



Peer to Peer Energy Trade Among Microgrids Using Blockchain Based Distributed Coalition Formation Method

Subhasis Thakur¹ · John G. Breslin¹

Received: 18 October 2017 / Accepted: 22 April 2018
© Springer Nature Singapore Pte Ltd. 2018

Abstract

Unpredictable environmental conditions make renewable energy generators of a microgrid unreliable. Energy trade among microgrids may reduce dependency on utility grid (which uses traditional energy generator that causes massive greenhouse gas emission). Coalition formation algorithms are popular tools to execute energy trade among microgrids. In this paper we develop coalition formation method using the blockchain mechanism. Advantages of our solution over existing coalition formation algorithms for microgrid energy trade are as follows:

- *Distributed execution*: We show how to find coalition structure in a distributed fashion. It improves robustness of the computation compared with centralized executions of coalition formation algorithms. This is because failure of a centralized computing entity will lead to the failure of energy trade among all MGs but in a distributed solution such problem can be avoided.
- *Asynchronous execution*: In our solution, multiple coalition formation algorithms are executed asynchronously. Most of the existing solutions execute coalition formation algorithms at regular time intervals in synchronous fashion. Our solution reduces the waiting time for a microgrid to trade energy with other microgrids. Energy requirement of microgrids may depend on weather and changes in their energy requirement may be not synchronous, i.e., energy requirement of all microgrids may not change at the same time and after regular intervals.
- *Scalable*: We show that the distributed execution of coalition formation algorithm is more scalable than centralized algorithms for coalition formation.
- *Convergence*: We show that the proposed distributed algorithm converges quickly.
- *Local energy trade*: We show that the proposed distribution algorithm promotes local energy trade to reduce energy loss due to long transmission. It should be noted that feasibility of local energy trade depends on satisfaction of constraints in distribution network. Our solution aim to reduce the distance between matched energy supplier and energy provider if the distribution network allows such an energy transfer.
- *Trust*: The trust between a microgrid and the computation entity who is responsible for generating coalition structure is ignored in the existing solutions. We use a blockchain to eliminate such trust establishment requirement.
- *Security and Privacy*: The existing literature ignores the security of the computation framework. A MG's energy surplus / deficit reveals its energy consumption patterns and it may cause security problems. We mitigate such security problems using the blockchain mechanism.

Keywords Microgrid · Blockchain · Coalition formation

Introduction

Renewable energy generators reduce our dependency on burning fossil fuel for energy. A smart Micro Grid (MG) is equipped with renewable energy generators such as solar panels, wind turbine etc. These energy generators can be affected by environmental factors such as the amount of sunshine, the wind speed etc. and may not consistently

✉ Subhasis Thakur
subhasis.thakur@nuigalway.ie

John G. Breslin
john.breslin@nuigalway.ie

¹ National University of Ireland, Galway, Ireland

produce energy. Such reliability issues force a MG to buy energy from the utility grid who uses traditional energy generators such as coal powered thermal power generator and causes massive greenhouse gas emission. Energy trade among MGs is a potential solution to reduce the energy to be bought from such polluting energy generators of the utility grid. Trade methods are developed to support such energy exchange among MGs.

Multi-agent systems and game theory is widely used to model such trade methods. Autonomous agents are developed to represent the MGs and they take trade decisions on behalf of the MGs. The decisions taken by the agents depend on the model of interactions among the MGs. Both cooperative and non-cooperative game theory are used for this purpose. In non-cooperative games each MGs attempt to improve its own utility. In cooperative games MGs aim to maximize the social utility or overall benefit of all MGs. Our focus on this paper is multi-agent systems based coalition formation as energy trade method. Such a coalition formation procedure is as follows:

- Energy surplus or deficit information from the MGs are collected.
- Using such information coalition structure is generated. A coalition structure partitions the MGs into coalitions. MGs in each coalition trade energy among themselves.
- Optimal coalition structure generation depends on various parameters such as energy loss due to long transmission, total excess or deficit energy in each coalition among others.

The state of art in a coalition formation for energy trade among MGs has a few shortcoming:

- **Robustness:** Most of these approaches use centralized coalition formation algorithm and this is not a robust technique.
- **Trust:** A MG must trust the entity who executes the computation for coalition structure generation. Such an entity may be malicious and may generate wrong coalition structure. Hence a MG must evaluate its trust on such a computation entity. Note that MG can be the entity who performs the computation. Hence the trust among MGs must be evaluated as a malicious MG can alter the result of coalition structure generation problem for its own benefit. Existing solutions [2–4, 14] for MG energy trade ignore such trust evaluation.
- **Scalability:** Most of the existing coalition formation [2–4, 14] algorithms for energy trade among MG process are not scalable. This is because these approaches are centralized, i.e., there is a centralized computing entity who gathers information from MGs and yields coalition structure. Scalability can be measured as the relation between efficiency of the solution and

increase size of the problem. In this case, efficiency can be time needed to process the coalition structure and problem size is the size of MG network. Computation time depends on the input size of the algorithm (beside the complexity of the algorithm). If there is a linear relation between efficiency and problem size then such a solution can be regarded as a scalable solution. Input size to centralized algorithm can be arbitrarily high while in distributed algorithm it can be evenly distributed among several computing units.

- **Synchronous:** Most of the existing solutions execute coalition structure generation algorithm in a synchronous fashion. For example, such algorithm collects energy requirement information from the MGs in every one hour and executes the coalition structure generation algorithm after every one hour. But this procedure is not suitable for frequent and unpredictable changes in the environmental factors which affect energy generation in the MGs.
- **Security and privacy:** The existing solutions for MG energy trade ignore the security and privacy problems associated with energy trade among MGs. As the MGs share information on energy surplus and deficiency, it reveals their energy consumption patterns. It can be used to compromise privacy of such MGs. Security problems can be created by constructing wrong coalition structure, by overwriting the information on energy trade and using standard Denial of service attack on the computation infrastructure responsible for generating coalition structure.

In this paper we develop a blockchain based distributed algorithm for coalition formation among MGs to mitigate the above problems. Our main contribution is a blockchain based distributed, scalable and secure computation infrastructure for coalition structure generation among MGs. Blockchain [15] mechanism allows us to securely store transaction records among peers of a peer to peer network. Security of blockchain maintained transaction record is guaranteed by its following characteristics:

- **Blockchain Data structure:** Blockchain uses a particular data structure known as UTXO (unspent transaction output). According to which inputs to a new transaction must be outputs of other transactions which have not been used as input to any other transaction. Also, blockchain groups a bunch of valid transaction as a block and maintains blocks with a time stamped parent-child relation. The advantage of this approach is that, to overwrite one transaction a peer has to prove validity of other transactions in the same block or child blocks. This problem is known to be computationally infeasible. Hence it is highly unlikely that data on blockchain

can be overwritten and the data kept in a blockchain is secure.

- **Consensus:** In blockchain a transaction is regarded as a valid transaction if all peer reach consensus that it is a valid transaction. Hence one or a handful of peers can not label an invalid transaction as a valid transaction.

As a result of these properties, blockchain eliminates the requirement of a trusted third party to verify a transaction between two parties. In this paper we propose a blockchain based solution for coalition formation among MGs. MGs form a blockchain peer to peer network and energy requirement (supply or demand) are announced as transactions from one MG to another MG. In this distributed algorithm any MG can execute the coalition formation algorithm on its unspent transactions i.e., supply or demand requirements it has received so far.

If it can find a solution to the coalition structure generation then its solution is added to blockchain and energy trade among corresponding MGs are executed using smart contracts. Otherwise the MG forwards its unspent transactions to a random neighbouring MG. Using experimental evaluation we show this distributed blockchain based coalition formation algorithm performs better than a centralized algorithm.

Note that blockchain peer to peer network is created by recognizing two MGs as neighbours if they are in close proximity. Being part of each others neighbourhood in the blockchain peer to peer network does not imply they are in a coalition. Coalition is decided by algorithms mentioned in “[Blockchain Based Solution for MG Management Problem](#)”.

The paper is organized as follows: In “[Related Literature](#)”, present a small review of relevant literature. In “[Coalition Formation for MG Energy Exchange](#)”, we discuss the coalition formation process for the MG management problem. In “[Blockchain Basics](#)”, we present a brief description of the blockchain mechanism. In “[Blockchain Based Solution for MG Management Problem](#)”, we show the blockchain based distributed coalition formation process. In “[Discussion](#)”, we discuss few implementation issues of such blockchain based coalition formation algorithm. In “[Experimental Analysis](#)”, we present experimental evaluation of the proposed solution and we conclude the paper in “[Conclusion](#)”.

Related Literature

Multi-agent systems is a common tool to implement energy trade among MGs. [6] used a multi-agent system to execute operational controls of MGs. Specifically they focus on optimal usage of local resources to support local loads within a MG. [13] used a multi-agent system to implement coordinated control for energy management among MGs.

[5] used a multi-agent system to implement auction as trade mechanism for energy trade among MGs. Similarly [7, 10] used multi-agent systems to develop control system for MG energy trade. [11] used multi-agent system for distributed energy resource management in MGs. [8] used a multi-agent systems for energy storage management which allows energy suppliers and consumers to coordinate their actions.

Coalition formation is a common tool to implement energy trade among MGs represented as a multi-agent systems. In such a solution MGs are partitioned into coalitions and MGs in each coalition trade energy among themselves. Optimality of such a partition is determined by various factors such as total energy surplus or deficiency in each coalition, energy loss in each coalition due to long transmission among others. [2] developed a cooperative energy exchange scheme in order to reduce the battery usage as well as improving the efficiency of houses locating in remote communities. [3] proposed hierarchical coalition formation for energy distribution among microgrids. The cooperative game theory based solutions need to find the stable coalitions and also need to find the energy sharing in each coalition. In most of these solutions Shapley Value is used to determine energy sharing among MGs in a coalition. [14] proposed dynamic coalition formation for this problem. [4] finds optimal coalition structure based on certain optimization criteria of the coalitions. [21] developed dynamic coalition formation algorithm that minimizes discomfort in MGs. [19] proposed trading strategies for energy market among MGs. [18] presented a learning strategy for the broker agent in the smart-grid market. [16] also proposed how the broker agent can learn the bidding price in the smart grid market. [17] proposed learning mechanisms for the electric suppliers to bid in the electric market. [12] proposed energy trading protocol based on a energy trading policy. [1] proposed multi-agent negotiation protocol for energy trading. [20] showed intelligent usage of energy storage to facilitate improvement of energy trading among MGs.

