Complexity Science and the Art of Trading

By

Dr. Paul Cottrell

April 18, 2016

Overview

This paper will provide more information of complexity science and how it can be used in finance. The following topics are covered in this paper: (a) complexity science, (b) agents, (c) automata, (d) the optimization problem, and (e) Strategy filtering. Within the optimization problem a section on trading strategies and simulation methods will be put forth. In the strategy filtering section a review of agent based models will be discussed to pick a maximizing solution. The goal of this paper is to introduce complexity science and system thinking to financial trading and economic problems.

What is Complexity Science?

Complexity science is the study of complex systems. The difference between the complex and complicated is usually misunderstood. Complexity is when a system has a reflexivity component, whereby feedback loops affect the behavior of the system—leading to unpredictable behavior. Complicated systems are structures that have many inter–links or intra–links, whereby it is hard to discern the actual mechanics of the system. I do not attribute reflexivity to a complicated system and self–organizing behavior is not present in this type of structure. A complex system does have self–organizing behavior. A complicated system can be called a cybernetic system.

There are three main behaviors of complex systems that I would like to explain: (a) agents have simple rules, (b) self–organization, and (c) interactions that can have magnifying effects. Agents can be described with simple rules but exhibit a vast array of behavior.When agents have the ability to interact with other agents and their surrounding environment emergence is exhibited. An example of a simple set of rules for an agent, in a financial paradigm, is the following:

- Agents are traders in a financial market
- Agents have a starting set of income
- Agents can be influenced by surrounding agents trading behavior
- Agents can lose all their income
- Agents have a changing degree of risk taking

With the above agent rules complex market interactions can result that are quite unpredictable. The example is not an exhaustive rule set, just a mere example of the simplicity in agent's rules.

Within the behavior of agents expressed with their respective rule set is the emergence of self–organization. See Figure 1 of a self–organizing structure. Fractal patterns can emerge from complexity, whereby self-similarity is present. See Figure 2 on a fractal pattern of self-similarity. Within fractal designs there is a feedback loop that can magnify the pattern. We can see this magnifying effect in financial markets through herd effects. These herd effects can be negative or positive for asset prices, but eventually the energy in the system fades and a mean reversion usually results in these financial complex systems. Within behavioral finance this reflexive property is the main reason that financial markets are nonlinear and are impossible to fully predict with a high degree of accuracy—unlike Newtonian laws of motion.

Another property of reflexivity is the self fulfilling behavior of financial markets. If other agents move in the same direction then, no matter what the true value of the asset is, the market will move in speculative equilibrium. We will now transition to some more discussions of agents.

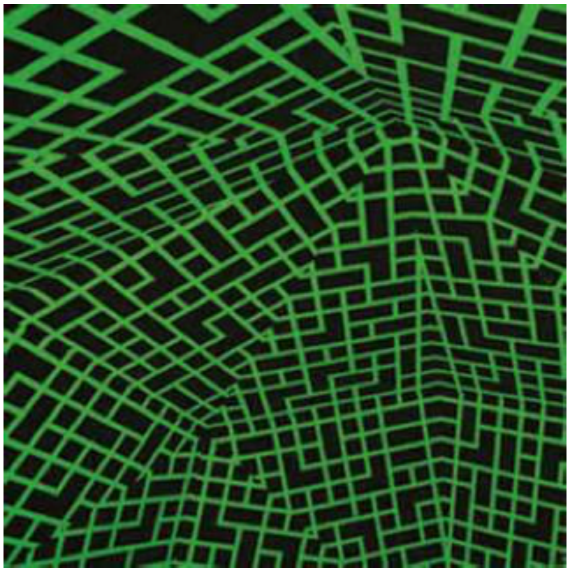


Figure 1. Self–organizing structure.

