Automation in Construction 139 (2022) 104264

Contents lists available at ScienceDirect

Automation in Construction

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

Rule-based generation of HVAC duct routing

Zhisen Chen [a], Hang Guan [a], Xiaolei Yuan [a], Tian Xie [b], Peng Xu [a,*]

[a] School of Mechanical Engineering, Tongji University, Shanghai, China
[b] School of Civil Engineering, Chongqing University, Chongqing, China

ARTICLE INFO

Keywords:
Duct routing
Automation
Traversal algorithm
Optimization
Fast resistance calculation
Obstacle avoidance

ABSTRACT

Traditional manual duct design can result in fan energy waste due to ducts network imbalance and human design error. This paper aims to develop an automated duct routing method, which is able to connect air diffusers across space with the consideration of duct pressure balance, construction obstacles, air resistance reduction and minimising construction costs. Treating air diffusers as nodal connections of a graph, the method uses a rule-based traversal algorithm (RBTA) and a fast resistance calculation model (FRCM) to generate the duct network. The methodology is tested on six different building layouts for the automatic design of ducts in a room and the results meet designer's requirements. Compared with the traditional manually designed duct network, the new duct network is more balanced and costs less. The automated approach cuts design time by more than ten folds and avoid human mistake and error in the traditional design process.

1. Introduction

Fan energy consumption of central air-conditioning units in public buildings is considerable. For example, in China, total fan energy use in public buildings is 270 billion kWh in 2011, which is 3.25 times greater than the annual power generation of the Three Gorges Hydropower Plant (the world's largest hydropower plant) [1,2]. Part of the fan energy consumption is caused by the poor duct design. Designers are unable to explore all duct design options due to time constraints. Engineers tend to design duct work based on their experience. In China alone, if the duct resistance loss is reduced by 1%, a total of 27 billion kWh will be saved and carbon dioxide emissions will be reduced by 26.9 million tons. Also, HVAC engineers must repeatedly change the duct design in response to changes in upstream architecture design. Duct drawing modifications are repetitive, simple, and error prone, and can be fully automated by the program. Automated duct design reduces the design workload and ensures optimal duct connections. This paper aims to provide an algorithm for fully automated duct design to achieve the optimal air diffusers connection network.

The distribution network in the heating, ventilation, and air-conditioning (HVAC) is commonly referred to as "duct" in air systems and "pipe" in water systems. Few studies have been conducted on the automated design of air systems; most previous studies have focused on the water system, mainly on algorithms for generating pipe routing. In 1961, Lee [3] proposed a maze algorithm that divided a region into a grid and searched for non-obstacle cells. The cells were labelled with different weights based on whether they were adjacent to the starting point; the next element was searched until the target cell was reached. However, this search method required a large amount of memory. In 1969, Hightower [4] suggested an escape algorithm that was fast and had a small memory requirement, but could not guarantee a solution. Beginning with Newell [5], studies on pipe routing generation algorithms have focused on pipelines with branches. To generate branching pipe routing, Park and Storch [6] considered a branch pipeline as a composite of two simple forms (end-forked and middle-forked forms), and presented a cell-generation method. Fan et al. [7] developed a branch pipe-routing algorithm based on the maze algorithm, splitting the multi-terminal connection into several two-terminal routings. Similarly, based on the method proposed by Lee [3], Sui and Niu [8] improved the genetic algorithm to generate branching pipeline routes in 3-D space.

Asmara and Nienhuis [9] combined particle swarm optimisation (PSO) (proposed by Kennedy and Eberhart [10]) and Dijkstra's algorithm (Dijkstra, 1951 [11]) to solve the routing of pipelines with branches. They used PSO to determine the order of connections, and then used Dijkstra's algorithm to sequentially connect the terminals. Kimura [12] regarded pipe branches as devices and routed the pipeline by removing them. Min et al. [13] applied an improved jump point search algorithm that considered the number of elbows and piping rules. The Steiner minimum tree (SMT) is another significant method that is well-suited to the branching pipe routing problem, but its solutions are

* Corresponding author.
E-mail address: xupeng@tongji.edu.cn (P. Xu).

https://doi.org/10.1016/j.autcon.2022.104264
Received 22 July 2021; Received in revised form 9 March 2022; Accepted 12 April 2022
Available online 23 April 2022

Nomenclature

Abbreviations

AD	Air diffuser
DCAD	Air diffusers of two rows connected directly in ADRC
ICAD	Air diffusers of two rows connected indirectly in ADRC
ADRC	Air diffuser row connection
FRCM	Fast resistance calculation model
HVAC	Heating, ventilation, and air-conditioning
LVPRC	Lower virtual point row connection
MCP	Must be connected point
MVPRC	Middle virtual point row connection
NCP	Non-connectible point
PSO	Particle swarm optimization
RBTA	Rule-based traversal algorithm
SMT	Steiner minimum tree
VP	Virtual point
VPRC	Virtual point row connection
3-D	Three-dimensional

List of symbols

A	Cross-sectional area of duct
$Adj_{I_i}^{-}$	Left adjacent inlet of i^{th} inlet
$Adj_{I_i}^{+}$	Right adjacent inlet of i^{th} inlet
B_{un}	Balance rate limit between duct branches
c	Character in inlet string
C	Connected point position indexes set
$C_{A,\ V}$	Set containing position indexes of all ADs and partial VPs
CV	Set containing all options of free VP connections;
D	Duct hydraulic diameter [m]
Dis	Set containing all possible AD assignments
$D_{install}$	Minimum installation distance [mm]
F	All-feasible inlet connections set
I	Inlet position index set
k	Number of free VPs
l	Length of duct cross-section [mm]
l_{min}	Length of smallest duct cross-sectional edge [mm]
L	Duct air flow rate [m^3/s]
L_{I_i}	Left-side inlets of i^{th} inlet
m	Number of inlets
M	MCP position index set
n	Number of ADs in AD row;
N'	AD number of ICAD segment
N	NCP position index set
NV	Set containing all free VPs in ICAD segment
O	Full set of position indexes
O'	Full set of position indexes for ICAD segment;
P	Potential connected point position index set
P_e	Perimeter of duct cross-section [mm]
PV_i	Set containing possible connections for i^{th} free VP
R_{I_i}	Right-side inlets of i^{th} inlet
s	Partial string of inlet string
S	Inlet string
S'	Partial inlet string for ICAD segment
S_i	i^{th} character value of inlet string
S_p	Set containing partial strings of inlet string
U	Set containing specific connection for all free VPs
v_{set}	Design air flow velocity of duct [m/s]
w	Width of duct cross-section [mm]
W	Set containing one connection approach for i^{th} free VP
x	Power of α, integer
X	Connected point position index set
Y_i	Set containing ADs distributed to i^{th} inlet
Z	Set containing all AD allocation combinations
ΔP_f	Friction resistance of duct [Pa]
ΔP_l	Local resistance of duct [Pa]
α	Length-to-width ratio of duct
η	Variance rate limit between two schemes
ξ	Local resistance coefficient

NP–hard and discrete problems. Many researchers have improved the optimisation algorithm to solve NP–hard and discrete problems. Hu et al. [14] and Luyet et al. [15] used the ant colony optimisation approach to obtain results for rectilinear SMT and SMT in graphs, respectively. Liu and Wang [16,17] used PSO to find the SMT of rectilinear and multi-terminal branch pipe routing. Currently, only Medjdoub and Bi [18] are concerned with the duct routing of air systems. They used a "branch and bound" [19] algorithm for the duct routing for ceiling-mounted fan coil systems in buildings. However, they have only implemented a semi-automated design that relies on user experience to select a partial connection solution.

Although ducts are connected similarly to pipes, the algorithms appropriate for pipes can produce unreasonable connections for duct routing. First, the nodes in pipe routing algorithms are terminal points, whereas most of the nodes in ducts are air outlets. These duct nodes are open nodes and imply a dimension in the vertical direction. As shown in Fig. 1, the same joint connection represents different components in the two systems, indicating an elbow and a 3-D tee in the pipe and duct, respectively. However, a 3-D tee does not exist in an actual duct system. Second, owing to the different characteristics of air and water, four-way connections are often used in ducts, but not in pipes, as they tend to cause hydraulic imbalances. Third, the main optimisation targets of previous pipe algorithms are minimum pipe length and number of bends, without considering hydraulic stability and pipe size. However, pressure balance is very important to duct design. An unbalanced design will concentrate airflow mainly on one duct, resulting in an oversized duct, which is not conducive to construction. At the same time, imbalance can lead to high or low air velocity at the air outlet, resulting in a blowing sensation or insufficient fresh air, which affects the user experience. The pressure characteristics between two points vary with the connection of all points, and cannot be calculated by existing pipe algorithms.

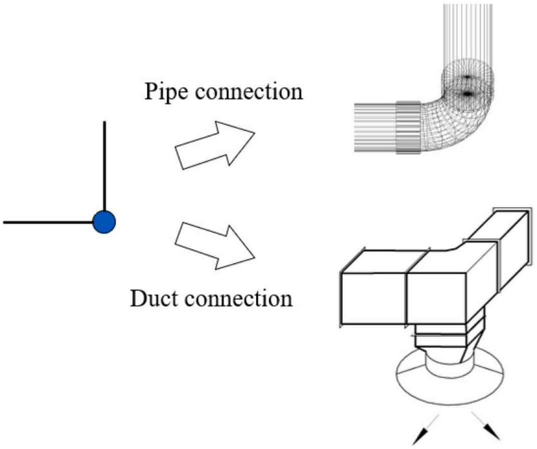


Fig. 1. Difference between pipe and duct with the same connection.

