Graph Database to Enhance Supply Chain Resilience for Industry 4.0

Young-Chae Hong, Ford Motor Company, USA
https://orcid.org/0000-0001-6330-6101

Jing Chen, Ford Motor Company, USA

ABSTRACT

Supply chain network in the automotive industry has complex, interconnected, multiple-depth relationships. Recently, the volume of supply chain data increases significantly with Industry 4.0. The complex relationships and massive volume of supply chain data can cause visibility and scalability issues in big data analysis and result in less responsive and fragile inventory management. The authors develop a graph data modeling framework to address the computational problem of big supply chain data analysis. In addition, this paper introduces time-to-stockout analysis for supply chain resilience and shows how to compute it through a labeled property graph model. The computational result shows that the proposed graph data model is efficient for recursive and variable-length data in supply chain, and relationship-centric graph query language is capable of handling a wide range of business questions with impressive query time.

KEYWORDS

Big Data, Graph Database, Industry 4.0, Risk Management, Supply Chain Resilience

INTRODUCTION

Globalization have stimulated automotive industry to develop globally interconnected and complex supply chain networks with greater physical distances. Henry Ford's supply chain was integrated and conceptually simple at the beginning of the twentieth century (The Economist, 2009). However, by the modern globalized economy, Ford accepted that they could not be the best in every field and began to develop interconnected and complex structures like *supply web* than supply chain (CSCMP's Supply Chain Quarterly, 2010). Thus, supply chain distribution risks increase with number of involved organizations in the network (Raghunath K. M., 2018). Supply chain data relationships become interconnected, and multi-tier rather than hierarchical or one-to-one. As the volume and complexity of data grow, supply chain managers require greater data transparency to analyze complex network behaviors to understand and support strategic decision makings. To achieve this, automotive industry companies need to digitize their supply chains to visualize better and understand how they work. However, due to its rapidly growing size and complexity of data, few companies have been able to apply

DOI: 10.4018/IJISSCM.2022010104

big analytics techniques to manage their supply chains. We developed a graph database framework to integrate multiple complex levels within an automotive supply chain network to enhance supply chain transparency.

One of the significant challenges for supply chain management is to mitigate risk by creating resilient supply chains. The literature has produced many definitions of supply chain resilience by several disciplines. Resilient supply chains require strong traceability systems for effective and timely decision making for external and internal changes. However, a recent survey shows that most supply chain companies still rely on spreadsheets to plan their supply chain process, making them less responsive and fragile (Supply Chain 247, 2018; Reuters Events, 2018). Recent advances in *big analytics* and AI support fast and effective decisions in many areas. In this paper, we evaluate one of the advanced databases in big data for supply chain management by testing its computational performance for big data analysis. Especially, we design a supply chain data model for graph database which tracks all the flow of raw materials from tier suppliers to finished products and their holistic interrelationships. Also, we introduce Time-to-Stockout analysis for supply chain resilience. The proposed Time-to-Stockout (TTS) performance metric could simplify the dynamic nature of the supply chain environment with respect to both market-side demand and supply-side inventory. It will offer a deeper knowledge of resilience and provide tools for managers to track and monitor the inventory risk, which can be propagated through supply chains. Several previous papers have evaluated graph databases and developed benchmarks. However, most of them used synthetic data or social network data. A few recent papers start to use real data to evaluate graph databases. As far as we know, this is the first paper to use a graph database for supply chain data. Especially, we evaluate it based on real data of Ford supply chain.

The remainder of the paper consists of five other sections. First, we introduce literature papers for the supply chain and graph database. Second, we present a way to model interconnected, multiple-depth supply chain data with a graph database. Third, we suggest the Time-to-Stockout performance metric and introduce the concept of Time-to-Stockout analysis. Fourth, we show the computational results obtained by applying a graph database to Ford supply chain data, and how do we support decisions for supply chain managers. Finally, we present conclusions and discussion.

LITERATURE REVIEW

Supply Chain Resilience

As globalization developed, supply chain resilience is currently an increasing concern since the supply chain is subject to diverse types of disruptions (Liao, Bayazit, & Wang, 2014; Ribeiro & Barbosa-Povoa, 2018). Today there are many definitions of supply chain resilience proposed by different authors in the operational management area (Pereira & Da Silva, 2015). Saenz et al. (2015) listed several definitions from 67 peer-reviewed articles from 2003 to 2013 on an emerging area of supply chain research. Even recent studies (Abidi, Bandyopadhayay, & Gupta, 2017; Lotfi, Mehrjerdi, Pishvaee, Sadeghieh, & Weber, 2019; Zare Mehrjerdi & Lotfi, 2019; Guoyi, Caiquan, Yubin, & Yunhui, 2020) suggest a sustainable and resilient closed-loop supply chain network. Briefly, resilient supply chains incorporate event readiness, are capable of providing an efficient response, and often are capable of recovering to their original state or even better post the disruptive event (Ponomarov & Holcomb, 2009). Therefore, a resilient supply chain needs to balance risk and costs to prevent or recover quickly from a multitude of dynamic and simultaneous risk-related disruptions (Deloitte, 2014).

One of key resilience factors is supply chain visibility (Ribeiro & Barbosa-Povoa, 2018). Supply chain visibility in multi-tier supply chains is characterized by traceability, mapping, and transparency. First, supply chain traceability is the ability to identify, trace and track the history, application or location of parts and products at any stage in the supply chain, as described by International Organization for Standardization (ISO) (2015). Second, supply chain mapping is a tool to show holistic picture of relationships within the supply chain. It enables graphic representation of all the potential dependency of parts and products within each step of the supply chain as it moves along the supply chain from supplier's parts to finished products. Carvalho et al. (2012) listed several types of supply chain mapping obtained from the literature. Finally, Supply chain transparency refers to the strategy to disclose information complying with internal governance and external regulations (Kraft, Valdés, & Zheng, 2018). The standardized management system turns out to be helpful in supply chain risk management (Zimon & Madzík, 2019).

To manage risk in multi-tier supply chains, it requires the ability to track and monitor supply chain events and the flow of materials from tier suppliers to end-users with the help of information technology. A lack of upstream visibility makes it difficult to take proactive, effective, and timely actions for external and internal changes. Unfortunately, a recent survey shows 65% of companies are still using spreadsheets in planning their supply chain process since spreadsheets are familiar, inexpensive, and convenient (Supply Chain 247, 2018; Reuters Events, 2018). A significant downside to spreadsheets is high maintenance cost and lack of transparency in multi-tier mapping, tracking, and reporting (Worldfavor, 2020). For reliable and transparent data in the supply chain, it needs to move towards an advanced analytics database.

Big Data Analytics in Supply Chain

The term 'Industry 4.0' is a new industrial revolution paradigm enabled by the introduction of the Internet of Things (IoT) into the production and manufacturing environment (Tjahjono, Esplugues, Ares, & Pelaez, 2017). Industry 4.0 involves digitization, connectivity, and intelligence in manufacturing environments and a variety of data analytics that enables flexible automation and rapid manufacturing for mass customization. Especially, by the negative impact of COVID-19 outbreak on the global supply chains, it accelerates to use several useful technologies of Industry 4.0 and the internet of things (IoT) (Javaid, et al., 2020). With the world moving towards Industry 4.0, the number of machines, processes, and services generating and collecting large quantities of data will increase significantly. It will give rise to *Big Data*, which is enormous amounts of data that cannot be processed with conventional computation techniques (Awwad, Kulkarni, Bapna, & Marathe, 2018).