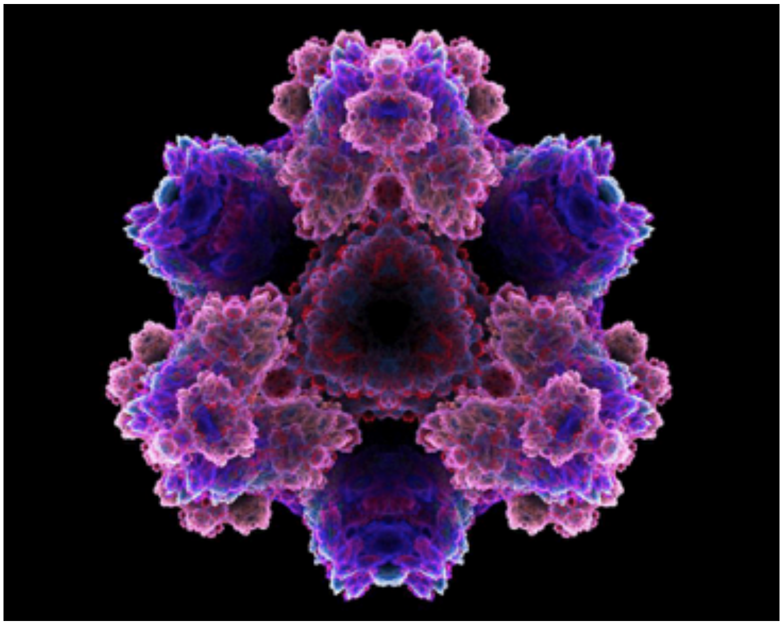


Figure 2. Fractal pattern. Source: www.http://www.funpeak.com/funnypics/fractals-pattern.jpg

Agents

Agents are the atoms of complex systems. These complex systems can be considered adaptive and exhibit emergent behavior. Agent–based modeling is a bottom–up strategy. These agents can be programmed to interact with external and internal environments. For example, researchers are using agent–based modeling to enhance stress testing in the banking sector. Standard orthodoxy for stress testing is to ask the individual banks how their balance would be affected by a ten percent reduction in assets or a thirty percent increase in loan defaults. The problem with the current orthodoxy model approach is that it does not capture a realistic contagion in the financial markets. Individually the banks many seem to pass a prescribed test, but in fact will actually fail if one considers the amplification of financial stressed counterparties. Therefore, modeling in an agent–based manner will allow for counterparty feedback loops and a more realistic simulation.

Complex behavior can emerge from agent–based modeling with just simple interaction rules. In our bank stress example, these agent rules and parameters could be total loans outstanding; payment dates and amounts between counterparties; capital reserve requirements; and short term funding rates. This might not be an exhaustive list of agent rules and parameters, but the main idea can be understood. With these simple rules the banking system will show weaker links than the current stress testing methodology—revealing that stronger banking firms might have much more counterparty risk than their risk models suggest. Through agent–based modeling of the banking sector you will notice how fragile the financial system is to asset depreciation.

Agents should be able to morph their behavior. Allowing for a morphing of behavior will add functionality to the complex adaptive system, producing a more anti–fragile system (Taleb, 2012). When banks are assessing their counterparty risks and determining what reserves are required this is a brittle strategy. They are trying to harden their balance sheet to absorb a negative shock, but this strategy does not allow them to adapt to a changing environment—leading to the needs of a lender-of-last-resort, (i.e., central bank intervention). An adaptive banking system, in theory, would increase their balance sheet reserves in good times and drawdown their reserves in bad times, which reduces the need for central bank intervention.

When agents can morph their behavior, evolutionary pathways are revealed and allows for heterogeneous populations. This morphing of behavior might be in the form of agents assessing their agent neighbor's environmental successes and trying to copy them. Within a large state–space there will be diverse groups optimized to their respective local environments.

Another promising field in agent–based modeling is the use of virtual reality. Agent–base modeling is a simulation, albeit of multiple dimensional–space. It is a natural progress to apply agents' interactions in a virtual world, whereby a researcher can explore the interactions in an immersive way. Game developers have been programming agents in virtual worlds for many years, but usually as first-person-shooting games. To apply this gaming technology to modeling financial markets is not that hard to envision. One just needs to program the agent with certain rules and allow the different agents to interact. The virtual world is just a visualization tool to understand the complexity that emerges.

AUTOMATA

There are two main types of automata: simple and complex. Simple automata are a cybernetic system. This type of automata does not evolve with their environment. Communication is also of limited capacity, which is usually unidirectional. Simple automata are well defined and are considered a bounded system, (i.e., a closed system). Since simple automata are a bounded system, emergent behavior does not exist within single automata, but emergence could be present within a cluster of cybernetic agents. This emergent behavior is quite limited in scope compared to complex automata. See Figure 3 of a simple automata design.

