Supply network design with uncertain demand: Computational cooperative game theory approach using distributed parallel programming

Emre Tokgöz a,*, Sonia Mahjoub b,**, Tarik El Taeib c, Khalid Bachkar d

a Industrial Engineering, School of Engineering, Quinnipiac University, Hamden, CT, USA
b LEMNA, Oniris, France
c Department of Security Systems & Law Enforcement Technology, Farmingdale SUNY State College, Farmingdale, NY, USA
d Department of International Business and Logistics, California State University Maritime Academy, CA, USA

ARTICLE INFO

Keywords:
Cooperative Game Theory
Core
Mathematical Programming
Supply Chain Management
Distributed Parallel Programming

ABSTRACT

In this work we investigate a supply chain design problem with uncertain final demands for the end products produced by a set of manufacturers. This network of manufacturers has the organizational decisions made internally by pooling resources in a cooperative manner and externally by determining the dominant strategic actors that are characterized by the retailer. A non-linear production game (NLPG) is formulated as a mathematical programming problem to describe coalition formation among the manufacturers based on initial contractual agreements. We show that NLPG is a grand coalition game when demand distribution has an increasing generalized failure rate. The conditions that impact the profit allocation in the game including core set, fairness, stability, superadditivity, least-core and ϵ-core are defined. A core allocation solution is generated by using an algorithmic approach. This algorithmic solution is tested in a distributed work station network and resulted in attaining strong computational results for the proposed mathematical programming problem; optimization results with 1000 players are determined in 6 min 40 s on a university computer network using parallel programming with 52 powerful work stations. The numerical results indicated O(log(−N)) complexity of the algorithmic solution up to 800 players. To the best of our knowledge, this work is the first of its kind in which distributed parallel processing is implemented in a university work station network with distributed parallel programming and processing for solving a production cooperative game.

1. Introduction

Supply chain stability and performance are challenged by intensive competition, products with short life cycle, environmental and social pressure and development of intelligent internet technologies. In this complex context, the supply chain network design is a key factor to cope with this new reality. According to Simchi-Levi et al. (2000), supply network design is the most basic decision of supply chain management, which impacts all other decisions concerning supply chain and has the most extensive effect on the chain’s return on investment and its overall performance.

This strategic decision (network design) which includes multi-decision makers is specifically addressed by Game.

Theory to predict the rational strategic behavior of individuals in competing or cooperating scenarios (Salcedo-Díaz et al., 2021; Drechsl, 2010; Leng and Parlar, 2005). The field of game theory can be broadly divided into cooperative game (all players share a common objective) and non-cooperative game theories (each player has an individual objective that usually will not coincide with the objective of the other players) (Salcedo-Díaz et al., 2021; Zamarripa et al., 2012). Arguably, non-cooperative game theory has been used extensively to model competition and coordination among players of a supply chain. Given the increasing supply chain costs, many articles used concepts and methods from non-cooperative game theory to model and solve certain problems such as inventory management (Halat and Haferzalkotob, 2019; Arda and Hennet, 2008; Cachon and Zepkin, 1999) and supply chain planning/pricing (Zamarripa et al., 2012; Leng and Parlar, 2010). Nevertheless, the main concern regarding of the application of non-cooperative game theory to supply chain network is whether using Nash and Stackelberg equilibrium concepts would provide a solution.
areas of applications in the literature; In particular, collaboration through sharing resources is a key decision to getting close to the customer and rapidly fulfilling variable market demands (Camarinha-Matos et al., 2009). Manufacturers’ network partners can share physical resources such as machine tools (Li et al., 2015) as well as virtual resources, such as data (Wang et al., 2019) and knowledge (Gao and Nee, 2017). Cooperation of manufacturers is investigated by Sandberg (2007) within Swedish manufacturing companies and their supply chains and determined that the increased levels of collaboration resulted in increased success in which case cooperative actions are defined to be information sharing type. Many firms establish cooperation and form Research Joint Venture (RJV) to promote R&D investment and information sharing and reduce the dramatic costs brought by R&D (Wu et al., 2021). Similarly, resource sharing for research and development as a part of organizations’ joint product development had shown to succeed for manufacturers by Garrette et al. (2009). The advantages of this kind of collaboration are outstanding, especially in high-tech firms. It helps firms to crack new markets, gain skills, and realize economies through reorganization and exploitation of complementarities, share costs and risks, and control competitive forces (Veugelers, 1998; Ge and Hu, 2008). From logistics performance perspective that also relate to manufacturing, Leitner et al. (2011) and Bahrami (2002) had shown the positive impact of resource sharing in industrial applications of logistics and how much it positively influences performance as a result of the corresponding performance indicators’ analysis. Positive outcomes on resource sharing across different partners are reported by Botsman et al. (2006) for mobility and housing and by Schaefer (2012) for car sharing services in Western Europe and North America and transportation logistics. For instance, civil airlines realized strategic alliances in order to increase their flight network and to enhance productivity as reported by Oum et al. (2004) that resulted in reduction of the number of journeys, transportation costs, and CO2-emissions (Cruijssen et al., 2007). In supply chains, the resources shared helped with productivity and profitability, and added value to creating activities to designing, manufacturing, marketing, sales, logistics, and services within the organization Oum et al. (2004). Within supply network design literature, cooperative game has received a great attention in the last two decades. For instance, Sobhan et al. (2019) have introduced a Sharing Economy-based cooperative platform with the aim of enabling organic small holders to overcome their specific set of challenges by sharing resources and aggregating peer-to-peer activities; The authors used a two-stage game theory modelling approach and jointly examined the production-inventory planning and pricing problems of multiple competing organic food supply chains (OFSCs) in two alternative scenarios: non-cooperative and cooperative game. Their various revealed that the Sharing Economy-based cooperative game theory approach yields a higher total profit compared with the sum of individual organic food supply chains profits achieved in the non-cooperative game theory scenario. This suggests that SE-based mechanism enables OFSCs to achieve greater financial gains and concurrently contribute toward sustainable development goals in developing countries due to its underlying innovative peer-to-peer sharing features. Hartman and Dror (2005), Chen and Zhang (2006), and Ben-Zvi and Gerchak (2006) have applied cooperative game theory to model inventory centralization of several retailers facing random demand. Guardiola et al. (2007) formulated a supply chain model composed of one supplier and several independent retailers as a cooperative game and showed that cooperation between retailers can increases their profits by obtaining a lower wholesale price from the supplier. Moreover, in the shipping industry, cooperative game theory tools are more suitable for modeling cooperation among liner companies that are engaged in cooperative agreements and alliances. Specifically, shipping alliances can help carriers reduce the cost associated with bunker costs and port charges and therefore provide low prices and broad service coverage through economies of scale and economies of scope (Notteboom et al., 2021).