According to Gartner (2020), *Big Data* is defined as "high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation". Recently, the big data concept is expanded by adding two more features – veracity and value (Oncioiu, et al., 2019). Hence, big data is characterized by $5V$ s, which are volume, velocity, variety, veracity, and value. The $5V$ s can be explained as follows: (1) volume refers to the magnitude of data that requires increased storage devices (Chen & Zhang, 2014); (2) variety is reflected by generating data from heterogeneous sources Internet of Things (IoT), online social networks, and structured, semi-structured, and unstructured formats (Tan, Zhan, Ji, Ye, & Chang, 2015); (3) velocity is given by the time to access, process, and use data in real-time (Assunção, Calheiros, Bianchi, Netto, & Buyya, 2015); (4) veracity reflects the importance of data quality and reliability (White, 2012; Gandomi & Haider, 2015); and (5) value is reflected by revealing unused data in big data and can support decision-making (Dijcks, 2013; Lee, Kang, Ye, & Wu, 2018). It is essential to understand the $5V$ s of supply chain data to leverage the full potential of Industry 4.0 in manufacturing.

Graph Database

Relational Databases have been dominating the computer industry since the 1980s, mainly for storing and retrieving data in tabular format (Batra & Tyagi, 2012). However, as the complexity of interconnected data increases with *Big Data*, the trends are driving change toward new database technologies called the *NoSQL movement* (Angles, 2012). Traditional databases are not efficient anymore to extract information from the graph-like data. Instead, Graph databases are quickly gaining popularity in the database community for relationship-rich data (ShefaliPatil & Bhatia, 2014).

A graph database is a *NoSQL* database model based on graph theory that stores data on relationship-rich data as a collection of nodes and edges (Coronel & Morris, 2016). Recently, most social networking sites and the hyperlink networks on the internet are highly complex and almost impossible to model efficiently in a relational database. Graph databases are optimized for these types of networks, as a graph is a natural way of storing connections between users (Batra & Tyagi, 2012). Other common use cases for graph databases are recommender systems, business relationships, network impact analysis, geospatial applications such as maps and route planning for rail or logistics, telecommunication or energy distribution networks, fraud detection, and many more (ShefaliPatil & Bhatia, 2014; Ferro & Sinico, 2018).

Graph Databases provide a natural data modeling technique, powerful relationship-centric query languages, structures, and algorithms for the graph-like data. Practically, there are a set of use cases and data patterns whose performance improves by one or more orders of magnitude when implemented in a graph, and whose latency is much lower than batch processing of aggregates. On top of this performance benefit, graph databases offer an extremely flexible data model, and a mode of delivery aligned with today's agile software delivery practices (Robinson, Webber, & Eifrem, 2013). Also, Vukotic et al. (2014) evaluate the performance of a particular query that finds friends-of-friends in a social network with a maximum depth of five. Robinson et al. (2013) test the query in both RDBMS and Neo4j with a database of 1,000,000 users, each with approximately 50 friends. The results strongly suggest that graph databases outperform RDBMS on connected data, as we see in Table 1.

In recent years, several papers have evaluated graph databases and developed benchmarks. Most of the papers used synthetic data to evaluate graph databases (Vicknair, et al., 2010; Ciglan, Averbuch, & Hluchy, 2012; Jouili & Vansteenberghe, 2013; McColl, Ediger, Poovey, Campbell, & Bader, 2014; Pacaci, Zhou, Lin, & Özsu, 2017). Abul-Basher et al. (2016) used two real datasets from the SNAP repository (Leskovec & Krevl, 2014), namely *Wiki-Talk* and *Slashdot*, to evaluate Neo4J, OrientDB and other systems. Ferro and Sinico (2018) developed a benchmark for graph database systems based on real data of Italian Business Register in October 2016, consisting of about 10.5 million companies and physical persons and about 5 million relationships among them. Kolomičenko et al. (2013) used both synthetic data and real data in the SNAP repository collected from *Amazon's co-purchasing network* to evaluate several systems (Leskovec & Krevl, 2014). Finally, the performance difference between a graph database and a traditional relational database is evaluated by Vicknair et al. (2010), ShefaliPatil and Bhatia (2014), Ferro and Sinico (2018). Table 2 and Table 3 summarize previous researches according to benchmark data and databases respectively.

In this paper, we evaluate the computational performance of a graph database with a traditional database by using the real data of the Ford supply chain consisting of about 12 million nodes and 4.5 million relationships instead of synthetic data as many of the papers above used. As far as we know, this is the first paper to compare a graph database with an established relational database using real supply chain data, which is not covered by previous works.

Table 1. Finding extended friends in a relational database versus Neo4j (Robinson, Webber, & Eifrem, 2013)

Depth	RDBMS execution time (sec)	Graph DB execution time (sec)
2	0.016	0.01
3	30.267	0.168
4	1,543.505	1.359
5	Not finished in 1 hour	2.132

Table 2. Literature review summary table: benchmark data (R: real-world, S: synthetic)

Author(year)	Data	Size (Max)	Application
Vukotic et al. (2014)	friends-of-friends (S)	1000 users, 50 friends	Social network
Robinson et al. (2013)	friends-of-friends (S)	1,000,000 users, 50 friends	Social network
Vicknair et al. (2010)	directed acyclic graph (S)	100,000 nodes	None
Ciglan et al. (2012)	directed graph (S)	1,000,000 nodes (16 edges per node)	None
Jouili et al. (2013)	directed graph (S)	500,000 nodes, 2,500,000 edges	None
McColl et al. (2014)	undirected graph (S)	256,000,000 edges	None
Pacaci et al. (2017)	LDBC's Social Network Benchmark (S)	34,000,000 nodes, 217,000,000 edges	Social network
Abul-Basher et al. (2016)	Wiki-Talk (R), Slashdot (R)	Wiki-Talk: 2,400,000 nodes, 5,000,000 edges Slashdot: 82,000 nodes, 549,000 edges	Wikipedia talk network, Social network
Ferro and Sinico (2018)	Italian Business Register (R)	10,000,000 nodes, 5,000,000 edges	Economic and business
Kolomičenko et al. (2013)	directed graph (S), Amazon (R)	Synthetic: 200,000 nodes (10 edges per node) Amazon: 260,000 nodes, 1,234,000 edges	Product co-purchasing network
ShefaliPatil and Bhatia (2014)	Facebook (R), Twitter (R)	Facebook: 4,000 nodes, 88,000 edges Twitter: 81,000 nodes, 1,800,000 edges	Social network

