A Multiagent-System-Based Intelligent Reference Governor for Multiobjective Optimal Power Plant Operation

Jin S. Heo and Kwang Y. Lee, *Life Fellow, IEEE*

***Abstract*—A large-scale power plant requires optimal set points, namely references, in several control loops for multiobjective optimal operation. In a 600-MW oil-fired drum-type boiler power unit, the set points considered are for the main steam pressure and reheater/superheater steam temperatures. The set points should be mapped with the varying unit load demand and satisfy the conflicting requirements in power plant operation. In practice, the set points are obtained using fixed nonlinear functions in the unit master control in a plant, which are designed for the single objective of load tracking with heat balance. However, it does not allow for process optimization under the multitude of conflicting objectives, which may be newly introduced and different from the initial design objective. This paper presents a methodology, multiagent-system-based intelligent reference governor (MAS-IRG), to realize the optimal mapping by searching for the best solution to the multiobjective optimization problem that tackles conflicting requirements. In searching for the optimal set points, a heuristic optimization tool, particle swarm optimization, is utilized to solve the multiobjective optimization problem. The IRG is designed based on the proposed MAS to operate at a higher level of automation, to execute asynchronous computations, and to reduce the computational complexity. The approach provides the means to specify optimal set points for controllers under a diverse operating scenarios online.**

***Index Terms*—Multiagent system (MAS), multiobjective optimization, optimal set point scheduling, particle swarm optimization (PSO), power plant control, reference governor, unit master control.**

I. Introduction

WHILE the demand in power is increasing, power plants are getting larger and more complex to run. In order to achieve an optimal operation, optimal set points, namely references, are required for the power plant control system. In practice, the set points are obtained by using fixed nonlinear functions in the unit master control in a plant, which are designed for the single objective of load tracking with heat balance. However, it does not allow for process optimization under the multitude of conflicting objectives, which may be newly introduced and different from the initial design objective. In general, the set points need to be scheduled by considering conflicting operational requirements such as minimization of load-tracking error, minimization of fuel consumption and heat loss rate, maximization of duty life, minimization of pollutant emissions, etc. These conflicting requirements can be tackled by a multiobjective optimization problem in generating optimal set points.

However, the multiobjective optimization problem for power plant operation lies not only in the generation of optimal set points but also in the design of architecture for control systems. Standard optimization methods for a large-scale multiple-input multiple-output (MIMO) nonlinear system result in a heavy computational burden if they are used for generating the optimal set points. Moreover, traditional optimization techniques may often become computationally unattractive or even unacceptable [1]. Control system architectures have been considered to reduce the computational complexity and manage the huge amount of distributed data and coupling problems among many subsystems.

Recently, there has been a growing interest in heuristic optimization techniques, genetic algorithm (GA) and particle swarm optimization (PSO), as variations of evolutionary algorithm (EA). It has been shown that it can provide quality solutions and fast convergences in many applications [4]–[13]. However, in previous studies [3]–[5], the performance of GA is lower than PSO techniques for the solution quality, convergence rate, and computational complexity. On the other hand, the study of multiagent systems (MASs) has become an important aspect of power system architecture in order to deal successfully with the problems of complexity and large-scale distributed systems. Each agent system has special functions in solving the distributed systems. In addition, in the MAS, the agents can work together to solve problems, which are beyond the capabilities or knowledge of an individual agent [26].

As an optimal set-point generator for optimal control actions, the reference governor has been developed using a goal programming (GP) method, GA, and PSO for small-scale power plants [2]–[5]. Moreover, the comparison among the variations of PSO has been investigated within the reference governor [4]. There have been a few researches for reference governor in other application areas. A reference governor was designed for a predictive control to provide the references in the prediction horizon [14]. The reference governor was also designed for systems with state and control constraints [15].

The previous studies [4], [5] presented the concept of the reference governor for a small-scale power plant that was a third-order nonlinear MIMO, fossil-fuel power plant model. For the small-scale power plant, the implementation of reference governor did not require the MAS concept to reduce computational

Manuscript received March 7, 2006; revised January 24, 2007. Current version published November 21, 2008. Paper no. TEC-00069-2006.

The authors are with the Department of Electrical & Computer Engineering, Baylor University, Waco, TX 76798 USA (e-mail: juh138@gmail.com; kwang_Y_Lee@baylor.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEC.2008.2001459

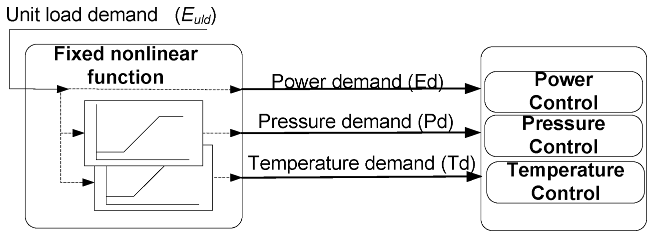


Fig. 1. Fixed nonlinear mapping for the references.

complexity and distributed and asynchronous problems. However, as the scale of the systems increases, control systems require a new framework to reduce the computational complexity and manage the huge amount of distributed data. Thus, the reference governor for a large-scale power plant is required to be developed based on the concept of MAS proposed in [28]. The fundamental concept of MAS is the cooperation of distributed multiple agents performing their jobs independently. Many applications of MASs or agent-oriented systems have been presented in control and monitoring systems to overcome the problems associated with large-scale distributed systems [21]–[30].

Although many studies have been done on the reference governor, it is developed for small-scale power plants. Design procedures for the reference governor are unconvincing for a large-scale power plant. Moreover, although there are many applications of MAS for distributed systems, little information is available for control system design of power plants. The primary focus of this paper is on the development of an intelligent reference governor (IRG) for multiobjective power plant operation with the proposed architectures of a single agent and an MAS. The multiobjective optimal power plant operation will be achieved by minimizing load-tracking error, fuel consumption, heat loss rate and pollutant emission, and maximizing duty life on equipment.

In a 600-MW oil-fired drum-type boiler power unit, the set points considered are for the main steam pressure and reheater/superheater steam temperatures. The set points should be mapped by varying the unit load demand (ULD), and they should satisfy the conflicting operation requirements of the power plant. In general, the set points obtained by using a fixed nonlinear function cannot provide optimal power plant operation. Fig. 1 shows the fixed nonlinear functions. The set points are obtained using fixed nonlinear functions that are designed for the single objective of load tracking with heat balance. However, it does not allow for process optimization under the newly introduced multitude of conflicting objectives, which are different from the initial design objective.

This paper presents a methodology, MAS-based IRG (MAS-IRG), to realize the optimal mapping by searching for the best solution to the multiobjective optimization problem that tackles the conflicting requirements. In searching for the optimal set points, a heuristic optimization tool PSO is utilized for the multiobjective optimization. The IRG is designed based on the proposed MAS to operate at a higher level of automation, execute asynchronous computations, and reduce the computational complexity. The approach provides the means to specify optimal set points for controllers under a diverse operating scenarios online.

The proposed MAS-IRG will be one of the functions in the MAS-based intelligent control (MAS-IC) that has several functions such as identification, fault-diagnosis, and modeling that provide efficient way to control locally and globally, and accommodate and overcome the complexity of large-scale distributed power systems. The MAS-IRG is based on the initial concept of MAS in [28], utilizes the steady-state model developed in [29], and can be used for fault-diagnostics and accommodation proposed in [30].

Following Section I, the power plant is described in Section II. Section III shows MAS. Section IV describes MAS-IRG. Section V shows simulation results to demonstrate the feasibility of the proposed approach and the final section draws some conclusions.

II. Power Plant

The power plant under consideration is a 600-MW oil-fired drum-type boiler–turbine–generator unit [17] shown in Fig. 2. It is a balanced draft, controlled recirculation drum boiler capable of delivering 4.2×10^6 lb/h of steam at a pressure of 2600 psig and at 1005 °F. Six recirculation pumps supply the required recirculation flow to provide sufficient flow for full-load operation. Two forced draft fans supply the primary air, and two induced draft fans are controlled to maintain a furnace pressure at a desired preset value. Two condensate pumps and a combined booster and main boiler feedpumps handle the feedwater flow.

