

# A Model-based Multi-agent Framework to Enable an Agile Response to Supply Chain Disruptions\*

Mingjie Bi<sup>1</sup>, Gongyu Chen<sup>2</sup>, Dawn M. Tilbury<sup>1,3</sup>, Siqian Shen<sup>2</sup> and Kira Barton<sup>1,3</sup>

**Abstract**—Due to the COVID-19 pandemic, the global supply chain is disrupted at an unprecedented scale under uncertain and unknown trends of labor shortage, high material prices, and changing travel or trade regulations. To stay competitive, enterprises desire agile and dynamic response strategies to quickly react to disruptions and recover supply-chain functions. Although both centralized and multi-agent approaches have been studied, their implementation requires prior knowledge of disruptions and agent-rule-based reasoning. In this paper, we introduce a model-based multi-agent framework that enables agent coordination and dynamic agent decision-making to respond to supply chain disruptions in an agile and effective manner. Through a small-scale simulated case study, we showcase the feasibility of the proposed approach under several disruption scenarios that affect a supply chain network differently, and analyze performance trade-offs between the proposed distributed and centralized methods.

## I. INTRODUCTION

The COVID-19 pandemic has tremendously disrupted the global supply chain in different aspects, such as customer demand, material supplies, manufacturing capabilities, labor shortage, etc. [1]. It also highlights the complexity of modern global supply chains [2]. Supply chain entities, such as customers, suppliers, and manufacturers, suffer from slow knowledge propagation of supply-demand mismatch amid lock-downs and other disruptions. Therefore, there is an urgent need for a dynamic response strategy that enables smart, agile, and resilient supply chains under unexpected disruptions [3].

In many supply chain networks, centralized models have been widely used to provide optimal solutions based on specific objectives (e.g., product flow cost), but have limited ability to adjust solutions due to disruptions in an agile manner [2]. As the complexity and scale of supply chains increase, it becomes more difficult to effectively manage supply chain networks under disruptions using centralized methods [2], [4]. To improve the agility and effectiveness of supply chain networks, researchers have proposed distributed control strategies, where multiple entities in the system make decisions via communication and collaboration [5].

\*This work was funded in part by The United States National Science Foundation (NSF) grant #CMMI-2034974.

<sup>1</sup>Mingjie Bi is with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109, USA, [mingjieb@umich.edu](mailto:mingjieb@umich.edu)

<sup>2</sup>Gongyu Chen and Siqian Shen are with the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA, [{chongongyu, siqian}@umich.edu](mailto:{chongongyu, siqian}@umich.edu)

<sup>1,3</sup>Dawn M. Tilbury and Kira Barton are with the Robotics Institute and the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA, [{tilbury, bartonk1}@umich.edu](mailto:{tilbury, bartonk1}@umich.edu)

Multi-agent control is a distributed method that enables intelligent decision-making in supply chains [4]. Each autonomous agent in a supply chain network either represents a physical entity, such as a supplier, or is responsible for a function, such as demand forecasting. Various agents communicate, make decisions based on their knowledge, and collaboratively solve supply chain problems [4], [6].

In this work, we focus on the problem of reacting to disruptions and recovering supply chain functions: given a supply chain network, existing product flows, and a disruption, we aim to identify a new product flow plan that satisfies the demand and minimizes penalties in terms of costs, time, etc. In the existing literature, most agent-based disruption reaction strategies are based on pre-defined disruption scenarios and reactive actions via a stochastic programming model [7], a Petri Nets model [8], or a case-based disruption reaction database [9]. In many examples, the disruption reaction performance of these methods is limited to a set of pre-defined scenarios, and it is difficult and even impossible to cope with unexpected disruptions outside of the predefined set.

To address this limitation, agents need to be equipped with model-based knowledge to make decisions dynamically. However, existing multi-agent approaches focus on either system-level architectures or rule-based agents [5]. The works [1], [10], [11] provide general descriptions of agent attributes and functions at a conceptual level. References [12]–[14] use rule-based reasoning logic to guide agent decisions, while [2], [13], [15] introduce case-based agents that provide pre-planned decision-making and coordinated behaviors for the agents. In these methods, relying on a rule-based strategy limits the ability of the agents to readily adapt to unexpected supply chain disruptions. Additionally, it is difficult to scale this rule-based approach to larger and more complex systems, with the addition of more rules reducing the flexibility of agent behaviors. Lastly, references [16], [17] introduce model-based agents for manufacturing systems; however, such approaches to define model-based agents for supply chain networks are lacking.

The main contributions of this paper include: (1) deriving a modeling structure within the knowledge base of supply-chain agents, (2) development of a dynamic agent communication infrastructure to respond to unknown disruptions, and (3) evaluation of potential trade-offs between centralized and the proposed distributed multi-agent methods through a simulated case study with various types of disruptive events.

