PROTOCOLOGICAL EXPERIMENTS TOWARD A CRITICAL GAMING
FURTHER NOTES ON DISTRIBUTED COMMUNICATIONS AND WEAPONIZED THEORY

M. Six Silberman

11 December 2009

We do what we must (because we can),
for the good of all of us (except the ones who are dead).
But there's no sense crying over every mistake;
you just keep on trying 'til you run out of cake.
And the Science gets done (and you make a neat gun)
for the people who are still alive.
—J. Coulton, "Still Alive" (credits song to *Portal*)

These ludic explorations in the abstract battlespace of net-work-war are offered in the spirit of Guy Debord, who later in life turned to making strategy games, and Alex Galloway, who made a Java version of Debord's *Kriegspiel*. The games here are slightly more complicated than Baran's simulations (specifically: the space of possible configurations and interactions is larger); they are intended both to give a flavor for the probabilistic machinery underlying Baran's simulations and to move 'beyond' them theoretically. The point is to ask: what can we describe with the network formalism as Baran deploys it? Later, as Galloway and Thacker deploy it in *The Exploit*? Later again, as, say, Lietaer, Ulanowicz, and Goerner deploy it in critical ecological-economic analysis of policy approaches to the contemporary 'economic crisis'?[1] What is foregrounded? What ontological characteristics and dynamics are imputed to the configuration under study through its application? What is hidden? What is lost completely, written out, de-inscribed? What, if anything, can be recovered, and with what kind of practices? What is at stake in the discussion of protocols and counterprotocols?

1 Concept

1.1 Initial conditions

We devise a game with two players. Each player manages a network which consists, at the beginning of the game, of a number of nodes n selected at random in the range $[2,N]$ connected by a number of edges ν selected at random in the range $[N, \frac{N^2-N}{2}]$. Players can allocate their edges in any way they like between their nodes prior to the start of 'play'. Players should alternate placing edges until only one player has edges left to place, at which point they should finish placing their edges. After all edges have been allocated, if any nodes are not connected, they are lost. (More precisely, all components other than the largest are discarded.) N is arbitrary, and should

[1] Lietaer et al., 2009. Options for managing a systemic bank crisis. *Sapiens* No. 2.

be selected according to the players' inclination to manage complexity and scale during gameplay. Each player begins with a random amount of available energy e in the range $[\frac{1}{2}E, E]$, where E is arbitrary (i.e., up to the players). 'Energy' is not universally available or applicable but is located physically at a node. Specifically, at the beginning of the game, $\frac{e}{n}$ is 'stored' at each node and all link capacities are set to $\lfloor \frac{e}{2n} \rfloor$.

1.2 Upkeep

Each node requires a constant maintenance upkeep at each turn. The per-turn node upkeeps are constant within a particular game, and are selected from a normal distribution with mean selected randomly in the range $[0, M]$ and variance selected randomly in the range $[0, V]$, where we require $V < M^2$ and recommend $M < \frac{E}{N}$, but for V and M otherwise arbitrary. At each turn each player receives an energy 'injection' of

$$A = X \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij}$$

where A is referred to as the 'ascendancy'[2] and X is the average mutual constraint,

$$X = k \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{T_{ij}}{\sum_i \sum_j T_{ij}} \log \left(\frac{T_{ij} \sum_i \sum_j T_{ij}}{\sum_i T_{ij} \sum_j T_{ij}} \right)$$

for an arbitrary constant k and T_{ij} the amount of energy transferred from node i to node j in the previous turn, distributed uniformly across the nodes.[3] That is, at each turn, each node receives an energy injection of amount $\frac{A}{n}$. This means that there is a discrepancy between the amount of energy received at each node (which is uniformly distributed across the network) and the amount of upkeep that must be paid (which is normally distributed across the network). This in turn means that energy must be moved across the network. If there is less energy at a node in a given turn than its per-turn upkeep, the node is destroyed. No energy is recovered and all links to it are lost.

1.3 Operations

On each player's turn, after receiving the resource injection and paying upkeep, they can select from any number of operations. Each player can perform each operation once on a given turn, in any order.

Internal operations are performed on one's own network:

- **Add node.** The new node's upkeep is inversely proportional to the initial energy investment expended in construction. This energy is taken from a designated node i. A new edge with some small capacity δ is constructed between node i and the new node.

[2] Ulanowicz et al. 2009, Quantifying sustainability: efficiency, resilience, and the return of information theory. *Ecological Complexity* **6**: 27-36.

[3] ibid.

- **Destroy node.** Energy is recovered inversely proportional to the node's upkeep. If the destroyed node was directly connected to m other nodes and its destruction results in the recovery of an amount of energy ε, an amount of energy $\frac{\varepsilon}{m}$ appears at its former neighbor nodes. All links to the destroyed node are lost.

