

# A general multi-agent control approach for building energy system optimization



Jie Cai<sup>a,\*</sup>, Donghun Kim<sup>a</sup>, Rita Jaramillo<sup>a</sup>, James E. Braun<sup>a</sup>, Jianghai Hu<sup>b</sup>

<sup>a</sup> Ray W. Herrick Laboratories, School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA

<sup>b</sup> School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

## ARTICLE INFO

### Article history:

Received 22 July 2015

Received in revised form 26 January 2016

Accepted 14 May 2016

Available online 18 May 2016

### Keyword:

Multi-agent control

Building energy system optimization

Distributed optimization

HVAC component coordination

## ABSTRACT

Penetration of advanced building control techniques into the market has been slow since buildings are unique and site-specific controller design is costly. In addition, for medium- to large-sized commercial buildings, HVAC system configurations can be very complex making centralized control infeasible. This paper presents a general multi-agent control methodology that can be applied to building energy system optimization in a “plug-and-play” manner. A multi-agent framework is developed to automate the controller design process and reduce the building-specific engineering efforts. To support distributed decision making, two alternative consensus-based distributed optimization algorithms are adapted and implemented within the framework. The overall multi-agent control approach was tested in simulation with two case studies: optimization of a chilled water cooling plant and optimal control of a direct-expansion (DX) air-conditioning system serving a multi-zone building. In both cases, the multi-agent controller was able to find near-optimal solutions and significant energy savings were achieved.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

More than 40% of the primary energy usage in the United States is related to energy consumption in buildings [1] and if buildings are not operated properly, a significant amount of energy is wasted. The energy savings opportunities for optimal building controls are becoming widely recognized leading to growing research efforts in the past few years. However, the deployment of advanced controls in buildings has been progressing very slowly due to several reasons: (1) buildings are unique in terms of both building construction and heating, ventilation and air-conditioning (HVAC) system configuration, which makes building-specific controller design costly; (2) optimal control of complex building energy systems is difficult because of the nonlinearities in the models as well as the large number of optimization variables.

Multi-agent control is a promising solution for building energy system management due to good modularity and performance in solving large-scale problems. Several efforts have been made to solve different building control problems using a multi-agent system. For example, some previous work focused on implementation of intelligent agents where the intelligence comes from some

heuristic rules that already existed for specific types of components [2–4]. As an example, Davidsson and Boman [3] utilized a room agent to setup or setback the room temperature setpoint depending on the presence of occupants to reduce HVAC energy consumption. A number of heuristic control strategies for different types of building energy systems can be found in Chapter 42 of the ASHRAE handbook—HVAC applications [5], and also in Refs. [6–8]. Although heuristic- or rule-based control is simple to implement and typically can be easily integrated within a device agent, general heuristic control rules do not exist for most HVAC devices and thus, the application of rule-based controls in a multi-agent controller is limited.

Some other researchers adopted a centralized-optimization-based multi-agent control approach, which mainly takes advantage of the good modularity of a multi-agent control system [9–11]. As an example, Zhao et al. [9] proposed a multi-agent control structure with an electricity agent (E-agent), a heating agent (H-agent) and a cooling agent (C-agent) where the E-agent manages the electrical power flow from electricity generator and the energy consumers are handled by the H-agent and C-agent. The decision making process therein still relies on centralized optimization and thus, this control approach may not be suitable for control of complex building energy systems.

Other work has investigated distributed decision making within a multi-agent controller to achieve good scalability. Most of the

\* Corresponding author.

E-mail address: [cai799174@gmail.com](mailto:cai799174@gmail.com) (J. Cai).

## Nomenclature

$A_{damper}$	Damper cross sectional area ( $m^2$ or $ft^2$ )
$A, C$	State-space matrices
$B_w, B_u$	State-space B matrices corresponding to disturbance and control inputs, respectively
$c_p$	Specific heat of air ( $kJ/kg\cdot K$ or $btu/lb\cdot F$ )
$ESP$	External static pressure of fan (in. W.C. or Pa)
$m_a$	Air mass flow rate across the cooling tower or DX coil ( $Kg/s$ or $lb/s$ )
$m_{a,i}$	Air mass flow rate through VAV box $i$ ( $Kg/s$ or $lb/s$ )
$m_{co}$	Condenser water mass flow rate mass flow rate ( $Kg/s$ or $lb/s$ )
$m_{cto}$	Cooling tower outlet water mass flow rate ( $Kg/s$ or $lb/s$ )
$m_{mains}$	Make-up water mass flow rate ( $Kg/s$ or $lb/s$ )
$Pow$	Power consumption (KW or $btu/hr$ )
$P_{ma}$	Cooling coil inlet air pressure (in. W.C. or Pa)
$P_{sa}$	Fan outlet air pressure (in. W.C. or Pa)
$P_{z,i}$	Air pressure in zone $i$ (in. W.C. or Pa)
$Q_{ev}$	Chiller load (KW or $btu/hr$ )
$SHR$	Sensible heat ratio of cooling coil
$Stage$	Compressor stage
$T_{amb}$	Ambient air (dry-bulb) temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{coi}$	Condenser inlet chilled water temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{coo}$	Condenser outlet chilled water temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{cto}$	Cooling tower outlet chilled water temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{db}$	Ambient air dry-bulb temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{evo}$	Evaporator outlet chilled water temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{la}$	Cooling coil outlet air temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{sa}$	Supply (fan outlet) air temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{wb}$	Ambient air wet-bulb temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{mains}$	Make-up water temperature ( $^{\circ}C$ or $^{\circ}F$ )
$T_{z,i}$	Air temperature of zone $i$ ( $^{\circ}C$ or $^{\circ}F$ )
$w_{ma}$	Cooling coil inlet air humidity ratio ( $kg\ water/kg\ air$ )
$X$	Vector of all local copies of the design variables
$x_k, y_k$	State and output vectors at time step $k$
$Y$	Dual variable vector
$y_{k+1}^{sp}$	Output setpoint vector at time step $k+1$
$Z$	Vector of the design variables
$X_i, Y_i$	Sub-vector corresponding to sub-problem ( $i$ )
$\alpha$	Step size for dual update
$\theta_i$	Damper opening in VAV box $i$ (%)
$\rho$	Air density ( $kg/m^3$ or $lb/ft^3$ )
$\sigma$	Factor for the augmented multiplier
$\varepsilon$	Convergence threshold

### Subscripts

<i>chiller</i>	Chiller
<i>ct</i>	Cooling tower
<i>DX</i>	DX unit
<i>fan</i>	Fan
<i>pump</i>	Pump

### Superscripts

( $i$ )	The $i$ th copy of the corresponding local variable
[ $i$ ]	The $i$ th iteration of optimization

these studies were primarily concerned with optimal load profile management and the HVAC system models were over simplified. This could reduce the actual energy savings since there are significant savings opportunities with optimal coordination of HVAC components (see Ref. [15] as an example).

The present study proposes a general multi-agent control approach for building energy systems that consists of two main elements: a multi-agent control framework and a multi-agent decision making procedure. Once a control project is configured within the framework for a building energy system, a centralized or distributed optimization problem is automatically composed depending on the user's specification and some symbolic manipulations are performed to eliminate the redundant design variables and equations. If the distributed control option is chosen, two different consensus-based optimization algorithms that are embedded within the framework are used to drive the intra-agent optimization and inter-agent coordination processes. The overall approach addresses the issues of low implementation cost and scalability in the following ways:

- 1 *Low implementation cost*: if the component agent models that represent device performance were integrated within HVAC devices (e.g., chillers) by manufacturers (models could adapt on the fly with continuous measurements), the proposed multi-agent control framework would automate the controller design process.
- 2 *Good scalability*: the distributed decision making process allows solution of a large-scale optimization problem in a distributed and parallel way.

The proposed multi-agent control approach was tested for two building control case studies. One case study focused on optimal control of a chilled-water cooling system and the other one concerned optimization of a direct-expansion (DX) air conditioning system serving a multi-zone building. The performance of the multi-agent control and the corresponding energy savings compared with other benchmarks were evaluated under different operating conditions.

## 2. Multi-agent control framework

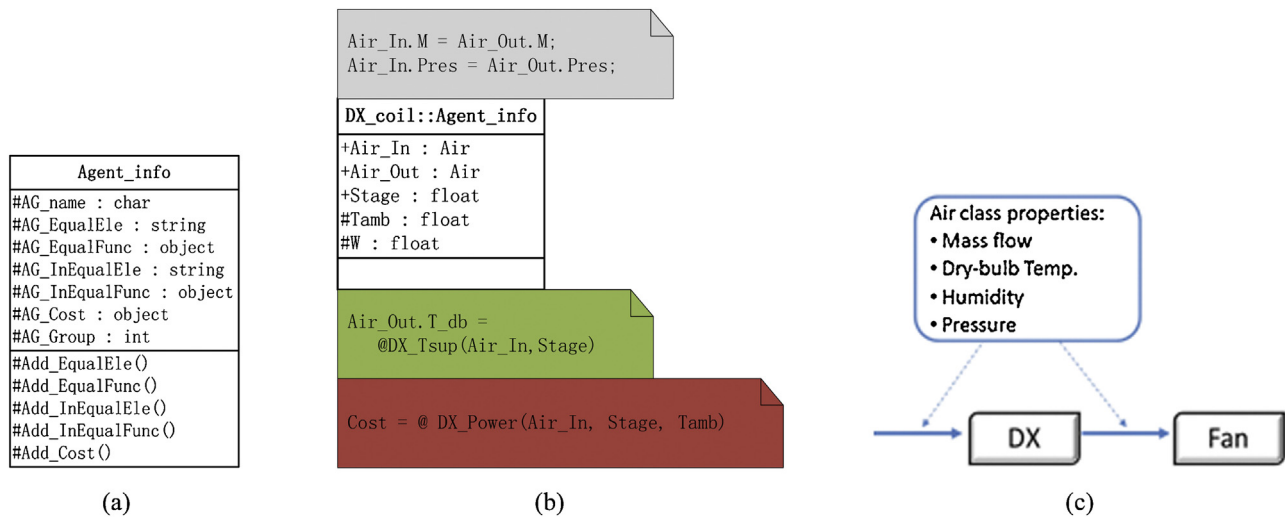
A prototype of the multi-agent control framework was developed using the Matlab object-oriented programming toolkit. It serves as a proof of concept in the software sense but can be replicated easily in other programming environments to support hardware implementation. The framework defines a general component agent structure as well as the flow connections between agents. To synthesize a multi-agent control system, a field engineer would only need to configure the inter-agent connections and the framework would compose the control algorithm automatically, assuming the component agents were integrated within the devices by their manufacturers.

### 2.1. Agent definition

Fig. 1(a) shows the structure of the backbone for a general component agent. It is written as a super class from which each component class can inherit the basic agent structure. A component agent is essentially an instantiation of the corresponding component class.

The properties of the agent class consist of a collection of cost functions, equality and inequality constraints that characterize the behavior of a specific component. Once a component agent is instantiated, the cost functions and equality/inequality constraints will be registered in the composed optimization(s). Note that if there are multiple cost functions in the same control project, the

work under this category focused on dynamic optimization problems under a distributed model predictive control (DMPC) scheme. Some examples can be found in Refs. [12–14]. However, most of



**Fig. 1.** (a) Component agent structure. (b) An example of component agents: direct-expansion (DX) unit agent. (c) An example of flow variables: air stream.