Table 3. Literature review summary table: performance test between databases

Author(year)	Database(s)	Findings
Vukotic et al. (2014)	Neo4j, MySQL	Execution time(s)
Robinson et al. (2013)	Neo4j, RDBMS	Execution time(s)
Vicknair et al. (2010)	Neo4j, MySQL	Execution time(s)
Ciglan et al. (2012)	Neo4j, DEX, OrientDB, Native/Sail RDF, SGDB	Load time(s), Traversal time(s)
Jouili et al. (2013)	Neo4j, OrientDB, Titan, DEX	Load time(s), Traversal time(s)
McColl et al. (2014)	Titan, OrientDB, Neo4j, Pegasus, SQLite, DEX, Bagel, Giraph, NetworkX, MTGL, Boost, STINGER (12 open source graph databases)	Shortest paths, Connected components, PageRank, Updates (four graph algorithms)
Pacaci et al. (2017)	Titan, Neo4j, Virtuoso, Postgres, Sqlg	Query latencies(s)
Abul-Basher et al. (2016)	Neo4j, Titan, OrientDB	Workload test: load, analytical, atomical query
Ferro and Sinico (2018)	ArangoDB, Neo4J, OrientDB, PostgreSQL	Execution time(s)
Kolomičenko et al. (2013)	Neo4j, DEX, MongoDB, OrientDB, InfiniteGraph, Titan, Native/Sail RDF, TinkerGraph, MysqlGraph	Graph operations time(s)
ShefaliPatil and Bhatia (2014)	Neo4j, MySQL, Vertica, VoltDB	Shortest paths, PageRank

TIME-TO-STOCKOUT ANALYSIS

Modern globalized supply chains are increasingly susceptible to various events, including natural and man-made disasters (e.g., earthquake, labor disputes, terrorist attacks, and political changes). Local events in one area of the world can cause supply disruptions and shortages. Various supply chain resilience definitions, measurements, frameworks, and quantitative models are presented to enhance the resilience against the supply chain risk in the literature (Ribeiro & Barbosa-Povoa, 2018). There are various metrics defined. Each supply chain performance metric gives a slightly different view of a piece of the supply chain.

In this paper, we suggest Time-to-Stockout (TTS) performance metric for monitoring stock levels of inventory relative to demands by tracking all inventory in the supply chain pipeline between a supplier and a final product. This method quantifies inventories across the entire supply chain between a supplier and a final product by a single numerical metric. Also, inherently and implicitly, it considers the holistic structure of the supply chain network. This metric offers a new way to measure "supply chain fitness" and provides critical insights for decision making by monitoring the inventory levels.

To be a good performance metric for effective supply chain management, it should be quantitively easy to understand so as to make the user take the correct action. Especially, useful performance metrics should be easy to calculate and collect from the system. Our TTS performance metric is a single numerical metric that has a very clear definition to understand intuitively. Especially, we will show how much it is easy and fast to compute the TTS through graph database based on the property graph model.

Supply Chain Data Modeling With Graph Database

A Supply chain in the automotive industry is an integrated process in which a set of several suppliers, warehouses, intermediate component plants (i.e., stamping, transmission, and engine plants), and final vehicle assembly plants work together. It purchases raw materials, converts raw materials to finished products, and sells them to customers (Carvalho, Silveira, & Ramos, 2010). Consequently, the automotive supply chain network has complex, interconnected, multiple-depth relationships. On the other hand, the relationship between upstream suppliers and downstream customers is essential to increase customer satisfaction and firm performance. Therefore, understanding supply chain relationships is a crucial driver of firm performance (Kannan & Tan, 2005). Also, effective supply chain management is vital to build and sustain a competitive advantage in the product and services of the firms (Gunasekaran & Ngai, 2004).

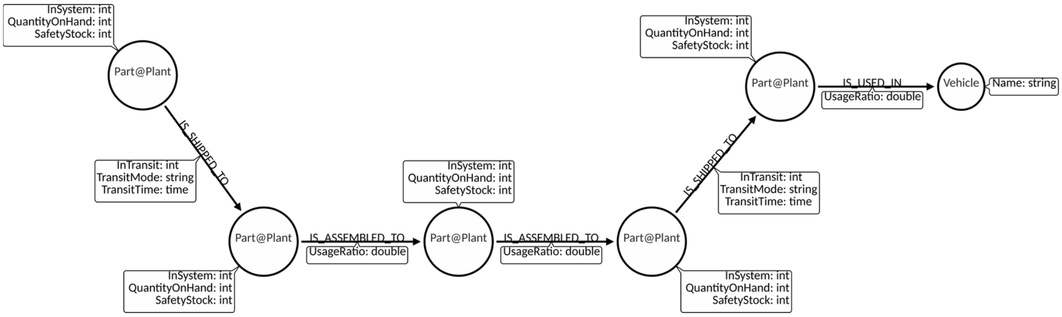
This paper suggests a new approach to model interconnected, multiple-depth supply chain data with the property graph model. A property graph is a directed graph where both nodes and Relationships can contain any number of properties. Nodes store properties in the form of arbitrary key-value pairs. Relationships connect and structure nodes. Like nodes, relationships can also have properties (Robinson, Webber, & Eifrem, 2013). The ability to add properties to relationships in the network is the main distinction to Resource Description Framework (RDF) Graphs (Miller, 1998). Since the supply chain needs to consider inventory in transit, we use the property graph model rather than RDF triple stores model. Also, it is known that Property Graph databases tend to be optimized for graph traversals (Alocci, et al., 2015). With RDF triple stores, the cost of traversing an edge tends to be logarithmic (Angles, Prat-Pérez, Dominguez-Sal, & Larriba-Pey, 2013).

When Transforming supply chain data into a property graph, each part in a plant is represented as a *"node"*, and transactions between nodes are represented as *"relationships"*. In automotive supply chains, there are two major transactions: *"IS_SHIPPED_TO"* and *"IS_ASSEMBLED_TO"*. *"IS_SHIPPED_TO"*, where transit volume, time, and mode are main properties for inventory control, represent shipping parts physically from one plant to another plant. *"IS_ASSEMBLED_TO"* represents a parent-component relationship in the bill of materials (BOM) to build a parent product and its quantity per assembly (QPA) for the relationship. For each part in a plant, it has inventory properties such as quantity in process, quantity on hand, and safety stock. The benefits of this supply chain data modeling are described in the following way.

Performance

One compelling reason for choosing a graph database is the performance benefit when dealing with connected data. While join-intensive query performance in relational databases will often deteriorate over time as the dataset grows both in size and connectedness, a graph database tends to offer relatively consistent performance regardless of the size and density of connections. It is because queries are

Figure 1. Supply Chain Data Modeling for Property Graph Database



localized to a portion of the graph. As a result, the execution time for each query is proportional only to the size of the part of the graph traversed to satisfy that query, rather than the size of the overall graph (Robinson, Webber, & Eifrem, 2013).

Relationship-CENTRIC Analytics

Traditional Relational Database Management Systems (RDBMS) are not designed for relationships among individual data points. By contrast, Graph Databases focus on the graph-like relationships between data points rather than the individual data themselves. It reveals valuable insights from data relationships by relationship-centric Graph Query Language (GQL). While graph traversals in RDBMS are much more complicated and can involve looping or recursing through the graph, possibly executing multiple expensive joins along the way, graph traversals are fairly simple in Graph Databases (Vicknair, et al., 2010). Furthermore, their inherent relationship-centric approach enables it to present data graphically in such a way as to make it understandable and intuitive to users.