The turbine is a tandem compound triple pressure steam turbine. It consists of three parts: a high-pressure turbine, an intermediate pressure turbine, and low twin pressure turbines rotating on a common shaft at a rated speed of 3600 r/min and at an exhausting pressure of 2-in Hg absolute. The generator is coupled with the turbine and features a 685 600 kV·A, three-phase, 60 Hz, 22 kV supply, with a power factor of 0.90 lagging.

There are many power plant models developed over the years [16]. The developed model represents an extension of some existing models [19], [20] in two primary areas. First, the condensate and feedwater side dynamics have been modeled, and second, the electrical prime movers that run fans and pumps and their dependence upon driving voltage and frequency have been modeled. The power plant model developed for the 600-MW unit [17] is validated in MATLAB environment [18]. There are four major modules consisting of 33 subsystems. Each I/O of subsystems is evaluated with data provided in the reference [17]. The model has 23 state variables and 12 control valves $(u_1, u_2, \ldots, u_{12})$ associated with physical processes [18]. In Fig. 2, the control valves are named as following: u_1: fuel flow, u_2: gas recirculation, u_3: induced draft fan, u_4: forced draft fan, u_5: combustor gun (burner) tilt, u_6: superheater spray flow, u_7: reheater spray flow, u_8: governor control valve, u_9: intercept valve, u_{10}: deaerator valve, u_{11}: feedwater valve, and u_{12}: feedpump turbine flow.

The set points are utilized at the distributed controllers. The model is grouped into four main modules, which are boiler

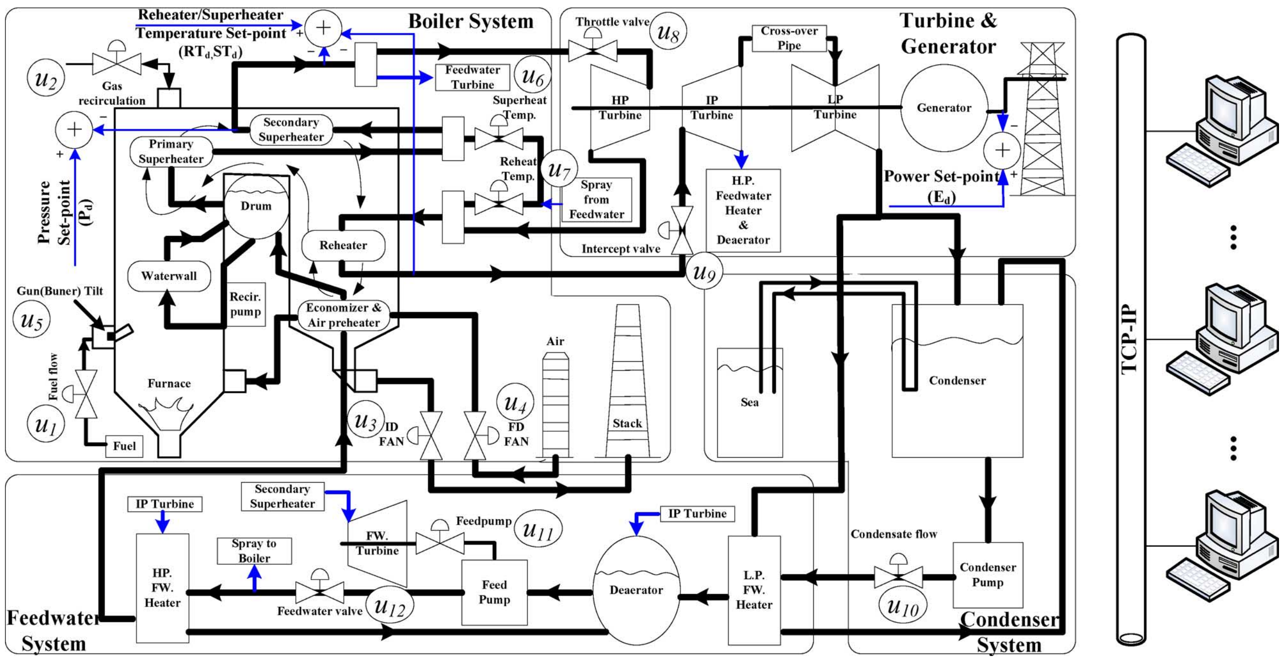


Fig. 2. Large-scale power plant model and MAS.

system, turbine–generator system, condenser system, and feedwater system [19]. The proposed MAS-IRG is one of the functional systems based on the MAS that is interconnected with the 31 subdivided and distributed subsystems that are components of the four main modules. Fig. 2 shows the large-scale distributed thermal power plant model and MAS. Most blocks are subsystems represented by the model. The proposed scheme will be applicable to other types of plants, including nuclear and fuel cell plants.

III. Multiagent System

An agent is a computer software program that is autonomous and situated in some distributed environments in order to meet its design objectives. Since the agents are faced with different environments, they are designed differently and properly for the given environment. Moreover, the agent is intelligent because it is reactive, proactive, social, flexible, and robust. In a large-scale distributed complex system, the agent's autonomous and intelligent properties can reduce the complexity by reducing the coupling problems between the subsystems. Furthermore, the proactive, reactive, and robust properties can be well suited for applications in a dynamic and unreliable situation [26], [27].

In order to design the control systems, design of architecture for a single agent and an organization for MAS are required in advance. First, the architecture of a single agent is shown in Fig. 3. Since the agent is situated in an environment that is the power plant, it needs a perceptor and an effecter to act and react. First, the sensed raw data are processed and mapped into a scenario, and then, an objective, which is a subgoal, is initialized under the situation to achieve the main goal that is the optimal operation. The initial objective is sent to other agents through the communicator for eliminating redundancy and conveying the mission of the agent to others. After confirming the objective,

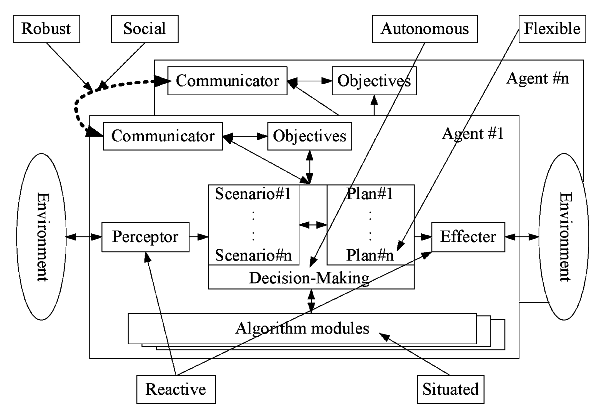


Fig. 3. Single agent architecture.

the best plan is chosen for the objective (subgoal) in the decision making. Depending on the plan, an algorithm module is selected to launch the plan. Finally, the action made by the algorithm module effects through the effecter into the environment. Most decisions are made in the decision-making process, which is like in a human brain [21], [22].

An MAS can be defined as a loosely coupled network (organization) of problem solvers (agents), which interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver (agent). In order to perform the cooperative works, it is presented to build multiple hierarchical structures for the MAS organization, as shown in Fig. 4. The organization has *low level*, *middle level*, and *high level*, and an agent in each level has a specific role in the society so that there is a conceptual idea of supervision for processing the tasks. In this paper, the high-level agents are the *task delegation* and *interface agents*, the middle-level agents are the *mediate* and *monitoring agents*, and the low-level agents are *intelligent agents*.

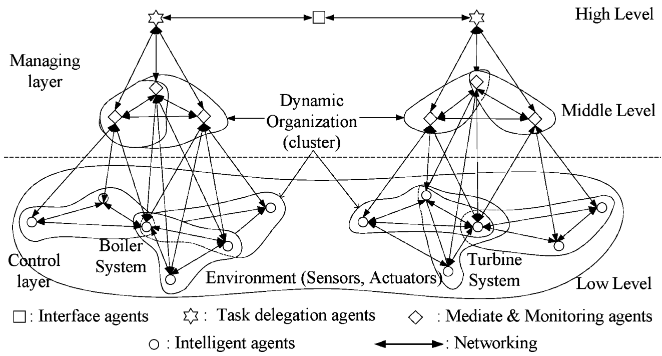


Fig. 4. Organization of MAS.

