

# **Parametric-based Distribution Duct Routing Generation Using Constraint-Based Design Approach**

Benachir Medjdoub\* and Gang Bi<sup>2</sup>

\*The Creative and Virtual Technologies Lab  
Nottingham Trent University, UK

<sup>2</sup>Roy Mark Technology (Shenzhen) Co. Ltd, China

---

**Abstract:** 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.

**Keywords:** Generative design; distribution routing; automation; parametric model; optimisation; interactivity; building services; constraints

## **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 more than 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 [7,8,9] 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 approaches 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 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 Figure 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 from Bentley Systems)

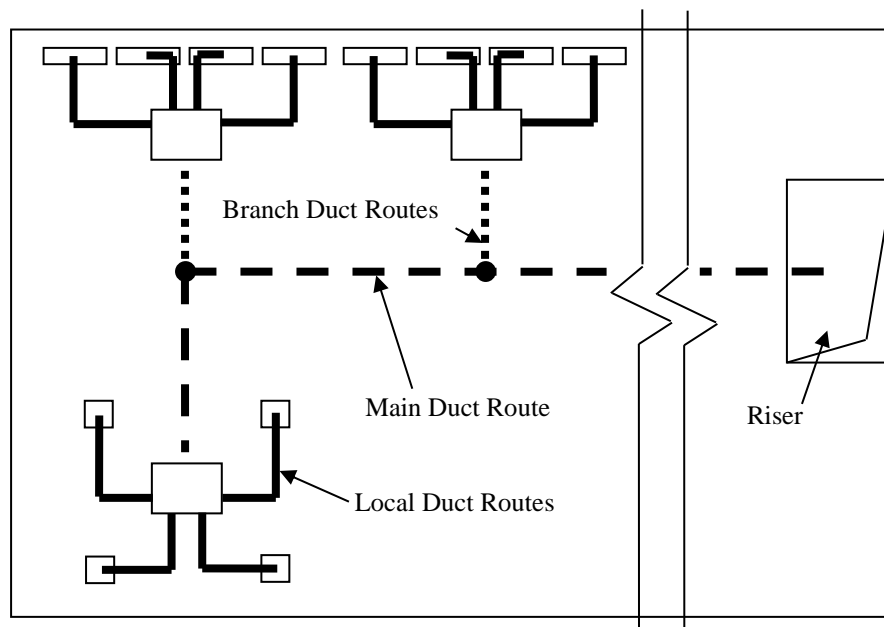


Figure 1: *Duct routing strategy with local ducts (solid lines), branch ducts (dot lines) and main duct (dash line)*

In sections 2 we present the distribution routing knowledge base. Next, in section 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 characterized 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. Figure 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.

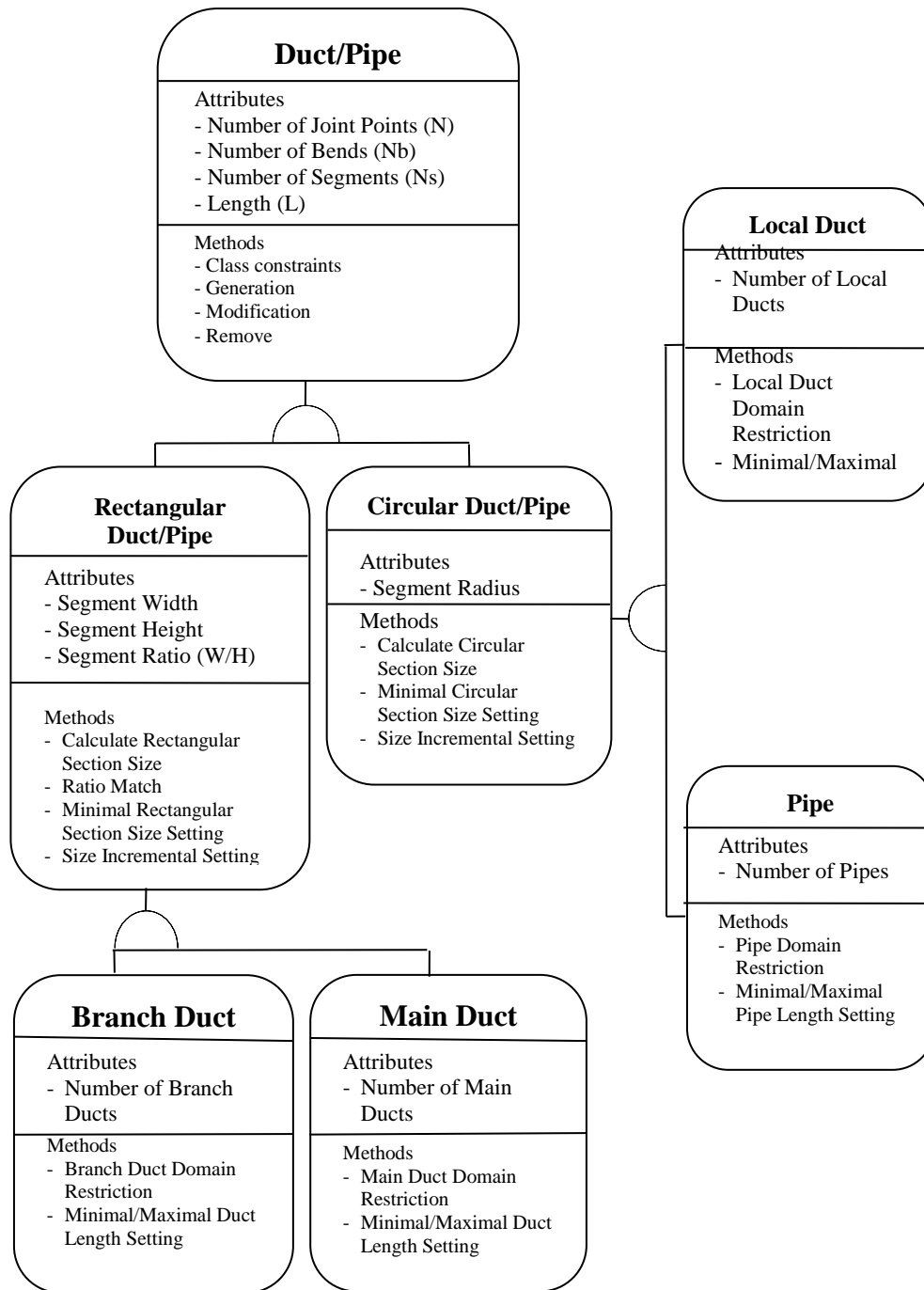


Figure 2: Duct/pipe object structure and the mechanism of inheritance

## 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 Figure 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 ( $N_s$ ) where each pair of successive joints defines a segment; (c) The number of bends between two successive segments ( $N_b$ ); The bending angle  $q_i$  (where  $i \in [0, n-3]$ ) between two adjacent segments where  $q_i$

$\theta \in [90^\circ, -90^\circ [$ ; and the total length  $L \in [L_{\min}, L_{\max}]$ . 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.