Flexibility

Although relational databases are more mature and secure than graph databases, they depend on a rigid schema, which makes it difficult to add new relationships between objects (Angles & Gutierrez, 2008) and less suitable when the data model evolves over time (Batra & Tyagi, 2012). Graphs are naturally additive, meaning we can add new kinds of relationships, new nodes, new labels, and new subgraphs to an existing structure without disturbing existing queries and application functionality. Because of the flexibility, it does not need to model a domain in exhaustive detail ahead of time. It enables us to add new subgraphs to an existing graph structure without disturbing existing functionality. The additive nature of graphs also means we tend to perform fewer migrations, thereby reducing maintenance overhead and risk (Robinson, Webber, & Eifrem, 2013).

Agility

By today’s agile trends, it is required to evolve our data model in step with incremental and iterative software development and any changing business requirements. Since modern graph databases are equipped for frictionless development and graceful systems maintenance, developing with graph databases aligns perfectly with today’s agile, test-driven development practices. In particular, the schema-free nature of the graph data model empowers us to evolve an application in a controlled manner (Robinson, Webber, & Eifrem, 2013).

Performance Benchmarks

In traditional RDBMS databases, the data is stored in a tabular format, where each table has a fixed number of columns, and each column has its own data type. However, given the table structure, graph traversal queries become quite slow as the size and depth of relationships increase. In contrast, graph databases connect nodes physically point to each other in the database (Robinson, Webber, & Eifrem, 2013). Thus, its performance stays constant even as data grows.

Time-to-Stockout

In this subsection, we introduce a metric called Time-to-Stockout (TTS) to measure the inventory fitness of the supply chain when the supply disruption happens. Inventory is a fundamental measure of the overall health of supply chain and logistics activities (Waller, Esper, & others, 2014). Thus, inventory-related costs and metrics are critical for supply chain resilience. In the real world, inventory for a specific product is stored across multiple facilities or sites in the supply chain network. We decompose the supply chain inventory into the following concepts:

- **Quantity in Process (QIP):** Unfinished items in the manufacturing or assembly process that have been partially completed through the production process.
- **Quantity in Transit (QIT):** Items that have been sent, but it has not been received at the other plant yet. They are still in transit status for the next manufacturing step.
- **Quantity on hand (QOH):** The total number of items above a safety stock level, which are physically available in a plant (including warehouse and consignment inventory), minus any items that have already been canceled or rejected.
- **Safety Stock (SS):** Reserved quantity of an item held in inventory to reduce the operational risk.
- **Bulk on Hand (BOH):** The total number of available component items they have for building future parent products, which is the sum of four concepts above (i.e., quantity-in-process, quantity-in-transit, quantity-on-hand, and safety stock).

It is often difficult to predict demand patterns with precision or accuracy. Also, a steady supply of raw materials and components is critical for manufacturing. The mismatch between supply and demand in the supply chain is one of the essential factors to assess supply chain risk. Thus, the inventory plays an important role in the supply chain to balance demand and supply. We define the TTS metric to consider both supply-side and demand-side changes together.

TTS is a metric that measures survival days that we can continue to make a final product under a specific production plan or demands even after some specific supplier part has in trouble to supply. Implicitly, it is defined between a tier supplier part and a finished product (i.e., vehicle and so on) in the assembly plant. To compute it, it first needs to know the entire connections between a tier part and a final product through the supply chain network. For each step in the supply, we need to collect respectively BOH and total demand for each intermediate part. However, total demand for an intermediate part is not trivial since it depends on the structure of the connected network, quantity per assembly for each parent-component relationship along the chain, and demands or a production plan for finished products that requires the intermediate part. Graph Database plays an important role in simplifying and accelerating this numerical computation based on the graph data model we suggest above. The Graph Database is queried through Graph Query Language (GQL), which is a declarative and efficient query language for the graph traversals. Then, we can quantify the following concepts under a specific production plan or demands:

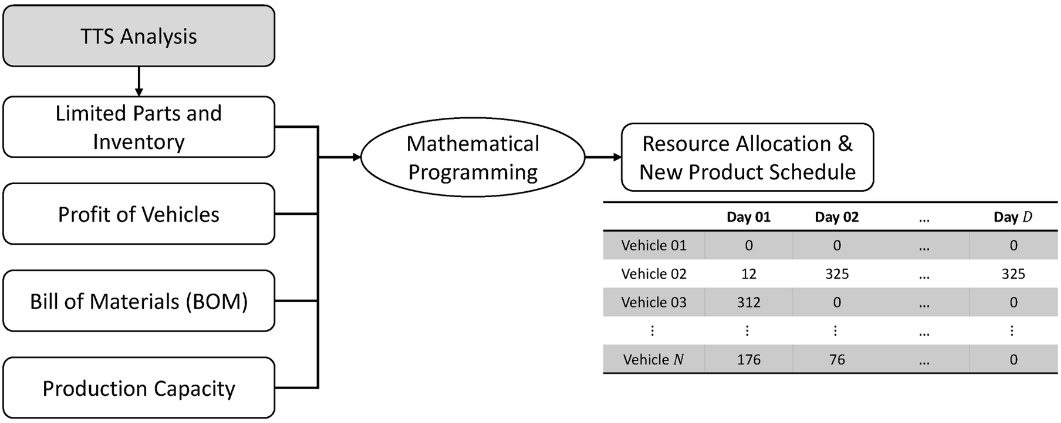
- **Demand per Unit Time (DPU):** The number of items we need to produce a planned volume of finished productions per unit time.
- **Days on Hand (DOH):** the number of calendar days the BOH quantity of a part will last relative to the demand for final productions. $DOH_i = \frac{BOH_i}{DPU_i}$, where i is a part in the supply chain data.
- **Time to Stockout (TTS):** the number of survival days that we can continue to make a final product using only pipeline inventory in the supply chain. $TTS_{uv} = \sum_i DOH_i$, where i is a part in the connection chain between a supplier part u and a final production v.

Time-to-Stockout analysis with the graph data model provides a deeper understanding of the underlying network structure and a comprehensive framework to measure and analyze supply chain resilience.

Resource Reallocation

The resource allocation problem seeks to find an optimal allocation of a fixed amount of resources to activities so as to minimize the cost incurred by the allocation (Katoh & Ibaraki, 1998). There are various optimization methods that have been proven successfully in solving

Figure 2. Resource Reallocation for TTS Analysis



	Day 01	Day 02	...	Day D
Vehicle 01	0	0	...	0
Vehicle 02	12	325	...	325
Vehicle 03	312	0	...	0
⋮	⋮	⋮	...	⋮
Vehicle N	176	76	...	0

resource allocation problems in industrial production planning (Lombardi & Milano, 2012). In this subsection, we demonstrate how to reallocate the limited resources identified by TTS analysis to maximize profit using mathematical programming. We consider net profit for each vehicle line, production capacity per day, and bill of materials described in Figure 2. Then, we get a daily production schedule for vehicles that need to be produced in an assembly plant by solving the following resource allocation problem.