Within a strategic situation involving coalitions, insuring stability to coalition partners was the central focus of supply network members. Arguably, cooperative game theory provides different solution concepts to determine what payoff is required to incite a player to collaborate with supply chain partners. Among these solutions proposed in the era of payoff allocations, the core, the Shapley Value (Shapley, 1953) and the Owen set (Owen, 1975) take their places. In the area of cooperation among supply chain partners, many studies dealt with the allocation of final payoffs awarded to each player in a fair manner. Drechsel and Kimms (2010) proposed an iterative row generation procedure based on mathematical programming techniques to compute a core element with a large number of players. The performance of the core element algorithm was tested in a procurement cooperative game with 150 players. Hennet and Mahjoub (2010) developed a technique to construct a stable and fair allocation scheme in the case of Linear Production Game (LPG). Using the LPG framework, the authors explored the existence of fair allocation of profit among a coalition of 10 partners who are pooling their resources. Frisk et al. (2010) developed the Equal Profit Method (EPM) as a new approach ensuring a profit allocation that provides as equal relative profit as possible among the participants. The proposed method provided stable cost allocations for the instances analyzed and these allocations are interesting as a basis for sharing costs or distributing savings (Frisk et al., 2010). Following the EPM approach developed by Frisk et al. (2010), Zheng et al. (2019) provided a new profit allocation scheme that minimizes the maximum pairwise satisfaction difference. The issue of fairness in payoff allocation is also studied in Mohr and Lin (2015) by introducing a criterion named Shared Capacity to resolve possible conflicts among network members of 30 suppliers. The proposed suppliers’ cooperation algorithm leads to both reliable and high level performance and enables suppliers to self-organize into independent disjoint coalitions and distribute obtained profit in a fair manner. However, the nature of the problems are usually NP-hard in this area of applications for the formulated problems; therefore, cloud computing (CC) was introduced recently as a new computational technique to determine supply chain solutions. In fact, most of the literature surrounding the CC use in supply chain postulates its significant impact on collaboration and coordination (Subramaniam and Abdurahman, 2017; Duan and Liu, 2016), resource allocation (Tao et al., 2014; Thekinen and Panchal, 2017), manufacturing integration (Lu and Xu, 2017; Brant and Sundaram, 2015) and resource sharing and production flexibility (Wang and Xu, 2013; Paniti, 2014; Liu et al., 2014). Hence, the aim of this article is to determine a solution to a complicated supply chain design problem in which the manufacturer’s network offers a wholesale price to an autonomous retailer facing uncertain final demands. In this game, the network of manufacturers pool their resources cooperatively and act as dominant strategic actors relatively to their retailer. We introduce a new non-linear mathematical model of the manufacturing network design named Non-Linear Production Game (NLPG). This model is an extension of the classical LPG game studied in Owen (1975). LPG game has a linear objective function with linear constraints while NLPG game has nonlinear objective function and linear constraints. In the Stackelberg game between manufacturers’ coalition and the retailer, the integration of the reaction function
of the follower (the retailer) in the objective function transforms linearity into nonlinearity. It is shown that the NLPG is a grand coalition game when demand distribution has an increasing generalized failure rate. Profit allocation in the designed NLPG game is determined by using the cost percentage commitment of each manufacturer’s contractual agreement. We show that this cost distribution insures superadditivity property and non-emptiness of the core for the profit allocation. To the best of our knowledge, this work is the first of its kind in which a core allocation solution is implemented by using distributed parallel processing for solving a production cooperative game with the corresponding algorithm. This solution can be tested in any connected network of work stations and we are choosing a university network in particular for attaining the numerical results. The software application architecture consisted of three-tier layers namely service, server and client layers in a network of 52 workstations, each having 32 cores (we note that a typical computer used for computational purposes has 3–7 cores in general). This methodology along with the designed algorithm had outstanding numerical performance and consistency when compared to the classic software application approaches. Hence, the main contribution of this study is the use of parallel programming in the supply chain framework to reduce computational time to attain solutions for the NLPG mathematical formulation that is known to be NP-hard. While supply chain solutions are attained by using computer networks, distributed computing, and cloud computing in the literature (see Sohrabi and Argomi, 2020 for a review), the algorithm we developed for distributed computing along with the network design distinguishes itself from the designed networks in the corresponding literature for solving large scale problems in real life applications. The distributed computing algorithm allowed computations by using all cores bit-by-bit with no time wasted during the computing process that speed up the process with the designed network of 50 work stations that had 32 cores. Our algorithmic approach reduces the computational complexity of the NLPG problem’s solution for a number of players. The performance of the proposed algorithm is tested for different combination of cooperative games with large number of players and products, leading to strong computational results when compared to other algorithmic solution methods used in the literature for cooperative games in supply chain (i.e., Drechsel and Kimms, 2010 -N|=150; Hennet and Mahjoub, 2010 -N|=10; Mohebbi and Li (2015) -N|=30). For this manufacturers’ cooperation algorithm, optimization results for a game up to 1000 players are tested and the corresponding solution is attained in 6 min and 40 s. The numerical results indicated O(log -N) complexity of the algorithmic solution up to 800 players and O(N) complexity of the algorithm when the number of players range from 800 to 1000. It is worth noting that the NP-hard nature of the problem does not change, however the proposed algorithm improves computational effectiveness of the complicated problem, indicating computational strength of our work in the supply chain framework. The type of production games with contractual agreements considered in this study exists in many real-life supply chain agreements. An example is Boeing company’s subsidiaries for aircraft development.

The remainder of this paper is organized as follows. In Section 2, the structure of the supply network is presented with the corresponding notation to be used throughout the paper. In Section 3, the non-cooperative game stage is covered with uncertainty in demand and the rules on the manufacturer’s network game. In Section 4, the coalition game with the characteristic function of the nonlinear production game (NLPG) and the core set are introduced. In Section 5, a model for the NLPG with cost commitment contract is introduced with the corresponding core set, fairness and stability conditions along with a numerical example. Algorithmic solution approach and computational complexity results are covered in Section 6. In Section 7, distributed network solution and numerical experimentation are presented. Finally, concluding remarks and future work are presented in Section 8.

2. The supply network presentation

Consider the supply chain design presented in Fig. 1 that consists of a network of manufacturers and a retailer facing an uncertain demand for each final product i with i ∈ {1, … , n}.

The manufacturers’ network contains N firms that are willing to cooperate by sharing resources (manufacturing plants, machines, work teams, robots, pallets, storage areas, etc) in order to produce various commodities and sell them in a market. All the firms of the manufacturers’ network compete to be partners in a coalition S ⊆ N that defines a supply chain.

Acting as an intermediate party between the manufacturers’ network and the final consumers, the retailer sells the products to the final consumers at a fixed price. Demands for products are uncertain and assumed mutually independent. One of the strengths of this study is determining an optimal profit allocation for supply chain partners by incorporating contractual agreement among them based on the cost distribution.

In order to formulate the supply network design problem, the following notations are used:

\begin{align*}
J & : \text{ The set of manufacturers or enterprises} \\
R & : \text{ The retailer} \\
r & : \text{ The set of resources with limited capacity} \\
B_r & : \text{ The amount of resource } r \text{ available at manufacturer } j \\
A_r & : \text{ The amount of resource } r \text{ necessary to produce one unit of product } i \\
x^M & : \text{ The profit of the manufacturers’ network} \\
x^R & : \text{ The profit of retailer}
\end{align*}

(continued on next page)
3. The non-cooperative game stage

At this stage, Stackelberg game is used for formulating the coalition between the manufacturers and the retailer under uncertain demand. In this game, we suppose that the manufacturers’ coalition acts as a leader and the retailer acts as a follower. The Stackelberg game with a dominant manufacturer is widely used in the areas of supply chain coordination, inventory management, and others. In this context, Hua et al. (2011) modeled the product design problem in a two-stage supply chain that includes one manufacturer and one retailer. This problem is formulated as a manufacturer-dominant Stackelberg game in order to derive the optimal design strategy for the manufacturer. Rahmani and Yavari (2019) considered a dual-channel supply chain that consists of one manufacturer which produces a new kind of green product and one retailer. The manufacturer sells the products to the retailer and directly to the customers. In the proposed model, the manufacturer acts as a Stackelberg leader and declared the wholesale price and green degree. Then, the retailer reacts to determine the selling price. Also, He and Zhou (2019) investigated a dual-channel green supply chain with a dominant manufacturer. As a follower, the retailer developed a promotion policy in order to cope with competition of direct channel. In the real world of supply chain, the growing role of the Internet in trade activities has prompted many manufacturers such as IBM, Pioneer Electronics, Nike, Dell and Estman to redesign their traditional channel structures through establishing direct sales channel (Rahmani and Yavari, 2019). Finally, Kwong et al. (2021) studied the Stackelberg game between the manufacturer who intends to develop one or more new products and a retailer. In this game, the manufacturer acts as leader in the new contract with the retailer who is already selling products provided by the manufacturer. The applicability of this game is tested on a case study of smartphone design. In the current research, the manufacturers’ coalition impose a take-it-or-leave-it wholesale price contract to the retailer. Then, the retailer reacts by fixing the optimal order quantity. It is assumed that the retailer demand agrees to conclude any contract that guarantees an expected profit greater than his opportunity cost in which is set to be equal to zero by convention.