Traditional duct design (air outlet placement, duct routing and sizing, and pressure balance) is commonly produced manually, according to the ASHRAE instructions from look-up tables. HVAC engineers also need to consider low construction costs and not overlap with building structures (obstacle avoidance) when designing ducts. Duct design needs to be automated, but there are no algorithms suitable for air diffusers connection. Therefore, this paper aims to generate a duct routing connecting air diffusers across space with the objectives of pressure balance, obstacle avoidance, resistance reduction and low construction costs. This study proposes a new method to address complex and combinatorial problems (NP-hard problems) in HVAC distribution system design to achieve more precise duct design through automated calculation of feasible solutions. The methodology proposed in this study is divided into two main parts: (a) achieving feasible duct routing based on practical connection rules by representing an air diffuser and a virtual node with binary characters '1' and '0', respectively; (b) fast resistance calculation of multiple connection results to obtain the best routing solution.

Compared to previous research, the innovations of this study are:

(1) Rule-based traversal algorithm (RBTA) and fast resistance calculation model (FRCM) are established, enabling fully automatic duct design for the first time.

(2) A duct routing algorithm is proposed that is suitable for different numbers of air outlets and virtual nodes to satisfy construction requirements.

(3) Resistance calculations are incorporated into the optimisation process; the optimisation objectives consider pressure balance, pipeline cost, and operational resistance.

(4) This approach has been tested for different optimisation objectives and different numbers and arrangements of air outlets.

The remainder of this paper is organized as follows. Section 2 presents the duct routing mathematical model and the knowledge base of traversal rules. The process of establishing RBTA and FRCM is described in Sections 3 and 4, respectively. Section 5 presents the running results of the duct-routing generation approach. A conclusion is presented in Section 6.

2. Knowledge model

The knowledge model in this study introduces the rules of duct connection, and the created principles and definitions for the routing algorithms. Fig. 2a shows the CAD design drawing of the duct connections for an all-air system in a large space. The rules that must be followed during duct routing generation can be summarised from the duct layout. Based on these rules, an algorithm can be proposed to traverse and find all matching options. Some principles are added to the traversal process to remove solutions that are unfit for the optimisation target. Screening options accelerates the searching process, and reduces the computational effort of the FRCM.

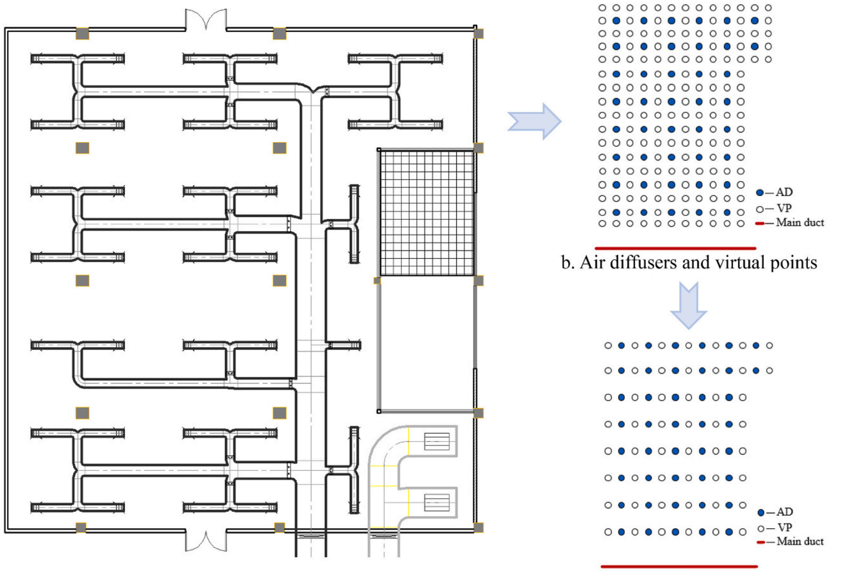
2.1. Problem transformation

As shown in Fig. 2a, duct connections share the following common features:

a. The main ducts (mostly in the corridor) into the room are on the same side of all air diffusers nodes (top, bottom, left, right), and can be converted to enter the room on the bottom side of all air diffusers by transforming the relative coordinates. Starting from the joints of the main ducts, the air diffusers are connected upward, row by row.
b. For convenience in construction and installation, ducts are aligned horizontally or vertically in the plane relative to the main duct (or corridor). Thus, the distribution of the potential connected points is fixed and uniform around the air diffuser.

Based on these characteristics, the duct routing problem can be described as finding the best solution from a set of points (Fig. 2b), where the air diffusers must be connected and other virtual points are indeterminate. Thus, spatial connectivity is transformed into a two-dimensional topology-generation problem. To find the optimal routing layout, the connection weights between points should be specified, which is not possible because the resistance characteristics must be calculated when all air diffusers are connected. Thus, all connections must be determined before resistance calculations are performed.

It is arithmetically intensive and requires a large amount of memory to directly search for all connection schemes for air diffusers (ADs) and virtual points (VPs). For a room with $n_x \times n_y$ ADs, the total number of nodes including VPs is $(2n_x + 1)(2n_y + 1)$, and the number of possible connected edges is $8n_xn_y + 2n_x + 2n_y$, as shown in Fig. 3. Each edge can be connected or not, resulting in a total of $2^{8n_xn_y+2n_x+2n_y}$solutions. Thus, directly traversing through all nodes is not reasonable, as the computational cost increases exponentially as the number of air diffusers increases; many solutions violating the AD connection rules are stored, which requires cumbersome filtering rules and involves numerous



b. Air diffusers and virtual points

a. CAD design drawings by engineer

c. Simplified AD and VP in algorithm

Fig. 2. Possible connection points.

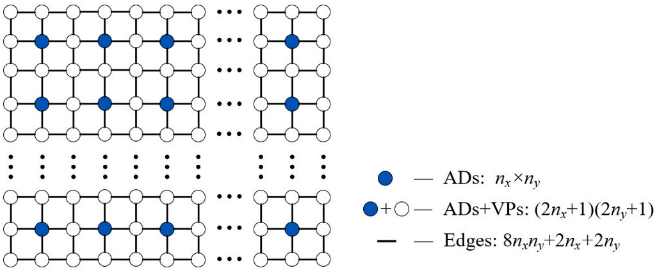

— ADs: $n_x×n_y$
●+○ — ADs+VPs: $(2n_x+1)(2n_y+1)$
— — Edges: $8n_xn_y+2n_x+2n_y$

Fig. 3. Possible connections for *n* rows and *n* columns of ADs.

unnecessary calculations and memory use. To address this issue, principles are established in the traversal process to remove schemes that are not suitable for actual connection, reducing the computational effort.

2.2. Rules of traversal

To discard node topologies according to the actual duct layout, and ones that are apparently non-optimal in terms of pressure balance, duct consumables, and operational resistance, some general rules are proposed for selecting duct connection solutions to establish traversal algorithms. These rules can be divided into two categories: duct connection rules that must be obeyed according to the needs of the actual project, and defined filtering rules that remove clearly not optimal results.

2.2.1. Duct connection rules

a) Each node has only one entry point, but multiple exits are allowed.
b) All branch terminal points in the connection topology can only be ADs, and not VPs.
c) To avoid a 3-D tee (Fig. 1), ADs cannot be connected with an elbow (Fig. 4).

2.2.2. Defined filtering rules

By transformation, the main duct is always at the bottom of the ADs; thus, an algorithm was proposed to search connections from the bottom to the top, row by row. The setpoints in Fig. 2b indicate two types of rows: AD rows contain both ADs and VPs, and VP rows contain only VPs. Based on the type of row, AD connections can be classified as AD row connections (ADRC) or VP row connections (VPRC). In ADRC, ADs are connected by the VP of the AD row or two ADs are connected directly (Fig. 5a). In VPRC, ADs of the same row are not connected to each other, and some ADs are connected by the VP of the VP row and supplied with air by one air inlet (Fig. 5b).

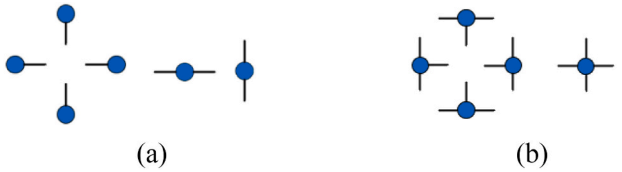


Fig. 4. (a) Ways ADs can be connected; (b) ways ADs cannot be connected.

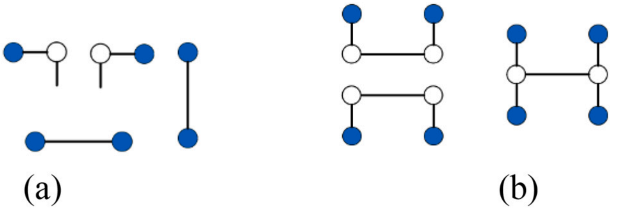


Fig. 5. (a) Connection of ADs in ADRC; (b) connection of ADs in VPRC (single/dual row).

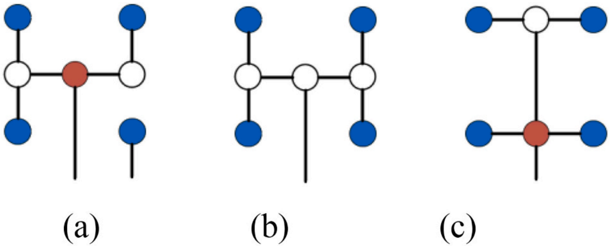


Fig. 6. (a) Combination connection; (b) VPRC; (c) ADRC.

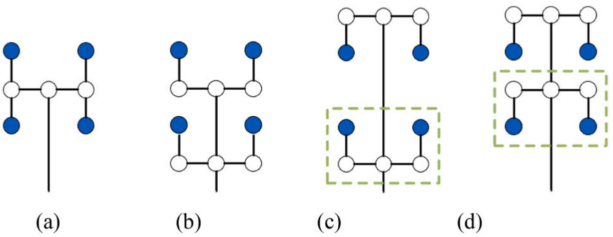


Fig. 7. (a) MVPRC of two AD rows; (b–d) single AD row VPRC for two AD rows.

