Resource Finding and Distribution in an Indefinitely Scalable Machine

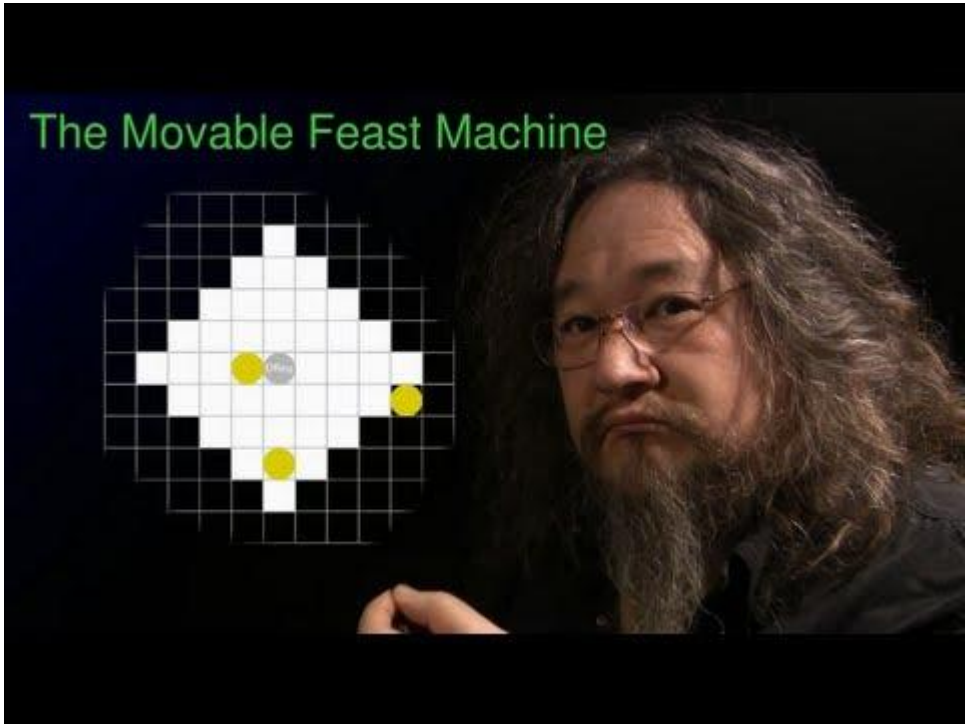
Ezra Stallings
University of New Mexico

An Indefinite Machine (1)

- Suppose we wanted to build an arbitrarily large computer.
- How many bits do we use for addressing?
- What happens when some memory is more light-seconds/minutes/hours away from some processors than other?
- What kind of architecture could solve these and other issues?

An Indefinite Machine (2)

- Movable Feast Machine (MFM) by Dave Ackley, UNM.
- Array of *tiles* plugged together.
- Sites contain *atoms,* instances of *elements.*



An Indefinite Machine (3)

- An *event* occurs when a tile picks a site to update.
- Events occur within an *event window*; a region of radius manhattan distance 4, centered on the updating site.
- *Behavior* function of element corresponding to atom at site is called.

An Indefinite Machine (4)

- MFM is indefinitely *scalable*:
- Each Tile updates asynchronously and randomly.
- Event windows restrict scope of all atoms to a small local memory space.
- Each tile only has to know immediate neighbors. No global addresses! Local memory and processing for all!

Limitations of MFM

- No global address space of any kind.
- Severely limited local scope.
- Unreliable order of operations.
- Unstable machine state.
- Good for enforcing robustness and scalability.
- Bad for writing programs for.

Networks

- Useful for communications, routing, resource sharing, and more.
- Difficult to do in the MFM. Need a way of passing messages over large distances.
- Seems desirable - if possible, does it even help?
- How to build a network to begin with?

Artificial Life

MFM's main strategy for making "useful" computation possible.

Inspiration: Ants & ant colonies.

- No awareness of world at large.
- Local behaviors combine to create order.
- Could we make a large-scale robust system?

Setup

Need to first build a network, then test usefulness. Resource distribution chosen as application of network.

Want to build a network where most (or all) network nodes are outside of each others' event windows.

