

Cloud Federation Formation in Oligopolistic Markets

by

Yash Khandelwal, Ganti Karthik, Suresh Purini

in

24th International European Conference on Parallel and Distributed Computing

Report No: IIIT/TR/2018/-1



Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2018

Cloud Federation Formation in Oligopolistic Markets

Yash Khandelwal¹, Karthik Ganti¹, Suresh Purini¹ and Puduru V. Reddy²

¹ International Institute of Information Technology, Hyderabad, India

² Indian Institute of Technology, Chennai, India

Abstract. In this paper, we study how an oligopolist influences the coalition structure in federated cloud markets. Specifically, we use cooperative game theory to model the circumstances under which a cloud provider prefers to join a cloud federation vis-a-vis consider taking a price offer made by an oligopolist.

Keywords: Federated Clouds, Oligopoly, Linear Production Games

1 Introduction

The current cloud computing market structure is akin to *oligopoly* as few mega cloud providers completely own the market share. Each of them individually or in collusion has the power to affect the market prices leading to what is called an *imperfect competition*. Further, due to the large scale of operations in the data centers associated with these oligopolists, there is an acute stress on electricity and other natural resources. Many studies indicated the resulting adverse impact on the environment due to carbon emissions and other pollutants.

Since computing has become a common commodity these days, it is easy to envisage a large number of micro cloud providers with small to medium scale data centers. With the presence of a large number of producers, an oligopolistic market leans towards *perfectly competitive market*. In a market with perfect competition, producers become price takers and it is not possible for one or few cloud providers to affect the market prices. Further, as these small data centers are geographically spread out, the stress on the local resources and the impact on the microclimate will be mitigated, especially by the usage of renewable energy resources and productive use of dissipated heat energy.

However, such micro cloud providers will be able to serve only moderate sized consumer requests due to the limited availability of resources in their data centers. In order to serve large consumer requests many micro cloud providers have to come together and form a coalition or a federation. The federation formation can happen in a peer-to-peer fashion leading to what is called an Peer-to-Peer Inter-Cloud Federation (refer Figure 1(a)) [1]. The other option is to use the services of a broker as in Figure 1(b) resulting in a Multi-Cloud federation model.

Given a set of cloud providers and a broker, in this paper, we study the question whether it is beneficial for cloud providers to form a peer-to-peer federation

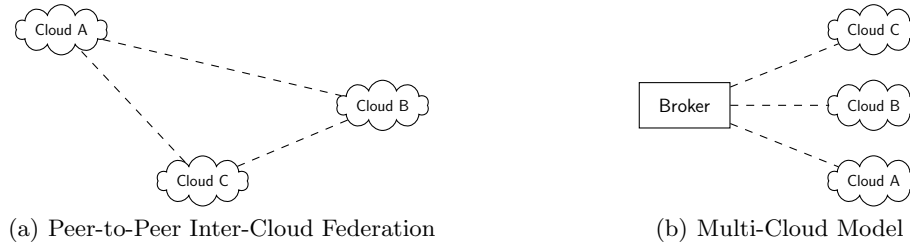


Fig. 1. Cloud Federation Models

or to subscribe to the services of a broker. We use cooperative game theory to address this interesting question. To the best of our knowledge, we did not find any prior work related to the proposed problem of study in this paper.

In Section 2, we provide the necessary background on cooperative game theory and linear production games; in Section 3, we formulate the cloud federation formation and payoff distribution using linear production games; in Section 4, we show the impact of an oligopolist on federation formation and how we can arrive at stable coalitional structures; in Section 5 we present our experimental analysis; Section 6 contains the related work; and finally we conclude with Section 7.

2 Background

We study the proposed problem in this paper using a special class of games called linear production games from the cooperative game theory [2, 3]. We provide the necessary game theoretic background in this section to understand the rest of the paper.

2.1 Cooperative Game Theory

Given a set of $N = \{1, \dots, n\}$ players, a subset $S \subseteq N$ of them can pool their resources and form a coalition to generate an utility or value $v(S)$. We say that the utility is *transferable* if it can be split among the coalition partners in an arbitrary fashion.

Definition 1. A cooperative n -person game in coalitional form is denoted by $G = (N, v)$ where $N = \{1, \dots, n\}$ and $v : 2^N \rightarrow \mathbb{R}^+$, with $v(\phi) = 0$. The function v is called the characteristic function of the game and $v(S)$ is called the value of the coalition S .