- **Add edge.** The new edge's capacity is linearly proportional to the construction investment.

- **Destroy edge.** The energy recovered is logarithmically proportional to its capacity. If ε is recovered upon the destruction of an edge linking nodes i and j, $\frac{\varepsilon}{2}$ appears at nodes i and j.

- **Expand edge capacity.** Capacity is expanded logarithmically with energy investment. If capacity is to be expanded on the edge joining nodes i and j, only energy located at nodes i and j can be invested in capacity expansion.

- **Move energy.** Players can make n energy moves in a single turn. An energy move consists in moving some amount of energy between two adjacent (i.e., linked) nodes. The amount of energy moved between two nodes in a single turn cannot exceed the capacity of the edge linking them.

- **Improve probability of attack success.** Probability of attack success can be improved asymptotically (it cannot reach 1) with energy expenditure. Energy invested in raising the probability of attack success at a node is expended at that node.

External operations are directed at the opponent's network:

- **Attack.** Players can attack up to n nodes (the number of nodes in the attacking player's network) in a single turn. Attacks have an origin node in the attacking player's network and a target node in the opponent's network. All attacks in a turn are carried out simultaneously, unless a single node originates multiple attacks, in which case the attacking player specifies the chronological order of targets. Each node is associated with a unique attack cost which is expended at the node at the time of attack, and with a unique probability of attack success. A node can make arbitrarily many attacks in a single turn as long as the required energy is available at the node and the total number of attacks made at all nodes in the turn does not exceed n.

To determine the outcome of an attack, a Bernoulli trial with $p = \hat{p}$, where $\hat{p}$ denotes the probability of attack success associated with the attacking node, is simulated. If the trial is successful, the attack is successful, and the target node in the opponent's network is destroyed. No energy is recovered and all links to it are lost. If the trial is unsuccessful, the energy expended to make the attack is lost, but there is no other result.

If a successful attack fractures the opponent's network into multiple disconnected components, only the component with the most nodes (or, in a variation, with the greatest ascendancy) is retained. All nodes in the disconnected components are lost, and no energy is recovered.

After one player has exhausted their operational choices for the turn (or chooses to pass), their turn ends and the other player receives a resource injection, pays upkeep, and selects from the set of operations.

1.4 Endgame

Players can choose from a number of ending conditions and objectives.

1.4.1 Annihilation

This is the traditional formulation of the game, owed to Baran's canonical 1964 memoranda on distributed communications. The objective is to reduce the number of nodes in the opponent's network by some fraction $f \in (0,1]$, agreed upon by both players at the start of play. The first player whose network is reduced to to $n - fn$ nodes loses.

1.4.2 Network growth in finite time

In this formulation, the total number of turns to be taken by each player is agreed upon at the start of play. After the predetermined number of turns has been taken, each player computes the value of $\frac{n'}{n}$ for their network, where n' denotes the number of nodes in the network at the end of play and n the number of nodes at the start of play. The player whose network is characterized by a greater value of $\frac{n'}{n}$ wins.

1.4.3 Ascendancy

In this formulation, network ascendancy (defined in Sec. 1.2, 'Upkeep') is more important than node count. As in the *annihilation* formulation, players define a fraction $f \in (0,1]$ at the start of play. The first player whose network ascendancy is reduced to $A_0 - fA_0$, for A_0 the ascendancy at the start of play, loses.

An *ascendancy in finite time* formulation modifies the *network growth in finite time* formulation such that the winner is the player whose network is characterized by the greatest ascendancy after the predetermined number of turns have elapsed.

1.5 Variations

1.5.1 Arbitrarily many players

The game can be played with arbitrarily many players. All rules are the same; this means (for example) that each player must select a single opponent (out of many possible) to attack each turn but that they may be attacked multiple times by multiple opponents before they can perform any internal operations.

1.5.2 Imperfect information

In the simplest formulation, the complete network state of each player is known to each other player at all times, and all internal operations are public; there is 'perfect information'. In *imperfect information* variations, constraints are placed on the availability of information about other players' networks. A simple imperfect information variation is as follows:

The number of nodes in all players' networks is public at all times, as is the number of edges and the ascendancy. Other players' network diagrams, however, are hidden at the start of play. As play proceeds, these network diagrams are revealed and updated. In general, a player maintains a representation of each other player's network diagram consisting of a subgraph of the network. At the 'edge' of the subgraph may be nodes linked to other nodes outside the subgraph. In the representation, the number of links is known but their capacity is not.