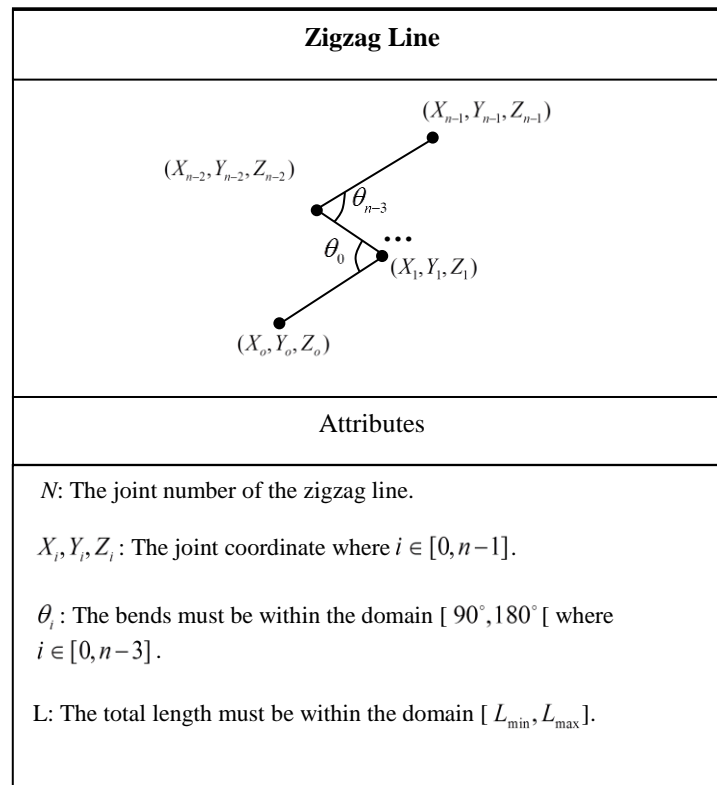


Figure 3: The attributes of the zigzag line

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 segment section. Figure 4 illustrates  $P_0P_1$ ,  $P_1P_2$  and  $P_2P_3$  segments of a duct/pipe, where  $R$ ,  $W$  and  $H$  of the duct/pipe segments will depend on the flow rate passing through that segment.

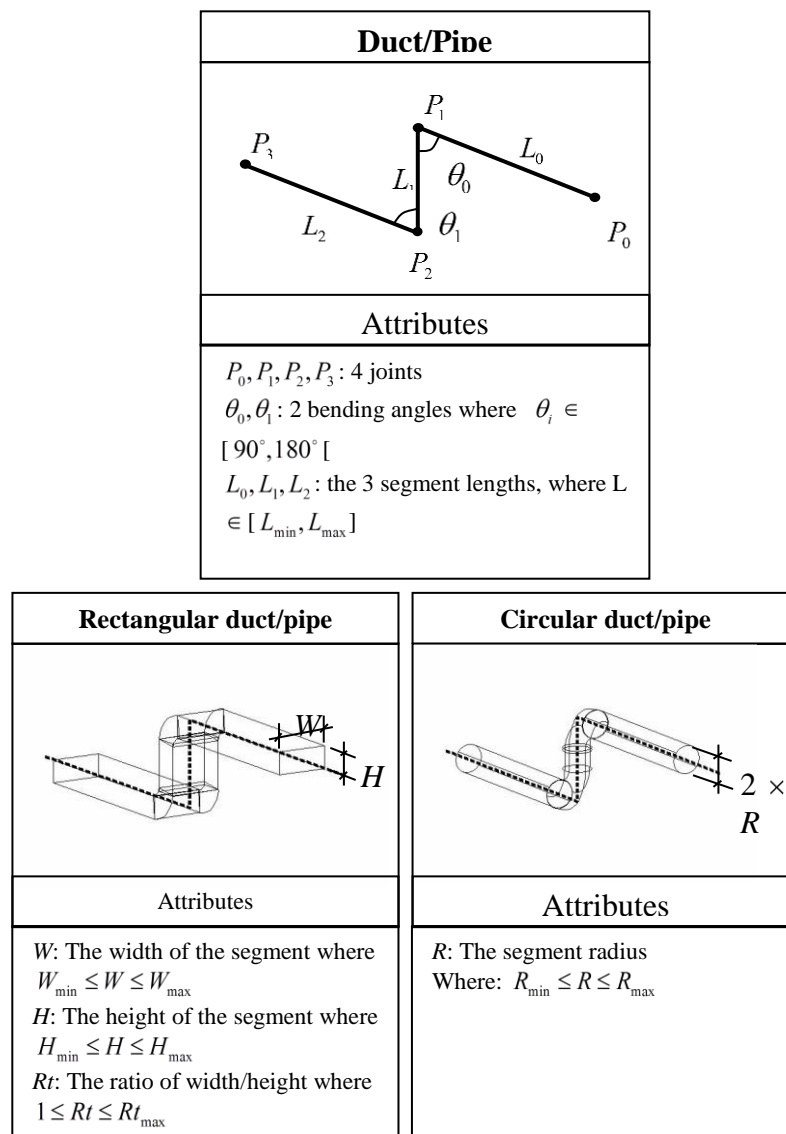


Figure 4: Duct/pipe with a set of 4 joints, 3 segments and 2 bending angles

## 3 The constraint model

Our constraint model makes the distinction between design *specification constraints* and *implicit constraints* which are implicitly applied during the solution search process in order to reduce the combinatorial complexity.

### 3.1 Design specification constraints

These constraints include dimensional and topological constraints over the pipes/ducts, fan coil equipment (fans, diffusers, etc.) and the building components (beams, walls, columns, etc.). Dimensional constraints are applied to the attributes of a single object whereas topological constraints are applied between two or more architectural objects and include inclusion constraint and non-overlapping constraints.

#### 3.1.1 Dimensional constraints

Dimensional constraints are applied to set the position of the starting and ending joints of the duct. For example, the starting joint of a local duct is fixed and corresponds to the ventilated outlet point of a fan coil. The end joint of the local duct is also fixed and corresponds to the inlet point of a diffuser. The relevant dimensional constraints can be defined as:

$$Duct.P_{start}(x, y, z) = FCU.P_i(x, y, z) \text{ where } i \in FCU.Pn$$

$$Duct.P_{end}(x, y, z) = Diffuser_i.P(x, y, z) \text{ where } i \in FCU.Diffn$$

Where  $Duct.P_{start}(x, y, z)$  = the start joint of the local duct

$Duct.P_{end}(x, y, z)$  = the end joint of the local duct

$FCU.P_i(x, y, z)$  = the outlet point  $i$  of the fan coil

$FCU.Pn$  = the number of the outlets of the fan coil

$Diffuser_i.P(x, y, z)$  = the inlet point of diffuser  $i$

$FCU.Diffn$  = the number of the diffusers per fan coil



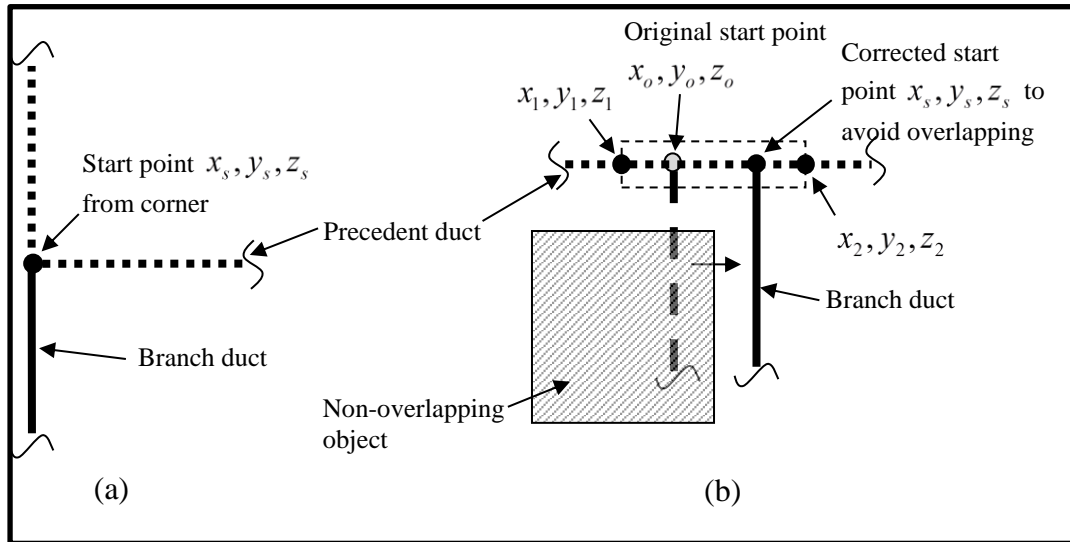


Figure 5: (a) Start joint of the branch duct (solid line) (b) Start joint of the branch duct (solid line)

### 3.1.2 Inclusion constraint

The inclusion constraint defines a 3D domain area where the duct joints must be located. It is a conjunction of six inequalities. The inclusion constraint is defined as:

$$Duct.X \geq Space.X - \frac{Space.L}{2};$$

$$Duct.X \leq Space.X + \frac{Space.L}{2};$$

$$Duct.Y \geq Space.Y - \frac{Space.W}{2};$$

$$Duct.Y \leq Space.Y + \frac{Space.W}{2};$$

$$Duct.Z \geq Space.Z - \frac{Space.H}{2};$$

$$Duct.Z \leq Space.Z + \frac{Space.H}{2};$$

Where  $Duct.X$  &  $Duct.Y$  &  $Duct.Z$  = the 3D coordinates of the joint

$Space.X$  &  $Space.Y$  = the 3D reference point of the space

$Space.L$  = the length of the space

$Space.W$  = the width of the space

$Space.h$  = the height of the space

All the variables are constrained variable on integer

### 3.1.3 Non-Overlapping Constraints

It is complex to solve a non-overlapping constraint between the duct and the other objects (equipment, building components, etc.), especially when the duct is oblique. If the duct lays out in the right direction (follow  $X$ -axis and/or  $Y$ -axis) and the non-overlapping objects have rectangular shapes, we can apply the non-overlapping constraints with 4 orientations  $\{North, East, South, \& West\}$  as described in Medjdoub and Yannou [15,16]. This constraint is a disjunctive constraint generating four possibilities and consequently increasing the complexity of the problem. Orthogonal geometry characterises the majority of the cases, but if the duct lays out in the oblique direction, we have developed the following constraints:

- **Non-overlapping constraint between an oblique duct and an object**

If the object has a rectangular shape, it will be characterised by its length  $L$  and width  $W$ . based on the geometrical construction illustrated in Figure 6(a). The duct will not overlap with the object if the distance  $O_1O_2$  between the central point  $O_1$  of the

rectangle and the duct segment  $D_1D_2$  is greater than  $\sqrt{\left(\frac{L}{2}\right)^2 + \left(\frac{W}{2}\right)^2}$ ;  $O_1O_2$  being

orthogonal to the duct. This constraint will remove few possible solutions as in fact the duct will not overlap with the enclosed circle but has the advantage to be a conjunction while decreasing the complexity. The same approach will be used to a non-overlapping constraint with an object with a non-orthogonal geometry where we use its enclosed circle to not overlap with the duct  $D_1D_2$  (see Figure 6(b))

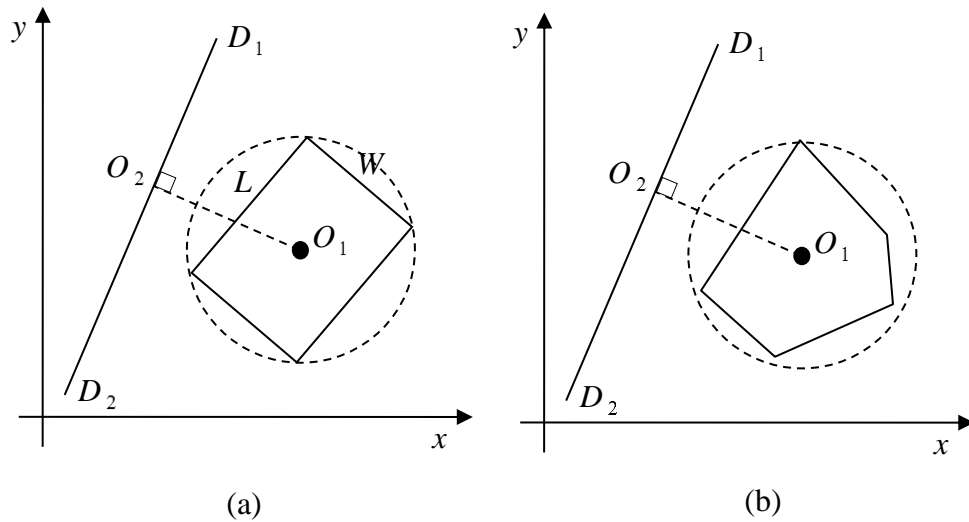


Figure 6: (a) non-overlapping between the duct and the square object, (b) non-overlap between the duct and the non-square object

The two duct segment do not intersect (see Figure 7) if and only if one segment lies entirely to one side of the line containing the other duct segment, this can be expressed in the following conjunctive constraints using the notion of a scalar cross product:

$$\vec{P_4P_2} \times \vec{P_4P_3} > 0 \ \& \ \vec{P_4P_1} \times \vec{P_4P_3} > 0 \ \& \ \vec{P_4P_2} \times \vec{P_2P_1} > 0 \ \& \ \vec{P_2P_3} \times \vec{P_2P_1} > 0$$

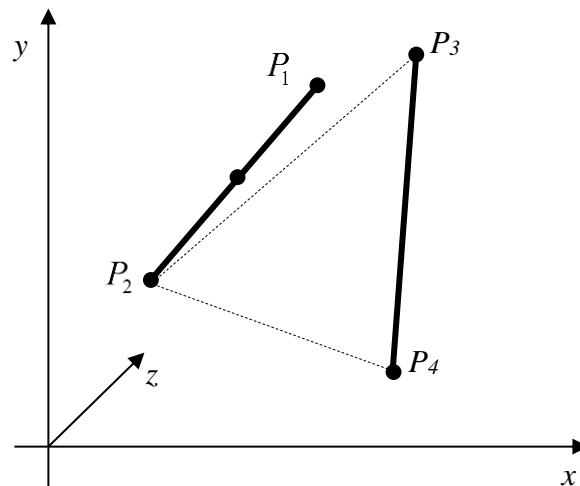


Figure 7: The non-overlapping constraint between two oblique duct segments

### 3.2 Approach to reduce the number of non-overlapping constraints

Non-overlapping constraints are disjunctive constraints increasing the complexity of the NP-complete problem. Therefore, we have suggested an approach to reduce, a

priory and before the solution search strategy, the number of non-overlapping constraints to reduce this complexity.