Bitcoin [15] introduced blockchain technology in 2008. Blockchain securely records transaction history among a set of peers in a peer to peer network over a chosen set of peers. Blockchain eliminates the need for a third party for establishing trust among two interacting parties such as banks. The first blockchain mechanism uses proof of work as the distributed consensus protocol. Peercoin (<https://peercoin.net/>) introduced the proof of stake protocol which use stake as the vote power instead of computing resource [9].

The existing research on MG energy trade shows that usage of multi-agent systems and coalition formation for energy trade among MGs can be useful and highlights the feasibility of implementing control systems and flexibility of energy distribution networks to support energy trade

among MGs. Yet, there are few shortcoming with coalition formation for MG energy trade as follows:

- Existing coalition formation solution is centralized and hence it is not robust as failure of the centralized computing entity may stop energy trade for all MGs. To mitigate this we propose a distributed solution.
- Existing solutions are synchronous as all MGs declare their energy needs and surplus are the same time and after regular interval. But availability of renewable energy may depend on weather and changes in energy requirement for all MGs may not occur at the same time and at regular intervals. To mitigate this we propose an asynchronous solution.
- The existing coalition formation solution ignores the requirement to evaluate trust among MGs. But if a MG collects energy requirement information from other MGs and decides the coalition structure then it must be trusted by all MGs to perform correct computation. Similarly if the computing entity is a centralized third party then MGs must evaluate their trust on it as it may be malicious. In this paper we propose a solution that erases the requirement to evaluate trust among MGs or on a third party.
- Scalability of coalition formation may be an issue with centralized coalition formation. But we show that our distributed approach is more scalable.
- Current coalition formation algorithms for MG energy trade ignore security and privacy problems in this trade. We use blockchain to make a more secure solution which also enhances privacy of MGs.

Coalition Formation for MG Energy Exchange

In this section, we describe the coalition formation process [3] for energy trade among MGs. We denote n MGs as $MG = M_1, \dots, M_n$. We denote discrete time instants as t_0, t_1, t_2, \dots where each consecutive time instances are dt time duration apart, i.e., $t_{i+1} - t_i = dt$. For each MG MG_i , D_i^j and S_i^j denote its predicted energy demand and energy supply at time t_j for next dt time duration. This means from time t_j to time t_{j+1} , the MG M_i will have energy demand D_i^j and based on the prediction of environmental factors expected energy production from its renewable energy generators is S_i^j . The energy requirement of a MG M_i at t_j is $E_i^j = D_i^j - S_i^j$. If E_i^j is positive then it implies that M_i is expected to produce more energy than the amount of energy it will consume for next dt time duration and it can sell such excess energy to other MGs. If E_i^j is negative then it implies that M_i is expected to produce less energy than the amount of energy it will consume for next

dt time duration and it may buy energy from other MGs. Utility of a MG M_i at time t_j is

$$U_i = \frac{1}{1 + |D_i^j - S_i^j|}.$$

Note that U_i increases as $|D_i^j - S_i^j|$ decreases. $|D_i^j - S_i^j|$ indicates either amount of surplus energy or amount of energy deficiency. The objective of coalition formation is to minimize $|D_i^j - S_i^j|$ for all MGs, i.e., coalition formation aims to minimize loss of surplus energy and energy deficiency.

There is a energy distribution network that connects the MGs. We denote such a network as an undirected graph $G = (M, E)$ where E is the set of edges. $W(e \in E)$ will denote the distance of the edge e . Energy loss due to transmission will be depend to the distance between two MGs.

Definition 1 A coalition structure Π for the set of MGs $M = \{M_i\}$ is a partition over them as $\Pi = (C_1, \dots, C_k)$ such that $C_i \subset M$, $\cup C_i = M$ and $C_i \cap C_j = \emptyset$ for any $C_i, C_j \in \Pi$.

MGs in a coalition trade energy among themselves. Next we define utility of a coalition in a coalition structure as follows:

Definition 2 The utility of a coalition $C \subset M$ is:

$$U_C = \frac{1}{1 + E_C + L_C}$$

where,

$$E_C = \sum_{M_i \in C} |D_i - S_i|$$

and L_C denotes the loss of the coalition C . L_C is a measure of loss of utility due to loss of energy due to transmission. L_C is calculated as follows:

- Let the F be all edges among any two $M_i \in C$ and $M_j \in C$ or edges among any micro grid in C and the utility grid UG .
- The loss of utility (due to the loss of energy) of an edge $e \in F$ is calculated as follows:

$$L(e) = \left[\frac{P(e)}{\Sigma} \right]^2 \times \alpha \times W(e) + \beta \times P(e)$$

where $P(\cdot)$ is the power required to move the energy, Σ is the carrying voltage of the transmission line, $W(e)$ is the length of the edge e and β is the transformer loss. The amount E will be decided after forming the

coalition (using Shapley value energy distribution on each edge in a coalition will be decided)

– Finally,

$$L(C) = \sum_{e \in E} L(e).$$

Coalition structure should be generated to maximize total utilities of the coalitions in a coalition structure. We will use shapley value for fair energy distribution in each coalition. Shapley Value for MG_i is:

$$\theta_i = \sum_{S \subseteq MG - MG_i} \frac{|S|!(|MG| - |S| - 1)!}{|MG|!} (v(S \cup i) - v(S))$$

The characteristics function v is defined as follows:

$$v(C \subseteq 2^{MG}) = \sum_{MG_i \in C} U_i - U_C \quad (1)$$

Blockchain Basics

In this section we present a brief description of the blockchain mechanism. Blockchain records transactions among peers of a peer to peer network where records of transactions validated and stored by peers. In blockchain, a peer announces a transaction to transfer funds to another peer. Such a transaction is validated and recorded by all peers. For example as shown in Fig. 1 A wants to send x\$ to G. A announces transaction of x\$ with G as the recipient to its neighbors B and C. B and C validate and forwards the transactions to their neighbors D and E. A transaction is valid if it is accepted and stored by all peers.

The Blockchain mechanism ensures security of the data and it prevents double spending i.e., fund can be spent only once. Network delay may create conflicting history of transactions by creating forks in the blockchain. Distributed consensus protocol of the blockchain mechanism ensures that all peers agree on one version of transaction history.

• **Transaction data structure** Blockchain uses Unspent Transaction Output (UTXO) data structure for transactions to prevent double spending. An example of UTXO (shown in Fig. 2) is as follows:

- Input of transaction created by A is the set of transactions T1,T2,..Tk.
- A is the recipient of T1,T2,..Tk.
- All transactions T1,T2,..Tk are unspent,i.e., not used as input to any other transaction.
- The amount of this transaction, i.e., x is at least same as the total amount of all transactions T1,T2,..Tk.

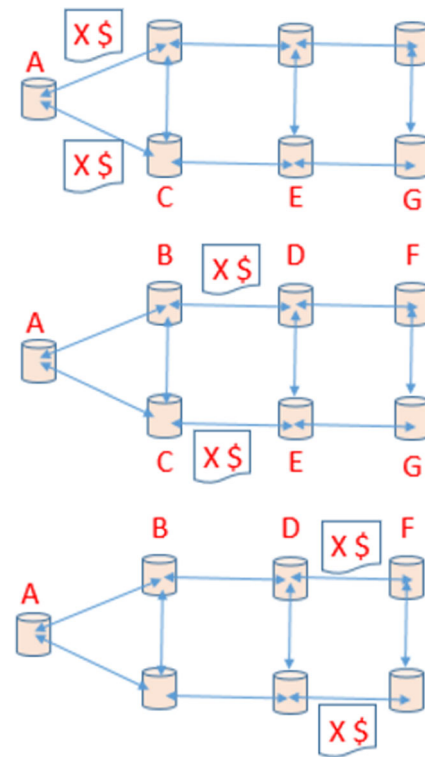


Fig. 1 Transaction in blockchain

After construction of the above transaction A identifies G by its public key and A signs the transaction with its private key. This ensures that A indeed wants to create this transaction and G is a correct recipient.

• Distributed consensus protocol

A block is a collection of valid transactions. A Blockchain is as chain of blocks where each block has a parent block except first block. The Block whose distance from the first block is maximum is regarded as the blockchain Head. The path from blockchain head to first block is regarded as the correct version of transaction history. Any peer can add a new block to the blockchain. Two peers may add a new block to the blockchain almost at the same time and network delay will cause a few peers to adopt one block as

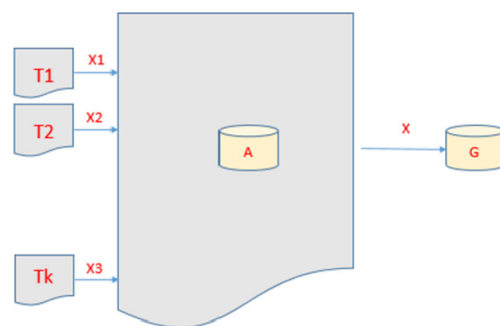


Fig. 2 Transaction data structure

a new blockchain head while the rest of the peers will regard the other block as the blockchain head. This phenomenon is called a fork in the blockchain. Distributed consensus protocol resolves forks. Proof of work based consensus protocol requires each peer to solve a puzzle (generate a Hash with particular pattern) before adding a new block to the blockchain. Proof of stake based consensus protocol uses low complexity puzzles (Fig. 3).