A new external operation, **probe**, is introduced. A probe has a target player. If the probing player has information about the target player's network (for example, from a previous successful probe), the probing player may select a particular node in the target player's network to probe. If the probing player has no information about the target player's network or declines to target a known node, a node is selected at random. The probing player selects a node in their network from which to deploy the probe. The *power* of the probe is logarithmically proportional to the energy invested in it, which is expended at the the node in the probing player's network from which it is deployed at the time of deployment. Upon deployment, the probe reveals a subtree of the target player's network rooted at the probed node. The depth of the revealed subtree is proportional to the power of the probe and the capacity of the edges connected to the probed node. (This is evaluated in a step-by-step fashion as follows. Suppose the probe is deployed with power ρ and the probed node is linked to two other nodes by edges with capacity c_1 and c_2. The probed node is revealed, if it was not already. The edges at the probed node are reviewed in a random order. Suppose the edge with capacity c_2 is reviewed first, and $\rho > \frac{\kappa}{c_2}$, for κ the probing constant, set at the start of play. Then the node at the other end of the edge with capacity c_2 from the probed node is also revealed, and the probe's power is reduced to $\rho - \frac{\kappa}{c_2}$. The probe is now 'at' the newly revealed node. The process repeats until the probe, now with reduced power ρ', is at some node connected to r edges with capacities $c_i, i = 1, 2, ..., r$ such that $\rho' < \frac{\kappa}{c_i}$ for all i, at which time and place the probe is lost to the probing player. An amount of energy $\mu\rho'$ is recovered by the probed player at the node at which the probe is lost, for $\mu \in [0, 1]$ the probe recovery constant, set at the start of play.) Additionally, the probing node in the probing player's network is revealed to the probed player. A simpler variation involves simply calculating the average capacity of the edges at the probed node and using this to determine the depth of the subtree revealed by the probe; a more complex one allows players to select their own algorithms for reviewing edges and allocating probe power.

A player may deploy n probes per turn.

1.5.3 Situated administrator

In an extension of the imperfect information variations, we can devise a *situated administrator* variation in which the player has a position in their network at all times and maintains a partial representation not only of other players' networks but also of their own. In contrast to the probe, the administrator can 'see' the nodes adjacent to the node at which they are located at any time. In this variation, two new internal operations are introduced:

Move administrator from their current node to any adjacent node. The administrator can move once per turn.

Autoprobe is similar to a normal probe, but has no target node. It is deployed along an edge from a deploying node within the probing player's own network. One autoprobe can be deployed per turn.

In this variation, if the network is broken into multiple components, all components other than the one the administrator is in are lost. (Compare to the standard game, in which all components other than the largest one or the one with the greatest ascendancy are lost.)

1.5.4 Single network

The quandary is this: no one controls networks, but networks are controlled.
—Galloway and Thacker, *The Exploit*, p. 39

In this variation, all play occurs within a single network. At the start of play, players agree on the range $[N_{min}, N_{max}]$ from which the total number of nodes n in the network will be selected at random. A random graph $G(n, p)$ (following the notation of the Erdős-Rényi model) is constructed, with each pair of nodes i and j being connected by an edge with probability $p \in (0, 1]$, agreed upon by the players prior to start of play. After G is generated, all components except the largest are discarded, resulting in a new random graph G' of size n'. G' is partitioned into k subgraphs of size $(1-\eta)\frac{n'}{k}$, where k is the number of players and $\eta \in (0, 1]$ is the fraction of nodes in G' left unassigned to a particular player at the start of play, to be agreed upon by the players prior to the start of play.

In this variation, we introduce a new node attribute: *control*. Each node, in addition to storing some amount of energy, is associated with a control vector $\mathbf{Q} = Q_j, j = 1, 2, \ldots, P$ for P the number of players. Valid configurations of the control vector of a particular node are

$$\begin{cases} \sum_{j=1}^{P} Q_j = 1, \text{ and} \\ Q_j \geq 0 \text{ for all } j \end{cases}$$

or $Q_j = 0$ for all j. A player j can perform internal operations only on nodes for which $Q_j \geq \gamma$, for $\gamma \in (\frac{1}{2}, 1]$ the control threshold, agreed upon prior to the start of play. At the start of play, all nodes in the kth subgraph partitioned in the random graph G' (i.e., the subgraph assigned to player k) are assigned control vectors with

$$\begin{cases} Q_k = 1 \\ Q_j = 0 \text{ for all } j \neq k \end{cases}$$

and all player-unassigned nodes are assigned control vectors with $Q_j = 0$ for all j.