**Table 1**  
Different Types of Constraints in a Component Agent Structure.

	Constraint type	Format	Examples
EqualEle	Linear equality	String	$m_{in} = m_{out}$ for airflow through a fan
EqualFunc	Nonlinear equality	Function handle	Chilled water temperature drop across a chiller
InEqualEle	Linear inequality	String	Interval type constraints on the design variables
InEqualFunc	Nonlinear inequality	Function handle	Chiller part-load ratio should be smaller than 1

sum of the cost functions will be minimized and thus, the physical quantities represented by different cost functions should be additive. The cost functions might be actual power consumptions that need to be minimized or some other performance metrics that need to be optimized, such as indoor space comfort. Different types of constraints are included to facilitate the symbolic manipulations later on. The detailed correspondence is listed in Table 1. Another important property in this agent structure is the group number for a specific agent, which is denoted by 'AG.Group'. This parameter is used in the setup of a distributed-optimization-based controller: one local controller will be assigned to control the components with the same group number and different local controllers will cooperate to find an overall optimal solution. This feature provides good flexibility in the design of a multi-agent controller topology. For example, one air handler unit (AHU) controller could be installed to control all the components in an AHU, say, direct-expansion (DX) cooling coil and supply fan, while another controller could be setup to manage the VAV box and zone comfort together. The grouping in this example depends on the physical distances among the devices but other factors could also be considered, such as network structure. Note that when all the devices are assigned with the same group number, a centralized controller will be synthesized. In addition to grouping multiple devices, the user is also allowed to group different function elements within a single device by using the group property 'Func.Group' within the cost functions or equality/inequality constraints (not shown in Fig. 1). If an individual device is very complex or if the functions within the device are computationally demanding, multiple local controllers could be specified where different controllers handle different elements within that single device.

Fig. 1(b) shows an example of the specification of a DX unit agent. The gray box on top specifies two element-wise linear equalities for the DX unit dictating that the entering air flow rate and pressure equal the leaving air flow rate and pressure. The white box contains the main properties for this agent, such as the compressor stage number, entering and leaving air properties and ambient

temperature. The green box corresponds to a nonlinear equality constraint, which correlates the leaving air temperature with entering air conditions and other design variables (e.g., Stage). The red box on the bottom specifies the compressor power consumption as a cost function for this device agent.

## 2.2. Inter-agent connections

To simplify the process of connecting different agents, a flow variable is defined to lump multiple properties of a fluid flow into one single variable (a similar idea is used in the Modelica programming language; see Ref. [16]). The flow variable represents a physical fluid flow through different components, which makes the project setup process more physically-based. This is critical for controller design of complex systems since the multi-agent topology is closely mapped from the actual system layout and it makes the multi-agent controller setup process more straightforward.

Fluids that are commonly used in HVAC systems include air, chilled and hot water and refrigerant. In the following case studies, separate classes are defined for air and chilled water that include all the necessary properties. As an example, Fig. 1(c) illustrates the usage of a flow variable to specify the air flow connection from the DX unit agent to the fan agent (from case study 2). The flow variable for air is defined as a class that has four properties: mass flow rate, dry-bulb temperature, humidity ratio and static pressure. The connection between the two agents corresponds to an air stream from a DX unit to a fan and this connection will add four equality constraints that correlate the corresponding properties between the DX outlet and fan inlet. Instead of linking the corresponding properties one by one, the flow variable only requires one simple link to enforce equalities among multiple properties. This reduces the engineering effort for inter-agent connection configurations significantly.

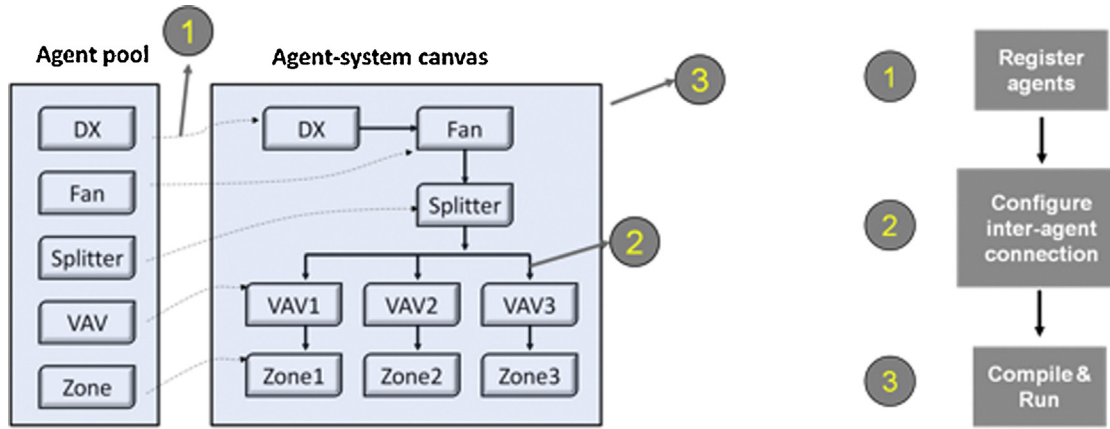


Fig. 2. Left procedure to setup a multi-agent system (from case study 2); right: program flow chart.

### 2.3. Program flow

The graph on the left hand side of Fig. 2 demonstrates the procedure to create a multi-agent system. Assuming all the component agents are already at hand, one can simply drag and drop them in a project canvas. This first step would register the agents and instantiate the component classes in the project. Once all agents are registered, inter-agent connections need to be specified in the second step by linking the associated agents in the project canvas. These two steps complete the multi-agent system setup and the framework would compile the code and compose an optimization problem automatically, as will be described in the following section.

### 3. Optimization problem composition

Once all the agents and their inter-connections are specified, the framework automatically constructs an optimization problem according to the configuration. This compilation process consists of several symbolic manipulation steps:

- 1 Allocate all the design variables.
- 2 Extract all the cost functions to construct a total cost function.
- 3 Eliminate the element-wise equality constraints, i.e., the constraints with the form of  $x_1 = x_2$ , and combine the two variables into one single variable.
- 4 Identify and eliminate redundant linear equality constraints by checking the linear dependence of the coefficient vectors: for two linear equality constraints  $\mathbf{E}_1 \mathbf{x}_1 = a_1$  and  $\mathbf{E}_2 \mathbf{x}_1 = a_2$  where  $\mathbf{E}_1$  and  $\mathbf{E}_2$  are two row vectors, there is a redundancy if  $\text{rank}([\mathbf{E}_1 | \mathbf{E}_2]) = 1$  and  $a_2 \mathbf{E}_1 = a_1 \mathbf{E}_2$ .
- 5 Identify and eliminate redundant weaker linear inequality constraints by checking the linear dependence of the coefficient vectors: for two linear inequality constraints  $\mathbf{E}_1 \mathbf{x}_1 < a_1$  and  $\mathbf{E}_2 \mathbf{x}_1 < a_2$ , there is a redundancy if  $\text{rank}([\mathbf{E}_1 | \mathbf{E}_2]) = 1$  and the corresponding elements of  $\mathbf{E}_1$  and  $\mathbf{E}_2$  have the same sign. The weaker constraint is the one with the larger value on the right hand side of the inequality after the left hand sides of both constraints are scaled by a positive quantity to have the same coefficient vector.
- 6 Eliminate the variables that are specified as boundary conditions and substitute their values in the associated function handles.

The steps above are mostly dedicated to reducing the dimension of the optimization problem as much as possible to lower the computational burdens. The steps need to be carried out with respect to

each sub-problem or group and the composed optimization problem is in the form:

$$\begin{aligned}
 & \min_{\mathbf{x}_1} f_1(\mathbf{x}_1) & \min_{\mathbf{x}_n} f_n(\mathbf{x}_n) \\
 & \text{s.t.} & \text{s.t.} \\
 & \mathcal{E}_1 \mathbf{x}_1 = B_1 & \mathcal{E}_n \mathbf{x}_n = B_n \\
 & \mathcal{F}_1 \mathbf{x}_1 < C_1 & \cdots \mathcal{F}_n \mathbf{x}_n < C_n \\
 & \mathcal{H}_1(\mathbf{x}_1) = 0 & \mathcal{H}_n(\mathbf{x}_n) = 0 \\
 & \mathcal{G}_1(\mathbf{x}_1) < 0 & \mathcal{G}_n(\mathbf{x}_n) < 0
 \end{aligned} \tag{1}$$

In addition to the problems formulated above, there are extra consensus constraints to enforce local copies of the same variable to match between different agents. Take the system in Fig. 1(c) as an example, the air leaving the DX coil is the same as air entering the fan (assuming negligible duct losses) and if these two devices are assigned different group numbers, there will be two sets of local variables corresponding to the same air property. The DX coil optimization would be in favor of higher leaving air temperature to save compressor power while in the supply fan problem lower air temperature would be beneficial since lower airflow, and thus lower fan power, is required to provide the same cooling capacity. So the two optimizations would drive the two local variables that represent the same physical quantity in opposite directions and the consensus constraints are necessary to enforce equalities among these local variables. The consensus constraints have the form:

$$\mathbf{X} = \mathbf{E}\mathbf{X} = \mathbf{F}\mathbf{Z} \tag{2}$$

where  $\mathbf{X}$  is a stack of all the local variables, i.e.,  $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$ ,  $\mathbf{E}$  is an identity matrix,  $\mathbf{Z}$  is a vector that contains the global variables and  $\mathbf{F}$  is a matrix such that the element in the  $i$ -th row and  $j$ -th column is equal to 1 if the  $i$ -th variable in  $\mathbf{X}$  is a local copy of the  $j$ -th variable in  $\mathbf{Z}$  and is equal to 0 otherwise. If all the components were assigned with the same group number, a centralized optimization problem would be composed.

### 4. Distributed optimization algorithms

Denote by  $\mathbf{E}_i$  and  $\mathbf{F}_i$  the sub-matrices of  $\mathbf{E}$  and  $\mathbf{F}$ , respectively, that contain only the rows corresponding to the constraints that belong to sub-problem  $i$ . Then  $\mathbf{E}_i \mathbf{x}_i = \mathbf{F}_i \mathbf{Z}$  and each sub-problem can be reformulated as

$$\begin{aligned}
 & \min_{\mathbf{x}_i} f_i(\mathbf{x}_i) \\
 & \text{s.t. } \mathbf{x}_i \in \mathbf{C}_i
 \end{aligned}$$



where  $\mathbf{C}_i$  is the feasible region of the local variables  $\mathbf{X}_i$ . Two consensus-based distributed optimization algorithms, subgradient method [17,18] and alternating direction multiplier method (ADMM) [18,19], have been adapted to address the optimization problem formulated in Eqs. (1) and (2). It is important to emphasize that the optimization problem in a building energy system might not be convex, as the case in the 2nd case study, but both of the algorithms only guarantee convergence for convex problems with some additional requirements. These algorithms are used here as local optimizers to provide a coordination mechanism for the multi-agent system; the issue of global convergence will not be addressed in this study. To increase the chances of getting a global optimum, a multi-start search scheme is used where multiple rounds of optimizations are performed with different initial guesses and the point with the minimum cost is used as the final solution.

