energies

MDPI

Article

A Distributed Optimization Method for Energy Saving of Parallel-Connected Pumps in HVAC Systems

Xuetao Wang, Qianchuan Zhao * and Yifan Wang

Department of Automation, BNRist, Center for Intelligent and Networked Systems, Tsinghua University, Beijing 100084, China; wang-xt17@mails.tsinghua.edu.cn (X.W.); wangyifa17@mails.tsinghua.edu.cn (Y.W.)

* Correspondence: zhaoqc@tsinghua.edu.cn

Received: 8 July 2020; Accepted: 27 July 2020; Published: 31 July 2020



Abstract: Motivated by the importance and challenges of the energy saving problem of parallel-connected pumps in heating, ventilation, and air-conditioning (HVAC) systems, we propose a distributed optimal control algorithm for on-off status and flow rate set points of parallel-connected pumps in HVAC systems. The proposed algorithm consists of two parts: First, in order to process the network information, we apply the breadth first search algorithm to construct a tree for exchanging messages. Second, all nodes coordinate with each other and randomly sample the speed ratios. To our best knowledge, the algorithm proposed in this paper is the first effort to address the challenges of existing studies at the same time. The algorithm solves the pump optimization problem in a distributed manner, achieves the minimum pump energy consumption and has the convergence guarantee. Even if some of the pumps break down, the whole system can still be working and have great flexibility. Simulation experiments on six parallel-connected pumps are provided for different working cases to demonstrate the effectiveness of the proposed algorithm and compare with the other four methods. The results show that our algorithm strictly satisfies the demand constraints and presents good energy saving potential, the convergence guarantee, and flexibility. The maximum energy saving can be up to 29.92%.

Keywords: distributed; parallel-connected pumps; speed ratio; optimal control; spanning tree

1. Introduction

The heating, ventilating, and air-conditioning (HVAC) system makes up approximately 50% of building energy consumption and 20% of total energy consumption [1]. Parallel-connected pumps, which provide as wide a flow range as possible, are the main power consumer devices in HVAC systems. The report of the European Commission shows that pump systems account for nearly 22% of the energy consumed by electric motors in the world [2]. The energy consumption of pumps to transport energy into rooms is responsible for a significant portion of HVAC energy consumption [3]. It accounts for about 40% of the total HVAC energy use [4]. These numbers demonstrate that the system with parallel-connected pumps has great potential for energy saving and there is increased attention associated with the operation of pumps. Many advanced computer control systems have been implemented in HVAC systems [5]. How to optimally control the number of running pumps and corresponding speeds is important to reduce the total energy consumption and exactly meet the demand.

Researchers have proposed various pump optimization strategies to reduce the energy consumption [6–22]. These methods play important roles and help human operators to save a lot of energy in systems with parallel-connected pumps [8,11]. However, there is still room to improve

existing methods. For example, sometimes people observe frequent adjustments of the terminal air-handling units in HVAC systems caused by the mismatch of flow rate demand [10]. Furthermore, without guarantee of optimality, pumps under control of existing algorithms still have energy saving potential. In addition, some of the pumps may be in maintenance (the interval of maintenance could be as long as three months [23]) or break down occasionally. As a result, how to adapt to such situations and continue to work in an energy saving way for pump systems might become a major challenge.

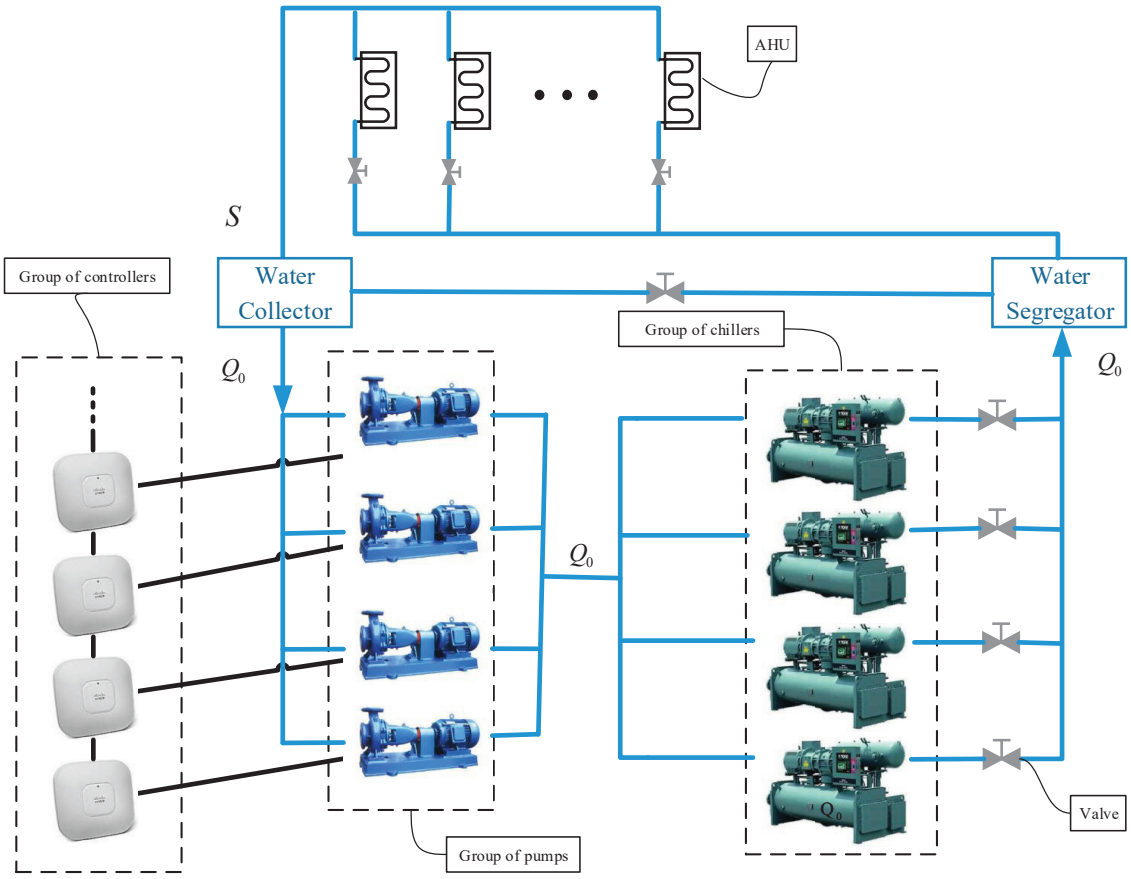


Figure 1. The distributed control framework for parallel-connected pumps.

By investigating the existing papers, we summarize them into four groups including rule based methods, mathematical programming, heuristic methods and distributed methods, as shown in Table 1. The first three groups are based on the centralized control framework, which replace the group of controllers, as shown in Figure 1, with a single controller. The last group is based on the distributed control framework, as presented in Figure 1.

Table 1. Optimization methods for the system with parallel-connected pumps.

Control Framework	Groups	Methods
Centralized	Rule based	Zhao et al. [8]; Ma et al. [9]; Gao et al. [10]
	Mathematical programming	Bonvin et al. [11]; Horváth et al. [12]; Koor et al. [13]
	Heuristic	Wang et al. [17]; Rong et al. [18]; Olszewski et al. [22]
Distributed	Distributed	Dai et al. [24]

1.1. Rule Based Methods

In the pump system, the operation still depends on human operators to manage pumps. To tackle this problem, some methods have been conducted on the adjusting pump working, including on-line adaptive control [6] and group control strategy [7]. Zhao et al. [8] defined the most unfavorable thermodynamic loop and used the variable differential pressure set point control strategy of pumps. The test results saved 47–58% water pump power consumption. Ma et al. [9] developed the pressure drop models for water networks and formulated an optimal pump sequence control strategy. The number of pumps in operation was set according to their power consumption and maintenance costs. In addition, Gao et al. [10] proposed a fault-tolerant and energy efficient sequence control strategy (SC) for chilled water pump systems with a differential pressure set-point. The strategy ensured that the water flow of the secondary loop met the primary loop. There was a series of operational problems for the terminal air-handling units and chillers due to the ungratified flow rate demand. Rule based methods are simple and show great performance to meet the demand. The accurate model of pumps is not needed in [8,10]. These methods are widely used in practical engineering. Sometimes pumps cannot work in the most energy efficient way because of the non-optimal working point.

1.2. Mathematical Programming Methods

Currently, many experts have made great efforts on developing and applying methods for pumps working in parallel to improve their energy efficiencies. Many typical research papers were reported by focusing on mathematical programming methods. Bonvin et al. [11] introduced a convex mathematical program for pump scheduling in water networks. It realized an average 3.0% optimality gap and significant saving according to operation costs and energy consumption. Horváth et al. [12] proposed a convex pump model based on pump curves and developed a mixed-integer optimization method. The applicability and results showed that the method was more energy efficient compared with current methods. The errors of the solution and the computation time were influenced by the help variables. In order to keep pumps running close to the best efficiency point, Koor et al. [13] applied the customized optimization software, which adopted the Levenberg–Marquardt algorithm to solve the complex optimization task. By analyzing three different theoretical scenarios, the optimal number of working pumps was estimated. The author in [14] studied the problem of energy consumption minimization for variable speed drive pumps and proposed a simplified convex relaxation of pump characteristics. Two small benchmark networks showed that the method was suitable for pump operations with lower energy costs. To develop an energy-oriented optimal control solution, Jepsen et al. [15] recommended a method to minimize the energy consumption of pump stations by two steps: First, the energy consumption for a pre-selected set of active pumps were formulated as a convex optimization problem, then the problem of choosing the number of active pumps was solved by a convex solver. The proposed solution presumed the performance curves remaining constant over time. Inspired by [15], we adopt the sequential least squares programming algorithm (SLSQP) to compare with our method. The sequential least squares programming algorithm has better computing efficiency [16]. These mentioned mathematical programming algorithms increase the ability to obtain the near-optimal solutions. Due to the equality constraints, there are some deviations to meet the demand. These methods are suitable only for specific systems and lack flexibility.