3.1. The retailer’s game under uncertain demand

For each final product sold on the market, the retailer faces an uncertain demand. Considering the retailer quantity order \( y_i \) for \( i = 1, \ldots, n \), the actual sales of a product \( i \) with given \( y_i \) and \( D_i \) is equal to \( \min(y_i, D_i) \). \( D_i \) is modeled over the reference period as a random variable with a continuous distribution function \( F(y_i) \) and a density function \( f(y_i) \). Then, the expected profit of the retailer over the reference period is

\[
\pi^R = \sum_{i=1}^{n} p_i E(\min(y_i, D_i)) - w_i y_i \quad \text{when } w_i \leq p_i
\]

where

\[
E(\min(y_i, D_i)) = \int_{y_i}^{\infty} x f(x) dx + \int_{0}^{y_i} y f(x) dx
\]

\[
= \int_{y_i}^{\infty} x f(x) dx + y_i F(y_i)
\]

with \( F(y_i) = 1 - F(y_i) \) satisfied. In this case, the retailer expected profit formula can be expressed by using the formula

\[
\pi^R = - \sum_{i=1}^{n} w_i y_i + \sum_{i=1}^{n} p_i \int_{y_i}^{\infty} x f(x) dx + \sum_{i=1}^{n} p_i y_i F(y_i) \tag{2}
\]

The optimality condition of retailer’s expected profit has the following form:

\[
\frac{\partial \pi^R}{\partial y_i} = - w_i + p_i (1 - F(y_i)) = 0
\]

Noting \( \frac{\partial \pi^R}{\partial y_i} = -p_i f(y_i) \leq 0 \), the problem is strictly concave and admits a single optimal solution. For each product, the optimal retailer’s order quantity is

\[
y^*_i = F^{-1}\left[ \frac{p_i}{p_i} w_i \right] \tag{3}
\]

3.2. The manufacturing network game

The supply network incurs manufacturing costs.

\[
c = (c_1, c_2, \ldots, c_n)^T \quad \text{per unit of each final product and supplies the retailer at the wholesale price vector } w = (w_1, \ldots, w_n)^T.
\]

The resource capacity constraints are expressed by the inequality

\[
A y \leq B e \quad \text{if } e \text{ is a characteristic vector of coalition } S \text{ such that } e_j = \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{if } j \not\in S \end{cases}
\]

Assuming we have a wholesale price contract, the manufacturers’ network acts as the Stackelberg leader and anticipates correctly the retailer’s reaction function. The manufacturers’ network faces the demand curve \( y(w) \) and determines \( w^* \) to maximize its profit formulated by solving the problem

\[
\max_{y \in \mathbb{R}^n} (w - c)^T y(w) \quad \text{subject to}\n\]

\[
y \in \mathbb{R}^n, e_S \in (0, 1)^S
\]

The manufacturers’ network profit is deterministic. It knows the retailer’s order for any price and it isn’t responsible for unsold products. All the uncertainty is supported by the retailer. At this stage, two different problems must be solved: the strategic problem of selecting the wholesale price vector \( w \) and the cooperative problem of optimizing the production vector \( y \) and the coalition characteristic vector \( e_S \). As pointed out by Lariviere and Porteus (2001), by using Eq. (3) we derive

\[
w_i(y_i) = p_i F(y_i) \quad \text{for } i = 1, 2, \ldots, n
\]

For a given vector \( w \), the problem (6) characterizes a cooperative game, named the Linear Production Game (LPG) studied in Owen (1975), Osborne and Rubinstein (1994), Hennet and Mahjoub (2010). The game we consider in this paper assumes \( w \) as the decision variables with optional values related to the optimal output values \( y_i \) through constraints of (6).

By substituting Eq. (7) into the objective function of problem given in (6) we obtain the following non-linear programming problem:

\[
\max_{y \in \mathbb{R}^n} \sum_{i=1}^{n} (p_i F(y_i) - c_i) y_i \quad \text{subject to}
\]
Characteristic function of the Nonlinear Production Game (NLPG)

In the context of a wholesale price contract, the manufacturers’ network can anticipate the retailer’s reaction function and integrate it in its profit function formulated as a mixed-linear programming problem. This fact leads to a new kind of production cooperative game named NLPG by using problem given in (8) that will be explained in this section.

The first-order condition for the unconstrained manufacturing network’s profit function is

$$\frac{\partial \pi(y)}{\partial y_j} = p_j F(y_j) \left(1 - \frac{y_j f(y_j)}{F(y_j)}\right) - c_i$$

where the term $g(y) = \frac{y f(y)}{F(y)}$ represents the generalized failure rate GFR that has an economic interpretation. The GFR measures the demand elasticity and gives the percentage decrease in the probability of a stock out from increasing the stocking quantity by 1% (Lariviére, 2006). A distribution has an increasing generalized failure rate (IGFR) if $g(y_i)$ is weakly increasing for all $y_i$ such that $F(y_i) < 1$.

In this context, this study considers the NLPG with demand distribution characterized by increasing generalized failure rate (IGFR). This class of distribution captures most common distributions like normal, exponential, gamma and weibull distributions. This assumption guarantees unimodality of the manufacturers’ network profit’s criterion.

In this cooperative game, each coalition of enterprises $S \subseteq N$ characterized by the vector $e_S$, has a value function $v(S)$ defined as the maximal profit that can be attained by this coalition. It is obtained as the solution of the problem given in Eq. (10) similar to problem stated in Eq. (8) except the fact that the vector $e_S$ turns into a fixed characteristic vector of the investigated coalition $S$, and not a vector of decision variables.

$$\max_{y_S} \pi(y) = (w - c)^T(y(w)$$

subject to

$$Ay - Be_S \geq 0, \quad y \in \mathbb{R}^n_+,$$

$$e_S \in \{0, 1\}^N$$

with $e_S$ given.

Property 1. The NLPG with IGFR is a grand coalition game

Proof. The characteristic vector of the grand coalition $N$ is $e_N = [1, \ldots, 1]^T$. The matrices $A$ and $B$ in (10) are componentwise nonnegative. For any set $S \subseteq N, e_S \leq e_N$ and $Be_S \leq Be_N$. Then, the optimal solution of (8) is feasible for (10) and the maximum expected profit, denoted by $v'$, can be obtained as the optimal solution of (10) defined by

$$\max_{y_S} \pi(y) = (w - c)^T(y(w)$$

subject to

$$Ay - Be_S \geq 0, \quad y \in \mathbb{R}^n_+,$$

$$e_S \in \{0, 1\}^N$$

Due to this property, the optimal global profit can be calculated by solving (10) characterized by continuous variables instead of the mixed-integer nonlinear programming problem (8).

Property 2. The NLPG with IGFR is non-convex.

Proof: Consider the following example with three manufacturers and one retailer who faces an independent uncertain demand: $D_1 \sim N(10; 4.5)$ and $D_2 \sim N(10; 5.75)$. In order to run the NLPG model, we propose the following numerical data:

$$A = \begin{bmatrix} 10 & 5 \\ 7 & 8 \end{bmatrix}, \quad B = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 25 & 25 \end{bmatrix}, \quad c = \begin{bmatrix} 50 \\ 50 \end{bmatrix}$$

The optimal profit $v' = 269.031$ for $y' = [2.82, 3.78]$ is generated by the grand coalition $N = \{1, 2, 3\}$. The profits of all valid coalitions are given in (12).

$$v_1 = 155.25, \quad v_2 = 155.25, \quad v_3 = 269.031$$

For $S = \{1, 2\}$ and $T = \{2, 3\}$, the non-convexity of the NLPG can be proven by using the fact that $269.031 = v_{1,2} \leq v_{1,1} + v_{1,3} - v_1 = 310.5$

4.2. The core of NLPG

A profit allocation method that splits the total profit, $v(N) = v'$, among the firms $j \in N$ is said to be efficient, that is $\sum_{j \in S} u_j = v'$ where $u_j$ is the profit allocated to firm $j$. A profit allocation is called individually rational if $u$ satisfies $u_j v'(j)$. The core of the game is defined by Gillies (1959) as those profit allocations, $u$, that satisfy the following:

$$\text{Core}(N, v) = \left\{ u \left| \sum_{j \in S} u_j = v' \quad \text{and} \quad \sum_{j \in N} u_j \geq v' \quad \forall \ S \subseteq N \right. \right\}$$

This implies that any coalition of firms should receive a profit that is higher than the coalition members acting alone. A profit allocation in the core is said to be stable.