As shown in Fig. 6, the combination of ADRC and VPRC (Fig. 6a) has more imbalance points (points in red) than VPRC (Fig. 6b), greater duct consumption than ADRC (Fig. 6c), and can generate many not optimal connections. Fig. 7 shows all VPRC solutions for two AD rows. Fig. 7a shows a middle VP row connection (MVPRC), which is obviously the best layout for double AD rows. Fig. 7b–d show solutions with dual AD rows connected separately via VPRC of a single AD row; these connections are pressure imbalance and require more ducts. The green boxes in Fig. 7 indicate two types of VPRC for a single AD row. The connection in Fig. 7c is much better because the entry point is on the lower side of the ADs. Based on the analysis, the stipulations were set as follows:

a) Combination connections are discarded during the traversal; all ADs in the same row must be connected to the same VP row in a two-row VPRC.
b) Two adjacent rows of ADs cannot be connected by VPRC in isolation, and only MVPRC is considered.
c) The VPRC of a single AD row only considers the lower VP row connection (LVPRC).

With the rules of traversal, all VPRCs selected in the algorithm are

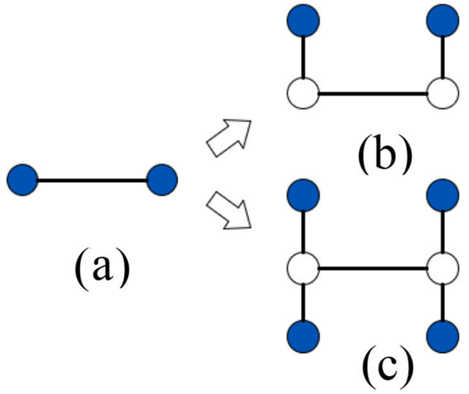


Fig. 8. (a) ADRC of single row; (b) LVPRC of single row; (c) MVPRC of two rows.

the LVPRC of a single AD row and the MVPRC of two AD rows (as shown in Fig. 8b and c), which can be considered as extensions of the ADRC of a single AD row (Fig. 8). All VPRC can be transformed from the ADRC, so the traversed set points are simplified from Fig. 2b and C.

3. Duct routing algorithm

Based on the rules proposed in Section 2, the duct routing algorithm (RBTA) is established to search for all reasonable AD connections; a flowchart is shown in Fig. 9, consisting of three main modules: (a) pre-treatment of ADs; (b) calculation of input for AD row; (c) Computation of AD row connections. The first module generates the string representing AD rows. The second module calculates the must be connected and non-connectable nodes in one AD row. The third module computes all reasonable connection results of an AD row. After computing all feasible connections in an AD row, the inputs and connections for the next AD row are calculated based on these results. This operation is repeated row by row to seek feasible solutions until the final AD row. Non-optimal solutions are discarded during traversal in part b and c. Part b filters unreasonable connections between two AD rows and part c removes the unreasonable connections in a single AD row.

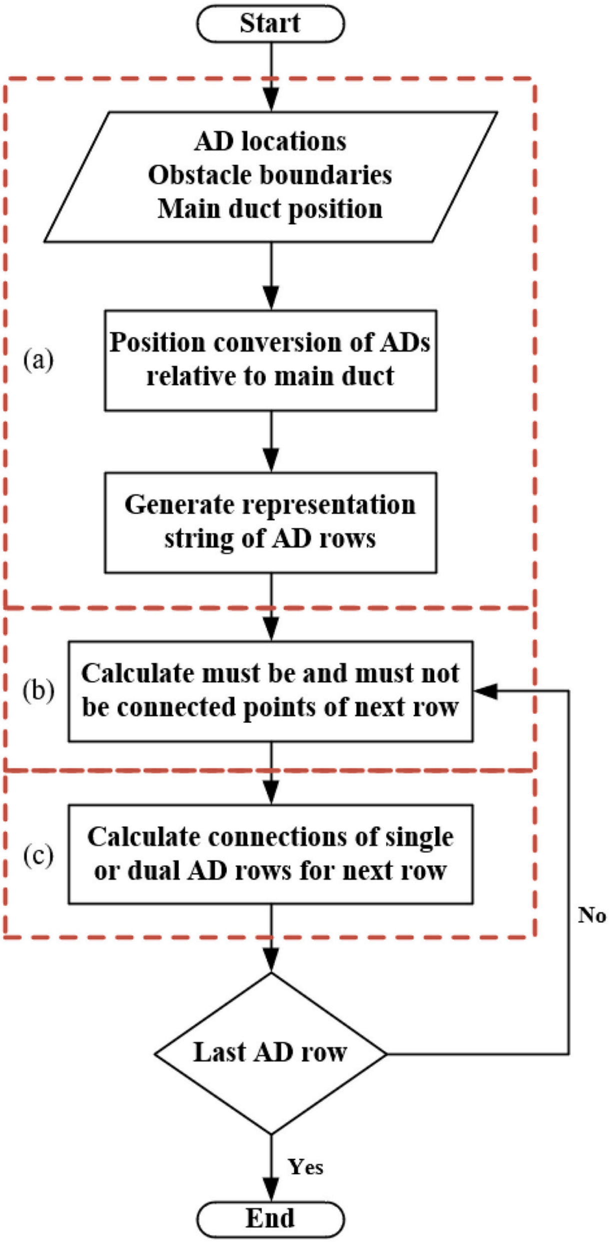

Start
AD locations
Obstacle boundaries
Main duct position
(a)
Position conversion of ADs relative to main duct
Generate representation string of AD rows
(b)
Calculate must be and must not be connected points of next row
(c)
Calculate connections of single or dual AD rows for next row
No
Last AD row
Yes
End

Fig. 9. RBTA flowchart.

3.1. Pre-treatment of ADs

Pre-treatment of ADs refers to the string conversion of ADs based on AD locations and main duct position. To achieve row-by-row traversal, all ADs are divided into rows and columns by relative coordinates using the main duct direction as the x-axis direction. The characters “1” and “0” are used for AD and VP, respectively; the AD rows can be represented by a string (e.g., “01010… 0.010”). Assuming that there are n ADs, the length of the AD row string is $2n + 1$. If an AD is missing, the missing point is replaced with a VP, and its character changes to “0”. Fig. 10 illustrates the process of adding VPs to a group of ADs and converting them into multi-row strings.

3.2. Calculation of input for AD row

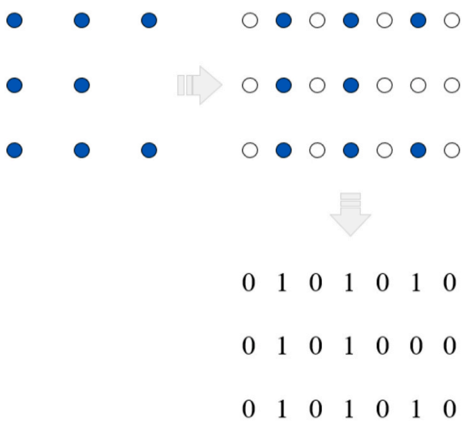
When filtering non-conforming connections between two AD rows, the searching process is greatly affected by the previous row connection, and more specifically by points that must or must not be connected to the next row. Based on the established rules (Section 2.2.1), the points must be connected (represented by MCPs) are the VPs acting as the end points of row connections (Fig. 11) and the user-specified main duct entry points. The non-connectible points (NCPs) include the horizontally connected ADs, the VPs not connected to the ADs in the row (Fig. 12), and the points overlap with obstacles in the room. The MCPs and NCPs are the input parameters for calculating the connection of next AD row and are determined by the connections of the previous AD row. MCPs and NCPs can only be obtained after the previous AD row connection has been determined. Thus, for the first row, MCPs and NCPs are generally not present unless the user defines them in advance.

3.3. Calculation of connections for AD row

After the MCPs and NCPs have been determined, the connection calculation for the next row begins by generating inlet strings that match the connection rules between two AD rows. The inlet strings show the number and location of points connected between two rows. The generated inlet strings may not conform to the single row connection rules, so the unfit part need to be removed. Within a single row, a feasible inlet string can generate multiple connection solutions that must be filtered to find all feasible solutions. Fig. 13 shows the calculation workflow of the row connections.

3.3.1. Inlet string calculation

To correspond to the AD row string, the AD row inlet is also defined



0 1 0 1 0 1 0

0 1 0 1 0 0 0

0 1 0 1 0 1 0

Fig. 10. Strings representing ADs and VPs.

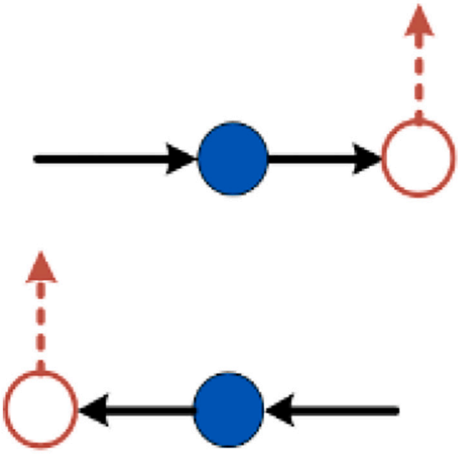


Fig. 11. Points that must be connected.

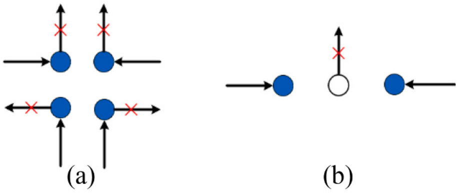


Fig. 12. Points that are non-connectible: (a) ADs connected horizontally; (b) VPs not connected.

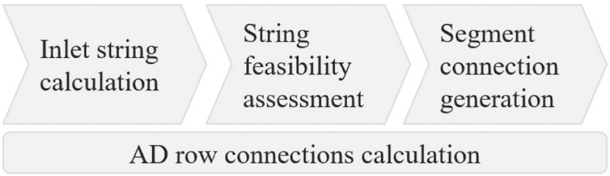

Inlet string calculation
String feasibility assessment
Segment connection generation
AD row connections calculation

Fig. 13. Calculation workflow of row connections.

in this study as a string, where “1” and “0” indicate an inlet and no connection with the previous row, respectively.