#### 4.1. Subgradient method

The Lagrangian for the distributed optimization problem is

$$\mathbf{L}(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) = \sum_{i=1}^n f_i(\mathbf{X}_i) + \mathbf{Y}^T (\mathbf{E}\mathbf{X} - \mathbf{F}\mathbf{Z}), \quad (3)$$

where  $\mathbf{Y}$  is the Lagrange multiplier vector. Let  $\mathbf{Y}_i$  be the sub-vector that corresponds to the sub-problem  $i$ . Then Eq. (3) becomes

$$\mathbf{L}(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) = \sum_{i=1}^n (f_i(\mathbf{X}_i) + \mathbf{Y}_i^T \mathbf{X}_i) - \mathbf{Y}^T \mathbf{F}\mathbf{Z}. \quad (4)$$

Assume the infimums are all obtainable within the feasible region for all problems considered and define

$$g_i(\mathbf{Y}_i) = \min_{\mathbf{X}_i} (f_i(\mathbf{X}_i) + \mathbf{Y}_i^T \mathbf{X}_i). \quad (5)$$

Then the dual function is

$$g(\mathbf{Y}) = \min_{\mathbf{X}, \mathbf{Z}} \mathbf{L}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \sum_{i=1}^n g_i(\mathbf{Y}_i) + \min_{\mathbf{Z}} (-\mathbf{Y}^T \mathbf{F}\mathbf{Z}) \quad (6)$$

For the equation above to be valid, we must have  $\mathbf{Y}^T \mathbf{F} = \mathbf{0}$ , or otherwise the second term in Eq. (6) would be unbounded below. Thus, the dual problem is

$$\begin{aligned} \max_{\mathbf{Y}} g(\mathbf{Y}) &= \sum_{i=1}^n g_i(\mathbf{Y}_i) \\ \text{s.t. } \mathbf{Y}^T \mathbf{F} &= \mathbf{0}. \end{aligned} \quad (7)$$

The slave problems in Eq. (5) and the master/dual problem in Eq. (7) can be solved iteratively. Note that the slave problems are totally separable and they have a one-to-one correspondence to the sub-problems in Eq. (1). So each sub-problem can be solved independently by an individual agent and the master problem can be tackled by some coordinator agent which collects and broadcasts information from and to the individual agents. The subgradient method discussed below is used to find the ascent direction to the master problem in Eq. (7).

**Definition.**  $s_g(\mathbf{Y}) \in \mathbb{R}^{n \times 1}$  is a subgradient of a convex function  $g : \mathbb{R}^n \rightarrow (-\infty, \infty)$  at  $\mathbf{Y} \in \text{dom}(g)$  if

$$g(\mathbf{Y}) + s_g(\mathbf{Y})^T (\bar{\mathbf{Y}} - \mathbf{Y}) \leq g(\bar{\mathbf{Y}})$$

for all  $\bar{\mathbf{Y}} \in \text{dom}(g)$ .

**Theorem.** Let  $\mathbf{X}_i^*$  be the optimal point in Eq. (5) corresponding to  $\mathbf{Y}_i$ . Then functions  $g_i$ 's are concave and  $-\mathbf{X}_i^*$  is a subgradient of  $-g_i$  at  $\mathbf{Y}_i$ .

**Proof.**  $g_i$  is the pointwise infimum of a family of affine functions of  $\mathbf{Y}_i$  so it is concave [20]. Let  $\bar{\mathbf{Y}}_i$  be any point in  $\text{dom}(g)$ . Then

$$\begin{aligned} g(\mathbf{Y}_i) + \mathbf{X}_i^{*T} (\bar{\mathbf{Y}}_i - \mathbf{Y}_i) &= f_i(\mathbf{X}_i^*) + \mathbf{Y}_i^T \mathbf{X}_i^* + \mathbf{X}_i^{*T} (\bar{\mathbf{Y}}_i - \mathbf{Y}_i) \\ &= f_i(\mathbf{X}_i^*) + \bar{\mathbf{Y}}_i^T \mathbf{X}_i^* \\ &\geq \min_{\mathbf{X}_i} \{f_i(\mathbf{X}_i) + \bar{\mathbf{Y}}_i^T \mathbf{X}_i\} = g(\bar{\mathbf{Y}}_i). \end{aligned}$$

Multiplying both sides of the inequality by  $-1$  completes the proof. ■

For non-differentiable functions, the subgradient plays the same role as a gradient does for differentiable functions. The subgradient does not need to be unique. Only for differentiable functions, the subgradient coincides with the gradient which becomes unique. Let subscript  $k$  denote the iteration number. Assuming  $\mathbf{X}_{k+1}^*$  contains all the optimal solutions in Eq. (5) at  $\mathbf{Y}_k$ , the ascent direction for the dual problem in Eq. (7) is simply the projection of  $\mathbf{X}_{k+1}$  onto the hyperplane defined by  $\mathbf{Y}^T \mathbf{F} = \mathbf{0}$ , which is  $(1 - \mathbf{F}(\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T) \mathbf{X}_{k+1}$ . From Eq. (2), an estimate of  $\mathbf{Z}$  would be

$$\mathbf{Z}_{k+1} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{X}_{k+1}.$$

Due to the special structure of  $\mathbf{F}$ ,  $\mathbf{Z}_{k+1}$  is essentially the average of the local copies of different variables from  $\mathbf{X}_{k+1}$ . So the ascent direction becomes  $\mathbf{X}_{k+1} - \mathbf{F}\mathbf{Z}_{k+1}$ , which is exactly the violation of consensus constraints in Eq. (2). The dual update is

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \alpha(\mathbf{X}_{k+1} - \mathbf{F}\mathbf{Z}_{k+1}),$$

which increases the penalty for larger element mismatches and aims at reducing the mismatches during the next iteration.  $\alpha$  is the step size that can be adjusted to change the penalty level.

#### 4.2. Alternating direction multiplier method (ADMM)

An augmented Lagrangian shown in Eq. (8)

$$\begin{aligned} \mathbf{L}(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) &= \sum_{i=1}^n f_i(\mathbf{X}_i) + \mathbf{Y}^T (\mathbf{X} - \mathbf{F}\mathbf{Z}) + (\sigma/2) \|\mathbf{X} - \mathbf{F}\mathbf{Z}\|_2^2 \\ &= \sum_{i=1}^n f_i(\mathbf{X}_i) + \frac{\sigma}{2} \|\mathbf{X} - \mathbf{F}\mathbf{Z} + \frac{\mathbf{Y}}{\sigma}\|_2^2 - \frac{1}{2\sigma} \|\mathbf{Y}\|_2^2 \end{aligned} \quad (8)$$

is considered in the alternating direction multiplier method (ADMM). It has an additional quadratic penalty to the consensus constraint violations compared with the Lagrangian in the preceding section. Again, let  $\mathbf{Y}_i$  denote the sub-vector of  $\mathbf{Y}$  corresponding to the sub-problem  $i$ . Then the augmented Lagrangian can be reformulated as

$$\mathbf{L}(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) = \sum_{i=1}^n \left\{ f_i(\mathbf{X}_i) + \frac{\sigma}{2} \|\mathbf{X}_i - \mathbf{F}_i \mathbf{Z} + \frac{\mathbf{Y}_i}{\sigma}\|_2^2 \right\} - \frac{1}{2\sigma} \|\mathbf{Y}\|_2^2. \quad (9)$$

It can be noticed that, unlike in Eq. (4), the slave problem in Eq. (9) is not decomposable between  $\mathbf{X}_i$  and  $\mathbf{Z}$  due to the existence of the quadratic term. An alternating direction procedure is taken which first solves the  $\mathbf{X}_i$  problem while fixing  $\mathbf{Z}$ , and then solves for  $\mathbf{Z}$  with fixed  $\mathbf{X}$ . It is trivial from Eq. (8) to see that the optimal value  $\mathbf{Z}^*$  satisfies

$$\mathbf{X} - \mathbf{F}\mathbf{Z}^* + \frac{\mathbf{Y}}{\sigma} = \mathbf{0},$$

which gives an estimate  $\mathbf{Z}^* = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T (\mathbf{X} + \frac{\mathbf{Y}}{\sigma})$ .

Similar to the subgradient method, the dual ascent direction is still  $\mathbf{X} - \mathbf{F}\mathbf{Z}$ . However, ADMM uses the penalty multiplier  $\sigma$  as the step size for the dual update.

#### 4.3. Convergence and stopping criterion

Two criteria are used to determine if convergence is reached and if the iterative optimization process can be terminated, which are defined as:

$$\varepsilon_k^1 = \|\mathbf{X}_k - \mathbf{FZ}_k\|_2^2 \quad (10)$$

$$\varepsilon_k^2 = \|\sigma \mathbf{F}(\mathbf{Z}_k - \mathbf{Z}_{k-1})\|_2^2 \quad (11)$$

As explained in Ref. [19], the optimal solution of the distributed optimization problem in Eqs. (1) and (2) need to satisfy primal

algorithms were tested for the case studies, the subgradient method failed frequently if the step size  $\alpha$  was not small enough. That was because for poorly chosen  $\alpha$ , the cost function in Step 5 of Algorithm 1 was unbounded from below and optimization drove the variable to infinity. Algorithm 2 does not have this issue since the cost function in Step 5 incorporates a quadratic penalty which makes sure the variable does not deviate too far from the center point. Although convergence of the algorithms is not considered in this paper, ADMM requires much weaker assumptions to guarantee convergence than the subgradient method. So ADMM is a preferred method and the results shown in the following two case studies were both obtained with ADMM.

#### Algorithm 1. Subgradient

---

```

1   Initialize  $\mathbf{Z}, \mathbf{X} = \mathbf{FZ}, \mathbf{Y} = \mathbf{0}, k = 1$ 
2   Loop
3       {%Solve each sub-problem%}
4       For  $i = 1$  to 11 do
5            $\mathbf{X}_i^{[k+1]} = \operatorname{argmin}_{\mathbf{X}_i} \left[ f_i(\mathbf{X}_i) + (\mathbf{Y}_i^{[k]})^T \mathbf{X}_i \right]$ 
6       End for
7       {%Estimate of the global variable%}
8        $\mathbf{Z}^{[k+1]} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{X}^{[k+1]}$ 
9       {%Dual update%}
10       $\mathbf{Y}^{[k+1]} = \mathbf{Y}^{[k]} + \alpha(\mathbf{X}^{[k+1]} - \mathbf{FZ}^{[k+1]})$ 
11      If (termination criterion met)
12          Break
13      Else
14           $k = k + 1$ 
15      Endif
16  Endloop

```

---

#### Algorithm 2. ADMM

---

```

1   Initialize  $\mathbf{Z}, \mathbf{X} = \mathbf{FZ}, \mathbf{Y} = \mathbf{0}, k = 1$ 
2   Loop
3       {%Solve each sub-problem%}
4       For  $i = 1$  to 11 do
5            $\mathbf{X}_i^{[k+1]} = \operatorname{argmin}_{\mathbf{X}_i} \left\{ f_i(\mathbf{X}_i) + \frac{\sigma}{2} \left\| \mathbf{X}_i - \mathbf{F}_i \mathbf{Z}^{[k]} + \frac{\mathbf{Y}_i^{[k]}}{\sigma} \right\|_2^2 \right\}$ 
6       End for
7       {%Estimate of the global variable%}
8        $\mathbf{Z}^{[k+1]} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \left( \mathbf{X}^{[k+1]} + \frac{\mathbf{Y}^{[k]}}{\sigma} \right)$ 
9       {%Dual update%}
10       $\mathbf{Y}^{[k+1]} = \mathbf{Y}^{[k]} + \sigma(\mathbf{X}^{[k+1]} - \mathbf{FZ}^{[k+1]})$ 
11      If (termination criterion met)
12          Break
13      Else
14           $k = k + 1$ 
15      Endif
16  Endloop

```

---

and dual feasibilities. The first criterion defined in Eq. (10) is the Euclidean norm of the primal residual, which corresponds to violations of the consensus constraints shown in Eq. (2). The second criterion defined in Eq. (11) is the Euclidean norm of the dual residual that represents the difference in the global variable between current and previous steps up to a scaling factor. So users specify a stopping threshold  $\varepsilon$  such that when both of the criteria  $\varepsilon^1$  and  $\varepsilon^2$  are below the threshold, the iterative process stops and the final iteration point is used as the optimal solution.