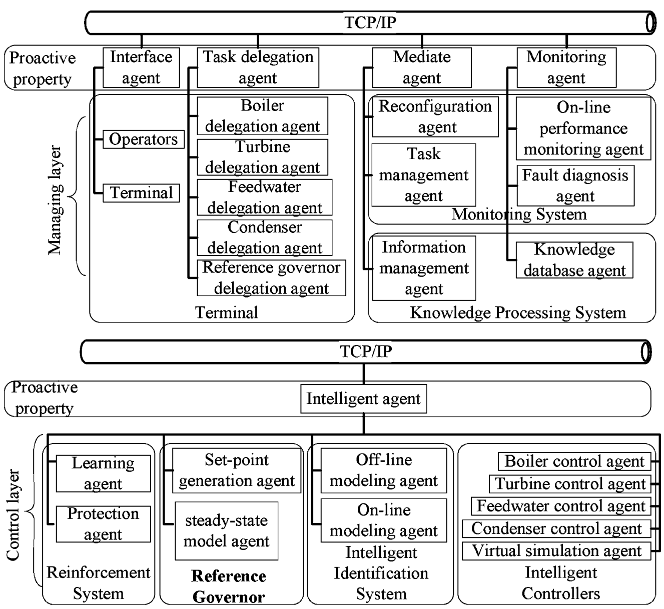


Fig. 5. Composition of MAS-ICS for 600-MW power plant.

The hierarchical structure that has three levels gives advantages for dynamic organization and autonomous systems. Moreover, the idea of multiple hierarchical structures is well suited for large-scale distributed systems [25], [26]. Although there are multiple hierarchical structures, each hierarchical structure has a different formation from others because the structures are constructed to fit for controlling real physical subsystems so that the organization is better optimized for a given power plant system [28]–[30]. Fig. 5 shows the composition of MAS-ICS for the 600-MW power plant.

With the proposed structure of a single agent and the architecture of MAS, the MAS is implemented for the control systems in the simulation environment. The agents are programs that are distributed in high-performance computers. The structure of the program is built upon the proposed single-agent structure. Fig. 6 shows an example for one of the agent programs. The example shows the offline modeling agent that is an agent in the intelligent identification system cluster. Since the power plant simulators recommend the use of distributed computation using PCs or workstations [32], the communication of MAS is developed by using the following proposed scheme. The agents

```
% off-line modeling agent
Function [New_network, New_performance] = off_Identification_agent (system_info, data)  % perceptor1
Data_filtering = smoothing_signal_filter (data)   % perceptor2
Pre_scaled_data = scaling (Data_filtering)  % perceptor3
Performance = Call_database ( system_info, data) % communicate with database agent to obtain old
                                                   % performance of the system
Type_of_system = decision1 (system_info)  % Dynamic Or Static system? Where is located in overall system?
                                            % What is its neighbor systems? Check its history!
Selection_of_plan = decision2 (Type_of_system, Performance)   % choose a algorithm module, error criteria
Objective = Algorithm_modules (Type_of_system,Selection_of_plan,Pre_scaled_data)
Evaluation = decision3 (Objective)  % check the sub goal
Check_other_problems = Call_database (Type_of_system)  % communicate with other systems to find faults
Report = effecter (Results, performance, Check_other_problems) % effect to the environment
```

Fig. 6. Example of one agent in the intelligent identification system cluster.

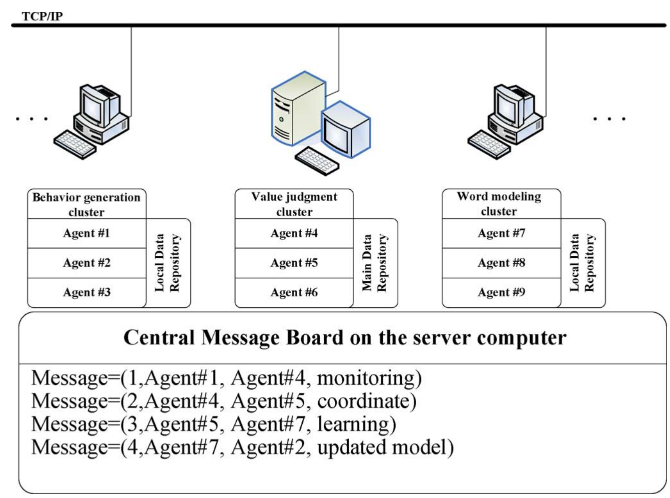


Fig. 7. Message communication in MAS.

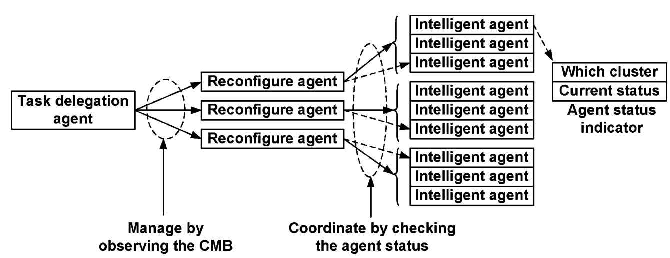


Fig. 8. Task delegation and reconfiguration agents.

are communicating with each other though the Central Message Board (CMB) that is managed by a server computer. Fig. 7 shows the message communication in MAS. The proposed communication protocol based on Transmission Control Protocol (TCP)/IP is designed to provide security, restoration, and status of agents. One of the agents in the clusters keeps checking the CMB and lets receiving agent know that the information has arrived. In order to communicate, all agents are unified by the proposed CMB. The TCP/IP supports the guaranty of delivery. The task delegation agent and reconfiguration agents that are the managing layer agents proceed by observing the CMB and the status of agents, respectively. Fig. 8 shows how the task delegation agent manage the task flow and the reconfiguration agents

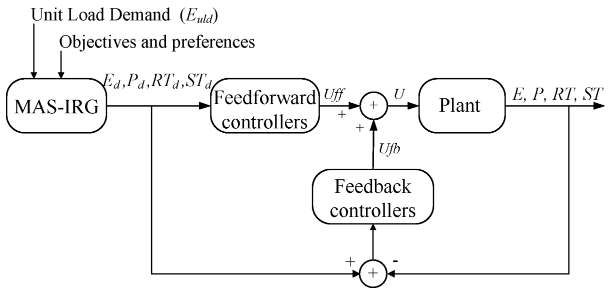


Fig. 9. Control structure of coordinated control.

coordinate the tasks. With the earlier proposed methodologies of MAS, MAS-IRG will be developed in detail in the next section.

IV. Multiagent-System-Based Intelligent Reference Governor

A. Overall Control Structure

There has been several control strategies for the power plant: boiler-following control, turbine-following control, and coordinated boiler–turbine control strategies [31]. In order to make the control system more response to load changes stable and faster, this paper uses the coordinated control scheme, which requires references (or set points) for power demand (E_d), main steam pressure demand (P_d), reheater temperature demand (RT_d), and superheater temperature demand (ST_d). The control structure of the coordinated control is shown in Fig. 9, where the distributed controllers are developed in three main modules: MAS-IRG, feedforward controllers, and feedback controllers. The multiobjective optimization is performed in the MAS-IRG. The results of the multiobjective optimization are the set points for the power, pressure, and temperatures (E_d, P_d, RT_d, ST_d) for the feedforward and feedback controllers. The outputs of the two controllers are added to become input to the power plant. The output of the power plant is fed back to the feedback controller, which regulates the output variations due to load disturbances and compensates for the variations in the load demand.

The essence of the MAS-IRG in the coordinated control is to design the optimal mappings from the ULD, $E_{\rm uld}$, to the set points E_d, P_d, RT_d, and ST_d:

$$SP_E: \ (E_{\rm uld}, t) \rightarrow (E_d, t)$$

$$SP_P: \ (E_{\rm uld}, t) \rightarrow (P_d, t)$$

$$SP_{RT}: (E_{\rm uld}, t) \rightarrow (RT_d, t)$$

$$SP_{ST}: (E_{\rm uld}, t) \rightarrow (ST_d, t)$$

which will be used to transform any ULD pattern $(E_{\rm uld}, t)$ into optimal set-point trajectories for the power (E_d, t), pressure (P_d, t), reheater temperature (RT_d, t), and superheater temperature (ST_d, t) control loops.