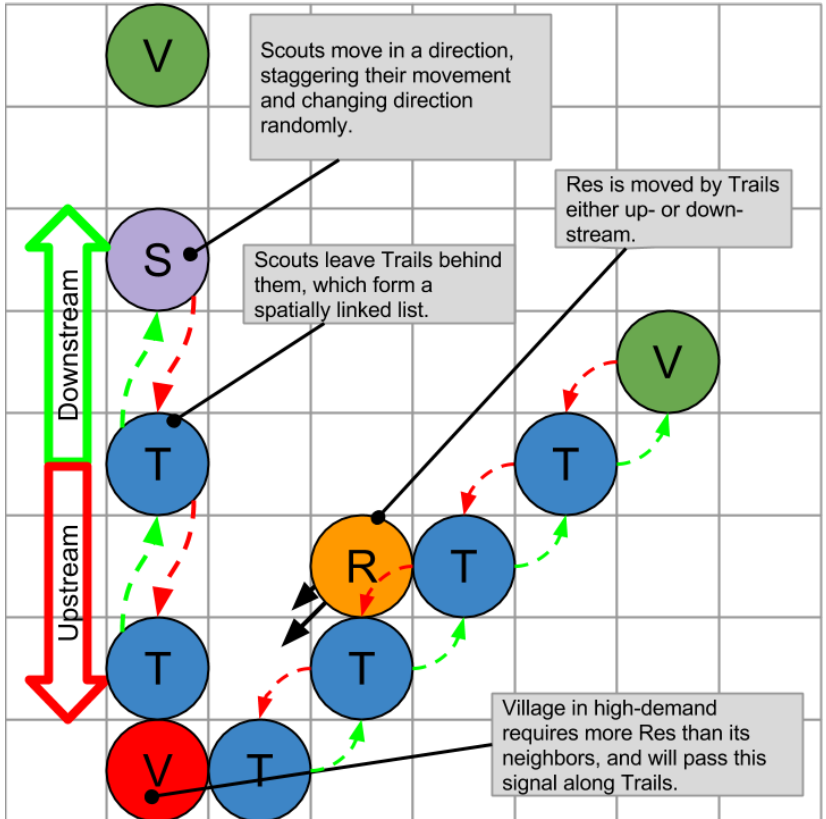
Growing a Network

- Villages make Scouts.
- Scouts leave Trails behind them as they wander. Scouts die when they see another Village.
- Trails are linked together into “paths”, like a spatial doubly linked list.
- Trails self-optimize their position to create straight lines from Village to Village.

Using the Network

- Villages randomly create/destroy Resource atoms based on supply/demand.
- Paths determine most urgent direction for Resource atoms to go based on demand from Villages.
- Villages are able to see demand of other directly connected villages.

Need-based Resource Distribution via Self-Optimizing Spatially Linked Lists.



Village [25 bits]	
ID	10 bits
Demand	4 bits
Supply	4 bits
Local Demand	5 bits
# of Neighbors	2 bits

Legend	
V	Village (Low Demand)
V	Village (High Demand)
T	Trail
R	Resource (Res)
S	Scout
	Downstream Link
	Upstream Link

Trail [61 bits]	
10 bits	Scout ID
10 bits	Index
10 bits	Prev_Index
10 bits	Next_Index
1 bit	Traffic Direction
1 bit	Endpoint
1 bit	Which End
4 bits	Upstream Demand
5 bits	Upstream Local Demand
4 bits	Downstream Demand
5 bits	Downstream Local Demand

Measuring Success

Villages will attempt to consume Res based on demand.

This will result in either a *Success* or *Failure*.

Prosperity Index:

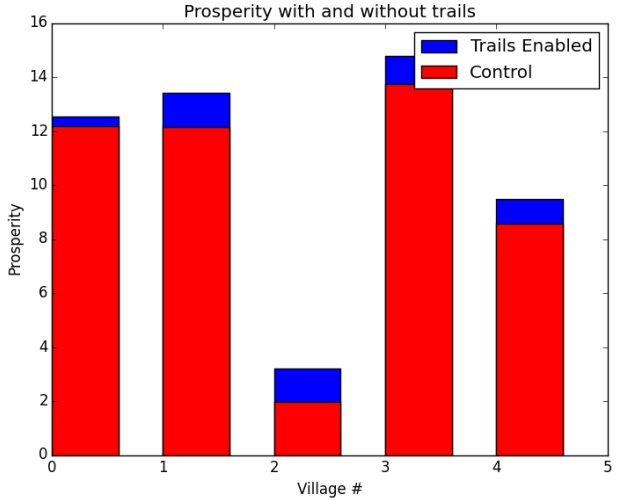
$$Prosp(v) = Demand_v^2 \times \frac{SuccessCount_v}{FailureCount_v}$$

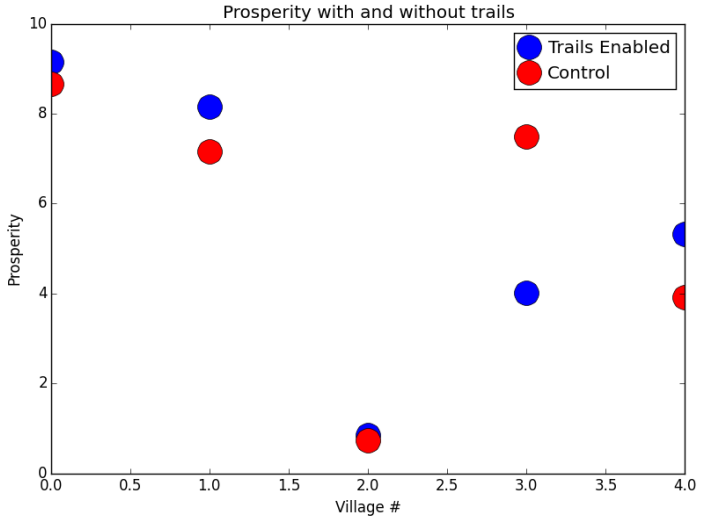
A metric for the success of a village, which greatly favors high-demand villages.

