

## DISTRIBUTED WARFARE AS A CO-OPTIMIZATION PROCESS

Juliette Zerick  
Department of Mathematics  
Graduate Group in Applied Mathematics  
contact: [jnzerick@math.ucdavis.edu](mailto:jnzerick@math.ucdavis.edu)

### ABSTRACT:

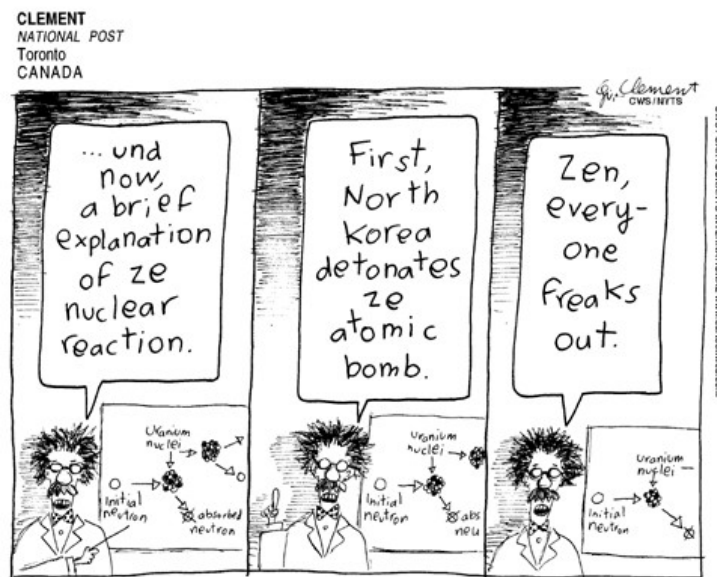
The Armageddon Equations were created to simulate the interactions between a state and an insurgency. Initially, two outcomes, or attractors, were found in the system: either the state quashed the insurgency, or the insurgency triumphed. Using genetic algorithms, a third, stable global attractor was located: the zero attractor. The Armageddon Equations included only a minimum of adaptation, so the lack of coexistence scenarios is unsurprising.

Using coupled immune system algorithms, a new simulation of distributed warfare is proposed. The optimization algorithms used (themselves dynamical systems) already employ competition and adaptation, while the equations used to direct the selection process incorporate the role that instability plays in insurgency. Preliminary results of a general-case implementation of the algorithm are described.

## INTRODUCTION

### *Motivation for the Topic*

Before starting graduate school, I was interested in pursuing research in counterproliferation. Last quarter, I set out to simulate the effectiveness of military and economic sanctions for the purpose of curbing proliferation, and their impact on the stability of a state. First, I found that the simulation of such a system would be *very* difficult. Second, I considered the use of weapons of mass destruction (WMD) highly unlikely, and thus a considerably smaller threat. The real impact of WMD is the emotional reaction they cause, neatly summarized in this editorial cartoon [1] published after North Korea detonated a second nuclear weapon on May 25 [2]:



*Figure 1: This editorial cartoon clearly explains the immediate threat of nuclear weapons testing by North Korea. In the long run, however, the potential for North Korea to become a supplier in the black market arms trade is a very real danger.*

It was my view that the instability caused by armed conflict (sans WMD) was a much greater threat. In the future, I expect armed conflict to be highly nonlinear and of much lower intensity, but no less destructive and disruptive. Large-scale conflict (between major powers) is expensive and messy, avoided at all cost. As for conflict between major and minor powers—which abounds—unless the minor power is suicidal, the minor power will engage in asymmetric tactics [3].

Regardless of the source, such low-intensity conflict is more dangerous now than during the Cold War. Rather than being contained by the two Great Powers, instability now spreads rapidly with unexpected consequences. It is in our interest to keep such conflict to a minimum, and “any methodology that can capture the essential dynamics of insurgency evolution at a

strategic level and can improve policy makers' mental models would appear desirable" [4]. Though, truly, study of the topic is not so much interesting as necessary.

