Endogenous Network Formation: Theory and Application

Allison Oldham Luedtke*

Saint Michael's College

August 10, 2019

Abstract

Economic networks have become a useful modeling tool. However, most of the current literature on economic networks takes the networks themselves as given. This paper presents a model of endogenous network formation which features a new definition of an equilibrium network that improves upon the definition of pairwise-stability. In the model, individual economic agents choose to form relationships with one another, thereby forming the links of a network. I describe an application of this model to the context of firms choosing input suppliers, thus forming a production network, and I analyze the outcome of a single firm losing its equilibrium input supplier. I show that when one of these firms loses its input supplier, aggregate output may actually increase. Simulations of the model indicate that this increase in output is more likely when (1) the firm that loses its supplier has fewer customers prior to losing its supplier and more customers after losing its supplier and (2) the production network as a whole is less interconnected prior to the input removal and more interconnected after the input removal. In the absence of this endogenous network formation, the answers to some economic questions may not only be quantitatively incorrect but qualitatively incorrect.

Keywords: Networks, Production, Network Equilibrium, Aggregate Output

JEL Classifications: C67, D85, E23.

Acknowledgements
This work was supported by the Jefferson Scholars Foundation, Charlottesville, VA [John E. Walker Endowed Fellowship].

*Contact Information: Dr. Luedtke can be reached by email at aluedtke@smcvt.edu or by physical mail at 34 Kinsington St., South Burlington, VT 05403.

1. Introduction

When the US automobile industry was failing, the president of Ford Motor Company supported the bailout of his competitors, General Motors and Fiat Chrysler Automobiles. He did this because if GM and Chrysler failed, their upstream input suppliers would fail, and Ford would no longer have access to those suppliers. Recent economic literature has begun investigating how the interconnectedness of agents determine aggregate outcomes. Acemoglu et al. (2012) explore how the sector level input-output network leads to aggregate fluctuations. di Giovanni, Levchenko, and Mejean (2014) analyze what percentage of aggregate volatility can be attributed to network linkages between firms. Most of the existing literature on economic networks, however, takes the networks themselves as given. When faced with an economic shock, economic agents adapt. They act to mitigate their losses or to improve their outcomes. Their decisions, and thus the links of the economic network, change. As a result, for some economic questions it is necessary to model the formation of the network rather than taking it as given.

I present a model of endogenous network formation wherein a finite set of individual economic agents choose to form relationships with one another and thereby form the links of an equilibrium economic network. This model allows me to ask and answer new questions. The endogeneity of the network formation allows for individual agents to react to economic shocks and for these reactions to determine a new network. I apply this model to the context of individual firms choosing intermediate input suppliers and thereby forming an equilibrium production network. Then, I use the model to find the effect on the production network when an individual firm loses its equilibrium input supplier. I present the model in a general context before explicitly discussing the application to production networks because the model - in particular the new equilibrium definition - is itself a contribution to the literature and is applicable to many applications beyond that of production. The model can be applied to any situation in which the relationships between a finite set of agents determine the payoffs to those agents. For example, it can be used to model the formation of trade networks, online or in-person social networks, or even networks of citations in research literature. I explore this particular application to demonstrate the importance of endogenous network formation.

In the model, each agent chooses whether to form a relationship with a set of available other agents, thereby forming the links of an equilibrium network. I define two network equilibrium definitions: a pairwise-stable equilibrium, introduced in Jackson and Wolinsky (1996), and a new refinement of the pairwise-stable equilibrium, a coordination-proof equilibrium. See Jackson (2010) for an excellent exposition and exploration of pairwise-stability. In the application to production networks, I also discuss the solution to the planner's problem. The equilibrium definition used prominently in the literature, a pairwise-stable equilibrium, is a network in which no possible pair of potentially-related firms would be made better off by deviating to a network in which that relationship is chosen. This is a restrictive definition; it does not allow for the consideration of more than one agent deviating to a different potential relationship at a time. As such, I define a coordination-proof equilibrium as a network such that no *set* of potentially-related pairs can be made better off by a multi-lateral deviation to a different network. That is, considering all possible combinations of agents - of size 1, 2, up to the entire set of agents and each of their alternative relationships - no set of potentially-related pairs would be made better off by switching to the network in which those links are chosen.

In economic contexts, there are often multiple pairwise-stable equilibrium networks in a given problem. Some of these networks are not stable in the following sense: multiple pairs of agents could be made better off by simultaneously (but independently) switching their relationships and thereby deviating to a different network. However, because pairwise-stability only considers one pair deviating at a time, it does not rule out such networks as equilibria. The definition of a coordination-proof network does rule out these networks. While similar in spirit to the concept of the core (see Shapley (1967) and Jackson (2010)), a coordination-proof network is not necessarily a member of the core because this equilibrium definition still considers only bilateral relationships. That is, coalitions consisting of more than two members are not considered. However, unlike pairwise-stability, the definition of a coordination-proof network considers every possible combination of two-member coalitions. The definition of a coordination-proof network combines the realistic and intuitive nature of the core with the computational tractability of pairwise-stability.

This new equilibrium definition differentiates my model from those in Oberfield (2013) and Acemoglu and Azar (2017). Additionally, Oberfield (2013) considers a continuum of firms, while this model features a finite set of agents. This finite set makes the direct computation of pairwise-stable and coordination-proof equilibrium networks possible. Taschereau-Dumouchel (2017) demonstrates a useful approximation technique for identifying equilibrium networks by smoothing the objective function, but acknowledges that the only method that *ensures* the identification of the equilibrium network or networks is the exhaustive approach of enumerating every option and checking each of them. This it the technique used in this paper. In a separate paper, I explore the extreme effect of small errors in network identification.

In this paper, I apply the model to the context of firms choosing intermediate input suppliers to form a production network. That is, the agents are firms and the relationships being chosen are the use of one firm's product by another firm in production. I analyze the effect of a firm losing its equilibrium input supplier on the network as a whole and on the aggregate output produced by the firms that make up the network.

We have evidence that firm choices are driven by the production network in which they reside. (diGiviovani et al. (2014), Baquee (2013)) As in the case of Ford, GM, and Chrysler, the links between a firm and its suppliers, as well as the links between other firms, play a role in the choices that a firm makes. Furthermore, firms may lose input suppliers in many ways. It may be the result of a natural disaster, as in the case of Toyota's Japanese factories when an earthquake in prevented the factories from getting necessary parts for production. (Dawson, 2018) It may be the result of a cyberattack, as in the case of many Ukranian businesses when the Ukranian shipping infrastructure was shut down due to a ransomware attack. (PBS News Hour, 2017) It may even be policy driven. Protectionist trade policies may prevent the use of oil from Saudi Arabia or avocados from Mexico in US production. Health and safety regulations led to the loss of asbestos as a major construction input. Finally, these changes at the firm level affect the macroeconomy. di Giovanni, Levchenko and Mejean (2014) find that a majority of aggregate volatility is driven by changes at the firm level and that this percentage is growing over time.

I show that, contrary to results of existing models, when a firm loses its equilibrium input supplier, aggregate output may actually increase. In the solution to the planner's problem, output will always decrease when a firm loses an input supplier. However, I show that there exist parameters of the model such that when an edge is deleted from a pairwise-stable equilibrium, it is possible for output to be greater in the new pairwise-stable equilibrium. In fact, this result survives in the coordination-proof equilibrium, which is a stronger equilibrium definition.

Output increases after an input is removed when the edge representing the input in the network is the only edge preventing a particularly high-output network from being an equilibrium network. In most cases, when an edge is deleted, the set of new equilibrium networks is a strict subset of the previous set of equilibrium networks and thus output is lower. However, there are situations in which the edge that is deleted was the only potentially-related pair preventing a new network from being an equilibrium. When that edge is deleted, the new network becomes an equilibrium network and if that new network has a higher output than the original equilibrium network then output will increase. Economically, this occurs when one buyer-supplier pair is being made better off at the expense of lower output in the economy as a whole. When this buyer-supplier relationship is no longer possible, the higher-output network is now stable to deviations and therefore an equilibrium.

I simulate the model and the results of this simulation suggest network characteristics which are correlated with output increasing when an input is removed. The level of connectivity in an economic network affects the aggregate outcomes of that network.[1] I measure the connectivity of a given network using the average distance of the shortest paths from each node to every other node. The average shortest path distance is the average distance of the shortest undirected path from each node of the network to every other node. The results of the simulations indicate that the probability that output increases is higher when the production network is less connected - that is, has a longer average shortest path distance - before the input is removed and more connected after the input is removed.

In addition to investigating the role of the connectivity of the network as a whole, I also investigate the role of the centrality of the individual firm that loses its input supplier. The results of the simulation indicate that aggregate output is more likely to increase when the firm that loses its input has more alternative suppliers from which to choose a new input. The increase in output is also more likely when the firm has relatively few customers before the input is removed and relatively many customers after the input is removed. These results taken together indicate that output is more likely to increase after an input is removed when the production economy is less interconnected before the input is removed and more interconnected after the input is removed. The intuition is this: more interconnectedness is a sign of a healthy production economy. When a production network starts out relatively disconnected, it has a lot of room for improvement. If it becomes relatively interconnected after the change in inputs, it is more likely that it realized that improvement potential.

In the next section of the paper, I describe the general network model and use the context of production networks as an illustrative example. In the following sections, I describe the production network formation model in more detail, use the model to investigate the effect of a firm losing access to its input supplier, and simulate the model to identify network and firm characteristics that play a role in increasing aggregate output.

[1] Acemoglu et al. (2015) analyzes the role of connectivity in the fragility of financial networks, for example.

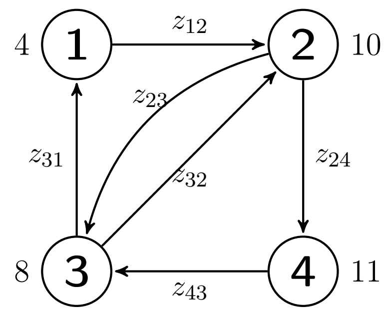
2. Network Model

In this section, I describe a general model of network formation. Throughout, I use the example of a production network to explain aspects of the model in a more specific context. A *network* consists of a set, S, of nodes and a set, E, of directed, weighted, edges between the nodes. Specifically, the elements that compose the set E consist of ordered pairs of nodes, and associated edge weights: $E = \{e = ((s_1, s_2), w_{s_1 s_2})\}$ where $s_1, s_2 \in S$ and $w_{s_1,s_2} \in \mathbb{R}$. If these nodes represent economic agents and the edges between the nodes represent economic relationships between the agents, then the network is an *economic network*. In such an economic context, there are different relationships available to each agent and the utility or payoffs of each agent depends on which of these relationships is in use. That is, the utility of a given economic agent in an economic network depends on which edges exist in the network. Furthermore, this utility depends not only on the edges connected to a given agent, but the edges connected to the other agents in the network as well.

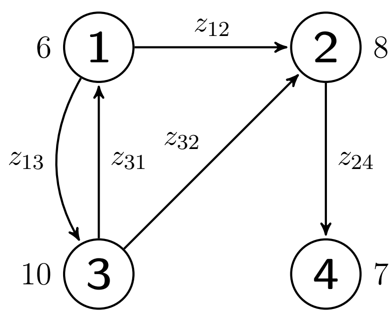
Example: Production Networks

In a production network, the nodes represent firms. These firms choose other firms' products to use as inputs in their own production process. The edges between them represent these input relationships and the weights on these edges represent productivity match values. The higher the match value, the better suited the input is for the production of that good. The profits of each firm depend on their choice of inputs, as well as their inputs' choice of inputs, and so on. In Figure 1 each node represents a firm, and an edge from one firm to another indicates that the latter firm is buying the former firm's input to use in production. Figure 1 compares two different production networks, N_1 and N_2. The numbers to the side of the nodes represent profits. In N_1, Firm 3 is using two inputs - Firm 2's product and Firm 4's product - and earns a profit of \$8. In N_2, Firm 3 switches to using only one input - Firm 1's product - and earns a higher profit of \$10.