In games with three or more players there are thus three categories of node with respect to control status: *uncontrolled* nodes with $Q_j = 0$ for all j, *controlled* nodes with $\sum_j Q_j = 1$ and some particular k for which $Q_k \geq \gamma$, and *contested* nodes for which $\sum_j Q_j = 1$ but $Q_j < \gamma$ for all j. (Because $\gamma > \frac{1}{2}$ there can be no contested nodes in games with only two players.) At the start of play, all nodes are either uncontrolled or controlled. A node comes under *threat* from a player k when the fraction of its neighbors under player k's control reaches the threat threshold $\beta \in (0, 1]$, agreed upon prior to the start of play. (Note this means an uncontrolled or contested node can be under threat from k players simultaneously if $\beta \leq \frac{1}{k}$.) If an uncontrolled node is under threat from player k, player k can take control of the node on their turn by moving any amount of energy into it. If a *contested* or *controlled* node is under threat from player k, player k can take control of the node on their turn by moving an amount of energy into it *greater* than the amount already present at the node.

Note that ‘energy’ is not player-specific. In this variation, instead of player-specific turns which include energy injection, upkeep, and internal and external operations, each round of play is divided into two phases. First, ascendancies are computed for the subgraphs of the network under each player’s control, and the energy injection is distributed among these nodes (neither the ascendancy calculation nor the energy injection includes contested nodes). Second, each player makes up to n_j energy moves, where n_j is the number of nodes controlled by player j. Third, upkeep is paid on all controlled and contested nodes in the network (no upkeep is paid for uncontrolled nodes) and nodes with insufficient energy present to pay upkeep are lost. Finally, players take turns making internal operations other than energy moves and external operations.

A number of variations on the standard endgame formulations apply to the single network game. A single network annihilation endgame counts n as the number of nodes under a player’s control (i.e., the size of their *control subgraph*) rather than the size of their network. Similarly, instead of a network growth endgame, a *control subgraph growth in finite time* endgame obtains in the single network variation. Finally, *control subgraph ascendancy threshold* and *control subgraph ascendancy growth in finite time* endgames are the single network analogues to the standard ascendancy endgame conditions.

In the single network game, a **local attack** is added to the operation set. It complements the standard attack operation and functions much like the autoprobe operation.

Like the standard game, the single network game can be played with arbitrarily many players; with imperfect information (with the addition of the probe operation); and with situated administrators (with the addition of the move administrator and autoprobe operations).

1.5.5 Programmed strategy

More a tool which changes one’s experience of the rules than a variation in them, the game includes a simple scripting language and programming interface which allows players to script macros and triggered responses to game conditions. If a script attempts an operation that is unexpectedly illegal (e.g., insufficient energy; target node destroyed; etc.) the operation is skipped and the script continues to run. Scripts can include self-termination conditions, and can be manually halted if the player maintains direct control over the subgraph where the script is running (see Sec. 1.5.6, ‘Independently operating components’).

1.5.6 Independently operating components

In this variation, players can attach a script to any subgraph in their network (including the entire network). If the network (or, in the single network variation, control subgraph) is fragmented into multiple components, the largest component (or the component in which the administrator is positioned, in the situated administrator variation) remains subject to internal and external operations directed by the player, while detached components or subgraphs are not destroyed (as in the standard game) but continue to operate under control of the script. For turn-taking purposes the components act as additional players. In multiple-network variations these players continue to play independently until their networks are destroyed; in the single-network variation, the node threat and control rules apply to nodes in their control subgraphs as to any other.

1.5.7 Fixed network

In this variation, the *add node*, *destroy node*, *add edge*, *destroy edge*, and *expand edge capacity* operations are disallowed.

1.5.8 Pathological variations

I am looking at a website—it bores me. I am delighted by my boredom.
—E. Kluitenberg, 'The pleasure of the medium: *jouissance* and the excess of writing'

A number (perhaps an infinity) of possible pathological variations exist. Some are difficult to implement exactly, but can be approximated; others are easy to implement but are likely to afford frustrating play experiences; some may be of theoretical interest from a variety of perspectives. They include *infinite single network*, *no information* (similar to *random strategy*), *random number of independently operating components with random strategies*, *random (mis)information*, and *remote administrator with noisy control channel*.

1.5.9 Topological investigations

In addition to the pathological and computationally impossible *infinite single network* variation, a number of theoretically interesting topological variations exist. The most obvious of these involve replacing the player-allocated edges (in the multiple-network games) or the single Erdős-Rényi random graph (in the single-network variation) with fixed graphs with particular topologies, for example *star* (or 'centralized') graphs, *trees*, *cycles*, *power* graphs, *complete* (or more generally *distributed*) graphs, and *decentralized* graphs with a 'backbone' of 'hubs'.

1.5.10 No attacks

In this variation, the *attack* operation is disallowed. Obviously, the annihilation endgames are nonsensical in this variation.