The characteristic function of the NLPG can be expressed by the formula

$$\max \sum_{j \in S} (p_j - c_j)y_j, \quad s.t. Ay - Be_S \geq 0$$

$$= \max \sum_{j \in S} (p_j - c_j)y_j, \quad s.t. Ay - Be_S \geq 0$$

$$\max \sum_{j \in S} (p_j - c_j)y_j, \quad s.t. Ay - Be_S \geq 0$$

In the cases of $F(y) = 0$ and $F(y) = 1$, the NLPG game transforms to the LPG game developed by Owen (1975) and has a non-empty core. We
introduce a modification of the problem given in (14) in the next section to avoid the core emptiness case and incorporate fairness condition.

5. The NLP with cost commitment contract

5.1. Model

A cost associated strategy is introduced by Wu and Chang (2020) with the cost of manufacturing to be the ratio of the manufacturer’s accepted price to the supplier’s offer price as the criterion for the scoring, and we are introducing a similar cost strategy to their cost strategy. The idea behind our approach is to determine cost distribution based on the cost percentage shared by the players for the product to be produced. We consider the cost percentage distribution parameters $a_{ij}$ of all players $j \in S$ that take place in producing each item $i$ such that $1 \leq i \leq n$. These percentages are based on the contractual agreement of the players and this distribution is going to be used for determining the profit distribution to each player. The player’s actual contract cost for each item $i$ divided by the average of the total of the costs shared by all the players in the contract gives the profit share of player $j$ for producing item $i$. Therefore, we assume

$$a_{ij} = \frac{cc_{ij}}{\sum_{j=1}^{n} cc_{ij}} \quad \forall \ j \in N \text{ and each } i \text{ such that } 1 \leq i \leq n$$

(15)

where $cc_{ij}$ is the contract cost of player $j$ in the game for producing item $i$. For example, if the initial distributed contract cost for item $i$ (i.e. $i = 1$) of the players are $cc_{11} = 360, cc_{12} = 900$, and $cc_{13} = 540$ then $a_{11} = 20\%$, $a_{12} = 50\%$, and $a_{13} = 30\%$ are the corresponding profit distribution elements respectively for players $\{1,2,3\}$ that invested in the cost of producing item $1$ and did produce item $1$. The same logic applies to all items $i, 1 \leq i \leq n$. In addition, it is easy to see that

$$\sum_{i=1}^{n} \sum_{j \in N} a_{ij} = \sum_{i=1}^{n} \sum_{j \in N} \frac{cc_{ij}}{\sum_{j=1}^{n} cc_{ij}} = \sum_{i=1}^{n} 1 = |N|$$

(16)

noting that there are $|N|$ number of players in the game and

$$\sum_{i=1}^{n} \left( \frac{cc_{ij}}{\sum_{j=1}^{n} cc_{ij}} \right) e_i = 100\% \quad \text{for each } j.$$

If $j$ is a player in the game for production of $i$ items then its profit share will be $100\%$ for production of all items. Eq. (15) indicates that there are $|N|$ number of players in the game for producing $i, 1 \leq i \leq n$, number of items that take place in production; In this equation the profit distribution to each player is $100\%$ for production of all $n$ items therefore the resulting outcome is $|N|$. The profit share of player $j$ for producing item $i$ in the game is calculated by multiplying profit per item $pF(y_{e}) - c_{i}$, the quantity of item $i$ sold, $y_{e}$, the profit percentage distribution of item $i$ to player $j, a_{ij}$, and the characteristic coalition vector element $e_{j}$; therefore the NLP model in (14) can be modified as follows:

$$\begin{align*}
\text{max} & \sum_{i=1}^{n} \left( p_{i} - c_{i} \right) y_{i} a_{ij} e_{j} \quad \text{s.t.} A_{ij} \leq B_{ij} \text{if } F(y_{e}) = 0 \\
\text{max} & \sum_{i=1}^{n} \left( p_{i}F(y_{e}) - c_{i} \right) y_{i} a_{ij} e_{j} \quad \text{s.t.} A_{ij} \leq B_{ij} \text{if } F(y_{e}) \in (0, 1) \\
\text{max} & \sum_{i=1}^{n} \left( -c_{i} \right) y_{i} a_{ij} e_{j} \quad \text{s.t.} A_{ij} \leq B_{ij} \text{if } F(y_{e}) = 1
\end{align*}$$

(17)

Eq. (17) differs from Eq. (14) and it depends on the contract cost responsibility of the player and the player’s involvement in the production of item $i$ and the corresponding $y_{e}$. Multiplication of the model in (14) by $e_{j}$ to derive the model in (17) is to ensure that the player $j \in S$ in the game will receive the profit credit $a_{ij}$ if the player $j$ takes place in the coalition $S$ (i.e. $e_{j} = 1$) for producing item $i$ and respecting the constraint $A_{ij} \leq B_{ij}$. Otherwise, the player will not get any credit for being a part of the game.

Property: The NLP with cost commitment contract is superadditive and therefore satisfy the condition

$$v(S) + v(T) \leq v(S \cup T)$$

whenever $S, T \subseteq N$ satisfy $S \cap T = \emptyset$ for the characteristic function $v$. This inequality simply means that the value of a union of disjoint coalitions is no less than the sum of the coalition’s separate values.

Proof. We assume $j_{1}$ is a player in set $S$ and $j_{2}$ is a player in $T$ such that $j = j_{1} \cup j_{2} \subseteq S \cup T$ and $S \cap T = \emptyset$ for proving superadditivity. The profit percentage of player $j_{1}$ in the coalition for producing item $i$ is represented by $a_{ij_{1}}$, and $a_{ij_{2}}$ represents the profit percentage allocated to player $j_{2}$ for producing item $i$ in the coalition. Therefore, these two profit distributions are independent from each other. Following these assumptions we have

$$v(S) + v(T) = \sum_{i=1}^{n} a_{ij_{1}} e_{j_{1}i} + \sum_{i=1}^{n} a_{ij_{2}} e_{j_{2}i} = \sum_{i=1}^{n} a_{ij} e_{ji}$$

In this case

$$a_{ij} = a_{ij_{1}} + a_{ij_{2}} \quad \text{for } j \in S \cup T$$

and each $i$ such that $1 \leq i \leq n$, noting that the profit allocation to the players $j_{1}$ and $j_{2}$ are independent from each other based on the contract, therefore their profit distributions $a_{ij_{1}}$ and $a_{ij_{2}}$ in the coalition would be independent from each other. The involvement of a player $j$ in a game depends on the $e_{j}$ value which is also a part of the resource constraint $A_{ij} \leq B_{ij}$. Next, we consider two cases related to the characteristic coalition vector that will be used in the proof of superadditivity condition in this section.