We say that a cooperative game is *super-additive* if $v(S \cup T) \geq v(S) + v(T)$ for all $S, T \in 2^N$ with $S \cap T = \phi$. When a game is super-additive, then the value $v(N)$ generated by the *grand coalition* N would be the maximum. However, the formation of a grand coalition or any other coalition depends on the payoff vector which determines the profit distribution among the coalition members.

Definition 2. A payoff vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ is called an imputation if it satisfies the following individual rationality and efficiency conditions.

1. Individual rationality: $x_i \geq v(\{i\}) \forall i \in N$.
2. Efficiency: $\sum_{i=1}^n x_i = v(N)$.

The set of imputations associated with a game $G = (N, v)$ is denoted by $I(G)$. For a payoff vector x and a coalition $S \subseteq N$, let $x(S)$ denote $\sum_{i \in S} x_i$.

Definition 3. The core of a game $G = (N, v)$ denoted as $C(G)$ is defined as follows.

$$C(G) = \{ x \in I(G) \mid x(F) \geq v(F) \forall F \subseteq N \}$$

If the payoff vector is from the core, then there is no incentive for any sub-coalition $S \subset N$ to deviate from the grand coalition N , thus ensuring stability. However, the core of a game is not necessarily non-empty. Bondareva [4] and Shapley [5] gave independently a characterization of games with a non-empty core. The characteristic vector e^S associated with a coalition $S \subseteq N$ is defined as $e_i^S = 1$ if $i \in S$ and $e_i^S = 0$ if $i \in N \setminus S$.

Definition 4. A map $\lambda : 2^N \setminus \{\phi\} \rightarrow \mathbb{R}^+$ is called a balanced map if

$$\sum_{S \in 2^N \setminus \{\phi\}} \lambda(S) e^S = e^N.$$

Definition 5. A cooperative game $G = (N, v)$ is called a balanced game if for each balanced map $\lambda : 2^N \setminus \{\phi\} \rightarrow \mathbb{R}^+$ the following condition holds good.

$$\sum_{S \in 2^N \setminus \{\phi\}} \lambda(S) v(S) \leq v(N)$$

A cooperative game $G = (N, v)$ can induce a subgame $G_S = (S, v_S)$ where $S \subseteq N$ and $v_S(T) = v(T)$ for all $T \subseteq S$.

Definition 6. A cooperative game $G = (N, v)$ is called totally balanced if every induced subgame $G_S = (S, v_S)$ for all $S \in 2^N \setminus \{\phi\}$ is balanced.

The following theorem due to Bondareva and Shapley characterizes the set of games with a non-empty core.

Theorem 1. A cooperative game $G = (N, v)$ will have a non-empty core if and only if it is a balanced game.

2.2 Linear Production Games

Consider a production situation where m different types of products P_1, \dots, P_m can be manufactured using q distinct kind of resources G_1, \dots, G_q . Further, there is a production matrix $A_{m \times q}$ whose $(j, k)^{th}$ entry a_{jk} denotes the number of units of resource G_k required to manufacture an unit of product P_j . Overall,

the j^{th} row of the matrix denoted by a_j gives the overall resource requirements per unit of product P_j . The linearity of the production situation comes from the fact that to manufacture α units of product P_j the corresponding resource requirements scale-up linearly to αa_j . Let the j^{th} entry of the price vector $c_{1 \times m} = (c_1, \dots, c_m)$ denote the price per unit of product P_j . Given a resource bundle $b_{q \times 1} = (b_1, \dots, b_q)^T$ with non-negative entries, the optimal production plan $x_{m \times 1} = (x_1, \dots, x_m)^T$ is obtained by solving the following linear programming problem.

$$\begin{aligned} & \underset{x}{\text{Maximize}} && c \cdot x \\ & \text{subject to} && A^T \cdot x \leq b \\ & && x \geq 0 \end{aligned}$$

Consider now an n -player game $G = (N, v)$ wherein the resource bundle owned by the i^{th} player is denoted by b_i . The resource bundle owned by a coalition $S \subseteq N$ is defined as $b(S) = \sum_{i \in S} b_i$. The value $v(S)$ associated with the coalition S is obtained by solving the following linear programming problem.

$$\begin{aligned} & \underset{x}{\text{Maximize}} && c \cdot x \\ & \text{subject to} && A^T \cdot x \leq b(S) \\ & && x \geq 0 \end{aligned}$$

The following is an important theorem which we use in this paper.