Non-overlapping constraints are applied between the equipment and the building components such as column, beams etc. It is clear that if we can reduce the number of those elements which should not overlap each other; we will decrease the complexity of the problem. One possibility we have explored is to group a set of small objects such as one fan and a set of diffuser in one cluster. This will have the advantage to apply the non-overlapping constraint between the duct and a set of clusters rather than every object and consequently reducing the complexity of the problem. Figure 8 illustrates an example: First, we combine the fan coil and its diffusers into one cluster. As we can see from figure 8, each fan coil and its diffusers are enclosed by a rectangle (dashed lines), and the distributed duct must not overlap with that rectangle instead of treating the fan coil and the diffusers separately. Therefore, five non-overlapping constraints (one fan coil and its four diffusers) are reduced into one.

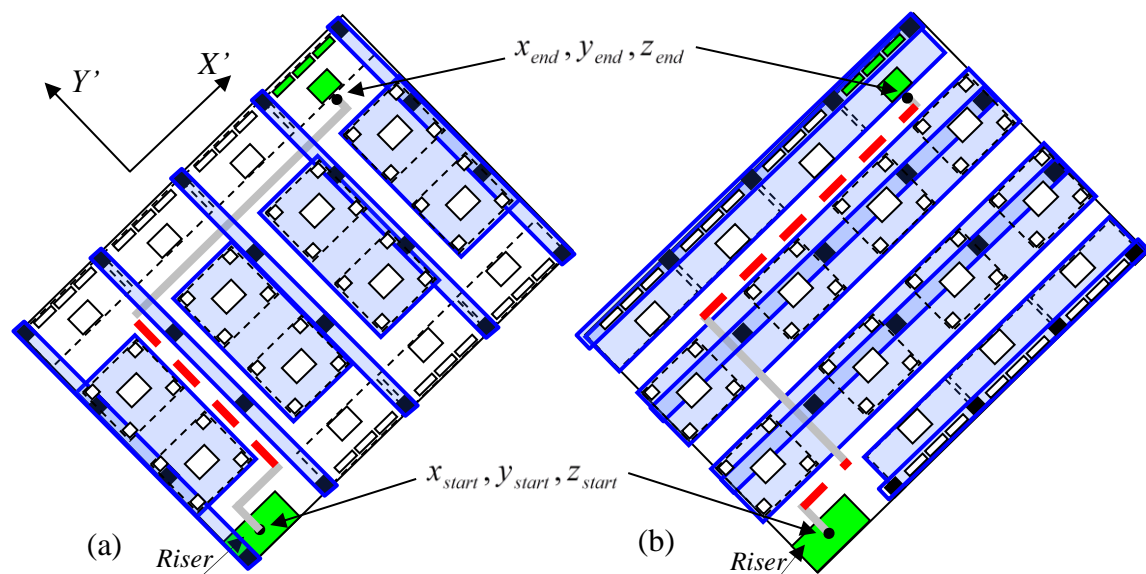


Figure 8: (a) 4 combined internal fan coil/diffuser clusters and 5 column clusters in the  $Y'$  direction, (b) 4 combined fan coil/diffuser clusters and 4 column clusters in the  $X'$  direction

Next, we found that these fan coil/diffuser cluster number still too high, and in order to reduce the number of clusters furthermore, we have combined clusters into bigger clusters. To do so, we have used the duct routing direction to optimise the problem.

In our approach and based on real practice, the duct route tends to follow the  $X'Y'$  axis

directions ( $X'$  and  $Y'$  are relative axis directions to the floor layout, see Figure 8). If we can group the clusters along the  $X'$  and  $Y'$  directions separately the number of non-overlapping constraints can be reduced furthermore. As indicated in Figure 8a, the red dash-line represents duct in the  $Y'$  axis direction. Next, we group all the objects within the  $Y'$  axis direction in one cluster and constrain it to not overlap with the duct. In this particular example, thirty-five original non-overlapping constraints (constrained with fifteen fan coil/diffuser groups and twenty columns in the figure) have been reduced to nine non-overlapping constraints. Figure 8b illustrates similar approach on the  $X'$  direction.

#### **4. Solution Generation**

The process of distribution routing consists of generating duct routes and sizing them. As described in section one, the generation process is done sequentially as follow:

1. Main duct routing between the ventilation riser and the selected fan coil
2. Branch duct routing between the selected fan coil and start point of precedent duct.
3. Local duct routing between the fan coils and the diffusers

##### 4.1 Main duct and local duct generation:

We proceed first by instantiating the number of bends, followed by the orientation and finally we minimize the length. We have used the “branch and bound” algorithm [17] which is the most common approach used in constraint programming to find the optimal solution. The procedure is described as: First, we create a constrained variable representing the objective function (total length) and find an initial solution. Then we introduce a new constraint that the value of the objective variable must be better than in the initial solution. We repeatedly solve the new problem and tighten the constraint on the objective variable until the problem becomes insoluble, and the last solution found is then the optimal solution. Each routing procedure for local and main ducts has its own 3D space domain and a set of objects to not overlap with. For the local duct, the 3D space domain is located strictly between the fan coil and the diffuser.

##### 4.2 Branch duct generation

Following the main duct generation, the system will calculate the number of connections and their positions on the main duct. This will be followed with the same

generation algorithm used for the main ducts. For the branch duct, the domain is located strictly between the precedent duct outlet and the fan coil.

## 5. Interactive parametric-based user interface

Our approach has been implemented as a plugin to Microstation from Bentley System. The constraint library and the generation algorithms have been implemented using JSolver constraint library. In this section, we present the interactivity of the system through a real case scenario consisting of an office floor with a rectangular shape (see Figure 9). In this example we will describe the system we have developed and its interactivity.