The rest of the paper is organized as follows. In Section II, we introduce a centralized optimization model to optimize

TABLE I  
A NOTATION SUMMARY FOR THE CENTRALIZED MODEL IN (2)

Known Parameters	
$d_{ik}$	demand of product $k$ at vertex $i$ ; by convention, $d_{ik} < 0$ .
$f_{ij}$	fixed transportation cost from vertex $i$ to $j$ .
$c_{ijk}$	unit transportation cost of flowing product $k$ from vertex $i$ to $j$ .
$q_{ij}$	mixed-flow capacity of edge $(i, j)$ .
$\bar{p}_i$	mixed-product production capacity available at vertex $i$ .
$e_{ik}$	production cost per unit product $k$ at vertex $i$ .
$r_{kk'}$	conversion rate from product $k$ to $k'$ , i.e. the unit of product $k$ consumed to produce one unit successor product $k'$ .
$\phi_i$	fixed cost of opening the production line of vertex $i$ .
$I_{ik}^0$	initial inventory of product $k$ at vertex $i$ at start of a time period.
$h_{ik}$	unit holding cost of product $k$ at vertex $i$ .
$\rho_{ik}^d$	penalty per unit of unsatisfactory demand $d_{ik}$ .
Decision Variables	
$y_{ijk}$	units of product $k$ flowing from vertex $i$ to $j$ .
$\beta_{ij}$	binary variable, equal to 1 if edge $(i, j) \in E$ is used to transport products, and 0 otherwise.
$x_{ik}$	actual satisfied demand of product $k$ at vertex $i$ .
$p_{ik}$	units of product $k$ produced at vertex $i$ .
$\zeta_i$	binary variable, equal to 1 if the production/assembly line opens at vertex $i$ , and 0 otherwise.
$I_{ik}$	inventory of product $k$ at vertex $i$ by the end of time period.
$\Delta_{ik}^d$	units of unsatisfied demand of product $k$ at vertex $i$ .

flow in supply chains. In Section III, we present the proposed distributed multi-agent framework. In Section IV, we describe an agent communication infrastructure for disruption response. Section V provides a case study for approach comparisons, with concluding remarks given in Section VI.

## II. CENTRALIZED MODEL

In this section we introduce a standardized, centralized model to serve as a comparison to the proposed multi-agent framework. In this work, we modify a traditional network-flow model that only considers product flow [18] and extend it to a generic mixed-integer linear programming (MILP) model that considers production, inventory, and transportation planning in response to disruptions for a multi-echelon, multi-product supply chain network.

### A. A centralized model for supply chain management

Consider a supply chain network  $G(V, E)$  with  $V$  being the set of vertices and  $E$  being the set of edges. The vertices represent the supply chain entities, such as suppliers, customers, etc., and the edges represent possible material or product flows between these entities. We denote  $K$  as the set of all product types within the system. Table I summarizes the parameters and decision variables used in this paper. We compute the total cost of operating the supply chain in Eq. (1), where  $y, x, I, p, \beta, \zeta, \Delta$  indicate the vector forms of the corresponding decision variables shown in Table I:

$$\mathcal{J}(y, x, I, p, \beta, \zeta, \Delta) \quad (1a)$$

$$= \sum_{(i,j) \in E, k \in K} c_{ijk} y_{ijk} + \sum_{i \in V, k \in K} h_{ik} I_{ik} + \sum_{i \in V, k \in K} e_{ik} p_{ik} \quad (1b)$$

$$+ \sum_{(i,j) \in E} f_{ij} \beta_{ij} + \sum_{i \in V} \phi_i \zeta_i + \sum_{i \in V, k \in K} \rho_{ik}^d \Delta_{ik}^d, \quad (1c)$$

where (1b) are costs incurred by transporting, manufacturing, and holding products, respectively; (1c) are fixed costs related to setting up transportation and manufacturing, as well as penalty costs of unsatisfied demand. We present the centralized model to compute optimal operations in (2):

$$\min_{y, x, I, p, \beta, \zeta, \Delta} \mathcal{J} \quad (2a)$$

$$\text{s.t.} \quad \sum_{j: (i,j) \in E} y_{ijk} - \sum_{j: (j,i) \in E} y_{jik} + \sum_{k' \in K} r_{kk'} p_{ik'} \\ - p_{ik} = x_{ik} + I_{ik}^0 - I_{ik}, \quad \forall i \in V, k \in K \quad (2b)$$

$$\sum_{k \in K} y_{ijk} \leq q_{ij} \beta_{ij}, \quad \forall (i, j) \in E \quad (2c)$$

$$\sum_{k \in K} p_{ik} \leq \bar{p}_i \zeta_i, \quad \forall i \in V \quad (2d)$$

$$\Delta_{ik}^d \geq x_{ik} - d_{ik}, \quad \forall i \in V, k \in K \quad (2e)$$

$$y_{ijk}, x_{ik}, I_{ik}, \Delta_{ik}^d \geq 0, \quad \zeta_i, \beta_{ij} \in \{0, 1\}, \\ \forall i \in V, (i, j) \in E, k \in K, \quad (2f)$$

where constraint in (2b) requires the flow balance of each product at each vertex; constraint in (2c) and (2d) limits the flow on each edge and production at each vertex by its given capacity; constraint in (2e) computes the unsatisfied demands of each product at each vertex; and constraint in (2f) specifies the domains of the decision variables.

