Automation in Construction 90 (2018) 104–116

Contents lists available at ScienceDirect

Automation in Construction

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

Parametric-based distribution duct routing generation using constraint-based design approach

Benachir Medjdoub[a,*], Gang Bi[b]

[a] The Creative and Virtual Technologies Lab, Nottingham Trent University, UK
[b] Roy Mark Technology (Shenzhen) Co. Ltd., China

A R T I C L E I N F O

Keywords:
Generative design
Distribution routing
Automation
Parametric model
Optimisation
Interactivity
Building services
Constraints

A B S T R A C T

In this paper, we present a generative design approach using constraint-based programming to handle the duct routing for ceiling mounted fan coil systems in buildings. This work utilises and builds on the result from previous approach using case-based reasoning and constraint satisfaction problem to deal with the space configuration of complex design problems for ceiling mounted fan coil systems in buildings. In this work, our approach automates the distribution routing using constraint-based approach. Comparatively to previous work, the system we have developed generates parametric-based models where further interactive modification and interaction is made possible for the end user. This approach has been tested in real case scenario working with our industrial partners.

1. Introduction

Standardisation is widely recognised as a key element in reducing the design time, cutting construction cost, and ensuring efficient design solutions. If the design of any complex artefact is suitably restricted by adhering to a library of predefined components and assembly details, it becomes possible to automate a great deal of the design process. In the case of pipe/duct routing, there are in any case good value-engineering reasons to use standard components and details. Barnard and Medjdoub and Medjdoub et al. [1,2] have shown that it is possible to define and implement standard solutions to produce designs comparable with the current practice. The main beneficiaries will be engineers, manufacturers, suppliers, building users, and managers. They all stand to benefit from standardised solutions, reduced capital costs, and improved performance. This work utilises and builds on the result from previous projects presented in Medjdoub et al. [3,4]. In [3] we have presented a hybrid approach using case-based reasoning and constraint satisfaction problem approach to deal with the space configuration of complex design problems for ceiling mounted fan coil systems in buildings.

This project is concerned with developing a new approach to deal with complex and combinatorial problems (NP-hard problem) in building services design and more precisely in pipe/duct routing. Currently, building services engineers solve these problems “by hand”. Starting from the fresh air load, a schematic solution is defined. The engineers proceed to equipment selection, and then equipment location, followed by pipe/duct routing governed by objective requirements (e.g. minimise number of bends, and minimise pipe length).

Algorithms to generate pipe/duct routing have been studied for > 50 years. In 1961 Lee [5] suggested the Maze algorithm which consists to divide a space into cells and labels and chooses the next cell until the target cell is reached. Next, Hightower [6] proposed the escape algorithm, also known as the line-search algorithm. More recent research tried to find the global optimum route path. Examples include an evolution-based algorithm [7–9] and an ant colony optimization scheme [10,11]. The target of route optimization is usually the minimum cost of the pipe routing path. In many studies, the cost consists of the pipe length cost and the cost of all bent parts, which require expensive bending fabrication or elbow fitting processes. Park [12], Kimura and Ikehira [13] and Ando and Kimura [14] also considered the operability costs such as the costs incurred to determine valve locations and safety clearances. However, there are still limitations when attempting to make use of the aforementioned approached to create a fully automatic routing system for actual building services design work. The main reason for this is that pipe routing algorithms generally do not consider the knowledge and the preference of the designer suitably as required in the actual design work. This type of limitation is not a matter purely related to the optimization algorithm itself. It is rather a matter of knowledge representation prior and during the design automation process. Therefore, the knowledge representation during the design phase is certainly becoming an important issue in the area of design automation. Moreover, from a practical point of view, it is also

* Corresponding author.
E-mail address: benachir.medjdoub@ntu.ac.uk (B. Medjdoub).

https://doi.org/10.1016/j.autcon.2018.02.006
Received 19 December 2016; Received in revised form 12 June 2017; Accepted 2 February 2018
Available online 22 February 2018
0926-5805/