Performing the ADRC traversal process for an AD row without a missing AD, the generation of inlet strings should follow the connection rules for two AD rows; all characters in the corresponding position in MCPs must be “1” and those in NCPs must be “0”. To ensure that each AD has only one entry, the number of inlets cannot exceed the number of ADs. For an AD row with n ADs, the rules can be expressed as

$$S_i = \begin{cases} 1, i \in C, C \in F \\ 0, \text{else} \end{cases}, i = 1, 2, \dots, 2n+1 \tag{1}$$

where S_i is the character value of a position in the inlet string. The connected point position index set C is an element belonging to the all-feasible inlet connection set F. F can be calculated as

$$F = \{X \cup M | X \subset P, |X| \leq n - |M|\} \tag{2}$$

where M is a set that describes the position indexes of the points in the MCPs. As a subset of P, X is a set that contains the position indexes of the connected points in potential points. Set P, containing the position indexes of all potential points that are not in MCPs or in NCPs, can be determined as

$$P = O \backslash (M \cup N) \tag{3}$$

where N is the position index set of the NCPs, and O is the full set of position indexes.

$$O = \{1, 2, \dots, 2n+1\} \tag{4}$$

To ensure consistency, for AD rows lacking ADs, inlet strings observing inlet connection rules can be obtained by collapsing, accomplished by replacing “00…0” with a single “0”, and performing the calculations (**Eqs.** 1–4).

For LVPRC and MVPRC, there is only one inlet for an AD row. Thus, the constraints for reasonable inlet connections change, and set F becomes

$$F = \{X \cup M | X \subset P, |X| \leq 1 - |M|\} \tag{5}$$

An empty set of F indicates that there is no feasible connection solution.

3.3.2. *String feasibility assessment*

The inlet strings generated for LVPRC and MVPRC have only one entry point, and all ADs are connected to the VP row. The connections represented by these inlet strings satisfy all connection rules, but the inlet strings generated for ADRC may not. Because the inlet strings for ADRC may have multiple inlets, and are directly connected to AD rows, the location of the inlets may cause some unconnected or over-connected ADs (e.g., AD row string = “01010”, inlet string = “11,000”). Thus, further assessment is required for single-row connections.

For ADRC, a lateral connection is not allowed after an AD is directly connected by the previous row, in which case the inlet can only supply air to one AD. However, an inlet connected to the VPs in an AD row can feed multiple ADs at the same time, and may have multiple connection solutions. Based on these differences, the inlet strings are split into partial strings of directly connected (DCAD) and indirectly connected (ICAD) ADs. Indirectly connected partial inlet strings should be considered further.

The characteristics of an inlet string can be used to ensure that each segment in an AD row without DCAD complies with the connection rules; the number of “1” in each indirectly connected partial inlet string must be greater than one and less than the number of corresponding ADs. Table 1 presents the algorithms used for the function. When the result of the algorithm is true, the inlet string meets all ADRC requirements.

3.3.3. *Segment connection generation*

Because there is only one inlet for LVPRC, MVPRC, and DCAD, the distribution of ADs is fixed, and all ADs are connected to the inlet. However, a reasonable inlet string for an ICAD segment can have several connections, as shown in Fig. 14. One AD can be assigned to different inlets and generate multiple solutions. Thus, the distribution of ADs must be determined according to the position of the inlets during segment connection generation.

The AD allocation process must also satisfy the established rules; each AD must be assigned to one adjacent inlet, and cannot be connected across inlets. In addition, each inlet must be connected to at least one AD. For an ICAD segment with n' ADs and an inlet string S', the AD distribution can be mathematically formulated as

$$\begin{gathered} Dis = \Big\{(Y_1, Y_2, \dots, Y_m) | \cup_{i=1}^{m} Y_i = \{2, 4, \dots, 2n'\}, \cap_{i=1}^{m} Y_i = \Phi, \\ Y_i \in Z, \forall y \in Y_i, Adj_{I_i}^{-} < y < Adj_{I_i}^{+}, i = 1, 2, \dots, m\Big\} \end{gathered} \tag{6}$$

where Dis is a set containing all possible AD assignments, and each of its elements $(Y_1, Y_2, \dots, Y_m)$ represents a feasible allocation of ADs; m is the number of inlets; I is a set that contains the position index of all inlets; Y_i is a set that contains ADs distributed to the i^{th} inlet I_i and belongs to set Z; Z is a set containing all AD allocation combinations; $Adj_{I_i -}$ and $Adj_{I_i +}$ denote the left and right adjacent inlets of the i^{th} inlet, respectively.

I, m, and Z are calculated as

Table 1
Algorithm for determining reasonableness of inlet string.

```
Input:  $S$, an inlet string
Output:  True or False, reasonability of $S$

$S_p :=$ a set that contains partial strings of $S$
for each character $c$ in $S$ do
  if $c$ is "1" and its position index is an even number then
    $S_p :=$ split $S$ at $c$ and remove $c$
  end if
end for
for each partial string $s$ in $S_p$ do
  if the number of "1s" in $s >$ (length of $s$ - 1) / 2 then
    return False
  end if
end for
return True
```

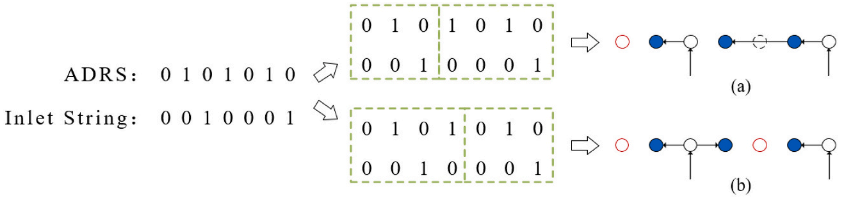


Fig. 14. One inlet string generating multiple connections.

$$I = \left\{ x \mid S_x^{'} = 1, x \in \{1, 3, \dots, 2n^{'} + 1\} \right\} \tag{7}$$

$$m = |I| \tag{8}$$

$$Z = \{Y | Y \subset \{2, 4, \dots, 2n^{'}\}, Y \neq \Phi\} \tag{9}$$

Adj_{I_i-}andAdj_{I_i+} are expressed as

$$Adj_{I_i}^{-} = \begin{cases} 1, & L_{I_i} = \Phi \\ max(L_{I_i}), & \text{else} \end{cases} \tag{10}$$

$$Adj_{I_i}^{+} = \begin{cases} 2n^{'} + 1, & R_{I_i} = \Phi \\ min(R_{I_i}), & \text{else} \end{cases} \tag{11}$$

where $L_{I_i}$and R_{I_i} are the left-side and right-side inlets of the i^{th} inlet, calculated as

$$L_{I_i} = \{x | x \in I, x < I_i\} \tag{12}$$

$$R_{I_i} = \{x | x \in I, x > I_i\} \tag{13}$$

In addition to the ADs, unconnected VPs (red circles in Fig. 14) also have multiple connection solutions. However, unlike ADs, these VPs do not have to be connected. Thus, one free VP normally has three connection options: unconnected, connected to the left AD, or connected to the right AD.

The set NV containing all free VPs in the ICAD segment can be calculated as

$$NV = O^{'} \backslash C_{A,V} \tag{14}$$

where O' is the full set of position indexes for an ICAD segment, $O' = \{1, 2, \dots, 2n' + 1\}$; $C_{A,\ V}$ is the set containing the position indexes of all ADs and partial VPs. The VPs are non-free VPs that act as inlets or are between two connected ADs (the VP indicated by the dashed circle in Fig. 14a). $C_{A,\ V}$ can be obtained as

$$C_{A,V} = \cup_{i=1}^{m} \{x | min(Y_i \cup \{I_i\}) \leq x \leq max(Y_i \cup \{I_i\}), x \in O^{'}\} \tag{15}$$

The possible connections for the i^{th} free VP can be formulated as

$$PV_i = \{\{NV_i - 1, NV_i\}, \{\}, \{NV_i, NV_i + 1\}\} \tag{16}$$

Based on the connection characteristics, all combinations of free VP connections can be expressed by Eq. (17).

$$CV = \left\{ U | U = \cup_{i=1}^{k} \{W | W \subset PV_i, |W| = 1\}, \cap_{i=1}^{k} U_i = \Phi, \cup_{i=1}^{k} U_i \subset O^{'} \right\} \tag{17}$$

where CV is a set containing all free VP connection options; its element set U denotes a specific connection for all free VPs; k is the number of free VPs, $k = |NV|$; W, like U_i, is a subset of PV_i with one element, that is, one connection approach for the i^{th} free VP; constraint $\cap_{i=1}^{k} U_i = \Phi$ limits each free VP to connect with no more than one AD; constraint $\cup_{i=1}^{k} \subset O'$ considers that an end-point-free VP has only two connection options.

For LVPRC and MVPRC, free VPs can only be endpoints that are not the inlet; $NV = \{x | x \in \{1, 2n' + 1\}, x \notin I\}$. Other calculations for possible connections are consistent with those for the ICAD segment.

Route generation is based on the assumption that one row performs one connection approach (VPRC or ADRC). In an actual project, an AD row can perform both approaches (Fig. 15). This combination can be achieved by dividing an AD row into multiple sections, with each section considered as a row using a single connection method. In this case, the VPs between the two sections cannot be simultaneously connected to the ADs for both sections.

After calculations to search for all feasible connections in single or double AD rows, the inputs and connections for the next AD row are

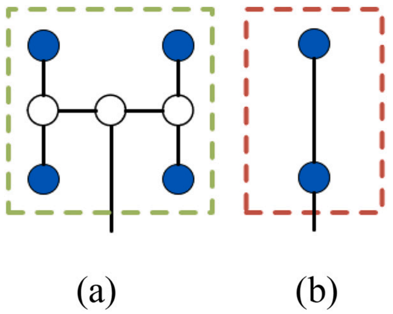


Fig. 15. One AD row with two connection methods: (a) VPRC; (b) ADRC.

calculated based on these results. This operation is repeated row by row to seek feasible solutions until the final AD row; the results represent possible routes of all ADs in accordance with the rules.