The set-point mappings *SP* are designed by solving a multiobjective optimization problem that takes into account the specified operation objectives and the steady-state model of the plant. The MAS-IRG performs the design process in three steps (see Fig. 10).

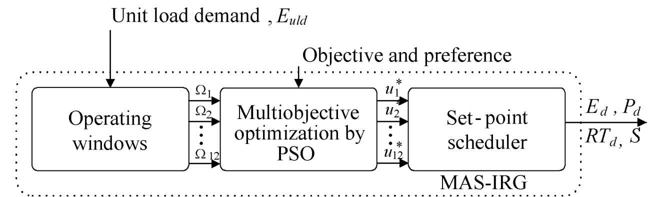


Fig. 10. Configuration of MAS-IRG.

1) Determination of the feasibility regions $(\Omega_1, \Omega_2, \ldots, \Omega_{12})$ for the decision variables $(u_1, u_2, \ldots, u_{12})$.
2) Solution of the multiobjective optimization problem to find optimal steady-state control signals $(u_1^*, u_2^*, \ldots, u_{12}^*)$.
3) Calculation of the set points (E_d, P_d, RT_d, ST_d) through evaluation of the steady-state model of the unit.

The decision variables are candidate steady-state control inputs for the control valves $u_1, u_2, \ldots, u_{12}$, which are shown in Fig. 2.

B. Implementation of MAS-IRG

The MAS-IRG realizes the optimal mapping between set points and varying ULD by searching for the best solution to the multiobjective optimization problem. The set points are considered for the main steam pressure and reheater/superheater steam temperatures in the power unit. The optimal set points are determined by solving the multiobjective optimization problem with conflicting requirements such as load following, fuel conservation, heat loss rate, life extension of equipments, reducing pollution, etc. The composition of MAS for the power plant is shown in Fig. 5. Although all agents are connected with the network, the reference governor cluster, which is made up of a set-point generation agent and a steady-state model agent, performs mainly for the MAS-IRG. However, the reference governor cluster will cooperate with the monitoring system, knowledge processing system, and reinforcement system clusters to obtain better performances. An operator will command and monitor the preference and status through the interface agent to/from the reference governor delegation agent who has all access for the MAS-IRG.

1) Feasibility Regions of Control Inputs: In order to realize the MAS-IRG, first, all feasible operating points, which satisfy all imposed constraints, need to be found using the online performance monitoring agent and virtual simulation agent. The virtual simulation agent simulates the power output responses with various set-point conditions. When system response is in steady state, the constant control inputs and static power, pressure, and temperature outputs form pairs of operating points, where the admissible power outputs can be obtained within an appropriate steam pressure and reheater/superheater temperature ranges. Fig. 11 shows the power output responses with different steam pressure values for the 450-MW power set point. The figure shows that the same power output (450 MW) can be obtained in the steady state with different steam pressure within the range from 1900 to 2900 psia. Similarly, when power set point is fixed at 600 MW that is nominal power, the admissible pressure values are from 2400 to 2800 psia. During the simulation by the virtual simulation agent, the online performance monitoring

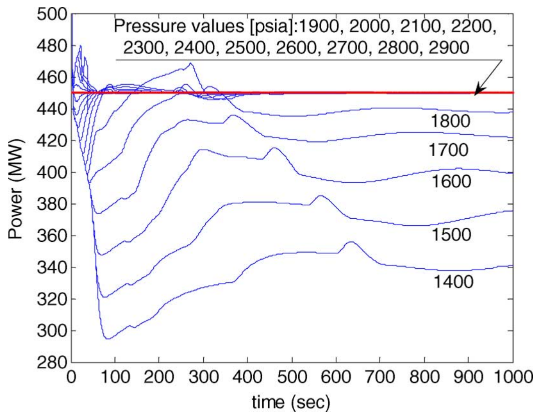


Fig. 11. Power output responses with various pressure set-point conditions for the 450-MW power set point.

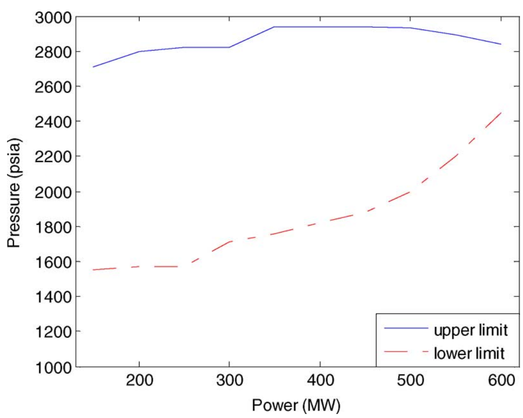


Fig. 12. Power–pressure operating window.

agent evaluates the operating points in order to find the admissible power, steam pressure, and reheater/superheater temperature operating points. The power–pressure operating window is obtained in Fig. 12, which shows that the 450 and 600 MW power are limited in the pressure range of 1900–2900 and 2400–2800 psia, respectively. The reheater/superheater temperature operating range is 1359.67–1459.67 °R (900–1000 °F) for all power ranges. Since the design and operation of reheater are essentially the same as those of the superheater, the reheater and superheater temperature set points are equal. Fig. 13 shows the corresponding power-control input operating windows.

2) Steady-State Model for Evaluation and Calculation of the Set Points: When the target system is a high-order complex system, it is a challenge to get the steady-state model with an analytical approach. Moreover, the model should be adaptive under the changing environment. In order to solve these problems, the steady-state model can be realized intelligently using distributed data analyzer, which can be the MAS. Thus, the next step is the development of approximators for steady-state models using the steady-state model agent in MAS (see Fig. 5). The main algorithm module of the steady-state agent is the neural network (NN), which is considered to be the best approximator for nonlinear systems. The steady-state models are called MAS-based intelligent steady-state models (MAS-ISSMs) [29]

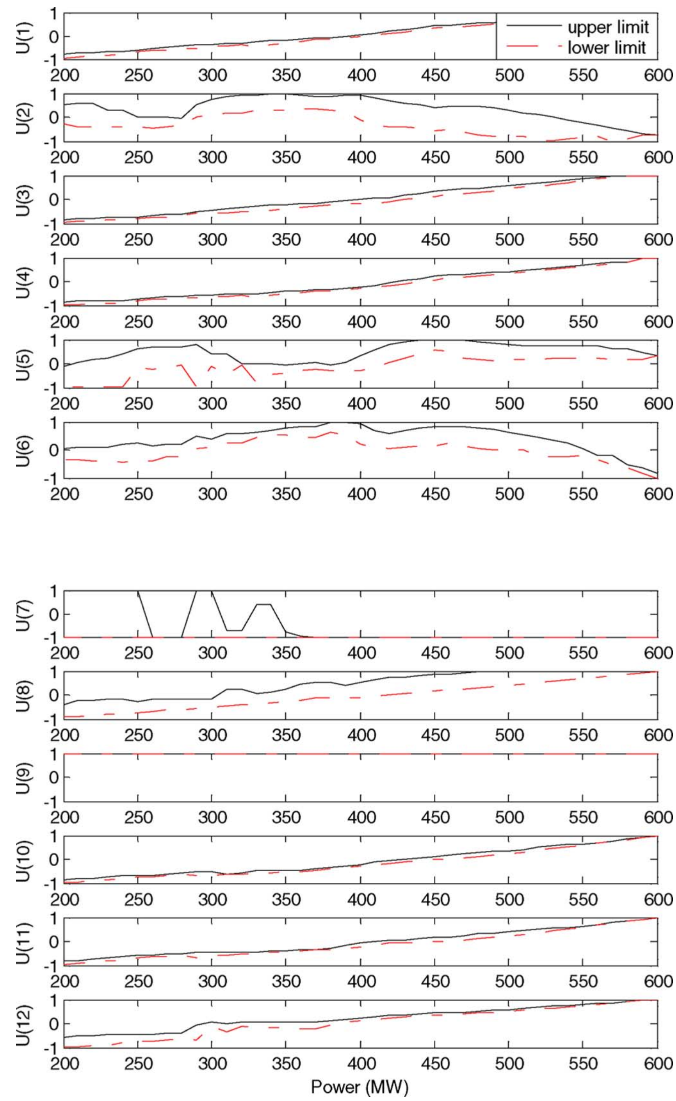


Fig. 13. Power-control input operating windows.