1.5.11 Real-time play

In this variation, players do not take turns but play continuously. Players receive energy injections and pay upkeep at regular intervals agreed upon by players prior to start of play.

2 Simple example game

This section details the configuration and first round-and-a-half of a simple game. Particular attention is paid to explore the ascendancy construct.

We play a standard annihilation game with two players, who select $N = 10$, $E = 100$, $M = 5$, $V = 4$, and $f = \frac{1}{2}$. To determine n_1, n_2, ν_1, ν_2, e_1, e_2, m, and v, we can run the following commands in MATLAB or Octave:

```
n1 = round(2 + 8 * rand)
n2 = round(2 + 8 * rand)
nu1 = round(n1 + (round(0.5 * (n1 ^ 2 - n1)) - n1) * rand)
nu2 = round(n2 + (round(0.5 * (n2 ^ 2 - n2)) - n2) * rand)
e1 = round(50 + 50 * rand)
e2 = round(50 + 50 * rand)
m = round(5 * rand)
v = round(4 * rand)
```

Of course, each time the machine will return a different result (this is the purpose of the `rand` command), but one result was:

```
n1 = 5
n2 = 10
nu1 = 10
nu2 = 37
e1 = 88
e2 = 86
m = 5
v = 3
```

So at the start of play, player 1 has 5 nodes, 10 edges each with capacity 8, and 88 energy units (or, rounding down, $\lfloor \frac{88}{5} \rfloor = 17$ units at each node), and player 2 has 10 nodes, 37 edges each with capacity 4, and 86 energy units ($\lfloor \frac{86}{10} \rfloor = 8$ at each node). Per-node, per-turn upkeep costs for both players will be drawn from a normal distribution with mean 5 and variance 3. We can draw 15 values (the first 5 for player 1's nodes and the remainder for player 2's) from this distribution with the command

```
round(5 + sqrt(3) * randn(15, 1))
```

which in this case returns

```
4    // node upkeeps for player 1
6
5
7
6
6    // node upkeeps for player 2
1
5
```

```
5
5
4
7
6
6
5
```

Player 1 has $\frac{5^2-5}{2} = 10$ edges, so they have a complete graph and no choice in edge layout (see Fig. 1). Player 2 has 37 edges, or 8 edges short of a complete graph ($\frac{10^2-10}{2} = 45$ and $45-37 = 8$). A plausible layout choice for player 2 is depicted in Fig. 2.

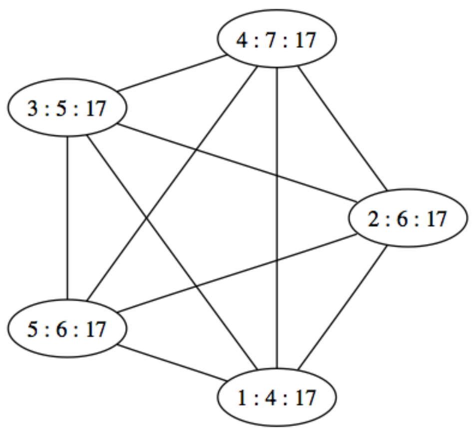

Figure 1: Player 1's network at start of play. Nodes are labeled in the format $i : u_i : e_i$, where i denotes the node index, u_i the per-turn node upkeep, and e_i the amount of energy at the node at the start of play. Link capacities are omitted for clarity; all are equal to 8.