#### 4.4. Comparison of subgradient method and ADMM

The key steps of the subgradient and ADMM algorithms are summarized in Algorithm 1 and Algorithm 2, respectively. The subgradient method is relatively easy to implement and the computational burden is slightly lower. In addition, the slave problems are totally decomposable with the subgradient method while for ADMM they are not and an alternating direction procedure is needed. However, ADMM has better robustness. When the two

### 5. Case study 1

The first case study concerns optimal control of a cooling plant (see Fig. 3) consisting of a centrifugal chiller having 7020 kW nominal cooling capacity, an evaporative counter-flow cooling tower with a variable-speed fan, a variable speed condenser water pump with 379 l/s nominal flow and a single speed chilled water pump with a flow rate of 202 l/s. The chiller is cooled by the condenser water, which in turn rejects heat to the air as it circulates through the evaporative cooling tower. The cold water is stored in a reservoir (cold well) where it is mixed with make-up water and then is circulated back to the condenser using the variable-speed pump. Cooling tower airflow  $m_a$  and condenser water flow  $m_{co}$  are the two control variables in a real control implementation. Increased cooling tower airflow, or equivalently, lower inlet condenser water temperature, and increased condenser water flow both lead to enhanced chiller efficiency and thus, reduce the chiller power consumption for fixed load. However, both of these actions require more power from the cooling tower fan and condenser water pump. So there is a tradeoff between the fan and pump power and the

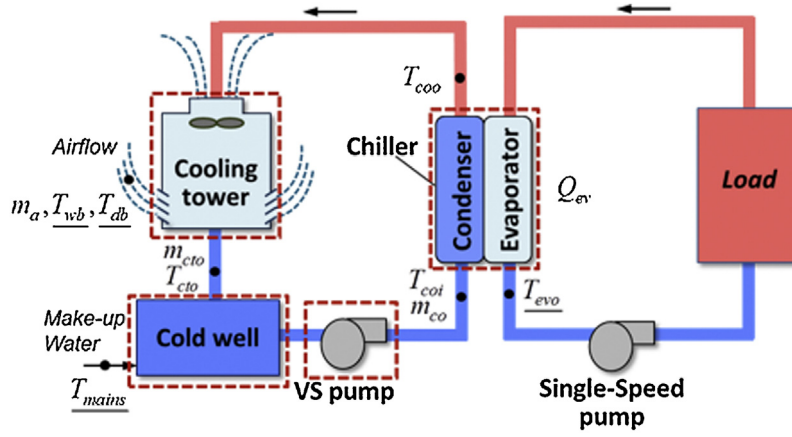


Fig. 3. Case study 1: water-cooled chiller plant control; each dashed box contains an individual component/agent.

chiller power and the optimal control solution will be a balancing point among these three quantities.

## 5.1. Agent models

### 5.1.1. Chiller

The centrifugal chiller is the main energy consumer within the cooling plant. In the chiller model, the power consumption is calculated by a bi-quadratic correlation to the load and the temperature lift (the difference between the condenser and the evaporator leaving water temperatures) where the correlation coefficients were obtained by linear regression applied to performance data. Then an energy balance is performed to correlate the condenser leaving water temperature to the inlet water conditions and building load. This balance is expressed as a nonlinear constraint in Eq. (12). Details can be found in Ref. [21].

$$\text{Cost : } Pow_{chiller} = Chiller_{pow}(Q_{ev}, T_{evo}, T_{coo}) \quad (12)$$

$$\text{NonlinConst : } T_{coo} = Chiller_{Tcoo}(Q_{ev}, T_{evo}, m_{co}, T_{coi})$$

### 5.1.2. Cooling tower

An effectiveness-based method is used to calculate the cooling tower heat transfer rate where the effectiveness is obtained using a similar relationship as for sensible heat exchangers but with modified parameter definitions, according to [22]. Then the cooling tower outlet water conditions are obtained based on mass and energy balances included in the nonlinear constraints in Eq. (13). The model parameters were trained with data from the cooling tower performance curves. Finally, the power consumption of the tower variable-speed fan is modeled as a cubic polynomial of the delivered airflow rate.

$$\begin{aligned} \text{Cost : } Pow_{ct} &= \alpha_0 + \alpha_1 m_a + \alpha_2 m_a^2 + \alpha_3 m_a^3 := CoolTow_{pow}(m_a) \\ \text{NonlinConst : } &\begin{cases} T_{cto} = CoolTow_{Tcto}(T_{wb}, T_{db}, m_{co}, m_a, T_{coo}) \\ m_{cto} = CoolTow_{mcto}(T_{wb}, T_{db}, m_{co}, m_a, T_{coo}) \end{cases} \end{aligned} \quad (13)$$

### 5.1.3. Pump

A variable-speed pump is used to deliver a variable condensing water flow rate. Neglecting the temperature rise in the delivered water, the pump agent only incorporates a cost function for the power consumption. The power is modeled as a cubic polynomial of the flow rate whose coefficients were obtained from regression of performance data at the rotor nominal speed. Pump laws are used to obtain the power consumption for other rotor speeds.

$$\text{Cost : } Pow_{pump} = \beta_0 + \beta_1 m_{co} + \beta_2 m_{co}^2 + \beta_3 m_{co}^3 := Pump_{pow}(m_{co})$$

### 5.1.4. Cold well

In the cooling tower, a portion of the condensing water evaporates and is taken away by the airflow. So the cooling tower outlet water flow rate is lower than the inlet and some make-up water is added to the cold well to compensate for this water loss. Neglecting the power associated with pumping the make-up water, the cold well agent incorporates a couple of simple constraints that come from mass and energy balances.

$$\text{LinConst : } m_{co} = m_{cto} + m_{mains}$$

$$\text{NonlinConst : } T_{coi} = \frac{m_{mains} T_{mains} + m_{cto} T_{cto}}{m_{mains} + m_{cto}} := ColdWell(m_{cto}, T_{cto}, T_{mains}, m_{mains})$$

## 5.2. Optimization problem composition

After all the components are registered in the framework, the following variables are allocated:

$$\begin{aligned} &Q_{ev}^{chiller}, T_{evo}^{chiller}, m_{co}^{chiller}, T_{coi}^{chiller}, T_{coo}^{chiller}, \\ &T_{wb}^{CTow}, T_{db}^{CTow}, m_{co}^{CTow}, m_a^{CTow}, T_{coo}^{CTow}, T_{cto}^{CTow}, m_{cto}^{CTow}, \\ &m_{co}^{pump}, \\ &m_{cto}^{CIWell}, T_{cto}^{CIWell}, T_{mains}^{CIWell}, m_{mains}^{CIWell}, m_{co}^{CIWell}, T_{coi}^{CIWell} \end{aligned}$$

where the superscript indicates which component the variable belongs to and the underscored variables are boundary conditions that are either measured directly via corresponding sensors or calculated based on available measurements. The inter-agent connections would add the following equality constraints:

$$\begin{aligned} m_{co}^{chiller} &= m_{co}^{CTow} = m_{co}^{pump} = m_{co}^{CIWell}; & T_{coo}^{chiller} &= T_{coo}^{CTow} \\ m_{cto}^{CTow} &= m_{cto}^{CIWell}; & T_{cto}^{CTow} &= T_{cto}^{CIWell}; & T_{coi}^{chiller} &= T_{coi}^{CIWell} \end{aligned} \quad (14)$$

## 5.3. Centralized formulation

If all the components are assigned with the same group number, i.e., the 'AG\_Group' property has the same value among all the

basic agents, a centralized optimization problem is formulated as below

$$\begin{aligned}
 & \min_{[m_a, m_{co}, T_{coi}, T_{coo}, m_{cto}, T_{cto}, m_{mains}]} \{Pow_{chiller} + Pow_{ct} + Pow_{pump}\} \\
 & \text{s.t.} \\
 & T_{coo} = Chiller_{T_{coo}}(Q_{ev}, T_{evo}, m_{co}, T_{coi}) \\
 & T_{cto} = CoolTow_{T_{cto}}(T_{wb}, T_{db}, m_{co}, m_a, T_{coo}) \\
 & m_{cto} = CoolTow_{m_{cto}}(T_{wb}, T_{db}, m_{co}, m_a, T_{coo}) \\
 & T_{coi} = ColdWell(m_{cto}, T_{cto}, T_{mains}, m_{mains}) \\
 & m_{co} = m_{cto} + m_{mains} \\
 & [m_a, m_{co}, T_{coi}, T_{coo}, m_{cto}, T_{cto}]^T \in \mathbf{FR}
 \end{aligned} \quad (15)$$

where  $\mathbf{FR}$  is a hyper-rectangular feasible region that represents interval type constraints for the design variables. Note that the originally composed problem had 19 design variables, 7 equality constraints from inter-agent connections as shown in Eq. (14) and 5 boundary conditions. The symbolic manipulation steps eliminated the redundant variables and constraints, resulting in a final design variable vector of dimension 7.

#### 5.4. Distributed formulation

If the four components are assigned with different group numbers, a distributed formulation is constructed by the framework as follows:

Sub-problem 1:

$$\begin{aligned}
 & \min_{[m_a^{(1)}, m_{cto}^{(1)}, T_{cto}^{(1)}, m_{co}^{(1)}, T_{coo}^{(1)}]} \{Pow_{ct}\} \\
 & \text{subject to} \\
 & T_{cto}^{(1)} = CoolTow_{T_{cto}}(m_{co}^{(1)}, m_a^{(1)}, T_{coo}^{(1)}) \\
 & m_{cto}^{(1)} = CoolTow_{m_{cto}}(m_{co}^{(1)}, m_a^{(1)}, T_{coo}^{(1)})
 \end{aligned}$$

Sub-problem 2:

$$\begin{aligned}
 & \min_{[m_{co}^{(2)}]} \{Pow_{pump}\}
 \end{aligned}$$

Sub-problem 3:

$$\begin{aligned}
 & \min_{[T_{coi}^{(1)}, m_{co}^{(3)}, T_{coo}^{(2)}]} \{Pow_{chiller}\} \\
 & \text{subject to} \\
 & T_{coo}^{(2)} = Chiller_{T_{coo}}(m_{co}^{(3)}, T_{coi}^{(1)})
 \end{aligned}$$

Sub-problem 4:

$$\begin{aligned}
 & \min_{[m_{cto}^{(2)}, T_{cto}^{(2)}, m_{co}^{(4)}, T_{coi}^{(2)}, m_{mains}^{(1)}]} \{0\} \\
 & \text{subject to} \\
 & m_{co}^{(4)} = m_{cto}^{(2)} + m_{mains}^{(1)} \\
 & T_{coi}^{(2)} = ColdWell(m_{cto}^{(2)}, T_{cto}^{(2)}, m_{mains}^{(1)})
 \end{aligned}$$

The consensus constraints are

$$\begin{aligned}
 m_a^{(1)} &= m_a \\
 m_{mains}^{(1)} &= m_{mains} \\
 m_{cto}^{(1)} &= m_{cto}^{(2)} = m_{cto} \\
 T_{cto}^{(1)} &= T_{cto}^{(2)} = T_{cto} \\
 m_{co}^{(1)} &= m_{co}^{(2)} = m_{co}^{(3)} = m_{co}^{(4)} = m_{co} \\
 T_{coo}^{(1)} &= T_{coo}^{(2)} = T_{coo} \\
 T_{coi}^{(1)} &= T_{coi}^{(2)} = T_{coi}
 \end{aligned}$$

where the superscript ( $i$ ) represents the  $i$ th copy of each global variable in  $\mathbf{X}$ . Note that in sub-problem 4, there are only constraints and no cost function. Any feasible point is optimal and there could be infinitely many solutions. However, within step 5 of either Algorithm 1 or Algorithm 2, the cost function is augmented with a penalty term coming from the consensus constraints, which remedies the illness in the original sub-problem. With this distributed formulation, the original optimization problem breaks down to several sub-problems with reduced dimensions and less constraints, which can be solved in parallel. The difficulty of solving each sub-problem is much lower than solving the original large problem. However, since some global variables have multiple copies of local variables, the total number of design variables is larger with a distributed formulation. A hardware implementation of this distributed decision making process is shown in Fig. 4. The bottom layer corresponds to the sensing network that collects the required operating conditions. Above the sensing layer is a basic agent layer that includes all the basic agents representing behaviors of all the components. The basic agents could be implemented by equipment manufacturers or could be identified on the fly from collected data. On top of the basic-agent layer, there is an optimizer agent layer, which is responsible for solving each sub-problem. Each optimizer agent calls the related basic agents iteratively to optimize its corresponding cost function independently and in parallel with the other optimization agents. The consensus requirements among the local variables need to be satisfied by a coordination layer as shown in the top that collects the local copies of all the variables, updates the dual variables accordingly and feeds the updated dual variables back to the optimizer agents to let them re-optimize with respect to the updated information. The iteration process continues until certain termination criteria are met.