Theorem 2. *Every linear production game $G = (N, v)$ is totally balanced. Hence not only the core $C(G)$ is non-empty but also the core $C(G_S)$ of every induced subgame $G_S = (S, v_S)$ where $S \subseteq N$ is also non-empty.*

3 Federation Formation and Payoff Distribution using Linear Production Games

In this section, we will present a model for peer-to-peer inter-cloud federation and an efficient payoff distribution scheme which gives a core allocation using linear production games.

3.1 Federation Formation Model

Let $\mathcal{I} = \{C_1, \dots, C_n\}$ be a collection of cloud providers. A cloud provider C_i owns a resource bundle $b_i = (b_i^c, b_i^m, b_i^s)^T$ where b_i^c is the total number of available compute cores; b_i^m and b_i^s denotes the total available main memory and secondary storage respectively. The cloud providers can offer m types of virtual machines denoted by VM_j , $1 \leq j \leq m$. The core, main memory and storage requirements for each virtual machine type is given by the production matrix $A_{m \times 3}$ whose j^{th} row, $a_j = (a_j^c, a_j^m, a_j^s)$, corresponds to the resource configuration vector of a virtual machine of type VM_j . Table 2 gives example virtual machine types and

the associated production matrix used in the experimental analysis section of this paper. The per unit market price of different types of virtual machines is denoted by the price vector $p = (p_1, \dots, p_m)$. Table 2 also provides the hourly rental price for various types of virtual machines considered. Given this market scenario, the cloud providers have to decide upon a federation structure such that each of them maximize their respective payoffs.

It can be observed that we can model this problem by constructing a linear production game which is exactly similar to the game $G_{lp} = (N, v)$ described in the Section 2.2. We denote the total pooled cores, memory and storage from a federation S by $b^c(S)$, $b^m(S)$ and $b^s(S)$ respectively. The value $v(S)$ associated with a federation S is obtained by solving the following linear programming problem OPTLP(S).

$$\text{Maximize}_x \quad \sum_{j=1}^m x_j p_j \quad (1a)$$

$$\text{subject to} \quad \sum_{j=1}^m x_j a_j^c \leq b^c(S) \quad (1b)$$

$$\sum_{j=1}^m x_j a_j^m \leq b^m(S) \quad (1c)$$

$$\sum_{j=1}^m x_j a_j^s \leq b^s(S) \quad (1d)$$

$$x_j \geq 0 \quad (1 \leq j \leq m) \quad (1e)$$

Constraints 1b, 1c and 1d denote the capacity constraints corresponding to core, memory and storage respectively. In fact, this game being super additive, we can infer that the grand coalition generates the maximum revenue, which is obtained by solving the linear programming problem OPTLP(N). Further, from Theorem 2, we know that there is a core allocation possible as it is a totally balanced game. In the next section, we show how we can do payoff distribution using a core allocation, thereby achieving the stability of the grand coalition.

3.2 Payoff Distribution

Owen [6] showed that we can compute a core allocation for a linear production game $G_{lp} = (N, v)$ by solving the following dual problem associated with the primal problem OPTLP(N).

$$\text{Minimize}_y \quad y_1 b^c(N) + y_2 b^m(N) + y_3 b^s(N) \quad (2a)$$

$$\text{subject to} \quad y_1 a_j^c + y_2 a_j^m + y_3 a_j^s \geq p_j \quad (\forall j, 1 \leq j \leq m) \quad (2b)$$

$$y \geq 0 \quad (2c)$$

We interpret the optimal solution $y_* = (y_*^c, y_*^m, y_*^s)$ to the dual problem as the shadow prices for cores, memory and storage. Owen proved that we can obtain a core allocation vector by paying the i^{th} player with the resource bundle $b_i = (b_i^c, b_i^m, b_i^s)^T$ as follows.

$$\alpha_i(N) = \sum_{j \in \{c, m, s\}} y_*^j b_i^j$$

We denote the payoff vector as $\alpha(N) = (\alpha_1(N), \dots, \alpha_n(N))$ where the parameter N indicates that the payoff corresponds to the grand coalition. The subset of core allocations which are formed using optimal dual solutions is known as the Owen set. In the next section, we will present how a broker or an oligopolist can intervene in the formation of a grand coalition by offering higher payoff to individual cloud providers.