### B. Disruption response using the centralized model

To analyze the performance of the proposed multi-agent framework, we first incorporate the effects of various disruptions into Model (2). For example, suppose that  $E_d \subset E$  is identified to be a set of edges that become unavailable due to disruptions. Correspondingly, we add the following constraints to Model (2):

$$y_{ijk} = 0, \quad \forall (i, j) \in E_d, k \in K. \quad (3)$$

Once the disruption is identified, a centralized decision-maker will re-run the centralized model with updated network structures, parameters, and constraints to determine the re-optimized decisions. The objective cost in (1) can be extended by adding costs related to the disruption and response, such as communication cost, network flow change, etc. We define  $\Delta_{ij}^E = |\beta_{ij} - \beta_{ij}^0|$  and  $\Delta_i^V = |\zeta_i - \zeta_i^0|$  as the usage status change on edge  $(i, j)$  and vertex  $i$ , where  $\beta_{ij}^0$  and  $\zeta_i^0$  are the binary indicators denoting whether the edge  $(i, j)$  and vertex  $i$  are used in the network before the disruption. We also incorporate  $\rho_{ij}^E$  and  $\rho_i^V$  as unit cost parameters for penalizing changes in the edges and vertices.

Given the example disruption, we update Model (2) and re-optimize the operations using the following formulation:

$$\min \quad \mathcal{J} + \sum_{(i,j) \in E} \rho_{ij}^E \Delta_{ij}^E + \sum_{i \in V} \rho_i^V \Delta_i^V \quad (4a)$$

$$\text{s.t.} \quad (2b) - (2f), (3) \quad (4b)$$

$$\Delta_{ij}^E \geq \pm (\beta_{ij} - \beta_{ij}^0), \quad \forall (i, j) \in E, k \in K \quad (4c)$$

$$\Delta_i^V \geq \pm (\zeta_i - \zeta_i^0), \quad \forall i \in V, k \in K \quad (4d)$$

$$\Delta_{ij}^E, \Delta_i^V \in \{0, 1\}, \quad \forall i \in V, (i, j) \in E. \quad (4e)$$

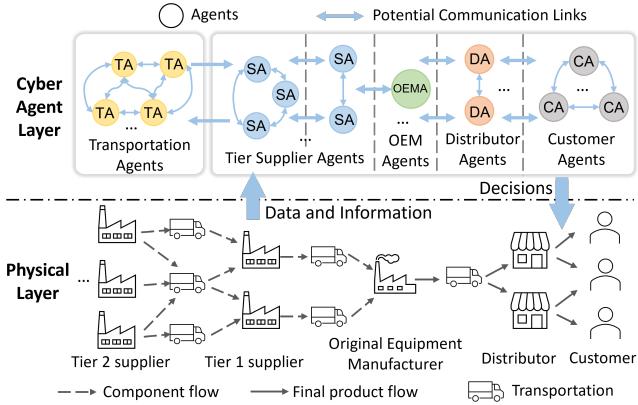


Fig. 1. Agents in an example supply network. Each agent represents an associated physical entity. Blue arrows represent potential communication.

The added terms in (4a) represent the total penalty costs of changing the used edges and vertices. In practice, these penalties can represent liquidated damage charges in contracts with local suppliers or transportation companies, or signing fees for new contracts. Constraints (4c) and (4d) linearize the definitions of  $\Delta_{ij}^E$  and  $\Delta_i^V$ , and constraint (4e) specifies their binary value domains.

The generic MILP model is flexible enough to capture different disruptions and incorporate disruption responses by modifying objectives and constraints. With the visibility of all entities and their status after the disruptions, this model can provide a globally optimal solution. In this work, this centralized model serves as a benchmark for evaluating the performance of the proposed multi-agent framework.

### III. MULTI-AGENT FRAMEWORK

In this section, we describe the agent network consisting of different types of agents. We also provide a detailed design of the model-based agent knowledge base, including capabilities, environments, and state dynamics.

#### A. Multi-agent supply chain network

Various types of agents have been developed in the existing literature for locally operating a supply chain [19], [20]. We summarize the most commonly used agent types and use the following types to build a supply chain agent network:

- Customer agent - orders products based on demand
- Distributor agent - receives the customer's order and sells products to customers
- Original equipment manufacturer (OEM) agent - assembles products using components from tier suppliers
- Tier supplier agent - supplies components to the OEMs
- Transportation agent - delivers components and products between physical entities

Each agent represents a physical entity and will be initialized with the physical entity characteristics, such as capabilities, goals, and risk tolerance. The agent obtains data from the physical world and is able to command changes to its corresponding physical entity, as shown in Fig.1.

TABLE II  
EXAMPLES OF MAPPING FUNCTIONS TO DESCRIBE CAPABILITIES

Agent capability	Example mapping function
DA: inventory	$Iv_{i,t} \rightarrow \mathbb{R}_+$ : mapping from a product type to the cost to hold unit product in inventory.
OEMA: production	$Pd_{i,t} \rightarrow K \times \mathbb{R}_+$ : mapping from a product type to the type and amount of needed components to produce unit product.
TA: transportation	$E \rightarrow \mathbb{R}_+$ : mapping from an edge to the capacity limit of transporting products.