Note that the proposed multi-agent framework provides ample flexibility in designing the control topology. By using the group number property in the basic agents, a user could easily design a grouping or topology of a multi-agent system in which different factors could be considered, such as the optimization dimensions, network speed and cost, physical distances between devices and cost of microcontrollers. With the agent grouping shown with dashed boxes in Fig. 3, the original centralized problem is fragmented into 4 sub-problems with lower dimensions as shown in Table 2. However, this distributed control structure requires significant network communications for information exchange, primarily between the optimizer layer and coordination layer in Fig. 4. If a different control topology were utilized, say, a 2-group controller with the 1st group containing the cooling tower, cold well and variable-speed pump and the 2nd group containing the chiller, the network requirements could be reduced although the sub-problem dimensions would be higher as shown in Table 2. The number of iterations required to reach convergence increases with the number of shared variables. The 4-group controller shown in Table 2 has four more shared variables than the controller with 2 groups, thus more iterations are needed to arrive at a consensus solution and heavier



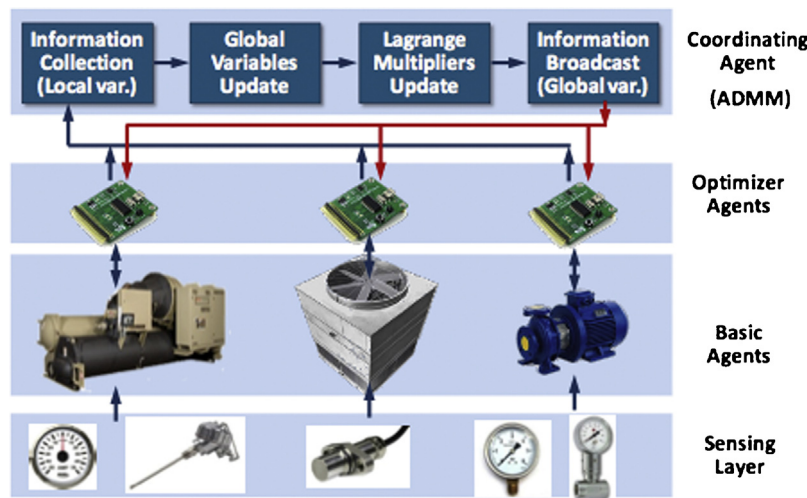


Fig. 4. Architecture of hardware implementation of the multi-agent controller.

**Table 2**  
Complexity comparison among different problem formulations.

	Centralized	Distributed with 4 groups	Distributed with 2 groups
Total num. of variables	7	14	10
Averaged num. of variables per sub-problem	7	3.5	5
Num. of network traffic channels	0	4	2

network traffic would result for the 4-group control topology. So a finer granularity in the grouping leads to good scalability but is more costly in the hardware implementation as well as network communication. A good controller design would need to leverage the scalability and network traffic requirements.

### 5.5. Optimization results

Before the optimization process, all the optimization variables, objective functions and constraints were scaled to assure that all these quantities had similar importance in the optimization. Fig. 5 presents the evolution of the iterative optimization process for a cooling load of 3517 kW and wet-bulb temperature of 26.67 °C. Fig. 5(a) shows the evolution of global optimization variables (vector  $\mathbf{Z}$ ) whereas the two plots in Fig. 5(b) show the evolution of normalized violations of agent behavior-related constraints from Eq. (15) and the total power consumption evaluated with the corresponding global variables, respectively. It can be observed that the optimization variables approach the optimal values in an oscillatory manner and the oscillation magnitudes diminish with more iterations. The oscillation is caused by the swinging weights between the actual cost function and the penalty term due to the consensus constraint violations in Step 5 of Algorithm 2. So the optimization process tries to minimize the actual cost function and enforce the consensus constraints, alternately. In this case study, the stopping tolerance used  $\varepsilon = 10^{-3}$  and the specific initial guess in the plot required 500 iterations to achieve convergence. This large number of iterations is caused by the poor initial guess that has been randomly generated. For example, the initial guess for the airflow rate  $m_a$  is 50% of the nominal flow, which differs significantly from the optimal level 75%. However, in a real implementation, generally the optimal operation would not change dramatically between two consecutive decision steps so the optimal operating point obtained in the previous step could be used as the initial guess for the current decision step which could reduce the number of iterations and also the computational burden. In addition, it can be observed from Fig. 5(a) that the optimization variables approach their opti-

mal levels very quickly (within approximately 70 iterations) and then oscillate around the optimal values. So in a real implementation when the decision time runs out but convergence has not been achieved, the most recent feasible point could be used as the final decision. Fig. 6 plots the intermediate feasible points along with the corresponding power consumptions within the optimization iterations shown in Fig. 5, where the feasible points are defined as those that have the sum of normalized constraint violations below  $10^{-3}$ . The first feasible point occurs at the 76th iteration and the corresponding total power is 742.1 kW, which is very close to the optimal power consumption 741.8 kW. The subsequent iterations do not reduce the energy consumption significantly but are necessary to reach convergence if time is allowed. For the example testing conditions shown in Fig. 5, the full 500 iterations used 41 s computation time while the 76 iterations took approximately 6 s on a laptop with Intel i3 CPU and 6 GB RAM implemented in a sequential manner where the four sub-problems were solved sequentially instead of in parallel. Since supervisory control commands typically get updated with time steps in the magnitude of hours, the proposed approach is amenable for real-time implementation in this case study.

To assess the energy savings, the multi-agent control approach was compared with a heuristic control strategy, which has been implemented at the Purdue Northwest Chiller Plant. The baseline (heuristic) strategy keeps the condenser water flow relatively constant while the fan speed is modulated to control a condenser water supply temperature set point of 22.8 °C. The performance map in Fig. 7 shows the comparison of the cooling plant (chiller, cooling tower and pump) energy efficiency (COP) attainable with the two control strategies at half load (3517 kW) and ambient wet-bulb temperature of 26.7 °C. Given that the centralized problem formulated in Eq. (15) has two degrees-of-freedom, the normalized values of the two independent variables (air flow and condenser water flow) were used to generate the contour plot in Fig. 7 whereas the values of all the remaining variables were calculated based on the equality constraints shown in Eq. (15). Under these specific operating conditions, the multi-agent control leads to a 12.2%

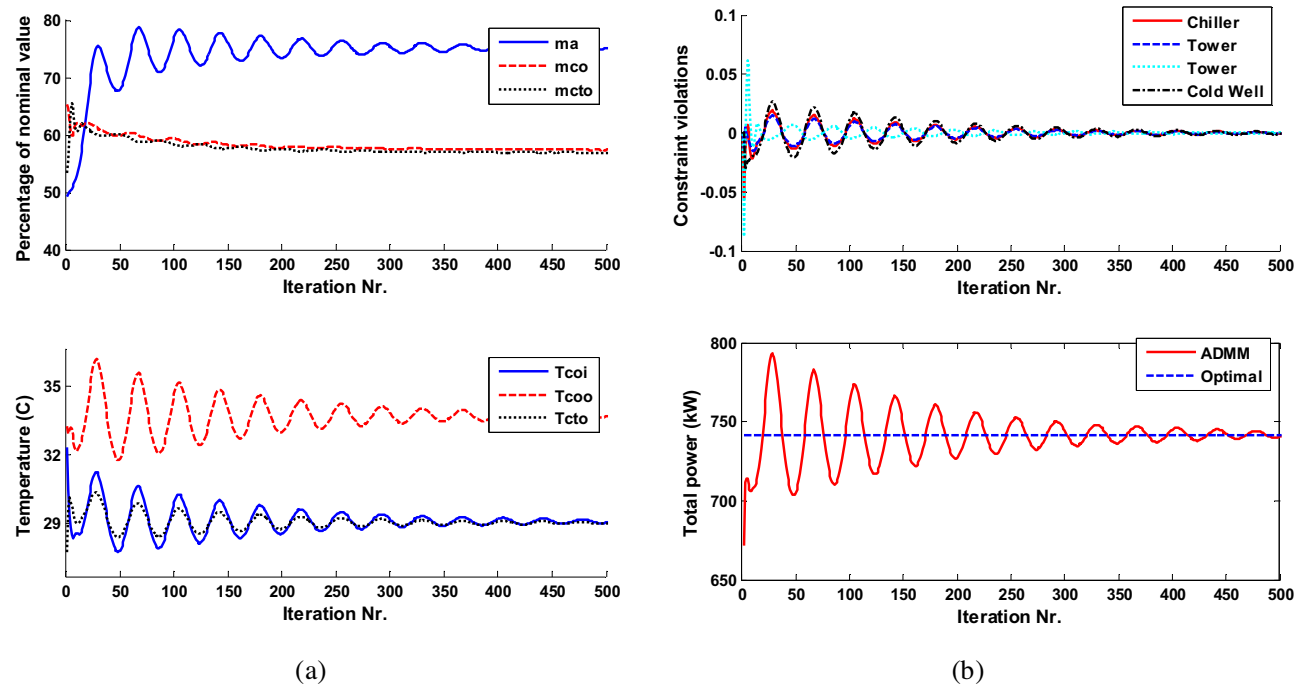


Fig. 5. (a) Evolution of the global optimization variables. (b) Evolution of normalized constraint violations and total power consumption.