### *A Note on Prior Research*

As mentioned above, I previously worked on simulations of armed conflict. I wound up creating a set of discrete dynamical equations to model the interactions between a state and an insurgency, an asymmetric conflict. A central assumption was that three behaviors were at the heart of such conflict: the state and insurgency *compete* in that the state acts to preserve itself, and the insurgency seeks to replace or disrupt the state; the state attempts to maintain security and stability with police and military forces, while the insurgency wields *instability* and public support as weapons against the state; the state and insurgency both *adapt* to the other (or compete more effectively), but the insurgency is able to do this with greater speed. As early results obtained during the construction of the Equations were rather dismal, I dubbed them the Armageddon Equations, as the name seemed appropriate at the time.

The lengthy time required to construct the Equations hinted at the complexity of the system. I had expected, at least, some sort of cyclic behavior or coexistence outcomes, but found only two behaviors: either the state quashed the insurgency, or the insurgency triumphed.

Considering the large number of parameters and initial conditions, I thought that perhaps I had not sufficiently explored the state space—all 47 dimensions of it. I later employed Goldberg's Simple Genetic Algorithm [5] to efficiently search the state space for possible fixed point solutions. I simplified the model by adding constraints obtained by the use of linear algebra methods; because of the way the Equations were constructed, I was able to restate some variables as linear combinations of others. Unfortunately, I did not manually check the algebra (had I checked, I likely would have found that the linear combinations would lead to contradiction or impossible parameter values). The results suggested that there were dozens of attractors.

I extended the algorithm to search for fixed points without the derived constraints. I found that the only globally stable fixed point was the zero attractor ("armageddon"). This should have been no surprise. The Equations gave the state and insurgency hardcoded behaviors with a minimum of adaptive responses; the opponents were barely sentient and had little incentive to spare the other (I should note that coexistence scenarios do exist in reality when corruption is present [6]). The best scenario the Equations could model would be the decades-long conflict between Sri Lanka and the Liberation Tigers of Tamil Eelam, which very recently ended with the state as the victor. Both sides will likely be indicted for war crimes [7].

Finally, I found that immune system algorithms could effectively serve as a tool for simulation of asymmetric warfare. Two crucial behaviors, competition and adaption, are inherent to the algorithm itself. The skirmishes between the state and insurgency could be encoded using governing equations that drive the direction of the adaption. However, I should note that the simulation, which is based on a clustering algorithm, as of yet can only simulate a small subset of asymmetric warfare: distributed warfare. In a single war, many battles are fought simultaneously; strategies are localized, but the sharing of good methods could potentially drive an emergent dynamic.

## BACKGROUND

### *Evolutionary Algorithms*

I have made use of two such algorithms: genetic algorithms and immune system algorithms. For the former, I used the Simple Genetic Algorithm by Goldberg [5], which I briefly outline below:

1. Spawn an initial population,  $P$ , of  $N$  Individuals
2. Calculate fitness of each Individual in  $P$
3. Repeat until termination condition is met:
4. Randomly pick  $\frac{N}{2}$  pairs of Individuals via “weighted roulette wheel” selection
5. Mate the pairs using crossover and mutation to produce new Individuals
6. Place the newborn Individuals in  $P_{new}$
7. Replace  $P$  with  $P_{new}$

Individuals are associated with their chromosomes, typically binary strings, which represent traits; in this case, traits in my fixed point-finding algorithm were values of parameters and initial conditions. The Individuals compete and the degree of advantage is determined by the Individuals' fitness; the fitness function rewards Individuals with desired traits with higher fitness values. I was interested in stable “traits”—fixed points. Thus I assigned fitness proportional to how many iterations of the Equations were performed using the parameters and initial conditions encoded in the Individual's chromosome. Orbits displaying stable behavior without violating any of the constraints (i.e. values representing the strength of the state and insurgency and such had to be positive) were given the highest fitness value.

The most fit Individuals are more likely to be selected for reproduction; selected Individuals are subjected to crossover, which shares desirable traits between Individuals (explores nearby state space), and mutation (effectively a random search of the space for attractors with small basins of attraction). After successive generations, the proportion of Individuals with the desired traits—those displaying fixed point behavior—increases and eventually dominates the population.

