Applied Soft Computing 13 (2013) 3751–3761

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Multi-agent modeling for solving profit based unit commitment problem

Deepak Sharma [a,*], Anupam Trivedi [b], Dipti Srinivasan [b], Logenthiran Thillainathan [b]

[a] Department of Mechanical Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, Assam, India
[b] Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore

ARTICLE INFO

*Article history:*
Received 27 August 2012
Accepted 2 April 2013
Available online 17 April 2013

*Keywords:*
Profit based unit commitment
Agent rules
Multi-agent modeling
Deregulation

ABSTRACT

Profit based unit commitment problem (PBUC) from power system domain is a high-dimensional, mixed variables and complex problem due to its combinatorial nature. Many optimization techniques for solving PBUC exist in the literature. However, they are either parameter sensitive or computationally expensive. The quality of PBUC solution is important for a power generating company (GENCO) because this solution would be the basis for a good bidding strategy in the competitive deregulated power market. In this paper, the thermal generators of a GENCO is modeled as a system of intelligent agents in order to generate the best profit solution. A modeling for multi-agents is done by decomposing PBUC problem so that the profit maximization can be distributed among the agents. Six communication and negotiation stages are developed for agents that can explore the possibilities of profit maximization while respecting PBUC problem constraints. The proposed multi-agent modeling is tested for different systems having 10–100 thermal generators considering a day ahead scheduling. The results demonstrate the superiority of proposed multi-agent modeling for PBUC over the benchmark optimization techniques for generating the best profit solutions in substantially smaller computation time.



## 1. Introduction

In the traditional vertically integrated power market, the power generator companies (GENCOs) are obliged to satisfy load demand by optimally scheduling power generating units at least operation cost. In this process of scheduling, ON–OFF (1–0) decision for every generating unit has to be taken by GENCOs for every hour on the planning horizon. Usually, the less expensive units are committed while satisfying the problem constraints. Once unit commitment (1–0) is decided for every generating unit, an economic dispatch from committed units is calculated in order to minimize the total operation cost. It is referred as unit commitment (UC) problem in the literature.

Now, the traditional power market is undergoing a radical transformation due to deregulation. Contrary to the regulated market, an independent system operator (ISO) would have no control on bids submitted by GENCOs. Therefore, GENCOs are not obliged to satisfy load demand constraint. GENCOs can decide generators scheduling plan (0–1) and the amount of power sold in the deregulated market that can maximize their individual expected profits. A scheduling plan involves ON–OFF (1–0) decision for every generating unit. An economic dispatch of power is calculated for maximizing its profit. This problem is referred as profit based unit commitment (PBUC) in the literature. Every GENCO solves PBUC problem independently and compete each other to sell their power in the power market by submitting their bids. Therefore, it is necessary to have an efficient algorithm available for GENCO that can generate best profit solutions.

Various techniques have been reported in the literature for solving UC and PBUC problems. Priority list [1] which is based on heuristics, is simple and computationally efficient but leads to sub-optimal solutions. Dynamic programming [2] can find near optimal solutions. But, it is mathematically complex, and it requires large computation time and memory size. Lagrange relaxation (LR) method [3] is viewed as an efficient and applicable method to even solve for a large system. However, it is sensitive to a choice of parameters such as Lagrange multipliers. Tuning of Lagrange multipliers itself is an optimization problem which can take different values for different power system.

Soft computing techniques have also been used for same purpose. Techniques like genetic algorithm (GA) [4], particle swarm optimization (PSO) [5,6], evolutionary programming (EP) [7], neural network [8], simulated annealing [9], ant colony optimization [10,11] to name a few, have been used to avoid premature

* Corresponding author. Tel.: +91 3612582661.
*E-mail addresses:* dsharma@iitg.ernet.in (D. Sharma), eleat@nus.edu.sg (A. Trivedi), elesd@nus.edu.sg (D. Srinivasan), g0700236@nus.edu.sg (L. Thillainathan).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.04.001

**Nomenclature**

| | |
|---|---|
| $PF$ | profit of GENCO |
| $PF_{it}$ | profit of GENCO in iteration '*it*' |
| $RV$ | revenue of GENCO |
| $TC$ | total operation cost of GENCO |
| $Pr_{i,t}$ | profit generated by generator $i$ at hour $t$ |
| $P_{i,t}$ | power generation of generator $i$ at hour $t$ |
| $R_{i,t}$ | reserve generation of generator $i$ at hour $t$ |
| $X_{i,t}$ | ON/OFF status of generator $i$ at hour $t$ |
| $LD'_t$ | forecast demand at hour $t$ |
| $SR'_t$ | forecast reserve at hour $t$ |
| $DL_t$ | demand left at hour $t$ |
| $RL_t$ | reserve left at hour $t$ |
| $P_i^{min}$ | minimum generation limit of generator $i$ |
| $P_i^{max}$ | maximum generation limit of generator $i$ |
| $N$ | number of generator units |
| $T$ | number of hours |
| $SP_t$ | forecast spot prices at hour $t$ |
| $RP_t$ | forecast reserve prices at hour $t$ |
| $ST_i$ | start-up cost of generator $i$ |
| $r$ | probability that reserve is called and generated |
| $T_i^{up}/T_i^{down}$ | minimum up/down time of generator $i$ |
| $t_i^{up}/t_i^{down}$ | duration during which unit $i$ is continuously ON/OFF |
| $a_i$, $b_i$, $c_i$ | cost coefficients of generator $i$ |

convergence. However, these techniques are computationally expensive as most of the techniques are populations based algorithms. The hybrid techniques are designed such LR-EP [12], LRGA [13], and EP-Tabu search [14], to preserve a global searching capacity of optimization techniques but in a reasonable computation time. An aim is to find an optimum set of parameters for parameter-sensitive techniques using soft computing.

Above mentioned techniques generated the best profit solutions till the multi-agent approaches have not been used for solving PBUC. Two multi-agent approaches have been reported in the literature. In first multi-agent approach [15], different random traveling routes are designed for visiting generator agents. A mobile agent negotiates profit and power with generator agents as per the traveling routes. But, it can possible that none of the traveling route can generate the best profit solution in a stipulated time. Moreover, this approach involves stochasticity that can generate different profit solutions when executes at different time for same load demand condition. Other multi-agent approach is our previous work [16] where maximum profit generating generator agents are committed for a given load demand. Thereafter, a negotiation stage is used to commit those generators which can produce positive profit for remaining load demand. These multi-agent studies reported the best profit solution when compared with the benchmark techniques. An efficient and improved version of our previous study [16] is proposed in this paper that produces the best profit solutions. In the following section, an overview of multi-agent system is given.

## 2. Overview of multi-agent system

A term 'agent' is defined in [17] as *"a software or hardware entity that is situated in some environment, and is capable of performing autonomous actions in that environment in order to meet its design objectives"*. An agent is characterized by its autonomy, social ability, reactive and protective behavior. Being autonomous an agent can independently perform any complex task. Social ability allows an agent to interact and negotiate with human or other agents to achieve its task. Reactive characteristic of agent helps it to perceive and respond toward a changing environment in a timely fashion. Proactive behavior of agent is not simply act in response to its environment but it is able to meet its goal by changing its behavior dynamically. Some other properties that are associated with agents include mobility, temporal continuity, collaborative behavior, etc. Agents which satisfy all or a few above mentioned properties, can be further categorized as weak or strong agency. Multi-agent system (MAS) is an extension of agent technology where agents are loosely connected and act in an environment to achieve their goals. Some benefits of using MAS technology for large system are [18]:

- Parallel computation and asynchronous operations can increase speed and efficiency of the operation.
- Fault-tolerance for graceful degradation of system when agents fail.
- Scalability and flexibility of adding or removing agents from a system whenever necessary.
- Reusability of agents that can be reused multiple times.

However, some critical challenges with MAS are:

- Concurrent learning and action of agents toward a changing environment can result in unstable behavior and can cause possible chaos.
- Limited visibility or information for agents distributed in the system can lead to sub-optimal solution.
- Action or decision taken by any agent may not be suitable for another agent that can be reduced by sharing the information on constraints, action preference and goal priorities. But the problem is when to communicate and to which of the agents.
- Difficulty in debugging and testing a massively parallel and distributed system.

Nevertheless, MAS has been used as a potential tool for various real world problems over the past few year. Broadly, MAS has been used as an approach to the building of robust, flexible and extensible hardware/software systems, or as a modeling approach [19]. In former approach, a system of multi-agents is required for those applications where agents can respond correctly in a changing environment, agents can be added as and when required, agents can be replaced in other system or be upgraded, and for graceful degradation of system when one or more agents fail. A few such applications can be found in [20–23].