We generate the agent network based on the physical network graph  $G(V, E)$  in Section II. Here, both the vertices  $V$  and edges  $E$  are a set  $N$  of nodes, which are connected by a set  $L_t$  of communication links. During each time period  $t$ , we denote the agent network as  $G_t^a(N, L_t)$ , where  $N = V \cup E$ . Since  $G$  is assumed to include all potential physical entities,  $N$  is static while  $L_t$  varies over time. For a specific agent  $A_i$ ,  $G_{i,t}^a(N_i, L_{i,t})$  represents its local communication network at time period  $t$ , where  $N_i \subseteq N$  is a set of nodes that  $A_i$  can communicate with via links in  $L_{i,t}$ . In this work, each time index  $t$  represents one time period where no disruptions occur and the network remains constant, such as flow rates. Once a disruption occurs, the response action requires changes to the network, and the supply chain enters the next time period  $t + 1$ .

#### B. Agent knowledge base

The proposed agent knowledge base consists of models for agent capabilities, environments, and states. These models describe the agent knowledge at time  $t$ .

1) *Capability model*: The capability model describes the behaviors that an agent can perform in the supply chain network. Based on the agent classification in Section III-A, agent capabilities are classified using three types: production, inventory, transportation. The capability model consists of capability knowledge and several associated mapping functions. We define the capability knowledge of agent  $A_i$  as a set  $M_{i,t} = \{Pd_{i,t}, Iv_{i,t}, Tp_{i,t}\}$ , where  $Pd_{i,t}, Iv_{i,t}, Tp_{i,t} \subseteq K$  define production, inventory, and transportation sets with which agent  $A_i$  can operate. For example, a product type  $k \in Pd_{i,t}$  indicates that  $A_i$  can produce product  $k$  at time  $t$ . Along with these high-level representations of the capabilities, several mapping functions are used to describe the characteristics of a capability, such as cost and capacity. Table II presents some example mapping functions.

The capability model is built when the agent is initialized based on its associated physical entity. This information is dynamically updated as the supply chain environment changes, such as cost increases, production line changes, etc.

2) *Environment model*: An agent's knowledge of other agents in the network is encapsulated in the environment model. As defined in Section III-A,  $G_{i,t}^a(N_i, L_{i,t})$  represents the communication network of agent  $A_i$  at time period  $t$ . Note that  $N_i \setminus A_i$  is the set of agents that  $A_i$  can communicate with. In the environment model, these agents are grouped

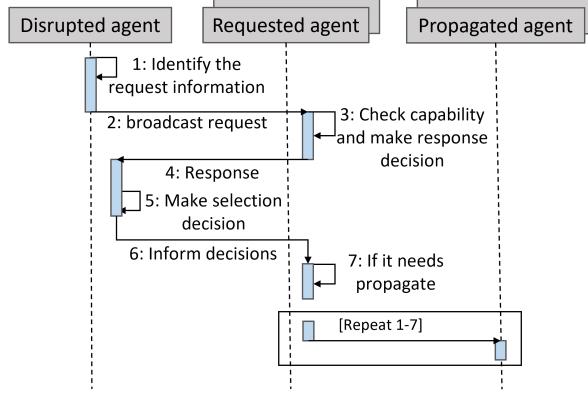


Fig. 2. Agent communication scheme to adapt to disruption.

into several sets based on their relationship with agent  $A_i$ . In this way, an agent is capable of identifying the subset of agents it needs to communicate and exchange information within the network. These relationships are modeled as the following mapping functions:

- $U_{i,t} : K \rightarrow 2^N$ : mapping from product types to upstream agents from which  $A_i$  can obtain the products
- $D_{i,t} : K \rightarrow 2^N$ : mapping from product types to downstream agents to which  $A_i$  can provide the products
- $T_{i,t} : K \rightarrow 2^N$ : mapping from a product type to the transportation agents that can reach agent  $A_i$ .
- $S_{i,t} : M_{i,t} \rightarrow 2^N$ : mapping from a capability in agent  $A_i$  to the agents that have the same capability

The mapping functions  $U_{i,t}$ ,  $D_{i,t}$ , and  $T_{i,t}$  identify the agents that can execute a physical product flow (e.g., obtaining products from agent  $A_j$ ) with agent  $A_i$ . The information about the flow (e.g., product type and amount) is stored in the environment model via other mapping functions associated with the agent sets. For example, a function  $U_{i,t}(K) \rightarrow K \times \mathbb{R}_+$  maps the upstream agent to the product type and amount of product flow with  $A_i$ . The mapping  $S_{i,t}$  identifies agents that perform the same behavior as agent  $A_i$  for replanning purposes.

The environment model is generated when the agent is initialized based on its associated physical entity. The agent sets and their associated information are dynamically updated if the supply chain environment changes, such as production capabilities of a given agent change. Note that changes may occur during a time period  $t$  if the changes only occur in agent knowledge without leading to changes in the network.