Figure 1: Production Networks



(a) N_1



(b) N_2

2.1. Equilibrium Networks

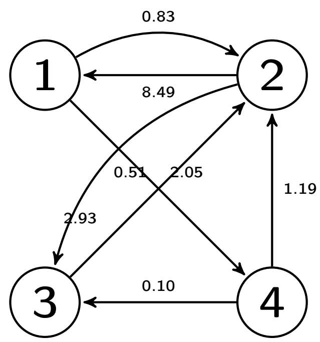
When the edges that exist in a network are the result of optimizing decisions of economic agents, we can define equilibrium networks. I will describe two equilibrium network concepts, one that has been discussed in the literature before and one that I developed as a refinement of the first.

The set of networks which could be equilibrium networks depends on the set of edges, or relationships, available to each economic agent. The *potential network* includes every edge available to each agent. This delimits all of the possible equilibrium networks.

Example: Potential Production Network

Figure 2 shows an example of a potential production network. Firm 2 has three options for inputs: Firm 1's product, Firm 3's product, and Firm 4's product. As a result, there are edges pointing to Firm 2 from Firms 1, 3, and 4 in the potential network. Which of these edges will point to Firm 2 in equilibrium will depend on the choices and payoffs of all of the other firms in the network. Firm 3, on the other hand, has two options: Firm 2's product and Firm 4's product.

Figure 2: Potential Production Network

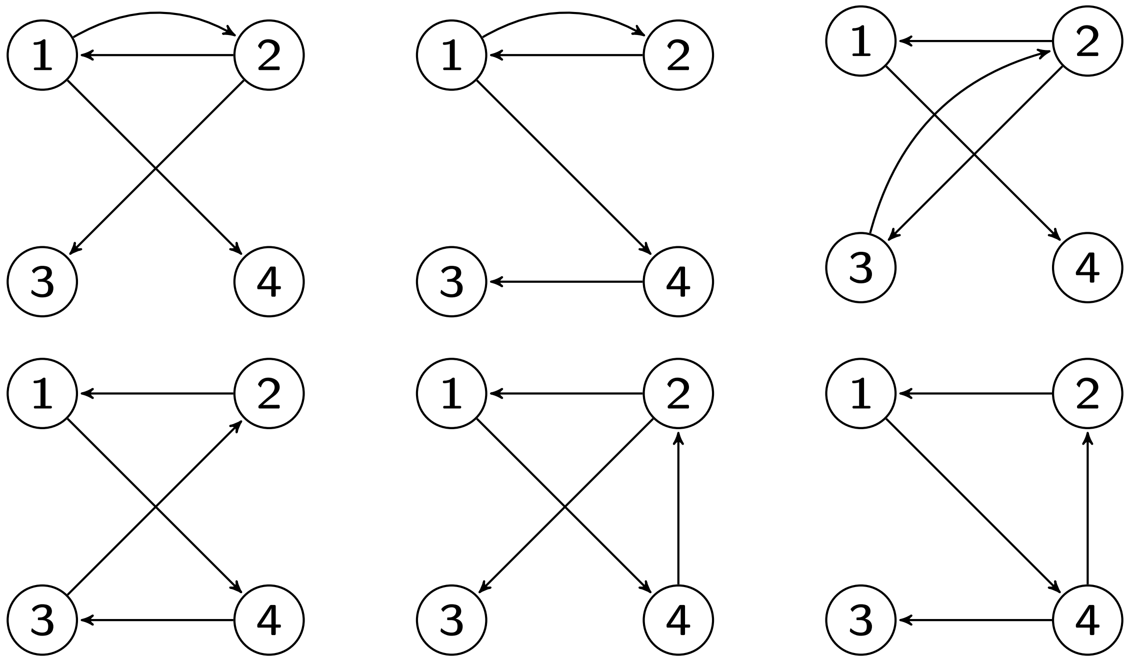


The set of feasible networks, $\mathcal{F}$, is the set of all networks that could be equilibrium networks. These are subnetworks of the potential network. Which subnetworks are feasible networks depends on the economic context and on the utility generating process. For example, using notation similar to that of Oberfield (2013), consider a production function for a firm j defined by the edge, $e = ((i,j), z(e))$, that points to that firm with the form $y_j = \frac{1}{\alpha^\alpha(1-\alpha)^{1-\alpha}} z(e)x(e)^\alpha l_j^{1-\alpha}$. Here the edge weight, $z(e)$, is an edge-specific match value, $x(e)$ is the amount of the input that firm j uses, l_j is the amount of labor that firm j uses, and α is common to all firms. Then, in equilibrium, each firm will use only one input. Thus, the set of feasible networks will consist of subnetworks of the potential network such that every node has exactly one edge pointing to it. This is discussed further in Section 3.

Example: Feasible Production Networks

Suppose the firms in the production network face the production function specified in the previous paragraph. As discussed, they will only choose one input in equilibrium and the set of feasible networks will be all of the subnetworks of the potential network such that every node has one edge pointing to it. Figure 3 enumerates all of the feasible networks for the potential production network in Figure 2 in this case. The edge weights are suppressed for clarity.

Figure 3: Feasible Production Networks with CRS Production Functions



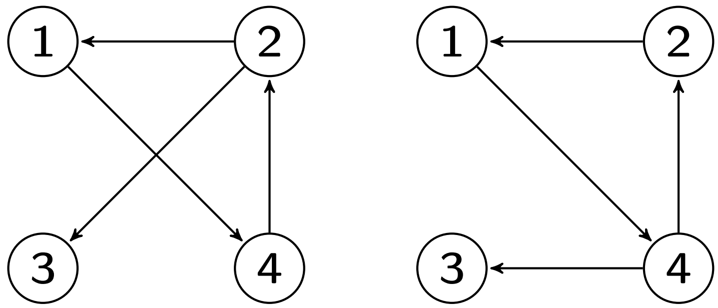
Note that the potential network could be the complete network, that is, the network such that there is an edge from every node to every node. However, by specifying a potential network, this model allows for contexts in which the complete network is not available. For example, in a production network, the complete network would not be a reasonable potential network. We would not expect to see an edge from a firm that produces green beans to a firm that produces truck tires.

To define an equilibrium network requires well defined deviations. If there are no utility-improving deviations from a given network, such a network is an equilibrium network. The deviations which must be ruled out determine the equilibrium network definition. For a given network, $N \in \mathcal{F}$, an i-adjacent network is another network, $\widetilde{N} \in \mathcal{F}$ that differs by exactly i edges. Such a network is a deviation from N. Note that the set of feasible networks is by definition closed under deviations; deviations can be made only to networks in the set of feasible networks.

Example: Adjacent Production Networks

Figure 4 depicts two feasible production networks, N_1 and N_2, that are 1-adjacent to each other. Firm 3 is using Firm 2 as a supplier in network N_1 and switches to using Firm 4 as a supplier in N_2.

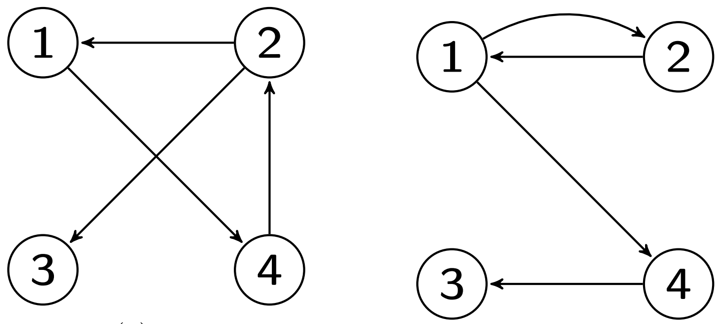
Figure 4: 1-Adjacent Production Networks



(a) N_1 (b) N_2

Figure 5 depicts two feasible production networks, N_1 and N_3, that are 2-adjacent to one another. Both Firms 2 and 3 switch suppliers.

Figure 5: 2-Adjacent Production Networks



(a) N_1 (b) N_3

For a given feasible network, N, let A be a subset of nodes in N. Define the ordered set $S_A = \{S(f) : f \in A\}$ to be a set of alternative nodes to which each node an A could be related to but are not related to in N. That is, S_A defines a specific set of deviations from N for the nodes in A. Let $\widetilde{N}^{A,S_A}$ denote the $|A|$-adjacent network to N associated with the agents in set A switching from the relationships in use in N to the relationships specified by the agents in the ordered set S_A. Using the example in Figure 5, $N_3 = \widetilde{N_1}^{\{2,3\},\{1,4\}}$.

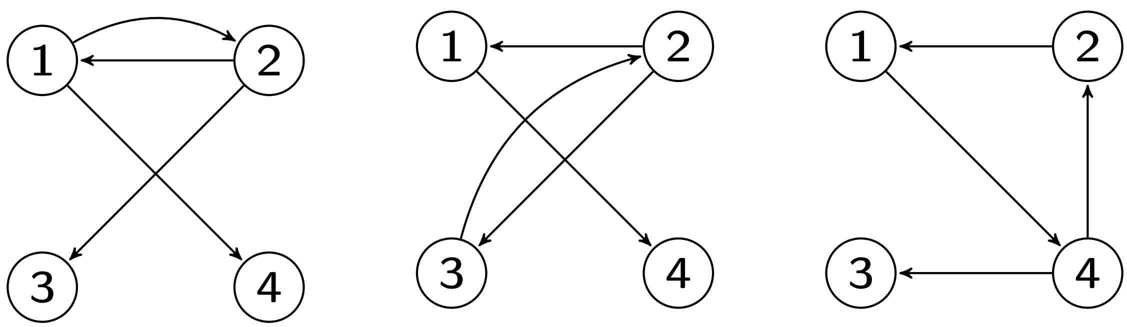
The first definition of network equilibrium that I describe is that of a pairwise-stable network. This is a definition that has been used in the literature previously. See Jackson (2010) and Oberfield (2013), for example. Let $I_j = \{s \in S : \{(s,j), w_{sj}\} \in E\}$ be the set of nodes which have edges pointing to node j in the potential network. That is, I_j is the set of economic agents with which j may form a relationship. Let $\{u_j^N\}_{j \in S}$ be the set of utilities for each agent in S for a given network

N. A pairwise-stable network is a network, $N^* \in \mathcal{F}$, such that no agent j, along with any potential relative $i \in I_j$, would be made better off by moving to the 1-adjacent network defined by j and i. A pairwise-stable network is any network with no utility-improving pairwise deviations. A pairwise deviation from a network N is any network that differs from N by one edge (formed by a pair of agents). To find a pairwise-stable network, all feasible networks that differ by exactly one edge must be ruled out as utility-improving. The set of deviations that must be considered and ruled out is the set of all 1-adjacent networks to N^*. A pairwise-stable network is any network that is stable to deviations by one edge only.

Example: Pairwise Deviations from a Feasible Production Network

Figure 6 depicts all of the pairwise deviations from the feasible production network, N_1, given the potential network depicted in Figure 2.