Blockchain Based Solution for MG Management Problem

The blockchain based distributed coalition formation algorithm (Fig. 4) is as follows:

- We form a blockchain peer to peer network from the MG network, i.e., if two MGs are neighbours in the MG network then they are neighbours in the blockchain peer to peer network.
- A MG announces its energy requirement (supply or demand) as a blockchain transaction and sends it to a neighbouring MG. There are two types of tokens, demand tokens and supply tokens to represent energy need of the MGs. A transaction of x demand tokens from MG M_i to MG M_j represents that M_i need x units of energy for the next dt (as described in “Coalition Formation for MG Energy Exchange”) time durations and M_j may find a MG with surplus energy to satisfy this energy need. Similarly, a transaction of x supply tokens from MG M_i to MG M_j represents that M_i wants to sell x units of energy for the next dt time durations and M_j may find a MG with energy deficit to sell this surplus energy. Hence unspent transactions

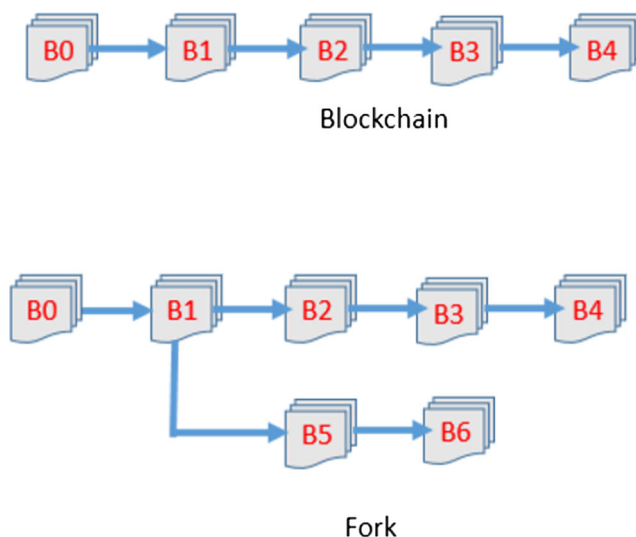


Fig. 3 Blockchain and forks

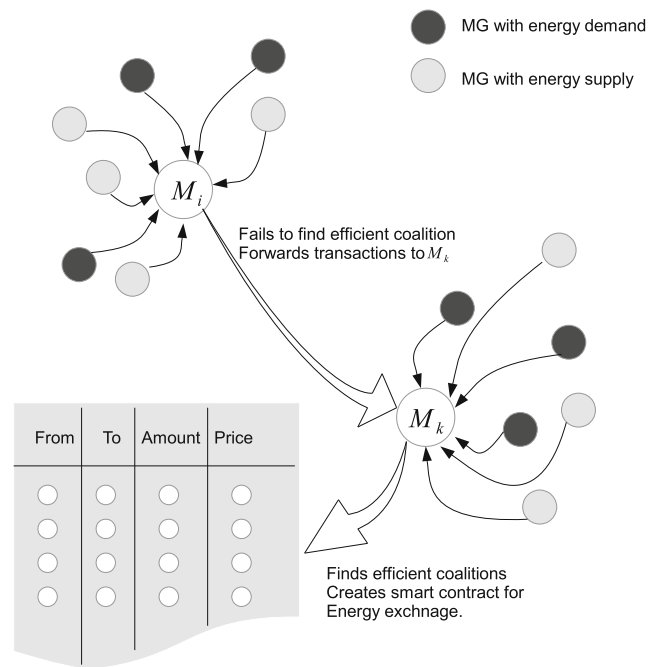


Fig. 4 Overview of the blockchain based solution for the MG management problem

of a MG indicate the collection of energy requirement requests from other MGs.

- Any MG can execute the coalition formation algorithm on its unspent transactions. This means a MG attempts to find coalition structure among MGs corresponding to its unspent transactions. If it can find a coalition structure in its unspent transactions then it generates a block which contains a transactions representing solutions of its coalition structure generation algorithm. Using such transactions relevant MGs create smart contracts which contain rules such as:

If M_i receives x units of energy from M_j within next y time duration then x' fund (i.e., BitCoin or Ether) should be transferred from M_i to M_j as the price for such energy transfer.

The MG announces such a smart contracts to the blockchain and it is executed with additional information from the energy grid such as ‘ M_i receives x units of energy from M_j within next y time’ which will trigger execution of the transaction ‘Transfer X units of fund from M_i to M_j ’.

- We will use proof of work as distributed consensus protocol. We introduce additional rules for transaction verification which ensure coalition structures with maximum social utility are generated. Transactions are stored as block and regarded as a legal solution to the coalition formation problem.

Next we provide a detailed description of this procedure.

Blockchain Network

First we create the blockchain peer to peer network as an undirected connected graph $G = (M, E)$ where $M = \{M_i\}$ be the set of MGs and E is the set of edges. For each MG M_i we add a fixed number of closest MGs as its neighbours. Hence we create a peer to peer network topology according to physical proximity of the MGs.

Transactions and Blocks

In this section we discuss blockchain's transaction structure to execute coalition structure generation algorithm. First, we will discuss modification to transaction data structure, then we will discuss transaction creation process after solving the coalition structure generation problem, next we will discuss the transaction forwarding procedure in case a MG fails to find coalition structure for its unspent transactions and finally we discuss block creation process.

Transaction Data Structure

We use two types of tokens, demand tokens and supply tokens. Note that a cryptocurrency token may remain alive (i.e., can be used as input to nother transaction) infinitely. But life span of tokens representing energy demand and supply must be restricted because energy demand or supply do not persist indefinitely. Hence we need to terminate (which ensures that a transaction can not be used as input to another transaction) demand and supply transactions after certain time duration. We make to following modifications to the blockchain's transaction data structure:

- **Type:** 1 will indicate that it is a demand token and 0 will indicate that it is a supply token.
- **Origin:** Any MG can create mint demand and supply tokens. It can forward such tokens to a MG in its blockchain peer to peer network neighbourhood. Transaction data structure will include a field 'origin' to represent the MG who had created the token.
- **Trial:** Any demand or supply transaction is terminated after a constant time duration. Trial will include the number of times a supply or demand information (as a sequence transactions where one transaction is used as the input of another transaction) is used to find coalition structure. θ will indicate the maximum trail number allowed. It means a θ is the maximum number of attempts to a energy demand requirement with one or more supply requirement.
- **Terminated:** 1 indicates that the transaction is terminated and can not be used as input to another transaction. Otherwise it is 0.
- **Matched:** It will include transaction ID of a complementary transaction (discussed in detail at the next section).

A transaction will be represented as the token

$\tau(< Types >, < Origin >, < Trail >, < Amount >)$.

τ^i will indicate the unspent transactions of the MG M_i .

Transactions for Coalition Structure Generation

In this section we will describe the hierarchical coalition formation algorithm [3]:

1. Initially every MG M_i acts as a coalition.
2. Divide the MGs into groups PG (MGs with positive energy supply and in descending order) and NG (MGs with negative energy supply and in ascending order).
3. For each MG M_i in NG we find the energy loss if energy is supplied by the MGs in PG .
4. We assign a score to each MG in PG based on the energy loss due to transmission.
5. We add MGs in the PG as member of the coalition containing the MG M_i if the energy loss is not significant.

Algorithm 1: Coalition structure generation algorithm

Data: $G = (M, E)$ as the blockchain network,
Locality of the miner $Locality(M_x)$, $R = \{R_i^j\}$
as the set of unresolved ordered transaction.

Result: Coalition structure CS

begin

```

     $CS \leftarrow$  coalition structure as list of
     $|Locality(M_x)|$  lists
     $PG \leftarrow$  subset of  $Locality(M_x)$  with positive
    energy supply and in descending order
     $NG \leftarrow$  subset of  $Locality(M_x)$  with negative
    energy supply and in ascending order
    for Each  $M_i \in NG$  do
         $Loss$  is a  $|PG|$ -element list with energy loss if
         $M_j \in PG$  supplies energy to  $M_i$ 
         $Score$  is a  $|PG|$ -element list as
         $Score[M_j] = Energy[M_j] - Loss[M_j]$ 
         $OrderedPG$  order  $PG$  with descending value
        of  $Score$ 
        for  $M_j \in OrderedPG$  do
            if  $Score[M_j] > 0$  then
                Add  $CS[M_j]$  to  $CS[M_i]$ 
                 $CS[M_j] \leftarrow \emptyset$  Subtract
                 $Min(Energy[M_i], Energy[M_j])$ 
                from  $Energy[M_j]$ 
                Add  $Min(Energy[M_i], Energy[M_j])$ 
                to  $Energy[M_i]$ 
            Update the list  $PG$ 
        if  $Energy[M_i] \geq 0$  then
            Break

```

Next a MG generates energy exchange records in each coalition of the coalition structure generated by Algorithm 2. The algorithm is as follows:

1. Similar to the previous algorithm we generate ordered set of MGs with positive and negative energy supply for each coalition as PG and NG .
2. We sequentially satisfy the energy requirement of each $M_i \in NG$ and record the energy exchanges.
3. If the energy requirement of a MG can not be fully satisfied then we use the utility grid for energy and record such transaction.

Algorithm 2: Algorithm to decide energy exchange in each coalition

Data: $G = (M, E)$ as the blockchain network, CS as the coalition structure

Result: Energy exchange records

begin

```

 $PG \leftarrow$  subset of  $Locality(M_x)$  with positive
energy supply and in descending order
 $NG \leftarrow$  subset of  $Locality(M_x)$  with negative
energy supply and in ascending order
for Each  $M_i \in NG$  do
  for Each  $M_j \in PG$  do
    Add 'Transfer
     $MIN(Energy[M_i], Energy[M_j])$  from
     $M_j$  to  $M_i$ '
    Subtract  $Min(Energy[M_i], Energy[M_j])$ 
    from  $Energy[MG_j]$ 
    Add  $Min(Energy[M_i], Energy[M_j])$  to
     $Energy[MG_i]$ 
    if  $Energy[M_i] \geq 0$  then
      Break;
  if  $Energy[M_i] < 0$  then
    Add 'Transfer ( $Energy[M_i]$ ) from  $UG$  to
     $M_i$ '
    Add  $Min(Energy[M_i], Energy[M_j])$  to
     $Energy[MG_i]$ 

```

After executing the above algorithms to generate coalition structure and corresponding energy exchanges, a MG uses the following transaction creation procedure: Let $\{\tau^i\}$ be the set of unspent transactions that the MG M_i uses to solve the coalition structure generation.