In modeling approach, MAS is used to represent a large and complex system which is difficult to model explicitly. The entities are designed as intelligent agents to simulate complex behavior. Many real world problems using this modeling approach can be found in studies like [24–27]. In power engineering domain, studies [19,28] summarized the concepts, approaches, and technical challenges, technologies, standards, and tools for building multi-agent systems for various applications. PBUC is one such problem from power system area which is high-dimensional, mixed variables and complex due to its combinatorial nature. Multi-agent modeling is done to solve PBUC problem by decomposing the problem into asynchronous operations. In the remaining paper, the formulation for PBUC problem is described in Section 3. Our proposed multi-agent modeling for PBUC is discussed in Section 4. The simulation results are discussed in Section 5 which is followed by conclusions in Section 6 with a note on future work.

## 3. PBUC problem formulation

A formulation of PBUC is adopted from study [12] where profit maximization of a GENCO in the deregulated market is modeled with the soft constraints on load demand and reserve. A description of objective function and constraints is given in the following sections.

### 3.1. Objective function

In the deregulated power market, objective for a GENCO is to maximize its expected profit as per Eq. (1).

$$\text{Maximize}: Profit(PF = RV - TC) \tag{1}$$

A GENCO can sell power in the energy market and the reserve (ancillary) markets, respectively. A purpose of maintaining reserve in the ancillary market is to cater uncertainty in load demand. The amount of power and reserve sold depends on the way reserve payments are made [29]. A payment is made for reserve allocated where GENCO receives the reserve price per generator of reserve for every time period that the reserve is allocated but not used. When reserve is used, GENCO receives the spot price for the generated reserve [12]. Thus, the revenue for GENCO is evaluated according to Eq. (2).

$$RV = \sum_{i=1}^{N}\sum_{t=1}^{T}(P_{i,t}.SP_t).X_{i,t} + \sum_{i=1}^{N}\sum_{t=1}^{T}((1-r)RP_t + r.SP_t)R_{i,t}.X_{i,t} \tag{2}$$

The term on the left hand side of addition in above equation represents the expected revenue from power selling. Second term on the right hand side of addition signifies the expected revenue from reserve. If reserve is used and then, GENCO can receive the spot price with probability '$r$'. Otherwise, the reserve price is received when reserve is allotted but not used.

Total operation cost for a GENCO is evaluated as per Eq. (3) that includes fuel cost and start-up cost for committed generators.

$$\begin{aligned} TC = {} & (1-r)\sum_{i=1}^{N}\sum_{t=1}^{T}F(P_{i,t}).X_{i,t} + \\ & r.\sum_{i=1}^{N}\sum_{t=1}^{T}F(P_{i,t}+R_{i,t}).X_{i,t} + ST_i.X_{i,t}.(1-X_{i,t-1}) \end{aligned} \tag{3}$$

Fuel cost ($F(P_{i,t})$) of generator $i$ at time $t$ is represented by a quadratic polynomial which is given in Eq. (4)

$$F(P_{i,t}) = a_i + b_iP_{i,t} + c_iP_{i,t}^2 \tag{4}$$

where $a_i$, $b_i$ and $c_i$ are cost coefficients of generator $i$.

### 3.2. Constraints

PBUC problem is subjected to various constraints which are as follows:

1. *Load demand constraint*: In the deregulated power market, this constraint is represented as soft constraint [29]. It suggests that the total power generated from the committed generators at time $t$ can be less than or equal to the forecast load demand at $t$.

$$\sum_{i=1}^{N}P_{i,t}X_{i,t} \le LD'_t \tag{5}$$

2. *Spinning reserve constraint*: It also suggests that the reserve generated from the committed generators at time $t$ can be less than or equal to the reserve requirement at $t$.

$$\sum_{i=1}^{N}R_{i,t}X_{i,t} \le SR'_t \tag{6}$$

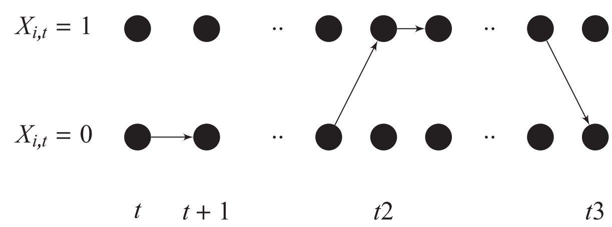


**Fig. 1.** Up and down time constraints for a generator$_i$.

3. *Generation limit constraints*: The power and reserve produced by any generator must be within its limited capacity as indicated below:

$$\begin{aligned} & P_i^{min} \le P_{i,t} \le P_i^{max} \\ & P_{i,t} + R_{i,t} \le P_i^{max} \end{aligned} \tag{7}$$

4. *Minimum up and down time constraints*: These constraints indicate that a generator must be ON/OFF for minimum number of hours before shut-down or start-up, respectively. These constraints are given by Eq. (8).

$$\begin{aligned} & (t_{up}(i) - T_i^{up})(X_{i,t-1} - X_{i,t}) \ge 0 \\ & (t_{down}(i) - T_i^{down})(X_{i,t-1} - X_{i,t}) \ge 0 \end{aligned} \tag{8}$$

Fig. 1 describes these constraints for a generator. Assuming generator$_i$ is OFF ($X_{i,t} = 0$) at time $t$. Before committing at time $t2$, generator$_i$ has to be shut down for minimum $T_i^{down}$ hours to satisfy the down time constraint, that is, $t2 = t + T_i^{down}$. Similarly, the same generator has to be committed for minimum $T_i^{up}$ hours before it shuts down at time $t3 = t + T_i^{down} + T_i^{up}$.

From the above formulation, we can observe that PBUC problem is high dimensional ($N$), has real and binary variables ($P_{i,t}$, $R_{i,t}$, $X_{i,t}$), a non-explicit objective function ($PF$) and various inequality constraints. Due to minimum up and down time constraints, PBUC problem becomes combinatorial in nature which is complex and difficult to solve. In the following section, a multi-agent modeling for PBUC problem is discussed which can maximize an expected profit of GENCO by communication and negotiation among the agents.

## 4. Multi-agent modeling for PBUC

JADE (Java Agent DEvelopment) [30] framework is used that conforms to the Foundation for Intelligent Physical Agents [31] standards for intelligent agents. JADE platform provides a set of functions and classes to implement agent functionality, such as agent management service, directory facilitator and message passing services. Agent management service (AMS) is responsible for managing agent platform, which maintains a directory of Agent Identifiers (AIDs) and agent states. Directory facilitator (DF) provides default yellow page services in the platform which allow agents to discover other agents in a network based on services they wish to offer or to obtain. Finally, message transport service (MTS) which is responsible for delivering messages between agents, provides services for message transportation in agent system.

### 4.1. Architecture of agents

Two types of agents are created using JADE platform that are coordinating agent (CA) and generator agents (GenAgents). Fig. 2 shows a simplified architecture of multi-agent system for PBUC where G1–G8 represent GenAgents. It can be seen from figure that GenAgents can only communicate and share information/data with

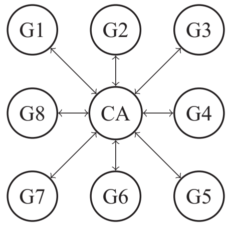


**Fig. 2.** Multi-agent system for thermal generation scheduling.

CA. However, any negotiation between GenAgents can possible via CA.

PBUC problem is decomposed in this paper so that a task can be distributed among CA and GenAgents. For example, Eqs. (2) and (3) show two summations over '*T*' and '*N*'. Summation over time is decomposed for CA so that CA can interact and negotiate with GenAgents for every time period '*t*'. Other functionalities for building CA are:

- CA can take decision for committing GenAgents which depends on profit values evaluated by GenAgents.
- Apart from communicating with GenAgents, CA also stores data which can be shared with GenAgents.
- CA is responsible to satisfy constraints given in (5) and (6) and to update $(DL_t, RL_t)$ every time when any GenAgent commits or shuts down.
- CA also asks GenAgents to check their up and down time constraints.

Therefore, CA is the most important agent of present multi-agent system.