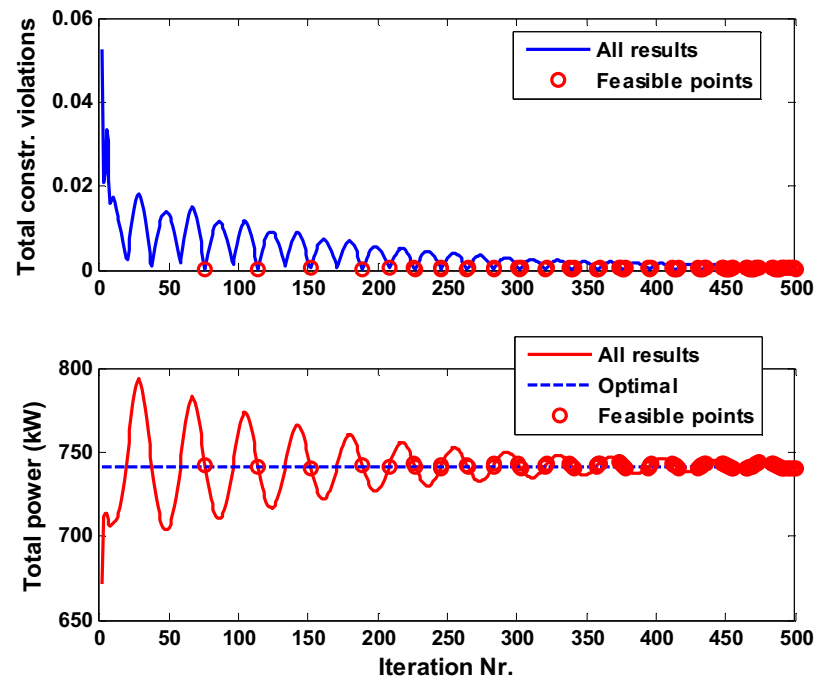


Fig. 6. Feasible points in the iterations of the multi-agent optimization process.

Table 3  
Energy comparisons between baseline and multi-agent controls.

Cooling Load (kW)	Wet bulb (C)	Current Control Strategy				Multi-Agent Control				Power Savings %	
		ma (relative)	mco (relative)	Power (kW)	COP	ma (relative)	mco (relative)	Power (kW)	COP		
7034	26.7	1.00	1	1381	5.10	0.92	0.76	1353	5.20	2.5	
7034	10	0.61	1	1175	5.99	0.92	0.64	1094	6.43	10.9	
3517	26.7	1.00	1	808	4.35	0.75	0.57	742	4.74	12.2	
3517	10	0.35	1	659	5.34	0.70	0.50	540	6.52	8.6	

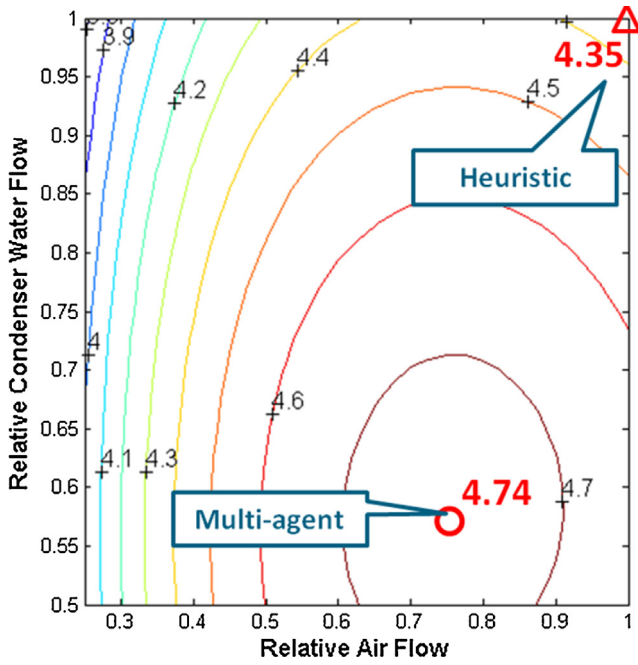


Fig. 7. Cooling plant COP contour plot for Load = 3517 kW and web-bulb temperature = 26.7 °C.

energy efficiency enhancement. It can be also noted that the multi-agent control is very close to the optimal. The multi-agent approach was also evaluated under three other operating conditions and the results are listed in Table 3. Compared to the baseline control, the proposed multi-agent control is able to achieve approximately 10% energy savings under the operating conditions considered, except for the case where the load is 7034 kW and the ambient wet bulb is 26.7 °C because the baseline control is already close to being optimal.

It is worth mentioning that the relationship between the total power and the two independent design variables is close to being convex, as can be observed in Fig. 7. This nice property within the considered problem provides good robustness for the multi-agent control algorithm. In the simulation tests convergence was achieved for almost all initial points that were generated randomly; although the convergence rate was highly dependent on the goodness of the initial guess. However, the optimization problem in building energy systems is in general not convex and convergence cannot be guaranteed, as illustrated in the next case study.

## 6. Case study 2

In the second case study, a typical centralized air conditioning system serving a multi-zone building is considered as shown in Fig. 8. Air is conditioned in the AHU and then supplied to each conditioned zone through a dedicated variable-air-volume (VAV) box. The zone air temperature is regulated by varying the entering airflow rate through modulation of the VAV damper. The return air (RA) from the space is circulated back to the AHU and mixed with the outdoor air (OA) before going through the heating/cooling coil for air conditioning. Supply air temperature ( $T_{sa}$ ) is controlled to a setpoint by changing the refrigerant evaporating temperature for direct-expansion (DX) systems. In this case study, a multi-stage DX unit is considered and the compressors are staged by feedback control to achieve a supply air temperature setpoint. Fan speed modulates to maintain a setpoint supply duct pressure ( $P_{sa}$ ). The supply air temperature and pressure setpoints are the two control variables in an actual implementation. There

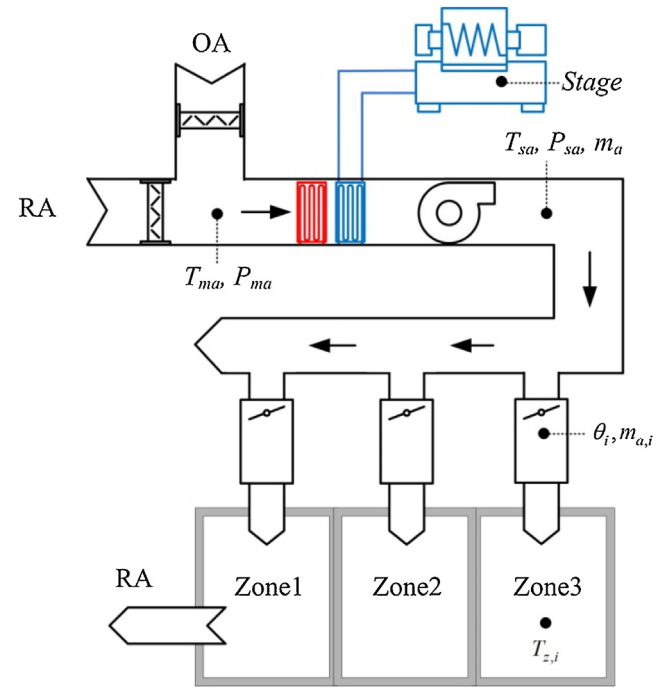


Fig. 8. System diagram of a centralized air-conditioning system for a multi-zone building.

is a tradeoff between the compressor and fan power in the supply air temperature control. To achieve a required sensible load, higher supply air temperature reduces the compressor power but leads to higher fan power to deliver more airflow. The optimal supply air temperature would be a balancing point between the compressor and fan power. The supply duct pressure setpoint determines the maximum airflow rate and thus, would affect the balance point indirectly. Outdoor air dampers are controlled to provide either a minimum ventilation rate or for economizer operation. Return air and relief air dampers coordinate to keep a constant zone air pressure ( $P_z$ ).

### 6.1. Component models

#### 6.1.1. DX unit

This study focuses on a direct-expansion cooling system in which the air exchanges heat directly with the refrigerant going through the cooling coil. There are six stages in the compressor excluding the off stage. By controlling the staging bandwidth, the system can achieve continuous capacity modulation; thus, Stage is a continuous variable with value between 0 and 6. A gray-box model was developed for this DX unit that was trained with field data; the details can be found in Ref. [23]. Given the corresponding boundary conditions ( $T_{ma}$  - mixed air temperature;  $w_{ma}$  - mixed air humidity ratio;  $m_a$  - air mass flow rate;  $T_{amb}$  - ambient temperature) and compressor stage, the DX unit model will output cooling coil outlet air temperature ( $T_{la}$ ), unit power consumption ( $Pow_{DX}$ ) and sensible heat ratio ( $SHR$ ).

$$[T_{la}, Pow_{DX}, SHR] = DX(T_{ma}, w_{ma}, m_a, T_{amb}, Stage)$$

#### 6.1.2. Fan

Energy is consumed by the fan to deliver conditioned air to the zone spaces. The instantaneous fan power is a function of the airflow rate ( $m_a$ ) delivered and pressure rise (external static pres-

sure, ESP) across the fan, which can be calculated as the pressure difference between the fan inlet and outlet:

$$ESP = P_{sa} - P_{ma},$$

where  $P_{ma}$  is the mixed air pressure. In this study, a quadratic polynomial form shown in the following equation is used and the coefficients were obtained through curve fitting using field data:

$$\begin{aligned} Pow_{fan} &= a_0 + a_1 m_a + a_2 m_a^2 + a_3 ESP + a_4 ESP^2 + a_5 ESP \cdot m_a \\ &:= Fan_{pow}(m_a, P_{sa}, P_{ma}) \end{aligned}$$

Fan energy contributes to a temperature rise in the air, which is calculated as

$$T_{sa} = T_{la} + \frac{Pow_{fan}}{m_a c_p} := Fan_T(T_{la}, m_a, P_{sa}, P_{ma})$$

where  $c_p$  is the specific heat of air (1000 J/kg·K).

### 6.1.3. Damper

The VAV box damper has a feedback control based on the space temperature. By varying the damper opening, the airflow rate that enters the zone space can be modulated to regulate the space temperature. So the VAV box damper model would predict the airflow rate given the damper opening and pressure drop. Let  $P_{sa}$  be the air pressure in the supply duct and  $P_z$  be the zone space pressure. Then the pressure drop across each air damper is

$$\Delta P = P_{sa} - P_z,$$

The airflow rate that is going through the air damper can be formulated as (see Ref. [24])

$$m_a = \text{sign}(\Delta P) A_{damper} \left( \frac{2\rho\Delta P}{\exp(a + b(1 - \theta))} \right)^{0.5} := \text{Damper}(P_{sa}, P_z, \theta),$$

where  $A_{damper}$  is the damper section cross area,  $\rho$  is the air density,  $\theta$  is the damper opening (%) and  $a, b$  are parameters that are associated with damper characteristics. These two parameters were determined from field data. Note that in the case study, the 3 dampers were assumed to have the same characteristics with different sizes.