4 Intervention of an Oligopolist in Federation Formation

In order to maintain market control, the oligopolists may intervene in the peer-to-peer federation formation, refer Figure 1(a), by offering incentives to the micro cloud providers to lend their resources to them. The oligopolists in turn use the lent resources to supply virtual machines to the end consumers potentially at a higher price due to their wider market reach. During this process, an oligopolist assumes the role of a broker leading to a multi-cloud architecture depicted in Figure 1(b). In the rest of this section, we study how an oligopolist can affect the structure of cloud federation and the resulting impact on the payoff to individual cloud providers.

Let an oligopolist offers a price m_i to rent the entire resource bundle b_i from the cloud provider C_i . In this paper, we study the restricted problem of an interaction between a single oligopolist and a set of cloud providers³. One simple way of considering more than one oligopolist is to set the price offer m_i made to the cloud provider C_i to the maximum of the offers made by different oligopolists in the market, and the rest of the theory proposed in this section holds good.

4.1 Core Allocation for Subgames

In Section 3.2, we described how the payoff distribution vector $\alpha(N)$ can be computed for the game $G_{I_p} = (N, v)$. Since, every subgame $G_S = (S, v_S)$ induced by G_{I_p} is also a linear production game, we can analogously compute the payoff distribution vector $\alpha(S)$ by solving the dual problem for the primal problem OPTLP(S). Overall, we have to solve $2^n - 1$ linear programming problems to compute the payoff distribution vectors for all the induced subgames, which is

³ An alternate way to view this problem is to consider the single oligopolist as a monopolist by ignoring the market influences due to other oligopolists which is not the subject matter of this paper.

computationally expensive. However, it can be noted from the constraints (2b) and (2c), the feasible region for the dual problem of OPTLP(S) is independent of the federation S and only the coefficients of the objective function change. Hence, for practical values of m , we can enumerate the basic feasible solutions, in other words, the extreme points of the polyhedra defined by the dual problem constraints. For different objective functions associated with different subgames, we can exhaustively check the list of extreme points and find the optimal solution.

4.2 Influence of the Oligopolist

Definition 7. *The marginal payoff for a cloud provider C_i with respect to a coalition S and a price offer m_i from an oligopolist is defined as*

$$\beta_i(S) = \alpha_i(S) - m_i.$$

A cloud provider has an incentive to deviate from a federation S and take up the offer of an oligopolist if and only if $\beta_i(S) < 0$. Thus the oligopolist may destabilize the grand coalition as all the cloud providers whose $\beta_i(N) < 0$ will break away from the coalition.

Definition 8. *For a cooperative game $G = (N, v)$ and a price offer vector $m = (m_1, \dots, m_n)$, a coalition $S \subseteq N$ is called a feasible coalition if and only if $\beta_i(S) \geq 0$ for all $i \in S$.*

From the discussion in Section 4.1, we can enumerate the list of all feasible coalitions in 2^N by computing the respective payoff distribution vectors.

Definition 9. *Given price offer vector m , we call a partition $CS = \{F_1, \dots, F_{k-1}, F^*\}$ of the player set N as a stable coalition structure if*

1. *The coalitions $F_i, 1 \leq i \leq k-1$ are feasible coalitions.*
2. *There exists no subset $S \subseteq F^*$ which is a feasible coalition. Thus all the cloud players from F^* take the price offer made by the monopolist.*

Note that if $m_i < v(\{i\})$, then cloud provider C_i is a feasible coalition by himself.

4.3 Finding a Stable Coalition Structure

There can be many possible stable coalition structures for a given price offer vector from the oligopolist. We may prefer one stable coalition structure to other based on certain criteria. For example, one criteria could be to minimize the number of cloud providers taking up oligopolist's offer, i.e., $|F^*|$. Another criteria could be to maximize the sum of payoffs of all the cloud providers, i.e., $\sum_{i=1}^{k-1} \sum_{j \in F_i} \alpha_j(F_i) + \sum_{j \in F^*} m_j$.

Definition 10. *For a feasible coalition F and a price offer vector $m = (m_1, \dots, m_n)$, we associate a goodness value $g(F)$ which is defined as follows.*

$$g(F) = \sum_{i \in F} (\alpha_i(F) - m_i) / |F|$$

In this paper, we propose the following simple greedy algorithm for stable coalition formation.