The immune system algorithm I used was Jerne's aiNET model, which was inspired by biological immune systems. Instead of Individuals, there are Antibodies, representatives of a host system, which defend against swarms of Antigens. aiNET is designed to be an efficient clustering algorithm. An outline is given below [8,9]:

1. Spawn an initial population of Antigens  $G$
2. Spawn an initial population of Antibodies  $B$
3. Repeat until the termination condition is met

4. For each Antigen in  $G$ :
5. Calculate the affinity between the Antigen and each Antibody in  $B$ , where  $d(Ab_i, Ag_j)$  is the distance between Antibody  $i$  and the Antigen  $j$ :
 
$$\text{affinity}(Ab_i, Ag_j) = \frac{1}{d(Ab_i, Ag_j)} \quad (1)$$
6. Select the highest-affinity Antibodies (as calculated in (1)) and store them in  $H$
7. Clone the Antibodies in  $H$  proportional to their affinity (the higher the affinity, the more clones the Antibody produces)
8. Assign  $H$  to  $C$
9. Mutate each Antibody in  $C$  in inverse proportion to its affinity, and store the mutated Antibodies in  $C^*$
10. For each Antibody in  $C^*$ , calculate the affinity with the Antigen as in (1)
11. Select from  $C^*$   $\xi\%$  Antibodies with largest affinities (as calculated in the previous step) and store them in  $M$ , the set of memorized clones
12. Eliminate clones from  $M$  whose distances exceed the threshold  $\sigma_d$ , i.e.:
 
$$d(Ab_m, Ab_n) > \sigma_d$$
13. Eliminate clones from  $M$  whose distances do not exceed the threshold  $\sigma_s$  (to avoid redundancy), i.e.:
 
$$d(Ab_m, Ab_n) < \sigma_s$$
14. Concatenate  $B$  with  $M$
15. Randomly generate new Antibodies, and place them in the set  $B_{diversity}$
16. Replace  $B$  with  $B \cup B_{diversity}$

Both Antibodies and Antigens are represented with chromosomes, typically bit strings, meant to store certain traits. Distances between Antibodies and Antigens are problem-specific, similar to the fitness functions of genetic algorithms. I will describe my specific implementation later.

### *Distributed Warfare*

Distributed warfare more generally describes battles with no defined front [10], typical in modern warfare with the change in weaponry and tactics. It is generally found in asymmetric warfare, as it makes little sense for a minor power to directly confront its opponent on the major power's terms [3].

However, this is also known as dissipative warfare, and more commonly refers to “network-centric warfare” (NCW), a dream pursued by the U.S. Military ([11, 12]). With huge leaps in mobile technology and uninterrupted broadband communication (hopefully encrypted), generals dream of turning soldiers into nodes in network, a hive mind with perfect situational awareness that can react at moment's notice. Such a powerful entity could overtake huge swaths of land in a very short period of time [13]. However, such a force is not designed to hold territory. The authors in [13] admit that the NCW model does not accommodate “hearts and minds” strategies. Considering that winning “hearts and minds” is a critical part of (successfully) waging asymmetric warfare [14], much more so than any purely military victory, I disregard the usefulness of NCW. I do, however, anticipate that the paradigm of distributed warfare, at least in the more general sense, will predominate.

## DYNAMICAL SYSTEM

The evolutionary algorithms themselves are dynamical systems. Consider the distribution of Individuals or Antigens as points in a state space. Then successive iterations of the algorithm produce a sequence of distributions that evolve over time. For genetic algorithms, the algorithm tends toward a fixed point—a limiting distribution [15]. If elitism is introduced in the selection process, convergence to the optimal solution is guaranteed (as long as there is also sufficient mutation to search the space) [16].