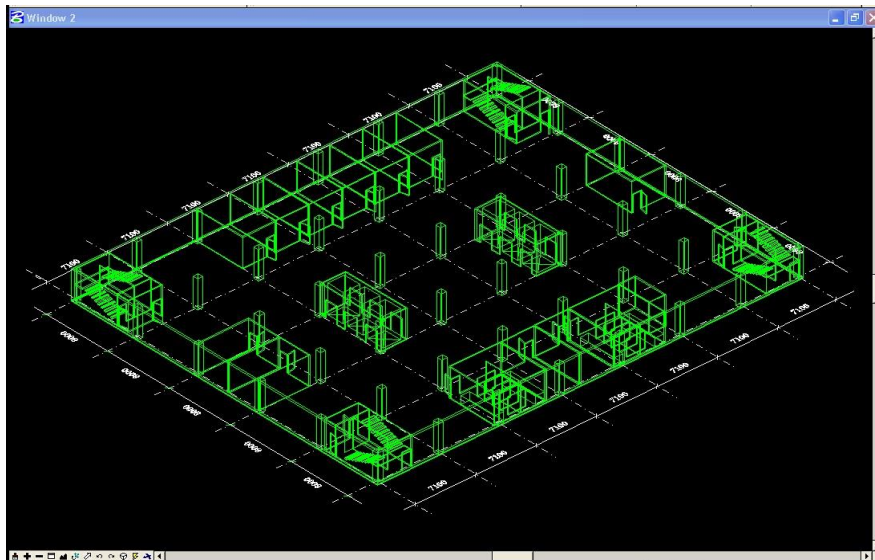


Figure 9: 3D BIM model (Microstation) of the office floor with rectangular shape

The process includes two stages: (1) first the system automatically size and locate the fans and the diffusers in the 3D ceiling void. We have used case-based reasoning approach combined with constraint-based adaptation process to generate the 3D layout of the equipment (i.e. fans and diffusers). This approach (See figure 10) has been presented in Medjdoub and Chenini [3]; (2) In stage two, the system will generate automatically the duct routing. This is a staged process generating first the main ducts followed by branch ducts and ending by local ducts (see figure 11).

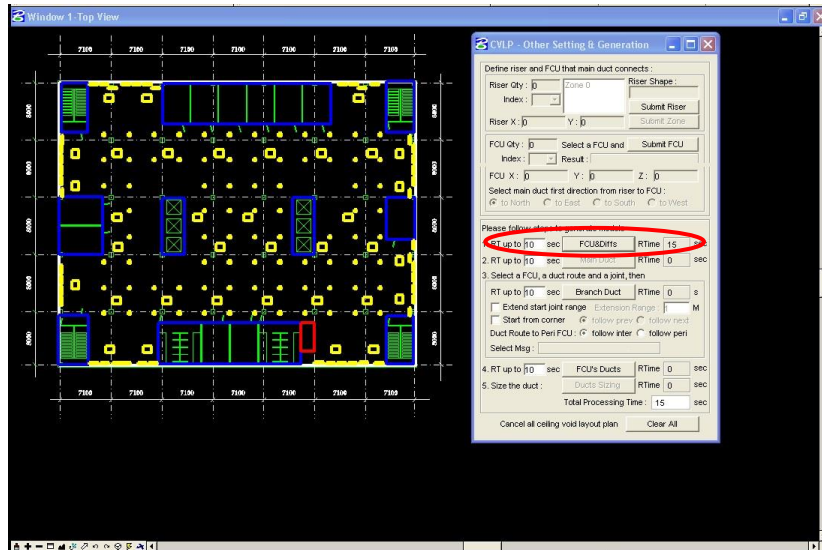


Figure 10: Output from stage 1 including the 3D layout of fans and diffusers

Before generating the main duct. The user will select interactively from the BIM model the riser and the farthest fan coil. The system will then generate a set of solutions from which the user can select one as indicated in Figure 11.

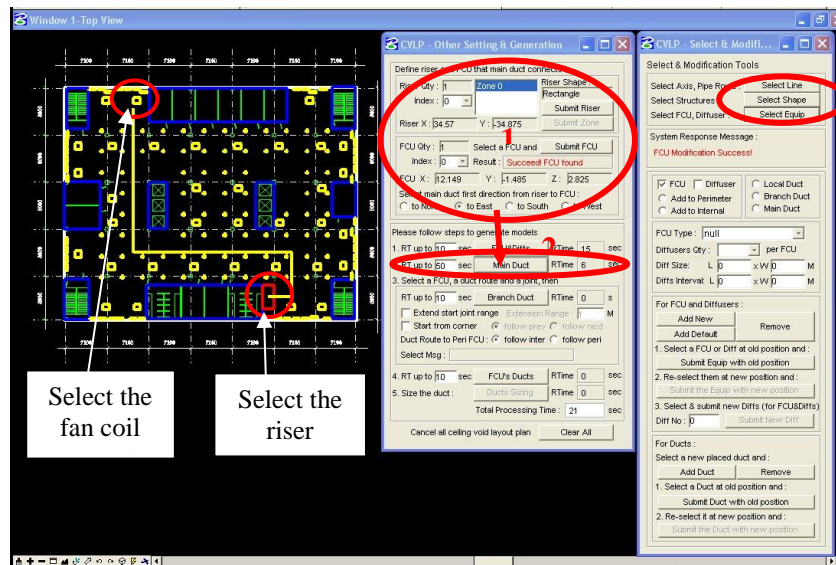
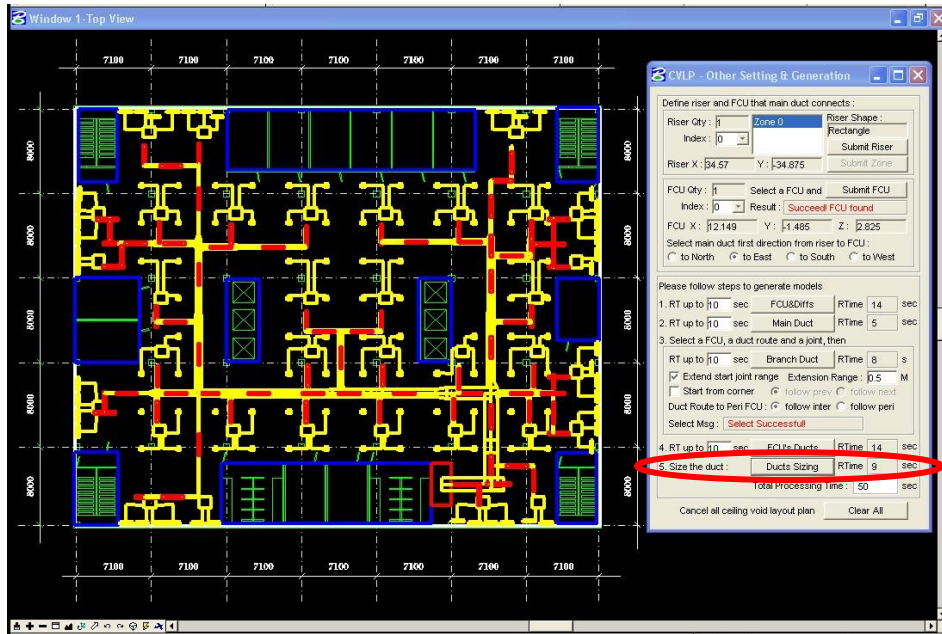