Case 1: If $j_{1} \in S$ then $j_{1} \in S \cup T$ indicating $e_{j_{1}i} = e_{ji} = 1$ holds naturally noting that we have $e_{ji} = 1$ if player $j_{1}$ is in the game and

$$1 = e_{ji} \leq e_{ji} = e_{ji} = 1$$

Case 2: If $j_{2} \in T$ then $j_{2} \in S \cup T$ indicating $e_{j_{2}i} = e_{ji} = e_{ji} = 1$ holds naturally noting that we have $e_{ji} = 1$ if player $j_{2}$ is in the game and

$$1 = e_{ji} \leq e_{ji} = e_{ji} = 1$$

(19)

Both cases indicate that the set of two players involvement in the game (i.e. $e_{ij_{1}j_{2}} = 1$) is always stronger than the individual involvement of player $j_{1}$ and $j_{2}$ in $S$ and $T$ noting that $S \cap T = \emptyset$. The superadditivity condition holds for $v$ by using Equations (18) and (19)

$$v(S) + v(T) = \sum_{i=1}^{n} a_{ij_{1}} e_{j_{1}i} + \sum_{i=1}^{n} a_{ij_{2}} e_{j_{2}i}$$

and using the general mathematical inequality
\[ \sum_{k \in S} a_k + \sum_{j \in S} b_j \leq \sum_{k \in S} (a_k + b_k) \]

5.2. Core conditions

Suppose we consider the case \( 0 < F(y_1) < 1 \). The core condition

\[ v(S) = \sum_{j \in S} \nu_j \sum_{e_j} v(S) | e_j = \sum_{j \in S} u_{i,j} \]

holds because the profit \( v_{ij} \) allocated to player \( j \) on the sale of item \( i \) at the end of the game is no more than the contractual percentage \( a_{ij} \) of the average of the total profit \( v(S) \) distributed to player \( j \) in \( S \) (i.e. \( v_{ij} \leq \alpha v(S) \)).

for all \( j \in S \) due to the contract. Therefore the total of the profit allocations \( u_{i,j} \) for all items \( i \) and all \( j \in S \) (i.e. the player \( j \) receiving \( a_{ij} \) portion of the total profit \( u_i \) at the end of the contractual agreement) is no less than the total profit \( v(S) \) that the players in \( S \) can receive based on the contract.

We assume \( u_{ij} = u_{i,j} \) for player \( j \) to receive the corresponding profit on the production of item \( i \). The percentage distribution values of \( a_{ij} \) for all players \( j \in S \) on each item \( i \) to be produced can be chosen in such a way that the game’s profit distribution can change if the players decide to change the rules of the game in the contract. Therefore, the proposed profit distribution is flexible if the rules of the game changes based on the cost at any given time. In addition, the constraint \( Ay \in B_e \) determines the involvement of players in set \( S \) which is reflected to the objective function in Eq. (17) by multiplying with the factor \( \xi_j \). For instance, if all the players’ profit share is equally distributed (i.e. \( u_i = \frac{v(N)}{n} \)) to all items \( i \) and if this profit is modified to include cost distribution function \( a_{ij} \) (i.e. player \( j \) receives \( a_{ij} \) as the profit amount for producing item \( i \) then the game would have

\[ \sum_{i \in I} a_{i,j} = 1 \text{ for each } j \text{ and } u_i = \frac{v(N)}{|N|} \]

with all the players involvement in the game (i.e. \( \xi_j = 1 \) for all \( j \in N \)) that would satisfy the second core condition

\[ \sum_{j \in S} \sum_{i \in I} u_{i,j} = \sum_{j \in S} \sum_{i \in I} u_{i,j} \xi_j = \sum_{j \in S} \sum_{i \in I} \frac{v(N)}{|N|} \xi_j \]

The profit distribution \( a_{ij} \) can be determined in such a way that the corresponding values can be entered in the contract.

5.3. Least- and \( \epsilon \)-core sets and solutions

The \( \epsilon \)-core and least-core set concepts are closely related to the problem introduced in this work. \( \epsilon \)-core set concept introduced by Shapley and Shubik (1966) is similar to the core-set concept that requires adjustment of the core-set definition by a parameter of \( \epsilon \in \mathbb{R} \) and defined as follows:

\[ \text{Core}_\epsilon(N,v) = \left\{ u \mid \sum_{j \in S} u_j = v \text{ and } \sum_{j \in S} u_j \geq v + \epsilon \forall S \subseteq N \text{ and } S \neq \emptyset \right\} \]

It is easy to see that the core-set is attained when \( \epsilon = 0 \). \( \text{Core}_\epsilon \) set is essentially useful when the solution to the problem is not in the core set and it requires a cost adjustment of \( \epsilon \) to redefine and solve the problem:

The new solution in the \( \text{Core}_\epsilon \) set of the game is feasible within the newly defined set \( \text{Core}_\epsilon \). This means that the grand coalition will remain stable in the case when the shared cost increases by an amount of \( \epsilon \) for which the solution is in the \( \text{Core}_\epsilon \) set. The grand coalition will be stable as long as the contract terms are redefined and the shared costs are distributed according to the adjusted rules of the game. \( \text{Core}_\epsilon \) set provides flexibility in the game to the players to adjust the cost that might be required for the game in the way the shared costs incur. The profit is re-distributed to the players by finding a solution in the \( \text{Core}_\epsilon \) if there exists no such feasible solution in the \( \text{Core} \) based on the initial contract cost distribution.

The least core concept is defined by Maschler et al. (1979):

\[ \text{Core}^\epsilon = \min \{ \epsilon : \text{Core}_\epsilon (v) \text{ is non – empty} \} \]

Least core represents the set by which the minimum possible \( \epsilon \) value for \( \text{Core}_\epsilon \) is attained to be as close as possible to the core-set. \( \text{Core}_\epsilon \) and \( \text{Core}^\epsilon \) need to be calculated in case the original game conditions do not change with the change in the contract. \( \text{Core}_\epsilon \) and \( \text{Core}^\epsilon \) require to add the constraint

\[ \sum_{j \in S} u_{i,j} \geq v_j - \epsilon \]

to the constraints of Eq. (17) while \( \text{Core}^\epsilon \) requires the additional objective min\( \epsilon \).

A numerical example is demonstrated in Section 5.5 as a part of the numerical example given for this section.

5.4. Fairness

In a coalitional game, the conventional profit allocation solution must be in the core set and satisfy fairness property for ensuring coalition stability and secure the members involvement in the coalition. Brink (2001) demonstrated that Shapley Value based on a marginal contribution concept is a fair solution satisfying the fairness condition “if to a game we add another game in which two players are symmetric then their payoffs change by the same amount”: Hennet and Mahjoub (2010) proposed a technique ensuring stability and fairness to the competitive solution developed by Owen (1975), Mohebbi and Li (2015) suggested a Shared Capacity Index criterion as a fair solution. Mohammaditabar et al. (2016) and Lozano et al. (2013) indicated that the least-core method (min–max core) is a fair imputation. Frisk et al. (2010) proposed a core allocation solution that satisfies fairness in a way that the participants’ relative profits are as equal as possible. The fairness condition in this case requires the following profit distribution for each item

\[ v(H) = \sum_{j \in S} v_{ij} \sum_{e_j} \frac{v(S)}{|S|} \xi_j \frac{v(S)}{|S|} |H| \]

hence we have

\[ \frac{v(H)}{|H|} \leq \frac{v(S)}{|S|} \iff v_{ij} = u_{i,j} = \frac{v(S)}{|S|} a_{ij} \]

for each item \( i, 1 \leq i \leq n \), and all \( j \in H \subseteq S \subseteq N \). Our fairness condition
indicates that the average profit share \( \frac{\Pi}{|N|} \) should be multiplied by \( a_{ij} \) to determine the profit \( \nu(H) \) of player \( j \) for producing item \( i \). This result is similar to the fairness solution attained by Drechsel and Kimms (2010) for the cost minimization model in the supply chain. We calculate the fairness of profit distribution based on items produced and the players’ cost percentage contribution for producing the item. In particular, if all players are assumed to have equal profit distribution then all the profits will be distributed for all items.

In the case when \( F(y_j) = 1 \) and \( F(y_i) = 0 \), the calculations for the case \( 0 < F(y_j) < 1 \) can be modified by choosing \( F(y_i) = 1 \) and \( F(y_j) = 0 \) by placing these values in the inequalities above.

5.5. Numerical Example

Suppose a cooperative game players signed a contract that determined the following costs at the beginning of a game and the contract requires the profit share distribution function \( a_{ij} \) values to be calculated by using the cost of player \( j \) for producing item \( i \) given by formula (15).