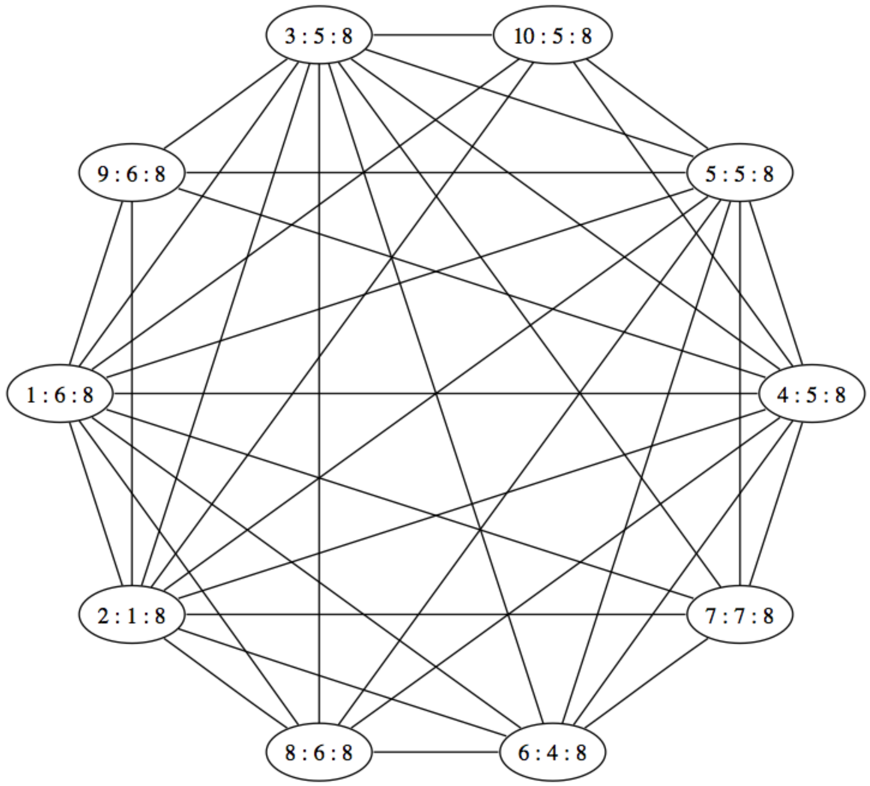

Figure 2: Player 2's network at start of play. All links have capacity 4.

For simplicity, suppose both players have a uniform attack cost of 5 energy units and a uniform attack success probability of 0.1 at the start of play. Suppose the attack success probability for an attack from a given node can be permanently improved to p' by a one-time investment of ξ energy units at the node in question, where $\xi = \lceil \frac{10}{1-p'} \rceil$.

Because no energy has moved in either network, both player's network ascendancies are zero and no energy injection is received on the first turn. Player 1 pays upkeep with the initial energy located at each node, leading to the following energy storage configuration:

Node	Energy
1	13
2	11
3	12
4	10
5	11

In order to receive an energy injection on the next turn, player 1 must move energy within their network to generate ascendancy. Moving one unit of energy from node 1 to nodes 2, 4, and 5 generates the following transition matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which yields an ascendancy of 0, so this ‘balancing’ plan is not great in terms of securing future energy injections. Consider the scheme of moving large amounts of energy to a single node, say node 4:

$$\begin{bmatrix} 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix}$$

This scheme also does not generate any ascendancy. Combining the two, however, does:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix}$$

yields $A = 7$. This is an indication that this pattern of energy movement is more ‘directed’ and ‘efficient’ (i.e., maximizing throughput *and* efficiency simultaneously) than the previous patterns. The pattern

$$\begin{bmatrix} 0 & 0 & 0 & 8 & 0 \\ 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 \\ 0 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{bmatrix}$$

is even more directed, yielding $A = 77$ (see Fig. 3), but this pattern is illegal, exceeding the number of energy moves ($n = 5$) allowed to player 1 in a single turn.

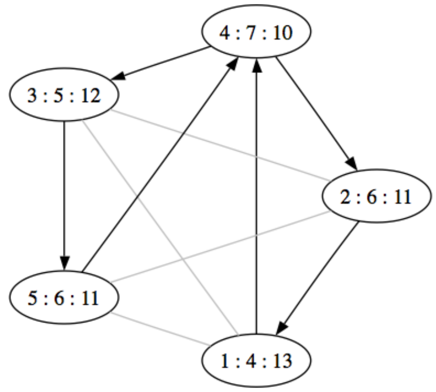

Figure 3: A more directed pattern, $A = 77$. Flows on black links are size 8, in the direction indicated by the arrowhead. There is no flow on gray links. Note this pattern is illegal, as it requires 6 energy moves.

Perhaps the most obvious (legal) strategy for maximizing A for player 1 is to move as much energy as possible around in a circle,

$$\begin{bmatrix} 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 8 \\ 8 & 0 & 0 & 0 & 0 \end{bmatrix}$$

a pattern which yields $A = 93$ (Fig. 4).

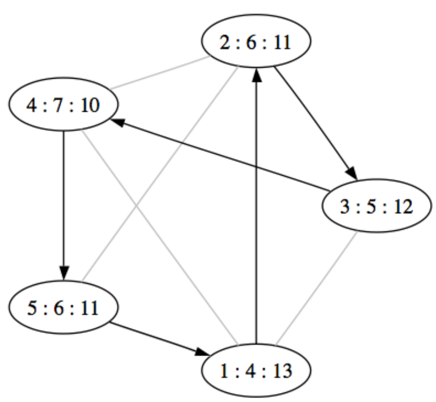

Figure 4: The ascendancy maximizing flow pattern for player 1 on the first turn (given the link capacities at the start of play) is a cycle, with $A = 93$. Flows on black links are size 8, in the direction indicated by the arrowhead. There is no flow on gray links.