Table 4 shows the list of sets and parameters for a manufacturing plant. Here, we have a set of limited parts and vehicles which use the parts. Then, given the planning period, it reallocates limited inventory I_p for part p based on the usage ratio between part p and vehicle v on the bill of materials (BOM). Finally, we use the production capacity limit for each vehicle to consider assembly line and labor constraints in the manufacturing plant.

Given the sets and parameters in Table 4, the resource reallocation problem for TTS analysis can be formulated as the following mathematical formulation in Table 5. Objective (1) maximizes the total profits over vehicles. Constraint set (2) ensures that each limited part can be used no more than the current available inventory. Constraint set (3) provides daily production capacity constraints for a plant. Finally, Constraint set (4) specifies that decision variables must be non-negative.

x_v^d is the decision variable representing the amount of vehicle v that will be produced on day d, $\forall v \in V, \forall d \in D$.

Table 4. Sets and Parameters

P	set of parts
V	set of vehicles
D	planning horizon (operation days)
w_v	profit for a vehicle v, $\forall v \in V$
C_d	daily production capacity on day d, $\forall d \in D$
I_p	inventory of part p, $\forall p \in P$
e_{pv}	usage ratio of part p that needs to make the vehicle v, $\forall p \in P, \forall v \in V$

Table 5. Formulation

$$\max \quad \sum_{v\in V}\sum_{d\in D} w_v x_v^d \qquad (1)$$

$$\text{subject to:} \quad \sum_{v\in V}\sum_{d\in D} e_{pv} x_v^d \le I_p \qquad \forall p \in P \qquad (2)$$

$$\sum_{v\in V} x_v^d \le C_d \qquad \forall d \in D \qquad (3)$$

$$x_v^d \in \mathbb{Z}_0^+ \qquad \forall v \in V, \forall d \in D \qquad (4)$$

COMPUTATIONAL RESULTS

The proposed graph database modeling in the previous section was implemented in Neo4j 3.5.5 and ran on Intel i7-8850H CPU 2.6 GHz laptop with 64.0 GB of RAM memory. To access supply chain data from Ford Hive data warehouse, we implemented ETL (extract, transform, load) tool in Java and Python with the help of H2 in-memory Java SQL database and PySpark to accelerate ETL process. We performed Time-to-Stockout analysis of all pairs of active suppliers and vehicle lines in the Ford supply chain, and its computational time was within a day. Figure 2 shows that TTS distribution over parts for a vehicle transmission in Ford. We observed that most of the parts for the transmission have less than 10 TTS days. However, we have found that a set of some parts clustered around 50 TTS days. Also, we can identify a few overstock parts that have more than 100 TTS days.

We also compare computational performance between traditional SQL and graph databases by using time-dependent supply chain data based on Figure 1. Part changes over time by evolving its function, color, design, material, and so on. For example, in Figure 3, tier-one part A evolves to part A' and part A'' by minor changes, and it changes to part B from part A family due to major changes. The change of tier-one parts impacts the assembly relationships of the whole downstream parts up to finished products. It causes the compatibility and safety issue with a final product when it switches from an obsolete part to a new replacement part. Therefore, tracking this kind of part replacement information over time is essential as a vehicle model evolves over the years.

Figure 3. Time-to-Stockout (TTS) Analysis of Transmission Parts

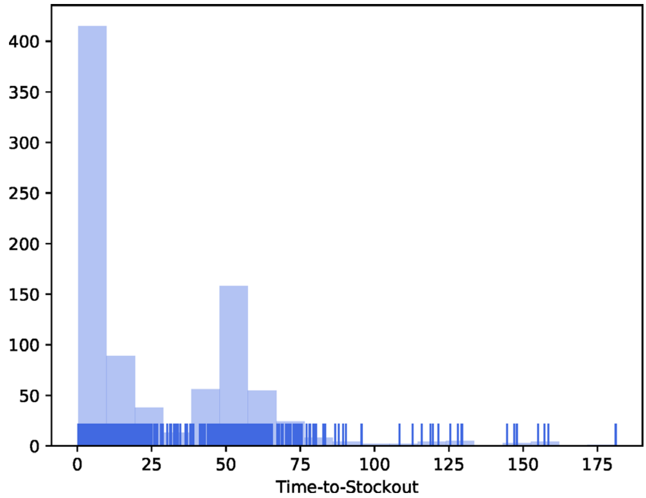
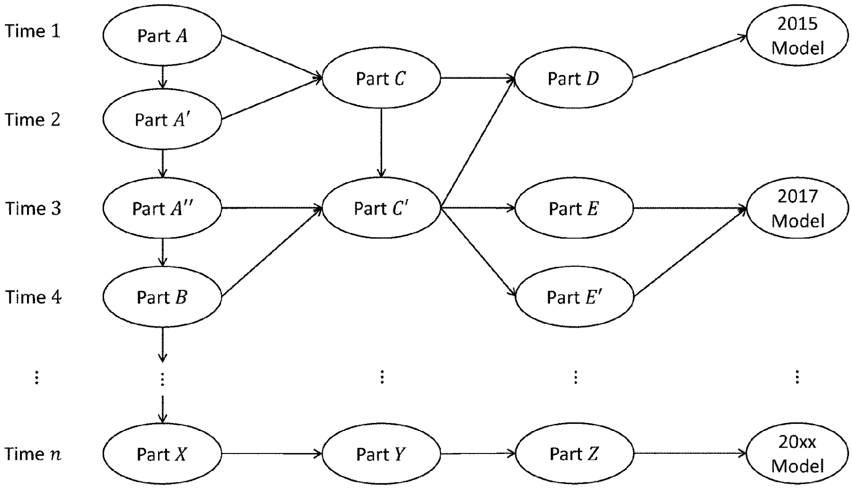


Figure 4. Time-dependent Part Replacement Supply Chain Network



In this paper, we could not show complete details about all supply chain data because of Ford data security. Instead, we demonstrate Ford part replacement supply chain data related to some commodities (i.e., engine, seat, transmission, and so on) to compare computational performance between traditional SQL and graph database. We highlight the key data characteristics, which can make it possible to reproduce a similar result, in Table 2 and Table 3. The supply chain test data contains two columns named *'Up'* and *'Down'* respectively. Each record represents a child-parent relationship between two different parts. To reduce the computational complexity, we calculated in advance all the possible root nodes and leaf nodes from the raw data by counting the number of child-parent relationships (e.g., a root part has no parents, or a leaf part has no children).

As the complexity of the relationship, Table 3 shows the histogram of the depth of each tree. Most of the child-parent relationships have a shallow tree. Only a few trees have more than 100 depths. The deepest tree of the test data has 162 levels.

Finally, this kind of data profiling to make Table 2 and Table 3 by discovering statistics or patterns in graph data is fairly simple in GQL. In contrast, SQL does not support concise traversal syntax like GQL. It even might not easy to find out the depth of the deepest tree since it needs to check the depth of all trees by traversing to find the deepest node in each tree, which is computationally expensive in RDBMS.