We consider the numerical values calculated in the proof of Property 2 with the modification of \( B \) as follows:

\[
B = \begin{bmatrix}
100 & 0 & 100 \\
0 & 30 & 30
\end{bmatrix}
\]

The solution to this problem by using the problem formulation in (14) is

\[
\begin{align*}
v_1 & = 0 \\
v_2 & = 340.9 \\
v_3 & = 0 \\
v_4 & = 340.9 \\
v_5 & = 0 \\
v_6 & = 340.9
\end{align*}
\]

in which case the core is empty due to the violation of the rationality property:

\[
u_1 + u_2 \geq v\{1,2\}, \\
u_3 \geq v\{3\}
\]

therefore

\[
u_1 + u_2 + u_3 \geq 2\nu^*
\]

However, if we formulate the problem by using (17) then we assume the profit distribution of each player is based on the percentage of the cost (or the resource) used by each player \( j \) on each item \( i \) for which case the contract is signed accordingly. In this example we use the values of \( B \) to calculate these percentages:

\[
a_{i1} = \frac{100}{100 + 30 + 100 + 30} = \frac{100}{260} = 38.462%
\]

\[
a_{i2} = \frac{130}{260} = 50%
\]

\[
a_{i3} = \frac{130}{260} = 11.538%
\]

The solution to (17) is

\[
\begin{align*}
v_1 & = 0 \\
v_2 & = 39.33 \\
v_3 & = 301.57 \\
v_4 & = 0 \\
v_5 & = 340.9 \\
v_6 & = 340.9
\end{align*}
\]

in which case the profit allocation for \( v\{1,2\} \) is calculated by using

\[
\sum_{i=1}^{3} \sum_{j=1}^{2} u_{ij}a_{ij}e_j = (340.9)(0.384) + (340.9)(0.5) = 301.57
\]

This would satisfy the core condition

\[
v\{3\} + v\{1,2\} = \nu^*
\]

The fairness condition (19) holds for this game’s players as follows:

\[
\frac{v(H)}{|H|} = \frac{v\{1,2\}}{2} = \frac{301.57 - 0.38462}{2} = 57.995% \frac{v(N)}{|N|} = \frac{340.9}{3} = 113.63
\]

\[
\frac{v(H)}{|H|} = \frac{v\{1,2\}}{2} = \frac{301.57 - 0.5}{2} = 75.393% \frac{v(N)}{|N|} = \frac{340.9}{3} = 113.63
\]

\[
\frac{v(H)}{|H|} = \frac{v\{3\}}{1} = \frac{39.33}{1} = 0.11538 = 4.537% \frac{v(N)}{|N|} = \frac{340.9}{3} = 113.63
\]

Considering the example in Section 4.5 and noting that \( v\{1,2\} = 301.57 \), the corresponding \( \epsilon - \) core set has the value \( \epsilon = 3.843 \).

The theoretical results attained so far are transformed into practice in the next section by using an algorithmic solution that will be used for distributed parallel programming in a university computer network.

6. Algorithmic Solution & Computational Complexity

In this section, we introduce an algorithm that will be used for computational experimentation of the problem formulated in Eq. (17) and its computational complexity. Let

\[
\begin{align*}
II & = [\sigma_{i, j} = |\{p \in F(y_j) - c\} y_j|]_{1 \times n} \\
\Omega & = [\sigma_{i, j} = |\{p \in F(y_j) - c\} y_j|]_{n \times 1} \\
e_\sigma & = [\sigma_{i, j} = |\{p \in F(y_j) - c\} y_j|]_{1 \times n}
\end{align*}
\]

for \( j \in S \) and \( 1 \leq i \leq n \). In this formula, \( I \) represents the vector consisting of the total profit \( pF(y_j) - c \) \( y_j \) for each product \( i \) based on the sale of \( n \) products with the decision variables \( y_j \). The matrix \( \Omega \) represents the matrix consisting of \( a_{ij} \) for all \( 1 \leq i \leq n \) and \( j \in S \), and \( e_\sigma \) represents the decision vector consisting of decision variables \( e_j \) for all \( j \in S \). Eq. (17) can be rewritten in the vector—matrix form

\[
\nu = \max_{\sigma_{i, j} \in \Omega} \Pi(e_\sigma)
\]

subject to

\[
A\nu \leq B e_\sigma \\
y \in R^+_n \\
e_\sigma \in \{0, 1\}^k
\]

In this problem the matrices \( \Omega \), \( A \) and \( B \) are known due to the contractual agreements. The subset \( S \) that determines the involvement of the players in the game, \( S \subset N \), is the important factor while determining \( \nu \) the variability of this set determines the Big-Oh complexity of the algorithmic solution to be introduced and explained next.

We let \( S_t \) be the subset of \( S \) such that the sets contained in \( S_t \) has either \( r \) number of players or less. For instance, the subset with 1 player, \( S_1 \), is defined to be

\[
S_1 = \{(1), (2), \ldots, \{N\} \}
\]

while \( S_2 \) would contain the sets consisting of 1 and 2 players, therefore \( S_1 \subset S_2 \subset \cdots \subset S_{|N|} = N \). The following algorithm can be used for determining a solution to \( \nu \) in the core to the proposed problem.

Algorithm 1. Coalition and Maximum Profit Allocation

1. Use the core set condition \( \sum_{j \in S} y_j - \nu = \nu - \nu^* \) to reduce the number of variables to \( |N| - 1 \).
2. Determine \( \nu \) for the coalition set \( S_i \). If \( B e_\sigma - A y \in \Omega \) and \( \sum_{j \in S} y_j - \nu \geq 0 \) hold, then the solution is in the core set. Otherwise, it is not.
3. For \( i = 2 \) to \( |N| - 1 \)
4. For \( j \in S_i \)
5. If \( \nu = \max \Pi(e_\sigma) = 0 \) then stop. The solution is infeasible. Go to Step 4.

(continued on next page)
Algorithm 1 is \(O \left( \left| \mathcal{N} \right| - 2 \right| 2^{N-1} \right) \) for \( |\mathcal{N}| > 2 \). It is shown in Section 7 for an example that this computational complexity is reduced to \( O(\log(|\mathcal{N}|)) \) when \( |\mathcal{N}| \in (1, 800) \) as a result of using a distributed parallel programming framework in a university computer network employing highly sophisticated 52 computers.

**Numerical Example**

The profit allocation of the coalition among the three players for two products requires solving the problem

\[
\nu = \max_{\Pi_{1,2}} \left( \left( \Pi_{2,3} \right) \left( \Pi_{1,2} \right) \right)
\]

\[
\text{subject to } \nu_{1} = \max_{\nu_{1}} \left( \left( \Pi_{1,2} \right) \left( \Pi_{1,2} \right) \right)
\]

The main goal of this constrained optimization program is to determine the maximum profit when the number of quantities ordered and the participation of the players in the game are the decision variables. The left-hand-side of the inequality constraint are the costs for each product while the right-hand side represents the resources available for production. Positive values of \( \nu \) force positive quantity orders from the retailer. Constraint \( \epsilon_{S} \in \{0, 1\} \) is the key element of the algorithmic solution impacting the objective function and the inequality constraint: this variable for the 3 players would require a maximum of 8 number of computations when \( j \in \mathcal{N} \).

Step 2.1 in Table 1 below; Step 2 of Algorithm 1, initiates the solution by using the set \( S = \{1, 2, 3\} \) for all \( i = 1, 2, 3 \) and reduces the original problem to

\[
\nu_{i} = \max_{\Pi_{1,2}} \left( \left( \Pi_{1,2} \right) \left( \Pi_{1,2} \right) \right)
\]

\[
\text{subject to } \nu_{1} = \max_{\nu_{1}} \left( \left( \Pi_{1,2} \right) \left( \Pi_{1,2} \right) \right)
\]

The solution to this problem finds \( y_{1} \) and \( y_{2} \) values for the particular values of \( \epsilon_{S} \in \{0, 1\} \), therefore identifies \( u_{1} \) and \( u_{2} \) to determine the validity of the core set constraints. The for loop of the algorithm requires

\[
\begin{array}{c|cccc|cccc|cccc|cccc}
\hline
\text{Step} & \epsilon_{(1)} & \epsilon_{(2)} & \epsilon_{(3)} & \epsilon_{(1,2)} & \epsilon_{(1,3)} & \epsilon_{(2,3)} & \epsilon_{(1,2,3)} \\
\hline
2.1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
2.2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
2.3 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
2.4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
2.5 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
2.6 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
2.7 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
2.8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
\end{array}
\]