3) *State model:* The agent dynamics are described as a flow balance of varying input and output products. The state model describes the dynamics in terms of flow, production, and inventory based on the flow balance equation:

$$I_{i,t+1} = I_{i,t} + u_{i,t} - z_{i,t} + h_{i,t}(I_{i,t}, u_{i,t}). \quad (5)$$

Note that each variable in the state model is a vector indexed by product type, denoted by  $[\cdot]_{k \in K}$ , where state vector  $I_{i,t} = [I_{ik,t}]_{k \in Iv_{i,t}}$  represents the amount of products held in the agent at time  $t$ ; input vector  $u_{i,t} =$

$\left[ \sum_{j \in U_{i,t}(k)} y_{jik,t} \right]_{k \in Iv_{i,t}}$  represents product flows into the agent from the upstream nodes  $U_{i,t}(K)$ ; output vector  $z_{i,t} = \left[ \sum_{j \in D_{i,t}(k)} y_{ijk,t} \right]_{k \in Iv_{i,t}}$  represents the product flows out of the agent to the downstream nodes  $D_{i,t}(K)$ ; and production function  $h_{i,t}(I_{i,t}, u_{i,t})$  defines the amount of components used to generate new products if the agent has production capabilities. Note that the variables in (5) are bounded by agent specific limits. For example,  $I_{i,t}$  is limited by the inventory capacity of  $A_i$ , while  $u_{i,t}$  and  $z_{i,t}$  are limited by transportation capacities.

#### IV. AGENT COMMUNICATION FOR DISRUPTION RESPONSE

In this section, we describe the proposed agent communication scheme presented in Fig. 2. We assume that agents can communicate no matter how the disruptions affect the physical entities, and a communication link could be established or removed based on the disruption. The agent communication for disruption response is assumed to be completed within the time period  $t$  when the disruption occurs. Future work will relax these assumptions. For simplicity, the time index is neglected in this section.

##### A. Agent communication

1) *Request:* In this work, we assume that agents are able to detect disruptions that occur in their associated physical entities and identify the effects of the disruption by periodically obtaining data and information from the physical entities. Disruptions can affect the flow ( $u, z$ ), inventory ( $I$ ), and/or production capabilities ( $h(I, u)$ ) of an agent. Therefore, when a disruption occurs, the disrupted agent ( $A_d$ ) must determine: 1) the product flows needed to recover the flow balance (e.g. satisfy downstream demand); 2) the agents with whom it should communicate for replanning. Note that if multiple disruptions occur or multiple agents are disrupted, each agent will send their requests separately; the coupling relationships between disrupted agents is outside the scope of this work.

Consider a case when an OEM loses all of the inventory and production equipment due to an unexpected catastrophic event. The request,  $Req = (y_d)$ , is defined as  $y_d = [y_{djk}]_{j \in D_d(k), k \in Iv_d}$ . In this request,  $y_d$  is a vector indexed by downstream agents. Each element  $y_{djk}$  represents the flow of product  $k$  from the disrupted OEM  $d$  to its downstream agent  $A_j$ . Note that the request could include other information related to the flow, such as budget, delivery due date, etc., as additional requirements. This type of request from OEM  $d$  is only sent to the agents that have the same production capabilities denoted by  $(S_d(M_d))$  to find alternative OEM agents that can provide the required products to the distributors.

2) *Response:* Agents that receive the request will then check their knowledge base and determine their maximum ability to recover the product flows. In the example of an OEM being unavailable, a response will include available product flow  $\bar{y}_{ijk}$  of product  $k$  that the requested agent  $A_i$  can provide to  $A_j$ . Agent  $A_i$ 's response to OEM  $d$  is  $\bar{y}_i = [\bar{y}_{ijk}]_{j \in D_d(k), k \in Iv_d}$ . Assuming flows are primarily limited

by the transportation capacity, the requested agent  $A_i$  can determine their response by solving the optimization problem in (6):

$$\min_{\bar{y}_i} \|\bar{y}_i - y_d\|_1 \quad (6a)$$

$$\text{s.t. } \sum_{k \in Iv_d} \bar{y}_{ijk} \leq q_{ij}, \forall j \in D_d(k), k \in Iv_d, \quad (6b)$$

where (6a) minimizes the difference between the request and response. Note that multiple agents can be used to provide the needed flow, and the objective and constraints can be extended if considering cost, production capacity, etc.

3) *Inform*: After receiving responses from all of the requested agents, the disrupted agent  $A_d$  determines the alternative flows  $y_r$  that can replace  $y_d$ :  $y_r = [\sum_{i \in P} y_{ijk}]_{j \in D_d(k), k \in Iv_d}$ , where  $i$  represents the index of the requested agent  $A_i$  and  $P$  is the set of requested agents. Note that  $y_r$  is a vector indexed by downstream agents. Assuming no other constraints, the disrupted agent can determine the alternative flow  $y_r$  by solving the optimization problem in (7):

$$\min_{y_r} C(y_r) + \|y_r - y_d\|_1 + \|y_r - y_0\|_1 \quad (7a)$$

$$\text{s.t. } y_{ijk} \leq \bar{y}_{ijk}, \forall k \in Iv_d, j \in D_d(k) \quad (7b)$$

$$y_r \leq y_d, \quad (7c)$$

where  $C(y_r)$  represents the total cost of choosing  $y_r$  and  $y_0$  represents the product flow vector before the disruption with the same indices as  $y_r$ . The objective (7a) is to minimize the cost, the unsatisfied total amount of flows, and the flow changes. The two constraints (7b) and (7c) denote that alternative flows cannot exceed their limitations determined by (6) and the request. Note that the objective can be extended to consider delivery time, product quality, etc. Other constraints, such as budget, delivery due date, etc., can also be added.