- For each outcome "Transfer x units of energy from M_x to M_y it creates a transaction τ_j whose inputs are $\tau_a \in \{\tau^i\}$ and $\tau_b \in \{\tau^i\}$ such that:
 - Origin of τ_a is M_x and its Type is 0.
 - Origin of τ_b is M_y and its Type is 1.

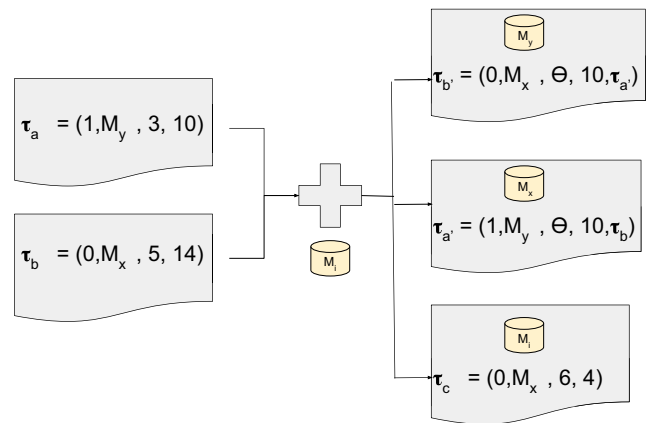


Fig. 5 Transaction creation process after solving coalition structure generation. Unspent transactions τ_a and τ_b are used to generate coalition structure. τ_a indicates that MG M_y needs 10 units of energy and τ_b indicates that M_x wants to sell 14 units of energy. M_i creates $\tau_{b'}$ as it transfers 10 energy supply tokens to M_y where M_x is marked as the origin, i.e., M_x will transfer the actual energy. Similarly, M_i creates $\tau_{a'}$ to send 10 demand tokens to M_x where M_y is marked as the origin, i.e., M_y will receive the actual energy. In both transactions Trail is θ . This indicates these transactions can not be further used as inputs to other transactions. Also, 'Matched' field of $\tau_{b'}$ is labelled with $\tau_{a'}$ and vice-versa. This helps the MGs to formulate the smart contract. Finally the excess supply tokens are sent back to M_i itself

- Amount of transfer of τ_j is less than $Min(Amount(\tau_a), Amount(\tau_b))$.
- M_i also sends transaction $\tau_{a'}$ and $\tau_{b'}$ to itself by increasing the trail number to θ .

Figure 5 shows an example of such transaction creation. Note that we increase the trail number to θ to prevent further usage of these transactions.

Smart Contract Creation

After receiving supply or demand tokens corresponding to coalition structures, the MGs establish smart contracts for energy trade. It is as follows:

- A MG checks if it has received supply or demand tokens in transactions where 'Matched' field is not empty and 'Trail' number is θ .
- Continuing from the example shown in Fig. 5, M_y received 10 supply tokens and M_x received 10 demand tokens.
- M_x and M_y engage in a smart contract creation process. A smart contract¹ is an executable code in the blockchain network. It is a set of transactions which can be triggered if certain conditions are satisfied. In this example such condition is M_y received 10 units of energy.

¹<https://en.bitcoin.it/wiki/Contract>

Transaction Forwarding

If a MG fails to find coalition structure using the hierarchical coalition formation algorithms discussed in previous section then it will send its unspent transactions to a neighbouring MG. Such transaction forwarding process is as follows: A MG M_i forwards an unspent transaction $\tau_x = (1, M_z, 10, 70)$ as follows:

- It chooses a neighbour M_j uniformly at random.
- It creates a transaction $\tau_y = (1, M_y, 11s, 70)$ where input transaction is τ_x and destination is M_y .
- Note that we trail number of τ_y is 1 more than the same for τ_x .

Block Creation and Distributed Consensus Protocol

In this section we discuss block creation process according to a distributed consensus protocol. A block is a set of verified transactions. A MG verifies a transaction according to the following rules:

- **Mint transaction:** Analogous to mint tokens, mint transactions are transactions without any input transactions. We assume existence of entities *Demand_Base* and *Supply_Base* who will issue mint tokens to create demand and supply transactions as MGs announce their energy requirements.
- **Forwarded transaction:** A forwarded transaction has the following pattern:
 - It has only one input transaction and
 - its ‘Type’ is same as the ‘Type’ of input transaction.

A forwarded transaction is valid if the following holds:

- Its trail number is 1 more than the same for its input transaction.
- Trail number of input transaction is θ .
- ‘Terminated’ field of the input transaction is 0.
- **Coalition structure transaction:** A coalition structure transaction has the following pattern:
 - There is at least two input transactions with different ‘Type’.
 - ‘Terminated’ field of all input transactions are 0.
 - ‘Trail’ of all input transactions are less than θ .
 - Amount of a transaction (say number of supply tokens) to a MG M_j is at most the total demand tokens for input transactions where *Origin* is M_j .

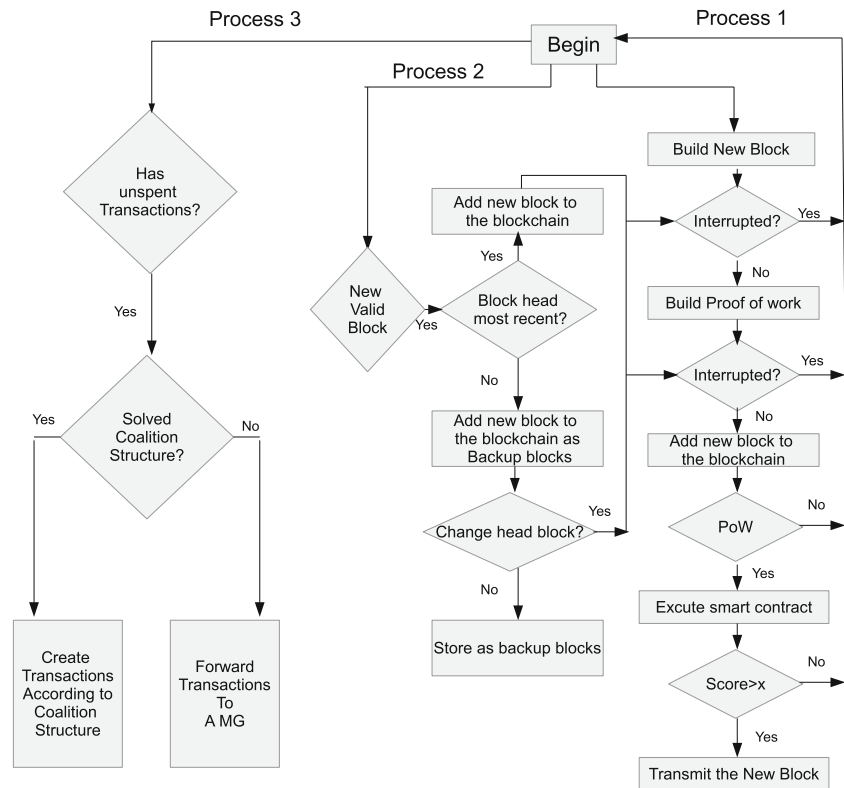
The block creation process in the blockchain mechanism is as follows:

- Each peer verifies transactions and group verified transactions in a block and announces such a block to the peer to peer network. Each block contains a parent block information which is the latest block of the blockchain.
- If a peer receives a block then it verifies it and performs the following steps:
 - If the parent block of the new block is the latest block known to it then it augments its blockchain with the new block otherwise it add the block as child of its parent block.
 - If the parent block of the new block is not the latest block known to it then it add the block as child of its parent block. If such a branch of the blockchain becomes longest (i.e., distance from the first block to the last block of this branch of the blockchain) then it treats this branch of the blockchain as the correct history of the blockchain.

Workflow of a MG

In this section we describe the workflow of each MG. As show in Fig. 6, workflow of miner has three processes. These processes are executed in parallel and may interrupt each other.

- **Process 1:** This process attempts to create a new block. A new block is a set of new transactions (not yet added to the blockchain). The block creation process is describe in “[Transaction Forwarding](#)”. This process can be interrupted by process 2.
- **Process 2:** This process a MG adds a new block created by another MG to the blockchain. After receiving a new block a MG first verifies the transactions in the new block. If it can verify that all transactions of the new block is valid then it use the following procedure:
 - If the new block’s parent block is the most recent block know to the MG then it adds the new block as the child of the most recent block of the blockchain and label the new block as the most recent block. This process interrupts Process 1 as the block creation process is restarted.
 - If the new block’s parent block is the not most recent block know to the MG finds such a parent block in the blockchain and adds the block as child block of that blockchain. Note that such a new block is not considered as valid block and such incident is regarded as a blockchain fork. If by adding the new

Fig. 6 Work flow of a MG

block to the blockchain, the distance (shortest path) from the new block to the first block is more than the same for the most recent block known to the MG then it labels such a branch of the blockchain (where last block is the new block) as the correct record of transaction history. This process is described in “[Blockchain Basics](#)”.

- **Process 3:** This process generates coalition structures. If there are unspent transactions then it attempts to solve the coalition structure generation problem as discussed in “[Transactions for Coalition Structure Generation](#)”.

Discussion

In this section we discuss few implementation issues for blockchain based coalition formation for energy trade among MGs.

Energy distribution network: The current energy distribution network does not completely support immediate implementation of peer to peer energy trade. Although [5–7, 10, 13] have shown how MG energy trade can be implemented, more research in energy distribution network is need to accommodate such trade for massive energy distribution network. A potential solution is to

create an energy distribution network dedicated to peer to peer energy trade. Such a solution is immediately applicable for isolated locations which are not connected with energy grid. Such a dedicated energy distribution networks can also be used in developing countries where there is no electricity distribution network. But it should be noted that development of dedicated energy distribution network should be evaluated for its financial feasibility. If we assume that MGs will pay for such distribution network and they will be rewarded with mint tokens from the blockchain then, we must estimate the financial benefit of such incentives to the MGs. The simulations developed in this paper can be used to perform such a study. We want to explore such a study on financial feasibility in the extension of this paper.