### 6.1.4. Building envelope model

A gray-box model with a resistance-capacitance thermal network was trained from field data to represent the zone thermal behaviors (see Ref. [25]). A discrete-time state-space representation of the model is

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}_w\mathbf{W}_k + \mathbf{B}_u\mathbf{Q}_{sen,k} \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k \end{aligned}$$

where  $\mathbf{x}$  is a state vector containing all the nodal temperatures in the thermal network;  $\mathbf{W}$  contains all disturbance inputs including weather conditions and internal heat gains from occupants and electrical devices;  $\mathbf{y} = [T_{z,1}, T_{z,2}, T_{z,3}]$  is the output vector containing the space air temperatures of the three zones;  $\mathbf{Q}_{sen} = [Q_{sen,1}, Q_{sen,2}, Q_{sen,3}]$  is a controllable input vector consisting of the sensible cooling/heating capacities of the zones; and the subscript  $k$  indicates the time step. The sensible capacity of each zone is calculated by

$$Q_{sen,i} = m_{a,i} c_p (T_{sa} - T_{z,i}),$$

In order to achieve a set of next step (step  $k+1$ ) zone air temperature setpoints  $\mathbf{y}_{k+1}^{sp}$ , the required sensible capacities can be calculated as

$$\mathbf{Q}_{sen,k} = [\mathbf{C}\mathbf{B}_u]^{-1} [\mathbf{y}_{k+1}^{sp} - \mathbf{C}\mathbf{A}\mathbf{x}_k - \mathbf{C}\mathbf{B}_w\mathbf{W}_k]. \quad (16)$$

### 6.1.5. Splitter

Assuming no heat or pressure losses, the splitter model is based on a simple mass balance:

$$m_a = \sum_{i=1}^3 m_{a,i}$$

### 6.2. Centralized formulation

When all devices are assigned with the same group number, a centralized optimization is formulated that tries to find the optimal operating point given the boundary conditions as well as the required cooling capacities such that the total power consumption is minimized. The problem needs to be solved at each decision step as the boundary conditions and capacity requirements change with time. The optimization variables (also the global variables in the distributed optimization problem) are

$$\mathbf{Z} = [m_a, \text{Stage}, T_{la}, T_{sa}, P_{sa}, \theta_1, \theta_2, \theta_3, m_{a,1}, m_{a,2}, m_{a,3}]^T.$$

The optimization problem to be solved at each time step is formulated as

$$\min_{\mathbf{Z}} \{Pow_{DX} + Pow_{fan}\} \quad (17)$$

subject to

$$m_a = \sum_{i=1}^3 m_{a,i} \} \text{LinConst} \quad (18)$$

$$\left. \begin{aligned} Q_{sen,i} &= m_{a,i} c_p (T_{sa} - T_{z,i}), i = 1, 2, 3 \\ m_{a,i} &= \text{Damper}(P_{sa}, P_{z,i}, \theta_i), i = 1, 2, 3 \\ T_{sa} &= \text{Fan}_T(T_{la}, m_a, P_{ma}, P_{sa}) \\ T_{la} &= \text{DX}_{Tla}(T_{ma}, W_{ma}, m_a, T_{amb}, \text{Stage}) \end{aligned} \right\} \text{NonlinConst} \quad (19)$$

$$\left. \begin{aligned} \theta_i &\in [0, 100], i = 1, 2, 3, \\ \text{Stage} &\in [0, 6], \\ P_{sa} &\in [0, 1.2](\text{inW.C.}) \end{aligned} \right\} \text{VariableBound} \quad (20)$$

The under-scored variables are boundary conditions provided at the beginning of each decision step. The zone space pressure  $P_{z,i}$  is normally maintained slightly above atmospheric pressure. In this study, values of  $P_{ma} = -0.3$  (inW.C.) and  $P_{z,i} = 0.1$  (inW.C.) gauge pressure were assumed. The required sensible capacity  $Q_{sen,i}$  was also taken as a boundary condition since it is calculated using Eq. (16) at the beginning of each decision step after the next step zone temperature setpoint has been prescribed. As a result, the time index  $k$  is not included in the notation for zone sensible cooling.

The cost function in Eq. (17) is the total power consumption for the DX unit and the supply fan. The equality constraints in Eqs. (18) and (19) correspond to the models described in the preceding section. The interval type constraints in Eq. (20) are due to the capacities of the physical components. Note that the boundary conditions are omitted in the remaining formulations for ease of notation.

### 6.3. Distributed formulation

In the centralized formulation, there are eleven optimization variables, nine (mostly nonlinear) equality constraints and five interval-type constraints. With a distributed formulation where different devices or even different elements within the same device are assigned different group numbers, the large-scale centralized problem can be broken down into several sub-problems as follows.

$$\begin{aligned} \text{Sub-problem 1: } \min & [Pow_{DX}(m_a^{(1)}, \text{Stage}^{(1)})] \\ \text{s.t. } & \text{Stage}^{(1)} \in [0, 6] \end{aligned}$$



$$\text{Sub-problem 2: } \min \left[ \text{Pow}_{fan} \left( m_a^{(2)}, P_{sa}^{(1)} \right) \right]$$

$$\text{s.t. } P_{sa}^{(1)} \in [0, 1.2]$$

$$\text{Sub-problems 3–5: } \min \left[ 1/\delta(Q_{sen,i} - m_{a,i}^{(1)} c_p(T_{sa}^{(i)} - T_{z,i})) \right] \text{ for } i = 1, 2, 3,$$

$$\text{Sub-problems 6–8: } \min \left[ 1/\delta(m_{a,i}^{(2)} - \text{Damper}(\theta_i^{(1)}, P_{sa}^{(i+1)})) \right] \text{ for } i = 1, 2, 3,$$

$$\text{s.t. } \theta_i^{(1)} \in [0, 100]$$

$$\text{s.t. } P_{sa}^{(i+1)} \in [0, 1.2]$$

$$\text{Sub-problem 9: } \min \left\{ 1/\delta \left( m_a^{(3)} - \sum_{i=1}^3 m_{a,i}^{(3)} \right) \right\}$$

$$\text{Sub-problem 10: } \min \left[ 1/\delta(T_{la}^{(1)} - DX_T(m_a^{(4)}, \text{Stage}^{(2)})) \right]$$

$$\text{s.t. } \text{Stage}^{(2)} \in [0, 6]$$

$$\text{Sub-problem 11: } \min \left[ 1/\delta(T_{sa}^{(4)} - \text{Fan}_T(T_{la}^{(2)}, m_a^{(5)}, P_{sa}^{(5)})) \right]$$

$$\text{s.t. } P_{sa}^{(5)} \in [0, 1.2]$$

The consensus constraints are:

$$m_a = m_a^{(i)}, \forall i \in \{1, 2, 3, 4, 5\},$$

$$\text{Stage} = \text{Stage}^{(i)}, \forall i \in \{1, 2\},$$

$$T_{la} = T_{la}^{(i)}, \forall i \in \{1, 2\},$$

$$T_{sa} = T_{sa}^{(i)}, \forall i \in \{1, 2, 3, 4\},$$

$$P_{sa} = P_{sa}^{(i)}, \forall i \in \{1, 2, 3, 4, 5\},$$

$$\theta_j = \theta_j^{(1)}, \forall j \in \{1, 2, 3\},$$

$$m_{a,j} = m_{a,j}^{(i)}, \forall i \in \{1, 2, 3\}, \forall j \in \{1, 2, 3\}$$

where

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ \infty, & x = 0 \end{cases}$$

is the Dirac delta function.

The distributed formulation was composed by the framework where each component was assigned a different group number. In addition, the DX unit and fan were each further divided into two groups, one with the cost function and the other one with the corresponding constraint. So this case study is a demonstration of a multi-agent controller with the finest grouping, although coarser granularity might be more suitable for practical implementation. With this distributed formulation, the original problem breaks down to eleven sub-problems most of which have dimension less than three. Sub-problems 9 and 11 have four design variables but Sub-problem 9 has an explicit solution that will be discussed shortly. In addition, each sub-problem with at most two constraints of interval type has a much lower solution complexity. This makes the proposed method scalable to increased problem size.

Note that sub-problems 3–11 add an infinite penalty to the cost function whenever the corresponding constraint is violated. This is another instance of the situation in sub-problem 4 of case study 1 where only constraints exist and there is no cost function. In either case, Step 5 of the ADMM algorithm is carried out as

$$\min_{\mathbf{X}_i} \|\mathbf{X}_i - \mathbf{F}_i \mathbf{Z}^{[k]} + \mathbf{Y}_i^{[k]}/\sigma\|_2^2$$

$$\text{s.t. } f_i(\mathbf{X}_i) = 0$$

which is also equivalent to the Euclidean projection of  $\mathbf{F}_i \mathbf{Z}^{[k]} - \mathbf{Y}_i^{[k]}/\sigma$  onto the constraint hyper-surface:

$$\mathbf{X}_i^{[k+1]} = \Pi_{C_i}(\mathbf{F}_i \mathbf{Z}^{[k]} - \mathbf{Y}_i^{[k]}/\sigma),$$

where  $C_i$  is the hyper-surface defined by  $f_i(\mathbf{X}_i) = 0$  or equivalently, by the corresponding equality constraint within Eqs. (18) and (19).

In addition, if the corresponding equality constraint is linear in  $\mathbf{X}_i$ , say  $\mathbf{D}_i^T \mathbf{X}_i - b = 0$ , an analytic solution can be obtained as

$$\mathbf{X}_i^{[k+1]} = \mathbf{F}_i \mathbf{Z}^{[k]} - \mathbf{Y}_i^{[k]}/\sigma - \left[ \mathbf{D}_i^T (\mathbf{F}_i \mathbf{Z}^{[k]} - \mathbf{Y}_i^{[k]}/\sigma) - b \right] \mathbf{D}_i / \|\mathbf{D}_i\|_2^2.$$

The equality constraint for sub-problem 9 is linear so the corresponding sub-problem can be easily solved. This is important because this specific constraint physically corresponds to an air splitter or merger and for a centralized air-conditioning system with a large number of air splits, the computational requirement does not increase much due to this property.

#### 6.4. Optimization results

In a real implementation, the optimization problem could be solved and applied for different decision intervals in response to varying operating conditions. However, this study only presents optimization results for one example operating condition since similar performance was observed at other conditions. The operating conditions that are considered are:  $w_{ma} = 0.009$  kg water/kg air,  $T_{ma} = 26^\circ\text{C}$ ,  $T_{z,1} = 25^\circ\text{C}$ ,  $T_{z,2} = 24.5^\circ\text{C}$ ,  $T_{z,3} = 24^\circ\text{C}$ ,  $T_{amb} = 31^\circ\text{C}$ ,  $Q_{sen,1} = 28$  kW,  $Q_{sen,2} = 26$  kW and  $Q_{sen,3} = 23$  kW.

As a first step, the variables were normalized to have comparative scales to make sure that penalties due to consensus violations were assigned fairly. Fig. 9 plots the variation of the normalized variables as the coordination procedure proceeds. Some variables have local copies among several optimizer agents and they are reflected by multiple curves within the same subplot. The stopping tolerance used  $\varepsilon = 10^{-2}$  and the algorithm stops at the 119th iteration. It can be seen that in the first few tens of iterations, different local copies of the same variable disagree since each sub-problem optimization is carried out for its own benefit. But the deviations diminish with more iterations due to inter-agent coordination. This reduction of disagreement can also be reflected in the top plot of Fig. 10 as discussed in the following paragraph.

Plugging the estimated global variables  $\mathbf{Z}$  into the centralized problem, the cost function as well as the constraint satisfaction were evaluated at different iterations. The top plot in Fig. 10 shows the normalized constraint violations and each curve corresponds to one constraint listed within Eqs. (18) and (19). The plotted constraint violations are also indicators of the disagreement level in the different local copies of the same variable since if a consensus is reached, the global variable in  $\mathbf{Z}$  should equal all the corresponding local variables which strictly satisfy the constraints within Eqs. (18) and (19). So in the top plot of Fig. 10, the local variables reach consensus in an oscillatory manner and the oscillation magnitude decreases with iterations. The bottom figure plots the evolution of the total power consumption, which is the output of the cost function in Eq. (17). A similar pattern could be observed in this iterative process compared to case study 1: the algorithm takes turns in reducing the power and enforcing the consensus constraints. When the constraint violation is small, the iterations move in a cost descent direction leading to a faster power drop. But these moves compromise consensus satisfaction and the next few iterations move towards a consensus reduction direction where the total power is not decreasing much or even increasing, as in the 1st case study. Under the demonstrated operating conditions, convergence is reached in the 119th iteration where the variable changes between iterations and consensus disagreement are below the threshold set in the stopping criterion. For the testing conditions shown in Fig. 9, it took 114s to reach convergence on a laptop computer with Intel i3 CPU and 6 GB RAM implemented in a sequential manner. More than 80% of the computation time was consumed in evaluating the DX unit model which involves internal iterations for dry coil conditions.