Common Table Expression

A recursive common table expression (CTE) is an SQL query that handles hierarchical model data. It can be used to traverse relations in a tree or graph. This recursive CTE consists of two main parts:

Table 6. Number of Records and Data Size for Ford Supply Chain Test Data

Test Data	Number of Records	Data Size (MB)
Parts	5,366,735	46.5
Relationships	4,469,041	73.3

Table 7. Frequency Table of Part Tree Depth

Depth	Number of Trees
0 - 9	803,408
10 - 19	75,928
20 - 29	15,026
30 - 39	4,426
40 - 59	2,206
60 - 79	562
80 - 99	138
100 - 200	48
Total	901,742

- **Anchor Member:** This query is initially executed once to return the base result of the CTE.
- **Recursive Member:** This query is repeatedly executed until the termination condition is met or no rows are returned.

By switching the role of *'Up'* and *'Down'* columns in the recursive execution, the CTE repeatedly executes, returns subgraphs, until it returns the complete graph. The following Figure 4 is an example of CTE in SQL for supply chain data.

Graph Database: Neo4j

Graph databases are focused on efficient storing and querying highly connected data (Pokornỳ, 2018). For relationship-centric data definition, query, and manipulation, graph database provides pattern matching query language that focuses on the relationships between entities. The most and first known GQL to target the property graph data model is Cypher of Neo4j Graph Database (Robinson, Webber, & Eifrem, 2013). Cypher query language is very expressive and efficient to handle connected data without explicitly writing traversal algorithms in the code. On the other hand, graph patterns are not easily expressible in SQL, and the complexity and cost of the recursive SQL query grow quickly as it calls additional joins. The following Figure 5 is an example of Cypher query to retrieve graph-structured information with variable-length relationships.

Compared to CTE query in the previous section, Cypher query language has rich and expressive syntax for graph traversals. It can easily match the pattern with variable-length relationships. Unlike SQL syntax that focuses on tables and columns, Cypher is a more human-readable description of relationships.

Computational Time

We performed a query to traverse data to retrieve a set of nodes for the deepest tree from a given root part a few times against a small data set of supply chain data described in the earlier section. At each test, we ran the query 5000 times – this was simply to warm up any caches that could help with performance. Total execution time was recorded, and we calculated the average execution time for each run. No additional database performance tuning was performed except for the composite index used in the second case in Table 4. Since it is not always possible to have an index on columns by the company's data governance policy and other data transaction constraints, we also tested it with no index in the third case in Table 4.

Figure 5. SQL Code for Common Table Expression (CTE)

```
SET STATISTICS TIME ON
DECLARE @root varchar(max);
set @root = ’PartID’;

-- Define the CTE expression name and column list.
WITH CTE_TABLE (Up, Down, Depth, Chain)
-- Define the CTE query.
AS (
--initialization
SELECT Up, Down, 0 as Depth, cast(@root as varchar(max)) as Chain
FROM [SQL_TABLE]
WHERE Up = @root
--initialization
UNION ALL
--recursive execution
SELECT a.Up, a.Down, r.Depth + 1, cast(r.Up + ’,’ + r.Chain as varchar(max))
FROM [SQL_TABLE] a INNER JOIN [CTE_TABLE] r
ON a.Up = r.Down
--recursive execution
)

SELECT * FROM CTE_TABLE
OPTION (MAXRECURSION 0)
SET STATISTICS TIME OFF
```

Figure 6. Cypher Code for Neo4j Graph Database

```
MATCH path = (root:Part)-[*]->(leaf:Part) WHERE root.FID = ’PartID’ AND leaf.numChild = 0
    RETURN nodes(path)
//Cypher version: CYPHER 4.0
```

Table 8. Query Time of GDB and RDBMS with Ford Supply Chain Test Data

Database	Number of Runs	Total Execution Time(ms)	Average Execution Time(ms)
Graph DB(Neo4j)	5,000	8,709	1.7418
RDBMS (MS SQL - indexing)	5,000	53,230	10.646
RDBMS (MS SQL)	5	524,400	104,880

The results of the experiment strongly suggest that a graph database is much faster than RDBMS for graph-structured data, as we see in Table 4. Neo4j is six times faster than RDBMS with indexing, and it is even 60,000 times faster than RDBMS if we are not able to use an index on relationship columns in RDBMS.

Finally, we test a computational time for the mathematical formulation described in Table 5 using various data sizes. We used plant data where four types of vehicles are produced by using approximately less than 40,000 parts per vehicle. We extrapolated total inventory volume for various planning horizons linearly from inventory data of five days. Computational time in Table 9 shows that the resource reallocation model for TTS analysis works with real-world scale data.

Table 9. CPU Time for Resource Reallocation Optimization Model with Various Data Sizes

CPU Time (sec.)	Planning Horizon (days)					
#Parts	5 days	10 days	15 days	20 days	25 days	30 days
100	3.11	3.69	2.94	2.9	3.03	3.2
500	3.31	3.44	3.16	3.24	3.08	3.46
1,000	3.2	4.63	3.34	3.62	3.65	5.27
5,000	5.96	7.38	7.53	8.25	8.35	8.64
10,000	9.83	12.37	13.27	13.33	15.28	16.37
20,000	17.54	21.56	21.9	22.13	24.61	25.26
30,000	28.32	31.52	32.68	32.98	33.64	34.54
40,000	38.99	39.98	40.54	42.04	45.02	46.13

CONCLUSION

Modern supply chains are vulnerable to wide various events (Blos, Quaddus, Wee, & Watanabe, 2009; Gaudenzi & Qazi, 2020), including natural and man-made disruptions. Local events in one area of the world can cause a ripple effect that lead to supply disruptions and shortages. To improve supply chain resilience and support supply chain management in the right direction, it is necessary to have a thorough and complete understanding of every element and relationship in the supply chain. End-to-end visibility over the entire supply chain will help a decision-maker respond properly when any unexpected event occurs. Furthermore, the volume and complexity of supply chain data have increased significantly and continue to trend up during the Industry 4.0 movement. In addition, the relationships between supply chain data can be more important than the individual data themselves to create insights and business values. Thus, supply chain management requires a database technology that can handle data relationships effectively. Unfortunately, traditional SQL-based RDBMS are poor at handling data relationships because their rigid schemas make it difficult to add new or different kinds of relationships or adapt to change in a fast-changing business environment.

We used the real data of Ford supply chain consisting of about 12 million nodes and 4.5 million relationships. To the best of our knowledge, this is the first paper to benchmark a graph database using real supply chain data. First, this research provides a new understanding of the supply chain graph database to obtain deeper insights from big data. Second, we introduce the TTS metric to improve supply chain resilience with respect to visibility, responsiveness, and readiness. Time-to-Stockout analysis can help manage and monitor inventory fitness with a deeper and complete understanding of the supply chain network. The computational results show that a graph database for supply chain data enhances computational performance significantly. Also, it allows calculating the relative inventory status of each part in real large-scale data through Time-to-Stockout analysis. It helps us to identify limited parts that have the relative inventory shortage given the demand forecast. Finally, given the list of limited parts identified by the Time-to-Stockout analysis, we propose the mathematical programming to reallocate the limited resources to maximize profit and test its scalability based on various supply chain data sizes.