## METHODS

With a few changes made to aiNET (appropriate to the simulated system), I first implemented a general version of the algorithm: Antibodies attempt to “herd” clusters of Antigens in two-dimensional Euclidean space; distance functions, naturally, are based on the standard (Euclidean) distance metric:

$$\text{for } x, y \in \mathbb{R}^N, d(x, y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (2)$$

Chromosomes of Antibodies and Antigens naturally are points in space; Antigens are represented with a pair of floating-point numbers while Antibodies' chromosomes are bit strings, which are mapped into two-dimensional space.

From the description given in class and in [9], I found that Jerne's algorithm generated too many Antibodies, far more than required for any decent level of “compression” of the data. As a result, in my implementation I chose to keep the number of Antibodies constant, which would mirror the fact that even the largest of war chests are not unlimited. If the weeding-out process of steps 12 and 13 in the algorithm did not pare down the population of Antibodies sufficiently, I selected Antibodies from  $M$  at random and discarded the rest. The most-represented Antibodies naturally would more likely be selected, thus preserving competitive advantage (and adaptation).

In order to promote faster clustering, but also prevent Antibodies from chasing after all the Antigens in the space, for the distance function used in step 10, I summed the distances of the Antibody to *every* Antigen within a fixed radius around the Antibody. Thus Antibodies are aware of their immediate surroundings, but are still assumed to be centrally controlled by the host (the “Selector” in the selection process). This mirrors the reality of distributed warfare, where perfect situational awareness is not a given.

The simulation couples two aiNET systems together—co-evolution as co-optimization. Antibodies are representatives of the state, and Antigens the insurgents (equivalently, representatives of a “shadow state” [17]). Chromosomes of Antibodies and Antigens would represent particular strategies to play against opponents. I should note that in the simulation, one Antibody would represent a small military force facing against a group of insurgents. Also, there are many more Antigens than Antibodies, which can be interpreted in several ways. First, with more Antigens, each representing a possible strategy executed by a group, the insurgency has

more chances to test strategies; thus at each generation, the insurgency benefits from what is effectively a faster *rate* of adaption. Alternatively, the Antibodies may simply be outnumbered, which is a plausible scenario: an insurgency can benefit from the support of the populace. Antibodies, meanwhile, are assumed to be more powerful, and can overtake an entire cluster of Antigens, given the right strategy. An outline of the algorithm is given below:

1. Spawn an initial population of Antigens  $G$
2. Spawn an initial population of Antibodies  $B$
3. Repeat until the termination condition is met
4. For each Antigen in  $G$ :
5. Calculate the affinity between the Antigen and each Antibody  $i$  in  $B$ , where the distance between them is  $d(Ab_i, Ag_j)$  :
 
$$affinity(Ab_i, Ag_j) = \begin{cases} \frac{1}{d(Ab_i, Ag_j)} & \text{if } d(Ab_i, Ag_j) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$
6. Select the highest-affinity Antibodies (as calculated in (3)) and store them in  $H$
7. Clone the Antibodies in  $H$  proportional to their affinity
8. Assign  $H$  to  $C$
9. Mutate each Antibody in  $C$  in inverse proportion to its affinity, and store the mutated Antibodies in  $C^*$

10. For each Antibody in  $C^*$ , calculate the affinity between that Antibody and every Antigen in  $G$ :

$$affinity(Ab_i) = \sum_{j=1}^{|G|} D(Ab_i, Ag_j) \quad (4)$$

where

$$D(Ab_i, Ag_j) = \begin{cases} \frac{1}{d(Ab_i, Ag_j)} & \text{if } d(Ab_i, Ag_j) < \sigma_d \\ 0 & \text{otherwise} \end{cases}$$

11. Replace  $M$  with  $C^*$
12. Eliminate clones from  $M$  whose distances exceed the threshold  $\sigma_d$ , i.e.:
 
$$d(Ab_m, Ab_n) > \sigma_d$$
13. Eliminate clones from  $M$  whose distances do not exceed the threshold  $\sigma_s$  (to avoid redundancy), i.e.:
 
$$d(Ab_m, Ab_n) < \sigma_s$$
14. If  $|M| > |B|$  :
15. Randomly select  $|B|$  Antibodies from  $M$  and store them in  $M^*$
16. Replace  $B$  with  $M^*$
- Else if  $|M| < |B|$  :
17. Randomly generate  $|B| - |M|$  new Antibodies, and place them in  $B_{diversity}$
18. Replace  $B$  with  $B \cup B_{diversity}$