3.4. Adjustment of some nodes position

The connection of all ADs can be considered as a directed graph with the main duct joint as the root node. Before calculating the resistance characteristics of each connection topology, the actual positions of each node need to be determined. For material saving and obstacle avoidance in actual projects, it is necessary to adjust some nodes position of the connection topology calculated by the RBTA.

Firstly, in the generated connection topology, the VP positions are relative; their exact locations must be determined according to the actual installation requirements (the VPs between two ADs change location depending on the connection (Fig. 16)). When a VP is located at the endpoint of the connected segment in an AD row, its position can be adjusted to the minimum installation distance $D_{install}$ from the AD.

Secondly, as obstacles (e.g., columns) may exist in the actual building rooms, ducts connected between two nodes may collide with them. The calculated optimal arrangement cannot be constructed in practice, so the avoidance of obstacles needs to be considered at the design stage. As shown in Fig. 17, the overlap of obstacles and connection solutions can be divided into two cases, edge and node overlaps. In order to follow the rules of duct connection, the nodes in one topology can be divided into fixed connected nodes and variable connected nodes (Fig. 17.**a**). A fixed connected node is a node whose location and connection direction (horizontal or vertical) are unchangeable, specifically a non-endpoint AD or a tee or a four-way. A variable connection node is a node whose position or connection direction can be changed, corresponding to the actual node being the terminal AD or elbow. The collision of an edge with an obstacle can be divided into three cases depending on the type of edge endpoints, and the specific methods of avoidance are shown in Fig. 17.**b**. As the AD arrangement will avoid the obstacle, the overlapping nodes can only be VPs. VPs in the connection scheme are either elbows, tees or four-ways. Since an obstacle cannot be circumvented by moving the four-way nodes, the four-way overlap is considered as an unreasonable solution, and then discarded. The method of obstacle avoidance for elbows and tees is shown in Fig. 17.**c**. The tee's obstacle avoidance mainly considers two parallel branches, and is handled in a similar way to the collision of edges. Both node and edge offsets follow a horizontal or vertical offset using the minimum installation distance as the offset distance.

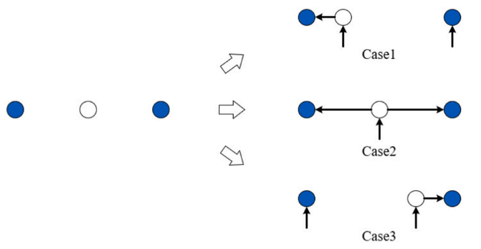


Fig. 16. Position of VPs in different connection cases.

4. Fast resistance calculation

After generating connections for all ADs, the optimal value must be determined through the optimisation objective. In actual projects, designers are mainly concerned with the pressure balance, duct material costs, and the operating expenses of a routing option. These objectives require resistance calculations for all schemes; engineers traditionally use look-up tables and interpolation methods to calculate, which is not reasonable for a large number of computational processes. Based on existing resistance calculation methods and some reasonable assumptions, an FRCM applicable to the automated routing optimisation process is established in this study, consisting of two parts: duct sizing and resistance calculation on a directed graph.

4.1. Duct sizing

To avoid the table look-up process, it is assumed that all duct sizes meet certain criteria: the length and width of all rectangular sections of the duct are in powers of α of the smallest edge l_{min}. From geometric knowledge, all sectional areas of the duct are also in powers of α of the smallest area (l_{min}^2). The formula for solving the cross-sectional area of the duct is

$$A = \frac{L}{v_{\text{set}}} \tag{18}$$

where A is the cross-sectional area of the duct at the design air velocity (m^2); L is the air flow rate of the duct (m^3/s); v_{set} is the design air flow velocity of the duct (m/s).

As the duct size must meet the standard size, $A \approx \alpha^x l_{\text{min}}^2$. Solving yields

$$x = \left[\frac{lgL - lgv_{\text{set}} - 2lgl_{\text{min}}}{lg\alpha}\right] \tag{19}$$

where x is the power of α and must be an integer; the symbol '[]' denotes the rounding function.

The targets of duct design include achieving minimum distribution resistance and saving space. For rectangular ducts, a more similar length and width (squarer) produces less resistance. However, for installation, a flatter duct section occupies a smaller space. Considering these requirements, the length-to-width ratio of the duct is specified as

$$\frac{l}{w} = \begin{cases} \alpha, & x = \text{odd number} \\ \alpha^2, & x = \text{even number} \end{cases} \tag{20}$$

Thus, the perimeter of the duct Pe is:

$$Pe = 2(l+w) = \begin{cases} 2l_{\text{min}}\left(\alpha^{\frac{x+1}{2}} + \alpha^{\frac{x-1}{2}}\right), & x = \text{odd number} \\ 2l_{\text{min}}\left(\alpha^{\frac{x}{2}+1} + \alpha^{\frac{x}{2}-1}\right), & x = \text{even number} \end{cases} \tag{21}$$

4.2. Resistance calculation on directed graph

After the nodes are located by the actual connection and the topology is transformed into a directed graph, the depth-first search algorithm [20] can be used to calculate the characteristics of all nodes and edges in the graph. The properties of each node and edge are shown in Fig. 18. With the features of the duct connections, each node can only have one predecessor and several successors (no more than three). Before calculation starts, the air flow rate must be assigned to all nodes, which is the design value for ADs and zero for VPs.

The depth-first search explores edges out of the most recently discovered node that still has unexplored edges leaving it. Once all the

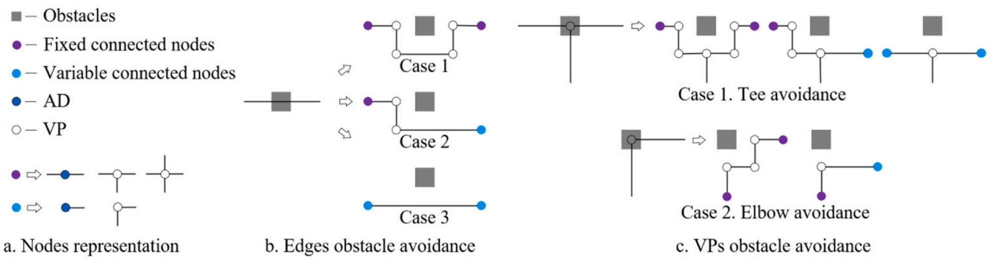


Fig. 17. Obstacle avoidance in two overlapping situations.

node's edges have been explored, the search "backtracks" to explore edges leaving the Fnode. This process continues until all nodes that are reachable from the original source node are discovered. Before the same operation is performed on nodes with the same rank, a series of operations must be performed. Each computation step involves calculation of the attributes of a node and its Fedge, specified as

$$Fedge.F = Node.F + \sum_{i=1}^{j} Sedge\_i.F;$$

$$Fedge.S = (l, w) = f(Fedge.F, l_{\min}, v_{set}, \alpha);$$

$$Fedge.L = \sqrt{(Fnode.X - Node.X)^2 + (Fnode.Y - Node.Y)^2};$$

$$Fedge.SA = 2 \times (l + w) \times Fedge.L;$$

$$Fedge.V = \frac{Fedge.F}{Fedge.A};$$

$$Fedge.FR = f(Fedge.F, Fedge.S, Fedge.L);$$

$$Sedge.LR = f(Fedge.V, Fedge.S, Sedge.V, Sedge.S);$$

$$Sedge.R = Sedge.LR + Sedge.FR;$$

$$Node.R = max(Sedge\_i.R + Snode\_i.R), i = 1, \dots, j;$$

$$Node.UB = \begin{cases} 0, \forall i \in [1, j], \dfrac{Node.R - (Sedge\_i.R + Snode\_i.R)}{Node.R} \leq B_{\text{un}} \\ 1, else \end{cases};$$

Where

$Fedge.F\ Sedge\_i.FNode.F =$ the corresponding air flow rate;
$j =$ number of $Snode$s;
$Fedge.S\ Sedge.S =$ the duct size, using the formula in Section 4.1 to calculate;
$Fnode.X\ Fnode.YNode.X\ Node.Y =$ the $2-$D coordinates of the node;
$Fedge.L =$ the length of the edge;
$Fedge.SA =$ the duct surface area of the corresponding edge;
$Fedge.V\ Sedge.V =$ the air velocity in the duct;
$Fedge.A\ Sedge.A =$ the cross $-$ sectional area of the duct;
$Fedge.FR\ Sedge.FR =$ the friction resistance of the duct;
$Fedge.LR =$ the local resistance of the duct;
$Sedge.R =$ the total resistance of a duct section;
$Node.R =$ the distribution resistance of the latter ducts;
$Node.UB =$ the unbalance rate of the connected duct sections;
$B_{\text{un}} =$ balance rate limit between duct branches (exceeding the value is considered unbalanced)

The resistance calculation consists of the calculation of friction resistance and local resistance of the duct, which can be quickly calculated using empirical formulas instead of look-up tables. Table 2 presents the empirical formulas used in this study. As the local resistance of each edge is influenced by the attributes of Fedge, in one calculation step, the frictional resistance of the Fedge and the local resistance of the Sedge can be calculated, but not the local resistance of the Fedge.