4) *Propagation*: After an agent is selected and informed, it will check whether additional products, such as components from an upstream agent, are needed to provide the new product flow. If so, the selected agent will send a request to the upstream agents. The communication follows the process detailed above, as shown in Fig. 2. It is possible that a solution may not be found due to the limited information of the agents. In such cases, the disrupted agent will request the centralized model to generate a new product flow plan using a global perspective.

### B. Network update

The agent knowledge base and communication network are updated dynamically based on changes within the network. For example, new product flow lines must be updated within the relevant agent knowledge base. For the capability model, the agents update the capability knowledge set  $M_i$  if a capability change occurs, such as changing the production line to produce a new product. The associated mapping functions are also updated if the characteristics of the capabilities change, such as increased production costs. For

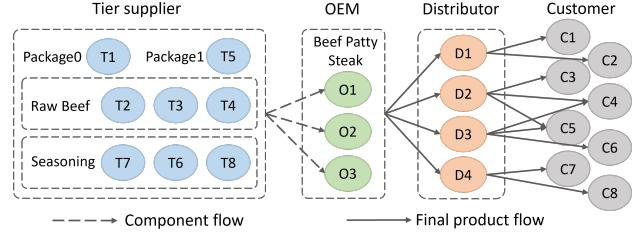


Fig. 3. Supply chain network used for case study. Arrows represent component and product flow via transportation.

the environment model, communication groups are updated if the agents obtain new knowledge of upstream agents, downstream agents, and agents that have the same capability. The state model is updated based on the new flow balance of the agent.

## V. CASE STUDY

To test the feasibility and performance of the proposed framework, we implemented the multi-agent framework and agent communication infrastructure in a simulated supply chain network. In this section, we describe the set-up and simulation results of the supply chain case study.

### A. Case study set-up

The supply chain network we use in this work is an adapted version from reference [21]. The simulated network contains 8 tier suppliers, 3 OEMs, 4 distributors, and 8 customers that are connected via distinct transportation, as shown in Fig. 3. The annotations represent the product types that the tier suppliers and OEMs can produce and hold in inventory. For example, the annotation for OEM indicates that all OEMs can produce and hold Beef Patties and Steak. The distributors have inventory capabilities for both the Beef Patties and Steak. The customers provide the demand for the Beef Patties or Steak. Each unit of Beef Patty needs 1 unit of Raw Beef, Seasoning, and Package0, and each unit of Steak needs 1 unit of Raw Beef, Seasoning, and Package1. Production and transportation in the network correspond to a cost and capacity constraint. Each entity in the network corresponds to an agent that follow the framework outlined in this work and illustrated in Fig. 3.

### B. Case study results

In this case study, the supply chain starts with an existing product flow plan determine by solving the centralized model (2). Based on the network topology, we chose three disruptions to evaluate the performance differences between the more general centralized method and our proposed distributed method: 1) C5 receives a new demand request; 2) T4 becomes unavailable; and 3) O1 becomes unavailable. Note that C5 is a downstream agent with only inflow, T4 is an upstream agent with only outflow, and O1 is an agent with both inflow and outflow. The performance of the framework based on the different disruption scenarios is evaluated using the following metrics:

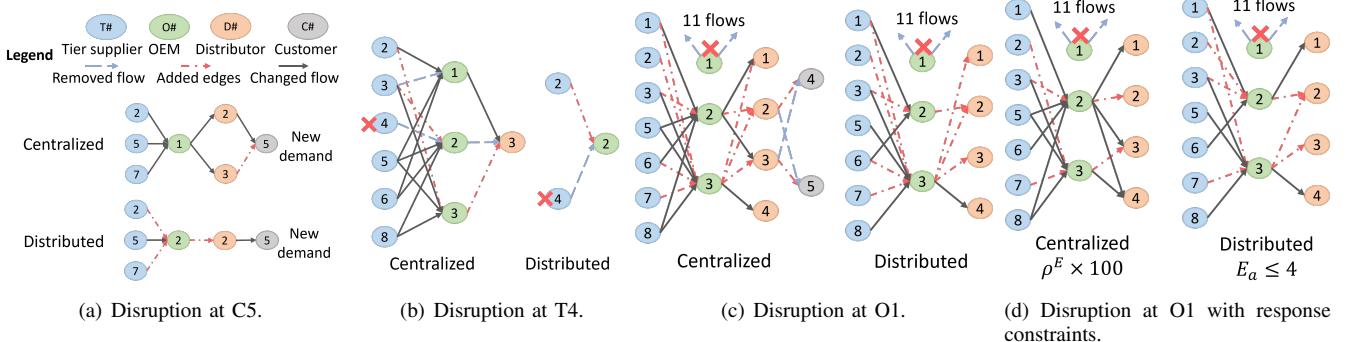


Fig. 4. Changes to product flow in the network by solving the centralized and distributed models to respond to different disruptions.