and expressed as follows:

$$\text{Power: } E_d = \phi_1 (u_1, u_2, \ldots, u_{12}) \tag{1a}$$

$$\text{Steam pressure: } P_d = \phi_2 (u_1, u_2, \ldots, u_{12}). \tag{1b}$$

Reheater/superheater temperatures:

$$RT_d = ST_d = \phi_3(u_1, u_2, \ldots, u_{12}). \tag{1c}$$

The MAS-ISSMs are modeled by the NN with inputs $(u_1, u_2, \ldots, u_{12})$ and outputs (E, P, and $RT = ST$) as defined in (1). During the simulation to find feasible regions, the monitoring agents collect all operating data in their database. The steady-state agent communicates with the knowledge database agent to train the NN. Whenever the knowledge database agent detects a new admissible operating point, it lets the steady-state agent know that the operating window is updated. The MAS-ISSMs are adaptively changed by learning using the updated operating windows that are adjusted to the conditions of the power plant. With a new dataset of control inputs and power/pressure/temperature outputs, the steady-state

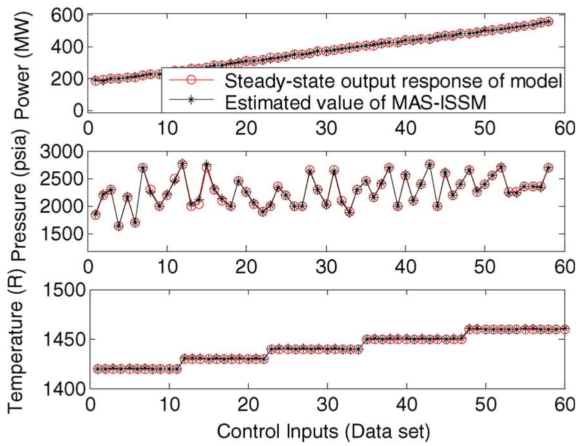


Fig. 14. Validation of MAS-ISSM for the power/pressure/temperature outputs.

agent is evaluated for validation. Fig. 14 shows the validation of MAS-ISSM for power, steam pressure, and reheater/superheater temperature outputs, respectively. Based on the examination of the results, the MAS-ISSM represents the approximator successfully, which shows very small error between the steady-state output responses of the large-scale model and the estimated values of MAS-ISSM.

3) Optimal Steady-State Control Inputs: With the operating windows and MAS-ISSMs, the multiobjective optimization problem can be tackled by the set-point generation agent and the cooperation of other agents. In this paper, the objective functions are accounting for the minimization of load-tracking error, fuel consumption, heat loss rate, pollutant emission, and extension of duty life on the equipment [2], [4]. Thus, the objective functions are formulated as

$$J_0(u) = |E_{\text{uld}} - E_d|, \quad J_1 = u_1, \quad J_2 = -u_2, \quad J_3 = u_3$$
$$J_4 = -u_4, \quad J_5 = -u_5, \quad J_6 = u_6, \quad J_7 = u_7, \quad J_8 = -u_8$$
$$J_9 = -u_9, \quad J_{10} = -u_{10}, \quad J_{11} = -u_{11}, \quad J_{12} = -u_{12} \quad (2)$$

where E_{uld} is the ULD. All values of the objective functions (2) are required to be minimized with respective preferences. Table I shows the explanation of the objective functions with respect to decision variables, which are control valve inputs $(u_1, u_2, \ldots, u_{12})$ described in Section II. The basic objective function $J_0(u)$ is to track the ULD as closely as possible. The rest of objective functions $J_1(u) - J_{12}(u)$ are simply the valve openings of respective controllers. For example, $J_1(u)$ represents the opening of the fuel valve, which reflects the fuel consumption. One way of reducing pollutant emission could be to minimize the induced draft fan speed $J_3(u)$. Since the induced draft fan is used to control the pressure inside the furnace, the optimal fan speed can be obtained by solving the multiobjective optimization problem. Some objective functions have a negative sign, which means to maximize the valve openings. For example, $J_8(u)$ is to maximize the pressure valve opening, which prevents the wear of the valve; thus, the equipment life can be extended and accidental breakdowns can be reduced.

TABLE I
EXPLANATION OF THE OBJECTIVE FUNCTIONS

Objective function	Description
$J_0(u)$	load-tracking error
$J_1(u)$	fuel consumption through the fuel flow valve
$J_2(u)$	use of flue gas through the gas recirculation valve
$J_3(u)$	pollutant emission through the induced draft fan valve
$J_4(u)$	use of air through the forced draft fan valve
$J_5(u)$	heat distribution through the burner tilt
$J_6(u)$	heat loss rate through the superheater spray valve
$J_7(u)$	heat loss rate through the reheater spray valve
$J_8(u)$	use of steam pressure through the governor valve
$J_9(u)$	use of steam pressure through the intercept valve
$J_{10}(u)$	use of condensate water through the deaerator valve
$J_{11}(u)$	use of water through the feedwater valve
$J_{12}(u)$	use of water through the feedpump turbine valve

* The minus sign means maximization. For example, u_2 needs to be maximized, in other word, $-u_2$ needs to be minimized.

When the ULD, E_{uld}, is given from a central dispatch center, the set-point generation agent creates the solution space, $\Omega_1, \Omega_2, \ldots, \Omega_{12}$, using the power-control input operating windows, as shown in Fig. 13. An operator provides the objective functions (2) and their preference values β for the multiobjective optimization problem through the interface agent. The reference governor delegation agent adjusts the preference values by investigating the condition of power plant with historical data. After confirming the preference values, the multiobjective optimization problem is ready to be solved.

In the multiobjective optimization, the objective functions are often in conflict with one another when performing the optimization. Thus, it is proposed to minimize the maximum deviation of the objective functions instead of directly minimizing the multiobjective functions [2]. The maximum deviation of the multiobjective functions is defined as follows:

$$\delta_m = \max_{i=1,\ldots,k} \delta_i, \quad \delta_i \geq 0 \tag{3a}$$

$$\delta_i = \beta_i \left| J_i(u) - J_i(u)^* \right|, \quad i = 1, 2, \ldots, k, \quad u \in \Omega \tag{3b}$$

$$J_i^* = \min\{J_i(u); u \in \Omega\}, \quad i = 1, 2, \ldots, k \tag{3c}$$

where δ_m is the maximum deviation of the multiobjective functions, δ_i is weighed deviation, β_i is the preference value, J_i^* is the minimum possible value of the single objective function J_i, and Ω is the solution space. The preference values give the relative priorities of the objectives in searching for the optimal solution. The relative priorities of the objectives are the weights of objectives that are desired by an operator. In this paper, the multiobjective optimization problem is solved by using the PSO algorithm since the ability of PSO was well shown in the small-scale power plant case [4]. With the objective functions (2) and maximum deviation function (3), the PSO technique will be performed to find the optimal input u^* in the set-point generation agent. The PSO algorithm is one of the algorithm modules in the set-point generation agent (see Fig. 5).

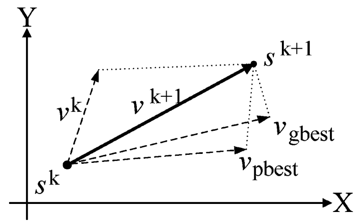


Fig. 15. Concept of modification of a search point by PSO.