5. Testing and evaluation

After the resistance calculation of the directed graphs, the optimal connection of the ADs can be selected according to different objectives. In an actual project, the optimal solution can be the most balanced pressure (*MBP*), the least use of duct material (*LUM*), and the minimum distribution resistance (*MDR*). These objectives can be expressed by the following equations:

$$MBP = min\left(\sum Node.UB\right) \tag{22}$$

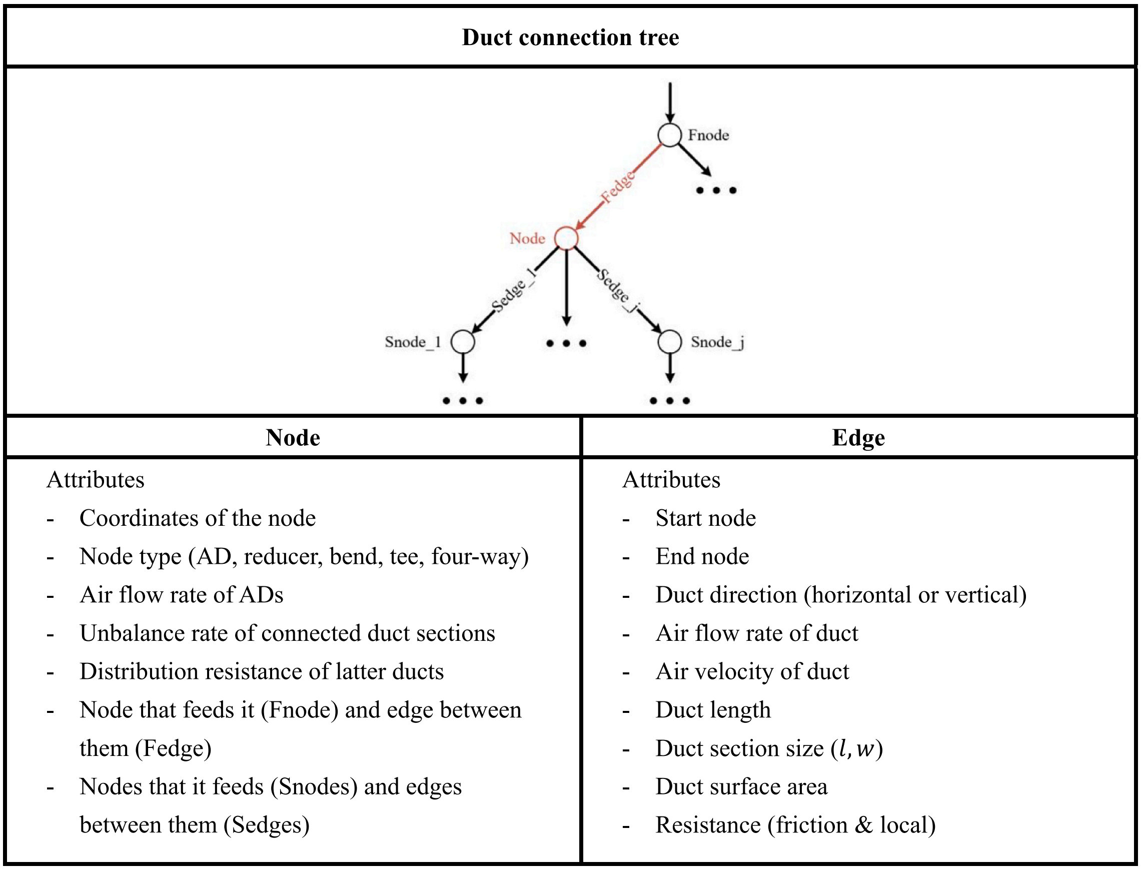
$$LUM = min\left(\sum edge.SA\right) \tag{23}$$

$$MDR = min(\text{root}\ Node.R) \tag{24}$$

According to user requirements, these objectives can be ranked in terms of importance; when the previous objectives of the two schemes are within the variance rate limit η (considered as equal values), the optimal solution of the next objective can be found.

This approach to generating the optimal duct route was tested and evaluated using different numbers and arrangements of ADs. The building structure and room sizes used in this paper comply with the building module requirements [22], while the room lengths can be 4.5, 9 or 12 m. The building existing structure, which may collide with the ducts, are beams or columns. The beams can be circumvented by adjusting the height of the ducts, and will not affect the connection scheme. Therefore, this paper only considered the influence of columns in the room on the ADs connection. The column network is arranged at a spacing of 9 m. The ADs in the rooms are arranged in an automatic and uniform way, with one AD in a 4–6 m area [21]. Specifically, we cut the rectangular room orthogonally with the corridor direction as the x-axis, and arrange the AD in the midpoint of the small rectangle. The AD rows are determined in parallel to the corridor, from the close to the distant. Python software was used to implement the algorithm (Sections 3 and 4); the relevant parameter values are presented in Table 3.

The objectives can be divided into stability objectives (*MBP*) and economic objectives (*LUM*, *MDR*) according to their characteristics. Ordering them in combination produces four target combinations: *PMR*



Node	Edge
Attributes	Attributes
- Coordinates of the node	- Start node
- Node type (AD, reducer, bend, tee, four-way)	- End node
- Air flow rate of ADs	- Duct direction (horizontal or vertical)
- Unbalance rate of connected duct sections	- Air flow rate of duct
- Distribution resistance of latter ducts	- Air velocity of duct
- Node that feeds it (Fnode) and edge between them (Fedge)	- Duct length
- Nodes that it feeds (Snodes) and edges between them (Sedges)	- Duct section size (l, w)
	- Duct surface area
	- Resistance (friction & local)

Fig. 18. Attributes of duct connection tree.

– [*MBP*, *LUM*, *MDR*]; *PRM* – [*MBP*, *MDR*, *LUM*]; *MRP* – [*LUM*, *MDR*, *MBP*]; *RMP* – [*MDR*, *LUM*, *MBP*]. The running results with different numbers of ADs and different optimisation objectives are shown in Tables 4 and 5. The air supply nodes of all rooms are in the corridor, and the corridor centreline is used as the main duct in this paper. The graphs in Table 4 show the results of ADs connections from the main duct to the individual rooms.

According to the calculation results for several cases, the algorithm and model proposed in this study have the following characteristics:

a. The RBTA is effective in filtering connection solutions and can remove greater than 99.9% of the schemes according to the duct connection. With a large filtering capacity, this algorithm can still compute quickly (on the order of 0.1 s).

b. The execution time of the FRCM increases exponentially with the number of ADs due to the rapid increase in connection options. As the number of VPs increases, more time is required to determine the actual positions of points (Section 0), leading to an increase in the conversion time from strings to directed graphs. In addition, the number of branches and directed graph node layers increases considerably, increasing the

Table 2
Empirical formulas for resistance calculation.

Item		Description/Formula [21]
Duct type/material		Rectangular, galvanized steel, duct absolute roughness = 0.15 mm
Friction resistance of Fedge		$\Delta P_f = 1.05 \times 10^{-2} \times D^{-1.21} \times (Fedge.V)^{1.925} \times Fedge.L$ $D = \dfrac{2lw}{l+w}$
Local resistance of Sedge		$\Delta P_l = \xi \dfrac{Sedge.V^2}{2g}$
Local resistance coefficient ξ	Reducer	$\xi = 0.065\dfrac{Fedge.A}{Sedge.A} - 0.036$
	Bend	$\xi = 0.11$
	Tee	$\text{Branch}: \xi = 0.5\left(\dfrac{Sedge.V}{Fedge.V}\right)^2 + 1$ $\text{Main}: \xi = 0.35\left(1 - \dfrac{Sedge.V}{Fedge.V}\right)^2$
	Four-way	Considered as a tee with two branches

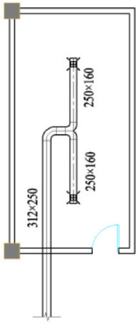
Table 3
Parameters used for testing.

Parameter	Value	Description [21]
α	1.25	Ducts in the room are small, so a smaller aspect ratio meets installation requirements. Choose a large aspect ratio for standard ducts and its square is also a standard duct
$l_{\min}$	100 mm	The shortest side in ASHRAE rectangular duct specification
Node.F	0.08 m^3/s	When the indoor noise level is 35– 50 dB, the design velocity of the AD is 1.5– 2.5 m/s; 2 m/s is chosen. Assuming an AD size that is commonly used in engineering, (200 mm × 200 mm), the air flow rate can be calculated
v_{set}	2 m/s	When the indoor noise level is 35– 50 dB, the design velocity of the duct branches is 2– 3 m/s; the middle value is chosen
$D_{install}$	1 m	Installation distance of duct local resistance components
$D_{arrange}$	5 m	AD alignment space
B_{un}	0.001	A smaller value produces a more accurate calculation, but increases the computational cost. If the value is too large, an unbalanced solution may be considered balanced. Use one order of magnitude smaller than the minimum local resistance
η	0.001	A smaller value produces a more accurate calculation, but increases the computational cost. If the value is too large, multiple solutions are generated. Use one order of magnitude smaller than the minimum order of magnitude of all optimization targets

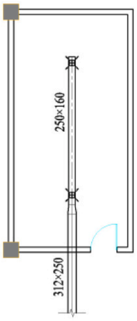
Table 4
Optimal duct routes for different numbers of ADs and different objectives.