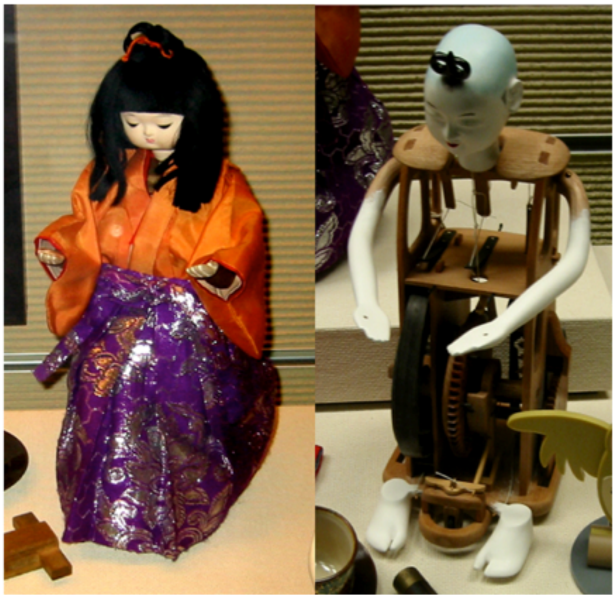


Figure 3. Simple Automata

Complex automata are evolving systems. Communication with internal and external environments is multi–directional for complex automata, due to their feedback loop properties. Complex automata have all the properties of complex adaptive systems: (a) simple rules for agents, (b) ability to communicate with the environment, (c) to have stochastic functions within their decision making, and (d) the ability to learn from doing. Complex automata are unbounded systems—actions are not defined completely to allow for emergent behavior. Since these automata are open systems, they are more adaptive to their environment and possess the ability to machine learn. See Figure 4 for a complex automata design.

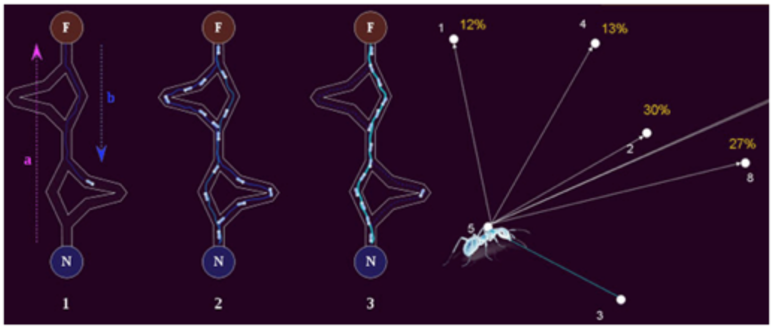


Figure 4. Complex Automat showing an ant algorithm.

In terms of optimization problems, automata could roam around a state–space to determine possible solutions. Many of these financial applications are utilizing complex automata, (e.g., ant algorithms, bee algorithms, and grasshopper algorithms). There are many variant automata algorithms but the main functions are similar—the automata is allowed to wander to find a solution that converges. These wandering complex automata are learning the solution landscape looking for a global optimum. Simple automata could be considered just calculators of certain tasks, but not actually learn from their environment.

The Optimization Problem

How do we optimize trading strategies? To answer this question we need to look into the concept of state–spaces and optimum areas of the state–space. A state–space is a landscape of multiple dimensions. The easiest state–space to envision is a two dimensional landscape, but there can be 2^n dimensions as well. Within this landscape there are possible solutions to a problem, from close approximations to an ideal solution;

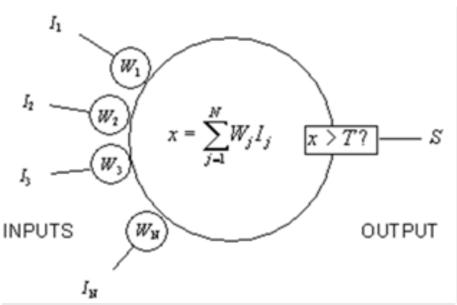
but solutions could also be very poor approximations. As solutions are investigated along this state–space there are local and global optimums. Local optimums are solutions to the problem being investigated with a limited range that does not include the complete state–space. Global optimums are solutions that are the best solution within the entire landscape. When investigating solutions along the landscape, algorithms need to be weighed with two criteria: time to converge to an optimum solution, and accuracy of the solution. Of course the shorter the time to achieve an optimum solution the more probable that solution is within a certain local optimum. But with low dimensional problems it is possible to reach global optimums.