Figure 6: Pairwise Deviations from Production Network N_1

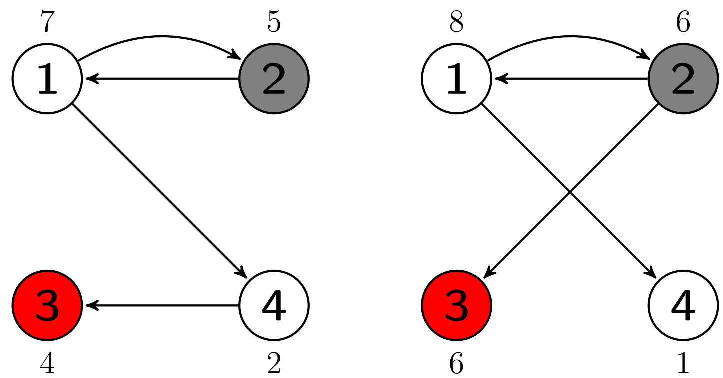


Any network for which there are no profitable pairwise deviations is a pairwise-stable network. For a given potential network, set of feasible networks, and associated utilities, there may be no pairwise-stable networks, a single pairwise-stable network, or multiple pairwise-stable networks.

Example: A Production Network that is Not Pairwise-Stable

Figure 7 shows a network, N_3, that is not pairwise-stable. When Firm 3 switches suppliers - from Firm 4 in N_3 to Firm 2 in N_4 - both Firm 3 and the alternative supplier, Firm 2, see higher profits. Because this profitable deviation exists, N_3 is not pairwise-stable.

Figure 7: N_3 is not Pairwise-Stable



(a) N_3 (b) N_4

Consider a network that is pairwise-stable. After checking every network that differs by exactly one edge, and comparing all of the appropriate utilities, there exist no pair of agents that would be made better off by switching exactly one edge. But suppose that by switching two edges, that is, if two agents change their relationships simultaneously, those two agents and their new relatives saw higher utilities. This is a reasonable deviation to consider in many contexts. However, in checking for a pairwise-stable network, this deviation is not considered. The definition of pairwise stability restricts the deviations that must be ruled out to only those that differ by one edge. It is reasonable to assume that sets of agents would independently choose to simultaneously switch input suppliers. Next I describe a new definition of an equilibrium network that is stable to deviations by all possible numbers of edges (that is, deviations from all existing relationships).

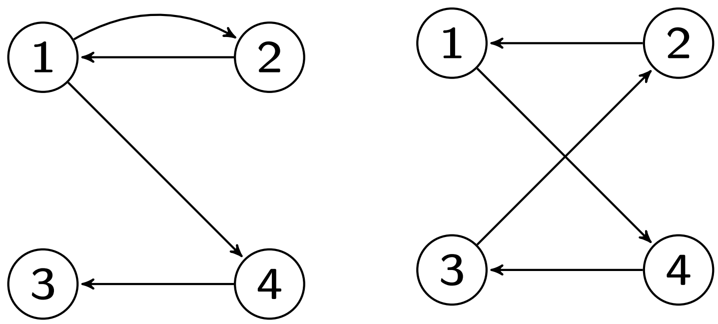
Let C_S^m be the set of all combinations of size m of the nodes in S. E.g., if $S = \{1,2,3,4\}$, then $C_S^2 = \big\{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\big\}$. Let $C_S = \big\{C_S^m\big\}_{m=1}^{|S|}$. For a given feasible network, $F \in \mathcal{F}$, let I_j^F be the set of edges pointing to j in F. A coordination-proof network, N^*, is a network such that no set of any size - from 1 to the number of firms - of firms can be made better off by jointly deviating to another feasible network. Formally, coordination-proof networks are $\{N^* \in \mathcal{F} : \forall C_S^m \in C_S, \forall j \in C_S^m, \forall I \in \mathcal{P}(I_j \setminus I_J^{N^*}), \neg\exists(j,k) k \in I, s.t. u_j^{\widetilde{N}_{jk}} > u_j^N \text{ and } u_k^{\widetilde{N}_{jk}} > u_k^N\}$. That is, it is a feasible network such that for every possible combination of agents and possible related agents, no such combination would see higher utility by deviating to the associated adjacent network. A coordination-proof network is a feasible network that does not have any utility-improving group deviations, where a group can consist of any combination of nodes. To find a coordination-proof network, all feasible networks that differ by any number of edges must be ruled out as utility-improving. This requires checking networks that differ by 1 edge, 2 edges, 3 edges, and so on. The set of deviations that must be considered is the set of all i-adjacent networks for $i = 1, 2, 3, \ldots M$, where M is the maximum number of edges that can be changed.

Because these deviations are independent, a coordination-proof network need not be in the core. These are not coalitions of agents jointly choosing to deviate, it is pairs of agents deviating simultaneously. A coordination-proof network is not necessarily coalition-proof.

Example: Group Deviations from a Feasible Production Network

Figure 8 depicts all of the group deviations for the set of firms $\{2, 3\}$ for production network N_1.

Figure 8: Group Deviations of $\{2, 3\}$ from N_1

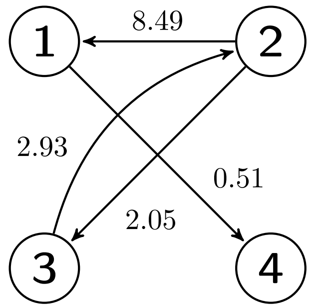


As with pairwise-stable networks, for a given potential network, set of feasible networks, and associated utilities, there may be zero or multiple coordination-proof networks. Any coordination-proof network is also pairwise-stable but not necessarily vice-versa. Therefore, the number of coordination-proof networks will be less than or equal to the number of pairwise-stable networks. In the case of the production network example discussed throughout this section, about one third of the pairwise-stable networks are coordination-proof.

2.2. Computation of Equilibrium Networks

To compute a pairwise-stable or coordination-proof equilibrium network requires a set of feasible networks and, for each such feasible network, a list of utilities for each agent. Networks are typically represented as matrices for the purposes of computation. The edges are represented as an *adjacency matrix*, $A = [a_{ij}]$. If there is an edge from node j to node i, $a_{ij} = 1$, otherwise $a_{ij} = 0$. The edge weights are stored in an associated matrix, $W = [w_{ij}]$, where w_{ij} is the weight on the edge from j to i if there is one, and zero otherwise. The network described by the below adjacency and edge weight matrices is depicted in Figure 9.

Figure 9: Adjacency Matrix, Weight Matrix, and Network



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

$$W = \begin{bmatrix} 0 & 8.49 & 0 & 0 \\ 0 & 0 & 2.93 & 0 \\ 0 & 2.05 & 0 & 0 \\ 0.51 & 0 & 0 & 0 \end{bmatrix}$$

A potential network can be used to enumerate every feasible network. As mentioned above, what constitutes a feasible network will depend on the economic context at hand. In the example discussed throughout this chapter, the set of feasible networks is the set of all networks such that every node has exactly one edge pointing to it. The code to recursively build this set from a given potential network is included in the Appendix.

The utility generating process is also context-specific. For the purposes of computing equilibrium networks, all that is required is a set of utilities for each agent associated with each feasible network. To find the set of equilibrium networks requires checking if each feasible network meets the conditions of the particular equilibrium definition. To do this, the possible deviations from the given feasible network are enumerated. For pairwise stability, this requires enumerating all the possible alternative relationships for each agent. For coordination proof networks, this requires enumerating all the possible alternative relationships for every *combination* of agents. Each set of alternative relationships defines a deviation to another feasible network.

Once the alternative feasible network is found, the utilities are compared. If both the agents and their potential alternative relatives have higher utility in the alternative feasible network, the original feasible network is not an equilibrium network. That feasible network is rejected as an equilibrium. If at least one of the agents or potential alternative relatives do not have higher utility in the alternative feasible network, the next possible deviation is checked. If no deviations are utility-improving, the current feasible network is an equilibrium network. This network is added to the set of equilibrium networks and the next feasible network is checked. The result is a set of equilibrium networks that may be the empty set or may contain multiple elements.

Here I present an outline for algorithms to compute the set of pairwise-stable equilibrium networks and to compute the set of coordination-proof networks. They take as input a potential network and return a set of networks which satisfy the conditions of pairwise-stable and coordination-proof networks, respectively. Note that this set may be the null set or it may contain multiple elements.

2.3. An Algorithm for Computing Pairwise-Stable Networks

1. Initialize the set of pairwise-stable networks as empty.

2. Enumerate the set of feasible networks.

3. For each feasible network:

   (a) Enumerate the pairwise deviations.

   (b) For each deviation:

      i. Check if both firms in the pair are made better off.

      ii. If they are, stop.

      iii. If they are not, check the next deviation.

   (c) If any of the deviations are profitable, this feasible network is not pairwise-stable; stop.

   (d) If there are no profitable deviations, this feasible network is pairwise-stable; add it to the set of pairwise-stable networks.

4. Return the set of pairwise-stable networks.

2.4. An Algorithm for Computing Coordination-Proof Networks

1. Initialize the set of coordination-proof networks as empty.
2. Enumerate the set of feasible networks.
3. For each feasible network:
   (a) Enumerate the group deviations.
   (b) For each deviation:
      i. Check if all of the firms in the group are made better off.
      ii. If they are, stop.
      iii. If they are not, check the next deviation.
   (c) If any of the deviations are profitable, this feasible network is not coordination-proof; stop.
   (d) If there are no profitable deviations, this feasible network is coordination-proof; add it to the set of coordination-proof networks.
4. Return the set of coordination-proof networks.

The Appendix contains the Matlab code for computing the set of pairwise-stable networks and coordination-proof networks for the production network context discussed in the examples throughout this chapter.

3. Production Networks and the Loss of an Input

In this section, I apply this model of network formation to the context of production networks and use it to understand if aggregate output necessarily decreases when a firm loses access to an input that it uses in production. Networks are a useful way to model production; they explicitly capture both the direct and indirect effect of firms' decisions on one another. When a firm loses access to its chosen input supplier, it must choose another supplier. Before this change, this new supplier must have been sub-optimal, otherwise it would have been used. Intuitively, when a firm switches to this previously sub-optimal input supplier, we would expect this firm's costs to increase and because of the interconnectedness of the network, all prices to increase and aggregate output to decrease. However, because firms can change their input decisions in the face of these new prices, a new equilibrium network emerges. Output need not fall in aggregate. In fact, we will see that there are situations where output increases.

Following notation similar to that in Oberfield (2013), let there be a finite set, J, of firms, represented by the nodes of the network. Each firm produces a single distinct good. All of the firms' output is consumed by either other firms as inputs or by a representative consumer. In the potential network, there is an edge from node i to node j if firm j can use firm i's product as an input. Each edge, e, from i to j has a weight, $z(e)$, which describes the technological match of that input in firm j's production. Each input available to firm j, that is, each edge pointing to j in the potential networks, defines a specific production function that depends on the edge weight, $z(e)$. The amount of firm j's product that it can produce using the input specified by edge e is

$$y_j = \frac{1}{\alpha^\alpha(1-\alpha)^{1-\alpha}} z(e)x(e)^\alpha l_j^{1-\alpha}$$

where $x(e)$ is the amount of the input that firm j uses and l_j is the amount of labor that firm j uses. The parameter, α, is common across all of the firms and input options.

Because these production functions exhibit constant returns to scale, in equilibrium each firm will choose only one input supplier. If a firm were using two inputs, one would offer the larger marginal return and the firm would substitute entirely over to that input. As a result, the set of feasible networks is the set of all subnetworks such that each node has exactly one edge pointing to it. Note that there are no restrictions on the number of edges pointing away from a given node. This will be determined by the equilibrium network. See the Appendix for a discussion of the effects of using a production function that leads to each firm only using one input.

Let y_j^0 be the amount of y_j that is consumed by the representative consumer. The representative consumer has preferences over the goods produced by the firms in J,

$$U(y_1, \ldots, y_{|J|}) = \Big(\sum_{j \in J} (y_j^0)^{\frac{\epsilon-1}{\epsilon}}\Big)^{\frac{\epsilon}{\epsilon-1}}$$

and supplies L units of labor, inelastically.