The first and second rows of Table 1 are contract based percentage distributions of the cost for the first and second products respectively. Entries \( B_{ij} \) of matrix \( B \) represent the amount of resources available at \( j \in \mathcal{S} \) for the two products \( r = 1.2 \). We assume uncertain demands \( D_{1} \) and \( D_{2} \) of products 1 and 2 are normally distributed with a mean value of 10 items with the standard deviations of 4.5 and 5.75 respectively (i.e. \( D_{1} \sim N(10; 4.5 \) and \( D_{2} \sim N(10; 5.75 \)) The general problem takes the following particular form by using these assumptions:

\[
\nu = \max_{\nu_{1}} \left( \left( \Pi_{1,2} \right) \left( \Pi_{1,2} \right) \right)
\]

subject to

\[
\begin{array}{c|cccc|cccc|cccc|cccc}
\hline
\text{Step} & \epsilon_{(1)} & \epsilon_{(2)} & \epsilon_{(3)} & \epsilon_{(1,2)} & \epsilon_{(1,3)} & \epsilon_{(2,3)} & \epsilon_{(1,2,3)} \\
\hline
2.1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
2.2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
2.3 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
2.4 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
2.5 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
2.6 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
2.7 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
2.8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
\end{array}
\]
to use the following table values to complete all possible combinations of $e_t \in \{0, 1\}$: where $e_{1,2,3} = 0$ in all steps. Each step mentioned in Table 1 needs to be generated as a mathematical programming problem by using the general formula given in Eq. (17). Keeping steps 2.1–2.8 same in Table 1 and adding 8 more steps in this table by changing $e_{1,2,3}$ to 1 would be the continuation of the algorithmic solution which is not entered in Table 1 for space reduction purposes. In the case if the solution cannot be determined, the assumption $e_{1} = e_{2} = e_{3} = 1$ can be revisited and changed to consider eight possibilities when $e_{2} \in \{0,1\}$ for all $j = 1, 2, 3$. Steps of Algorithm 1 need to be repeated for a combination of these values.

It is natural that the matrix $\Omega$ can change due to the nature of the contractual agreement for each step displayed in Table 1. $\Omega$ can also be assumed to be a probabilistic matrix as a part of a Markov Chain if the contractual agreement changes periodically. However, we will not assume this to be the case in this work and solution of the proposed problem under consideration.

7. Distributed Network Solution & Numerical Experiments

7.1. Computer Network Description & Algorithm Architecture

Distributed application is one of the greatest commonly applied software techniques used in software industry that has proven its effectiveness and practicality. In this section, the applications and numerical results for the theoretical work developed are introduced to show the strength of the theory. The algorithmic solution to Eq. (17) is explained in detail. The numerical results are attained as a result of a distributed three-tier layered software application architecture developed; this methodology proved its outstanding performance and consistency when compared to the classic software application approach in this work.

In this software architecture, “service layer” is the middle-tier between clients (client-tier) and server-tier that is responsible to connect these two layers. This three-tier design facilitates a sharp separation between each one of the three layers so that they do not impact each other’s performances and computational efficiency. The three-tier approach can be deployed over any network as a distributed application to start a synchronous interaction and it has proven its high performance via parallel computations in this work. In fact, the designed framework, 3-Layer Distributed Game Theory Application (3-LDGTA), is a service-oriented application that works as a layer network-distributed application. The interoperability feature of 3-LDGTA between the server and the clients is the main strength of the computational effort. In addition, the ability to exchange messages during the communications through a variety of protocols such as HTTP and TCP/IP are significant tools that make 3-LDGTA work efficiently regardless of the transport technique used. The components of three layers, service, main (server) and clients, communicate with each other through messages that got sent one another and this is one of the important features of 3-LDGTA. All three layers of 3-LDGTA communicate in order to solve the game theory problem proposed in this work.

The service layer initiates the first step of the computational effort and distributes the problem to the clients. Each optimization problem that needs to be solved is one of the maximization problem that is determined for the game theory problem proposed in this work. The service layer receives the solutions from the clients and processes the results in a comparative manner. Communication between the client and the server is established via messages that are sent and received by both the clients and the server. In our design, there are 50 clients with multiple applications that run in these clients that we will explain the details later in this section.

The most powerful aspect of 3-LDGTA is exploitation of all possible resources available in the network and improve resource utilization in order to solve large scale optimization problems.

For instance, the server of 3-LDGTA has a total of 32 cores and running this server is equivalent to computational power of five computers with five cores and one computer with seven cores on a network. In our case we have 52 of such workstations. Therefore, processing of the data is fast and the numerical results attained for the optimization problems yield to improvement of the computational complexity of the solution methodology that we proposed. The data processing speeds up as the total number of clients exploited by the middle tier (service) increases.

The service layer provides interface between the server and the clients. It sends and receives messages on the problems to be solved by the clients. These messages carry information and tasks that required to be performed by each client. The client sends messages back to the server throughout the service layer regarding to the overall solution of the problem. The clients, the server, and the service layer must communicate over the network using multiple TCP/IP sessions. Therefore, the server and the clients have their own individual IP addresses.

The service layer does not code among the clients and the server, thus decreases the system coupling. The communication between the server and the clients provide many benefits such as reuse and data persistence of operations, reduced duplication of code, and enhanced system deployment and effectiveness.

The three components (server, client, and service) communicate with each other using IP addresses which provide effective transportation technique over network using TCP protocol. Initially, the service sends the calling request to the client. The client responses to the service if it is available to receive a task for completion. Next, this availability status is passed on to the server. Thus, the server can determine the number of available clients that are able to distribute the tasks to the available clients. The service runs on the server side along with the server application. The system coupling is occurring as the service communicates with the clients and the server, which gives the advantage of high-speed communication with robust security. This 3-LDGTA consists of three parts that are presented in Fig. 2.
The following algorithms are followed for the network implementation.

**Algorithm 2.** Server Operations

1. Input number of players $N$ and number of products $n$
2. Calculate the set of possible combination of players $2^N - 1$
3. Set the fixed number of clients to be used in the network.
4. Create contract matrix $X_{i,m}$
5. Create resource amount matrix $B_{i,m}$
6. Sales price product vector $P_{i,1}$
7. Create cost vector $c_{i,1}$
8. Set matrix necessity resource $A_i$
9. Create a binary decision matrix of size $m \times n$; this generates the values in Table 1.
10. Distribute objective function $v_i$ to clients via the network by using the matrices, vectors and variables introduced in Steps 1–8 above.
11. Receive objective function calculations from the clients and compare the values to update the $v_i$ iteratively.

**Algorithm 3.** Client Operations (WCF)

1. Launch the Sever App and the Client Apps
2. Set number of players $N$
3. Set $N \times \max = 2^{N-1}$; the number of maximum values to be calculated for objective function $v_i$.
4. Server assigns each maximization problem $\max$ ($\max_i$ for a specific instance chosen) to an available client.

(continued on next column)

<table>
<thead>
<tr>
<th>Players</th>
<th>Results</th>
<th>Instance solution time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>5.724325E + 07</td>
<td>180</td>
</tr>
<tr>
<td>200</td>
<td>1.562078E + 08</td>
<td>211</td>
</tr>
<tr>
<td>300</td>
<td>7.892668E + 07</td>
<td>230</td>
</tr>
<tr>
<td>400</td>
<td>1.144828E + 08</td>
<td>242</td>
</tr>
<tr>
<td>500</td>
<td>7.625234E + 07</td>
<td>247</td>
</tr>
<tr>
<td>600</td>
<td>1.680292E + 08</td>
<td>259</td>
</tr>
<tr>
<td>700</td>
<td>8.663870E + 07</td>
<td>260</td>
</tr>
<tr>
<td>800</td>
<td>1.409773E + 08</td>
<td>269</td>
</tr>
<tr>
<td>900</td>
<td>5.730934E + 07</td>
<td>329</td>
</tr>
<tr>
<td>1000</td>
<td>8.751037E + 08</td>
<td>400</td>
</tr>
</tbody>
</table>

**7.2. Algorithms and Computational Results**

Next, we present the experimental results based on the 10 experiments conducted and comparison of these results with the ones in the literature. The following algorithms are followed for the network implementation.