Who owns the blockchain? In this paper we proposed a blockchain based energy trade mechanism. Thus a natural question arises that who owns such a blockchain peer to peer network?. There are few solutions:

- **Public blockchain:** We may use a public blockchain network such as Ethereum. There may be certain privacy and security issues with using public blockchain to keep record of energy trade as these ledgers are accessible to all and may reveal energy consumption patterns. MGs can act as miners in such a blockchain. Certain financial benefits similar to blockchain mining should be added to the blockchain

based trading mechanism to cover the cost of equipments for mining.

- Private blockchain: We may also use a private blockchain for peer to peer energy trade. A potential implementation of such a private blockchain is as follows: Energy providers (may not be renewable energy), energy distribution network maintainers may act as the miners (they invest in mining equipments) and MGs may only have blockchain wallets.

Payment system: We propose two payment schemes.

- Cryptocurrency: We can immediately use cryptocurrency for payments. We may use a public blockchain such as Ethereum or Bitcoin where the peer to peer energy trade will be implemented as a sidechain. Although this method can be easily implemented but fluctuating exchange rate for cryptocurrency may be an obstacle.
- Compensation by energy providers: In another solution the traditional energy providers may invest in building the energy distribution network for peer to peer energy trade and may also act as miners. Houses may form microgrids to participate in peer to peer energy trade and they may also retain connection to energy distribution network which serves traditional energy sources. In such a settings, a microgrid may be affiliated with an energy provider. Suppose microgrid A is affiliated with energy supplier C . Microgrid A may send energy to another MG B on behalf of energy supplier C . C will adjust A 's electricity bill based on the energy A transferred to B .

Forks: Delay in blockchain communication network may cause blockchain fork. As discussed in “[Coalition Formation for MG Energy Exchange](#)”, forks are resolved by discarding one branch of the blockchain as the transactions in the discarded branch is again processed. It is possible to execute such reversal operation for cryptocurrency. But, once energy is transferred it not possible to reverse it. A possible solution is to completely ignore transactions in the discarded branch of the blockchain. Note that it takes time to resolve a fork, for example in Bitcoin it may take time up to 6 block creation to resolve a fork. This is estimated to be close to one hour. Energy surplus and deficiency are time dependent. Hence the life-span of corresponding transactions are limited. Thus we may completely ignore transactions in discarded blocks are energy requirement information in such blocks are no longer valid.

Security: Blockchain based distributed coalition formation method is more secure than existing methods for the same problem because:

- Energy surplus or deficiency information are announced using blockchain transactions. The associated encryption technology (private/public key) ensures authenticity of such information.
- UTXO data structure ensures appropriate existence of appropriate energy surpluses for energy deficiencies.
- Distributed consensus protocol insures that every MG correctly computes the coalition structure.

Trade of between security and speed: Security of the blockchain maintained solution depends on size of the network. Larger the size more secure the solution. This is because a malicious entity needs 50% of all computation capability of the blockchain peer to peer network. In small network a MG may have such computation power. But speed of computing the coalition structure decreases as the network grows. This is because input size of coalition formation algorithm increases with larger network.

Algorithm 3: Algorithm to Simulate Distributed Coalition formation

Data: Blockchain peer to peer network

Result: Energy trade records

begin

for Every iteration **do**

for Every MG M_i **do**

if $\text{Modulus}(\text{Number-of-Iteration}, 5) == 0$

then

$\text{Energy}_i \in [-100, 100]$ chosen

 uniformly at random

$M_j \in \text{Neighbour of } M_i$ chosen

 uniformly at random

$\tau(M_i, M_j, \text{Energy}_i)$ Send transaction to M_j

if There is unspent transactions $\{\tau^i\}$ for M_i

then

 Solve coalition structure generation for $\{\tau^i\}$

if Successfully solved coalition

 structure generation **then**

 Create smart contract to announce

 energy trade among MGs

else

$M_j \in \text{Neighbour of } M_i$ chosen

 uniformly at random

 Send transactions to $\{\tau^i\}$ M_j

Experimental Analysis

We simulate the proposed blockchain based solution for energy trade among MGs using agent based modelling. We use ‘SimPy’ library in Python to simulate asynchronous processes. We simulate the peers of a blockchain peer to peer network as autonomous agents and workflows of such agents are modelled using separate asynchronous processes. We use three datasets with 300, 500 and 700 MGs. Each pair of MGs have fixed distance between them.

Algorithm 3 describes the simulation procedure. It is as follows:

1. A MG sends its energy requirement information to a random neighbour. We form a neighbourhood among MGs by choosing 5 closest MGs of a MG as its neighbour.
2. Lines 4 to 7 describe the process of creating transactions to announce energy surplus or deficiency. Note that in a synchronous settings all MGs after every 5 iterations announce their energy requirement. In asynchronous settings, we randomly choose a subset of MGs to announce their energy requirement at every iteration.
3. Lines 8-14 describe the process of solving coalition structure generation problem. If a MG can solve it then, it announces a smart contract which describes energy transfer between the MGs. Otherwise it sends the unspent transactions to a neighbouring MG. Note that unspent transactions indicate energy surplus or deficiency information are not used in computation of coalition structure.
4. Lines 4-14 describe workflow for each MG. Note that we execute this part of the algorithm as separate asynchronous process for each MG.

First we compare the performance of the distributed coalition formation method against a centralized coalition formation algorithm [3] when the energy demands and surpluses are announced synchronously, i.e., at time instant t_i , all MGs announce their energy deficit and surplus till the next time instant t_{i+1} . For example, at 2 PM all MGs announce how much energy they want to sell or buy for next 1 hour. Based on such information, coalitions must be formed to trade energy for 1 hour approximately from 2 PM to 3 PM. Again at 3 PM they will announce their energy requirements.

First we analyse the convergence time of the distributed coalition formation mechanism which is an iterative process. If a MG fails to find coalition structure with its unspent supply and demand requests then it forwards such supply and demand requests to a MG in its neighbourhood. Such a neighbour attempts to solve the coalition formation problem at the next iteration. Hence we measure the convergence time of the coalition formation process as the

number of iterations needed to match each demand request to one or more supply requests. The convergence time for MG networks with 300, 500 and 700 MGs are shown in Figs. 7, 8 and 9 respectively. We found average convergence times for these networks are 4, 5 and 21. Hence it proves that the distributed coalition formation process terminates quickly.

Next, we compare utility of the coalitions formed by the distributed and the centralized coalition formation mechanism. Figures 10, 11 and 12 show the utility of coalitions formed by the distributed coalition formation mechanism compared with average utility of the coalitions formed by the centralized coalition formation mechanism. We found that on average utility of coalitions formed by the distributed algorithm is more than the same for the centralized coalition formation mechanism. Our distributed coalition formation algorithm attempts to find coalition among MGs in proximity if it fails to find such coalition in proximity then it gradually includes MGs located at distant locations in the order of increasing distances. Hence it reduces energy loss due to long transmission which is considered a critical parameter for measuring utility of coalitions.

Next we analyse scalability of the distributed coalition formation mechanism. We measure scalability as the input size (number of supply and demand requests) per

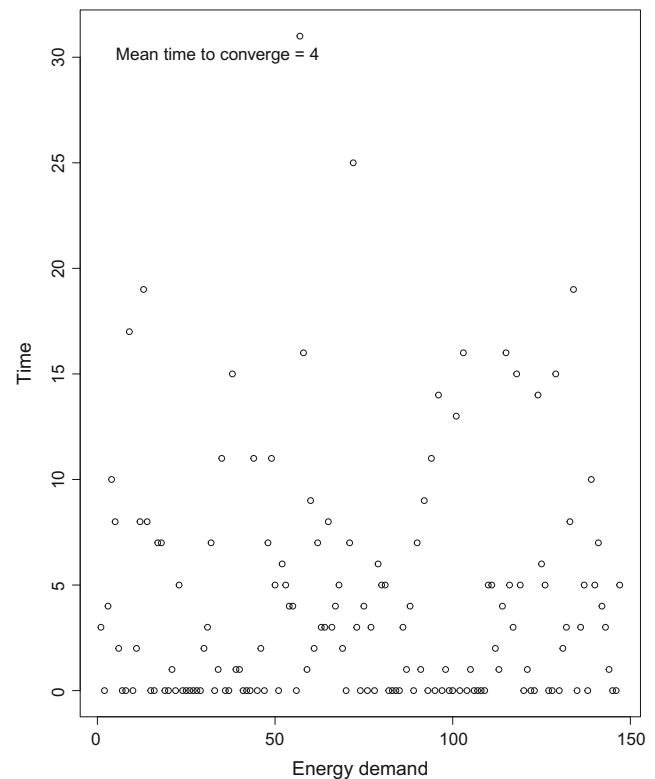


Fig. 7 Convergence time for distributed coalition formation process in MG network with 300 MGs

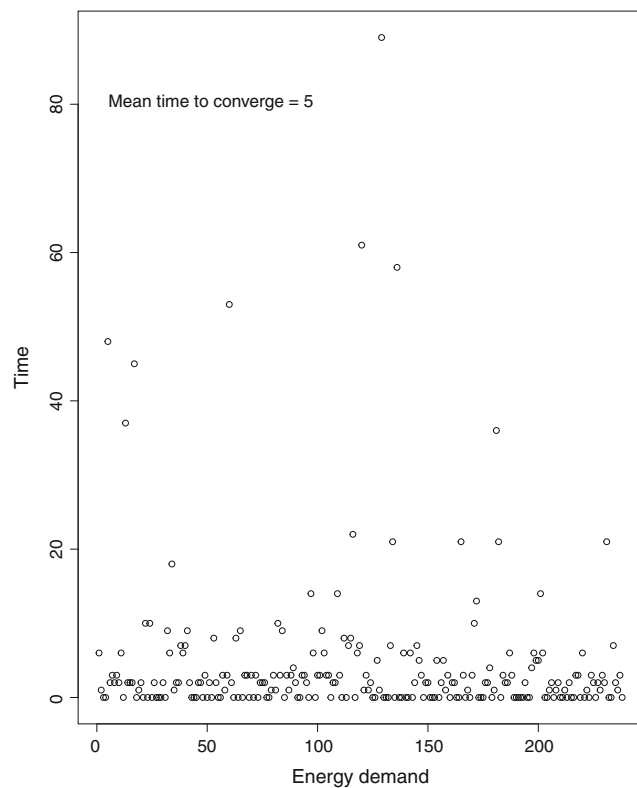


Fig. 8 Convergence time for distributed coalition formation process in MG network with 500 MGs