Demo!

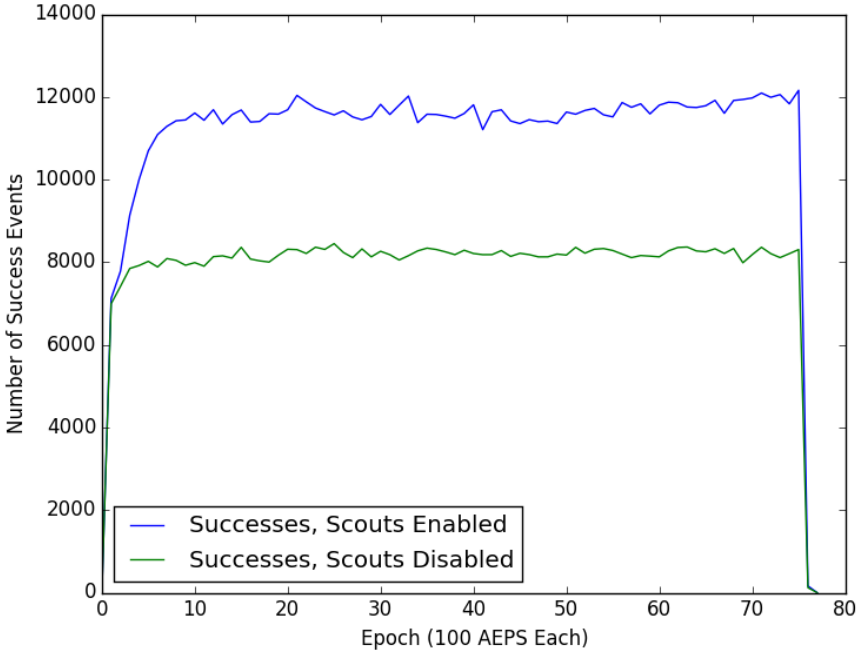
Results

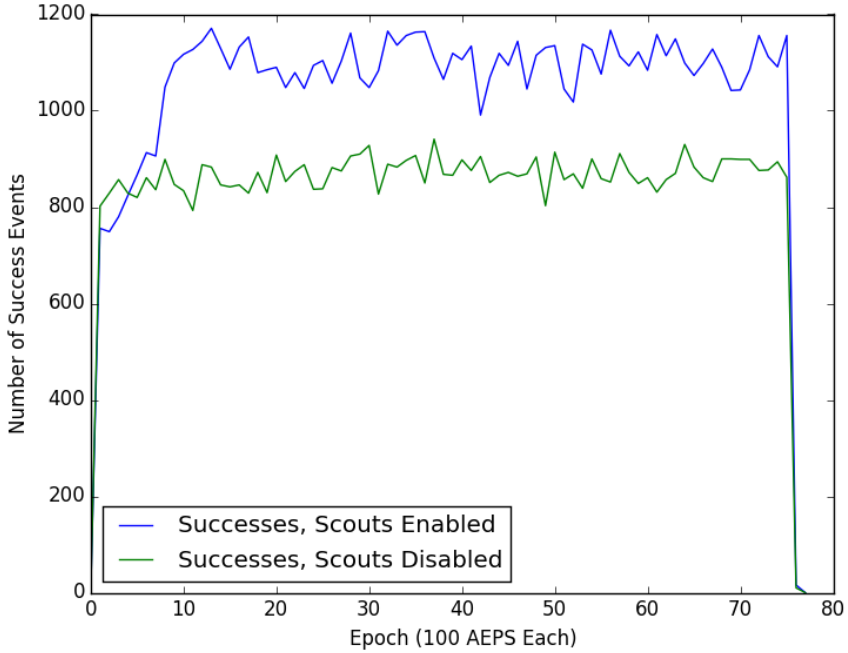
On a local scale: Does not always make things better, especially with a small number of villages.





Results


14000
12000
10000
8000
6000
4000
2000
0
Number of Success Events
0
10
20
30
40
50
60
70
80
Epoch (100 AEPS Each)
Successes, Scouts Enabled
Successes, Scouts Disabled


1200
1000
800
600
400
200
0
Number of Success Events
0
10
20
30
40
50
60
70
80
Epoch (100 AEPS Each)
Successes, Scouts Enabled
Successes, Scouts Disabled

Conclusion

- Works best for moderate numbers of villages.
- Not, however, very robust!
- Is it useful?