A numerical experiment is implemented in Quinnipiac University’s internal network. The network was dedicated for this numerical experimentation. The server and the work stations are very powerful and have high specifications that allowed to have supercomputing methodologies implemented. All work stations and the server had the following same specifications:

- Windows 10
- Memory: 6 GB
- GPU: nVidia M10 GRID card

There are 50 work stations (i.e. clients), a server and a service app employed for attaining the following numerical results in Table 2. These results are attained for a randomly generated instances that takes place in the game that follows the rules that we specify in this article. The problems solved in the literature cannot be used in this work noting that the network system that we are using is very powerful, therefore capable of solving any game with players $|N|$ ranging from 100 to 1000 (with 100 increments) in a timeframe of approximately 180 to 400 s. There were no computational issues from a time perspective therefore all the results for the problem instances are completed and outlined in Table 2. The number of products is chosen to be the same as the number of players to attain these numerical results. These numerical experiments are strong indicators that the distributed processing approach we used is a very strong and sound method for solving game theory problems in a fast and efficient manner with very good results. In comparison to the literature, we are not reporting the Shapley Value in this setting noting that the CPU effectiveness of the network was extremely high to solve the problems with the corresponding instances.

It is crucial to point out that we are attaining a solution for as many as $|N| = 1000$ players requiring a computational effort of $2^{1000} - 1$ runs in the network. We see that the number of iterations grows, if the number of players grows, but still very large instances can be solved with the proposed algorithmic solution within a reasonable CPU time as low as 6 min 40 s for 1000 players. The results are even more impressive when compared to the existing literature noting that 150 instances were

![Graph](image-url)

Fig. 3. The best fit natural logarithmic model for the time to solve instances in seconds.
considered by Drechsel et al. (2010) for game theoretic solutions and some of these solutions were halted after 20,000 iterations therefore the need for calculating the Shapley Parameter (SP) value arose in these studies; however, the computational results we attained are strong and sound therefore a need for calculating the SP values appeared to be redundant in our numerical value reporting. In the literature, as we pointed out in the introduction, CC has been used recently for attaining computational results however CC has downtime costs, inefficiency, limited control, and cost of runtime (www.cloudacademy.com, 2020). Our distributed parallel network solution eliminates all these shortcomings. We must also note that running such a network is expensive and require extensive amount of programming practice to design the software architecture. The results indicated no need for \( \varepsilon \)-core or least core to be calculated for determining the solutions to the problems.

The computational complexity of the designed network solution is simplistic noting that \( |N| \) number of players are distributed to \( m \) work stations with Server and Service layers supporting the work station computations. This indicates \( 2^{\frac{|N|}{m}-1} \) number of problems to be distributed to each work station on average. Noting that Algorithm 1 is utilized with complexity ranges from \( \Omega(|N|) \) to \( O((|N| - 2)2^{\frac{|N|}{m}-1}) \) when \( |N|/2 \) for one work station, the complexity of each work station when \( m \) work stations are available in a range from \( \Omega\left(\frac{|N|}{m}\right) \) to \( O\left(\left(\frac{|N|}{m} - 2\right)2^{\frac{|N|}{m}-1}\right) \) for \( |N|/2 \).

However, the hardware features of the computers that we used in the network boost the computational complexity and resulted in either natural logarithmic or linear behavior of the computational times to
calculate the proposed problems. Figs. 3 and 4 below display the distributions fitting to the runtime of the program to compute values in Table 1. In the case when the number of players range from 100 to 800, the computational time displayed in Fig. 3 follows a logarithmic distribution with the fitting equation of $y = 41.977 \ln(x) - 11.702$ seconds and $R^2$ value of 99.46%, indicating a strong fitting of the curve to the corresponding data. We can conclude that the computational complexity is $O(\log(x))$ for the particular domain $1 < x < 800$.

The computational complexity after 800 number of players starts to have a constant rate of change therefore a better fit in this case is a linear distribution indicating $O(x)$ complexity for $800 < x < 1000$. The corresponding model is determined to be $y = 0.655 x - 256.83$ with $R^2$ values of 99.77%.

Novais et al. (2019) pointed out the need for the use of IoT in Supply Chain (SC) and the need to have better methods for handling big data in SC as a research gap. Our work invests in this area of improvement in a big-data SC framework for 1000 players to achieve strong computations results using 3-LDGTA. To the best of our knowledge, this is the first work in the literature that solves a non-linear production game in SC with 1000 players in a timeframe as low as 400 s. Fig. 5 displays the computational complexity values as a means of Big-Oh computation for three different scenarios; $N \in \{1100, 1200\}$. $N \in \{1200, 1300\}$and $N \in \{1100, 1200\}$, when the number of work stations $m$ range between 50 and 60.

This indicates the exponential decay in the computational complexity of the corresponding problem when the number of work stations in the network increases linearly from 50 to 60. While increasing the work stations increase the cost of the solution to the problem, the corresponding time to compute the exact result is decreasing exponentially. It is important to note here that the time it takes for an NP hard problem to be solved in our proposed solution method depends on $m$ and $N$ along with the complexity degree of the problem; therefore, while we present that the general Big-Oh complexity of the problem is reduced dramatically with our proposed method, the specific turning point (which is $N = 800$ in the example given) from logarithmic complexity to linear complexity depends on $m$, $N$, and the nature of the problem’s complexity degree to be solved, therefore this particular point cannot be determined for all problems with a specific formula.

8. Conclusions & Future Work

In this paper, a supply network design problem under uncertain demands has been studied. We have considered a network of manufacturers in a dominant position relatively to a retailer. The foremost contribution of this study is modeling a new kind of cooperative game named nonlinear production game (NLPG). Furthermore, we showed that the NLPG is a grand coalition game. The stability, fairness and superadditivity of the formulated maximization problem is illustrated for supply chain partners by incorporating contractual agreement among them based on the cost distribution. As an extension to the core implementation, least-core and $\epsilon$ – core are also shown in this work. An algorithmic solution for the formulated optimization problem that worked effectively in a distributed network setting is presented. A sub-network of Quinnipiac University’s computer network consisting of 52 computers is utilized for attaining the computational results. Using the proposed algorithms and the sub-network, computational results up to 1000 number of players are attained within 400 s, a very impressive result that was not attained in the literature. Our findings can incite small- to large-sized enterprises managers to cooperate by sharing logistics resources in order to deal with powerful retailers. In fact, as the number of players increase, the computation of the allocations becomes cumbersome and tedious due to the NP-hard nature of the problem. Hence, our algorithm can be used for much larger games for robust and scalable solutions based on distributed parallel programming that can generate stable and fair allocation. For instance, the production of Boeing 737 required a large number of manufacturers to take place in the game; approximately 600 suppliers manufactured components for the production of the plane (Slotnick, 2019). Boeing’s 787 Dreamliner is assembled in the United States with parts sourced worldwide with 45 big companies delivering parts as the subcontracts (Kavilanz, 2013).

In comparison to the literature, we are not reporting the Shapley Value for the computational results noting that the CPU effectiveness of the network was extremely high to solve the problems with the corresponding instances. In addition, CC has been used recently for attaining computational results however CC has downtime costs, inefficiency, limited control, and cost of runtime (Larkin, 2019). Our distributed parallel network solution eliminates all these short comings.

As a possible extension of this study, we may consider a symmetric case in terms of leadership by shifting the power from the manufacturers’ network to the retailer. This is another important research theme since in practice we observe many retail companies in a strong position, i.e Wal-Mart and Carrefour. Secondly, we assumed complete information sharing between supply chain partners. This assumption can be relaxed by considering that partners have private information of customer demand and logistics costs as a part of future research considerations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project is supported by Quinnipiac University. We would like to thank Fabiano Iacucso, Associate Director of Systems, Technology & Planning; Harry Daddio, Systems Administrator; Andy Soliwoda, Senior Network Engineer of Quinnipiac university for their extensive support. The authors would like to thank Dr. Bruce Hartman for his useful and insightful comments on an earlier draft of this manuscript.

References