1. Let the initial coalition structure be $CS = \phi$. Repeat the following step until it terminates.
2. (i^{th} iteration)
 - (a) Among all the feasible coalitions, choose a coalition F_i with a maximum goodness value $g(F_i)$ and $F \cap F_i = \phi$ for all $F \in CS$.
 - (b) If there exists no feasible coalition which is disjoint with the already chosen feasible coalitions, then set

$$\begin{aligned} F^* &= N - \cup_{F \in CS} F \\ CS &= CS \cup \{F^*\} \end{aligned}$$

and exit the algorithm.

We can easily note from the above algorithm, that different goodness functions will yield different coalition structures. In the next section, we do an experimental analysis on the influence of the oligopolist on stable coalition formation and overall payoff distribution.

5 Experimental Results

In this section, we study how increasing price offers from the oligopolist to the individual cloud providers impact the structure of stable coalitions formed. We consider a set of 12 cloud providers $\mathcal{I} = \{C_1, \dots, C_{12}\}$ whose resource capacities are given in the Table 1. These resource capacities are randomly chosen, first by choosing one of the three buckets: small, medium and large; and then choosing a capacity randomly within a range determined by that bucket type. Inspired from Microsoft Azure, we let each cloud provider offer four types of virtual machines: General Purpose (B2S), Storage Optimized (L4), Memory Optimized (E8 v3), and Compute Optimized (F16 v2). The resource requirements of each type of virtual machine is given in the Table 2. The same table also provides the hourly rental price for each type of virtual machine.

We consider $l = 45$ different market scenarios. In the i^{th} market scenario, M_i , $1 \leq i \leq l$, the oligopolist makes a price offer $m = (m_1, \dots, m_j, \dots, m_{12})$ wherein

$$m_j = \left(1 + \frac{i}{100}\right) \times v(\{C_j\}). \quad (3)$$

That means the oligopolist is offering a price which is $i\%$ greater than the value a cloud provider can generate by working all alone. For small values of i , a cloud provider can potentially get better payoff by forming a coalition; whereas for larger values of i he may be better off taking up the oligopolist's offer. This can be observed from the Figure 2 which depicts how the stable coalition structure evolves with the increasing price offers from the oligopolist. The stable

Table 1. Resource capacity of cloud providers. vCPUs are expressed in 100s of cores, memory and storage in 100 GB units.

	vCPU	Memory	Storage
C_1	36	44	1845
C_2	55	74	1704
C_3	120	165	548
C_4	15	133	1906
C_5	61	490	2100
C_6	110	503	3164
C_7	119	900	3468
C_8	181	150	3900
C_9	182	986	6814
C_{10}	210	610	4654
C_{11}	166	531	13000
C_{12}	239	850	4100

Table 2. VM instance types, their resource configurations and hourly rental prices.

	vCPU	Memory	Storage	Price
	(in GB)	(in GB)	(in GB)	(per hour)
General Purpose	2	4	8	0.047\$
Storage Optimized	4	32	678	0.312\$
Memory Optimized	8	64	200	0.532\$
Compute Optimized	16	32	128	0.716\$

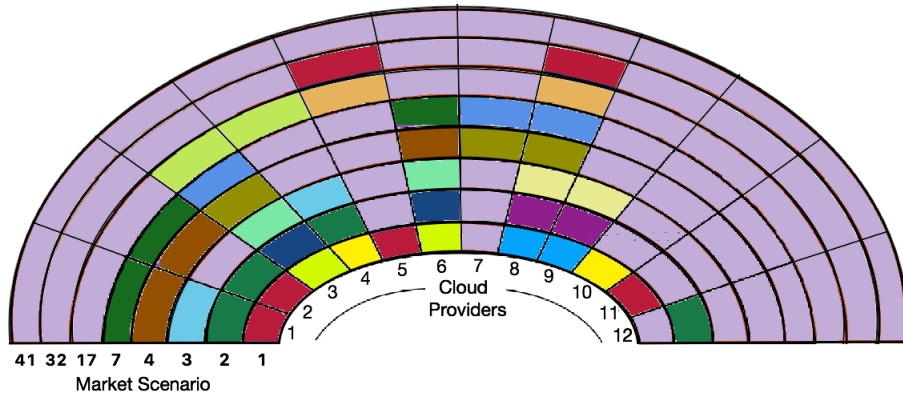


Fig. 2. Evolution of coalition structure with increasing price offers going from market scenario M_1 to M_{45} (refer Equation (3)).