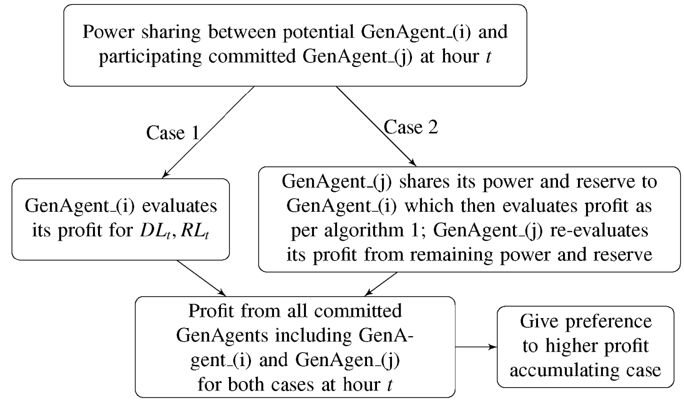


**Fig. 4.** Power sharing between GenAgent_(i) and GenAgent_(j) at hour *t*.

Summation over '*N*' is broken down by generating '*N*' GenAgents. Eq. (1) is now simplified and these GenAgents can evaluate profit according to (9).

$$\begin{aligned} Profit(Pr_{i,t}) &= [\{P_{i,t}.SP_t + R_{i,t}.((1-r).RP_t + r.SP_t)\} \\ &- \{(1-r).F(P_{i,t}) + r.F(P_{i,t}+R_{i,t})\}].X_{i,t} \end{aligned} \tag{9}$$

Constraints given in (5) and (6) are modified for these GenAgents as given in (10) in which values of $DL_t$ and $RL_t$ are to be supplied by CA. However, the generation limit constraints are same as given in (7).

$$\begin{aligned} &P_{i,t}.X_{i,t} \le DL_t \\ &R_{i,t}.X_{i,t} \le RL_t \end{aligned} \tag{10}$$

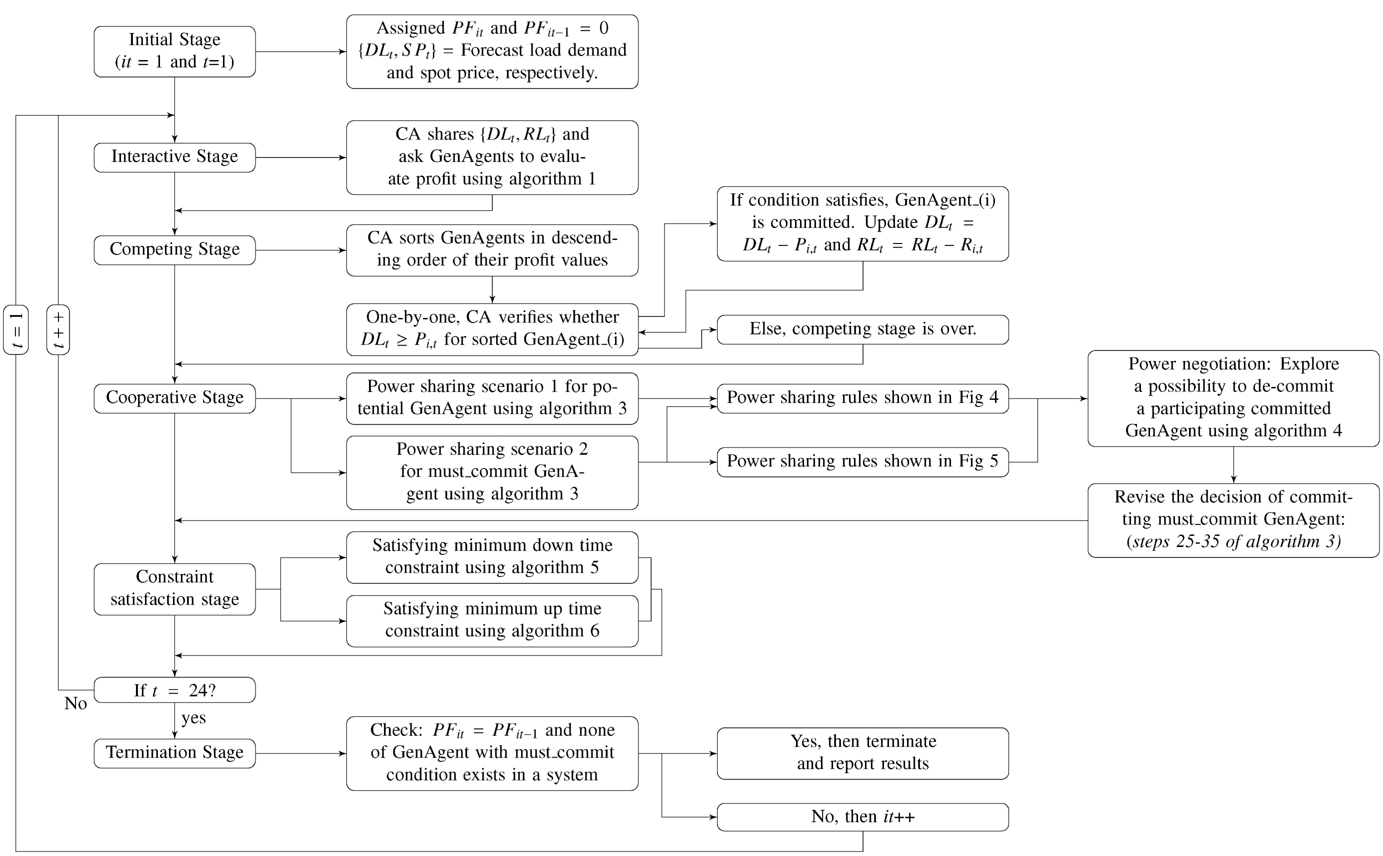


**Fig. 3.** Block diagram of agents communication and negotiation.

It can be seen here that a desired amount of power and reserve is decided by GenAgents, whereas CA tries to commit those GenAgents for which the profit for GENCO can be maximized. GenAgents also satisfy up and down time constraints given in (8) when asked by CA.

### 4.2. Communication and negotiation

For intelligent scheduling of generators, CA communicates GenAgents at various stages as shown in Fig. 3. These stages are broadly divided into six stages which are discussed in the following sections.

#### 4.2.1. Initial stage

This stage is designed to initialize important parameters. CA assigns zero value to current and last iteration's profit of GENCO, that is, $\{PF_{it}, PF_{it-1} = 0\}$. Similarly, $\{DL_t, SP_t\}$ = forecast load demand and spot price, respectively. $RL_t$ is set to 10% of $DL_t$ and $RP_t = 0.001 \times SP_t$ for benchmarking the results. Starting iteration ($it$) and time ($t$) are equal to one in this stage.

#### 4.2.2. Interactive stage

This stage is designed for GenAgents to evaluate their profit values using Algorithm 1. Values of $\{DL_t, RL_t, SP_t, RP_t\}$ which are stored by CA, are shared with GenAgents. In this process, every GenAgent evaluates $\{Pr_{i,t}, P_{i,t}, R_{i,t}\}$ values and send them to CA. GenAgents which are already committed or cannot satisfy constraint given in (10), will not evaluate profit.

Algorithm 1 which evaluates $\{Pr_{i,t}, P_{i,t}, R_{i,t}\}$ is motivated from the reserve payment method described in Section 3.1 where $RP_t \ll SP_t$. In this algorithm, every GenAgent tries to deliver as much power in hour $t$ depending on $DL_t$. If generating capacity further allows, then reserve gets scheduled depending on $RL_t$ and unit capacity as given in Eq. (7). Profit, power and reserve values for every GenAgents are then stored by CA that helps in committing them in following stage.

**Algorithm 1.** Profit evaluation by GenAgent_(i)

```
if DL_t >= P_i^max then
  P_i,t = P_i^max and R_i,t = 0 for selling
else
  if DL_t >= P_i^min then P_i,t = DL_t
    if P_i,t + RL_t > P_i^max then R_i,t = P_i^max − P_i,t
    else
      R_i,t = RL_t
    end if
  end if
end if
Evaluate PR_i,t from equation (9)
```

**Algorithm 2.** Rule for committing potential GenAgent_(i)

```
if Profit of potential GenAgent_(i) >0 for remaining {DL_t, RL_t} then
  Current_profit = Last_profit = 0
  for every committed GenAgent at hour t do
    Apply power sharing rules shown in Fig. 4
    Current_profit = Pr_i,t + Pr_j,t + Σ_{k=1, k≠i, k≠j}^{M} Pr_k,t
    if Current_profit > Last_profit then
      Last_profit = Current_profit
      Store updated Pr_i,t, P_i,t, R_i,t, Pr_j,t, P_j,t, R_j,t
    end if
  end for
  Commit GenAgent_(i) with updated values of Pr_i,t, P_i,t, R_i,t.
Also, assign updated Pr_j,t, P_j,t, R_j,t values for previously
committed GenAgent_(j) of the best Last_profit
combination
  Update DL_t and RL_t
end if
```