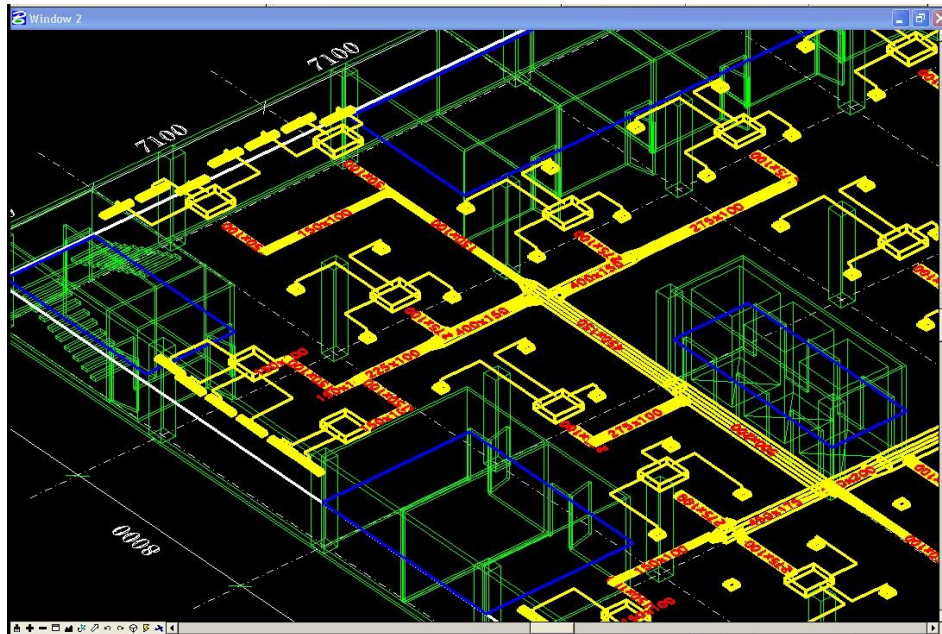
Figure 11: Riser, fan coil selection and main duct routing

Next, the branch duct and local duct will be generated sequentially using the same generation algorithms, this will have the advantage to divide the complex problem in a set of sub-problems more likely to be generated in a reasonable response time.

Figure 12 display the final solutions of the office building.



(a)



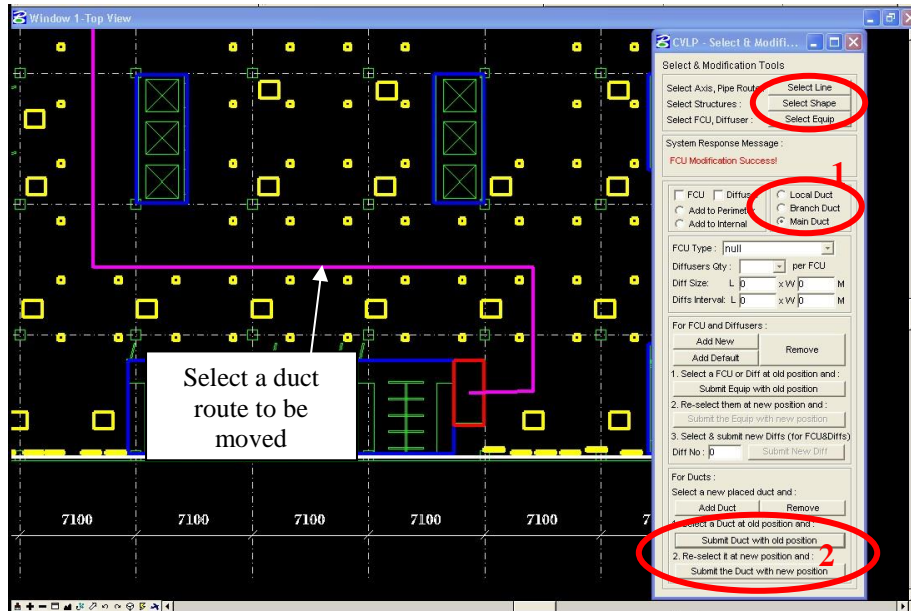
(b)

Figure 12: (a) duct generation, (b) isometric view of the BIM solution

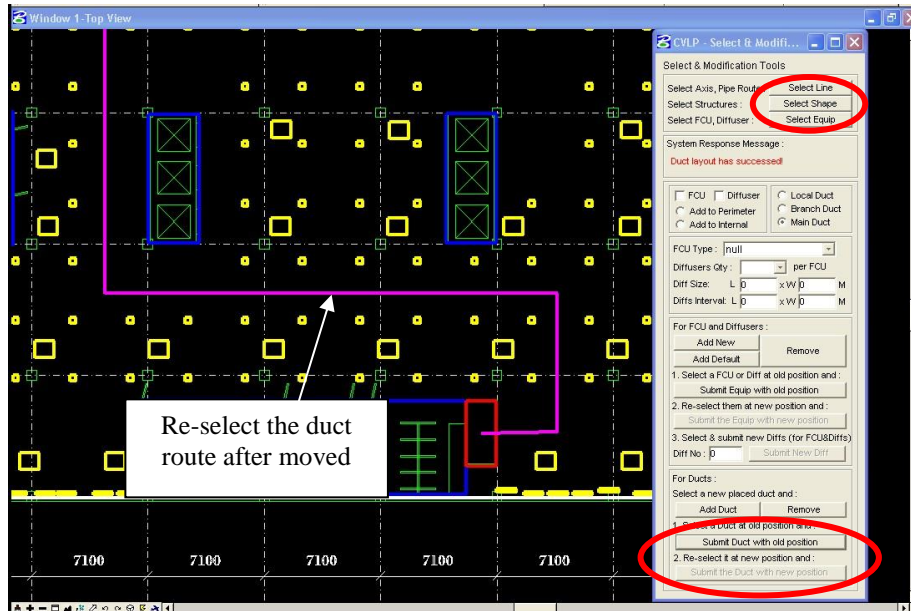
The generated solution is an instantiation of the none-oriented graph of constraints as defined in section 4 and corresponding to an oriented graph of constraints defining a parametric model. Comparatively to previous works on duct/pipe routing, this approach has an advantage in offering the possibility to the user/engineer to improve the generated solution interactively. Thus as indicated in figure 13, if the user is not completely satisfied with the solution (Figure 13a), he/she can just by dragging routine,



for example, move the duct up. The system will update the solution automatically (see Figure 13b). Any change of routing is driven by the parametric model of the solution. The clashes are solved by the automatic application of a non-overlapping constraint (see section 3.1.3) between any modified or additional duct/service and the surrounding objects.



(a)



(b)

Figure 13: (a) Select and move the duct up, (b) The system will update automatically the new solution

The final solution includes the full information model including the type of equipment size and location, the ducts type, size and location, and manufacturer information.

## 6. Testing and evaluation

This approach has been evaluated using two other real case scenario by the engineers of our industrial partners AECOM. We have undertaken a qualitative evaluation, which is more appropriate to check the validity of the solution generated by our system. Thus, the engineers have assessed the solution by keeping the good duct routes and modifying the ones needing improvement. The final result (Table 3) shows that the solution can be compared with the ones produced by the engineers where between 4 and 8% improvement was required. The engineers have also suggested some improvements to the system.