4) Overview of the Basic PSO: Basically, the PSO was developed through simulation of birds flocking in 2-D space [8]. The position of each bird (called particle) is represented by a point in the X–Y coordinates and also the velocity is similarly defined. Bird flocking is assumed to optimize a certain objective function. Each particle knows its best value so far (*pbest*) and its current position. This information is an analogy of the personal experience of a particle. Moreover, each particle knows the best value so far in the group (*gbest*) among *pbests* of all particles. This information is an analogy of a particle knowing how other particles around it have performed. Each particle tries to modify its position using the concept of velocity. The velocity of each particle can be updated by the following equation:

$$v_i^{k+1} = wv_i^k + c_1 \, \mathrm{rand}_1 \times (pbest_i - s_i^k) + c_2 \, \mathrm{rand}_2 \times (gbest - s_i^k) \tag{4}$$

where v_i^k is velocity of particle i at iteration k, w is weighting function, c_1 and c_2 are weighting factors, rand_1 and rand_2 are random numbers between 0 and 1, s_i^k is current position of particle i at iteration k, $pbest_i$ is the *pbest* of particle i, and *gbest* is the best value so far in the group among the *pbests* of all particles. The first term in the right-hand side of (4) is for diversification in the search procedure, which keeps on trying to explore new areas. The second and third terms are for intensification in the search procedure. They help in moving toward the *pbests* and/or *gbest* [12]. The method has a well-balanced mechanism to utilize diversification and intensification efficiently in the search procedure. The following weighting function is usually utilized in (4):

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{5}$$

where $w_{\max}$ is the initial weight, $w_{\min}$ is the final weight, $iter_{\max}$ is the maximum iteration number, and $iter$ is the current iteration number. Using the previous equations, a certain velocity, which gradually brings the particles close to *pbest* and *gbest*, can be calculated. The current position (search point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1}. \tag{6}$$

The model using (4) is called the *Gbest model.* The model using (5) in (4) is called the inertia weights approach (IWA). Fig. 15 shows the concept of modification of a search point by the PSO. The PSO is applied by replacing the control input u_i with the particle s_i^k in the multiobjective optimization problem (3).

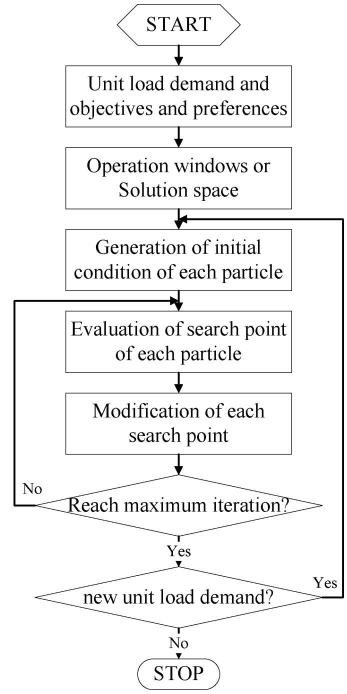


Fig. 16. Total flow chart of PSO in the MAS-IRG.

5) Implementation of PSO in MAS-IRG: The PSO algorithm is applied to solve the multiobjective optimization problem in the MAS-IRG. Fig. 16 shows the flow chart of the PSO in designing the MAS-IRG with the set-point generation agent (cf. Fig. 5). The role of the set-point generation agent is to solve the multiobjective optimization problem in the MAS-IRG. In order to solve the conflicted operating requirements in (2), the set-point generation agent utilizes the maximum deviation function (3) with the basic PSO. The PSO searches for the best input values to minimize the maximum deviation of multiobjective functions.

Initialization: The first step of the PSO for the MAS-IRG is random generation of the particles in the solution space, which is the feasible input regions, $\Omega_1, \Omega_2, \ldots, \Omega_{12}$ generated with the given ULD by using power-input operating window, as shown in Fig. 13. The particles represent the search points in the solution space, which are expressed by controls $u_1, u_2, \ldots, u_{12}$. Moreover, the initial velocities are also generated randomly within the same space. Whenever the ULD is changed, the initial particles and velocities are created in the solution space corresponding to the given ULD. To find the PSO parameter values, experiments are performed by trial and error, by testing the convergence rate with many different parameter values of c_1, c_2, $W_{\max}$, and $W_{\min}$. Fig. 17 shows the examples of evaluation for the convergence rate by changing the parameter values. In order to speed up the search for an optimal solution, c_1 and c_2 are set to 2, $w_{\max} = 0.8$, and $w_{\min} = 0.3$. The number of particles is 40 and the number of iterations is 130. If the values of parameters for PSO are not properly given, the convergence occurs too early and the PSO cannot provide the optimal solution. The initial *pbests* are equal to the current search points and *gbest* is found by comparing the *pbests* among the particles.

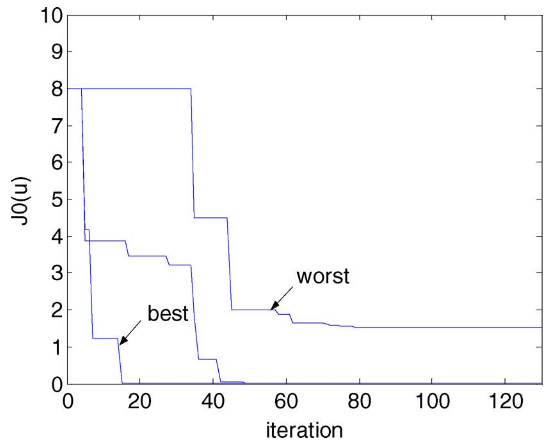


Fig. 17. Evaluation of convergence rate with different values of parameters.

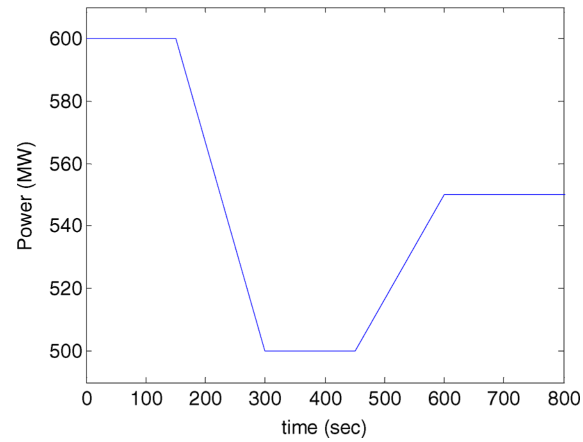


Fig. 18. ULD.

Evaluation: The evaluation of search point for each particle is performed using maximum deviation function (3) in the PSO algorithm. During the search for the solution, one of the MAS-ISSMs $E_d = \phi_1(u_1, u_2, \ldots, u_{12})$ is utilized to evaluate the load-tracking error. While searching for the optimal solution, the maximum deviation is getting smaller through the iteration of the control valve inputs $u_1, u_2, \ldots, u_{12}$.

Modification: The modification of current search point is performed by (4)–(6) in every iteration.

6) Calculation of Set Points: After finding the optimal solution $u_1^*, u_2^*, \ldots, u_{12}^*$, using the PSO, the MAS-ISSMs are applied to map the optimal solution into demand power (E_d), steam pressure (P_d), reheater temperature (RT_d), and superheater temperatures (ST_d) using (1). The set-point scheduler block (see Fig. 10) processes the task under the observation of the set-point generation agent.

V. Simulation Results

In the following, the results of the MAS-IRG will be shown and the analysis of time response will be discussed. Simulations deal with three different cases.

Case 1: Minimize $J_0(u)$ only.

Case 2: Minimize $J_0(u), J_1(u), J_2(u)$.

Case 3: Minimize $J_0(u), J_1(u), J_2(u), \ldots, J_{12}(u)$.

The objective functions are given in (2) and a vector of preference values is given by an operator. Fig. 18 shows a ULD that resembles a typical load cycle. It has different rising and falling slopes and different levels of constant powers. With the given ULD, the solution space is obtained using the power-input operating windows (see Fig. 13). Fig. 19 shows the solution space $(\Omega_1, \Omega_2, \ldots, \Omega_{12})$ for the given ULD. The gaps between upper and lower limits are the solution space for the optimization process. Next step is to perform the PSO for the multiobjective optimization with predefined objective functions and preference values. Finally, the power, pressure, and temperature set points are obtained by set-point scheduler (1) as shown in Figs. 20–22, respectively. The demand power set point (E_d) is almost the same for all cases as the ULD (see Fig. 18). The demand pressure (P_d) and temperature set points (RT_d, ST_d) mapped for different number of objective functions are shown in Figs. 21 and 22. It is interesting to note that while the demand power set-point profile is almost the same for all cases, the demand pressure set-point profiles differ significantly from case to case.