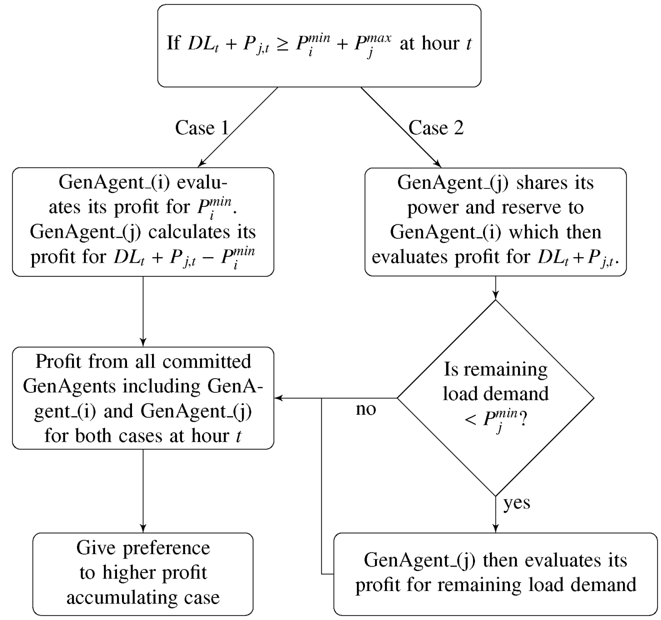


**Fig. 5.** Power sharing between a must_commit GenAgent_(i) and a participating committed GenAgent_(j).

#### 4.2.3. Competing stage

This stage is referred as a competing stage for GenAgents where every GenAgent tries to commit first. This stage can help GENCO to accumulate a *significant* profit by committing maximum profit generating GenAgents. Therefore, CA sorts GenAgents in descending order of their profit values and commit them one-by-one. When any GenAgent_(i) is committed, $\{DL_t, RL_t\}$ values are updated as $\{DL_t = DL_t - P_{i,t},\ RL_t = RL_t - R_{i,t}\}$. The process of committing GenAgents continues till $DL_t \geq P_{i,t}$. Otherwise, this stage gets over.

#### 4.2.4. Cooperative stage

After executing rules of competing stage, some GenAgents may available in a system of GenAgents which can still generate positive profit for remaining $\{DL_t, RL_t\}$. These GenAgents are referred as potential GenAgents. In some other cases, a few GenAgents have to satisfy minimum up/down time constraint by committing for a specified period. These GenAgents are designated as must_commit GenAgents for identification and further processing in next iteration. The details of assigning must_commit condition is described later in Section 4.2.5.

In cooperative stage, a potential or a must_commit GenAgent cooperatively negotiates power and reserve with already committed GenAgents. The aim of this stage is to further accumulate profit for GENCO, after-all every generating unit belongs to same GENCO. Based on a potential or a must_commit GenAgent, two power sharing scenarios are developed here. Power sharing scenario 1 is developed to commit potential GenAgent after negotiating power and reserve with committed GenAgents. The rules of negotiation for must_commit GenAgent and committed GenAgent are designed in power sharing scenario 2. These scenarios are discussed in following sections.

*4.2.4.1. Power sharing scenario 1.* This power sharing scenario is designed to explore a best profit combination for potential GenAgent_(i) when all committed GenAgents participate in power sharing. The rules are designed in Algorithm 2 where two power sharing cases for a potential GenAgent_(i) and a participating committed GenAgent_(j) are shown in Fig. 4. Based on higher profit accumulating case, the values of current_profit and last_profit values are updated as and when required. When every committed

GenAgent takes part in this power sharing process, GenAgent_(i) is committed with the updated values of $Pr_{i,t}$, $P_{i,t}$, $R_{i,t}$. The participating committed GenAgent_(j) of the best profit combination also updates its $Pr_{j,t}$, $P_{j,t}$, $R_{j,t}$ values.

**Algorithm 3.** Rules for must_commit GenAgent_(i)

```
1:  Current_profit = Last_profit = 0
2:  for every committed GenAgent at hour t do
3:    if DL_t ≥ P_i^min then
4:      Apply power sharing rules shown in Fig. 4
5:    else
6:      Apply power sharing rules shown in Fig. 5
7:    end if
8:    Prefer high profit generating case
9:    if Pr_{i,t} + Pr_{j,t} ≥ 0 then
10:     Store updated Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t}
11:   else
12:     if Pr_{i,t} ≥ Pr_{i,t} + Pr_{j,t} for case 1 || Pr_{i,t} < 0 for case 2 from
        both Figs. 4 and 5 then
13:       Store updated Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t}
14:     else
15:       Explore a possibility of de-committing participating
          committed GenAgent_(j) using Algorithm 4
16:     end if
17:   end if
18:   if Current_Profit > Last_profit then
19:     Last_profit = Current_profit
20:     Store updated Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t}
21:   end if
22: end for
23: Store updated ON–OFF status of best Last_profit
    combination and Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t} values of
    GenAgent_(i) and GenAgent_(j).
24: Revise the decision of committing must_commit
    GenAgent_(i)
25: if Minimum up time constraint of must_commit
    GenAgent_(i) is satisfied from t to t − T_i^up then
26:   Profit_sum of GenAgent_(i) = Σ_{k=t}^{t+T_i^down} Pr_{i,k}
27:   if Profit_sum of GenAgent_(i) < Pr_{j,t} of GenAgent_(j) then
28:     Remove must_commit condition and shut down
        GenAgent_(i) from t to t + T_i^down.
29:     Participating committed GenAgent_(j) remains
        committed with its original Pr_{j,t}, P_{j,t} and R_{j,t} values
30:   else
31:     Commit GenAgent_(i) and GenAgent_(j) with updated
        values of Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t} of the best profit
        combination
32:   end if
33: else
34:   Commit GenAgent_(i) and GenAgent_(j) with updated
      values of Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t} of the best profit
      combination
35: end if
36: Update DL_t and RL_t
```

*4.2.4.2. Power sharing scenario 2.* When any GenAgent with must_commit condition exists in a system of GenAgents, this power sharing scenario is executed. The rules are designed in Algorithm 3 with the same motivation of finding a best profit combination for a must_commit GenAgent when all committed GenAgents participate in this power sharing process. However, there exist two conditions for power sharing cases which depend on available $DL_t$ and capacity ($P_i^{min}$) of must_commit GenAgent_(i). First power sharing condition corresponds to the same rules as shown in Fig. 4. Second power sharing condition triggers the rules which are shown in Fig. 5. These rules are similar to the rules in Fig. 4, but must_commit GenAgent_(i) is either allowed to commit for $P_i^{min}$ or for more power.

*4.2.4.3. Power negotiation.* Once a higher profit generating case is chosen, a possibility of complete power sharing from a participating committed GenAgent_(j) to a must_commit GenAgent_(i) is explored. It is referred as de-committing a participating committed GenAgent_(j) and conditions are shown in Algorithm 4. This possibility is explored so that the profit for GENCO can be maximized further by allowing must_commit GenAgent to commit for more power. However before shutting committed GenAgent_(j), its minimum up and down time constraints as given in (8) are to be satisfied for specified time period as mention in Algorithm 4. Whether participating GenAgent_(j) is de-committed or not, updated $Pr_{i,t}$, $P_{i,t}$, $R_{i,t}$, $Pr_{j,t}$, $P_{j,t}$, $R_{j,t}$ values are stored. In a same manner, every committed GenAgent participates in power sharing and a best profit combination for must_commit GenAgent_(i) is explored. ON–OFF status and updated $Pr_{i,t}$, $P_{i,t}$, $R_{i,t}$, $Pr_{j,t}$, $P_{j,t}$, $R_{j,t}$ values of GenAgent_(i) and GenAgent_(j) of the best profit combination are stored.