19. For each Antibody in  $B$ :
20. Calculate the affinity between the Antibody and each Antigen in  $G$ , as referenced in (3) (note that distance metrics are symmetric):

$$\text{affinity}(Ab, Ag_j) = d(Ab, Ag_j) \quad (5)$$

21. Select the highest-affinity Antigens (as calculated in (5)) and store them in  $J$
22. Clone the Antigens in  $J$  proportional to their affinity
23. Assign  $J$  to  $F$
24. Mutate each Antigen in  $F$  in inverse proportion to its affinity, and store the mutated Antigens in  $F^*$

25. For each Antigen in  $F^*$ , calculate the affinity between that Antigen and every Antibody in  $B$ :

$$\text{affinity}(Ag_j) = \sum_{i=1}^{|B|} D(Ab_i, Ag_j) \quad (6)$$

where

$$D(Ab_i, Ag_j) = \begin{cases} d(Ab_i, Ag_j) & \text{if } d(Ab_i, Ag_j) < \sigma_d \\ 0 & \text{otherwise} \end{cases}$$

26. Replace  $P$  with  $F^*$
27. Eliminate clones from  $P$  whose distances exceed  $\sigma_d$  from every other Antigen, i.e.:  
 $d(Ag_p, Ag_q) > \sigma_d$
28. Eliminate clones from  $P$  which are not within distance  $\sigma_s$  of at least one other Antigen, i.e.:

$$d(Ag_m, Ag_n) > \sigma_s$$

29. If  $|P| > |G|$  :
30. Randomly select  $|G|$  Antigens from  $P$  and store them in  $P^*$
31. Replace  $G$  with  $P^*$
- Else if  $|P| < |G|$  :
32. Randomly generate  $|G| - |P|$  new Antigens and store them in  $G_{diversity}$
33. Replace  $G$  with  $G \cup G_{diversity}$

Distances between like representatives (Antibodies and Antibodies, Antigens and Antigens) would give the difference in strategies between representatives. Between Antibodies and Antigens, distance would represent the effectiveness of one strategy against another (seen in (3) and (4), and (5) and (6), respectively), the quantity based on the outcome of a few iterations of the governing equations (perfect foresight is *not* given). If the state or insurgency thought a strategy to be lacking, believing that their strength would decrease further, the representative would likely not survive the next generation. Mutation is a change in strategy and would not be haphazard; for the sake of fairness (and realism), shifts in strategy would be limited in magnitude and directed by the Antibodies' and Antigens' respective controllers.

In terms of behavior, the Antibodies still try to “herd” the Antigens, finding a strategy effective against a range of Antigens. The Antigens, however, flee from the Antibodies but attempt to stay in a group, as the insurgency requires some degree of cohesion and a base of

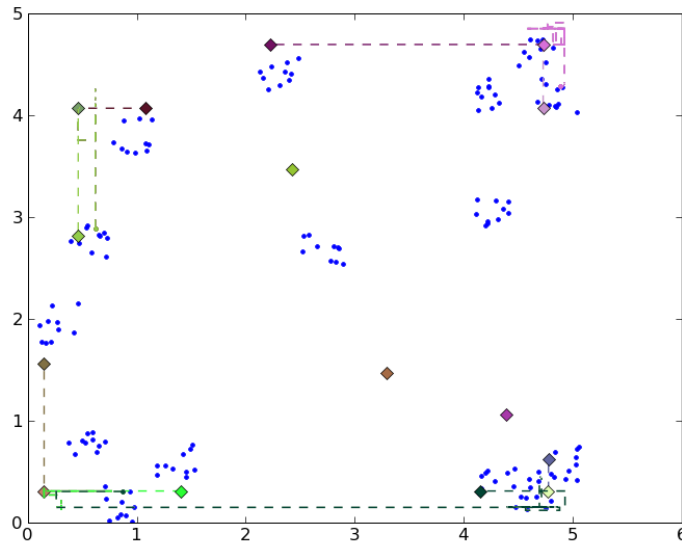


support (whether it is within the insurgency or from the populace).