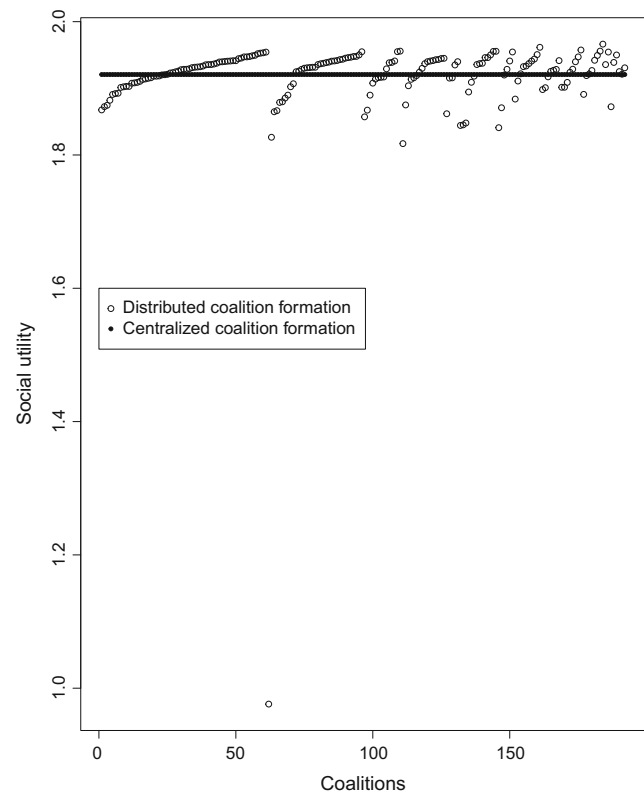


Fig. 10 Utility of coalitions formed by the distributed coalition formation process in MG network with 300 MGs

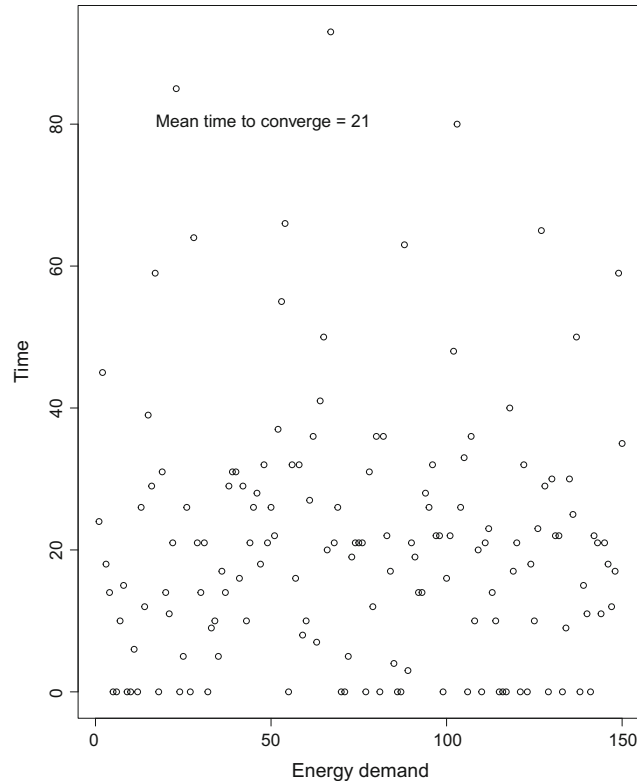


Fig. 9 Convergence time for distributed coalition formation process in MG network with 700 MGs

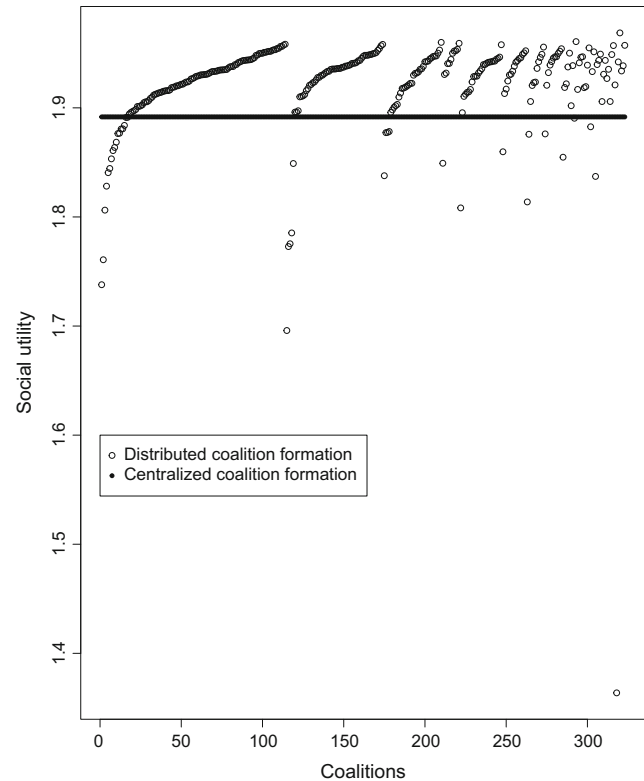


Fig. 11 Utility of coalitions formed by the distributed coalition formation process in MG network with 500 MGs

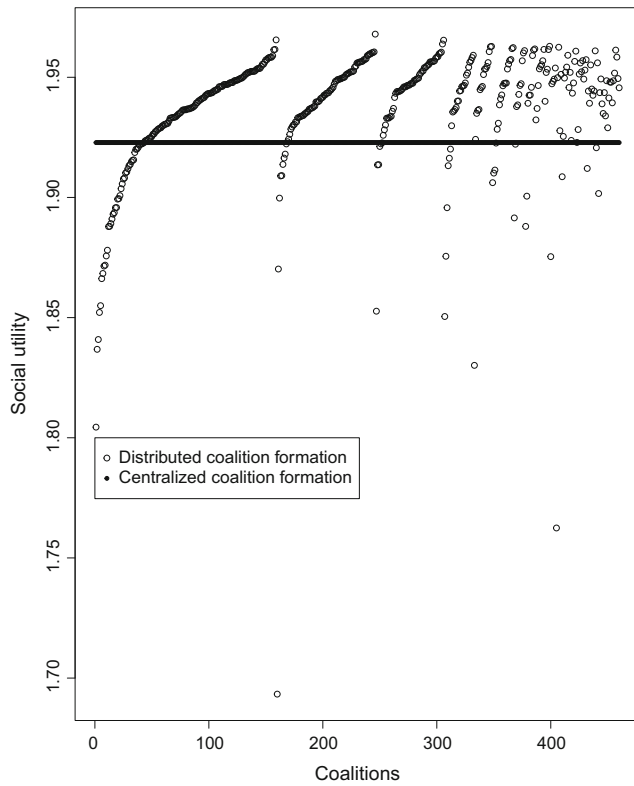


Fig. 12 Utility of coalitions formed by the distributed coalition formation process in MG network with 700 MGs