3.1. The Planner's Problem

Here I describe the planner's problem to both provide a basis for comparison for other equilibrium outcomes and to build intuition for the model. The planner considers all of the feasible networks and for each solves a standard consumer utility maximization problem.

$$\max_{\left\{\{y_j^0, x(e_j), l_j\}_{j \in J}\right\}_{N \in \mathcal{N}}} \Big(\sum_{j \in J} (y_j^0)^{\frac{\epsilon-1}{\epsilon}}\Big)^{\frac{\epsilon}{\epsilon-1}} \equiv Y^0$$

s.t.

$$y_j^0 + \sum_{e \in \hat{D}_j} x(e) \leq \frac{1}{\alpha^\alpha (1-\alpha)^{1-\alpha}} z(e_j) x(e_j)^\alpha l_j^{1-\alpha} \qquad \forall j \in J$$

$$\sum_{j \in J} l_j = L$$

Each $N \in \mathcal{N}$ defines a different edge pointing to each firm j, labeled e_j, and a different set of edges pointing away from firm j to each of its customers, labeled $\hat{D}_j$. Thus, the output of each firm is split into the amount consumed by the representative consumer and the amount consumed as inputs by other firms: $y_j = y_j^0 + \sum_{e \in \hat{D}_j} x(e)$. The first constraint is the technology constraint: the consumer and the customers of firm j cannot consume more than firm j produces using the input defined by N. The second constraint is the labor constraint: all of the firms in J use only the labor supplied by the representative consumer.

Each of the feasible networks in $\mathcal{N}$ defines a different maximization problem across $\{y_j^0, x(e_j), l_j\}_{j \in J}$, each of which the planner solves. The planner then selects the network and choice variables corresponding to the largest Y^0. The Lagrangian for each feasible is:

$$\mathcal{L} = Y^0 + \sum_{j \in J} \lambda_j \Big[\frac{1}{\alpha^\alpha (1-\alpha)^{1-\alpha}} z(e_j) x(e_j)^\alpha l_j^{1-\alpha} - y_j^0 - \sum_{e \in \hat{D}_j} x(e)\Big] + \mu \Big[L - \sum_{j \in J} l_j\Big].$$

Here μ is the shadow cost of labor and λ_j is the shadow cost of product j. Define the individual efficiency of each firm, $q_j \equiv \frac{\mu}{\lambda_j}$. This will be useful in characterizing individual and aggregate outcomes, as defined in the following theorems.

Theorem 1. *In the solution to the planner's problem the efficiency of a given firm is a function of the efficiency of the input supplier of that firm. Specifically,* $q_j = z(e_j) q_{s(e_j)}^\alpha$, *where* $s(e_j)$ *is the* *identity of the input supplier used by j.*

Proof. The technology constraint gives

$$\lambda_j = \frac{1}{z(e_j)}\lambda^{\alpha}_{s(e_j)}\mu^{1-\alpha}.$$

Using the definition of q_j and then rearranging,

$$\frac{\mu}{q_j} = \frac{1}{z(e_j)}\big(\frac{\mu}{q_{s(e_j)}}\big)^{\alpha}\mu^{1-\alpha}$$

$$q_j = \mu\Big(z(e_j)\big(\frac{q_{s(e_j)}}{\mu}\big)^{\alpha}\frac{1}{\mu^{1-\alpha}}\Big)$$

$$q_j = z(e_j)q^{\alpha}_{s(e_j)}$$

□

Theorem 2. *The measure of aggregate output, Y^0, is a function of the efficiencies of all of the individual firm efficiencies.*

$$Y^0 = \big(\sum_{j\in J} q_j^{\epsilon-1}\big)^{\frac{1}{\epsilon-1}} \cdot L$$

See the Appendix for a proof. The proofs of Theorems 1 and 2 are very similar to those in Oberfield (2013), but are done for a discrete set of firms rather than a continuum and therefore require separate proofs.

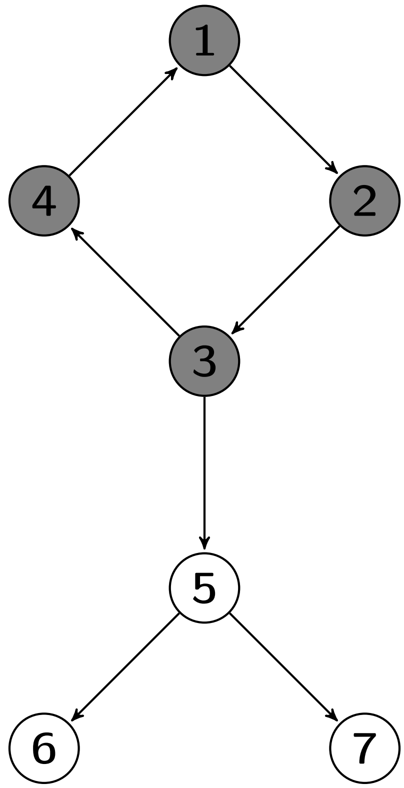
3.2. Equilibrium Production Networks

When exactly one edge is pointing to each firm, there are only two possible network shapes that can make up each connected component of the entire network. These are cycles with branches. A cycle is a set of nodes such that the in-degree and out-degree of each node are both exactly one. A branch is a set of nodes such that the in-degree of each node is one but the out-degree is unrestricted. See Figure 10. Any connected component must contain exactly one cycle and any branch in that connected component must have its root on the cycle.[2] These two shapes are critical

[2]If it had no cycle then there would need to exist one node with no supplier and if there was more than one cycle then there would exist some node with more that one supplier.

in calculating the prices.

Figure 10: The gray nodes form a cycle; the white nodes form a branch.



Firms set prices for both the portion of their output consumed by the representative consumer and the portion consumed by each of their network customers - the other firms which use their good as an input. Label the price of y_j^0 as p_j^0. For each network customer of firm j, j charges a two-part tariff. That is, j sets $\{p(e), \tau(e)\}_{e \in \hat{D}_j}$, for each $\hat{D}_j$ defined by each $N \in \mathcal{N}$. Here $\tau(e)$ is fixed fee and $p(e)$ is the price per unit of product j.

The per-unit price, $p(e)$, that firm j charges the customer to which edge e points is firm j's marginal cost of production.[3] As a result of this, the per-unit price firm j charges is a function of the marginal cost of the input supplier used by firm j, $p(e) = MC_j = \frac{1}{z(e_j)} MC_{s(e_j)} w^{1-\alpha}$, where w is the price of labor to all firms. Because the price charged by each firm can be written in terms of the supplier's marginal cost, all of these prices can be calculated using only the network structure and $z(e_j)$'s. The price charged by any firm on a cycle can be traced back through each supplier

[3]For a proof of this, see Oberfield (2013). While the proof is presented in the context of a continuum of firms, this relationship depends entirely on discrete, firm-to-firm links and will be essentially equivalent in the context of this paper.

until it is expressed in terms of itself, thus there is a closed form solution for any price on a cycle. The price charged by any firm on a branch can be traced up to the root node of the cycle, which must be on a cycle, thus any such price can be calculated. See the Appendix for formal derivation.

The profit maximization problem each firm j solves is

$$\max_{p_j^0, y_j^0, x(e_j), l_j} p_j^0 y_j^0 + \sum_{e \in \hat{D}_j} [p(e)x(e) + \tau(e)] - [p(e_j)x(e_j) + \tau(e_j)] - wl_j$$

s.t.

$$y_j^0 + \sum_{e \in \hat{D}_j} x(e) \leq \frac{1}{\alpha^\alpha (1-\alpha)^{1-\alpha}} z(e_j) x(e_j)^\alpha l_j^{1-\alpha}.$$

Each $N \in \mathcal{N}$ defines a different profit maximization problem for each firm and the solutions to these produce a set of profits for each firm in each feasible network. These profits are the payoffs that determine the pairwise-stable and the coordination-proof equilibria, as defined in the previous section.

Just as the efficiency of an individual firm in the planner's problem is defined as the ratio of the two shadow costs μ and λ_j, define the efficiency of the individual firms in this case as, $\widetilde{q}_j \equiv \frac{w}{MC_j}$. As in the case of the planner's problem, this can be written in terms of the efficiency of firm j's input supplier.

Theorem 3. *The efficiency of a given firm is a function of the efficiency of the input supplier of that firm, $\widetilde{q}_j = z(e_j)\widetilde{q}^\alpha_{s(e_j)}$, where $s(e_j)$ is the identity of the input supplier used by j.*

Proof. Cost minimization and the result above,

$$MC_j = \frac{1}{z(e_j)} MC^\alpha_{s(e_j)} w^{1-\alpha}.$$

The definition of $\widetilde{q}_j$ gives,

$$\frac{w}{\widetilde{q}_j} = \frac{1}{z(e_j)} \Big(\frac{w}{\widetilde{q}_{s(e_j)}}\Big)^\alpha w^{1-\alpha}$$

$$\widetilde{q}_j = z(e_j)\widetilde{q}_{s(e_j)}.$$

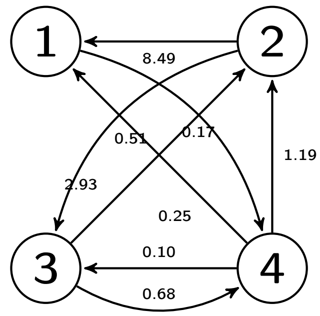
□

3.3. Input Removal

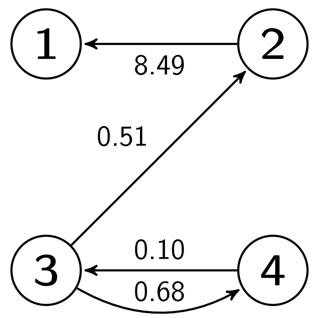
I compare the aggregate output generated by an equilibrium network before and after an edge of the network is removed. Let e^* be the deleted edge and j^* be the identity of the firm to which e^* points. That is, j^* uses e^* in its production. When this edge is deleted, it creates a new potential network, and thus a new set of feasible networks. This new set of feasible networks is a strict subset of the original set. The new equilibrium network is determined from among this new set of feasible networks.

An intuitive analysis of the result from this edge deletion may suggest a path-dependent cascade of the drop in j^*'s efficiency along all of the edges emanating from j^* and its customers, holding the other equilibrium network links fixed. However, this is not necessarily an equilibrium. See Figure 11 for an example. Figure 11(a) shows the potential network and Figure 11(b) shows the original coordination-proof equilibrium. Note that this is also then a pairwise-stable equilibrium. If the edge from firm 2 to firm 1 is deleted and the other edges are held fixed, while firm 1 chooses the lowest marginal cost supplier available to it - firm 4 - then the network will be as shown in Figure 11(c). However, this network is neither pairwise-stable nor coordination-proof. The coordination-proof equilibrium that results from deleting the edge from firm 2 to firm 1 is shown in Figure 11(d).

