed on a few variables at the modellers pleasure. Parallelism is really a point of view that allows the modeller to add order to the system, so as to make sense of it. In this case, we have attempted to introduce a fair allocation of priority between colonies.

5.3 Fitness

Fitness is determined by the number of tiles in a colony, and even in the case of species where the outcome of interactions between colonies does not depend on fitness, the growth priority is still determined by fitness.

5.4 Boundary Effects

Because the simulated surface is a finite size, when a colony grows to the edge it simply stops growing. This can lead to buffering problems, which will ripple through the system to affect colonies away from the edge, via colony interactions. However, boundaries are common in real life systems, such as pier pylons or small rocks colonised by marine sessile animals. In fact natural communities of sessile organisms are almost always patchy (Connell and Keough 1985); organisms colonise and compete in bounded areas. If an unbounded environment is needed, the only way to solve the ripple problem is to recruit only to a small inner segment of the environment and halt the simulation when the first colony reaches the edge. This limits the size of the environment and the time the simulation runs (Simon 1992).

The use of a doughnut environment is only a partial solution. Even though there are then no edges, effects would ripple through the system when the opposite edges of colonies meet, but there would be more time before the simulation must halt to prevent rippling. Note that this is a realistic model for a pier pylon if only two edges are connected, to form a cylinder.

5.5 Spawning

In real life some colonies would only produce larvae from a central area of the colony (Jackson and Wertheimer 1985). In other systems, such as reef corals, colonies may all spawn synchronously, or colonies may allocate all their resources into spawning before they die. These cases are not modelled by SESSIM.

6. ACKNOWLEDGMENTS

We would like to thank John Stern who introduced us, encouraged us, and clarified mathematical issues. RD studied a biological system on which the model is based as part of a PhD study at the University of Sydney, with support from the Australian Museum.

7. REFERENCES

Buss, L.W., Competitive intransitivity in size-frequency distributions of interacting populations, *Proc. Natl. Acad. Sci. USA*, 77(9), 5355-5359, 1980.

Buss, L.W., Competition and community organisation on hard surfaces in the sea, in *Community Ecology*, edited by J. Diamond and T.J. Case, pp. 517-536, Harper and Row, New York, 1986.

Chandler, R., J. Koplick, K. Lerman, and J.F. Willemsen, Capillary Displacement and Percolation in Porous Media, *J. Fluid Mech.*, 119, 249-267, 1982.

Connell, J.H. and M.J. Keough, Disturbance and patch dynamics of subtidal marine animals on hard substrata, in *The Ecology of Natural Disturbance and Patch Dynamics*, edited by S.T.A. Pickett and P.S. White, pp. 125-151, Academic Press, New York, 1985.

Crisp, D.J., Factors influencing the settlement of marine invertebrate larvae, in *Chemoreception in Marine Organisms*, edited by P.T. Grant and A.M. Mackie, pp. 177-265, Academic Press, London, 1974.

Day, R.W. The Ecology of Settling Organisms on the Coral Reef at Heron Island, Queensland, PhD thesis, The University of Sydney, 356 pp., 1977.

Day, R.W., The effects of refuges from predators and competitors on sessile communities on a coral reef. *Proc. Fifth Internat. Coral Reef Congr.*, 4, 41-45, 1985.

Day, R.W., and R.W. Osman, Predation by *Patiria miniata* (Asteroidea) on bryozoans: prey diversity may depend on the mechanism of succession, *Oecologia (Berl.)*, 51, 300-309, 1981

Grosberg, R.K., Competitive ability influences habitat choice in marine invertebrates, *Nature*, 290, 700-702, 1981.

Harel. D., *Algorithmics: the Spirit of Computing*, Addison Wesley, 476 pp., New York, 1992.

Harris. V.A., *Sessile Animals of the Seashore*, Chapman and Hall, 379 pp., London, 1990.

Jackson, J.B.C., Morphological strategies of sessile animals, in *Biology and Systematics of Colonial Organisms*, edited by G. Larwood and B.R. Rosen, pp. 499-555, Academic Press, London, 1979.

Jackson, J.B.C., and S.P. Wertheimer, Patterns of reproduction in five common species of Jamaican reef-associated bryozoans, in *Bryozoa: Ordovician to Recent*, edited by C. Nielsen and G.P. Larwood, pp. 161-168, Olsen & Olsen, Fredensborg, Denmark, 1985.

Karlson, R.H., and L.W. Buss, Competition, disturbance and local diversity patterns of substratum-bound clonal organisms: a simulation, *Ecol. Modelling*, 23, 243-255, 1984.

Karlson, R.H., and J.B.C. Jackson, Competitive networks and community structure: a simulation study, *Ecology*, 62, 670-678, 1981.

Medvinsky, A.B., M.A. Tsyganov, V.Yu. Shakhbazian, I.B. Kresteva, and G.R. Ivanitsky, Formation of Stationary Demarcation Zones Between Population Autowaves Propagating Towards Each Other, *Physica D*, 64, 267-280, 1993.

Palumbi, S.R., and J.B.C. Jackson, Aging in modular organisms: ecology of zooid senescence in *Steginoporella* sp. (Bryozoa; Cheilostomata), *Biol. Bull.*, 164, 267-278, 1983.

Penrose, R., *Shadows of the Mind' a Search for the Missing Science of Consciousness*, Oxford University Press, 457 pp., Oxford, 1994.

Poundstone, W., *The Recursive Universe: Cosmic Complexity and the Limits of Scientific Knowledge*, Oxford University Press, 252 pp., Oxford, 1987.

Russ, G.R., Overgrowth in a marine epifaunal community: competitive hierarchies and competitive networks, *Oecologia (Berl.)*, 53, 12-19, 1982.

Ryland, J.S., Physiology and Ecology of Marine Bryozoans, *Adv. Mar. Biol.*, 14, 285-443, 1977.

Sedgwick, R., *Algorithms*, 2nd ed., Addison Wesley, 657 pp., New York, 1992.

Simon, H.D., *Parallel Computational Fluid Dynamics*, MIT Press, London, 1992.

Silander, J.A., and S.W. Pacala, The application of plant population dynamic models to understanding plant competition, in *Perspectives on Plant Competition*, edited by J.B. Grace and D.R. Tilman, pp. 67-91, Academic Press, San Diego USA, 1990.

Toffoli, T., and N. Margolus, *Cellular Automata Machines: a New Environment for Modelling*, MIT Press, 259 pp., London, 1987.

Waldrop, M.M., *Complexity: the Emerging Science at the Edge of Order and Chaos*, Viking Press, 380 pp., New York, 1992.