1.3. Heuristic Methods

Considering the complexity of the problem, some researchers have turned their attention to heuristic optimization methods. By adopting the genetic algorithm, Wang et al. [17] proposed a bi-objective pump scheduling approach. Rong et al. [18] presented a crowding genetic algorithm to decide optimal pump configuration. Cimorelli et al. [19] studied the optimal regulation of pump systems by the genetic algorithm. In addition, Barán et al. [20] introduced a multi-objective evolutionary algorithm to solve the optimal pump scheduling problem with four objectives.

Hashemi et al. [21] introduced an ant-colony optimization method to improve the optimization process and reduced the feasible solutions in the search space. The optimization process was repeated many times leading to different costs and the lowest cost was selected. In order to enhance energy conservation and analyze control strategies, Olszewski et al. [22] studied three different strategies for complex pumping stations with a genetic algorithm (GA) and pointed out that minimizing power consumption was the most energy efficient. The ungratified flow rate was affected by the population size. The heuristic optimization methods are powerful tools to solve the optimization problem without any derivatives and continuities assumption. The above methods adopt the advantage of population sampling to solve these combination problems. However, these methods lack convergence guarantee and cannot be used to assess the minimum energy. Under equality constraints, these methods are hard to find a feasible solution. If the system with parallel-connected pumps changes, they need to do the algorithm again.

1.4. Distributed Methods

Recently, we proposed a new building control architecture Insect Intelligent Building (I^2B) [25]. This new distributed control framework is convenient for pump units. It is open, flexible and has a low configuration cost. There are some early attempts to carry out the distributed control in HVAC systems. Liu et al. [26] developed a decentralized optimization algorithm for optimal set point control for HVAC systems. Dai et al. [24] introduced a novel decentralized transfer algorithm (TA) as a solution to a typical control problem for parallel-connected pumps in HVAC systems. In fact, the idea of distributed control or optimization has been applied widely in other large engineering systems such as the distributed multi-area economic dispatch for active distribution networks [27], or multi-agent traffic light control systems [28].

The above literature review shows that there is great potential in reducing pump energy consumption. Advanced control strategies are effective ways to achieve high energy efficiency. There are three types of methods based on centralized control structures: rule based methods, mathematical programming and heuristic methods. There have also been some distributed methods in recent years that show better flexibility. Dai et al.'s work [24] is an initial attempt to solve the optimization problem for parallel-connected pumps in a distributed manner. We observe that existing studies can be further improved. First, there are some violations of demand constraints. So, strictly speaking, the solutions are not feasible. Second, existing algorithms have no guarantee of convergence. The solution may not be optimal and extra energy is consumed.

In this paper, we consider a distributed pump system, as presented in Figure 1, and extend our previous work [29]. Compared with our previous work, the main differences can be summarized as follows: First, our work considers the controllers connected in the more general topologies and not limited in the chain topology. Second, our work establishes the theoretical convergence results. Motivated by the distributed and peer-to-peer algorithm framework proposed in [25], we put forward a novel distributed optimization method that is suitable for general topologies and establishes theoretical convergence results. Our method consists of two parts: First, in order to process the network information, we apply the breadth first search algorithm to construct a tree to exchange messages. Second, all nodes coordinate with each other and randomly sample a population of speed ratios. In addition, we prove that our method has the guarantee of convergence. Based on a numerical example of pump systems, we compare our approach with other methods. It turns out that our method is able to exactly satisfy the actual demand, minimize energy consumption and has good flexibility. The hardware testing shows that our method can perform on low-cost Raspberry Pi3 and reduce the cost when applied in practical systems.

The remaining part of the paper is organized as follows. In Section 2, we define the speed ratio optimization problem for parallel pumps. In Section 3, we present a novel distributed algorithm for the problem. In Section 4, we prove the convergence of the proposed algorithm. To demonstrate the advantages of the new algorithm, in Section 5, we show numerical results of the algorithm and

compare it with the state of art algorithms. We also study the flexibility of the algorithm and test the algorithm on a hardware platform. In Section 6, we conclude the paper.

2. Problem Formulation

Consider the control of parallel-connected pumps in an HAVC system, as shown in Figure 1. We assume that there are m centrifugal variable-speed pumps connected in parallel and the size of pumps may be different. The demand is a stable value during a period of time (for example, one hour, as given in paper [9]). Our problem is to optimize the number of running pumps and their speed set-points to meet the demand with minimum energy consumption. We also assume that underlying controllers, such as those given in [2], can regulate pump speed to the new set-points in real time.

2.1. Pump Models

Let us first formulate the relationship between pump speed and pump energy consumption. There are various methods to establish the pump model in application, such as dynamic pump model [30], pump approximation model [22]. In order to obtain better precision and convenient calculation, we choose polynomial functions to reflect pump characteristics. It is vital to know pump characteristics to implement optimal pump control. To describe a pump, we need to consider relationships between pump head, flow rate, and efficiency. When pumps work at their rated speeds, the relationships can be indicated by the equations as below.

$$H_i = a_{P1,i}Q_i^2 + a_{P2,i}Q_i + a_{P3,i} \tag{1}$$

$$\eta_i = b_{P1,i}Q_i^2 + b_{P2,i}Q_i + b_{P3,i} \tag{2}$$

$$P_i = \frac{\rho g Q_i H_i}{1000\eta_i \eta_{M,i} \eta_{VFD,i}} \tag{3}$$

We use H_i to denote the head of pump $i, i = 1, 2, \ldots, m$ where m is the total number of pumps. Q_i is the flow rate of pump i. η_i is the mechanical efficiency, P_i is the power consumption of pump i. P_i depends on three sub-efficiency variables, including the mechanical efficiency η_i, the motor efficiency $\eta_{M,i}$, the variable frequency efficiency $\eta_{VFD,i}$. The $a_{P1,i}, a_{P2,i}, a_{P3,i}, b_{P1,i}, b_{P2,i}, b_{P3,i}$ are pump performance parameters. These parameters in the equations may be accessed by the pump performance curves or actual measurement data. The data is measured when pumps work at the rated speed. If we obtain the data, it is possible to utilize the least square method to fit the polynomial functions.

We use n_i to denote the actual speed of pump i. $n_{0,i}$ is the rated speed of pump i. We define speed ratio ω_i to depict the ratio between the actual speed and the rated speed.

$$\omega_i = \frac{n_i}{n_{0,i}} \tag{4}$$

In general, the pump performance parameters are related to the speed and the operating conditions. When pump works in variable speed status and the operating conditions do not change, we can attain a series of pump characteristic curves through the pump affinity laws [31], as presented in Equation (5).

$$\begin{aligned} Q_i(n_{0,i}) &= \frac{Q_i(n_i)}{\omega_i} \\ H_i(n_{0,i}) &= \frac{H_i(n_i)}{\omega_i^2} \\ \eta_i(n_{0,i}) &= \eta_i(n_i) \end{aligned} \tag{5}$$

where $Q_i(n_{0,i})$ is the rated flow rate at rated speed n_0, $Q_i(n_i)$ is the actual flow rate at actual speed n_i. $H_i(n_{0,i})$ is the rated head at rated speed n_0. $H_i(n_i)$ is the actual head at actual speed n_i. $\eta_i(n_{0,i})$

is the rated mechanical efficiency at rated speed n_0. Using the pump affinity laws and Equations (1) and (2), we can reach the equations of pump head and flow rate, efficiency and flow rate at a speed n_i including rated and non-rated, which are as below.

$$H_i = a_{P1,i}{Q_i}^2 + a_{P2,i}\omega_i Q_i + a_{P3,i}{\omega_i}^2 \tag{6}$$

$$\eta_i = b_{P1,i}(\frac{Q_i}{\omega_i})^2 + b_{P2,i}(\frac{Q_i}{\omega_i}) + b_{P3,i} \tag{7}$$

Although paper [32] indicated that $\eta_{M,i}, \eta_{VFD,i}$ depended on ω_i, when the speed ratios meet $0.4 \leq \omega_i \leq 1$, $\eta_{M,i}, \eta_{VFD,i}$ could supposedly be constants. The reason is that ω_i has little influence on them. Therefore, we can dispense with the impact of $\eta_{M,i}, \eta_{VFD,i}$ and only consider the mechanical efficiency η_i. As a result, the pump power consumption equation can be rewritten as below.

$$P_i = \frac{\rho g Q_i H_i}{1000\eta_i} \tag{8}$$

When the demand is given, in Figure 1 the total demand pressure set point H_0 and the total demand flow rate Q_0 meet the resistance characteristic equation of the pipe net as follows [33]:

$$H_0 = SQ_0^2 \tag{9}$$

where S is the resistance of the pipe net. This equation makes sure that if the total flow rate on the main pipe is Q_0 and S does not change, it can keep the total pressure equal to the demand pressure set point H_0. Once S changes, it leads to a different demand.

2.2. Optimization Problem Formulation

Now let us formulate the pump speed ratio optimization problem (PSROP).

Problem 1. *Pump Speed Ratio Optimization Problem (PSROP).*

$$\min_{\omega_i, i\in\{1,2,\ldots m\}} \sum_{i=1}^{m} P_i(\omega_i) \tag{10}$$

$$s.t. \quad H_0 = H_i \tag{11}$$

$$Q_0 = \sum_{i=1}^{m} Q_i \tag{12}$$

$$\omega^- \leq \omega_i \leq 1 \text{ or } \omega_i = 0, \quad i \in \{1, 2, \ldots, m\} \tag{13}$$

The objective function presents that the optimization objective of this paper is to minimize the total power consumption of pumps in parallel. The objective function indicates that we should try to find a combination of ω_i to content the corresponding constraints at the lowest energy consumption. Constraint (11) is the equality relationship between the head of the pump and the demand total pressure H_0. Constraint (12) is to guarantee that the amount of flow rate is equal to the total demand flow rate Q_0 on the main pipe. Constraint (13) is the range of the speed ratio. ω^- is the lower limitation of the speed ratio. If the speed ratio is equal to 0, it means that the pump i is shut down.

3. Proposed Algorithm

In this section, we propose a novel algorithm, breadth first search random sampling (BRS for short), to solve the optimization problem PSROPred (Pump Speed Ratio Optimization Problem). Our algorithm includes two steps.

First, in order to process the network information, we apply the breadth first search algorithm (BFS) [34] to construct a spanning tree supporting the communication in the pump intelligent node network. For details of the BFS algorithm, see Appendix A.1 in Appendix A.