This paper is not limited to academic research in supply chain management. The proposed approaches are tested as a supply chain industry platform, including graph database, visualization, performance metrics for bottleneck detection, and resource reallocation optimization to support operational decisions. Especially, this paper has successfully demonstrated that a graph database can help process complex, interconnected, multiple-depth supply chain data quicker than traditional SQL databases.

Unlike academic research, people in the industry field need crisp and clear key performance indicators (KPIs) to gain business insights from complex data and communicate with non-technical background decision-makers and stakeholders. Therefore, it is beneficial in the industry field if the performance metric is simple to understand without training instead of complicated black-boxed performance metrics suggested by academic papers. The need for intuitive KPI motivates us to propose the Time-to-Stockout metric to keep track of inventory in the supply chain network. Also, there is always a time limit for business decisions. In many cases, it needs to be almost real-time or a few days. Therefore, it requires operations analytical tools to have scalability for massive supply chain data set to gain insights quickly and test various decision scenarios. They may not have days to investigate and verify the KPI's assumptions. The proposed supply chain data modeling with a graph database and resource reallocation formulation satisfied the time and simplicity requirements.

REFERENCES

Abidi, N., Bandyopadhayay, A., & Gupta, V. (2017). Sustainable supply chain management: A three dimensional framework and performance metric for Indian IT product companies. *International Journal of Information Systems and Supply Chain Management*, *10*(1), 29–52. doi:10.4018/IJISSCM.2017010103

Abul-Basher, Z., Chignell, M. H., Godfrey, P., & Yakovets, N. (2016). TGDB: towards a benchmark for graph databases. *Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering*, 257–267.

Alocci, D., Mariethoz, J., Horlacher, O., Bolleman, J. T., Campbell, M. P., & Lisacek, F. (2015). Property graph vs RDF triple store: A comparison on glycan substructure search. *PLoS One*, *10*(12), 10. doi:10.1371/journal.pone.0144578 PMID:26656740

Angles, R. (2012). A comparison of current graph database models. *2012 IEEE 28th International Conference on Data Engineering Workshops*, 171–177.

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, *40*(1), 1–39. doi:10.1145/1322432.1322433

Angles, R., Prat-Pérez, A., Dominguez-Sal, D., & Larriba-Pey, J.-L. (2013). Benchmarking database systems for social network applications. *First International Workshop on Graph Data Management Experiences and Systems*, 1–7. doi:10.1145/2484425.2484440

Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A., & Buyya, R. (2015). Big Data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, *79*, 3–15. doi:10.1016/j.jpdc.2014.08.003

Awwad, M., Kulkarni, P., Bapna, R., & Marathe, A. (2018). Big Data Analytics in Supply Chain: A Literature Review. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 418–425.

Batra, S., & Tyagi, C. (2012). Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering*, *2*, 509–512.

Blos, M. F., Quaddus, M., Wee, H. M., & Watanabe, K. (2009). Supply chain risk management (SCRM): A case study on the automotive and electronic industries in Brazil. *Supply Chain Management*, *14*(4), 247–252. doi:10.1108/13598540910970072

Carvalho, H., Cruz-Machado, V., & Tavares, J. G. (2012). A mapping framework for assessing supply chain resilience. *International Journal of Logistics Systems and Management*, *12*(3), 354–373. doi:10.1504/IJLSM.2012.047606

Carvalho, M. F., Silveira, A., & Ramos, J. (2010). A Methodology for Supply Chain Design An Application to Auto-part Industry. *Management and Control of Production and Logistics*, 139–143.

Chen, C. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314–347. doi:10.1016/j.ins.2014.01.015

Ciglan, M., Averbuch, A., & Hluchy, L. (2012). Benchmarking traversal operations over graph databases. *2012 IEEE 28th International Conference on Data Engineering Workshops*, 186–189.

Coronel, C., & Morris, S. (2016). *Database systems: design, implementation, & management*. Cengage Learning.

CSCMP's Supply Chain Quarterly. (2010). *From supply chain to ``supply web"*. Retrieved from From supply chain to ``supply web": https://www.supplychainquarterly.com/archives/2010/04/

Deloitte. (2014). *Supply Chain & Operations Leverage the backbone of your business as a source of competitive advantage*. Retrieved from Supply Chain & Operations Leverage the backbone of your business as a source of competitive advantage: https://www2.deloitte.com/content/dam/Deloitte/be/Documents/Operations/201411_SupplyChainOperations_2014.pdf

Dijcks, J.-P. (2013). Oracle: Big data for the enterprise. *Oracle white paper*, 16.

Ferro, N., & Sinico, L. (2018). *Graph Databases Benchmarking on the Italian Business Register*. SEBD.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. doi:10.1016/j.ijinfomgt.2014.10.007

Gartner. (2020). *Gartner IT Glossary*. Retrieved from Gartner IT Glossary: https://www.gartner.com/it-glossary/big-data/

Gaudenzi, B., & Qazi, A. (2020). Assessing project risks from a supply chain quality management (SCQM) perspective. *International Journal of Quality & Reliability Management*, *38*(4), 908–931. doi:10.1108/IJQRM-01-2020-0011

Gunasekaran, A., & Ngai, E. W. (2004). Information systems in supply chain integration and management. *European Journal of Operational Research*, *159*(2), 269–295. doi:10.1016/j.ejor.2003.08.016

Guoyi, X., Caiquan, D., Yubin, Z., & Yunhui, Z. (2020). Research on Multi-Period Closed-Loop Supply Chain Network Equilibrium Based on Consumers' Preference for Products. *International Journal of Information Systems and Supply Chain Management*, *13*(4), 68–94. doi:10.4018/IJISSCM.2020100104

International Organization for Standardization (ISO). (2015). *ISO 9000, 2015. Quality management systems–fundamentals and vocabulary*. Retrieved from ISO 9000, 2015. Quality management systems–fundamentals and vocabulary: https://www.iso.org/standard/45481.html

Javaid, M., Haleem, A., Vaishya, R., Bahl, S., Suman, R., & Vaish, A. (2020). Industry 4.0 technologies and their applications in fighting COVID-19 pandemic. *Diabetes & Metabolic Syndrome*, *14*(4), 419–422. doi:10.1016/j.dsx.2020.04.032 PMID:32344370

Jouili, S., & Vansteenberghe, V. (2013). An empirical comparison of graph databases. *2013 International Conference on Social Computing*, 708–715. doi:10.1109/SocialCom.2013.106

Kannan, V. R., & Tan, K. C. (2005). Just in time, total quality management, and supply chain management: Understanding their linkages and impact on business performance. *Omega*, *33*(2), 153–162. doi:10.1016/j.omega.2004.03.012

Katoh, N., & Ibaraki, T. (1998). Resource allocation problems. In *Handbook of combinatorial optimization* (pp. 905–1006). Springer. doi:10.1007/978-1-4613-0303-9_14

Kolomičenko, V., Svoboda, M., & Mlỳnková, I. H. (2013). Experimental comparison of graph databases. *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, 115–124.