Case 1: AD number = 2

$n_x \times n_y = 1 \times 2$

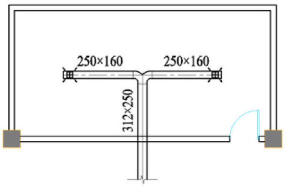


PMR & PRM

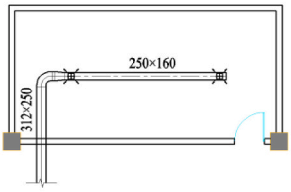


MRP & RMP

$n_x \times n_y = 2 \times 1$



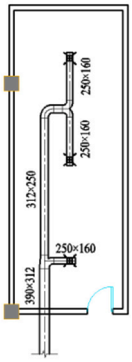
PMR & PRM



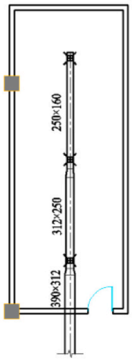
MRP & RMP

Case 2: AD number = 3

$n_x \times n_y = 1 \times 3$

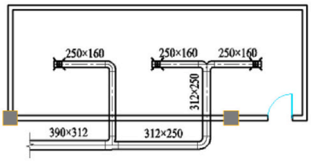


PMR & PRM

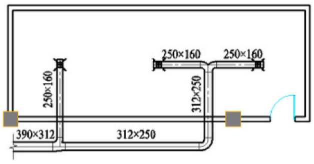


MRP & RMP

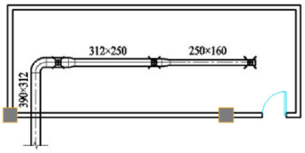
$n_x \times n_y = 3 \times 1$



PRM



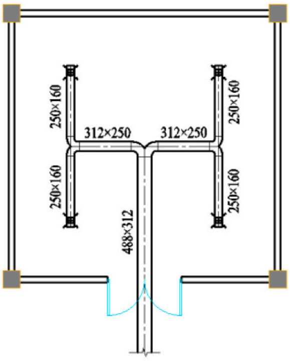
PMR



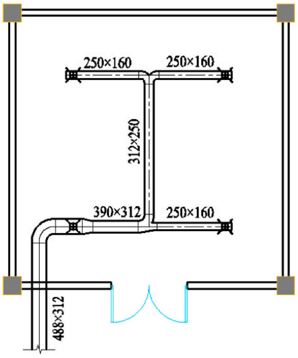
MRP & RMP

Case 3: AD number = 4

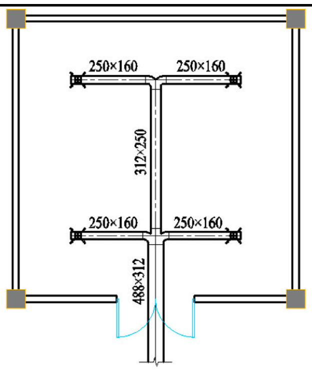
$n_x \times n_y = 2 \times 2$



PMR & PRM



MRP

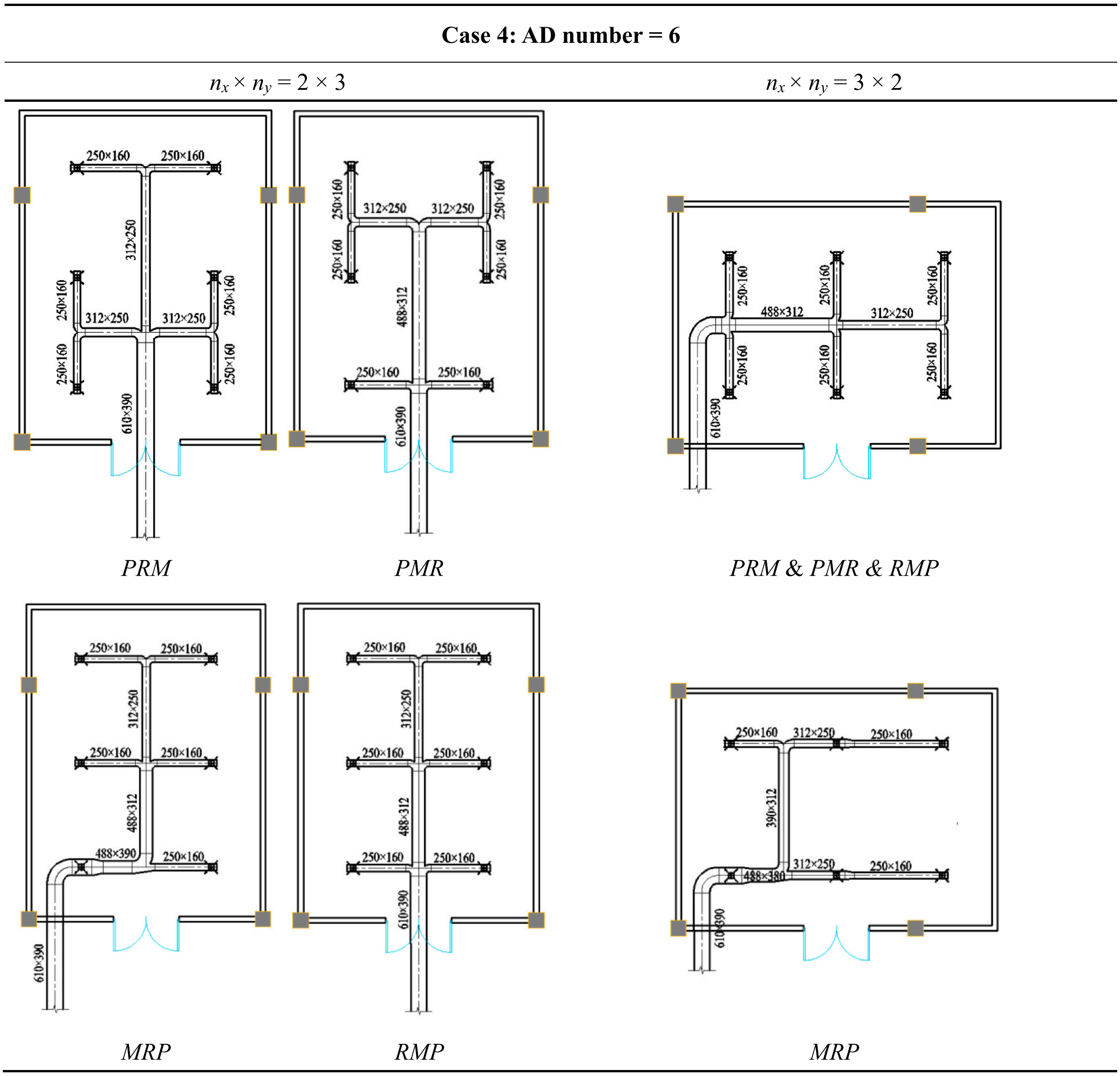


RMP

graph computation time. When the number of schemes selected by the RBTA is less than 5000, the optimisation running time is within 5 min, which better meets the time spent target of automated design.

c. The optimal design solution varies with the objectives. With stability as the primary objective, results with duct material consumption or operating resistance as secondary optimisation targets vary little, and in many cases have the same routes. However, with economy as the primary goal, the best solutions considering duct material cost or resistance as the first objective usually vary greatly.

d. Fast resistance calculations with a fixed duct aspect ratio produce some non-standard duct sizes; the resulting duct size is close to a standard duct as the aspect ratio is used in the majority of standard ducts. The non-standard duct of the optimal solution can be replaced with a standard size; this replacement has little impact on the pressure



characteristics of the design solution.

e. The limitation of this approach is the long execution time when the number of ADs is too large. For a room with more than nine ADs, automated duct routing has no advantage over manual drawing. For duct route generation in most rooms (less than 400 m^2), this method is effective.

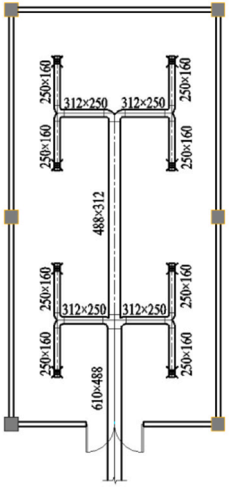
The proposed automatic design algorithm only involves solving the connection of ADs within a single room. For rooms with no internal columns, the design can be fully automated without human intervention. For rooms with columns inside, when the connection topology collides with the building structure, automatic obstacle avoidance can be achieved using the method in Section 3.4. However, when obstacles cannot be avoided automatically (e.g., overlapping with a four-way), the AD position needs to be adjusted manually.

There are few researches on the automated design of ducts, and the only available study is the semi-automated designs by Medjdoub and Bi [18]. In their study, the ducts in the room are defined as local ducts and the routing algorithm they use to solve Case 3 takes around 20s to calculate. For an experienced designer dealing with the same case would take several minutes to design and calculate the pressure characteristics of the ducts using design software. However, the algorithm proposed in this paper runs in less than 1 s, which is a significant improvement in design speed. This advantage becomes even more apparent when the number of rooms increases.

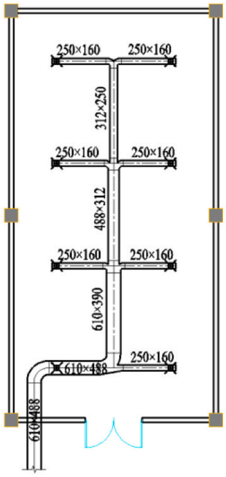
The application of the RBTA and FRCM to duct automation for the entire area is still in the research process and our team is working on its implementation. The automatic design of an entire HVAC duct system also requires the automation of many complex issues (e.g., the automatic determination of main duct locations, the division of supply air zones, the appropriate setting of local resistance components, the location of AHU rooms, the coordination between return and supply ducts, the avoidance of load-bearing walls and columns). Therefore, only the automated duct arrangements in one room is currently being tested and evaluated.