Second, based on the tree, we randomly sample the speed ratios. All the nodes coordinate with each other, carry out parallel calculations to optimize the speed ratio of each node so that the total energy consumption is minimized. Before introducing the algorithm in detail, we make an assumption as follows:

Assumption 1. *In our distributed pump control system, every pump is an intelligent node equipped with a controller. The pump model and an identical computing program have been stored in the node (controller) for the pump. The node can communicate with neighbors and do calculations. The network of the pump intelligent nodes forms a connected graph.*

With this assumption, if node i and node j are not neighbors, there is at least one path between them.

Figure 2 gives an example of constructing a BFS tree for a network with six nodes (controllers for six pumps connected in parallel). From the root node 1, the search message indicated by red links gradually spreads to leaf nodes. Based on the BFS tree, every node performs an algorithm enabling coordinations with its neighbors.

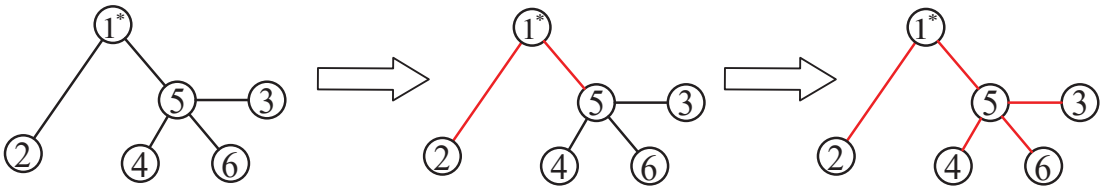


Figure 2. The process of constructing a breadth first search algorithm (BFS) tree from the root node marked with the star.

Our algorithm is formally defined as Algorithm 1. Algorithm 1 uses the BFS tree generated by the BFS algorithm defined in Appendix A.1 in Appendix A as input and calls the Algorithm defined in Appendix A.2 in Appendix A to do the message processing. A message processing example is shown in Figure 3 based on the BFS tree given in Figure 2. The messaging mechanism between nodes can be described as follows. Every node has six input message queues. Each message queue records the message. Once the message is read, it will be deleted from the queue. The communication between nodes is synchronous. During the k-th round, each node sends messages to its neighbors and reads from its input message queues. As a result, the messages of all child nodes are collected.

Below is a description of the main procedure in Algorithm 1. Every node obtains its node type in the BFS tree (as root node, intermediate node, or leaf node). Then every node initializes its pump performance parameters, the range of speed ratio, the sample adjustment parameter τ, the child nodes set $\mathcal{N}_i$ and the total iteration parameter K.

Every leaf node independently generates speed ratio samples following the uniform distribution over the interval $[\omega^- - \tau, 1]$ with $\omega_{i,k}$ as the k-th sample of node i. If the speed ratio sample $\omega_{i,k} \geq \omega^-$, it will evaluate corresponding $Q_{i,k}, P_{i,k}$. If not, the node will set the sample to 0, which means the pump will be shut down. Then the node will transmit the $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$ to its parent node. If it does not receive end instruction, the leaf node will continue to do the above operation. When the leaf node receives end instruction from the parent, it will stop generating samples, and acquire its optimal speed ratio. The intermediate node performs the same operation that generates samples. The difference is that the intermediate node will calculate the partial summation flow rate $Q_{T,i,k}$ and the partial summation power consumption $P_{T,i,k}$ over the subtree of node i basing on the Q_{T,j,k_j}, P_{T,j,k_j} provided by its child nodes $\mathcal{N}_i$. The Q_{T,j,k_j}, P_{T,j,k_j} is read from the input message queue. Next, if the

partial summation flow rate $Q_{T,i,k} \leq Q_0$, the intermediate node will transmit $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$ to its parent node. If not, the node will delete the $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$. As for the root node, it will compute its $Q_{i,k}$ through the total flow rate Q_0 and the Q_{T,j,k_j} provided by its child nodes. This operation makes sure that the constraint is satisfied. In view of the Q_i, the root node utilizes the Equations (6)–(8) and H_0 to compute $\omega_{i,k}$. When $\omega_{i,k}$ satisfies the speed ratio range constraint, it will calculate the total energy consumption $P_{T,i,k}$. P_k^* is the currently optimal total energy consumption after k rounds selection of speed ratio samples. $\widetilde{\boldsymbol{\omega}}_k^*$ denotes the optimal estimation combination of speed ratio samples in the k-th round for the root node. In addition, if $P_{T,i,k} \leq P_k^*$, the root node will update P_k^* and $\widetilde{\boldsymbol{\omega}}_k^*$ to save the currently optimal estimation combination of speed ratios for the whole network. By doing so, after K rounds the root node supposes that the best combination of speed ratios is achieved, and therefore ends the entire algorithm and sends the best combination of speed ratios $\widetilde{\boldsymbol{\omega}}_k^*$ to child nodes. Once the node receives the end message and $\widetilde{\boldsymbol{\omega}}_k^*$, it will stop calculation and send them to child nodes.

From the steps of Algorithm 1, we can observe that the BRS algorithm makes sure the solution exactly meets the demand (the total flow rate constraint). Every node maintains its own speed ratio $\widetilde{\omega}_i^*$. The whole adjustment is an iterative optimization process with the cooperation of each node. Constraints are satisfied in the entire iteration process. Furthermore, it has the flexibility due to the distributed control structure. We will prove that the total energy consumption can be minimized.

Algorithm 1: Breadth first search random sampling algorithm (BRS)

Input: the BFS tree, the pump parameters $a_{P1,i}, a_{P2,i}, a_{P3,i}, b_{P1,i}, b_{P2,i}, b_{P3,i}$, the total flow rate Q_0, the head H_0, sample adjustment parameter τ, the child nodes set $\mathcal{N}_i$, the range of speed ratio ω^- and K ;

Output: every intelligent node's optimal estimation of speed ratio $\widetilde{\omega}_i^*$;

if *node is leaf node or intermediate node* **then**
- Initialize $k = 0, Q_{T,i,k-1} = [], P_{T,i,k-1} = [], \boldsymbol{\omega}_{T,i,k-1} = []$;
- **while** *does not receive end instruction* **do**
  - Sample $\omega_{i,k}$ and $\omega_{i,k} \sim U[\omega^- - \tau, 1]$;
  - **if** $\omega_{i,k} \geq \omega^-$ **then**
    - Use Equations (6)–(8) and H_0 to compute $Q_{i,k}, P_{i,k}$;
  - **else**
    - $\omega_{i,k} = 0, Q_{i,k} = 0, P_{i,k} = 0$;
  - Do ***Algorithm A.2*** and get $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$;
  - **if** $Q_{T,i,k} \geq Q_0$ **then**
    - Delete $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$;
  - $k = k + 1$;
- Send end instruction and $\widetilde{\boldsymbol{\omega}}_k^*$ to child nodes ;
- Goto final ;

if *node is root node* **then**
- Initialize $k = 0$, $P_k^* = \infty$, $\widetilde{\boldsymbol{\omega}}_k^* = []$, $Q_{T,i,k-1} = []$, $P_{T,i,k-1} = []$, $\boldsymbol{\omega}_{T,i,k-1} = []$;
- **while** $k < K$ **do**
  - $Q_{i,k} = [], P_{i,k} = [], \omega_{i,k} = []$;
  - Do ***Algorithm A.2*** and get $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$;
  - $Q_{i,k} = Q_0 - Q_{T,i,k}$;
  - Use Equations (6)–(8) and H_0 to compute $\omega_{i,k}, \boldsymbol{\omega}_{T,i,k}, P_{T,i,k}$;
  - **if** $P_{T,i,k} < P_k^*$ **then**
    - $\widetilde{\boldsymbol{\omega}}_k^* = \boldsymbol{\omega}_{T,i,k}, P_k^* = P_{T,i,k}$;
  - $k = k + 1$;
- Send end instruction and $\widetilde{\boldsymbol{\omega}}_k^*$ to child nodes ;
- Goto final ;

final ;

return its speed ratio $\widetilde{\omega}_i^*$;

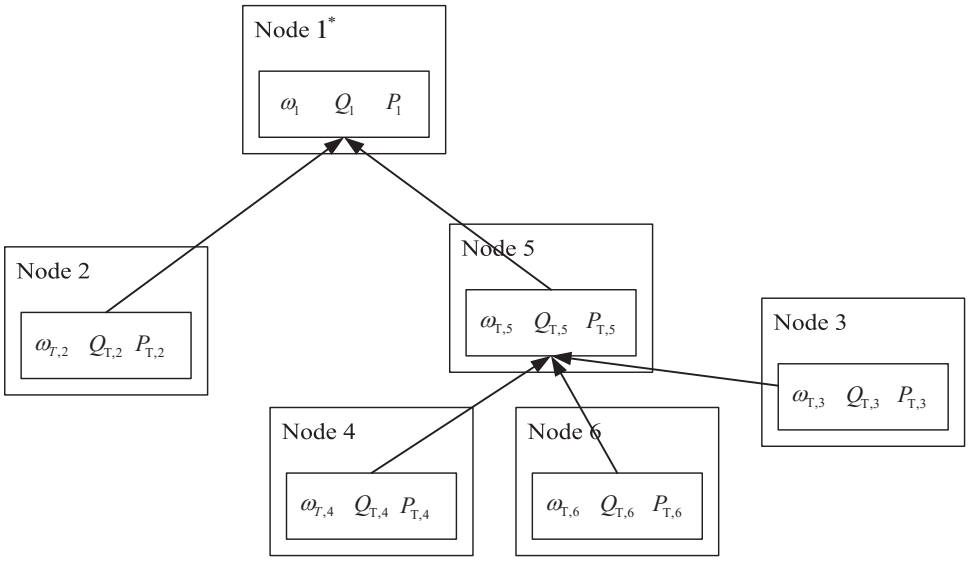


Figure 3. The message processing of the variables among intelligent nodes.

4. Theoretical Analysis of the Proposed Algorithm

We carry out a theoretical analysis of the convergence for the proposed BRS algorithm. According to the definition of the algorithm, the sample process at node i (for pump i) of speed ratio can be viewed as an i.i.d. stochastic process $\{\omega_{i,k}, k = 1, 2, ...\}$ where $\omega_{i,k}$ is the k-th sample of the speed ratio of pump i. The speed ratio sample of pump i follows the uniform distribution

$$U[\omega^- - \tau, 1] \tag{14}$$

with $\tau > 0$ as a given parameter such that $\omega^- > \tau$.