**Example 1:** Figure 14 illustrates the plan of an office building we have used as our first case study. Figure 15a illustrated the solution generated by our system and figure 15b illustrates the solutions generated by the engineers. Table 1 shows the response time of the solution.

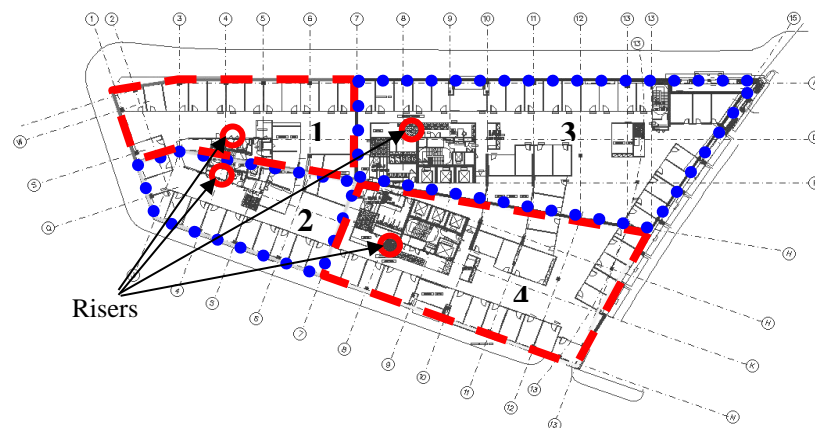
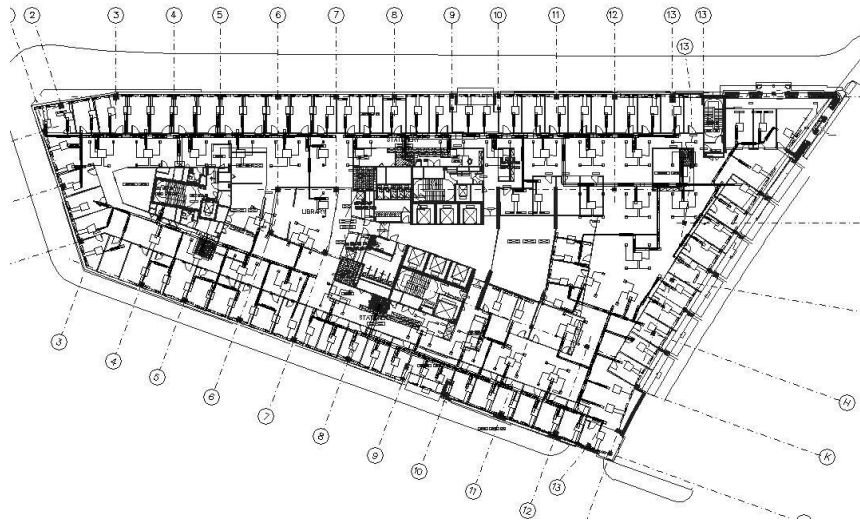
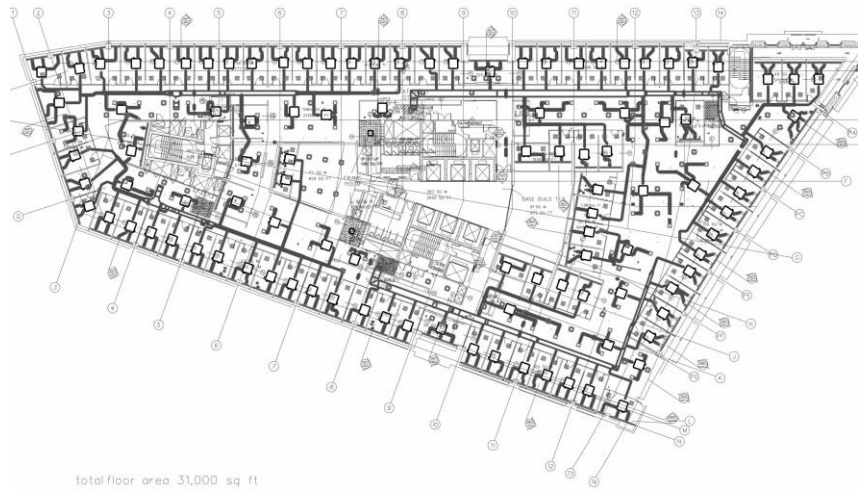


Figure 14: *Floor zoning (4 zones)*



(a)



(b)

Figure 15: (a) Solution generated by our system, (b) Solution generated by the engineers

Table 1: Processing time (PC – quad Core Processor i5-6400 (3,2GHz))

<b>Main Duct</b>	
- Processing Time for Facility Layout	101 sec
<b>Branch Duct</b>	
- Processing Time for Facility Layout	66 sec
<b>Local Duct for FCU</b>	
- Processing Time for Facility Layout	21 sec

**Example 2:** Figure 16 illustrates the plan of an office building we have used as our second case study. Figure 17a illustrated the solution generated by our system and figure 17b illustrates the solutions generated by the engineers. Table 2 shows the response time of the solution.

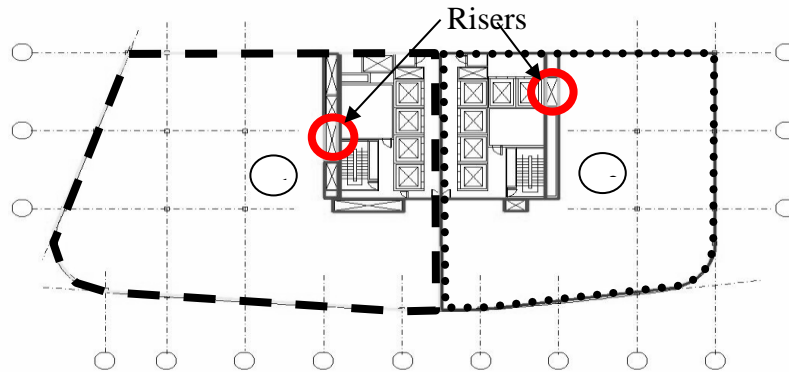
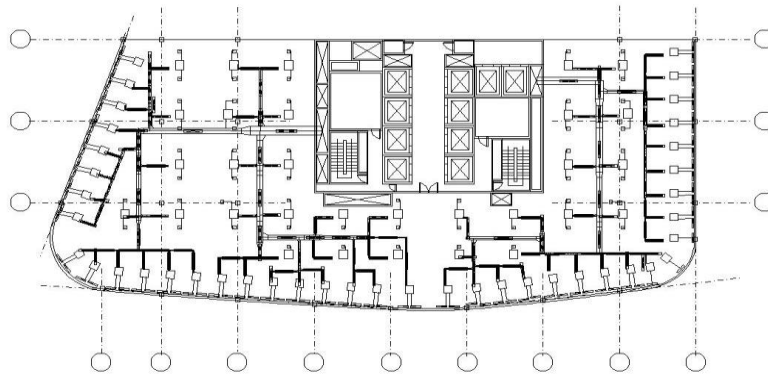
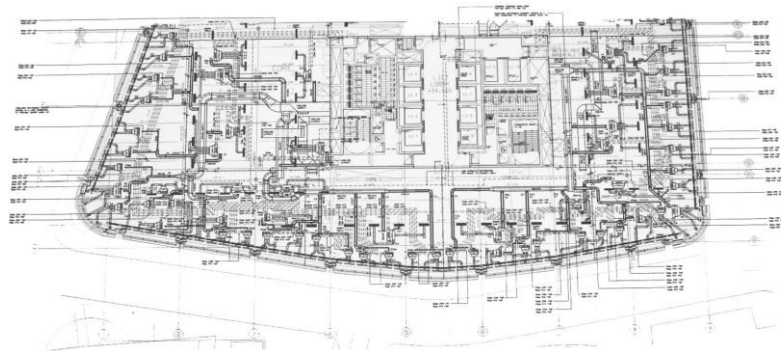