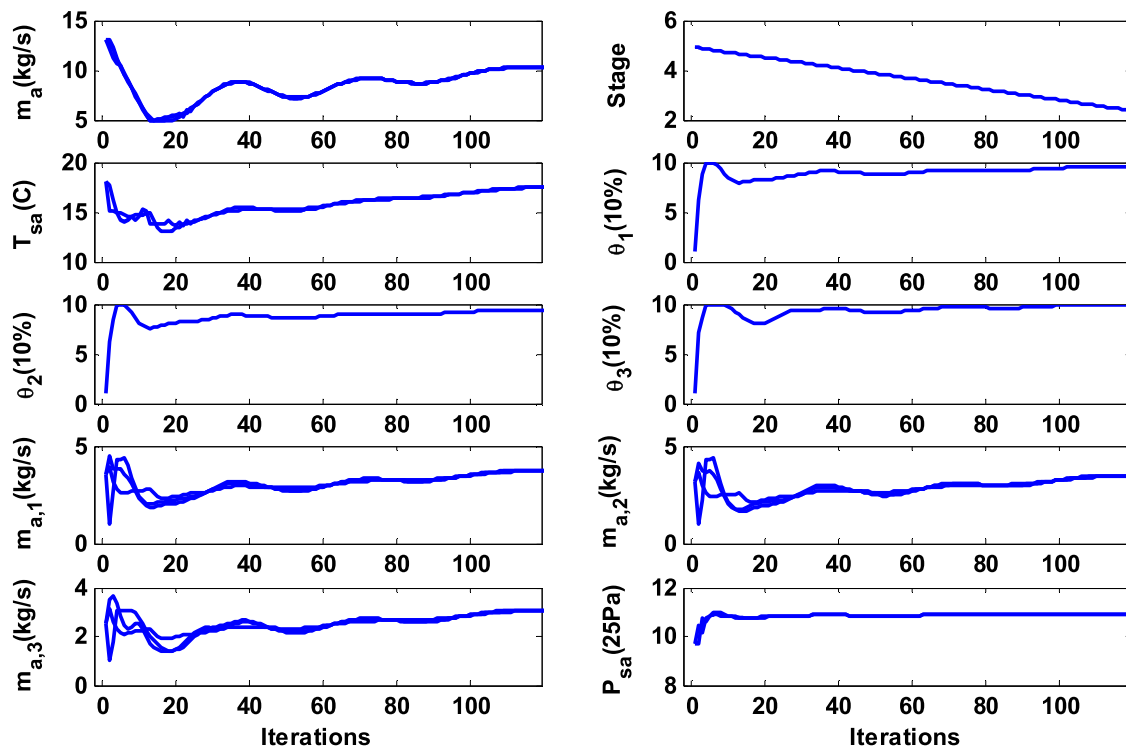


Fig. 9. Evolution of local variables.

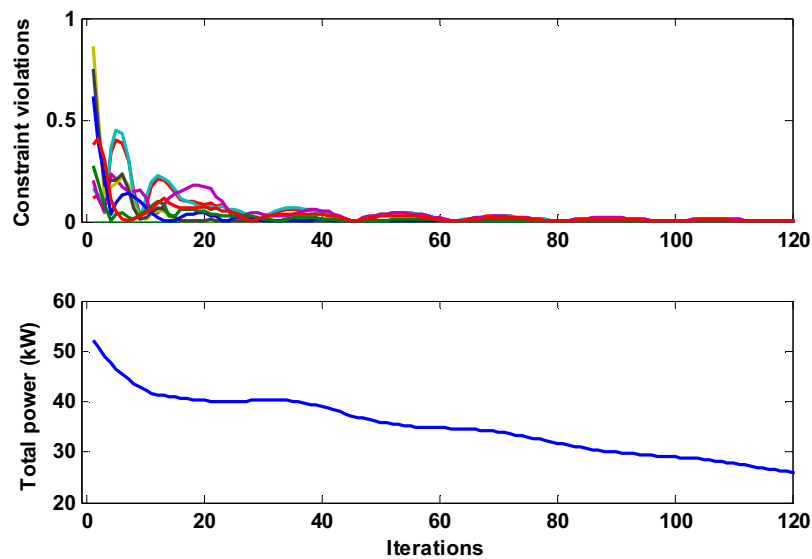


Fig. 10. Evolution of power and constraint violations.

The centralized optimization has eleven variables and many constraints. Through some engineering, the problem was reduced to a 2 degree-of-freedom optimization that is relatively easy to solve and the optimal point for this specific operating condition was:  $Stage = 2.19$ ,  $T_{sa} = 17.8^\circ\text{C}$ ,  $\theta_1 = 96\%$ ,  $\theta_2 = 94\%$ ,  $\theta_3 = 100\%$ ,  $P_{sup} = 300\text{ Pa}$  with a minimum total power consumption of 24.9 kW. To assess the energy savings, a baseline control strategy for the conventional control was considered with  $T_{sup} = 14^\circ\text{C}$  and  $P_{sa} = 280\text{ Pa}$ , which had been implemented physically in the case study building before the year of 2013 (see Ref. [23]). The energy consumption with the baseline strategy is 43.5 kW. So there is a 42.7% energy savings potential and the multi-agent control was able to find a solution with 25.9 kW power consumption which covers 94.6% of

the maximum energy savings. The savings potential is not fully recovered because the total power is not sensitive to the supply pressure setpoint which creates a lot of local minimums and the obtained solution is a local minimum.

In contrast to the 1st case study, the problem considered here is non-convex so convergence is not guaranteed. In the simulation tests, it was observed that different initial guesses gave quite different types of solutions (e.g., local minimums, points without consensus or even divergent solutions). A practical approach is to implement a multi-start scheme and find the consensus solution with the minimum power. Testing results show that with a multi-start scheme where the start points were randomly generated, 60% of the start points led to some consensus solution that covers at

least 92% of the maximum energy savings. Also, using the optimal solution from the previous step as the initial guess for the current time step should improve the performance since optimal operating conditions do not change significantly between decision steps.

## 7. Conclusion & discussion

This paper presented a general multi-agent control approach for the optimization of building energy systems. The proposed approach consists of two main elements: a multi-agent control framework and multi-agent decision making algorithms. With the help of the framework, a multi-agent controller can be easily set up with limited engineering effort. Then the distributed decision making algorithms can be used to optimally control and coordinate different components to reduce the overall energy consumption within a building energy system. Two case studies were considered to test the proposed approach. Testing results show that the multi-agent control was able to find near-optimal control solutions under different operating conditions and significant energy savings were achieved compared to baseline control strategies.

Note that this paper is primarily concerned with the algorithm design of a multi-agent controller. However, some issues related to hardware implementation have also been considered such as control topology design and inter-controller communications. The framework provides good flexibility in control topology design where multiple devices can be grouped together and controlled by one local controller or a single complex component could be controlled by multiple local controllers to leverage control scalability and network traffic requirements.

While developed framework should handle a wide range of building energy systems, the multi-agent control algorithm has some limitations and might not be directly applicable to some types of equipment. For example, many HVAC systems have multiple operating modes and their optimal control becomes a mixed-integer programming problem that cannot be solved with the proposed algorithms. Consensus-based optimization algorithms are typically much more efficient than other distributed optimization schemes, such as the ping-pong scheme, since optimizations are carried out in parallel with multiple computing engines. But the main drawback within actual implementation of a consensus-based algorithm is that most of the intermediate iterations are non-consensus and thus, are not feasible. This poses a potential risk that the decision time runs out but no consensus point has been reached. These issues need to be addressed in future work to make the approach more robust.

## Acknowledgment

This research was supported by the National Science Foundation under Grant No. 1329875.

## References

- [1] US Department of Energy (DOE), Buildings Energy Data Book, 2008.
- [2] Z. Mo, A. Mahdavi, An agent-based simulation-assisted approach to bi-lateral building systems control, 8th International IBPSA Conference (2003).
- [3] P. Davidsson, M. Boman, Distributed monitoring and control of office buildings by embedded agents, *Inf. Sci.* 171 (2005) 293–307.
- [4] L. Zhang, G. Zhang, B. Shen, X. Xie, Q. Yan, Building energy saving design based on multi-Agent system, 5th IEEE Conference on Industrial Electronics and Applications (2010).
- [5] ASHRAE, HVAC applications. ASHRAE Handbook, Fundamentals, (2011) Atlanta, GA, USA.
- [6] J.E. Braun, G.T. Diderrich, Near-optimal control of cooling towers for chilled-water systems, *ASHRAE Trans.* 96 (1990) 9–13.
- [7] K.H. Drees, J.E. Braun, Development and evaluation of a rule-based control strategy for ice storage systems, *HVAC&R Res.* 2 (1996) 312–334.
- [8] Geoffrey van Moeseke, Isabelle Bruyère, André De Herde, Impact of control rules on the efficiency of shading devices and free cooling for office buildings, *Build. Environ.* 42.2 (2007) 784–793.
- [9] P. Zhao, S. Suryanarayanan, M.G. Simes, An energy management system for building structures using a multi-agent decisionmaking control methodology, *Ind. Appl. IEEE Trans.* 49 (2013).
- [10] R.Yang Wang, L. Wang, Multi-agent control system with intelligent optimization for smart and energy-efficient buildings, *IECON-36th Annual Conference on IEEE Industrial Electronics Society* (2010).
- [11] S.J. Treado, An Agent-Based Methodology for Optimizing Building HVAC System Performance, *ASHRAE Trans.* (2010) 116, 2, p124–134.
- [12] P. Morosan, R. Bourdais, D. Dumur, J. Buisson, Distributed model predictive control based on Benders' decomposition applied to multisource multizone building temperature regulation, 49th IEEE Conference on Decision and Control (2010).
- [13] P.B. Sun, Q. Luh, Z. Jia, F.Wang Jiang, C. Song, An integrated control of shading blinds, natural ventilation, and HVAC systems for energy saving and human comfort, 6th annual IEEE Conference on Automation Science and Engineering (2010).
- [14] M.Y. Lamoudi, M. Alami, P. Beguery, Distributed constrained Model Predictive Control based on bundle method for building energy management, 50th IEEE Conference on Decision and Control and European Control Conference (2011).
- [15] J. Cai, J.E. Braun, A generalized control heuristic and simplified model predictive control (MPC) strategy for direct-expansion (DX) air-conditioning systems, *Sci. Technol. Built Environ.* (2015).
- [16] M. Tiller, *Introduction to Physical Modeling with Modelica*, USA: Springer, New York, NJ, 2001.
- [17] A. Nedic, A. Ozdaglar, Cooperative distributed multi-agent optimization, *Convex Optimization in Signal Processing and Communications* (2010).
- [18] D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.
- [19] S. Boyd, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (2011) 1–122.
- [20] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [21] J.E. Braun, Performance and control characteristics of large central cooling systems, *ASHRAE Trans.* 93 (Part 1) (1987).
- [22] J.E. Braun, Methodologies for the design and control of chilled water systems, in: Ph.D Thesis, University of Wisconsin, Madison, WI, 1988.
- [23] J. Cai, J.E. Braun, Gray-box modeling of multi-stage direct-expansion (DX) units to enable control system optimization, *ASHRAE Trans.* 121 (2015) p203–p216.
- [24] Lawrence Berkeley National Laboratory, Modelica Buildings Library, URL: [simulationresearch.lbl.gov/modelica](http://simulationresearch.lbl.gov/modelica).
- [25] J. Cai, J.E. Braun, A practical and scalable inverse modeling approach for multi-zone buildings, 9th International Conference on System Simulation in Buildings (2014).