Assumption 2. *Both the energy consumption function $P_i(\omega_i)$ and flow rate function $Q_i(\omega_i)$ of a pump are continuous functions on the domain $\{0\} \cup [\omega^- - \tau, 1]$ with $P_i(0) = Q_i(0) = 0$. The flow rate function $Q_i(\omega_i)$ of a pump is also a monotonously increasing function.*

Now we are ready to show the main theoretical result: the speed ratios of the m pumps output by our BRS algorithm will converge to the optimal solution of the PSROP problem (defined in Section 2.2 by Equations (10)–(13) if K goes to infinity. According to the steps of the BRS algorithm, under Assumption 1, all pump nodes are divided into three types on the spanning tree T. Denote $\mathcal{I}$ as the set of intermediate nodes, $\mathcal{L}$ as the set of leaf nodes, $\mathcal{R} = \{r\}$ as the set of a single root node. Then we can rewrite PSROP in an equivalent form.

Problem 2. *Alternative Pump Speed Ratio Optimization Problem (PSROP')*

$$\min_{\omega_i, i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}} \sum_{i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}} P_i(\omega_i) \tag{15}$$

$$s.t. \quad H_0 = H_i \quad i \in \mathcal{L} \cup \mathcal{I} \cup \{r\} \tag{16}$$

$$Q_0 = \sum_{i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}} Q_i \tag{17}$$

$$\omega^- \leq \omega_i \leq 1 \text{ or } \omega_i = 0, \quad i \in \mathcal{L} \cup \mathcal{I} \cup \{r\} \tag{18}$$

It is trivial that the optimal value of PSROP' is the same as that of PSROP since both objective functions and constraints are the same when we re-order the speed ratios. Below, in order to present our analysis, we define the optimal pump speed ratios according to node types of the spanning tree T as follows

$$\boldsymbol{\omega}_T^* = [\omega_i^*, i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}] \tag{19}$$

Similarly, we denote

$$\boldsymbol{\omega}_T = [\omega_i, i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}] \tag{20}$$

as the combination list of decision variables for the PSROP' problem and

$$\widetilde{\boldsymbol{\omega}}_{T,K}^* = [\widetilde{\omega}_{i,K}^*, i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}] \tag{21}$$

as the speed ratios of the m pumps output by our BRS algorithm after K rounds to emphasis the role of node type on the tree T. We note that due to the equality constraint Equation (17), the optimal speed ratio ω_r^* of the root node r can be written as a function of speed ratios of other nodes. In fact, we have (due to the monotone property of flow rate function by Assumption 2)

$$\omega_r^* = Q_r^{-1}(Q_0 - \sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_i^*)) \tag{22}$$

Similarly, we have

$$\widetilde{\omega}_{r,K}^* = Q_r^{-1}(Q_0 - \sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\widetilde{\omega}_{i,K}^*)) \tag{23}$$

according to the step in Line 20 of Algorithm 1 running at the root node.

Assumption 3. *$\omega_r^* \in (\omega^-, 1)$ and $\sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_i^*) > 0$.*

Theorem 1. *Suppose Assumptions 1, 2, 3 are satisfied. Let $\varepsilon > 0$ be any given positive number, then*

$$\lim_{K \to \infty} Pr\{|P(\widetilde{\boldsymbol{\omega}}_{T,K}^*) - P(\boldsymbol{\omega}_T^*)| < \varepsilon\} = 1 \tag{24}$$

where $\widetilde{\boldsymbol{\omega}}_{T,K}^* = [\widetilde{\omega}_{i,K}^*, i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}]$ are speed ratios output by the algorithm after K rounds.

Proof of Theorem 1. From Assumption 3, we know that $\omega_r^* \in (\omega^-, 1)$ and $\sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_i^*) > 0$. In this case, we know that $Q_r(\omega_r^*) > 0$ and $\exists i_0 \in \mathcal{L} \cup \mathcal{I}$ such that

$$Q_{i_0}(\omega_i^*) > 0 \tag{25}$$

We must have $Q_r(\omega_r^*) \in [Q_r(\omega^-), Q_0)$ and $Q_{i_0}(\omega_{i_0}^*) \in (0, Q_0)$

Since both Q_r and P_r are continuous (by Assumption 2), for any $\varepsilon_1 > 0$, there exists a $\delta_r(\varepsilon_1) \in (0, 1 - \omega^-)$ such that $\forall \omega_r \in [\omega_r^-(\varepsilon_1), \omega_r^*]$, where $\omega_r^-(\varepsilon_1) = \omega_r^* - \delta_r(\varepsilon_1)$. It is true that

$$Q_r(\omega_r) \in (Q_r(\omega_r^*) - \varepsilon_1, Q_r(\omega_r^*)] \tag{26}$$

$$P_r(\omega_r) \in (P_r(\omega_r^*) - \varepsilon_1, P_r(\omega_r^*) + \varepsilon_1) \tag{27}$$

Similarly, since both Q_i and P_i are continuous (by Assumption 2), when $Q_i(\omega_i^*) > 0$(then $\omega_i^* \in [\omega^-, 1]$), for any $\varepsilon_2 > 0$, there exists a $\delta_i(\varepsilon_2) \in (0, 1 - \omega^-)$ such that $\forall \omega_i \in \boldsymbol{B_i} = [\omega_i^*, \omega_i^+(\varepsilon_2)]$,

$$\boldsymbol{B_i} = [\omega_i^*, \omega_i^+(\varepsilon_2)] \tag{28}$$

and $\omega_i^+(\varepsilon_2) = \omega_i^* + \delta_i(\varepsilon_2)$, it is true that

$$Q(\omega_i) \in [Q_i(\omega_i^*), Q_i(\omega_i^*) + \varepsilon_2) \tag{29}$$

$$P_i(\omega_i) \in (P_i(\omega_i^*) - \varepsilon_2, P_i(\omega_i^*) + \varepsilon_2) \tag{30}$$

When $\omega_i^* = 0$, then $Q_i(\omega_i^*) = 0$, we set $\boldsymbol{B}_i = \{0\}$. For this case, $\forall \omega_i \in \boldsymbol{B}_i$, Equations (29) and (30) still hold.

Since the optimal solution is feasible, we have

$$Q_0 = Q_r(\omega_r^*) + \sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_i^*) \tag{31}$$

For a given $\varepsilon_1 > 0$ and corresponding $\delta_r(\varepsilon_1)$, we choose ε_2 in Equations (29) and (30) such that

$$(m-1)\varepsilon_2 \leq Q_r(\omega_r^*) - Q_r(\omega_r^-(\varepsilon_1)) \tag{32}$$

or equivalently

$$\sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_i^*) + (m-1)\varepsilon_2 \leq Q_0 - Q_r(\omega_r^-(\varepsilon_1)) \tag{33}$$

For $(\omega_i, i \in \mathcal{L} \cup \mathcal{I})$ satisfying $\cap_{i \in \mathcal{L} \cup \mathcal{I}} B_i$, we have

$$\omega_r = Q_r^{-1}(Q_0 - \sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_i)) \tag{34}$$

$$\omega_r = Q_r^{-1}(Q_r(\omega_r^*) + \sum_{i \in \mathcal{L} \cup \mathcal{I}} (Q_i(\omega_i^*) - Q_i(\omega_i))) \geq Q_r^{-1}(Q_r(\omega_r^*) - (m-1)\varepsilon_2) \tag{35}$$

$$\omega_r \geq Q_r^{-1}(Q_r(\omega_r^-(\varepsilon_1))) = \omega_r^-(\varepsilon_1) \tag{36}$$

From the monotone property of Q_r^{-1} (based on the monotone property of flow rate function by Assumption 2), we also have

$$\omega_r \leq Q_r^{-1}(Q_r(\omega_r^*)) = \omega_r^* \tag{37}$$

So

$$\omega_r \in [\omega_r^-(\varepsilon_1), \omega_r^*] \tag{38}$$

From Equation (30) and $P_i(\omega_i)$ is set to 0 if $\omega_i = 0$, we know that

$$\sum_{i \in \mathcal{L} \cup \mathcal{I}} |P_i(\omega_i) - P_i(\omega_i^*)| < (m-1)\varepsilon_2 \tag{39}$$

Due to Equation (28), for ω_r in Equation (34) we have

$$P_r(\omega_r) \in (P_r(\omega_r^*) - \varepsilon_1, P_r(\omega_r^*) + \varepsilon_1) \tag{40}$$

In sum, we have

$$|\sum_{i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}} P_i(\omega_i) - P(\boldsymbol{\omega}^*)| = |\sum_{i \in \mathcal{L} \cup \mathcal{I} \cup \{r\}} (P_i(\omega_i) - P(\omega_i^*))| < Q_r(\omega_r^*) - Q_r(\omega_r^-(\varepsilon_1)) + \varepsilon_1 \tag{41}$$

For given $\varepsilon > 0$, we can choose ε_1 such that

$$Q_r(\omega_r^*) - Q_r(\omega_r^-(\varepsilon_1)) + \varepsilon_1 < \varepsilon \tag{42}$$

For nodes $i \in \mathcal{L} \cup \mathcal{I}$, define $\boldsymbol{A}_{i,k}$ as the event that the kth sample $\omega_{i,k}$ generated at node i belongs to the interval $\boldsymbol{B}_i$ defined in Equation (28) with the understanding that $\boldsymbol{B}_i = \{0\}$ if $\omega_{i,k} = 0$ (corresponding to the case $u_{i,k} \in [\omega^- - \tau, \omega^-)$. According to the algorithm design, we have

$$Pr\{\boldsymbol{A}_{i,k}\} = Pr\{\omega_{i,k} \in \boldsymbol{B}_i\} = \frac{\omega_i^+ - \omega_i^*}{1 - (\omega^- - \tau)} \tag{43}$$

if $Q_i(\omega_i^*) > 0$, or

$$Pr\{\boldsymbol{A}_{i,k}\} = Pr\{\omega_{i,k} \in \boldsymbol{B}_i\} = Pr\{u_{i,k} \in [\omega^- - \tau, \omega^-)\} = \frac{\tau}{1-(\omega^- - \tau)} \tag{44}$$

if $Q_i(\omega_i^*) = 0$, it means the pump will be shut down. Since the samples generated at different nodes in $\mathcal{L} \cup \mathcal{I}$ are independent, we have

$$Pr\{\bigcap_{i \in \mathcal{L} \cup \mathcal{I}} \boldsymbol{A}_{i,k}\} = \prod_{i \in \mathcal{L} \cup \mathcal{I}} Pr\{\boldsymbol{A}_{i,k}\} \geq [\frac{d}{1-(\omega^- - \tau)}]^{m-1} \tag{45}$$

where d is a constant no larger than τ and $(\omega_i^+ - \omega_i^*)$ for nodes $i \in \mathcal{L} \cup \mathcal{I}$ such that $Q_i(\omega_i^*) > 0$.