coalition structures are computed using the greedy algorithm proposed in Section 4.3. Each track of the semi-circle represents the coalition structure for a given price offer. The purple colored cloud providers are those who take up the oligopolists offer. Similar colored cloud providers in a track belong to the same coalition. For example, at one percent price offer, the coalition structure is $CS = \{\{1, 2, 5, 11\}, \{3, 6\}, \{4, 10\}, \{8, 9\}, \{7, 12\}\}$. The members of the last set $F^* = \{7, 12\}$ are those who accepted the offer made by the oligopolist. Further,

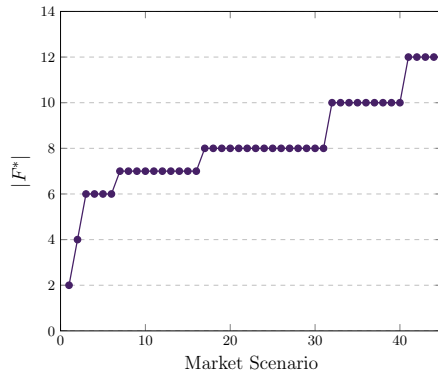


Fig. 3. Number of cloud providers taking up the offer made by the oligopolist with increasing price offers.

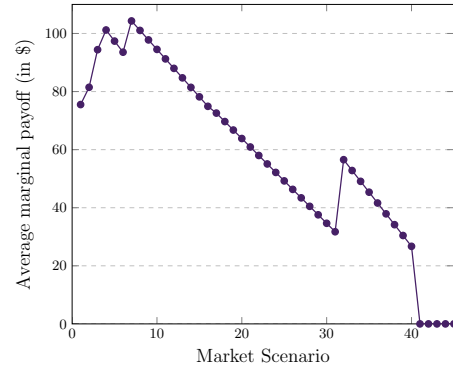


Fig. 4. Average marginal payoff with changing market scenarios.

the semi-circle shows only those tracks where there is a change in the coalition structure from the previous market scenario. For example, since the coalition structure did not change from the market scenario M_7 till M_{17} , the intervening coalition structures are not represented. We can notice the increasing purple color as we move from inside to outside in the semi-circle indicating that with increasing price offers more cloud providers will lean towards the oligopolist. This is further illustrated by the graph in Figure 3 which shows the size of F^* , $|F^*|$, with increasing price offers. Another interesting observation is that a cloud provider may take an oligopolist's offer in market scenario M_i but may change his mind in $M_{i'}$ where $i' > i$. This is due to the overall change in the coalition structure. This phenomenon can be observed by looking at the sector corresponding to the cloud provider 5 in Figure 2.

Figure 4 shows the average marginal payoff of the cloud providers who preferred to form a peer-to-peer coalition. For a given market scenario, CS is a stable coalition structure (refer definition (9)), then the average marginal payoff is defined as $\sum_{F_i \in CS \setminus F^*} \sum_{j \in F_i} \beta_j(F_i) / |N \setminus F^*|$. As expected, with the increasing price offer from the oligopolist, the marginal payoff goes down. However, it need not be monotonic, as it may increase locally due to the changes in the stable coalition structure. Figure 5 shows the total time taken for the computation of the stable coalition structure for a given market scenario. It can be noted that overall it is in the order of milliseconds and hence computationally feasible problem to solve. Further, with the increasing price offers, the number of feasible coalitions go down, which makes the greedy algorithm converge faster.

For a coalition, we know that $v_S(S)$ is the total payoff available for the coalition S . The combined payoff from an oligopolist to a coalition S is $\sum_{i \in S} m_i$. Figure 6 compares the coalitional payoff and the combined broker payoff for all the coalitions in the market scenario M_1 . For cloud providers 7 and 12, who

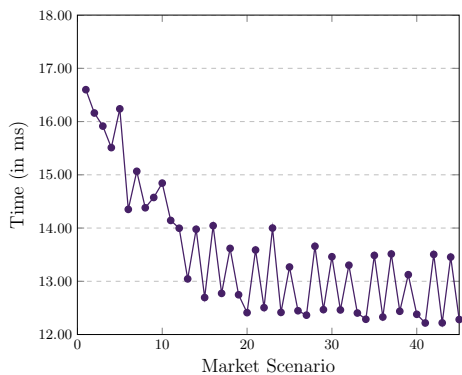


Fig. 5. Total time taken to compute the stable coalition structure for a given market scenario.