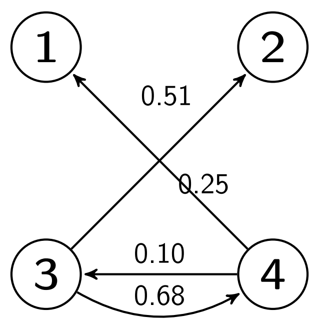
Figure 11: Deleting an edge and holding the other edges fixed is not neccessarily an equilibrium



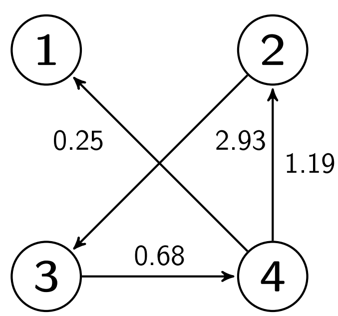
(a) Potential Network



(b) Original Equilibrium Network



(c) Holding other edges fixed



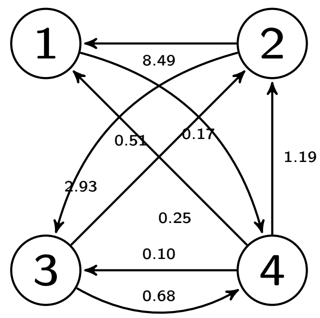
(d) The equilibrium network when the edge is deleted

While in the case of a solution to the planner's problem, the new output will be lower than the original output, this is not necessarily true in the case of a new pairwise-stable equilibrium, and, in fact, this result survives the equilibrium refinement of the coordination-proof equilibrium.

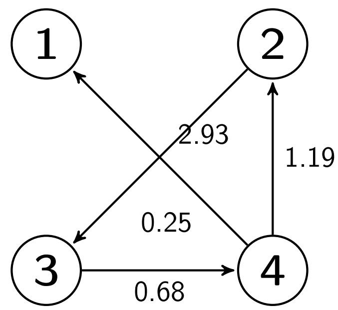
Result 1. *There exist parameters of the model such that the output produced by a pairwise-stable equilibrium or by a coordination-proof equilibrium increases when an edge is deleted and a new equilibrium of the same type is determined.*

This will occur when the edge that is deleted is the only edge preventing a higher-output network from being pairwise stable or coordination proof. If, when an edge is deleted the set of new equilibrium networks is a strict subset of the original set of equilibrium networks, then certainly output will decrease. However, this need not be the case. There are situations in which deleting an edge makes it possible for a new network to be pairwise stable or coordination proof. If the output produced by such a new network is higher than the output in the original network, then output will increase. See Figure 12 for an example. There are three coordination-proof (and pairwise-stable) networks corresponding to the potential network shown in Figure 12(a). Of those, the one that offers the highest output, 0.1430, is depicted in Figure 12(b). When the edge from firm 4 to firm 1 is deleted, the new set of coordination-proof equilibria consists of five networks. From those, the highest possible output is now 0.1904. The network that produces this output is depicted in Figure 12(c).

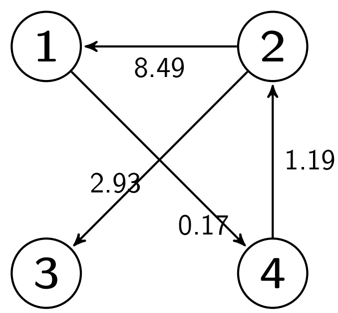
Figure 12: Output increases when the edge from firm 4 to firm 3 is deleted.



(a) Potential Network



(b) Original Equilibrium Network, Output = 0.1430



(c) New Equilibrium, Output= 0.1904

3.4. Simulation Results

I simulate this model by generating potential production networks and then finding a solution to the planner's problem, a pairwise-stable equilibrium, and a coordination proof equilibrium. I create the potential network by drawing a number of possible input suppliers for each firm from a Poisson(3) distribution. The identity of each supplier is drawn uniformly with replacement from the other firms. The productivity parameter, $z(e)$, for each edge e is drawn from a Pareto$(0.2, -1.8)$ distribution. This parameterization is motivated by the Carvalho (2012) survey on Input-Output analysis. Note that drawing the supplier identities with replacement allows for multiple edges from a given firm. However, because the productivity parameters are realizations of continuous random variables, the edge with the higher z will always be chosen. These simulations consisted of the creation of $1,000$ potential networks, each of which consisted of five firms. Enough simulation repetitions were conducted to ensure statistical significance of the coefficient estimates except in one case noted below. In 554 of these potential networks, a solution to all three problems was found.

This simulation builds a random sample of potential production networks with five firms from the universe of potential production networks consisting five firms. The number of available input suppliers, the identities of the input suppliers, and the pairwise match values (edge weights) are created using a Sobol Set quasi-random number generator. As noted in the previous section, there may be multiple pairwise-stable or coordination-proof equilibrium networks. When this occurs, I select the network that creates the largest aggregate output. By doing this, I ensure that any increase in output after an input relationship is removed is not the result of multiple equilibria.

3.5. Removal of an Input Supplier

To investigate the removal of an input supplier, I run edge deletion experiments on three different sets of networks: solutions to the planner's problem, pairwise-stable networks, and coordination-proof networks. I find one of each of these for each potential network created. Then for each network, I remove each edge, one at a time. I do this by removing that edge from the potential network and then finding a new solution of each type. This resulted in $2,770$ edge deletion experiments. In $2,691$ of these edge deletion experiments, a new solution to all three allocations is found. I take the ratio of output after the edge is deleted to output before the edge is deleted. Label this "relative output" for each allocation: planner, pairwise-stable, and coordination-proof.

I measure the connectivity of each original equilibrium using the average shortest path distance. I do this by calculating the length of the directed path from each node to every other node and taking the average across all such paths. A larger average shortest path distance corresponds to a less connected network and a shorter average shortest path distance corresponds to a more connected network. For each edge deletion, i, I regress the relative output of each allocation on the average shortest path distance of the corresponding original equilibrium network. That is, I estimate the following regression equation.

$$\widehat{\text{relative output}}^{E} = \beta(\text{avg. shortest path distance})$$

for each $E \in \{\text{Planner's Solution, Pairwise-Stable, Coordination-Proof}\}$ using ordinary least squares. The results are reported in Table 1.

Table 1: Output and Connectivity

E	$\hat{\beta}$
Planner's Solution	0.3907
Pairwise-Stable	0.2812
Coordination-Proof	0.3803

Each of the three estimated regression coefficients is positive, indicating that a larger average shortest path distance is correlated with a larger relative output. This means that more connected equilibrium networks are correlated with a smaller relative output after an edge is deleted, and this result holds over all three allocations.

Each edge deletion experiment, i, defines a firm, j^*, that loses its input supplier. Label the number of input suppliers available to j^* in the potential network as $\#\text{sup}_i$ and the number of customers j^* has in the original equilibrium network $\#\text{cust}_i$. For each equilibrium type, $E \in \{\text{Planner's Solution, Pairwise-Stable, Coordination-Proof}\}$, I estimate

$$\widehat{\text{relative output}}^{E} = \gamma_1(\#\text{sup}) + \gamma_2(\#\text{cust})$$

using ordinary least squares. The results are reported in Table 2.

Table 2: Output and Centrality		
E	$\hat{\gamma}_1$	$\hat{\gamma}_2$
Planner's Solution	0.2615	0.0325
Pairwise-Stable	0.2014	-0.0175
Coordination-Proof	0.2518	0.0178

The estimated regression coefficients on the number of available suppliers are positive for all three equilibrium definitions. This indicates that a larger number of available input suppliers is correlated with a larger relative output after an input is deleted.

The estimated regression coefficients on the number of customers in the original equilibrium network are all positive except for in the pairwise stable case. These positive coefficients indicate that the more customers j^* has when it loses its input supplier, the higher the relative output will be. However, in the case of the pairwise-stable and coordination-proof equilibria, the confidence intervals of $\hat{\gamma}_2$ include zero, and this was true for every size of simulation that I conducted. While it may be the case that in the planner's problem, a larger number of customers is correlated with a larger relative output, this cannot be concluded for pairwise-stable nor coordination-proof networks.

3.6. Increased Output

I investigate the role that network connectivity and firm centrality play in the probability that output increases after an input is removed from the production network. In doing this, I restrict my focus to the coordination-proof edge deletion experiments because any coordination-proof equilibrium is also pairwise stable.[4] To understand the role of network connectivity, I consider the average shortest path distance of the potential network, the original equilibrium production network, and the new equilibrium production network that results after the input is removed. I also include characteristics of j^*. Specifically, I consider the number of suppliers from which j^* can choose a new input, the number of firms that buy j^*'s product in the original equilibrium production network, and the number of firms that buy j^*'s product in the new equilibrium production network. These six characteristics are the explanatory variables in a binary Logistic regression for which the dependent variable is the probability that relative output is greater that one; that is, that output

[4]The solution to the planner's problem is omitted because the output always decreases when an input is removed.

increased after the edge was removed. The marginal effects of the explanatory variables in this regression are reported in Table 3.

Table 3: Logit Regression Marginal Effects

Characteristic	ME
Potential Network Average Shortest Path Distance	-0.0280
Original Equilibrium Average Shortest Path Distance	0.0671
New Equilibrium Average Shortest Path Distance	-0.0260
Number of Possible Suppliers	0.0073
Original Number of Customers	-0.0317
New Number of Customers	0.0253

The results regarding the connectivity of the production network indicate the following. First, the more connected the potential network is, the higher the likelihood that output will increase. A one-link increase in the average shortest path distance of the potential network is associated with a 2.8 percentage point decrease in the likelihood that output increases. Second, the less connected the original equilibrium production network and the more connected the new equilibrium production network, the higher the likelihood that output will increase. A one-link increase in the average shortest path distance of the original equilibrium is associated with a 6.71 percentage point increase in the likelihood that output increases, while a one link increase in the average shortest path distance of the new equilibrium is associated with a 2.6 percentage point decrease in the likelihood that output increases.

The firm characteristic results indicate the following. First, the more suppliers available to j^*, the higher the likelihood that output will increase. One more supplier is associated with a 0.73 percentage point increase in the probability that output increases. Second, the fewer firms buying from j^* in the original equilibrium production network and the more firms buying from j^* in the new equilibrium production network, the higher the likelihood that output increases. One more customer in the original equilibrium is associated with a 3.17 percentage point decrease in the probability that output will increase, while one more customer in the new equilibrium is associated with a 2.53 percentage point increase in the probability.

4. Conclusion

The key contributions of this paper are a new network model which features a finite number of firms and endogenous network determination, a refinement of the standard network equilibrium, and a better understanding of the role network connectivity and firm centrality play in the determination of aggregate outcomes. I apply the model to investigate the effect of a firm losing an input supplier in the production network and find that when this happens the resulting aggregate output can be higher. Simulation results indicate the following. First, on average, the more connected a production network is, the smaller the decrease in aggregate output will be when a firm loses an input supplier. Second, on average, the more alternative suppliers that firm has, the smaller the drop in output will be when that firm loses its input supplier. Finally, it is more likely output will increase when (1) the firm that loses its supplier goes from having fewer customers before it lost its supplier to having many customers afterwards and (2) the network as a whole goes from less connected before this input is removed to more connected afterward this input is removed.

Acknowledgements

This work was supported by the Jefferson Scholars Foundation, Charlottesville, VA [John E. Walker Endowed Fellowship].

References

[Acemoglu and Azar, 2017] Acemoglu, Daron, and Azar, Pablo D. (2017). Endogenous Production Networks. *National Bureau of Economic Research* No. w24116

[Acemoglu et al., 2012] Acemoglu, Daron, Carvalho, Vasco M., Ozdaglar, Asuman, and Tahbaz-Salehi, Alireza. (2012). The Network Origins of Aggregate Fluctuations. *Econometrica.* Vol. 80 No. 5

[Baqaee, David Rezza, 2013] Baqaee, David Rezza. (2013). Cascading Failures in Production Networks. *London School of Economics and Political Science.*

[Carvalho and Voigtländer, 2015] Carvalho, Vasco M. and Voigtländer, Nico. (2015). Input Diffusion and the Evolution of Production Networks. *Economic Working Papers 1418*

[Chartrand, 1977] Chartrand, Gary. (1977). Introductory Graph Theory. *Dover Publication. New York, NY.*

[Dalal et al., 2008] Dalal, Ishaan L., Stefan, Deian, and Harwayne-Gidansky, Jared. (2008). Low Discrepancy Sequences for Monte Carlo Simulations on Reconfigurable Patters. *Application-Specific Systems, Architectures and Processors, 2008. ASAP 2008. International Conference on. IEEE, 2008*

[Dawson, 2018] Dawson, Chester. (2018). Toyota to Suspend Output at Japanese Plants After Earthquake. *Wall Street Journal.* September 7, 2018.