From Equation (33), we know that if $\omega_{i,k} \in \boldsymbol{B}_i$, $\forall i \in \mathcal{L} \cup \mathcal{I}$, it is true that

$$\sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_{i,k}) < Q_0 \tag{46}$$

As a result, $\boldsymbol{\omega}_{T,k} = [\omega_{i,k}, i \in \mathcal{L} \cup \mathcal{I}; \omega_{r,k}]$ where

$$\omega_{r,k} = Q_r^{-1}(Q_0 - \sum_{i \in \mathcal{L} \cup \mathcal{I}} Q_i(\omega_{i,k})) \tag{47}$$

forms a feasible solution with total energy consumption as

$$P(\boldsymbol{\omega}_{T,k}) = \sum_{i \in \mathcal{L} \cup \mathcal{I} \cup r} P_i(\omega_{i,k}) \tag{48}$$

From the selection of $\widetilde{\boldsymbol{\omega}}_{T,k}^*$ as the feasible solution with the lowest total energy consumption for the first k-th rounds, we have

$$P(\widetilde{\boldsymbol{\omega}}_{T,k}^*) \leq P(\boldsymbol{\omega}_{T,k}) \tag{49}$$

From the optimality of $\boldsymbol{\omega}^*$ and Equation (41), we have

$$P(\boldsymbol{\omega}^*) \leq P(\widetilde{\boldsymbol{\omega}}_{T,k}^*) \leq P(\boldsymbol{\omega}_{T,k}) \leq P(\boldsymbol{\omega}^*) + \varepsilon \tag{50}$$

The theorem is proved. □

As a result, the BRS algorithm can not only exactly meet the constraints and have great flexibility but also has the guarantee of convergence. By applying the BRS algorithm, the solution $\widetilde{\boldsymbol{\omega}}_{T,k}^*$ converges to the optimal speed ratio $\boldsymbol{\omega}^*$ in probability. Hence, under Assumption 1, BRS algorithm is easy to realize and can converge to the optimal solution.

Remark 1. *In reality, we cannot judge whether Assumption 3 holds in advance. To make sure of the convergence to the global true optimal solution, we can run the algorithm with each node as the root node in turn and compare the final results.*

5. Results and Discussion

In this section, we will verify the properties of the proposed BRS algorithm including the demand satisfaction, minimum energy consumption, the flexibility in adapting to changing the topology of pump systems and the low-cost hardware test. We compare BRS with other methods in demand satisfaction and energy consumption. This section is organized as follows: First, we introduce the study cases, the simulation environment and compared algorithms. Second, we study the violation of constraint satisfaction among BRS and the compared algorithms. Third, we compare the performance among BRS and other algorithms. Then, we study the flexibility of BRS under a different number of intelligent nodes. Finally, we test BRS based on low-cost hardware.

5.1. Set up of Study Cases

We adopt the performance parameters of pumps that are introduced in paper [35]. Table 2 shows the specific parameters and performance curves of the pumps that are presented in Figure 4. The parameters are collected from the RXL (Hot cold water vertical) type pumps.

Table 2. Performance parameters of pumps.

Pump	Performance Parameters					
	$a_{P1,i}$	$a_{P2,i}$	$a_{P3,i}$	$b_{P1,i}$	$b_{P2,i}$	$b_{P3,i}$
PUMP-A	−0.0046	0.0696	60.271	−0.0002	0.0254	0.0616
PUMP-B	−0.0112	0.1358	54.841	−0.0005	0.0316	0.2582

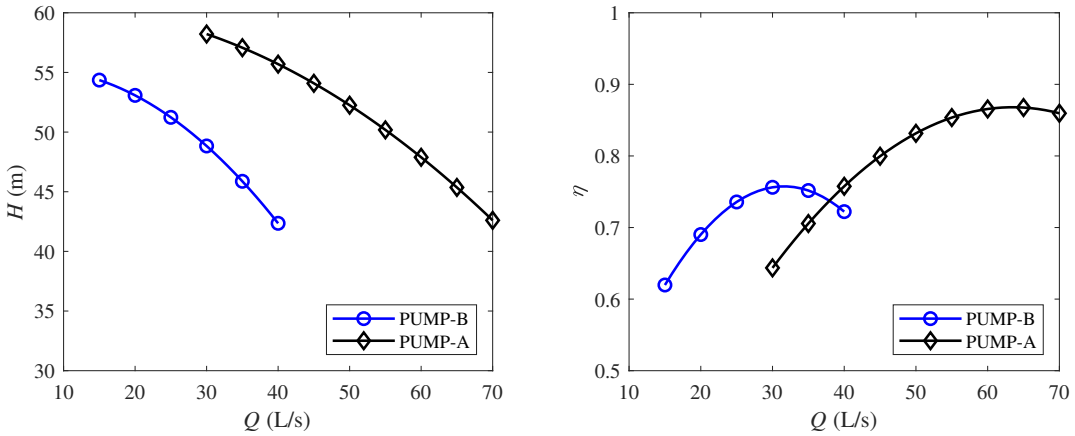


Figure 4. The performance curves of PUMP-A and PUMP-B.

Here, we establish an HVAC system including six pumps in parallel. Four of the pumps are PUMP-A (big pump) and two of the pumps are PUMP-B (small pump). The pump with a relatively higher capacity of flow rate is named big pump and the other one is small pump [35]. We adopt four different total flow rate and head demand cases in an HVAC system, which represent the typical stable working conditions during a period of time, as shown in Table 3. These cases are from light to heavy. In order to investigate a method in depth, the optimal working positions for parallel pumps are different under these four cases.

Table 3. Different demands of the heating, ventilation, and air-conditioning (HVAC) system.

	H_0(m)	Q_0(L/s)
Case 1	26	86
Case 2	29	117
Case 3	36	248
Case 4	39	288

5.2. Software Simulation Environment

The software simulation environment is shown in Table 4. The centralized algorithms SC, SLSQP (Sequential least squares programming algorithm) and GA are built with an open-source software Python with version 2.7 to carry out the simulation. We employ the distributed simulation platform (DSP). (DSP is a simulation engine written in Python that can simulate distributed algorithms synchronous to all nodes through UDP (User Datagram Protocol) communication.) To investigate the performance of both BRS and TA. The operating environment is Win10x64, Intel (R) Core (TM) i7-7700 @3.60GHz, DSP1.0, and computer memory is 8 GB. Based on the DSP, we can simulate a distributed

pump system to test BRS and TA algorithms under different HVAC system demands. DSP can generate different kinds of topologies. TA is designed under controllers connected in the chain topology and relaxes the constraint. If the topology changes, TA cannot obtain the violation of constraints. All nodes are connected one by one for distributed methods in Section 5.4. In Section 5.6, we adopt a tree topology. The population size and iterations are the same for GA and BRS.

Table 4. The simulation environment for different methods.

Algorithm	Control Framework	Simulation Environment
BRS	Distributed	DSP
SC	Centralized	Python
SLSQP	Centralized	Python
GA	Centralized	Python
TA	Distributed	DSP

5.3. Other Algorithms to Be Compared with BRS

The compared algorithms include centralized algorithms and distributed algorithms. We pick up three typical centralized algorithms including sequence control strategy (SC) [10], sequential least squares programming algorithm (SLSQP) [16] and genetic algorithm (GA) [22]. The distributed transfer algorithm (TA) was proposed in [24]. These algorithms are selected based on a previous literature review. They are well represented characteristics of the four class methods and show good performance. The details of these algorithms are referred to in the related literature.

5.4. The Violation of Constraint Satisfaction

Table 5 presents the calculation results for five algorithms listed in Section 5.3 and for four cases listed in Section 5.1. Δ represents the violation of constraint (12), which means that $\Delta = \sum Q_i - Q_0$. The total flow rate exceeds the actual demand if Δ is positive. The total flow rate is less than the actual demand if Δ is negative. From Table 5, Δ related to BRS and SC are all 0 in the four cases and there are no violations to meet the constraint. As can be seen, Δ for the other three algorithms are positive or negative.

The reason for the performance difference among algorithms in constraint satisfaction for pump systems is that their principles are different. Our BRS algorithm and SC deal with the equality constraint directly. The BRS algorithm uses the advantage of population sampling and the equality constraint to find feasible solutions. SC adopts the PID (Proportion Integration Differentiation) feedback control to eliminate the deviation and meet the constraint. The other three methods relax the equality constraint and adopt the penalty function. They cannot meet the constraint. Regardless of cases, Δ is distributed randomly.

Experimental results prove that most methods have some violations to meet the demand. Furthermore, data comparison of Δ confirms that our BRS algorithm can exactly satisfy the demand constraints, which is difficult for most methods.

From Table 5, we can also see that when the demand changes, BRS can automatically regulate the number of working pumps and their speed ratios. In addition, we apply BRS to a very light demand $H_0 = 20, Q_0 = 20$. The result of our BRS algorithm only opens a PUMP-B and its $\omega_5 = 0.6438, Q_5 = 20$.

Table 5. Numerical comparison for different study cases.