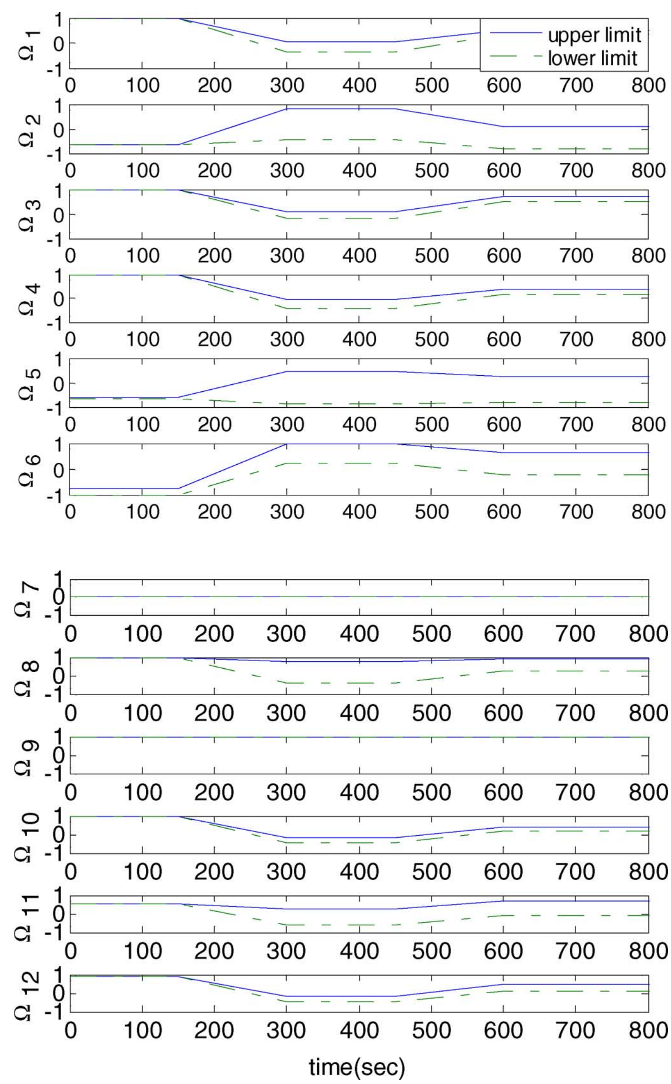


Fig. 19. Solution space $(\Omega_1, \Omega_2, \ldots, \Omega_{12})$ for the given ULD.

TABLE II
COMPARISON THE OBJECTIVE VALUES AMONG THE CASES

Objectives	$J_0(u)$	$J_1(u)$	$J_2(u)$	$J_3(u)$	$J_4(u)$	$J_5(u)$	$J_6(u)$	$J_7(u)$	$J_8(u)$	$J_9(u)$	$J_{10}(u)$	$J_{11}(u)$	$J_{12}(u)$
Case 1	8.132	3.630	-3.346	4.30	2.745	-5.002	0.841	0	6.25	7	2.308	3.029	1.965
Case 2	18.915	3.515	-4.358	4.30	2.841	-4.137	1.350	0	6.25	7	2.459	3.611	1.724
Case 3	22.313	3.287	-4.339	4.30	2.955	-4.228	1.341	0	6.25	7	2.662	3.611	1.689

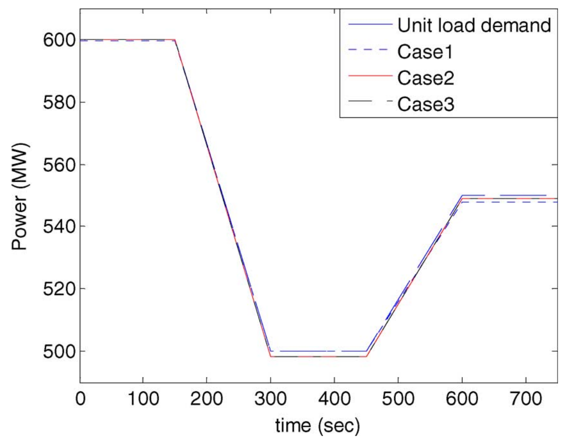


Fig. 20. Demand power set-point trajectories.

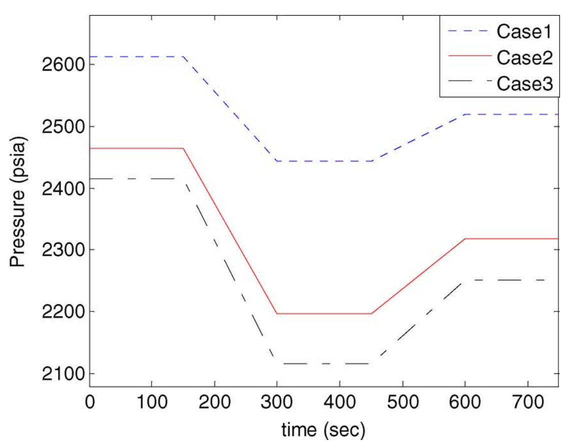


Fig. 21. Demand pressure set-point trajectories.

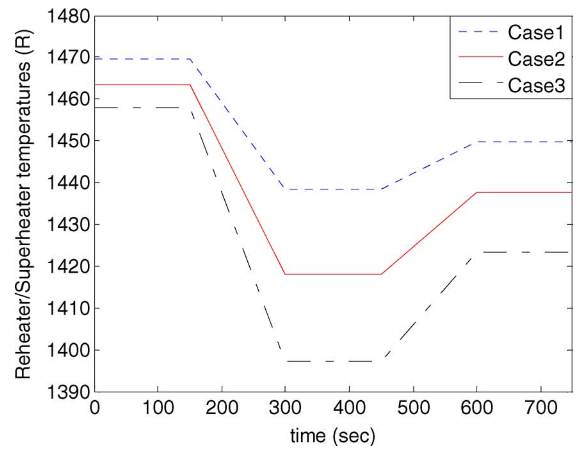


Fig. 22. Demand reheater/superheater temperature set-point trajectories.

This is because the power–pressure operating window is quite large and the same amount of power can be produced on a wide range of pressure, as shown in Fig. 12. Moreover, the different temperatures of reheater/superheater can produce the same amount of power. As additional objective functions are added in the optimization, the plant is operating more conservatively in lower pressure and temperatures. The demand powers (E_d) are almost the same as the ULD, as shown in Fig 18; however, the conflicting requirements cause slight difference between the demand power and the ULD. Thus, all simulation results show that the MAS-IRG can perform well in the multiobjective optimization problem and also in the online implementation since the pressure and temperatures set points need to be updated only when the ULD is changed during the load cycle. Moreover, distributed computing, which is the advantage of the MAS, reduces the computing time for online implementation. The goal of the MAS-IRG is to design the optimal set points when there are several conflicting objectives. The design is based on the simulation of the power plant model, which does not depend upon data; hence, the usual problem of noise is not of concern. In practical implementation, this design can be done offline and the set points can be stored in a lookup table. In case when the ULD is given in advance, which is usually the case, the set points can also be computed in advance offline for real-time application.

VI. CONCLUSION

A new concept of the IRG based on the MAS is presented for multiobjective optimal power plant operation. In order to deal with the difficulty of handling a large-scale system, architecture of a single agent and an organization of MAS are designed as basis for the IRG. The proposed MAS reduces the coupling problems of subsystems by intelligent and asynchronous computation. The optimal mappings between the varying ULD and the power, steam pressure, and reheater/superheater temperature set points are realized in an online implementation with the help of MAS. The MAS-IRG provides the optimal set points for the feedforward and feedback control loops in coordinated control system by the cooperation of MAS-ISSM. The MAS-ISSM is continuously adapted to the current condition of the power plant with the cooperation of agents. As one of algorithm modules, PSO is well suited for finding optimal solution in the multiobjective optimization problem under a diversity of operating scenarios. Table II compares the average performance of the PSO technique while increasing the number of objective functions. The case I minimizes only J0 that is indicated with a highlight box. Similarly, the other highlight boxes shows which objectives are considered for minimization in the cases II and III. Due to the confliction of requirements, the performance values of objective functions are changed while increasing the objective functions. Therefore, the PSO and MAS are efficient methodologies to design the IC system for a complex large-scale power plant.