TABLE III

THE PERFORMANCE OF CENTRALIZED AND PROPOSED DISTRIBUTED METHOD UNDER DIFFERENT DISRUPTIONS

Scenarios		Metrics				
Disrupted agent	Methods	$C_f$	$C_p$	$E_a$	$F_c$	$C_e$
C5	Centralized	+1599.54	+5691	1	6	54
	Distributed	+1722.39	+5657.89	3	2	29
T4	Centralized	-1742.72	+5889.56	3	10	56
	Distributed	-1361.46	+6039.89	1	0	6
O1	Centralized	+1981.49	-596.31	11	10	61
	Distributed	+2215.31	-246.9	9	6	46
	Centr. ( $\rho^E \times 100$ )	+2154.78	-625.31	5	11	61
	Centr. ( $E_a \leq 4$ )	unsatisfying demand				
	Distr. ( $E_a \leq 4$ )	+2309.86	-201.91	7	7	46

- Flow cost change ( $C_f$ ): change in the costs associated with a given flow
- Production cost change ( $C_p$ ): change in the production costs
- Added edges ( $E_a$ ): number of edges that are added to address the disruption
- Changed flows ( $F_c$ ): number of flow channels that are changed in terms of type and/or amount of product
- Communication effort ( $C_e$ ): number of communication exchanges used to derive a response to the disruptions

The calculations of  $C_f$ ,  $C_p$ ,  $E_a$  and  $F_c$  are straightforward by definition. The communications effort  $C_e$  in the centralized method includes the request for re-running the model, the requests to and responses from all the nodes in the supply chain to collect information, and the notifications to the nodes whose flow and/or production plan need to change. In the distributed method,  $C_e$  includes all agent requests, responses, and inform messages, as defined in Section IV-A.

Table III shows the performance of the centralized and proposed distributed methods under different disruptions. Figure 4 represents the changes in product flow within the network. In the scenario with a disruption at C5, the centralized method re-distributes the product flow to avoid adding edges, which leads to more overall changes to the existing flow edges, as shown in Fig. 4(a). The proposed distributed

method minimizes undesirable local changes without increasing the cost drastically. However, the distributed approach adds more new edges since the local agents cannot provide knowledge of the added edges of other agents. For example, in the simulation, agent D2 does not know that O2 needs new edges from T2 and T7 to satisfy its request. The results also show that new demands from customers require minimal flow changes from downstream to upstream.

The scenario of a disruption at T4 illustrates the benefit of using the proposed multi-agent method to respond to disruptions at an upstream node. Although the flow and production costs are higher than the centralized method, the agent communication only occurs within the agents that have the same capability as T4 and related OEMs, which results in a faster response. Fig. 4(b) shows the distributed method only results in a small change in the network.

In the scenario of a disruption at O1, both the centralized and distributed methods lead to lots of edge and flow changes with a large communication effort, as shown in Fig. 4(c). Since O1 has a combined 11 inflow and outflow edges before the disruption, the distributed method cannot recover the complete flow demand within a local communication environment. In practice, adding new flow edges can lead to high costs, thus we increase the penalty  $\rho^E$  on the addition of new edges and add a constraint to  $E_a$  to check the performance. Figure 4(d) shows that the centralized method provides a solution with a smaller  $E_a$  when a larger  $\rho^E$  is applied, while the cost becomes higher. With the hard constraint on  $E_a$ , the centralized model fails to provide a solution that can satisfy the customer demand. However, in the distributed method, since agents only have their local view, the value of  $E_a$  only considers their connected edges. Thus, the distributed method can provide a feasible solution, as shown in Fig. 4(d).

Overall, both the centralized and distributed methods lead to higher costs to satisfy the demand. The centralized method provides lower-cost solutions since it has a global view of the network. However, it often leads to more production and flow changes with larger communication efforts. The distributed method responds to the disruptions with less communication and maintains the original flow as much as possible, especially under a small disruption. The results

from network changes also show that the proposed multi-agent method performs better when the disruption occurs in an upstream node or a node with few initial flows. An enterprise can make a decision depending on whether they aim for a quick response, a low-cost solution, or a minimal-change solution.

### C. Insight from case studies

The case study presented in this section showcases the feasibility of using a model-based multi-agent framework to drive the behavior of the entities in the supply chain network. In the case study, both centralized and distributed models seek to minimize the number of edge changes for decision-making. However, the objective functions can include additional information such as desired inventory level or communication costs.

Additionally, we initialize the local view of the agents with a subset of the global supply chain network in the case study. In practice, however, an agent may have knowledge of some entities outside of the network or the centralized model. Incorporating this knowledge and designing an agent communication strategy for a new entity joining the network can be studied in future work.

The case study illustrates that disruptions at different nodes lead to different performance outcome. We will investigate nodes' attributes that result in these differences in future work. Future work may also include the analysis of the effects of network topology on response performance, and the design of an optimization framework for selecting response strategies.