**Algorithm 4.** Conditions for de-committing a participating committed GenAgent_(j) for GenAgent_(i)

```
if Minimum up time constraint of GenAgent_(j) is satisfied
from t to t − T_j^up then
  if Minimum down time constraint of GenAgent_(j) is
  satisfied from t to t + T_j^down, assuming its current status
  'OFF' then
    GenAgent_(j) is assumed de-committed
    Update DL_t = DL_t + P_{j,t} and RL_t = RL_t + R_{j,t}
    GenAgent_(i) then evaluates and store its Pr_{i,t}, P_{i,t}, R_{i,t}
    for updated DL_t, RL_t
  else
    Profit_sum of GenAgent_(j) = Σ_{k=t}^{t+T_j^down} Pr_{j,k}
    if Profit_sum of GenAgent_(j) < best stored Pr_{i,t} of
    GenAgent_(i) then
      Update DL_t = DL_t + P_{j,t} and RL_t = RL_t + R_{j,t}
      GenAgent_(i) then evaluates and store its Pr_{i,t}, P_{i,t}, R_{i,t}
      for updated DL_t, RL_t
    else
      GenAgent_(j) cannot de-commit
      Store updated Pr_{i,t}, P_{i,t}, R_{i,t}, Pr_{j,t}, P_{j,t}, R_{j,t}
    end if
  end if
end if
```

*4.2.4.4. Condition for revising the decision of committing must_commit GenAgent_(i): steps 25–35 of Algorithm 3.* During a process of committing GenAgent with must_commit condition, there is a possibility that must_commit GenAgent_(i) generates higher negative profit and none of participating committed GenAgents is ready to de-commit as described earlier in Algorithm 4. At this stage, the decision of committing GenAgent_(i) with must_commit condition is revised. It is done by evaluating profit of GenAgent_(i) from $t$ to $t+T_i^{down}$ hours and then, compare it with *original* $Pr_{j,t}$ of GenAgent_(j). Based on the comparison, either must_commit GenAgent is de-committed from $t$ to $t+T_i^{down}$ hours and participating GenAgent_(j) is restored with its original values of $Pr_{j,t}$, $P_{j,t}$ and $R_{j,t}$. Or, GenAgent_(i) and GenAgent_(j) of the best profit combination are committed with their updated values of $Pr_{i,t}$, $P_{i,t}$, $R_{i,t}$, $Pr_{j,t}$, $P_{j,t}$, $R_{j,t}$.

*4.2.5. Constraint satisfaction stage*

Next stage is checking the minimum up and down time constraints given in (8) for all committed GenAgents. Rules are designed either to assign must_commit condition or shut down GenAgent so that the loss in revenue can be minimized.

Rules designed for minimum up and down time constraints are shown in Algorithms 5 and 6. Both algorithms are similar except a specified time period that depends on $t_i^{up}$ and $t_i^{down}$ hours for up and down time constraints, respectively. Based on profit, either must_commit condition is assigned to GenAgent_(i) or shut it down.

**Algorithm 5.** Rules to satisfy minimum down time constraint for GenAgent_(i) at time $t$

Suppose GenAgent_(i) is de-committed for $t_{down}$ hours ($< T_i^{down}$)
*Profit_sum* of GenAgent_(i) = 0
**for** $k = t - t_{down} \rightarrow t - t_{down} + T_i^{down}$ **do**
  **if** $k < t$ **then**
    **if** $DL_t \geq P_i^{min}$ **then**
      *Profit_sum* of GenAgent_(i) = *Profit_sum* +$Pr_{i,t}$ evaluated by Algorithm 1 for given $DL_t$
    **else**
      *Profit_sum* of GenAgent_(i) = *Profit_sum* +$Pr_{i,t}$ evaluated by Algorithm 1 for $P_i^{min}$
    **end if**
  **else**
    *Profit_sum* of GenAgent_(i) = *Profit_sum* +$Pr_{k,t}$
  **end if**
**end for**
**if** *Profit_sum* of GenAgent_(i) >$ST_i$ of GenAgent_(i) **then**
  Assign must_commit condition to GenAgent_(i) from $t - t_{down}$ to $t - 1$ hours
**else**
  Shut-down GenAgent_(i) from $t$ to $t - t^{down} + T_i^{down}$ hours
  Update $DL_t$ and $RL_t$ for $t$ to $t - t^{down} + T_i^{down}$ hours
**end if**

**Algorithm 6.** Rules to satisfy minimum up time constraint for GenAgent_(i) at time $t$

Suppose GenAgent_(i) is committed for $t_{up}$ hours ($< T_i^{up}$)
*Profit_sum* of GenAgent_(i) = 0
**for** $k = t - t_{up} \rightarrow t - t_{up} + T_i^{up}$ **do**
  **if** $k < t$ **then**
    *Profit_sum* of GenAgent_(i) = *Profit_sum* +$Pr_{k,t}$
  **else**
    **if** $DL_t \geq P_i^{min}$ **then**
      *Profit_sum* of GenAgent_(i) = *Profit_sum* +$Pr_{i,t}$ evaluated by Algorithm 1 for given $DL_t$
    **else**
      *Profit_sum* of GenAgent_(i) = *Profit_sum* +$Pr_{i,t}$ evaluated by Algorithm 1 for $P_i^{min}$
    **end if**
  **end if**
**end for**
if *Profit_sum* of GenAgent_(i) >$ST_i$ of GenAgent_(i) **then**
  Assign must_commit condition to GenAgent_(i) from $t$ to $t - t_{up} + T_i^{up}$ hours
**else**
  Shut-down GenAgent_(i) from $t - t^{up}$ to $t - 1$ hours
  Update $DL_t$ and $RL_t$ for $t - t^{up}$ to $t - 1$ hours
**end if**

#### 4.2.6. Termination conditions

CA terminates the proposed multi-agent approach when two conditions are satisfied simultaneously. First condition satisfies when profit values from committed GenAgents of current and last iterations are same. Second termination condition triggers when none of GenAgent with must_commit condition exists in a system of GenAgents. If any of these conditions is not met, an iteration count increases by 1. The procedure of Fig. 3 starts from interaction stage and $t = 1$ h.

## 5. Simulation results and discussion

Simulations using proposed multi-agent modeling for solving PBUC are performed for different systems consisting ten to 100 thermal generating units. These systems of generators can be owned by a GENCO for a day ahead scheduling. The ten generation units data and forecast data is taken from [12]. Table A.4 in Appendix A shows generating unit's data required to perform the simulations. In this table, the data of every row is self explanatory from nomenclature except "Ini." which suggests that a generator is either ON (positive integer) or OFF (negative integer) for given number of hours at time $t = 0$. The data of forecast load demand and forecast spot price for ten generators' system is given in Table A.5. For testing out a larger system, the thermal generator data is obtained by duplicating the ten generators' data. The value of forecast load demand is adjusted in proportion to the generator size. In following paragraphs, the results of proposed multi-agent approach are discussed and benchmarked against the commonly used techniques such as Lagrangian relaxation (LR), genetic algorithm (GA) and hybrid LRGA. A comparison of distributed artificial intelligence based multi-agent approach with the mathematical technique, the meta-heuristic algorithm and the hybrid stochastic method can help us to analyze an overall performance of proposed approach on the solution quality and on the execution time as well. Note that the ramp up and down constraints are not considered for benchmarking purpose.

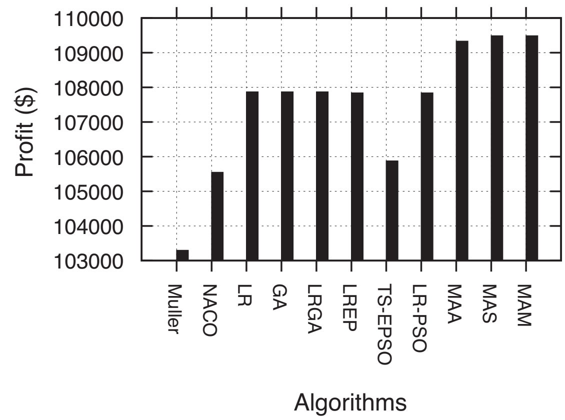


**Fig. 6.** Best profit solutions obtained for ten generator system by different algorithms.

### 5.1. Parameter setting

For LR, a relative duality gap ≤0.01 is set. Lagrangian multipliers are tuned for every system to generate best profit results. For economic power dispatch, Lambda-iteration method is used. GA proposed in study [4] is used which terminates when either 100 generations are over or variation in the best fitness solutions over consecutive 20 generations is ≤0.1. For ten to 60 generators, population of 100 is used, whereas for larger generator sets population of 150 members are used to evolve the best profit solutions. The hybrid LRGA [32] terminates when a duality gap ≤0.001. For updating Lagrangian multiplier, GA is executed for 50 generations with the population of 100. The source codes are developed on MatLab platform and executed on 64 bit Intel CPU @ 2.66GHz with 16 GB RAM. For benchmarking the results of proposed multi-agent modeling against [10,12,15,16,32–35] studies, PBUC problem is solved for $r = 0.005$ and $RP_t = 0.01 \times SP_t$.