[di Giovanni et al., 2014] di Giovanni, Julian, Levchenko, Andrei A., and Mejean, Isabelle. (2014). Firms, Destinations, and Aggregate Fluctuations. *Econometrica.* Vol. 82 No. 4

[Dupor, 1999] Dupor, Bill. (1999). Aggregation and Irrelevance In Multi-Sector Models. *Journal of Monetary Economics* Vol. 43 Issue 2

[Hildenbrand and Kirman, 1988] Hildenbrand, W. and Kirman, A.P. (1988). Equilibrium Analysis. *Elsevier Science Publishers B.V. Princeton, NJ.*

[Horvath, 1998] Horvath, Michael. (1998). Cyclicality and Sectoral Linkages: Aggregate Fluctuations from Independent Sectoral Shocks. *Review of Economic Dynamics.* Vol. 1 Issue 4

[Jackson, 2010] Jackson, Matthew O. (2010). Social and Economic Networks. *Princeton University Press.*

[Jackson and Wolinksy, 1996] Jackson, Matthew O. and Wolinsky, Asher (1996). A Strategic Model of Social and Economic Networks. *The Journal of Economic Theory.* Vol. 71 Article No. 0108 p. 44 - 74

[Oberfield, 2013] Oberfield, Ezra. (2013). Business Networks, Production Chains, and Productivity: A Theory of Input-Output Architecture. *Working Paper Series Federal Reserve Bank of Chicago*, WP-2011-12.

[PBS Newshour] "How a sophisticated malware attack is wreaking havoc on Ukraine." *PBS Newshour*, PBS. WETA, Washington, D.C. 28 June 2017. Television.

[Shapley, 1967] Shapley, Lloyd S. (1967). On Balanced Sets and Cores. *Naval Research Logistics* Vol. 14 Issue 4

[Taschereau-Dumouchel, 2017] Taschereau-Dumouchel, Mathieu (2017). Cascades and fluctuations in an economy with an endogenous production network.

[Tucker, 2007] Tucker, Alan. (2007). Applied Combinatorics. *John Wiley and Sons. Hoboken, NJ.*

The data generated in the simulations are available from the author upon reasonable request.

Declarations of interest: none

5. Appendix

5.1. Proof of Theorem 2: Aggregate Output and Efficiency in The Planner's Problem

Each firm j uses a number of supply chains in the production process. A supply chain is a string of firms, each producing intermediate goods for the next firm in the chain. Label the set of supply chains which lead to firm j as $\mathcal{S}_j$. Partition firm j's final output, y_j^0, by final output produced by each supply chain, $s \in \mathcal{S}_j$. That is, $\sum_{s \in \mathcal{S}_j} y_j^0(s) = y_j^0$. To make $y_j^0(s)$, firm j uses labor $l_j^0(s) \leq l_j$ and intermediate input $x_j^0(s) \leq x_j$. This $x_j^0(s)$ is produced by firm j's supplier using $l_j^1(s)$ and $x_j^1(s)$, this $x_j^1(s)$ is produced using $l_j^2(s)$ and $x_j^2(s)$, and so on up the supply chain. In general, I write $l_j^{k+1}(s)$ and $x_j^{k+1}(s)$ are the labor and intermediate input amounts used to make $x_j^k(s)$, along supply chain s to make firm j's output for final consumption. The lower subscript describes the good at the end of the supply chain and the superscript describes the step up the supply chain,

Let $\lambda_j^0(s)$ be the marginal social cost of producing $x_j^0(s)$ and let μ be the marginal social cost of labor, following the earlier notation. The optimal choices of $x_j^0(s)$ and $l_j^0(s)$ give:

$$\frac{\lambda_j^k(s) x_j^k(s)}{\alpha} = \frac{w l_j^k(s)}{1-\alpha}$$

$$\frac{\lambda_j^{k+1}(s) x_j^{k+1}(s)}{\alpha} = \frac{w l_j^{k+1}(s)}{1-\alpha}.$$

Using the technological constraint,

$$\lambda_j = \frac{1}{z(e_j)} \lambda_{s(e_j)}^{\alpha} \mu^{1-\alpha}$$

where $s(e_j)$ is the supplier of edge e_j. This means that the marginal social cost of producing product j is determined by the marginal social cost of firm j's supplier. This will be necessary for connecting efficiency across the entire network. The network structure gives $\lambda_j^{k+1}(s)x_j^{k+1}(s) = \alpha\lambda_j^k(s)x_j^k(s)$, so the optimality condition for the $(k+1)$th step up the supply chain becomes $\lambda_j^k(s)x_j^k(s) = \frac{wl_j^{k+1}(s)}{1-\alpha}$. Substituting this into the optimality condition for the kth step gives $l_j^{k+1}(s) = \alpha l_j^k(s)$. That is, the labor used in each step along the supply chain to make intermediate goods for $y_j^0(s)$ is a constant share of the labor used in the previous step.

Let $l_j(s)$ be the total labor used along supply chain s to make $y_j^0(s)$ such that $l_j(s) = \sum_{k=1}^{\infty} l_j^k(s) = \sum_{k=1}^{\infty} \alpha^k l_j^0(s) = \frac{l_j^0(s)}{1-\alpha}$. From the optimality condition for the final step in the supply chain,

$$\lambda_j y_j^0(s) = \frac{wl_j^0(s)}{1-\alpha} = wl_j(s).$$

Finally, write $y_j^0 = \sum_{s\in\mathcal{S}_j} y_j^0(s) = \sum_{s\in\mathcal{S}_j} \frac{w}{\lambda_j} l_j(s) = \frac{w}{\lambda_j}\sum_{s\in\mathcal{S}_j} l_j(s) = \frac{w}{\lambda_j} l_j$. From the definition of q_j, I can now write $y_j^0 = q_j l_j$.

From here, I use the solution to this planner's problem to show that $Y^0 = L\Big[\sum_{j=1}^{J} q_j^{\epsilon-1}\Big]^{\frac{1}{\epsilon-1}}$. Let $\Big[\sum_{j=1}^{J_t} q_j^{\epsilon-1}\Big]^{\frac{1}{\epsilon-1}} \equiv Q$. First, the first order condition with respect to firm y_j^0 is $\big(Y^0\big)^{\frac{1}{\epsilon}}(y_j^0)^{-\frac{1}{\epsilon}} = \lambda_j$. Rearranging this gives $\lambda_j = \Big(\frac{y_j^0}{Y^0}\Big)^{-\frac{1}{\epsilon}}$. This expression can be used to show that $\sum_{j=1}^{J_t} \lambda_j^{1-\epsilon} = 1$.

Rewriting the definition of firm efficiency for q_j allows me to write $\sum_{j=1}^{J_t}\big(\frac{w}{q_j}\big)^{1-\epsilon} = 1$. Then, using the definitions of q_j and Q, I can write $\frac{y_j^0}{Y^0} = \Big(\frac{q_j}{Q}\Big)^{\epsilon}$.

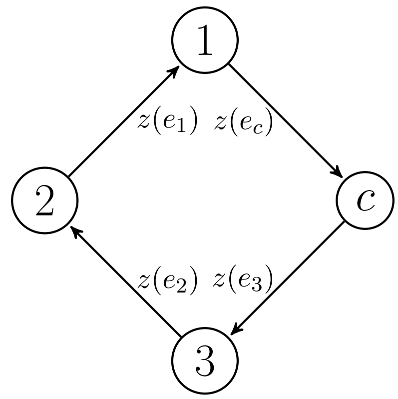
Finally, I use the labor constraint to show $L = \sum_{j=1}^{J_t}\Big(l_j\Big) = \sum_{j=1}^{J_t}\Big(\frac{y_j^0}{q_j}\Big) = Y^0 Q^{-\epsilon}\sum_{j=1}^{J_t} q_j^{\epsilon-1} = Y^0 Q^{-\epsilon} Q^{\epsilon-1} = \frac{Y^0}{Q}$. Rewritten, this is the expression I need: $Y^0 = QL$.

5.2. Derivation of Price Expressions

Both of the following derivations are driven by the fact that the price that each firm pays for each unit of the input they use is the marginal cost of the supplier of that input, as proved above.

1. Price Paid by Firms on a Cycle

For a cycle of length c, there are c firms and c edges. Label these firms $1, \ldots, c$. Without loss of generality, we find the price of firm c and label the supplier c uses as 1, the supplier 1 uses as 2 and so on.



The price that c pays for its input is $p(e_c) = \frac{1}{z(e_1)} p(e_1)^\alpha w^{1-\alpha}$, where $p(e_1)$ is the price firm 1 pays for its input. This price is $p(e_1) = \frac{1}{z(e_2)} p(e_2)^\alpha w^{1-\alpha}$, where $p(e_2)$ is the price firm 2 pays for its input. Continuing in this way I can write the price that firm $c-1$ pays as $p(e_{c-1}) = \frac{1}{z(e_c)} p(e_c)^\alpha w^{1-\alpha}$. Substituting each price expression into the previous one gives:

$$p(e_c) = p(e_c)^{\alpha^c} \left[\frac{1}{z(e_1)} \frac{1}{z(e_2)^\alpha} \cdots \frac{1}{z(e_c)^{\alpha^{c-1}}} \right] w^{(1-\alpha)+\alpha(1-\alpha)+\ldots\alpha^{c-1}(1-\alpha)}.$$

Solving for $p(e_c)$ gives:

$$p(e_c) = \left(\left[\frac{1}{z(e_1)} \frac{1}{z(e_2)^\alpha} \cdots \frac{1}{z(e_c)^{\alpha^{c-1}}} \right] w^{(1-\alpha)+\alpha(1-\alpha)+\ldots\alpha^{c-1}(1-\alpha)} \right)^{\frac{1}{1-\alpha^c}}$$

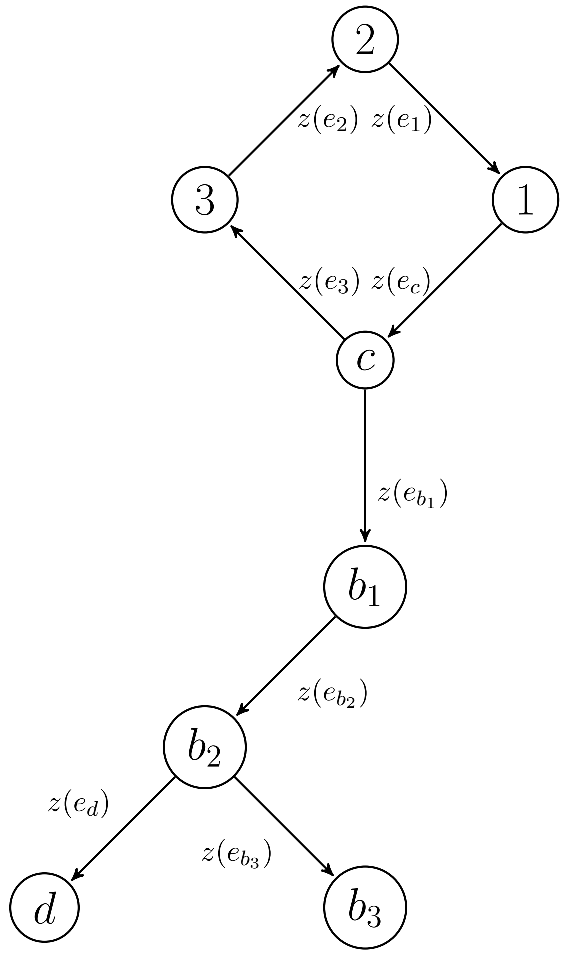
$$= w^{\frac{1-\alpha}{1-\alpha^c} \sum_{k=1}^{c} \alpha^{k-1}} \prod_{i=1}^{c} \left(\frac{1}{z(e_{i+1})} \right)^{\frac{\alpha^{i-1}}{1-\alpha^c}}$$

$$= w^{\frac{1-\alpha}{1-\alpha^c} \frac{\alpha^c - 1}{\alpha - 1}} \prod_{i=1}^{c} \left(\frac{1}{z(e_{i+1})} \right)^{\frac{\alpha^{i-1}}{1-\alpha^c}}$$

$$= w \prod_{i=1}^{c} \left(\frac{1}{z(e_{i+1})} \right)^{\frac{\alpha^{i-1}}{1-\alpha^c}}.$$

2. Price Paid by Firms on a Branch

Each connected component of the network has one cycle, and potentially many branches emanating from that cycle. Thus, each branch has a root node on the cycle. Because each firm pays the marginal cost of its supplier, the cost of any branch firm can be traced back and written in terms of the price of this root node. Let firm d be d edges down the branch from the node where $d > 1$.