REFERENCES

[1] H. C. Pietrobom and K. H. Kienitz, "Using surrogates to reduce time expenditure for optimization in systems and control," in *Proc. 40th IEEE Conf. Decis. Control*, Orlando, FL, 2001, vol. 4, pp. 3178–3183.

[2] R. Garduno-Ramirez and K. Y. Lee, "Multiobjective optimal power plant operation through coordinate control with pressure set point scheduling," *IEEE Trans. Energy Convers.*, vol. 16, no. 2, pp. 115–122, Jun. 2001.

[3] J. Chang, K. Y. Lee, and R. Garduno-Ramirez, "Optimal task decomposition agents of the multiagent power plant control system," presented at the 2003 Int. Conf. Intell. Syst. Appl. Power Syst., Lemnos, Greece, 2003, Paper CD ISAP03-085.

[4] J. S. Heo, K. Y. Lee, and R. Garduno-Ramirez, "Multiobjective control of power plant using particle swarm optimization techniques," *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 552–561, Jun. 2006.

[5] J. S. Heo, K. Y. Lee, and R. Garduno-Ramirez, "Multiobjective optimal power plant operation using particle swarm optimization technique," presented at the IFAC Congr., Prague, Czech Republic, 2005, Paper 04833, Tu-M06-TO/4.

[6] K. Y. Lee and M. A. El-Sharkawi, Eds., *Tutorial on Modern Heuristic Optimization Techniques With Applications to Power Systems*. Piscataway, NJ: IEEE Power Engineering Society, 2002 (IEEE Catalog Number 02TP160).

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE Int. Conf. Neural Netw.*, vol. 4, pp. 1942–1948.

[8] C. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.

[9] P. Angeline, "Using selection to improve particle swarm optimization," in *Proc. 1998 IEEE Int. Conf. Evol. Comput.*, pp. 84–89.

[10] V. Miranda and N. Fonseca, "EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems," in *Proc. 2002 IEEE Transmiss. Distrib. Conf. Exhib.*, vol. 2, pp. 6–10.

[11] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proc. 1999 IEEE Int. Conf. Evol. Comput.*, pp. 1951–1957.

[12] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. 2000 Congr. Evol. Comput.*, pp. 84–88.

[13] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with non-smooth cost functions," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 34–42, Feb. 2005.

[14] D. Angeli and E. Mosca, "Command governors for constrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 44, no. 4, pp. 816–820, Apr. 1999.

[15] E. Gilbert and I. Kolmanovsky, "Fast reference governors for systems with state and control constraints and disturbance inputs," *Int. J. Robust Nonlinear Control*, vol. 9, no. 15, pp. 1117–1141, 1999.

[16] A. Ordys, M. Johnson, R. Katebi, and M. Grimble, *Modelling and Simulation of Power Generation Plants*. London, U.K.: Springer-Verlag, 1994.

[17] P. B. Usoro, "Modeling and simulation of a drum-turbine power plant under emergency state control," M.S. thesis, Massachusetts Inst. Technol., Cambridge, 1977.

[18] S. J. Al-Nasurs, "Simulation of a fossil-fuel power plant unit in matlab environment," M.S. thesis, Dept. Electr. Eng., The Pennsylvania State Univ., University Park, 2001.

[19] J. P. McDonald, H. G. Kwatny, and J. H. Spare, "A non-linear model for reheat boiler-turbine generator systems; part I—General description and evaluation," in *Proc. 12th Joint Autom. Control Conf.*, 1971, pp. 219–226.

[20] H. G. Kwatny, J. P. McDonald, and J. H. Spare, "A non-linear model for reheat boiler-turbine generator systems; part II—Development," in *Proc. 12th Joint Autom. Control Conf.*, 1971, pp. 227–236.

[21] W. Junpu, C. Hao, X. Yang, and L. Shuhui, "An architecture of agent-based intelligent control systems," in *Proc. 3rd World Congr. Intell. Control Autom.*, 2000, pp. 404–407.

[22] T. Wittig, N. R. Jennings, and E. H. Mamdani, "ARCHON: Framework for intelligent co-operation," *IEEE Trans. Intell. Syst. Eng.*, vol. 3, no. 3, pp. 168–179, Autumn 1994.

[23] J. Kosakaya, A. Kobayashi, and K. Yamaoka, "Cooperative multi-agent-based control technology for supervisory control and data-acquisition systems," in *Proc. Emerging Technol. Factory Autom. Conf.*, 2003, pp. 19–26.

[24] J. Jung and C.-H. Liu, "Multi-agent system technologies and an application for power system vulnerability," in *Proc. IEEE Power Eng. Soc. General Meeting*, Denver, CO, 2004, vol. 1, pp. 61–64, Paper PESGM2004-000975.

[25] J. A. Hossack, J. Menal, S. D. J. McArthur, and J. R. McDonald, "A multiagent architecture for protection engineering diagnostic assistance," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 639–647, May 2003.

[26] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems*. Hoboken, NJ: Wiley, 2004.

[27] M. Wooldridge, *An Introduction To Multiagent Systems*. Chichester, U.K.: Wiley, 2002.

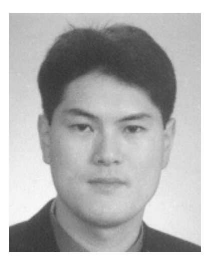
[28] J. S. Heo and K. Y. Lee, "Multi-agent system-based intelligent control system for a power plant," in *Proc. IEEE Power Eng. Soc. General Meeting*, San Francisco, CA, 2005, vol. 2, pp. 1050–1055, Paper CD PESGM2005-000858.

[29] J. S. Heo and K. Y. Lee, "Multi-agent system-based intelligent steady-state model for a power plant," in *Proc. 13th Int. Conf. Intell. Syst. Appl. Power Syst.*, Arlington, VA, 2005, pp. 419–424.

[30] J. S. Heo and K. Y. Lee, "A multi-agent system-based intelligent identification system for control and fault-diagnosis for a large-scale power plant," presented at the IEEE Power Eng. Soc. General Meeting, Montréal, QC, Canada, 2006.

[31] H. C. Gery, "The evolution of coordinated control," *Inst. Soc. Amer. Power Symp.*, St. Petersburg, FL, 1988, pp. 109–112, paper no. 88-0417.

[32] R. Fray, "Compact simulators for fossil-fueled power plants," *IEEE Spectr.*, vol. 32, no. 2, pp. 46–51, Feb. 1995.



Jin S. Heo received the B.S. and M.S. degrees in electronics engineering from Inje University, Gimhae, Korea, in 1999 and 2001, respectively, and the Ph.D. degree in electrical engineering from the Pennsylvania State University, University Park, in 2006.

His current research interests include multiobjective optimization in control system, intelligent distributed control, multiagent system, modeling and control of power plants, and real-time embedded system.

Kwang Y. Lee (S'70–M'72–SM'86–F'01–LF'08) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1964, the M.S. degree in electrical engineering from North Dakota State University, Fargo, in 1968, and the Ph.D. degree in system science from Michigan State University, East Lansing, in 1971.

He has been with Michigan State University, Oregon State University, and the University of Houston. He has also been with The Pennsylvania State University, University Park and Baylor University, where he is currently a Professor and Chair of electrical and computer engineering and the Director of Power Systems Control Laboratory. His current research interests include power system control, operation, planning, and intelligent system applications to power systems.

Dr. Lee is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the Editor of the IEEE TRANSACTIONS ON ENERGY CONVERSION. He is also a Registered Professional Engineer.