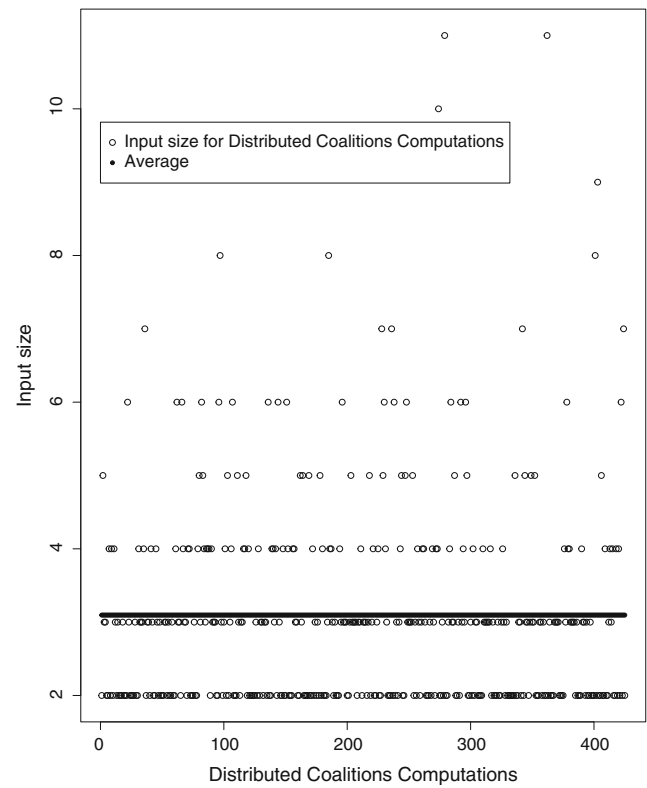


Fig. 14 Input size for distributed algorithm with MG size 500

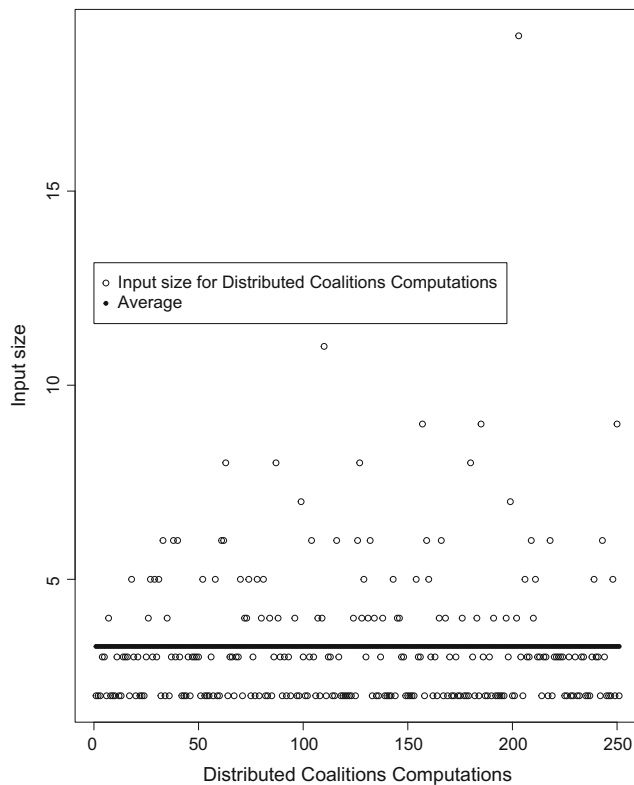


Fig. 13 Input size for distributed algorithm with MG size 300

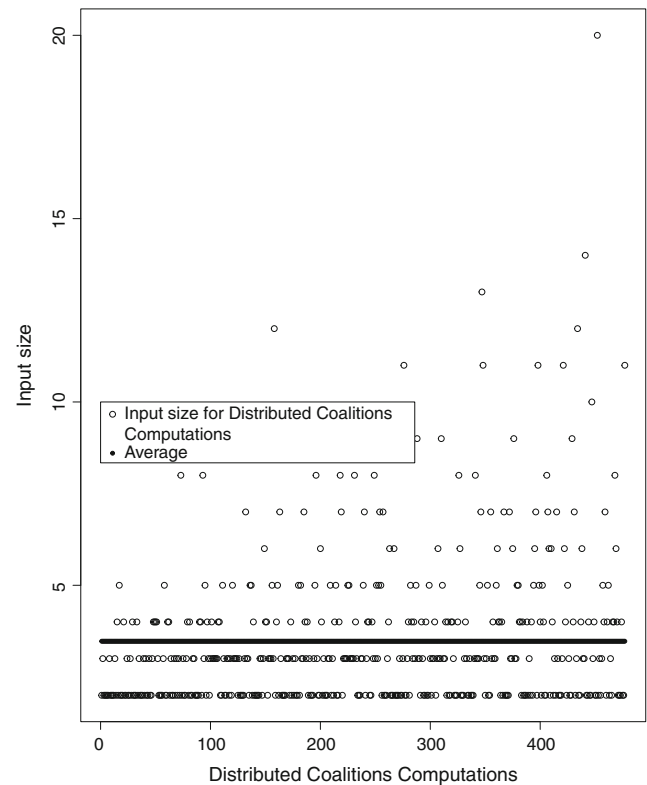


Fig. 15 Input size for distributed algorithm with MG size 700

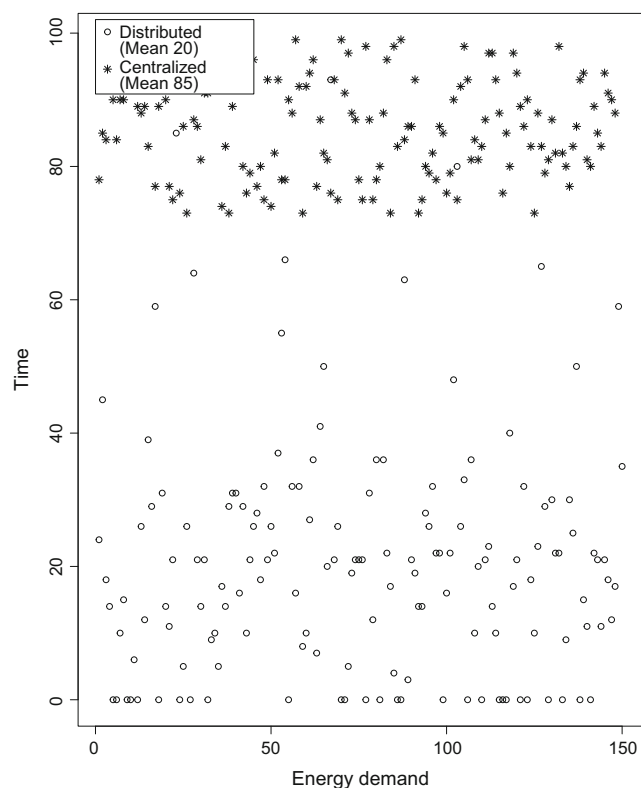


Fig. 16 Convergence time for distributed and centralized algorithm with asynchronous input and MG size 300

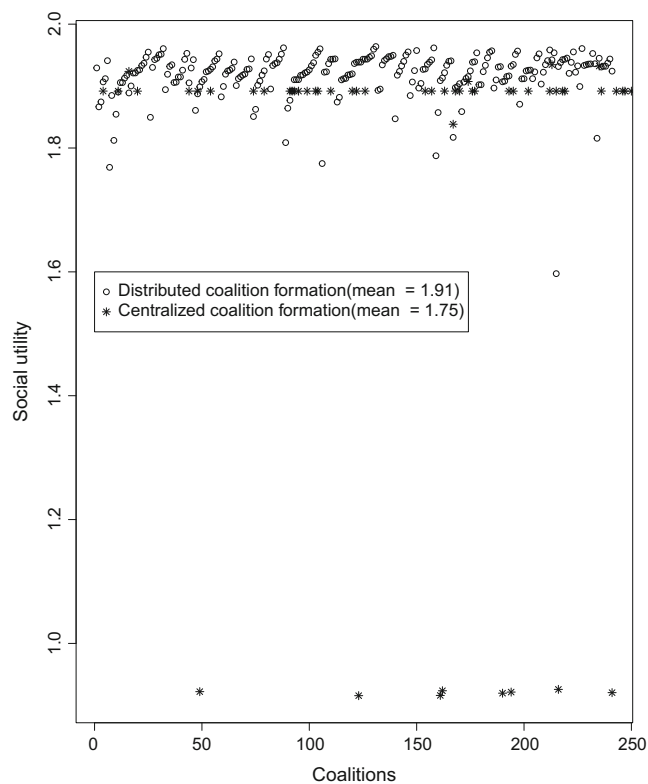


Fig. 18 Utility of coalitions formed by the distributed and centralized algorithm with asynchronous input and MG size 300

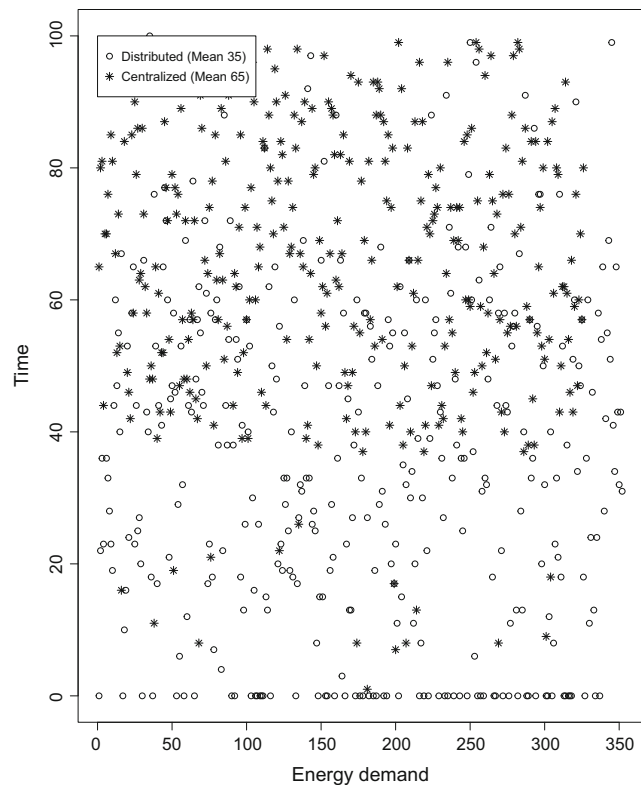


Fig. 17 Convergence time for distributed and centralized algorithm with asynchronous input and MG size 700

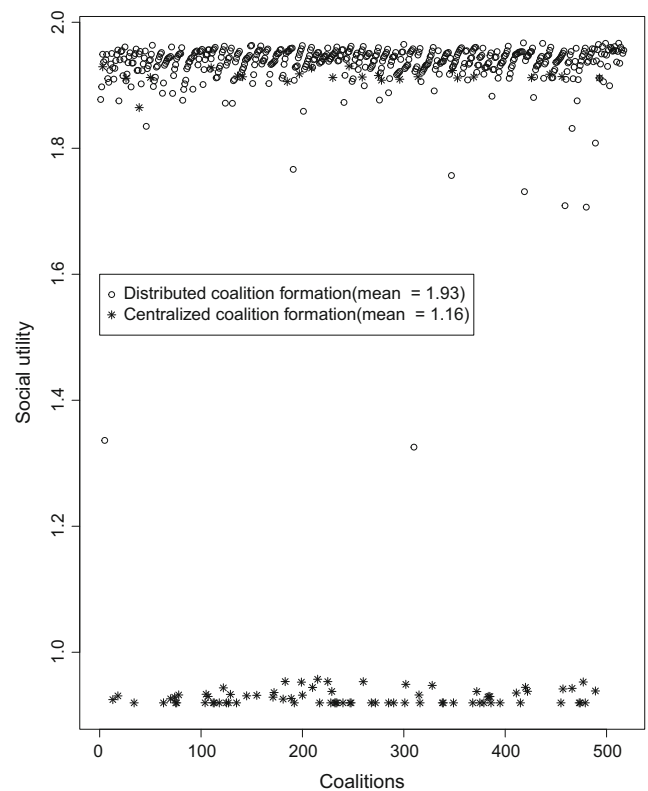


Fig. 19 Utility of coalitions formed by the distributed and centralized algorithm with asynchronous input and MG size 700

coalition formation process for the distributed algorithm. Figures 13, 14 and 15 show the input size for networks with 300, 500 and 700 MGs. We found that on average each coalition formation process has input size 4. Hence, despite increasing the network size from 300 to 700 computational load for individual coalition formation process remains constant. Hence the distributed coalition formation process is scalable than a centralized coalition formation algorithm whose input size would be 300, 500 and 700 respectively.