Case 5: AD number = 8

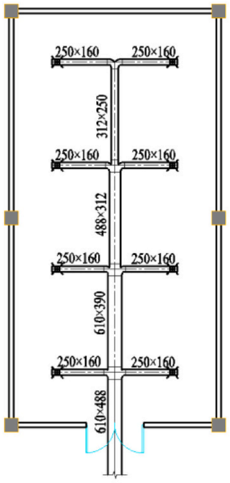
$n_x \times n_y = 2 \times 4$



PRM & PMR

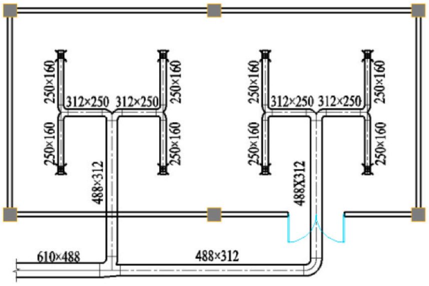


MRP

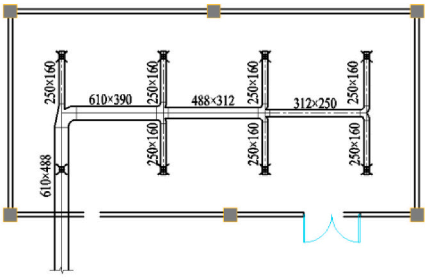


RMP

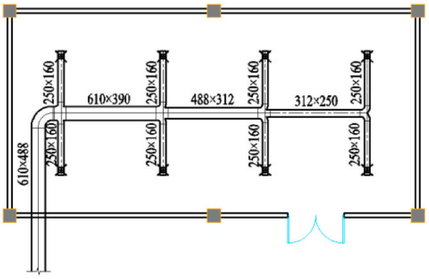
$n_x \times n_y = 4 \times 2$



PRM & PMR



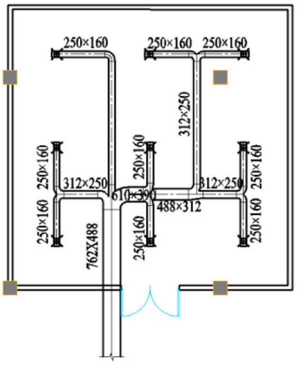
MRP



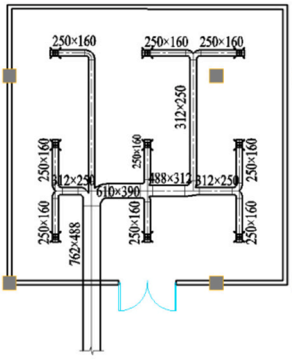
RMP

Case 6: AD number = 9

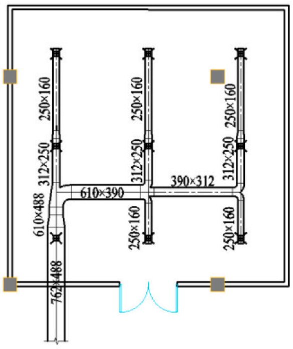
$n_x \times n_y = 3 \times 3$



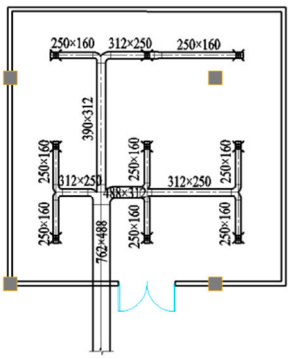
PRM



PMR



MRP



RMP

Table 5
Running results with different numbers and arrangements of ADs.

ADs ($n_x \times n_y$)	Number of connection schemes	Number of duct connection schemes	Execution time (s)	
			RBTA	FRCM
1 × 2	4.19×10^6	7	0.0020	0.011
2 × 1	4.19×10^6	11	0.0010	0.034
1 × 3	4.29×10^9	19	0.0020	0.042
3 × 1	4.29×10^9	39	0.0030	0.375
2 × 2	1.10×10^{12}	79	0.0060	0.606
2 × 3	2.88×10^{17}	646	0.0430	13.861
3 × 2	2.88×10^{17}	343	0.0160	12.780
2 × 4	7.56×10^{22}	5226	0.3651	271.249
4 × 2	7.56×10^{22}	4360	0.1700	631.052
3 × 3	1.93×10^{25}	8160	0.4611	1740.523

6. Conclusion

This paper proposed a new approach for the automatic design of HVAC ducts, generating duct routings for the first time in a fully automated process. Compared with the traditional manual duct design, the new duct network is more balanced. The automated approach cuts design time by ten folds, avoids human mistake and improves the quality of duct design. The process of this method is as follows. Firstly, the RBTA is used to filter out the solutions that are suitable for actual project according to the rules for duct connections. Each solution is then obstacle-avoided so that all solutions conform to the building structure and are guaranteed to be constructible. Finally, each of the remaining options is then subjected to multi-objective planning and the FRCM is used to solve for the optimal solution. The algorithm can be extended to optimise the duct connections for multiple rooms, providing an algorithmic basis for the automated design of duct systems.

The proposed RBTA is the first algorithm applicable to duct connections; it considers a variable quantity of connected VPs and the specificity of ADs, while effectively avoiding solutions that cannot be used in actual projects. The RBTA uses strings to represent AD connections and filters solutions based on rules that can quickly manage the NP-hard problem.

The presented FRCM is applied to perform fast calculations in RBTA-generated scenarios. This model simultaneously computes the pressure characteristics, duct material consumption, and operational resistance in the resistance calculation process. By fixing the duct aspect ratio to avoid table look-up and interpolation, the FRCM can perform fast and accurate computations on directed graphs of AD connectivity.

This study has some limitations. The FRCM is more suitable for small-quantity AD connection optimisation problems as the execution time increases with an increase in the number of ADs. In addition, ranking several objectives to search for an optimal solution may not be generalised for different duct systems. Thus, future research should apply the FRCM to large-quantity AD routing optimisation problems, and find a generic integrated optimisation target for different duct systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This paper has received funding from the Science and Technology Project of China Southern Power Grid Company (Grant No. GDKJXM20212099).

References

[1] Building Energy Research Center of Tsinghua University, 2014 Annual Report on China Building Energy Efficiency (in Chinese), China Building Industry Press, Beijing, China, 2014. ISBN:9787112164837.

[2] R. Gao, S. Chen, A. Li, Z. Yang, B. Cong, Computational fluid dynamics study on the drag and flow field differences between the single and coupled bends, Build. Serv. Eng. Res. Technol. 38 (2) (2017) 163–175, https://doi.org/10.1177/0143624416670695.

[3] C.Y. Lee, An algorithm for path connections and its applications, IRE Trans. Electron. Comput. EC-10 (3) (1961) 346–365. ISSN: 0367-9950, https://doi.org/10.1109/TEC.1961.5219222.

[4] D.W. Hightower, A solution to line routing problems on the continuous plane, in: A. R. Newton (Ed.), DAC88: Design Automation Conference, Association for Computing Machinery, New York, NY, United States, 1988, pp. 11–34, https://doi.org/10.1145/62882.62883.

[5] R.G. Newell, An interactive approach to pipe routing in process plants, in: C. V. Freiman, J.E. Griffith, J.L. Rosenfeld (Eds.), IFIP Congress (2) 1971 Vol. 2, North-Holland Publishing Co., Ljubljana, Yugoslavia, 1971, pp. 1080–1085. ISBN: 0-7204-2063-6.

[6] J.-H. Park, R.L. Storch, Pipe-routing algorithm development: case study of a ship engine room design, Expert Syst. Appl. 23 (3) (2002) 299–309, https://doi.org/10.1016/S0957-4174(02)00049-0.

[7] J. Fan, M. Ma, X. Yang, Research on automatic laying out for external pipeline for aeroengine (in Chinese), J. Mech. Des. 20 (7) (2003) 21–23, https://doi.org/10.3969/j.issn.1001-2354.2003.07.008.

[8] H. Sui, W. Niu, Branch-pipe-routing approach for ships using improved genetic algorithm, Front. Mech. Eng. 11 (3) (2016) 316–323. ISSN: 2095-0233, https://doi.org/10.1007/s11465-016-0384-z.

[9] A. Asmara, U. Nienhuis, Automatic piping system in ship, in: H.T. Grimmelius (Ed.), Proceedings of the 5th International Conference on Computer and IT Applications in the Maritime Industries, COMPIT'06, IMarEST Benelux Branch, Oegstgeest, The Netherlands, 2006, pp. 269–280. ISBN-10: 90-810065-3-3, URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.354&rep=rep1&type=pdf (accessed on 09 March 2022).

[10] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks vol. 4, IEEE, Perth, WA, Australia, 1995, pp. 1942–1948. ISBN: 0–7803–2768-3, https://doi.org/10.1109/ICNN.1995.488968.

[11] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271, https://doi.org/10.1007/BF01386390.

[12] H. Kimura, Automatic designing system for piping and instruments arrangement including branches of pipes: multi-objective optimization of piping material costs and valve operability, J. Jpn. Soc. Nav. Archit. Ocean Eng. 14 (2011) 165–173. ISSN: 1880-3717, https://doi.org/10.2534/jjasnaoe.14.165.

[13] J.-G. Min, W.-S. Ruy, C.S. Park, Faster pipe auto-routing using improved jump point search, Int. J. Nav. Archit. Ocean Eng. 12 (2020) 596–604. ISSN: 20926782, https://doi.org/10.1016/j.ijnaoe.2020.07.004.

[14] Y. Hu, T. Jing, Z. Feng, X.L. Hong, X.D. Hu, G.Y. Yan, ACO-Steiner: ant colony optimization based rectilinear Steiner minimal tree algorithm, J. Comput. Sci. Technol. 21 (1) (2006) 147–152. ISSN: 1000-9000, https://doi.org/10.1007/s11390-006-0147-0.

[15] L. Luyet, S. Varone, N. Zufferey, An ant algorithm for the Steiner tree problem in graphs, in: M. Giacobini (Ed.), Applications of Evolutionary Computing vol. 4448, Springer, Berlin, Heidelberg, Berlin, 2007, pp. 42–51. ISBN: 978–-3-540-71804-8, https://doi.org/10.1007/978-3-540-71805-5_5.

[16] Q. Liu, C. Wang, A discrete particle swarm optimization algorithm for rectilinear branch pipe routing, Assem. Autom. 31 (4) (2011) 363–368. ISSN: 0144-5154, https://doi.org/10.1108/01445151111172952.

[17] Q. Liu, C. Wang, Multi-terminal pipe routing by Steiner minimal tree and particle swarm optimisation, Enterp. Inf. Syst. 6 (3) (2012) 315–327. ISSN: 1751-7575, https://doi.org/10.1080/17517575.2011.594910.

[18] B. Medjdoub, G. Bi, Parametric-based distribution duct routing generation using constraint-based design approach, Autom. Constr. 90 (2018) 104–116. ISSN: 09265805, https://doi.org/10.1016/j.autcon.2018.02.006.

[19] G. Carpaneto, M. Dell'Amico, P. Toth, Algorithm 750: CDT: a subroutine for the exact solution of large-scale, asymmetric traveling salesman problems, ACM Trans. Math. Softw. 21 (4) (1995) 410–415. ISSN: 00983500, https://doi.org/10.1145/212066.212084.

[20] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, Third edition, The MIT Press, London, England, 2009. ISBN-13: 978-0-262-03384-8.

[21] Y. Lu, S. Tang, W. Ji, M. Zhou, Q. Wang, Practical Heating and Air-conditioning Design Manual, China Building Industry Press, Beijing, China, 2008. ISBN: 978-7-112-09749-4.

[22] Z. Fan, The Principle and Design of Building Construction, Fourth edition, Tianjin University Press, Tianjin, China, 2009. ISBN978-7-5618-2040-7.