## RESULTS

In reality, neither a state nor an insurgency would continue the use of a losing strategy—they would adapt as fast and as radically as possible. This was the failing of the Armageddon Equations.

I was unsure of how fast to allow the insurgents to adapt, or how far the Antigens could move. A “fair fight” would have to allow the Antibodies (the state) to keep pace. I tested the algorithm by introducing a small perturbation to each Antigen every generation; though the Antibodies were trailing, they did not become entirely lost, as seen in Figure 2:



*Figure 2: Results of the general-case implementation of the coupled immune system algorithm are shown above. Clusters of Antigens (blue dots) periodically shift. Meanwhile, Antibodies attempt to track the clusters of Antigens. The “movement” of Antibodies via mutation toward the clusters of Antigens are shown as dotted lines, with circles denoting the starting points and diamonds as the end points, or most recent position of the Antibodies.*

I assume that directed (adaptive) movement by Antibodies and Antigens, as intelligent representatives of their respective states, would yield a “fair fight.” The speed with which Antibodies and Antigens shift strategies could be modulated to simulate highly-adaptive or more rigid organizations. To balance out an increase in speed, a decrease in strength or efficiency of attacks would be introduced. An entity with great willingness to adapt presumably would be more decentralized, and thus have a harder time coordinating a response, and thus be less effective. On the other hand, a top-down organization would have a longer response time, and

take significantly longer to adapt, but would muster a much stronger “punch.” However, it has been proven that a bureaucracy, past a certain tipping point, decreases in efficiency [18]. So perhaps a dreaded strike would turn out to be dud that falls into the sea [19].

## CONCLUSIONS AND FUTURE WORK

Like last quarter, construction of algorithms took the majority of the time. Small mistakes are often difficult to track down, but do yield valuable insight into the behavior of the algorithms.

Despite only locating a single, stable attractor in the Armageddon Equations (even with 47 dimensions to work in), I was amazed at the efficiency of the Simple Genetic Algorithm. Even tackling the full problem, the attractor was easily located within 50 generations. This took anywhere from a few minutes to an hour, depending on the precision used; run times would improve if I calculated fitness values with a C module. Though not as much information is yielded as compared to rendering basins of attraction, the speed with which the algorithm can find the existence of fixed points is incredible. I will certainly be using these algorithms in any future research.

As for the simulation, I was only able to implement the more general version of aiNET; as of yet, the code only allows for Antibodies to herd Antigens in Euclidean space, not “strategy space.” Time permitting, I hope to finish the simulation and see how it behaves. I expect the Antibodies and Antigens to chase each other around basins of attraction, but I do not know what limiting behavior might emerge.

## REFERENCES

- [1] Gary Clement. *National Post* (Toronto, Canada). Published through Views of the World (service), available on *Slate*. Accessed June 6, 2009. Available online: <<http://cartoonbox.slate.com/hottopic/?image=11&topicid=109>>.
- [2] Blaine Harden. “North Korean Nuclear Blast Draws Global Condemnation.” *The Washington Post*, May 26, 2009. Available online: <<http://www.washingtonpost.com/wp-dyn/content/article/2009/05/25/AR2009052501672.html>>.
- [3] Andrew Mack. “Why Big Nations Lose Small Wars: The Politics of Asymmetric Conflict.” *World Politics*, Vol. 27, No. 2 (January 1975). pp. 175-200.
- [4] Edward G. Anderson Jr. “A Proof-of-Concept Model for Evaluating Insurgency Management Policies Using the System Dynamics Methodology .” *Strategic Insights*, Vol. 6, Issue 5 (August 2007) . Available online: <<http://www.ccc.nps.navy.mil/si/2007/Aug/andersonAug07.asp>>.
- [5] Rao Vemuri. Class Lecture. Genetic Algorithms (EAD 221, Spring 2009). University of California at Davis, Davis, California. April 1, 2009.