The price that d pays for its input is the marginal cost of its supplier. The price the supplier pays is the marginal cost of his supplier and so on up to the root node, whose price was found in the above derivation. Label $MC_r = \frac{1}{z(e_r)} p_r^{\alpha} w^{1-\alpha}$. Then the price that firm d pays is

$$p(e_d) = MC_r^{\alpha^{d-1}} \left[\prod_{i=1}^{d-1} \frac{1}{z(e_{b_i})^{\alpha^{d-i-1}}} \right] w^{(1-\alpha)\sum_{k=0}^{d-2} \alpha^k}$$

$$= MC_r^{\alpha^{d-1}} \left[\prod_{i=1}^{d-1} \frac{1}{z(e_{b_i})^{\alpha^{d-i-1}}} \right] w^{(1-\alpha)\frac{\alpha^d - 1}{\alpha - 1}}$$

$$= MC_r^{\alpha^{d-1}} \Big[\prod_{i=1}^{d-1} \frac{1}{z(e_{b_i})^{\alpha^{d-i-1}}} \Big] w^{1-\alpha^d}.$$

For a firm that is only one edge away, for example firm b_1 in the figure, the price that firm pays is the marginal cost of the root node, MC_r.

5.3. Robustness to Multiple Inputs: A Case Study

Consider a production function that specifies two inputs for firm j, e_j^1 and e_j^2,

$$y_j = z(e_j^1)z(e_j^2)\Big[\big(\frac{x(e_j^1)}{\gamma}\big)^{\gamma}\big(\frac{x(e_j^2)}{1-\gamma}\big)^{1-\gamma}\Big]^{\alpha} l_j^{1-\alpha}.$$

The set of feasible networks expands to include not just networks in which every firm has one input but networks in which every firm has either one or two inputs. If in a given feasible network, firm j has one input, it solves the profit maximization problem described in the main body of the paper. If instead, a firm has two inputs, it solves the following profit maximization problem.

$$\max_{p_j^0, y_j^0, x(e_j^1), x(e_j^2), l_j} p_j^0 y_j^0 + \sum_{e \in \hat{D}_j} [p(e)x(e) + \tau(e)] - [p(e_j^1)x(e_j^1) + \tau(e_j^1)] - [p(e_j^2)x(e_j^2) + \tau(e_j^2)] - wl_j$$

s.t.

$$y_j^0 + \sum_{e \in \hat{D}_j} x(e) \leq z(e_j^1)z(e_j^2)\Big[\big(\frac{x(e_j^1)}{\gamma}\big)^{\gamma}\big(\frac{x(e_j^2)}{1-\gamma}\big)^{1-\gamma}\Big]^{\alpha} l_j^{1-\alpha}$$

I run an edge deletion experiment on a specific potential network as a case study in the robustness of this model with respect to the assumption of constant returns to scale. I find initial equilibrium networks for the two different production functions, and then compare the outcomes when edges are deleted. The potential network is depicted below, in Figure 13, with the edge weights specified in matrix W, below.

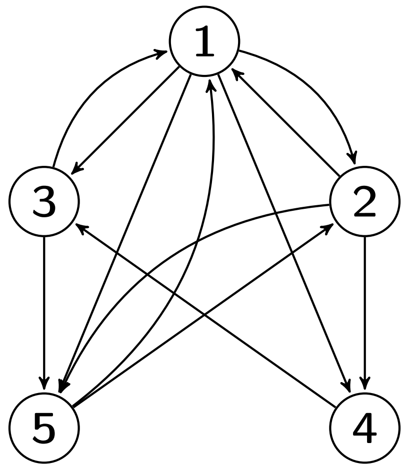


Figure 13: Potential Network

$$W = \begin{bmatrix} 0 & 0.16 & 2.84 & 0 & 0.08 \\ 0.33 & 0 & 0 & 0 & 0.98 \\ 0.21 & 0 & 0 & 17.31 & 0 \\ 0.10 & 1.26 & 0 & 0 & 0 \\ 5.03 & 0.25 & 0.07 & 0 & 0 \end{bmatrix}$$

Firms 1 and 5 have three available suppliers while the rest of the firms have two available suppliers. There are $3 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 72$ feasible networks in which every firm has only one input supplier. There are $[3 + \binom{3}{2}] \cdot [2 + \binom{2}{2}] \cdot [2 + \binom{2}{2}] \cdot [2 + \binom{2}{2}] \cdot [3 + \binom{3}{2}] = 972$ feasible networks in which firms have either one or two input suppliers.

When firms are restricted to one input, there are four coordination-proof networks, shown in Figure 14. These four constitute 5.6% of the 72 feasible networks. The mean output across these is 0.1586. The network with the highest output, 0.3117, is depicted in Figure 14(b).

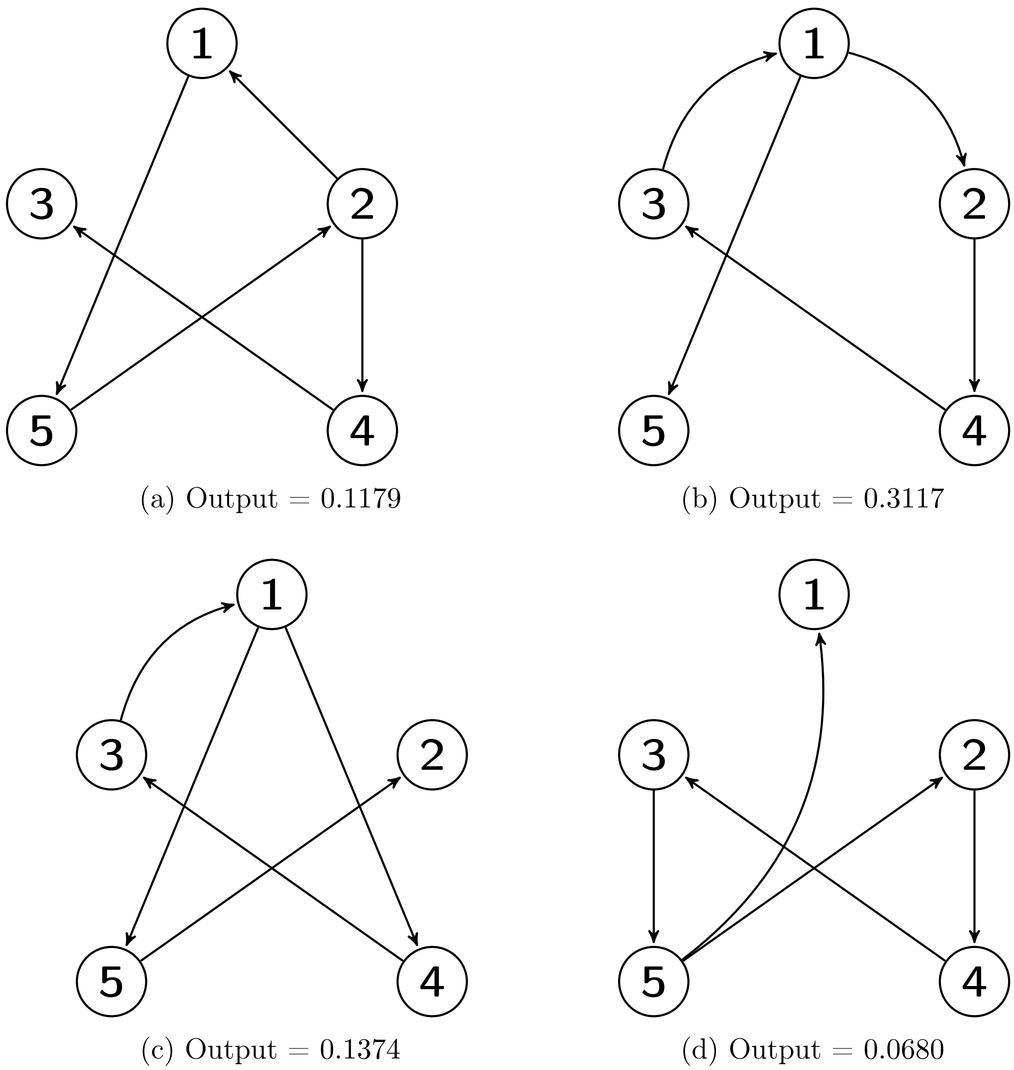


Figure 14: Coordination-Proof Networks for the 1 - Input Model

When firms can have either one or two inputs, there are 53 coordination-proof networks, constituting 5.5% of the 972 feasible networks. This percentage is very similar to the one-input case. Furthermore, the network with the highest output in the multiple-input case is the same network as in the one-input case, the network shown in Figure 14(b), with an output of 0.3117. The mean output across the 53 coordination-proof networks is 0.0895, lower than the mean output in the one-input case.

I delete two edges from this equilibrium - using both the one-input model and the two-input model - and compare the outcomes. I delete Firm 1 and Firm 5's input suppliers, one at a time. Both Firm 1 and Firm 5 have two more suppliers to choose from, but they differ in that Firm 1 has two network customers in the original equilibrium and Firm 5 has none.

First, I delete the edge from Firm 3 to Firm 1 from the original potential network and find the coordination-proof networks under both the one-input assumption and allowing for two inputs. When only one input is possible, the new set of coordination-proof networks consists of only two networks, both of which are in the original set of one-input, coordination-proof networks. They are the networks depicted in Figure 14(a) and (d). No new one-input equilibrium networks are created by deleting this edge. The mean output across the two fell to 0.0930. The highest output given by either of these is 0.1179. The network depicted in Figure 14(a) produces this output. In this case, when an edge is deleted, the resulting equilibrium network is the same whether firms are restricted to one input or whether they can have two inputs.

When two inputs are possible, the new set of equilibrium networks consists of nine networks, more than one of which has firms using two inputs. The mean across these fell to 0.0384. However, the network that produces the highest output in this case is the same network as in the one-input case, the network depicted in Figure 14(a), with an output of 0.1179.

Next, I delete the edge from Firm 1 to Firm 5 from the original potential network and find the coordination-proof networks for the one-input case and two-input case. When only one input is possible, there are three coordination proof networks. The mean across these is 0.0543. Two new coordination-proof networks are created when this edge is removed, but the network that produces the highest output is one of the four original coordination-proof networks, depicted in Figure 14(d). This network produces an output of 0.0680.