### 5.2. Best profit solutions

#### 5.2.1. Ten generators system

For ten generators system, various algorithms have been reported in the literature for finding best profit solutions of PBUC problem. These algorithms are muller method by [33], nodal any colony optimization (NACO) by [10], LR and GA by [4], LRGA by [32], LREP by [12], tabu search EPSO (TS-EPSO) by [34], LR-PSO by [35], multi-agent approach (MAA) by the authors of this paper [16], and multi-agent system (MAS) given by [15]. The proposed multi-agent modeling for PBUC is referred as "MAM" in this paper. The best profit solutions by these algorithms are shown in Fig. 6. It can be observed from figure that the approaches based on multi-agent system generated the higher profit solutions than the mathematical

**Table 1**
Optimal schedule for ten generator system ($r$ = 0.005 and $RP_t$ = 0.01$SP_t$).

| Power (MW)/reserve (MW) | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 | TC ($) | Profit ($) |
| 1 | 455.0/0.0 | 245.0/70.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 13689.23 | 1838.95 |
| 2 | 455.0/0.0 | 295.0/75.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 14561.05 | 1963.62 |
| 3 | 455.0/0.0 | 395.0/60.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 16307.15 | 3348.57 |
| 4 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 17353.30 | 3258.20 |
| 5 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 17353.30 | 3804.20 |
| 6 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 3094.04 |
| 7 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 3186.04 |
| 8 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 2822.04 |
| 9 | 455.0/0.0 | 455.0/0.0 | 130.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 23105.76 | 3020.24 |
| 10 | 455.0/0.0 | 455.0/0.0 | 130.0/0.0 | 130.0/0.0 | 162.0/0.0 | 68.0/12.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 28769.61 | 11255.65 |
| 11 | 455.0/0.0 | 455.0/0.0 | 130.0/0.0 | 130.0/0.0 | 162.0/0.0 | 80.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 29047.98 | 13523.82 |
| 12 | 455.0/0.0 | 455.0/0.0 | 130.0/0.0 | 130.0/0.0 | 162.0/0.0 | 80.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 29047.98 | 15641.82 |
| 13 | 455.0/0.0 | 455.0/0.0 | 130.0/0.0 | 130.0/0.0 | 162.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 26851.61 | 5915.59 |
| 14 | 455.0/0.0 | 455.0/0.0 | 130.0/0.0 | 130.0/0.0 | 130.0/32.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 26187.36 | 5674.36 |
| 15 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 160.0/2.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 23918.06 | 3082.62 |
| 16 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 2978.04 |
| 17 | 455.0/0.0 | 415.0/40.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 19516.28 | 2747.03 |
| 18 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 2718.04 |
| 19 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 2874.04 |
| 20 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 3342.04 |
| 21 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 3810.04 |
| 22 | 455.0/0.0 | 455.0/0.0 | 0.0/0.0 | 130.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 20213.96 | 3654.04 |
| 23 | 455.0/0.0 | 445.0/10.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 17178.79 | 3299.61 |
| 24 | 455.0/0.0 | 345.0/80.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 0.0/0.0 | 15434.42 | 2632.55 |
| ST ($) | 0.00 | 0.00 | 550.00 | 560.00 | 900.00 | 170.00 | 0.00 | 0.00 | 0.00 | 0.00 | – | – |
| TC ($) | 203179.77 | 203005.95 | 17350.80 | 48631.20 | 21769.10 | 6310.74 | 0.00 | 0.00 | 0.00 | 0.00 | 500247.59 | – |
| PF ($) | 55760.79 | 42891.32 | 3295.70 | 4193.30 | 2879.79 | 464.33 | 0.00 | 0.00 | 0.00 | 0.00 | – | **109485.23** |

techniques, the meta-heuristic algorithms and the hybrid stochastic methods reported in the literature. MAM and MAS generated the best profit solution for ten generators system. The reason behind an improved solution for PBUC is the developed rules for intelligent scheduling the generators. However, the modeling of multi-agent system for MAM is based in fixed rules, whereas stochastic rules are developed for MAS.

A day-ahead scheduling by MAM is shown in Table 1. During the process of scheduling, the power sharing rules were negotiated 43 times between potential GenAgents and participating GenAgents via CA. Moreover, one of the participated committed GenAgents was shutdown for some period to maximize an expected profit of GENCO. For example in Table 1, G3 is de-committed from hour 15 which allows G4 to commit at its $P_i^{max}$ at the same hour. The power sharing rules further assist G5 to commit for more load demand. Another good example of power sharing can be seen at hour 17 when G2 shares its power with G4.

#### 5.2.2. Simulation results for large systems

The above techniques reported in the literature have been tested on ten generator system only. However, a performance of these algorithms can be really tested when they are executed on a larger system where the profit solution and time to generate this solution are equally important. There are some benchmark techniques which are frequently referred in the literature and are used for unit commitment problem in a vertically integrated power system. In this paper, LR, GA and LRGA techniques have been chosen for benchmarking our results for larger and scaled-up systems. A comparison of best profit solutions obtained by MAM and benchmarked algorithms is shown in Fig. 7. It can be seen from figures that MAM generated the best profit solutions which indicates its supremacy over the benchmarked algorithms for a wide range of thermal generator systems. In Fig. 7(f), the gain in profit values using MAM is shown against LRGA. Among the benchmarked techniques, LRGA is chosen because it generated the next best profit solutions for every system of generators. It can be seen that the gain in profit values ranges from $1613.6 for ten generator system to $18985.94 for 100 generator system that a generator company can earn in a single day. It suggests that MAM can help GENCO to accumulate a huge annual profit.

### 5.3. Working behavior of MAM

In this section, a posterior analysis is done to capture the communication and negotiation among the agents that define a working behavior of MAM. Various data is shown in Table 2. Second column of table shows number of executions for the power sharing rules shown in Figs. 4 and 5. An increasing trend of calling the power sharing rules by CA indicates their significance for exploring and finding the intelligent schedule corresponding to the best profit solution.

**Table 2**
Posterior analysis.

| No. of generators | No. of power sharing rules execution | No. of must_commit GenAgents to be de-committed | No. of participating committed GenAgents to be de-committed |
|---|---|---|---|
| 20 | 157 | 0 | 2 |
| 40 | 687 | 0 | 5 |
| 60 | 1262 | 0 | 7 |
| 80 | 3221 | 1 | 9 |
| 100 | 3929 | 4 | 13 |