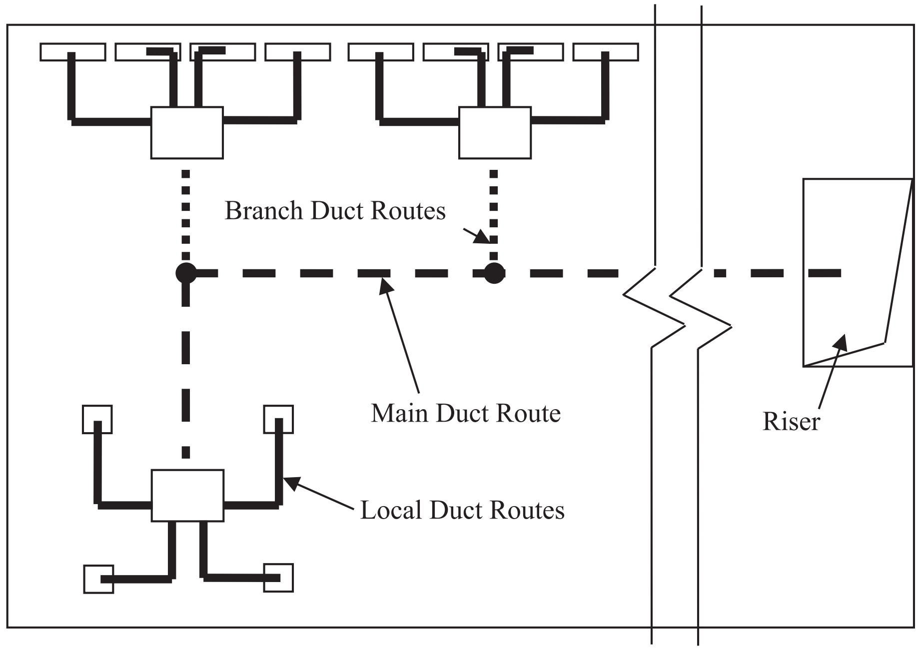

Fig. 1. Duct routing strategy with local ducts (solid lines), branch ducts (dot lines) and main duct (dash line).

important that the implemented routing algorithm can be utilized effectively in real case scenario and interfacing with building information modelling systems.

In this paper, we focus on the duct route generation for ceiling mounted fan coil systems. Three main tasks are carried out to generate the duct route (see Fig. 1), which are: (a) Main duct routing between the ventilation riser and the selected fan coil; (b) Branch duct routing between the selected fan coil and start point of precedent duct; and (c) Local duct routing between the fan coils and the diffusers. Our approach automates the distribution routing using constraint-based approach. Comparatively to previous work: (1) the system we have developed generates parametric-based models where further interactive modification and interaction is made possible for the end user; (2) This approach has been tested in real case scenario working with our industrial partners; (3) The software prototype has been implemented imbedded in a BIM system (Microstation form Bentley Systems).

In Section 2 we present the distribution routing knowledge base. Next, in Sections 3 and 4, we describe the constraint model and the solution generation approach. The interactive parametric-based user interface and the benchmarking exercise are presented in Sections 5 and 6. Finally, the conclusion is presented in Section 7.

2. Knowledge model

The knowledge model holds the main pipe/duct classes. Each defined class is characterised by a set of attributes and class constraints. Based on the inheritance mechanism, there are two sub-classes of the duct/pipe class including: rectangular duct/pipe class and circular duct/pipe class. The rectangular duct/pipe class represents the duct with rectangular section, and its sub-classes include branch duct class and main duct class. The circular duct/pipe class represents the duct or pipe with circular section, and its sub-classes include local duct class and pipe class. Fig. 2 illustrates the class structure of our knowledge model. The knowledge and constraint models have been implemented in Java using the constraint library JSolver. We have used Java Native Interface and MicroStation's API method to access Bentley DGN files in order to interface the knowledge and constraint models with the BIM model.

2.1. Duct/pipe class

The duct/pipe class is characterised by a zigzag line defined by a polyline including bends and segments. As illustrated in Fig. 3, the attributes of the duct/pipe class are: (1) The zigzag line defined by a set of joints (N joints) characterised by 3D point coordinates (X_i, Y_i, Z_i); (2) Number of segments (Ns) where each pair of successive joints defines a segment; (c) The number of bends between two successive segments (N_b); The bending angle θ_i (where $i \in [0, n-3]$) between two adjacent segments where $\theta_i \in [90^\circ, -90^\circ]$; and the total length $L \in [L\text{min}, L\text{max}]$. All these attributes are integer-constrained variables. We used an arc-consistency on integers constraint programming technique which explains the need for a distance increment; but this is not too limitative as engineer and architects are used to reasoning with dimensional modules.

The two following class constraints have been defined to ensure the geometrical consistency of the duct/pipe class:

- (c1) $N_b = N - 2$
- (c2) $N_s = N - 1$

A modification of a variable domain composing the constraint (c1) or (c2) entails the modification of variable domains of the other related variables, thanks to the *arc-consistency on integers* that we used. Arc-consistency technique asserts that these constraints will always be respected for a specific instantiation and try to rule out variable domain values which have no chance to be in a solution. But this technique does not reduce a domain variable to its minimal size; solutions are complete but they are not all consistent. This is a problem we will have to deal with when generating *the solutions*.

2.2. Duct/pipe sub-classes

We know that a local duct connects the diffuser with the fan coil and a branch duct connects the selected fan coil with the duct, while the main duct connects the selected fan coil with the riser. The duct/pipe sub-classes (i.e. the branch and main duct) have additional parameters such as W (width) and H (height) to represent the section size of each segment, meanwhile, the local duct and the pipe have additional parameter such as R (radius) to represent the circular size of the