- [6] Elizabeth Dickinson. "Who is blowing up what and why in Nigeria." *ForeignPolicy.com*, May 14, 2009. Available at: <[http://blog.foreignpolicy.com/posts/2009/05/14/who\\_is\\_blowing\\_up\\_what\\_and\\_why\\_in\\_nigeria](http://blog.foreignpolicy.com/posts/2009/05/14/who_is_blowing_up_what_and_why_in_nigeria)>.
- [7] Louis Charbonneau. "U.N. Chief urges Sri Lanka war crimes investigations." *Reuters*, June 5, 2009. Available online: <<http://www.alertnet.org/thenews/newsdesk/N05538468.htm>>.
- [8] Rao Vemuri. Class Lecture. Genetic Algorithms (EAD 221, Spring 2009). University of California at Davis, Davis, California. May 18, 2009.
- [9] Leandro Nunes de Castro and Fernando José Von Zuben. "aiNet: An Artificial Immune Network for Data Analysis." Idea Group Publishing. Draft available online: <<ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/DMHA.pdf>>.
- Published in *Data Mining: A Heuristic Approach*. Ed. Hussein A. Abbass, Ruhul A. Sarker, and Charles S. Newton. Idea Group Publishing, USA. 2001. Preview available online through Google Books: <[http://books.google.com/books?id=vmV2B\\_bzyD8C](http://books.google.com/books?id=vmV2B_bzyD8C)>.
- [10] David A. Fastabend. "That Elusive Operational Concept." *Army Magazine*, June 2001. Available online: <<http://www3.ousa.org/webpub/DeptArmyMagazine.nsf/byid/CCRN-6CCRXXQ>>.
- [11] John Ferris. "Netcentric Warfare, C4ISR and Information Operations." Published in: *Understanding intelligence in the twenty-first century*. Ed. Leonard Victor Scott and Peter Darron Jackson. Routledge, 2004. Available online through Google Books: <[http://books.google.com/books?id=md\\_DQHXY9AC](http://books.google.com/books?id=md_DQHXY9AC)>.
- [12] Thomas P.M. Barnett. "The Seven Deadly Sins of Network-Centric Warfare." *Proceedings of the U.S. Naval Institute*, January 1999, pp. 36-39. Available online: <<http://www.thomaspmbarnett.com/published/7d.htm>>.
- [13] Yedidia Yaari and Haim Assa. "Dynamic Molecules: The Theory of Diffused Warfare." *Pointer: Journal of the Singapore Armed Forces*, Vol. 31, No. 3. Available online: <<http://www.mindef.gov.sg/imindef/publications/pointer/journals/2005/v31n3/features/feature1.html>>.
- [14] John A. Nagl and David H. Petraeus. *The U.S. Army / Marine Corps Counterinsurgency Field Manual*. University of Chicago Press, 2007. pp. 294-295.
- [15] Rao Vemuri. Class Lecture. Genetic Algorithms (EAD 221, Spring 2009). University of California at Davis, Davis, California. April 22, 2009.
- [16] Colin R. Reeves and Jonathan E. Rowe. *Genetic Algorithms—Principles and Perspectives*. Springer, 2003. Available online through Google Books: <<http://books.google.com/books?id=7Gy3yevFd0kC>>.

- [17] David Kilcullen. "Countering Global Insurgency." *Journal of Strategic Studies*, Vol. 28, No. 4 (August 2005), pp. 597–617. Available online: <http://www.smallwars.quantico.usmc.mil/search/Articles/CounteringGlobalInsurgency.pdf>.
- [18] Peter Klimek, Rudolf Hanel, and Stefan Thurner. "Parkinson's Law quantified: three investigations on bureaucratic inefficiency." *Journal of Statistical Mechanics*, March 2009.
- [19] William J. Broad. "North Korean Missile Launch was a Failure, Experts Say." *The New York Times*, April 5, 2009. Available online: <http://www.nytimes.com/2009/04/06/world/asia/06korea.html>.