After this movement of energy, player 1's network has energy stored in the same configuration as before:

Node	Energy
1	13
2	11
3	12
4	10
5	11

Player 1 has moved energy 5 times, so they cannot move energy any more. They could perform internal operations, or attack player 2, but suppose player 1 ends their turn here. Player 2 then pays upkeep, resulting in the following configuration:

Node	Energy
1	2
2	7
3	3
4	3
5	3
6	4
7	1
8	2
9	2
10	3

Recall that player 2's network has a capacity of 4 on all links at this time. They can move energy in a circle to generate ascendancy, but they have less energy to move at some nodes, and their low link capacity acts as a bottleneck at others:

$$\begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Still, this pattern yields $A = 125$ (see Fig. 5), more than enough energy to pay upkeep for all of player 2's nodes on their next turn.

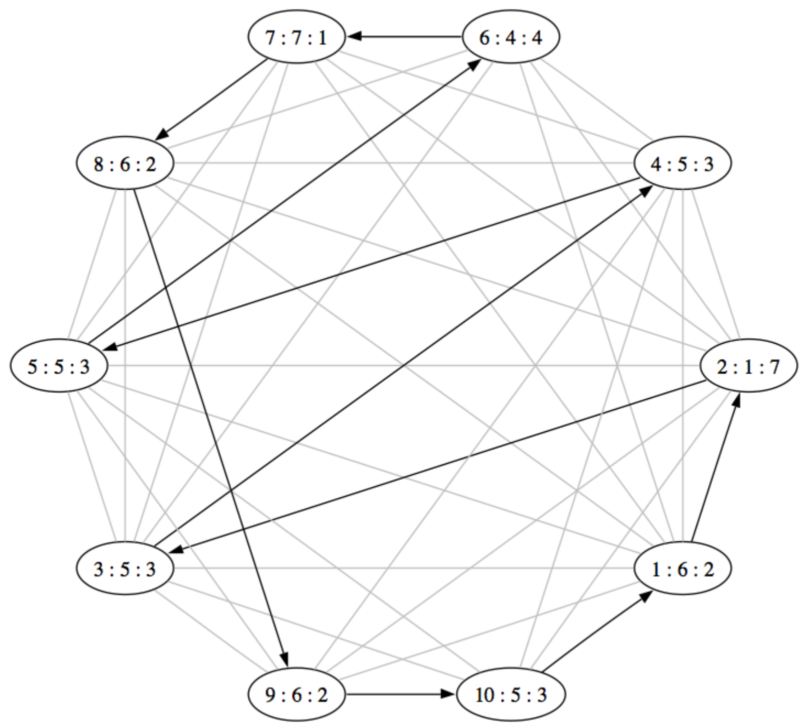

Figure 5: A high-ascendancy flow pattern for player 2 on the first turn.

Suppose player 2, following player 1's lead, refrains from performing internal operations or attacking on this first turn. Now player 1 receives their energy injection, $\lfloor \frac{A}{n} \rfloor = \lfloor \frac{93}{5} \rfloor = 18$ units at each node, and pays upkeep, resulting in the following configuration:

Node	Energy
1	27
2	23
3	25
4	21
5	23

Clearly player 1 can afford to invest in additional edge capacity before moving energy to generate ascendancy for their next turn, but suppose they choose to attack player 2 first. Suppose they attack from node 1, expending 5 energy to target node 6 in player 2's network. All of player 1's nodes have an attack success probability of 0.1, so we can call the MATLAB/Octave command `rand` to simulate a Bernoulli trial; if it returns a value less than 0.1, the trial (and the attack) is successful; otherwise it is not. In this case the call to `rand` returns

```
0.11078
```

so the trial is unsuccessful, and the attack fails. Now player 1 has 22 energy units at node 1, and they decide to improve their attack success probability by directing all of it to this purpose.

$\xi = \lceil \frac{10}{1-p'} \rceil$, and $\xi = 22$, so after this expenditure player 1's node 1 has a new attack success probability of $p' = \frac{22-10}{22} = 0.55$. Player 1 uses 1 energy move to move 8 energy from node 2 to node 1, and attacks player 2's node 6 again. Now the attack is successful if the call to `rand` is less than 0.55. In this case it returns

```
0.21804
```

so the attack is successful, and player 2's node 6 is destroyed. Player 1 uses a second energy move to move 8 more energy units from node 2 to node 1, and attacks player 2's node 10. The call to `rand` returns

```
0.97974
```

so the attack is unsuccessful. 6 energy units remain at player 1's node 1, so they attack player 2's node 10 again. The call to `rand` returns

```
0.14771
```

so the attack is successful, and player 2's node 10 is destroyed. Player 1 has 3 remaining energy moves, 3 remaining attacks, and the following energy configuration:

Node	Energy
1	1
2	9
3	25
4	21
5	23

$f = \frac{1}{2}$, so if player 1 can destroy 3 more of player 2's nodes with their remaining attacks they can win the annihilation endgame outright before the turn ends. If the attacks are unsuccessful, however, player 1 may be in a compromised position energetically. The game proceeds from here.