Now, we evaluate the performance of the distributed coalition formation algorithm with asynchronous or online supply or demand information. In these experiments at every iteration 10 MGs announce their energy requirement. The distributed coalition formation process continuously executes the coalition formation algorithm. Figures 16 and 17 show the convergence time for the distributed and the centralized algorithms in networks with 300 and 700 MGs. We observe average convergence time for distributed algorithm is 20 and the same for the centralized algorithm is 85 for a network with 300 MGs. And, average convergence time for distributed algorithm is 35 and the same for the centralized algorithm is 65 for a network with 700 MGs.

Next we evaluate the utility of coalitions formed by the distributed algorithm against the same for the centralized algorithm. We observe in Figs. 18 and 19 that the distributed algorithm performs much better.

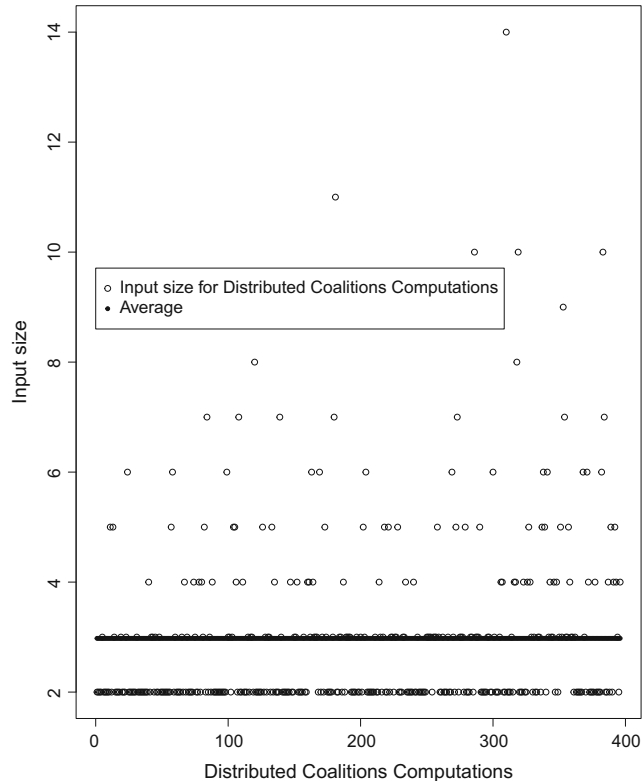


Fig. 20 Input size for distributed algorithm with MG size 300 and asynchronous input

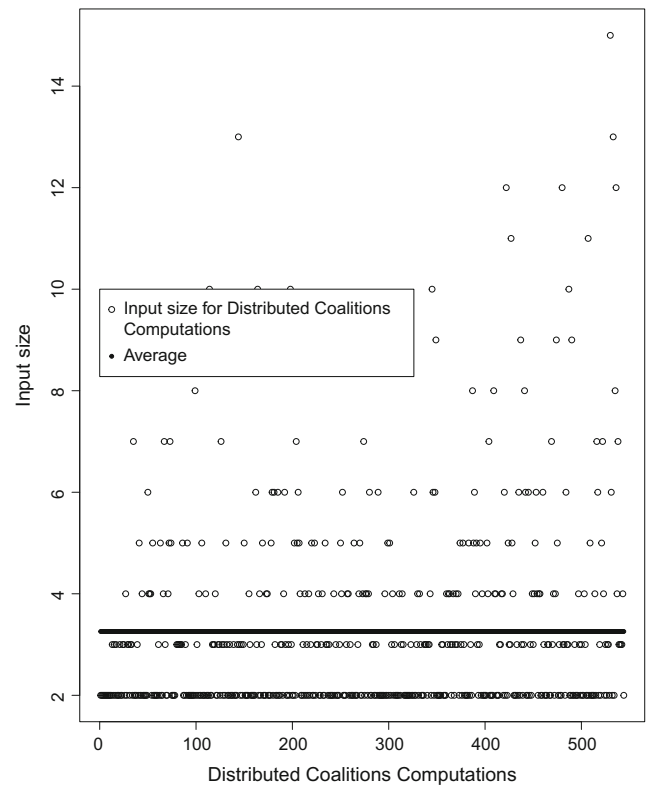


Fig. 21 Input size for distributed algorithm with MG size 700 and asynchronous input

Finally, as shown in Figs. 20 and 21 we evaluate scalability of the distributed algorithm with asynchronous inputs. We found that despite increasing MG network size from 300 to 700, average input size for each coalition formation computation remains 3. Hence the distributed algorithm remains scalable if energy demands and supply information are announced in an asynchronous fashion.

Conclusion

In this paper we have proposed a blockchain based solution for MG energy trade. We have developed a blockchain based distributed coalition formation algorithm where any MG can execute the coalition formation algorithm and blockchain mechanism ensures that a MG correctly executes such coalition formation algorithm. Through experimental evaluation we have shown that this distributed algorithm quickly converges, generates better coalition structure and more scalable than a centralized coalition formation algorithm. We have also shown that these results hold when the energy requirements are announced in an asynchronous fashion.

Acknowledgment This research is funded by Science Foundation Ireland grant number SFI/12/RC/2289.

References

1. Alam M, Gerding EH, Rogers A, Ramchurn SD (2015) A scalable interdependent multi-issue negotiation protocol for energy exchange. In: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, pp 1098–1104. AAAI Press
2. Alam M, Ramchurn SD, Rogers A (2013) Cooperative energy exchange for the efficient use of energy and resources in remote communities. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS '13, pp 731–738. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC
3. Chakraborty S, Nakamura S, Okabe T (2014) Scalable and optimal coalition formation of microgrids in a distribution system. In: IEEE PES Innovative Smart Grid Technologies, Europe, pp 1–6. <https://doi.org/10.1109/ISGTEurope.2014.7028761>
4. Chakraborty S, Nakamura S, Okabe T (2015) Real-time energy exchange strategy of optimally cooperative microgrids for scale-flexible distribution system. *Expert Syst Appl* 42(10):4643–4652
5. Cintuglu MH, Martin H, Mohammed OA (2015) Real-time implementation of multiagent-based game theory reverse auction model for microgrid market operation. *IEEE Trans Smart Grid* 6(2):1064–1072. <https://doi.org/10.1109/TSG.2014.2387215>
6. Dimeas AL, Hatziargyriou ND (2005) Operation of a multiagent system for microgrid control. *IEEE Trans Power Syst* 20(3):1447–1455. <https://doi.org/10.1109/TPWRS.2005.852060>
7. Dou C, Yue D, Li X, Xue Y (2016) Mas-based management and control strategies for integrated hybrid energy system. *IEEE Trans Ind Inf* 12(4):1332–1349. <https://doi.org/10.1109/TII.2016.2569506>
8. Ishowo-Oloko F, Vytelingum P, Jennings N, Rahwan I (2012) A storage pricing mechanism for learning agents in masdar city smart grid. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS '12, pp 1167–1168. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC. <http://dl.acm.org/citation.cfm?id=2343896.2343902>
9. King S, Nadal S (2012) ppcoin: peer-to-peer crypto-currency with proof-of-stake. <http://www.peercoin.net/assets/paper/peercoin-paper.pdf>
10. Logenthiran T, Naayagi RT, Woo WL, Phan VT, Abidi K (2015) Intelligent control system for microgrids using multiagent system. *IEEE J Emerging and Selected Topics in Power Elect* 3(4):1036–1045. <https://doi.org/10.1109/JESTPE.2015.2443187>
11. Logenthiran T, Srinivasan D, Wong D (2008) Multi-agent coordination for der in microgrid. In: 2008 IEEE International Conference on Sustainable Energy Technologies, pp 77–82. <https://doi.org/10.1109/ICSET.2008.4746976>
12. Luo Y, Itaya S, Nakamura S, Davis P (2014) Autonomous cooperative energy trading between prosumers for microgrid systems. In: 39th Annual IEEE Conference on Local Computer Networks Workshops, pp 693–696. <https://doi.org/10.1109/LCNW.2014.6927722>
13. Mao M, Jin P, Hatziargyriou ND, Chang L (2014) Multiagent-based hybrid energy management system for microgrids. *IEEE Trans Sustainable Energy* 5(3):938–946. <https://doi.org/10.1109/TSTE.2014.2313882>
14. Mihailescu R, Vasirani M, Ossowski S (2011) Dynamic coalition adaptation for efficient agent-based virtual power plants. In: Multiagent System Technologies - 9th German Conference, MATES 2011, Berlin, Germany, October 6–7, 2011. Proceedings, pp 101–112
15. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>
16. Peters M, Ketter W, Saar-Tsechansky M, Collins J (2013) A reinforcement learning approach to autonomous decision-making in smart electricity markets. *Mach Learn* 92(1):5–39
17. Rahimi-Kian A, Sadeghi B, Thomas RJ (2005) Q-learning based supplier-agents for electricity markets. In: IEEE Power Engineering Society General Meeting, 2005, pp 420–427, vol 1. <https://doi.org/10.1109/PES.2005.1489514>
18. Reddy PP, Veloso MM (2011) Strategy learning for autonomous agents in smart grid markets. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two, IJCAI'11, pp 1446–1451. AAAI Press
19. Yasir M, Purvis M, Purvis M, Savarimuthu BTR (2014) An intelligent learning mechanism for trading strategies for local energy distribution. Springer International Publishing, Cham, pp 159–170
20. Yasir M, Purvis M, Purvis M, Savarimuthu BT (2014) Intelligent battery strategies for local energy distribution. In: Revised Selected Papers of the COIN 2013 International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems IX - Volume 8386. Springer New York, Inc., New York, pp 63–80
21. Yasir M, Purvis M, Purvis M, Savarimuthu BTR (2015) Agent-based community coordination of local energy distribution. *AI & SOCIETY* 30(3):379–391. <https://doi.org/10.1007/s00146-013-0528-1>