Case	Method	PUMP-A		PUMP-A		PUMP-A		PUMP-A		PUMP-B		PUMP-B		Δ (L/s)	P^* (KW)
		ω_1	Q_1	ω_2	Q_2	ω_3	Q_3	ω_4	Q_4	ω_5	Q_5	ω_6	Q_6		
Case 1	BRS	0.7340	43.476	0.7304	42.524	0	0	0	0	0	0	0	0	**0**	25.378
	SC	0.7199	39.623	0	0	0	0	0	0	0.9	46.377	0	0	**0**	32.970
$Q_0 = 86$	SLSQP	0.7322	42.995	0.7322	42.995	0	0	0	0	0	0	0	0	-0.01	**25.377**
	GA	0.8046	59.632	0	0	0	0	0	0	0.7528	26.342	0	0	0.026	26.902
	TA	0.7354	43.831	0	0	0	0	0	0	0.7269	21.296	0.7269	21.296	0.423	27.288
Case 2	BRS	0.8285	58.509	0.8284	58.491	0	0	0	0	0	0	0	0	**0**	**38.757**
	SC	0.9066	74.038	0	0	0	0	0	0	0.9	42.962	0	0	**0**	45.697
$Q_0 = 117$	SLSQP	0.8285	58.509	0.8285	58.509	0	0	0	0	0	0	0	0	0.018	38.764
	GA	0.7835	48.045	0.8798	68.960	0	0	0	0	0	0	0	0	-0.005	39.539
	TA	0.7790	46.901	0.7803	47.255	0	0	0	0	0.7710	23.201	0	0	0.357	39.614
Case 3	BRS	0.9051	61.208	0.9111	62.552	0.9076	61.775	0.9107	62.465	0	0	0	0	**0**	101.322
	SC	0.9	60.036	0.9	60.036	0.9046	61.098	0	0	0.9	33.415	0.9	33.415	**0**	105.609
$Q_0 = 248$	SLSQP	0.9086	61.991	0.9086	61.991	0.9086	61.991	0.9086	61.991	0	0	0	0	-0.036	**101.317**
	GA	0.9516	71.161	0.8425	45.289	0.9798	76.739	0.8780	54.792	0	0	0	0	-0.019	103.741
	TA	0.8833	56.087	0.8762	54.349	0.8762	54.349	0.8832	56.076	0.8665	27.381	0	0	0.242	102.540
Case 4	BRS	0.9264	60.082	0.9417	63.626	0.9597	67.610	0.9667	69.121	0.8972	27.561	0	0	**0**	**129.493**
	SC	0.9	53.510	0.9	53.510	0.9	53.510	0.9	53.510	1.0	44.156	0.9086	29.804	**0**	134.518
$Q_0 = 288$	SLSQP	0.9742	70.722	0.9742	70.722	0.9742	70.722	0.9742	70.722	0.8402	5.106	0	0	-0.006	131.494
	GA	0.9497	65.410	0.8780	47.457	0.9587	67.390	0.9593	67.523	0.9717	40.189	0	0	-0.031	131.937
	TA	0.9193	58.378	0.9193	58.378	0.9193	58.378	0.9120	56.568	0.9018	28.494	0.8973	27.574	-0.23	130.327

5.5. Performance Comparison

Table 5 summarizes the total energy consumption P^* for five algorithms listed in Section 5.3 and for four cases listed in Section 5.1. The bold value of P^* in each cases is the minimum energy consumption. When comparing P^* of BRS to that of SC in the four cases, it is clear that the energy consumption of BRS is the minimum to exactly meet the demand constraint. Comparing P^* among BRS and other three algorithms in the four cases, the energy consumption of BRS is the minimum in Case 2 and Case 4. The energy consumption of SLSQP is slightly lower than BRS in Case 1 and Case 3. The reason is that the solutions of SLSQP are not feasible and the total flow rate is less than the actual demand. Due to providing less flow rate, the energy consumption of SLSQP is minimum. The differences of P^* between SLSQP and BRS are 0.001 and 0.005 in Case 1 and Case 3. Table 5 shows that SC can satisfy the demand constraint. However, SC does not optimize the energy consumption and pumps work in the non-optimal points. As a result, SC consumes much more energy.

According to these findings, due to lacking the guarantee of convergence, these methods cannot work optimally in most cases. Our BRS algorithm makes sure to satisfy the demand and minimize the energy consumption of pumps. It consumes less energy in the four cases and verifies the guarantee of convergence.

From Table 5, it can be observed that there are different combinations of pumps for type PUMP-A and type PUMP-B in the four proposed cases. In Case 1, we can choose the first option to open two PUMP-A, the second option to open one PUMP-A and two PUMP-B or the third option to open one PUMP-A and one PUMP-B. BRS algorithm makes sure to satisfy the demand and have a minimum energy consumption. The combination of pumps with BRS is supposed to be optimal.

Table 5 shows that only BRS and SC can exactly meet the demand constraints. SC is wildly used in practical engineering. In order to show the energy saving potential between two algorithms, we define the energy saving rate ξ_P as follows. Where P_i^{SC} is the energy consumption of pump i under SC. P_i^{BRS} is the energy consumption of pump i under BRS.

$$\xi_{P_{SC}} = \frac{\sum P_i^{SC} - \sum P_i^{BRS}}{\sum P_i^{BRS}} \times 100\% \tag{51}$$

Table 6 summarizes the energy save rate of pumps in the four cases under different algorithms, clearly showing that BRS is more energy efficient than SC. During the four work cases, the maximum energy saving of BRS is about 29.92%, which indicates that BRS can save more energy in practical engineering. This energy saving ability reflects that BRS has a great ability to find good enough solutions.

Table 6. The energy saving rate for different cases.

	Case 1	**Case 2**	**Case 3**	**Case 4**
$\xi_{P_{SC}}$	29.92	17.91	4.23	3.89

The theoretical analysis shows that the speed of convergence for the BRS algorithm is exponential with the K increasing. The iteration process of P^* in Figure 5 shows that the BRS algorithm may obtain a good enough solution in the first 20 iterations. While the demand changes, it almost has little influence on the speed to find a solution. Every sample communication costs time t_p. The population size is l. For a node i in the system, the time complexity generated by communication of K rounds is $O(t_p lK)$. In our BRS algorithm, the node generates the sample and calculates the flow rate and the power consumption. Then the node calculates its partial summation flow rate and the partial summation power consumption based on child node messages. Let c be the largest number of children nodes. The count of executions of multiplication is $8lK$ and the count of executions of addition is $(5 + 2c)lK$. The time complexity is $O(clK)$. The total time complexity is $O((c + t_p)lK)$. The running time of BRS under the four cases is within three seconds. Compared with the period

of demand changing, BRS running time is quite small. This analysis further presents that the BRS algorithm has the guarantee of convergence and is more fitted to practical engineering.

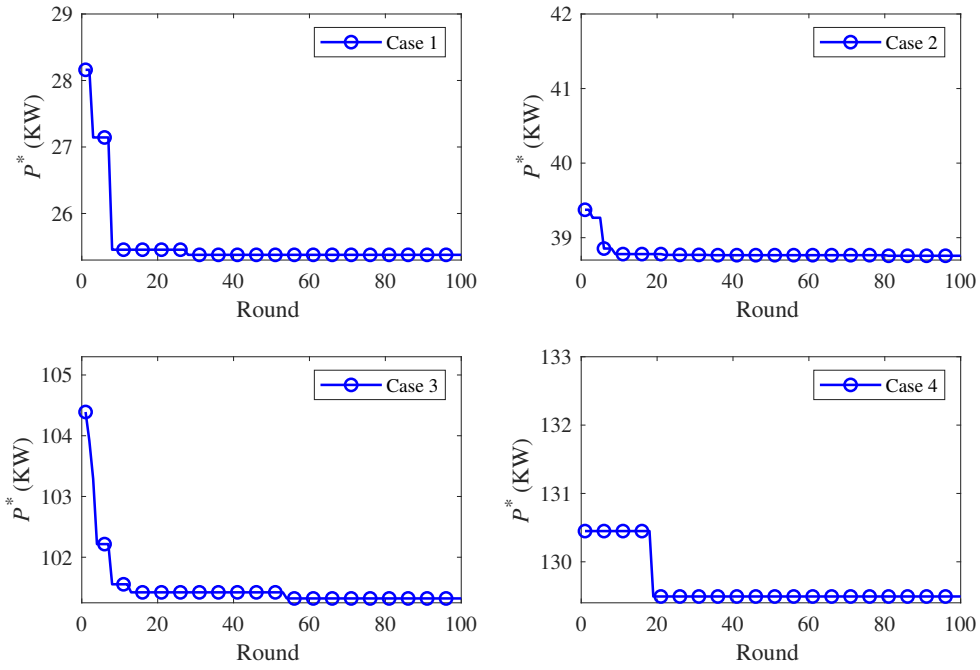


Figure 5. The iteration of BRS under four cases.

5.6. Test of Flexibility

The testing scenario is as follows: First, all nodes work well, as shown in Figure 6a. Then, Node 4 PUMP-A breaks down or is in maintenance, as shown in Figure 6b. Next, Node 4 is repaired, as shown in Figure 6c. The total flow rate and head demand on the main pipe is Case 3 listed in Section 5.1.

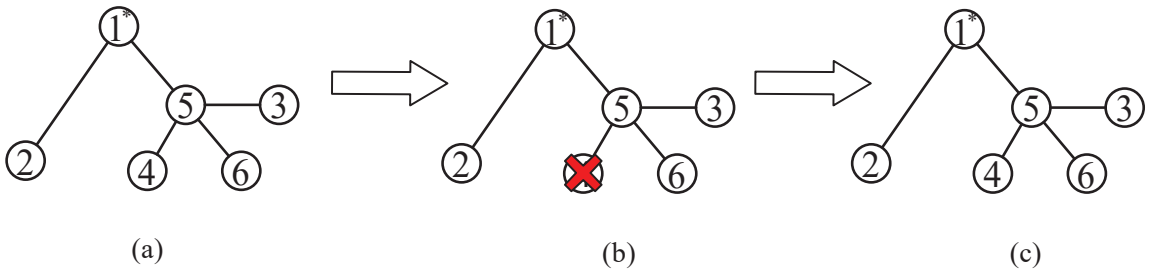


Figure 6. The topologies of pump intelligent nodes. (**a**) Toplogy with all 6 pumps working in normal; (**b**) Topolgy with Pump 4 out of service; (**c**) Topolgy with Pump 4 back to normal.

Figure 7 shows the iteration process of BRS under a different number of nodes corresponding to the ones in Figure 6. From Figure 7, P^* tends to converge to the optimal points. When all pumps work well, as shown in Figure 6a,c, P^* converges to the same optimal point. As expected, the number of nodes changing does not hinder the capacity of our algorithm to cooperate with neighbors and perform the on-line optimization. BRS algorithm does not need to stop the whole pump system and the pump is plug-and-play. In contrast, the centralized control methods need to explicitly reformulate the problem to be solved manually and are hard to automate in practice.