Figure 16: Floor zoning (2 zones)



(a)



(b)

Figure 17: (a) Solution generated by our system, (b) solution generated by the engineers

Table 2: Processing time (PC – quad Core Processor i5-6400 (3,2GHz))

<b>Main Duct</b>	
- Processing Time	44 sec
<b>Branch Duct</b>	
- Processing Time	35 sec per duct
<b>Local Duct for FCU</b>	
- Processing Time	17 sec per duct

Table 3: Qualitative evaluation of the solutions.

	Duct type	Engineers' Evaluation	
		Result	To improve
1	Main Duct Layout (duct connects riser and selected fan coil)	- 96% of branch duct routes are good. - The engineer improved interactively 4% of the branch routes.	The response time is too long (2min) with more complex configuration (250 objects and more).
2	Local Duct Layout (duct connects fan coil and diffuser)	- 93% of the branch duct routes are good. - The engineer improved interactively 8% of the branch routes.	
3	Branch Duct Layout (duct connects precedent duct and selected fan coil)	- 92% of branch duct routes are good. - The engineer improved interactively 8% of the branch routes.	Duct end points should be near the central inlets of fan coils.

Another limitation of our system is due to the complexity of the building geometry. Thus the pipe routing in non-rectangular shape is less accurate. This is mainly due to the complexity of the algorithm used and

## **7. Conclusion**

The approach presented in this paper shows an interactive system to support *pipe/duct routing in a fan coil system*. The prototype developed is simple to use providing interactive modification functionality of the 3D parametric model. This approach has shown the potential to significantly reduce design costs by reducing design time by 50% and improve the quality of the solution. The solution can produce additional benefits elsewhere in the supply chain while providing the full building information model of the technical solution including equipment and pipes/ducts information, their type, specifications, size, location and manufacturer.

On the computational side, the constraint-based design approaches used did achieve a good synergy between the size/complexity of the problem, the geometry of the building, and the response time, while producing results that are comparable with the Engineers' solutions.

However, we experienced some limitations when dealing with buildings with complex geometries (e.g. curves), more the building geometry is complex more the response time is higher. This is due to the non-overlapping constraints used in the case of non-orthogonal geometries which introduce more disjunctions (i.e. a node which increases the size of the tree search). Another advantage of this approach is the generation of parametric models providing the user the possibility to further improve the solution interactively.

Finally, the methodology developed can be transferable to other BIM platforms; our approach has been implemented as a plugin totally independent from any BIM application. As mentioned previously, we have used JNI and Microstation's API to interface our application and have access to the DGN file and the BIM model on Microstation. We can therefore use the same methodology to interface our application with existing BIM platforms.

## **References**

- [1] Barnard B., Medjdoub B. and Richens P., "Building Services Standard Solutions Implemented in CAD", CIBSE Journal, December 2001.

- [2] Medjdoub B., Richens P. and Barnard B. "Generation of Variational Standard Plant Room Solutions". *Automation in Construction Journal*, 12(2), Elsevier, p. 155-166, 2003.
- [3] Medjdoub, B. and M Chenini, 2015, "A constraint-based parametric model to support building services design exploration", in *Architectural Engineering and Design Management*, Taylor & Francis, P123-136, Volume 11, Issue 2, 2015.
- [4] Medjdoub B., "Constraint-based Adaptation for Complex Space Configuration in Building services", In *ITCon Journal*, Volume 243/2007, p.627-636, 2009.
- [5] Lee, C.Y., 1961. An algorithm for path connections and its applications. *IEEE Trans on Electron Computers*, 10(3), pp.346-365.
- [6] Hightower, D.W., 1969. A solution to line routing problems on the continuous plane. *Proceedings of Sixth Annual Design Automation Conference*. IEEE, pp.1-24.
- [7] Ito, T., 1999. A genetic algorithm approach to piping route path planning. *Journal of Intelligent Manufacturing*, 10, pp.103-114.
- [8] Ikehira, S., Kimura, H. and Kajiwara, H., 2005. Automatic design for pipe arrangement using multi-objective genetic algorithms. *International Conference on Computer Applications in Shipbuilding (ICCAS)*. Busan, Korea, 23-26 August 2005, 2, pp.97-110.
- [9] Kimura, H., 2011. Automatic designing system for piping and instruments arrangement including branches of pipes. *International Conference on Computer Applications in Shipbuilding (ICCAS)*. Trieste, Italy, 20-22 September 2011, 3, pp.93-99.
- [10] Xiaoning, F., Yan, L. and Zhuoshang, J., 2006. The ant colony optimization for ship pipe route design in 3D space. *World Congress on Intelligent Control and Automation (WCICA)*. Dalian, China, 21-23 June 2006, pp.3103-3108.
- [11] Xiaoning, F., Yan, L. and Zhuoshang, J., 2007. Ship pipe routing design using the ACO with iterative pheromone updating. *Journal of Ship Production*, 23(1), pp.36-45.
- [12] Park, J.H., Storch, R.L., 2002, Pipe-routing Algorithm Development: Case Study of a Ship Engine Room Design, *Expert Systems with Applications*, 23, 299-309.
- [13] Kimura, H. and Ikehira, S., 2009. Automatic pipe arrangement design considering operability valve operability. *International Conference on Computer*

Applications in Shipbuilding (ICCAS). Shanghai, China, 1-3 September 2009, 2, pp.121-126.

- [14] Y. Ando, and H. Kimura, H. An automatic piping algorithm including elbows and bends. Proceeding of the International Conference on Computer Applications in Shipbuilding (ICCAS). Trieste, Italy, 20-22 September 2011, 3, pp.153-158.
- [15] Medjdoub B. and Yannou B., "Separating topology and geometry in space planning". *Computer Aided Design*; 32(1), Elsevier, p. 39-61, 2000.
- [16] Medjdoub B. and Yannou B., "Dynamic space ordering at a topological level in space planning", *Artificial Intelligence in Engineering*, 15(1), Elsevier, p. 47-60, 2001.
- [17] Carpaneto, G., Dell'amico, M., Toth, P., 1995, "A branch-and-bound algorithm for large scale asymmetric travelling salesman problems", *ACM Transactions on Mathematical Software*, 21 (1995), p. 410–415.