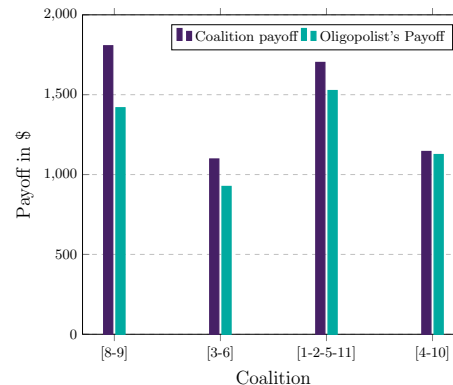


Fig. 6. Comparison between the coalitional payoff and combined broker price offer for the market scenario M_1 .

take up the oligopolist's offer, these two values are almost the same (one percent difference).

6 Related Work

Grozev et al. [1] provided a systematic taxonomy of various inter-cloud architectures. The peer-to-peer inter-cloud architecture and the broker based multi-cloud architecture considered in this paper are based on their taxonomy. There has been several works on federation formation and payoff distribution using cooperative game theory [7–12]. However, none of these works consider the impact of an oligopolist or a monopolist on coalition formation which is the main focus of this paper. Niytao et al. [9] proposed the usage of stochastic linear programming games for payoff distribution among coalition members. The payoff distribution scheme we presented in this paper using linear production games is similar to their work. The closest work related to ours in literature is due to Fragnelli [13]. The author studied a market scenario which is very similar to that of ours but the specific problem addressed is the pricing strategy to be adopted by the players. Innes and Sexton [14] also studied very similar market scenario but the problem they studied is the pricing strategy to be adopted by the monopolist to deter coalition formation.

7 Conclusions

In this paper, we showed how we can model the influence of an oligopolist in a cloud market where multiple cloud providers can potentially come together to form a federation in order to increase their market reach. Further, we introduced the notion of stable coalition structures in the presence of oligopolists and a

greedy algorithm for computing them. We believe that our work paves way for further research in this less studied facet of federated cloud computing.

References

1. Grozev, N., Buyya, R.: Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience* **44**(3) (2014) 369–390
2. Curiel, I.: *Cooperative Game Theory and Applications*. Springer, Boston, MA (1997)
3. Tijs, S.: *Introduction to Game Theory*. Hindustan Book Agency (2003)
4. Bondareva, O.N.: Some applications of linear programming methods to the theory of cooperative games. *Probl Kibern* 10 (1963) 119–139
5. Shapley, L.S.: On balanced sets and cores. *Naval Research Logistics Quarterly* **14**(4) (1967) 453–460
6. Owen, G.: On the core of linear production games. *Mathematical Programming* **9**(1) (Dec 1975) 358–370
7. Mashayekhy, L., Nejad, M.M., Grosu, D.: Cloud federations in the sky: Formation game and mechanism. *IEEE Transactions on Cloud Computing* **3**(1) (Jan 2015) 14–27
8. Khandelwal, Y., Purini, S., Reddy, P.V.: Fast algorithms for optimal coalition formation in federated clouds. In: *Proceedings of the 9th International Conference on Utility and Cloud Computing. UCC '16, New York, NY, USA, ACM* (2016) 156–164
9. Niyato, D., Vasilakos, A.V., Kun, Z.: Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach. In: *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. (May 2011) 215–224
10. Romero Coronado, J.P., Altmann, J.: Model for incentivizing cloud service federation. In Pham, C., Altmann, J., Bañares, J.Á., eds.: *Economics of Grids, Clouds, Systems, and Services*, Cham, Springer International Publishing (2017) 233–246
11. Samaan, N.: A novel economic sharing model in a federation of selfish cloud providers. *IEEE Transactions on Parallel and Distributed Systems* **25**(1) (Jan 2014) 12–21
12. Guazzone, M., Anglano, C., Sereno, M.: A game-theoretic approach to coalition formation in green cloud federations. *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)* (May 2014) 618–625
13. Fragnelli, V. In: *A Note on the Owen Set of Linear Programming Games and Nash Equilibria*. Springer US, Boston, MA (2004) 205–213
14. Innes, R., Sexton, R.J.: Customer coalitions, monopoly price discrimination and generic entry deterrence. *European Economic Review* **37**(8) (1993) 1569 – 1597