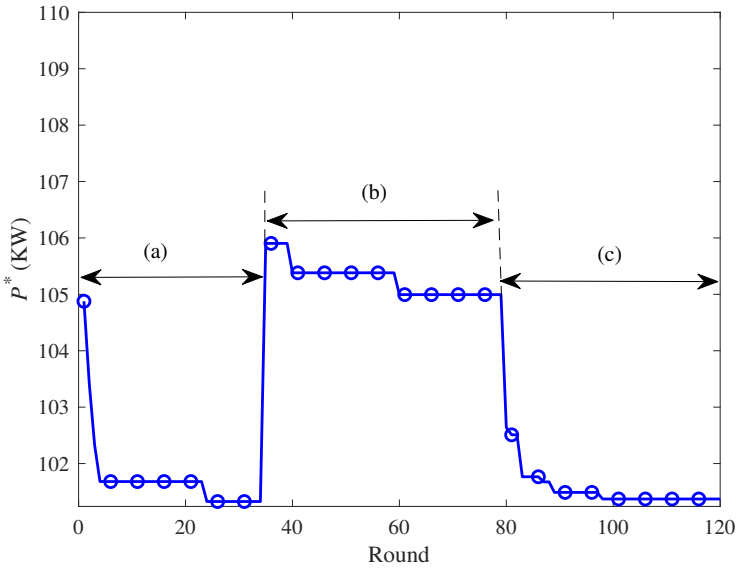


Figure 7. The iteration of BRS under different number of nodes.

These experiments validate that our BRS algorithm can adapt to the system dynamic changing by reallocating the flow rates in an on-line fashion.

5.7. Hardware Test of BRS

Furthermore, we developed a distributed low-cost hardware platform to test the algorithm performance. The hardware platform is based on Raspberry Pi3, Model B, 1GB RAM. Every Raspberry Pi3 represents an intelligent node and has the wireless and Ethernet communication interface. It can connect with neighbor nodes through a wireless communication interface. Every node stores the pump model and simulates a smart pump. Figure 8 presents the topology of the pump nodes. The total flow rate and head demand on the main pipe is Case 3 listed in Section 5.1.

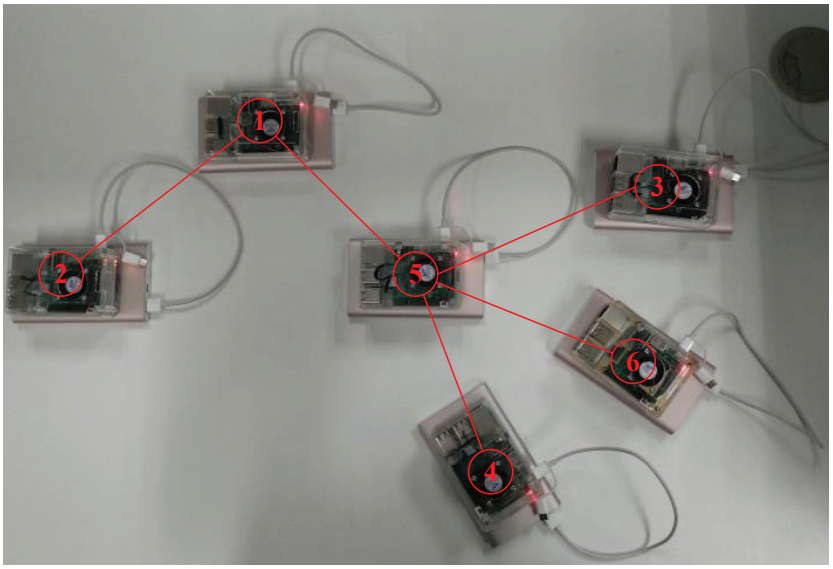


Figure 8. The hardware platform for test distributed algorithm.

Figure 9 shows the iteration process of the BRS algorithm on the hardware platform. BRS algorithm can converge to the optimal point in the first few iterations, which clearly shows that the BRS algorithm can perform on Raspberry Pi3-based low-cost hardware platform. The whole hardware

system is fully distributed and there are not any central monitoring hosts. When the number of pumps increases, traditional centralized methods need a high-cost computing platform to perform the centralized optimization. As a result, the BRS algorithm can reduce the system cost.

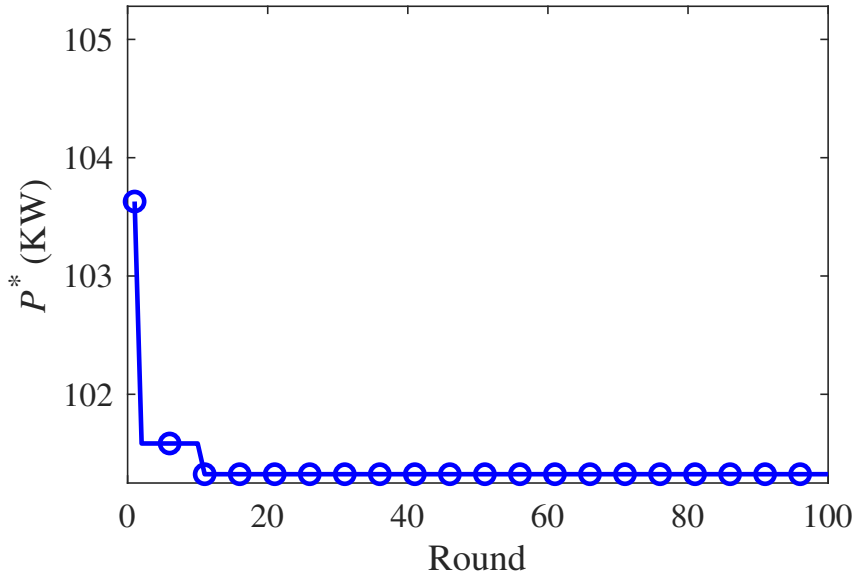


Figure 9. The iteration of BRS on hardware platform.

6. Conclusions

The distributed optimization problem with parallel-connected pumps is studied in this paper. We propose a distributed optimal control algorithm BRS for on-off status and flow rates of parallel-connected pumps in HVAC systems. The proposed BRS algorithm consists of two parts: First, in order to process the network information, we apply the breadth first search algorithm to construct a tree for exchanging messages. Second, all nodes coordinate with each other and randomly sample the speed ratios. The intelligent nodes communicate with their neighboring nodes and run collaboratively to perform the proposed distributed approach. Compared with existing methods, the solutions of our BRS algorithm meet the total flow rate demand without any deviations and achieve the minimum pump energy consumption. BRS algorithm has the convergence guarantee and shows the great flexibility in adapting to the system dynamic changing.

Based on the BRS algorithm, we successfully solve the distributed optimization problem with parallel-connected pumps and achieve the minimum pump energy consumption. Compared with previous works, our BRS algorithm can exactly meet the total flow rate demand and has the convergence guarantee. Our BRS algorithm can be more fitted to practical engineering, work on low-cost hardware and the pump is plug-and-play. However, in this paper we only consider that the pump network is synchronous. As future work, we will try to extend the algorithm to asynchronous networks. If the condition is allowed, we will test the BRS algorithm in practical systems. Also, we believe that our method can be extended to optimize the set points of parallel-connected heat pumps. It will be our possible future work.

Author Contributions: All authors made contributions to the research of this paper. X.W. wrote the manuscript. Q.Z. revised the findings of this work. Y.W. developed the distributed simulation platform. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Project of China (No. 2017YFC0704100 entitled New generation intelligent building platform techniques, and 2016YFB0901900) and the National Natural Science Foundation of China (No. 61425027), the 111 International Collaboration Program of China under Grant BP2018006 and the BNRist Program under Grant No. BNR2019TD01009.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

$a_{P1,i}, a_{P2,i}, a_{P3,i}, b_{P1,i}, b_{P2,i}, b_{P3,i}$	pump performance parameters
H_i	head of pump i (m)
H_0	total demand head (m)
m	total number of pumps
n_i	actual speed of pump i
$n_{0,i}$	rated speed of pump i
$\mathcal{N}_i$	set of child nodes of node i
P_k^*	optimal estimation of total power consumption after the k-th sample
P_i	power consumption of pump i
$P_{T,i,k}$	partial summation power consumption over subtree of node i
$Q_{T,i,k}$	partial summation flow rate over subtree of node i
Q_i	flow rate of pump i (L/s)
Q_0	total demand flow rate (L/s)
$\boldsymbol{R}_{T,i,\delta}$	combination of node's optimal speed ratio neighborhood over the subtree of node i Greek letters
ω_i	speed ratio of node i
ω^-	lower limitation of speed ratio
$\omega_{i,k}$	the k-th speed ratio sample of node i
$\boldsymbol{\omega}_{T,i,k}$	the combination of the speed ratio over subtree of node i
ω_i^*	optimal speed ratio
$\widetilde{\omega}_{i,k}^*$	optimal estimation of speed ratio after the k-th sample
$\boldsymbol{\omega^*}$	optimal combination of speed ratio
$\widetilde{\boldsymbol{\omega}}_{\boldsymbol{k}}^*$	optimal estimation combination of speed ratio after the k-th sample
ξ_P	energy saving rate
η_i	mechanical efficiency
$\eta_{M,i}$	motor efficiency
$\eta_{VFD,i}$	variable frequency efficiency
Δ	violation of $\sum Q_i - Q_0$
Subscripts	
i	pump i
k	the k-th sample
T	the subtree

Abbreviations

The following abbreviations are used in this manuscript:

HVAC	Heating, ventilating, and air-conditioning system
BRS	Breadth first search random sampling algorithm
BFS	Breadth first search algorithm
SC	Sequence control strategy
SLSQP	Sequential least squares programming algorithm
GA	Genetic algorithm
TA	Transfer algorithm
DSP	Distributed simulation platform

Appendix A

Appendix A.1

Algorithm A1: Breadth First Search algorithm [34]

Input: Choose one node i as a root

Output: Every node appoint a parent variable to convey its parent

Initialize: i sends search message to neighbors ;
if *an unmarked node accepts message* **then**
    Mark itself;
    Appoint one node that sends a search to it as a parent;
    Send search to neighbors at the next step ;
    Goto final;
final ;
return its parent;