3 Commentary

As a way of concluding this exercise, I will return to the animating questions at the beginning of the text and sketch a few links between this 'game' (or, if you like, this 'theory') of networks and ongoing discussions and previous investigations from which it draws inspiration. Serious attention has been paid in these conversations to the cultural and political consequences of particular understandings of the network formalism or metaphor (and shortcomings of these), and for good social, cultural, economic, and political reasons. I should reiterate however that I have been motivated in this exercise to explore not what cannot be said or thought in or through particular 'languages' built from particular understandings of networks but to begin to trace what *can* be said, and specifically what can be said *consistently* and *coherently*, in the space delineated by a relatively simple formalism. In the space that follows I offer a few arguments and highlight explicitly some of the conceptual resources that the game has drawn on.

First, this exercise shows the extensibility of the network formalism. It is important to stress that this is a 'demonstration' of the richness of the *formalism*, not just the network *metaphor*, because the formalism and the space of interactions, structures, and processes that it can represent (at least somewhat) consistently and coherently indicates a way of thinking—about structure, agency, process, relation, power, and so on. When and where that way of thinking is 'appropriate' or 'useful' is a separate (but important) question. It is to some extent an empirical question and to some extent a matter for debate and critique. It is a debate that must draw on a historical analysis of the network metaphor and its attendant formalisms—an analysis which I have omitted here.

Second, Galloway and Thacker are not wrong to foreground the analytic difficulties raised by the network formalism's reliance on a strong distinction between nodes and edges (*Exploit*, p. 33). Their complaint about a so-called "diachronic blindness" of the formalism (ibid.), however, reflects not a constraint of the formalism but rather an inadequate understanding of its conceptual richness. Modeling approaches to random processes in graph theory (especially the Erdős-Rényi model, for example) do make it easier to understand processes as unfolding in discrete time than in continuous time. But this is merely an accounting difficulty, not an impossibility.

Third, and perhaps surprisingly, the ecological appropriation of analytical tools from information theory (Ulanowicz et al. 2009, op. cit.) provides a nondualist approach to theorizing much-worried-over dualities like structure/agency, control/freedom, order/disorder, efficiency/diversity, and so on. In the ecological (and perhaps rather Hegelian; locating the boundary of Hegel's long shadow exceeds my theoretical expertise) view, these constitute each other, and there exists a quantitative optimum that describes a balance between them that assures ongoing 'evolution' of the 'system' as a whole. But this 'ecological balance' should not be romanticized: 'sustainability', the name given to this optimum, is not a human value but a statistical property of a system.

Fourth and relatedly, the 'moderate' ecological view of networks and their evolution (ibid.) is an implicit critique of any political theory of networks fashioned as a lens with which to find an 'exploit' or (to use explicitly military terminology) *schwerpunkt*—respectively, a procedural strategy which redirects the energies associated with the network's own 'proper' functioning to disrupt it; or a point at which, owing to the structure of the network, the application of a comparatively small amount of force can collapse the entire edifice. In this view, the latter terms in the dualities above have been romanticized in cultural and political discourse for at least a century. There have been, of course, good reasons for this, but—to put it explicitly—two political implications of the ecological theory of networks are, first, that 'revolution' is not merely undesirable but also meaningless, but second, that opportunities for 'evolving' alternative structures for 'moving energy around' (to put it abstractly) or for fulfilling "real wants and desires"[4] have been underestimated.

Fifth, the consequences of the 'irreducible complexity' of networks have been underestimated in building concrete theories about particular networks, especially networks of political and social importance. The result of this complexity is that 'counterprotocols'—or simply *other* protocols, or more accountable or responsible/"response-able" "social logics"—must be developed over time through rich engagement in *particular* networks: they cannot be designed in the abstract.

[4]Galloway and Thacker, 2004. The limits of networking. *nettime-l*.

Appendices

A Computing ascendancy

In MATLAB and Octave, the ascendancy of a network with n nodes represented by an n-by-n matrix $\mathbf{T} = T_{ij}$, with $i, j = 1, 2, ..., n$, can be computed by the following function:

```
function a = ascendancy(T)

n = size(T, 1);
a = 0;
for i = 1:n
  for j = 1:n
    if (T(i, j) > 0)
      a += T(i, j) * log2(T(i, j) * sum(sum(T)) / (sum(T(i, :)) * sum(T(:, j))))
    end
  end
end
```

This function was used to compute ascendancies for the example game described in Sec. 2. Following Ulanowicz et al. (2009; op. cit.), the logarithm base used is 2.