Kraft, T., Valdés, L., & Zheng, Y. (2018). Supply chain visibility and social responsibility: Investigating consumers' behaviors and motives. *Manufacturing & Service Operations Management*, *20*(4), 617–636. doi:10.1287/msom.2017.0685

Lee, A. H., Kang, H.-Y., Ye, S.-J., & Wu, W.-Y. (2018). An integrated approach for sustainable supply chain management with replenishment, transportation, and production decisions. *Sustainability*, *10*(11), 3887. doi:10.3390/su10113887

Leskovec, J., & Krevl, A. (2014). SNAP Datasets: Stanford Large Network Dataset Collection. *SNAP Datasets: Stanford Large Network Dataset Collection*.

Liao, K., Bayazit, O., & Wang, F. (2014). Building a natural disaster risk index for supply chain operations. *International Journal of Information Systems and Supply Chain Management*, *7*(4), 20–30. doi:10.4018/ijisscm.2014100102

Lombardi, M., & Milano, M. (2012). Optimal methods for resource allocation and scheduling: A cross-disciplinary survey. *Constraints*, *17*(1), 51–85. doi:10.1007/s10601-011-9115-6

Lotfi, R., Mehrjerdi, Y. Z., Pishvaee, M. S., Sadeghieh, A., & Weber, G.-W. (2019). A robust optimization model for sustainable and resilient closed-loop supply chain network design considering conditional value at risk. *Numerical Algebra, Control & Optimization*, 0.

McColl, R. C., Ediger, D., Poovey, J., Campbell, D., & Bader, D. A. (2014). A performance evaluation of open source graph databases. *Proceedings of the first workshop on Parallel programming for analytics applications*, 11–18. doi:10.1145/2567634.2567638

Miller, E. (1998). An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, *25*(1), 15–19. doi:10.1002/bult.105

Oncioiu, I., Bunget, O. C., Türkeș, M. C., Căpușneanu, S., Topor, D. I., Tamaș, A. S., Rakoș, , & Hint, M. Ș. (2019). The impact of big data analytics on company performance in supply chain management. *Sustainability*, *11*(18), 4864. doi:10.3390/su11184864

Pacaci, A., Zhou, A., Lin, J., & Özsu, M. T. (2017). Do we need specialized graph databases? Benchmarking real-time social networking applications. *Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems*, 1–7.

Pereira, C. R., & Da Silva, A. L. (2015). Key Organisational Factors to Building Supply Chain Resilience: A Multiple Case Study of Buyers and Suppliers. *Journal of Operations and Supply Chain Management*, *8*(2), 77–95. doi:10.12660/joscmv8n2p77-95

Pokornỳ, J. (2018). Functional querying in graph databases. *Vietnam Journal of Computer Science*, *5*(2), 95–105. doi:10.1007/s40595-017-0104-6

Ponomarov, S. Y., & Holcomb, M. C. (2009). Understanding the concept of supply chain resilience. *International Journal of Logistics Management*, *20*(1), 124–143. doi:10.1108/09574090910954873

Raghunath, K. M., & Devi, S. L. (2018). Supply Chain Risk Management: An Invigorating Outlook. *International Journal of Information Systems and Supply Chain Management*, *11*(3), 87–104. doi:10.4018/IJISSCM.2018070105

Raghunath, K. M., & Devi, S. L. T. (2018). Supply Chain Risk Management: An Invigorating Outlook. *International Journal of Information Systems and Supply Chain Management*, *11*(3), 87–104. doi:10.4018/IJISSCM.2018070105

Reuters Events. (2018). *47% of supply chain executives still rely on Excel says survey*. Retrieved from 47% of supply chain executives still rely on Excel says survey: https://www.eft.com/technology/47-supply-chain-executives-still-rely-excel-says-survey

Ribeiro, J. P., & Barbosa-Povoa, A. (2018). Supply Chain Resilience: Definitions and quantitative modelling approaches–A literature review. *Computers & Industrial Engineering*, *115*, 109–122. doi:10.1016/j.cie.2017.11.006

Robinson, I., Webber, J., & Eifrem, E. (2013). *Graph databases*. O'Reilly Media, Inc.

Saenz, M. J., Koufteros, X., Hohenstein, N.-O., Feisel, E., Hartmann, E., & Giunipero, L. (2015). Research on the phenomenon of supply chain resilience. *International Journal of Physical Distribution & Logistics Management.*

ShefaliPatil, G., & Bhatia, A. (2014). Graph Databases-An Overview. *1Student, ME Computers, Terna College of Engg. Navi Mumbai*, *2*, 657–660.

Supply Chain 247. (2018). *Isn't It Time to Finally Retire Your Supply Chain Spreadsheets?* Retrieved from Isn't It Time to Finally Retire Your Supply Chain Spreadsheets?: http://www.supplychain247.com/article/isnt_it_time_to_finally_retire_your_supply_chain_spreadsheets/news

Tan, K. H., Zhan, Y., Ji, G., Ye, F., & Chang, C. (2015). Harvesting big data to enhance supply chain innovation capabilities: An analytic infrastructure based on deduction graph. *International Journal of Production Economics*, *165*, 223–233. doi:10.1016/j.ijpe.2014.12.034

The Economist. (2009). *Moving on up: Is the recession heralding a return to Henry Ford's model?* Retrieved from Moving on up: Is the recession heralding a return to Henry Ford's model?: https://www.economist.com/node/13173671

Tjahjono, B., Esplugues, C., Ares, E., & Pelaez, G. (2017). What does industry 4.0 mean to supply chain? *Procedia Manufacturing*, *13*, 1175–1182. doi:10.1016/j.promfg.2017.09.191

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010). A comparison of a graph database and a relational database: a data provenance perspective. *Proceedings of the 48th annual Southeast regional conference*, 1–6. doi:10.1145/1900008.1900067

Vukotic, A., Watt, N., Abedrabbo, T., Fox, D., & Partner, J. (2014). *Neo4j in action*. Manning Publications Co.

Waller, M. A., & Esper, T. L. et al. (2014). *The definitive guide to inventory management: Principles and strategies for the efficient flow of inventory across the supply chain*. Pearson Education.

White, M. (2012). Digital workplaces: Vision and reality. *Business Information Review*, *29*(4), 205–214. doi:10.1177/0266382112470412

Worldfavor. (2020). *Do You Still Use Spreadsheets for Supply Chain Sustainability?* Retrieved from Do You Still Use Spreadsheets for Supply Chain Sustainability?: https://blog.worldfavor.com/do-you-still-use-spreadsheets-for-supply-chain-sustainability

Zare Mehrjerdi, Y., & Lotfi, R. (2019). Development of a mathematical model for sustainable closed-loop supply chain with efficiency and resilience systematic framework. *International Journal of Supply and Operations Management*, *6*, 360–388.

Zimon, D., & Madzík, P. (2019). Standardized management systems and risk management in the supply chain. *International Journal of Quality & Reliability Management*, *37*(2), 305–327. doi:10.1108/IJQRM-04-2019-0121