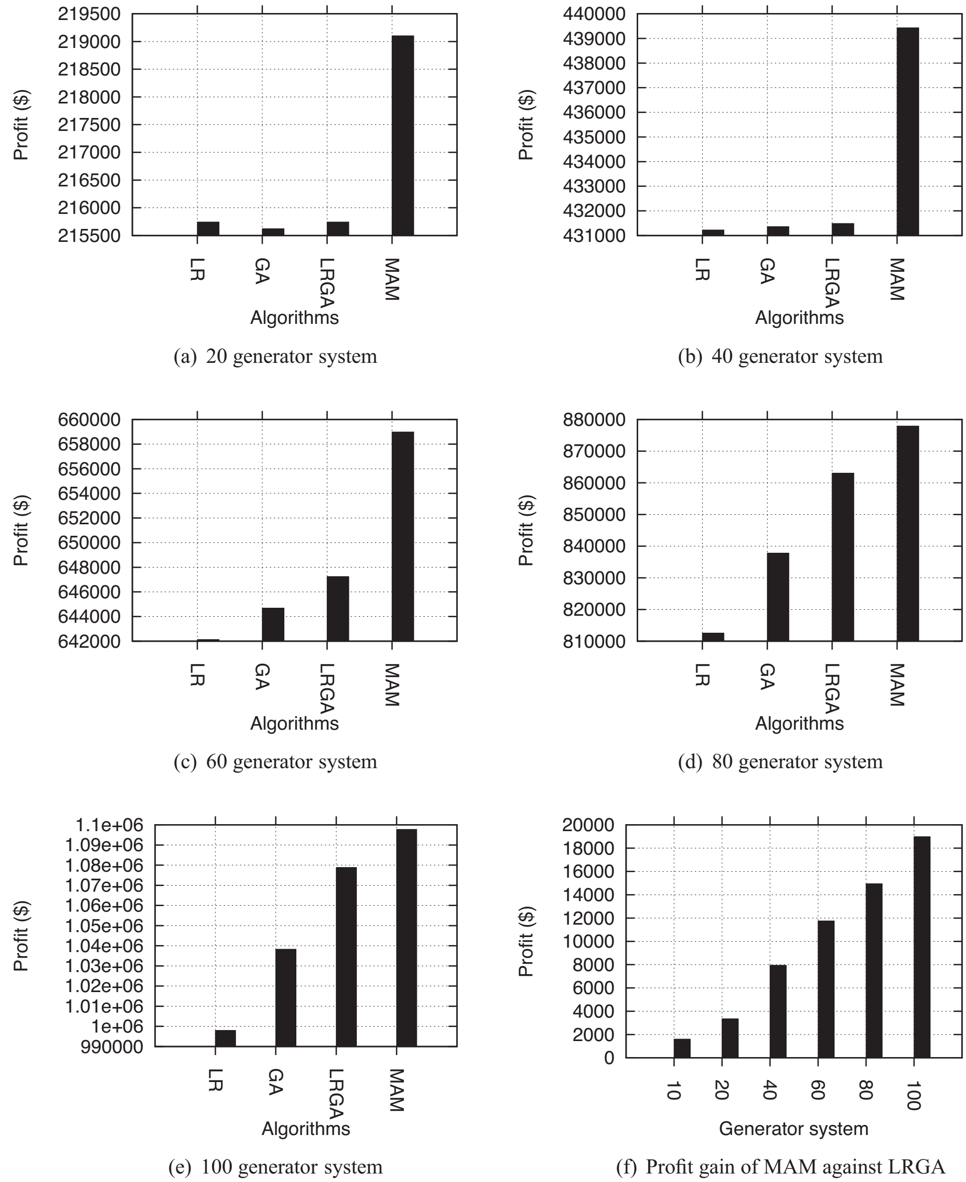


**Fig. 7.** Profit generated by algorithms for scaled-up systems of generators.

In Section 4.2.4.3, a possibility to de-commit a participating committed GenAgent is explored that can help in accumulating profit for a GENCO. Fourth column of Table 2 indicates an increasing values of such generators so that must_commit GenAgent can commit for more power.

In Section 4.2.4, a decision of committing must_commit GenAgent is revised and a possibility to de-commit it is explored. Third column of Table 2 reveals the importance of this rule when a GENCO owns a large number of generating units.

**Table 3**
Comparison of execution time (s).

| Techniques | LR | GA | LR-GA | MAM |
|---|---|---|---|---|
| No. of generators | | | | |
| 10 | 111 | 752 | 1112 | 6 |
| 20 | 198 | 1323 | 1230 | 7 |
| 40 | 368 | 2321 | 1365 | 10 |
| 60 | 571 | 3821 | 1498 | 13 |
| 80 | 851 | 5099 | 1954 | 19 |
| 100 | 1298 | 8235 | 2537 | 27 |

### 5.4. Execution time

Table 3 shows a comparison of execution time which suggests a remarkable improvement in the CPU time of MAM against LR, GA and LRGA techniques. MAM consumes less time because the rules are executed which depend on examining various conditions. On the other hand, the optimization techniques improve quality of solution iteratively where more function evaluations is required to generate an optimal solution. Moreover, the computation time suggests that MAM can be used for a shorter period scheduling than a day ahead.

### 5.5. Comparison of performance

#### 5.5.1. Comparison between LR and proposed MAM

It is interesting to note that LR solves generation scheduling problem by decomposing Lagrange formulation [12]. Lagrangian function is minimized for each generator separately without considering any affect on other generators. Similarly, MAM also commits maximum profit generating GenAgents. For economic power dispatch, LR based methods use optimization technique where Lagrangian multiplier are to be tuned for different generator system. However, MAM delivers the desired power and reserve based on the rules developed under cooperative stage as discussed in Section 4.2.4. Moreover, MAM is not sensitive to any parameter setting.

*5.5.2. Comparison with other multi-agent system based studies*

As mentioned in Section 1, two multi-agent system based studies for solving PBUC have been reported in the literature. In first study by [15] which is referred as MAS, the negotiation of profit and power is designed on random traveling routes for a mobile agent to visit generators agents. However, the fixed rules are designed for communication and negotiation with generator agents in this paper. Stochasticity in random traveling routes is involved in MAS approach that can generate different solutions for same load demand condition when execute at different time. On the other hand, MAM can give consistent results.

Other study was our previous work [16] which is referred as MAA. In MAA, the maximum profit generating generator agents are committed for remaining load demand. It is similar to the competing stage (Section 4.2.3) of MAM, but the working behavior is different. For example, when a maximum profit generating generator agent is committed, the load demand is updated by MAA. The generator agents which are currently OFF, then evaluate their profit, power and reserve using genetic algorithm for the remaining demand and reserve. In MAM, once a profit is evaluated according to Eq. (9) by GenAgents at their $P_i^{max}$, these GenAgents are sorted and committed one-by-one till $DL_t \geq P_i^{max}$. By doing this, number of function evaluations or computation time is saved. In our previous study, a heuristic repair operator is used. However in this paper, the cooperative stage (Section 4.2.4) with different power sharing scenarios and cases is designed. The rules to de-commit generator agents (Sections 4.2.4.3 and 4.2.4.4), and assigning must_commit condition for satisfying minimum up and down time constraints (Section 4.2.5) are also introduced for MAM. The rules and conditions designed in this paper make MAM effective and able to generate the best profit solutions against the benchmarked techniques.

## 6. Conclusion

A multi-agent modeling for solving PBUC has been proposed for GENCO which is an independent and autonomous generation utility in the deregulated power system. MAM is developed for profit maximization via intelligent scheduling the generators and allocating power economically. The proposed MAM for GENCO has been tested on different systems having small to large number of generating units where it generated the *best profit solutions* in *less computation time* against the benchmark techniques. The rules designed for various agents were able to explore many scenarios for profit maximization. Compared to the parameter sensitive techniques like LR, the proposed MAM is parameter-less. Moreover, MAM can give consistent results as its working principle is defined by communication and negotiating rules which do not involve heuristic or stochasticity like GA, LRGA, etc. As MAM consumes very little computation time, it can be used for a shorter-term scheduling than a day-ahead. Future work will consider further addition of new intelligent rules to make MAM more efficient while incorporating ramp and network constraints.

## Acknowledgment

Authors acknowledge the support from National Research Foundation, Singapore (Grant no. NRF-2007EWT-CERP01-0954 (R-263-000-522-272)).

## Appendix A.

See Tables A.4 and A.5.

**Table A.4**
Generating unit data.

| | Unit 1 | Unit 2 | Unit 3 | Unit 4 | Unit 5 |
|---|---|---|---|---|---|
| $P_i^{max}$ | 455 | 455 | 130 | 130 | 162 |
| $P_i^{min}$ | 150 | 150 | 20 | 20 | 25 |
| $a$ | 1000 | 970 | 700 | 680 | 450 |
| $b$ | 16.19 | 17.26 | 16.60 | 16.50 | 19.70 |
| $c$ | 0.00048 | 0.00031 | 0.00200 | 0.00211 | 0.00398 |
| $T_i^{up}$ | 8 | 8 | 5 | 5 | 6 |
| $T_i^{down}$ | 8 | 8 | 5 | 5 | 6 |
| $ST_i$ | 4500 | 5000 | 550 | 560 | 900 |
| Ini. | 8 | 8 | −5 | −5 | −6 |

| | Unit 6 | Unit 7 | Unit 8 | Unit 9 | Unit 10 |
|---|---|---|---|---|---|
| $P_i^{max}$ | 80 | 85 | 55 | 55 | 55 |
| $P_i^{min}$ | 20 | 25 | 10 | 10 | 10 |
| $a$ | 370 | 480 | 660 | 665 | 670 |
| $b$ | 22.26 | 27.74 | 25.92 | 27.27 | 27.79 |
| $c$ | 0.00712 | 0.00079 | 0.00413 | 0.00222 | 0.00173 |
| $T_i^{up}$ | 3 | 3 | 1 | 1 | 1 |
| $T_i^{down}$ | 3 | 3 | 1 | 1 | 1 |
| $ST_i$ | 170 | 260 | 30 | 30 | 30 |
| Ini. | −3 | −3 | −1 | −1 | −1 |

**Table A.5**
Forecast demand (FD) and spot price (SP) for ten-unit 24-period system.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Hour | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| FD (MW) | 700 | 750 | 850 | 950 | 1000 | 1100 | 1150 |
| SP ($/MW-H) | 22.15 | 22.00 | 23.10 | 22.62 | 23.25 | 22.95 | 22.50 |
| Hour | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| FD (MW) | 1200 | 1300 | 1400 | 1450 | 1500 | 1400 | 1300 |
| SP ($/MW-H) | 22.15 | 22.80 | 29.35 | 30.15 | 31.65 | 24.60 | 24.50 |
| Hour | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| FD (MW) | 1200 | 1050 | 1000 | 1100 | 1200 | 1400 | 1300 |
| SP ($/MW-H) | 22.50 | 22.30 | 22.25 | 22.05 | 22.20 | 22.65 | 23.10 |
| Hour | 22 | 23 | 24 | | | | |
| FD (MW) | 1100 | 900 | 800 | | | | |
| SP ($/MW-H) | 22.95 | 22.75 | 22.55 | | | | |