When two inputs are possible, the new set of coordination-proof networks consists of 41 networks. The mean output across these is 0.0383, which is very similar to the mean output when the previous edge is deleted and the two-input equilibrium is found, 0.0384. The maximum output created by any of these 41 networks is 0.1349. Note that this is a smaller decrease from the initial equilibrium than when firms are restricted to one input. The network that produces this output is depicted in Figure 15.

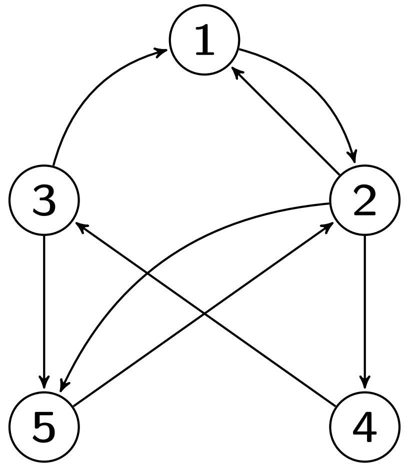


Figure 15: Highest Output Network, 2nd Edge Deletion, Two Inputs Possible

While further research is needed to understand the distributional effects of allowing for more than one input, this case study illuminates several facts. First, even when multiple inputs are allowed, the maximal output network may still have every firm using only one input. Second, despite dramatically increasing the number of feasible networks, the percentage of feasible networks that are coordination-proof does not necessarily dramatically change. Finally, allowing for two inputs may lead to a smaller drop in output when an edge is removed but the possibility of output increasing remains.

5.4. Matlab Code

```
%This function takes as input a potential network adjacency matrix, P, and
%returns a matrix, Pot_Eq, containing all the possible lists of suppliers
(that
%would make up an equilibrium network's adjacency matrix). It calls the
%recursive function rec_matrix.

function [Pot_Eq] = Find_Potentials(P)

%get function variables
Potential = P;

%How many firms?
num_firms = size(Potential,1);

%Construct the matrix of potential supplier indices
sup_matrix = zeros(num_firms, num_firms - 1);
%And count how many rows you'll need for the storing matrix.
num_rows = 1;
for i = 1:num_firms
    %What are the indices of the nonzero elements (suppliers)?
    index_list = find(P(i,:));
    %How many are there?
    num_options = length(index_list);
    temp_rows = num_rows;
    num_rows = temp_rows*num_options;
    %For each of those options put the index in the supplier matrix
    for s = 1:num_options
        index = index_list(s);
        sup_matrix(i,s) = index;
    end
end
%sup_matrix

%construct the temporary storing matrix
temp_store = zeros(num_rows,num_firms);

%construct initial vector
init_vector = zeros(num_firms,1);

%call the function for firm 0
return_mat = rec_matrix(0,init_vector,sup_matrix,temp_store);

Pot_Eq = return_mat;

end


%This is a recurseive function that enumerates all of the different
%   combinations of suppliers possible for a given potential network
%   adjacency matrix.

%It takes as input
%   - firm (row) we're on
%   - a working vector naming the suppliers used
%   - the potential adjacency matrix
```

```
%   - matrix to store the finished list for each possible eq. network

function [matrix] = rec_matrix(f, sup_vec, sup_mat, mat)

firm = f; %which firm are we on
supplier_vector = sup_vec; %list of suppliers we're building
supplier_matrix = sup_mat; %Matrix describing the suppliers available to
                            %each firm
matrix = mat;   %matrix where we're storing the final product

%How many firms are we working with?
num_firms = size(matrix,2);

%RETURN CONDITION

%If we're on the last firm
if firm == num_firms

    %Print the list of supplier's we've created on this round
    %supplier_vector

    %What's the next open spot in the matrix?
    next_spot = find(matrix(:,1)==0,1,'first');

    %Store it in the matrix
    %Only make changes to this matrix if you're on the last firm and have
    %completed a list of suppliers
    matrix(next_spot,:) = supplier_vector;

    %Return

else
%Otherwise, for each potential supplier of the next firm, call the function


    %What are the supplier options for the next firm?
    firm_suppliers = nonzeros(supplier_matrix(firm+1,:));
    %How many?
    n = length(firm_suppliers);

    %For each such supplier, add him to the vector of suppliers and call the
    %function with it.
    for i = 1:n
        %create a temp vector so you don't ruin it
        temp_vec = supplier_vector;
        %put potential supplier i in the supplier vector
        temp_vec(firm+1) = firm_suppliers(i);

        %call the function with the new working vector for the next firm
        matrix = rec_matrix(firm+1,temp_vec,supplier_matrix,matrix);

    end

end
```

```
%This function determines which of the potential equilbrium networks are
%pairwise stable equilbria.

%It takes as input:
%   - The list of Potential Equilibrium networks, P
%   - The Payoffs for each firm for each such network, Pay
%   - The possible suppliers for each firm, Sup

%It returns a matrix describing the equilbrium, E
%   -This is a list of all equilibria


function [E] = Pick_Equilibrium_Mult(P,Pay,Sup)

%Get the function values
Pot_Matrix = P; %lists the potential eq networks
Payoffs = Pay;  %lists the payoffs for each firm for each network
Suppliers = Sup; %lists the suppliers for each firm

dim = size(Pot_Matrix);

%Create a boolean indicating whether you found a profitable deviation
found_dev = 0;


%How many options for eq network are there?
num_options = dim(1);

%How many firms are there?
num_firms = dim(2);

%initialize E as a matrix of zeros num_options by num_firms
E = zeros(num_options, num_firms);
%create a counter to store where the next available row is.
storage_counter = 1;

%create vectors to hold the current and other eq network options
current = zeros(num_firms,1);
other = zeros(num_firms,1);

%create vectors to hold the payoffs for the two networks
current_payoffs = zeros(num_firms,1);
other_payoffs = zeros(num_firms,1);

%Starting from the first option, check if each matrix is pairwise stable
o = 1;
while o <= num_options
    %Get the current network
    current = Pot_Matrix(o,:);

    %Get the payoffs for the current network
    current_payoffs = Payoffs(o,:);

    %set the indicator to 0
    found_dev = 0;
```

```
    %For each firm, check to see if there is a deviation they would take
    f = 1;
    while f <= num_firms
        %What other suppliers does this firm have
        sup = nonzeros(Suppliers(f,:));
        num_sup = length(sup);


        %For each supplier that isn't the one being used right now,
        %construct a row describing the alternative matrix
        s = 1;
        while s <= num_sup
            %only if the supplier is different from the current one
            %current(f)
            %other = zeros(1,num_firms);

            if sup(s) ~= current(f)
                %copy current
                other = current;

                %put the alternative supplier option in for firm f in the
                %alternative network
                other(f) = sup(s);

                %find the index of the other vector in the potential
                %network matrix
                other_index = find(ismember(Pot_Matrix,other,'rows'));

                %get the payoffs associated with that network
                other_payoffs = Payoffs(other_index,:);
                Payoffs(other_index,:);
                %Payoffs

                %Are both the current firm and the potential new
                %supplier made better off?
                if other_payoffs(f) > current_payoffs(f) &&
other_payoffs(sup(s)) > current_payoffs(sup(s))
                    %set the indicator to 1, break the supplier loop
                    found_dev = 1;
                    %disp('found dev')
                    break;
                else
                    %otherwise move to the next supplier
                    %disp('did not find dev')
                    temp_s_2 = s;
                    s = temp_s_2 + 1;
                end
            else
                %If it IS the current supplier, just go to the next
                %supplier
               temp_s = s;
               s = temp_s + 1;
            end
        end
```

```
        %If you found a deviation, break the firm loop, you don't need to
keep
        %looking
        if found_dev == 1
            break
        else
            %otherwise go on to the next firm
            temp_f = f;
            f = temp_f + 1;
        end
    end

    %If you found a deviation, increase o becuase the one you tried wasn't
    %an eq.
if found_dev == 1
    temp_o = o;
    o = temp_o + 1;
 %Otherwise you didn't find a deviation and this is a pairwise equilbirum
   %     set the next row of E to be current, incremenet the storage
   %     counter and increment o.
else
    E(storage_counter,:) = current;
    %increment storage counter
    temp_count = storage_counter;
    storage_counter = temp_count + 1;
    %increment o
    temp_o_2 = o;
    o = temp_o_2 + 1;
end

end

%trim E
%find the index of the first 0
zero_index = find(~E(:,1),1);
%trim E from the zero_index to num_options
if zero_index > 1
    E(zero_index:num_options,:) = [];
end
end
```

```
%This function returns the coalition-proof equilibrium network (if one
%exists and a list of -1's if one does not). It takes as input:
%   - The list of possible equilibrium matrices
%   - The list of associated payoffs
%   - The sup matrix that lists the possible suppliers for each firm

%It returns a network, e, that lists the supplier of each firm (or -1's)

function [E] = Find_CP_Eq_Mult(Pot_Mat, Pay, Sup)

%get the function variables
Pot = Pot_Mat; %The list of possible eq matrices
Payoffs = Pay; %The list of payoffs associated with each such matrix
Suppliers = Sup; %Lists the possible suppliers for each firm

%get the dimensions of the pot matrix
pot_dim = size(Pot);
num_opt = pot_dim(1);
num_firms = pot_dim(2);

%Initialize E as a list of zero vectors
%allowing for every network to be CP
%trim at the end
E = zeros(num_opt,num_firms);
%create a storage counter
storage_counter = 1;

%create a counter to count the matrices visited
m = 1;
%while there are still matrices to check
%EQ MATRIX LOOP
while m <= num_opt

    %for each matrix, check if it is coalition proof
    %   --> check each coaliton size
    %get the current possible eq network we're trying
    poss_eq = Pot(m,:);

    %COALITION SIZE LOOP
    %check each coalition size
    %if any of them have a profitable deviation, set coaliton_flag = 1 and
found_dev = 1 and end
    coalition_size = num_firms;
    coalition_flag = 0;
    found_dev = 0;
    while coalition_flag == 0
        %check each size
        %check the coalitions of the current coalition_size
        %coalition_size
        found_dev = check_coalition(poss_eq, m, Pot, Payoffs, Suppliers,
coalition_size);
        %coalition_size

        %if there was a profitable deviation, give up on this network
        if found_dev == 1
```

```
            coalition_flag = 1;
        else
            %decrease the coalition size
            %if it's 1, set the flag to 1 to to end the loop
            if coalition_size == 1
                coalition_flag = 1;
            else
                temp_co_size = coalition_size;
                coalition_size = temp_co_size - 1;
                %coalition_size
            end
        end
    %END COALITION CHECKING LOOP
    end

    %if found_dev = 1, you found a deviation and need to increase the m
    %counter to go to the next network
    if found_dev == 1
        %go to next poss eq network
        temp_m = m;
        m = temp_m + 1;
    else
        %if you didn't find a deviation that means you got through all the
        %coalition sizes and no profitable deviations
        %   --> this is a coalition proof network
        %store it in E
        E(storage_counter,:) = poss_eq;
        %increment the storage vector
        temp_counter = storage_counter;
        storage_counter = temp_counter + 1;

        %increment m
        temp_m = m;
        m = temp_m + 1;
    end
    %If you never find a deviation, you never store anything, and E is just
    %a thing of zeros

%END OF MATRIX LOOP
end

%trim E
%find the first 0
zero_index = find(~E(:,1),1);

%trim E from the zero_index to num_options
if zero_index > 1
    E(zero_index:num_opt,:) = [];
end

end
```