Appendix A.2

Algorithm A2: Message processing algorithm

Input: the child nodes set $\mathcal{N}_i$, the flow rate $Q_{i,k}$, the power consumption $P_{i,k}$, the speed ratio $\omega_{i,k}$, $Q_{T,i,k-1}, P_{T,i,k-1}, \boldsymbol{\omega}_{T,i,k-1}$;

Output: every intelligent node's the partial summation flow rate $Q_{T,i,k}$,the partial summation power consumption $P_{T,i,k}$ and the combination of the speed ratio $\boldsymbol{\omega}_{T,i,k}$ over subtree of node i ;

Initialize $Q_{T,i,k} = Q_{i,k}$, $P_{T,i,k} = P_{i,k}$, $\boldsymbol{\omega}_{T,i,k} = [\boldsymbol{\omega}_{T,i,k}, \omega_{i,k}]$;
Send $Q_{T,i,k-1}, P_{T,i,k-1}, \boldsymbol{\omega}_{T,i,k-1}$ message to parent's input message queue and receive child node $j, j \in \mathcal{N}_i$ message ;
$Q_{T,i,k} = Q_{T,i,k} + \sum_{j \in \mathcal{N}_i} Q_{T,j,k_j}$;
$P_{T,i,k} = P_{T,i,k} + \sum_{j \in \mathcal{N}_i} P_{T,j,k_j}$;
$\boldsymbol{\omega}_{T,i,k} = [\boldsymbol{\omega}_{T,i,k}, \bigcup_{j \in \mathcal{N}_i} \boldsymbol{\omega}_{T,j,k_j}]$;
return $Q_{T,i,k}, P_{T,i,k}, \boldsymbol{\omega}_{T,i,k}$;

References

1. Pérez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
2. Arun Shankar, V.K.; Umashankar, S.; Paramasivam, S.; Hanigovszki, N. A comprehensive review on energy efficiency enhancement initiatives in centrifugal pumping system. *Appl. Energy* **2016**, *181*, 495–513. [CrossRef]
3. Liu, M.; Ooka, R.; Choi, W.; Ikeda, S. Experimental and numerical investigation of energy saving potential of centralized and decentralized pumping systems. *Appl. Energy* **2019**, *251*, 113359. [CrossRef]
4. Kusiak, A.; Li, M.; Tang, F. Modeling and optimization of HVAC energy consumption. *Appl. Energy* **2010**, *87*, 3092–102. [CrossRef]
5. Ali, A.; Coté, C.; Heidarinejad, M.; Stephens, B. Elemental: An Open-Source Wireless Hardware and Software Platform for Building Energy and Indoor Environmental Monitoring and Control. *Sensors* **2019**, *19*, 4017. [CrossRef] [PubMed]
6. Wang, S.; Burnett, J. Online adaptive control for optimizing variable-speed pumps of indirect water-cooled chilling systems. *Appl. Therm. Eng.* **2001**, *21*, 1083–1103. [CrossRef]
7. Li, H.; Shi, J.; Li, Z.; Yang, C.; Zhang, Y.; Liang, Z. A Group Control Method of Central Air-conditioning Chilled Water Pumps with the Aim of Energy Saving. In Proceedings of the 2018 International Conference on Power System Technology (POWERCON), Guangzhou, China, 24–26 October 2018; pp. 4517–4524. [CrossRef]
8. Zhao, T.; Ma, L.; Zhang, J. An optimal differential pressure reset strategy based on the most unfavorable thermodynamic loop on-line identification for a variable water flow air conditioning system. *Energy Build.* **2016**, *110*, 257–268.

9. Ma, Z.; Wang, S. Energy efficient control of variable speed pumps in complex building central air-conditioning systems. *Energy Build.* **2009**, *41*, 197–205. [CrossRef]
10. Gao, D.C.; Wang, S.; Sun, Y. A fault-tolerant and energy efficient control strategy for primary-secondary chilled water systems in buildings. *Energy Build.* **2011**, *43*, 3646–3656. [CrossRef]
11. Bonvin, G.; Demassey, S.; Le Pape, C.; Maïzi, N.; Mazauric, V.; Samperio, A. A convex mathematical program for pump scheduling in a class of branched water networks. *Appl. Energy* **2017**, *185*, 1702–1711. [CrossRef]
12. Horváth, K.; van Esch, B.; Vreeken, D.; Pothof, I.; Baayen, J. Convex Modeling of Pumps in Order to Optimize Their Energy Use. *Water Resour. Res.* **2019** , *55*, 2432–2445. [CrossRef]
13. Koor, M.; Vassiljev, A.; Koppel, T. Optimization of pump efficiencies with different pumps characteristics working in parallel mode. *Adv. Eng. Softw.* **2016**, *101*, 69–76. [CrossRef]
14. Menke, R.; Abraham, E.; Stoianov, I. Modeling Variable Speed Pumps for Optimal Pump Scheduling. In Proceedings of the World Environmental and Water Resources Congress 2016, West Palm Beach, FL, USA, 22–26 May 2016.
15. Jepsen, K.L.; Hansen, L.; Mai, C.; Yang, Z. Power consumption optimization for multiple parallel centrifugal pumps. In Proceedings of the 1st Annu IEEE Conf Control Technol Appl CCTA 2017, Kohala Coast, HI, USA, 27–30 August 2017; pp. 806–811.
16. Kraft, D. *A Software Package for Sequential Quadratic Programming. Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*; DFVLR: Cologne, Germany, 1988.
17. Wang, J.Y.; Chang, T.P.; Chen, J.S. An enhanced genetic algorithm for bi-objective pump scheduling in water supply. *Expert Syst. Appl.* **2009**, *36*, 10249–10258. [CrossRef]
18. Rong, Z.G.; Wu, G.; Ye, T. Global optimization of pump configuration problem using extended crowding genetic algorithm. *Chin. J. Mech. Eng.* **2004**, *17*, 1.
19. Cimorelli, L.; Covelli, C.; Molino, B. Optimal Regulation of Pumping Station in Water Distribution Networks Using Constant and Variable Speed Pumps: A Technical and Economical Comparison. *Energies* **2020**, *13*, 2530. [CrossRef]
20. Barán, B.; Von, Lücken C.; Sotelo, A. Multi-objective pump scheduling optimization using evolutionary strategies. *Adv. Eng. Softw.* **2005**, *36*, 39–47. [CrossRef]
21. Hashemi, S.S.; Tabesh, M.; Ataeekia, B. Ant-colony optimization of pumping schedule to minimize the energy cost using variable-speed pumps in water distribution networks. *Urban Water J.* **2014**, *11*, 335–347. [CrossRef]
22. Olszewski, P. Genetic optimization and experimental verification of complex parallel pumping station with centrifugal pumps. *Appl. Energy* **2016**, *178*, 527–539. [CrossRef]
23. Beckerle, P.; Schaede, H.; Rinderknecht, S. Fault Diagnosis and State Detection in Centrifugal Pumps—A Review of Applications. *Mech. Mach. Sci.* **2015**, *21*, 387–398. [CrossRef]
24. Dai, Y.; Jiang, Z.; Shen, Q.; Chen, P.; Wang, S.; Jiang, Y. A decentralized algorithm for optimal distribution in HVAC systems. *Build. Environ.* **2016**, 95, 21–31. [CrossRef]
25. Zhao, Q.; Jiang, Z. Insect Intelligent Building (I2B): A New Architecture of Building Control Systems Based on Internet of Things (IoT). In *Advances in Intelligent Systems and Computing, Proceedings of the International Conference on Smart City and Intelligent Building, ICSCIB 2018, Hefei, China, 15–16 September 2018*; Fang, Q., Zhu, Q., Qiao, F., Eds.; Springer: Singapore, 2019; Volume 890, pp. 457–466.
26. Liu, Z.; Chen, X.; Xu, X.; Guan, X. A decentralized optimization method for energy saving of HVAC systems. In Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, USA, 17–21 August 2013; pp. 225–230. [CrossRef]
27. Zheng, W.; Wu, W.; Zhang, B.; Li, Z.; Liu, Y. Fully distributed multi-area economic dispatch method for active distribution networks. *IET Gener. Transm. Distrib.* **2015**, *9*, 1341–1351. [CrossRef]
28. Timotheou, S.; Panayiotou, CG.; Polycarpou, MM. Distributed Traffic Signal Control Using the Cell Transmission Model via the Alternating Direction Method of Multipliers. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 919–933. [CrossRef]
29. Zhao, Q.; Wang, X.; Wang, Y.; Jiang, Z.; Dai, Y. A P2P Algorithm for Energy Saving of a Parallel-Connected Pumps System. In *Advances in Intelligent Systems and Computing ICSCIB 2018, Proceedings of the International Conference on Smart City and Intelligent Building, Hefei, China, 15–16 September 2018*; Fang, Q., Zhu, Q., Qiao, F., Eds.; Springer: Singapore, 2019; Volume 890, pp. 385–395.

30. Campana, PE.; Li, H.; Yan, J. Dynamic modelling of a PV pumping system with special consideration on water demand. *Appl. Energy* **2013**, *112*, 635–645. [CrossRef]
31. Pumps, S. *Centrifugal Pump Handbook*, 3rd ed.; Elsevier Science: Amsterdam, The Netherlands, 2010.
32. Ulanicki, B.; Kahler, J.; Coulbeck, B. Modeling the efficiency and power characteristics of a pump group. *J. Water Resour. Plan. Manag. ASCE* **2008**, *134*, 88–93. [CrossRef]
33. Wang, S.; Xing, J.; Jiang, Z.; Dai, Y. A Decentralized Swarm Intelligence Algorithm for Global Optimization of HVAC System. *IEEE Access* **2019**, 7, 64744–64757. [CrossRef]
34. Lynch, N.A. *Distributed Algorithms*; Elsevier: Amsterdam, The Netherlands, 1996.
35. Dai, Y.; Jiang, Z.; Xin, S.; Chen, P.; Li, S. Optimal control of variable speed parallel-connected pumps. In Proceedings of the 13th International Conference on Indoor Air Quality and Climate, Hong Kong, China, 7–12 July 2014; pp. 87–94.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).