## References

[1] T. Senjyu, K. Shimabukuro, K. Uezato, T. Funabashi, A fast technique for unit commitment problem by extended priority list, IEEE Transactions on Power Systems 18 (2) (2003) 882–888, http://dx.doi.org/10.1109/TPWRS.2003.811000.
[2] W.L. Snyder, H.D. Powell, J.C. Rayburn, Dynamic programming approach to unit commitment, IEEE Transactions on Power Systems 2 (2) (1987) 339–348, http://dx.doi.org/10.1109/TPWRS.1987.4335130.
[3] A. Merlin, P. Sandrin, A new method for unit commitment at electricite de france, IEEE Transactions on Power Apparatus and Systems PAS-102 (5) (1983) 1218–1225, http://dx.doi.org/10.1109/TPAS.1983.318063.
[4] J. Richter, C.W.G. Sheble, A profit-based unit commitment GA for the competitive environment, IEEE Transactions on Power Systems 15 (2) (2000) 715–721, http://dx.doi.org/10.1109/59.867164.
[5] T. Ting, M. Rao, C. Loo, A novel approach for unit commitment problem via an effective hybrid particle swarm optimization, IEEE Transactions on Power Systems 21 (1) (2006) 411–418, http://dx.doi.org/10.1109/TPWRS.2005.860907.
[6] I. Jacob Raglend, C. Raghuveer, G. Rakesh Avinash, N.P. Padhy, D.P. Kothari, Solution to profit based unit commitment problem using

particle swarm optimization, Applied Soft Computing 10 (4) (2010) 1247–1256, http://dx.doi.org/10.1016/j.asoc.2010.05.006.

[7] K. Juste, H. Kita, E. Tanaka, J. Hasegawa, An evolutionary programming solution to the unit commitment problem, IEEE Transactions on Power Systems 14 (4) (1999) 1452–1459, http://dx.doi.org/10.1109/59.801925.

[8] V. Dieu, W. Ongsakul, Enhanced augmented lagrangian hopfield network for unit commitment, IEE Proceedings – Generation, Transmission and Distribution 153 (6) (2006) 624–632, http://dx.doi.org/10.1049/ip-gtd:20050460.

[9] G. Purushothama, L. Jenkins, Simulated annealing with local search – a hybrid algorithm for unit commitment, IEEE Transactions on Power Systems 18 (1) (2003) 273–278, http://dx.doi.org/10.1109/TPWRS.2002.807069.

[10] C.C. Columbus, K. Chandrasekaran, S.P. Simon, Nodal ant colony optimization for solving profit based unit commitment problem for gencos, Applied Soft Computing 12 (1) (2012) 145–160, http://dx.doi.org/10.1016/j.asoc.2011.08.057.

[11] K. Vaisakh, L.R. Srinivas, Evolving ant colony optimization based unit commitment, Applied Soft Computing 11 (2) (2011) 2863–2870, http://dx.doi.org/10.1016/j.asoc.2010.11.019.

[12] P. Attaviriyanupap, H. Kita, E. Tanaka, J. Hasegawa, A hybrid LR-EP for solving new profit-based UC problem under competitive environment, IEEE Transactions on Power Systems 18 (1) (2003) 229–237, http://dx.doi.org/10.1109/TPWRS.2002.807080.

[13] C.-P. Cheng, C.-W. Liu, C.-C. Liu, Unit commitment by lagrangian relaxation and genetic algorithms, IEEE Transactions on Power Systems 15 (2) (2000) 707–714, http://dx.doi.org/10.1109/59.867163.

[14] C. Rajan, M. Mohan, An evolutionary programming-based tabu search method for solving the unit commitment problem, IEEE Transactions on Power Systems 19 (1) (2004) 577–585, http://dx.doi.org/10.1109/TPWRS.2003.821472.

[15] J. Yu, J. Zhou, B. Hua, R. Liao, Optimal short-term generation scheduling with multi-agent system under a deregulated power market, International Journal of Computational Cognition 3 (2) (2005) 61–65.

[16] D. Sharma, D. Srinivasan, A. Trivedi, Multi-agent approach for profit based unit commitment, in: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, USA, 2011, pp. 2507–2513.

[17] M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, Knowledge Engineering Review 10 (2) (1995) 115–152.

[18] P. Tichý, P. Kadera, R.J. Staron, P. Vrba, V. Mařík, Multi-agent system design and integration via agent development environment, Engineering Applications of Artificial Intelligence (0) (2011), http://dx.doi.org/10.1016/j.engappai.2011.09.021.

[19] S. McArthur, E. Davidson, V. Catterson, A. Dimeas, N. Hatziargyriou, F. Ponci, T. Funabashi, Multi-agent systems for power engineering applications-part I: concepts, approaches, and technical challenges, IEEE Transactions on Power Systems 22 (4) (2007) 1743–1752, http://dx.doi.org/10.1109/TPWRS.2007.908471.

[20] P. Dhavachelvan, G. Uma, Multi-agent-based integrated framework for intra-class testing of object-oriented software, Applied Soft Computing 5 (2) (2005) 205–222, http://dx.doi.org/10.1016/j.asoc.2004.04.004.

[21] S. Mukkamala, A.H. Sung, A. Abraham, Hybrid multi-agent framework for detection of stealthy probes, Applied Soft Computing 7 (3) (2007) 631–641, http://dx.doi.org/10.1016/j.asoc.2005.12.002.

[22] C.-J. Su, C.-Y. Wu, Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring, Applied Soft Computing 11 (1) (2011) 315–325, http://dx.doi.org/10.1016/j.asoc.2009.11.022.

[23] R. Erol, C. Sahin, A. Baykasoglu, V. Kaplanoglu, A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems, Applied Soft Computing 12 (6) (2012) 1720–1732, http://dx.doi.org/10.1016/j.asoc.2012.02.001.

[24] M.S. Hamdi, Masacad: A multi-agent approach to information customization for the purpose of academic advising of students, Applied Soft Computing 7 (3) (2007) 746–771, http://dx.doi.org/10.1016/j.asoc.2006.02.001.

[25] A.J. Kulkarni, K. Tai, Probability collectives: A multi-agent approach for solving combinatorial optimization problems, Applied Soft Computing 10 (3) (2010) 759–771, http://dx.doi.org/10.1016/j.asoc.2009.09.006.

[26] A. Mata, J.M. Corchado, D.I. Tapia, Cros: A contingency response multi-agent system for oil spills situations, Applied Soft Computing 11 (3) (2011) 3147–3159, http://dx.doi.org/10.1016/j.asoc.2010.12.017.

[27] L.M. Hercog, Better manufacturing process organization using multi-agent self-organization and co-evolutionary classifier systems: the multibar problem, Applied Soft Computing (0) (2012), http://dx.doi.org/10.1016/j.asoc.2012.04.033.

[28] S. McArthur, E. Davidson, V. Catterson, A. Dimeas, N. Hatziargyriou, F. Ponci, T. Funabashi, Multi-agent systems for power engineering applications-part II: technologies, standards, and tools for building multi-agent systems, IEEE Transactions on Power Systems 22 (4) (2007) 1753–1759, http://dx.doi.org/10.1109/TPWRS.2007.908472.

[29] E. Allen, M. Ilic, Reserve markets for power systems reliability, IEEE Transactions on Power Systems 15 (1) (2000) 228–233, http://dx.doi.org/10.1109/59.852126.

[30] JADE,;1; Java agent development framework, http://jade.tilab.com/index.html

[31] FIPA,;1; Foundation for intelligent physical agents, http://www.fipa.org/

[32] L. Thillainathan, D. Srinivasan, LRGA for solving profit based generation scheduling problem in competitive environment, in: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, USA, 2011, pp. 1153–1159.

[33] K. Chandram, N. Subrahmanyam, M. Sydulu, Improved pre-prepared power demand table with muller method for solving profit based unit commitment, in: IEEE Region 10 Conference TENCON, 2008, pp. 1–6.

[34] H. Mori, K. Okawa, A new meta-heuristic method for profit-based unit commitment under competitive environment, in: IEEE PowerTech., 2009, pp. 1–6.

[35] P. Sriyanyong, Application of particle swarm optimization technique to a profit-based unit commitment problem, in: 2nd International Conference on Education Technology and Computer (ICETC), 2010, pp. 1–6.