What are some of the current methods to determine an optimized trading strategy? The dependent variable for evaluating trading strategies could be the profit and loss performance, (i.e., a higher profit is a more optimal strategy). For example, a MACD vs. an RSI trading strategy would evaluate the profit and loss performance between these trading technical indicators to determine which indicate should be more utilized. Other possible trading strategies could be a moving average vs. a Fibonacci retracement or the evaluation of different quant models, (e.g., Black Scholes vs. the Levy process). But the common strategy to evaluate the performance of the profit and loss is done by brute force, (i.e., by trying to select all possible scenarios for each of the parameters within each trading strategy). This common method is very time consuming and not all possible trading strategies can be evaluated efficiently. Therefore, the selection set is limited for evaluation in a reasonable timeframe.

Simulation methods are very good to explore the landscape of solutions to a problem compared to brute force. One such simulation method is the use of Ant algorithms. Ant algorithms are program methods where an agent crawls the landscape to find a solution. This agent stores the location of the solution with a pheromone trail. Stronger pheromone scents are considered the most optimized. This simulation is run multiple times to achieve convergence of the solution. The ant algorithm simulation method can produce local optimums, in which an adjustment of the search criteria would most likely achieve better approximations for the solution with higher convergence time —albeit with better convergent times compared to the brute force method for most problems.

Other simulation methods are stochastic simulations, artificial neural networks, and genetic algorithms. Stochastic simulations select randomly the parameter values and add a stochastic process to evaluate the profit and loss of a portfolio. Artificial neural networks are used to determine optimum weights for inputs to produce the best trading signal. The artificial neural networks are trained with historical time series and implemented in a live trading environment to evaluate the profit and loss of a portfolio. A genetic algorithm takes a solution population and ranks them in order of best to worst profit. Then within the genetic algorithm, a combination of the top 10 percent is evaluated to see if better profits are generated. This genetic algorithm is repeated until a solution population runs out or convergence is met.

See Figure 5 on an artificial neural network. Inputs are presented and weights are adjusted to train a neuron. The output of the neuron is to approximate the training set. See Figure 6on a genetic algorithm. It is important to note that artificial neural networks and genetic algorithms can be combined to find possible optimized solutions to problems.



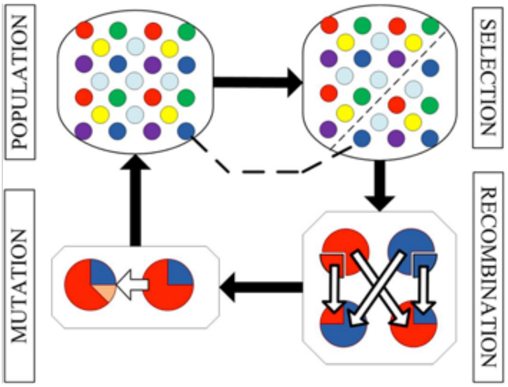


Figure 5. Artificial Neural Network

Figure 6. Genetic Algorithm

Strategy Filtering

How to pick the best strategy? I believe that the best way to answer this question is to utilize a complexity science framework. This can be accomplished by letting agents provide a solution. Agents can be programmed with simple trading rules, which are the following: (a) random selection of risk behavior at the beginning of the simulation, (b) start with equal equity in the trading account, (c) agents select a particular strategy from a defined strategy landscape, and (d) agents evaluate if there trading has created a profit. During the simulation the agents store their strategies and profit performance into a data array with the parameters that the respective agents used.

Through many simulations a global optimum should be reached. These simulations for strategy filtering can also implement artificial neural networks, genetic algorithms, and other agent–based methods for the agent's trading strategy. It is important to note that the simulation's price action can be a stochastic function or based on historic time series, but verification of strategy should be conducted with out-of-sample testing to establish external validity—assuring that stylized facts within the time series did not affect the overall validity of a trading strategy.

Conclusion

Complexity science can help with optimization for trading strategies and the evaluation of parameters within trading models or technical indicators. Brute force can be used for determining best strategy for trading, but this method is not computational efficient when evaluating the full landscape of a multiple dimensioned state–space. Agents can be programmed with certain personalities and can evolve through time. When applying agent–based methods, especially adaptive agents, unexpected knowledge can be obtained about the parameter for certain trading strategies. By using complexity science for solving the trading strategy problem one can see that an agent–based model is a form of machine learning and systems can be built on top of the framework and develop higher performing strategy filtering.

Reference List

Taleb, N. N. (2012). *Antifragile: Things that gain from disorder*. New York, NY: Random House.