## VI. CONCLUSION

Various multi-agent frameworks have been studied to cope with disruptions in supply chain systems. In this paper, we introduce a multi-agent framework where agents are equipped with a model-based knowledge base and dynamic communication network. The proposed agents have the knowledge of their capabilities, environments, and state dynamics. Based on agent knowledge, this paper develops an effective agent communication strategy to recover a new product flow plan without requiring prior knowledge of the potential disruptions. The proposed work can be used to create supply chain models that enable dynamic and agile responses to disruptions and allow for greater agent flexibility and scalability to larger systems when compared to rule-based architectures. Showcased through a simulated case study, the proposed work provides an agile response to supply chain disruptions and maintains the original flow as much as possible. Future work will include generalizing the agent model and communication infrastructure for different disruptions, analyzing how the different disrupted agents and network topology affect the performance, and designing an optimization framework for determining different strategies to respond to disruptions.

## REFERENCES

- [1] T. Rahman, F. Taghikhah, S. K. Paul, N. Shukla, and R. Agarwal, "An agent-based model for supply chain recovery in the wake of the covid-19 pandemic," *Computers & Industrial Engineering*, vol. 158, p. 107401, 2021.
- [2] M. Giannakis and M. Louis, "A multi-agent based system with big data processing for enhanced supply chain agility," *Journal of Enterprise Information Management*, 2016.
- [3] M. Ben-Daya, E. Hassini, and Z. Bahroun, "Internet of things and supply chain management: a literature review," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4719–4742, 2019.
- [4] G. Baryannis, S. Validi, S. Dani, and G. Antoniou, "Supply chain risk management and artificial intelligence: state of the art and future research directions," *International Journal of Production Research*, vol. 57, no. 7, pp. 2179–2202, 2019.
- [5] L. Xu, S. Mak, and A. Brintrup, "Will bots take over the supply chain? revisiting agent-based supply chain automation," *International Journal of Production Economics*, vol. 241, p. 108279, 2021.
- [6] R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, and M. Fischl, "Artificial intelligence in supply chain management: A systematic literature review," *Journal of Business Research*, vol. 122, pp. 502–517, 2021.
- [7] S. Torabi, M. Baghersad, and S. Mansouri, "Resilient supplier selection and order allocation under operational and disruption risks," *Transportation Research Part E: Logistics and Transportation Review*, vol. 79, pp. 22–48, 2015.
- [8] M. F. Blos, R. M. da Silva, and H.-M. Wee, "A framework for designing supply chain disruptions management considering productive systems and carrier viewpoints," *International Journal of Production Research*, vol. 56, no. 15, pp. 5045–5061, 2018.
- [9] M. Giannakis and M. Louis, "A multi-agent based framework for supply chain risk management," *Journal of Purchasing and Supply Management*, vol. 17, no. 1, pp. 23–31, 2011.
- [10] A. Fiedler, "An agent-based negotiation protocol for supply chain finance," *Computers & Industrial Engineering*, vol. 168, p. 108136, 2022.
- [11] M. Hajian Heidary, "Agent-based simulation-optimization model for a bi-objective stochastic multi-period supply chain design problem," *Journal of Industrial Engineering and Management Studies*, vol. 8, no. 2, pp. 175–195, 2022.
- [12] P. Ghadimi, C. Wang, M. K. Lim, and C. Heavey, "Intelligent sustainable supplier selection using multi-agent technology: Theory and application for industry 4.0 supply chains," *Computers & Industrial Engineering*, vol. 127, pp. 588–600, 2019.
- [13] K. Pal and B. Karakostas, "A multi agent-based service framework for supply chain management," *Procedia computer science*, vol. 32, pp. 53–60, 2014.
- [14] G. Wang, T. Wong, and X. Wang, "An ontology based approach to organize multi-agent assisted supply chain negotiations," *Computers & Industrial Engineering*, vol. 65, no. 1, pp. 2–15, 2013.
- [15] N. Shukla and S. Kiridena, "A fuzzy rough sets-based multi-agent analytics framework for dynamic supply chain configuration," *International Journal of Production Research*, vol. 54, no. 23, pp. 6984–6996, 2016.
- [16] I. Kovalenko, D. Tilbury, and K. Barton, "The model-based product agent: A control oriented architecture for intelligent products in multi-agent manufacturing systems," *Control Engineering Practice*, vol. 86, pp. 105–117, 2019.
- [17] M. Bi, I. Kovalenko, D. M. Tilbury, and K. Barton, "Dynamic resource allocation using multi-agent control for manufacturing systems," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 488–494, 2021.
- [18] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows: theory, algorithms, and applications," 1993.
- [19] P. Ghadimi, F. G. Toosi, and C. Heavey, "A multi-agent systems approach for sustainable supplier selection and order allocation in a partnership supply chain," *European Journal of Operational Research*, vol. 269, no. 1, pp. 286–301, 2018.
- [20] R. de Souza Henriques, "Multi-agent system approach applied to a manufacturer's supply chain using global objective function and learning concepts," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1009–1019, 2019.
- [21] S. P. Willems, "Data set—real-world multiechelon supply chains used for inventory optimization," *Manufacturing & service operations management*, vol. 10, no. 1, pp. 19–23, 2008.