ICCEPM 2017
The 7th International Conference on Construction Engineering and Project Management
Oct. 27-30, 2017, Chengdu, China

A BIM-based Automated Framework for Formwork Planning on Construction Sites

Maozeng Xu[1], Zhongya Mei [1*], Yi Tan [2]

[1] *School of Economics and Management, Chongqing Jiaotong University, Chongqing 400074, China.*
[2] *Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.*
E-mail address: meizy@mails.cqjtu.edu.cn

Abstract: Considering its significant impact on the cost and schedule of construction projects, formwork as one part of temporary facility categories in construction should be arranged precisely. Current practice in the formwork planning is often conducted manually and repetitively, causing low efficiency and time waste. This study proposes an automated framework to generate more accurate and detailed formwork plans by utilizing information from building information modeling (BIM) considering the adequate geometric and semantic information provided by the BIM model. The dimensions and quantities information of elements in a building can be extracted automatically. Then, a rule is prepared for calculating the required forms erected around elements based on the contact areas. Finally, an algorithm of integrating first fit decreasing (FFD) with coordinated bottom left (CBL) is applied to automatically generate the formwork plan. The BIM-based automated planning framework is demonstrated by an illustrative example. The results show that the proposed framework can generate the formwork plan accurately and automatically, and significantly improve the efficiency in the formwork plan and reuse.

Key words: concrete formwork plan, building information modeling (BIM), first fit decreasing (FFD), coordinate bottom left (CBL)

1. INTRODUCTION

Formwork is the total system of support for freshly placed concrete, including the mold or sheathing that contacts the concrete and all supporting members, hardware, and necessary bracing. Formwork system has significant impact on the cost and schedule of construction projects. The research report by ACI committee [1] showed that the cost of formwork in the United States could be as much as 60% of the total cost of the complete concrete structure in place. Huang et al. [2] also pointed out in their research that the formwork erection in a reinforced concrete building projects takes up to 15% of the total construction cost or 1/3 of the total concrete cost, and the formwork system may be characterized as a labor-intensive and time-consuming operation. The increasing acceptance of concrete as a major construction material means that more efficiency and accuracy in formwork plan and reuse will be needed.

1.1 Formwork using process

The report about formwork safety risks and reliability assessment from CPWR (the Center for Construction Research and Training) highlights that the general site life-cycle of vertical concrete formwork includes up to 18 steps [3]. Many researches have been conducted relating to the design, selection, erection, failure, and safety and reliability of formwork. Nemati [4] proposed some considerations for both horizontal and vertical formwork design, including safety, cost, unique design challenges, and the contractor. In order to improve formwork utilization on the construction site, Krawczyńska-Piechna [5] used the MCDA method for the formwork selection and developed a simple interactive simulator to analyze the efficiency of the formwork utilization. Zhang et al. [6] used a reliability analysis for formwork shoring systems, considering reasons for the failures in concrete structure. Despite the aforementioned studies on formwork using process, the research on formwork plan and reuse is still limited.

1.2 Building information modeling (BIM)

With rich information, BIM has been widely used in AEC industry to obtain construction quantity take-offs (QTO) measuring the materials in the project. Wei et al. [7] developed a BIM-based method to calculate the auxiliary materials in housing construction, generating the dosage report, construction drawings, and processing drawings, but the reuse of auxiliary was not discussed. A schedule approach integrating BIM with MS Project was presented by Liu et al. [8], which facilitated the automatic generation of optimized activity level construction schedules for building projects under resource constraints. Therefore, this paper will also focus on the BIM-based approach integrating concrete work schedule to generate formwork plan automatically.

1.3 Two-dimensional cutting stock problem (2D-CSP)

According to the distinct dimensions of elements in a building, the formwork purchased in the market may not be the same with the elements completely. Some preprocessing activities must be done on sites, meaning that a given set of forms (smaller items) with distinct length and width, should be placed into one or more formwork (larger objects) with the same dimensions, without overlap so as to minimize the total number of the formwork. This is a combinatorial optimization problem similar to 2D-CSP, and this NP-hard problem has been studied for decades in the manufacturing industry. Brandão et al. [9] presented pattern-based methods making the FFD solutions for extremely large cutting stock instances found in a very short time. Xu et al. [10] combined the PSO algorithm with bottom-left-fill heuristic placement strategy to allocate irregular items, and the results showed that the algorithm could bring out good solutions for most of the cutting stock problem with pieces of irregular shape. Therefore, this paper will integrate the FFD with CBL algorithm to assist the automated calculation of formwork demand. For convenience, items and objects will be respectively used to represent small forms and large formwork.

2. BIM-BASED AUTOMATED PLANNING FRAMEWORK

As shown in Fig.1, the proposed BIM-based automated planning framework consists of three main modules: (1) elements information extraction module, (2) items calculation module, and (3) objects plan module. The elements information extraction module (Part 1) is based on BIM model of the buildings, construction elements and their dimensions are extracted. In the items calculation module (Part 2), based on the output of the Part 1, a rule-calculating approach for elements have been proposed and implemented to obtain the dimensions and quantities of all items. Based on the calculation results of the Part 2, the MS Project is used to generate the reuse plan in the objects plan module (Part 3), in which an improved algorithm, integrating First Fit Decreasing (FFD) with Coordinate Bottom Left (CBL), has been used to automatically obtain the object quantities. The final results of the proposed framework include items reuse plan, objects demand quantities, and objects purchasing time. Each of these three main Module is discussed in detail in the following sections.

2.1 Elements information extraction module

BIM has been increasingly applied through the life-cycle of building projects considering rich geometric and semantic information it supplies. Industry Foundation Classes (IFC) is currently the main neutral file format for data exchange among AEC/FM software [11]. IFC is developed based on EXPRESS language. Building elements such as columns, slabs, walls and beams are usually defined as IFC entities. For example, column is represented as *IfcColumn* and wall can be represented as both *IfcWall* and *IfcWallStandardCase*. Since this study aims at formwork planning for the main structure of building projects during construction, four main building elements: column, slab, wall, and beam are considered. The purpose of this module is to extract the geometry information, especially for dimensions, of these building elements to plan formwork usage during construction. More details of IFC entity representations and EXPRESS data structure of these building elements are introduced in the next sub-sections.

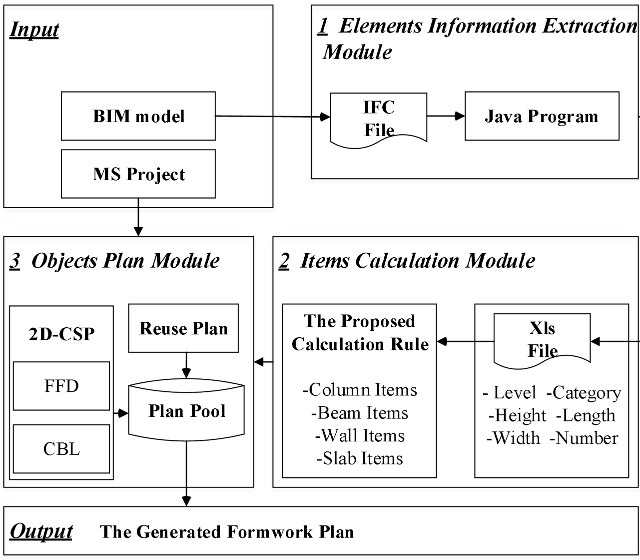


Fig. 1. The proposed BIM-based automated planning framework for formwork

2.1.1 Column IFC entities [12]

The column (*IfcColumn*) is a vertical structural member of a building. It is often aligned with a structural grid intersection. IFC specification provides two entities for column occurrences: *IfcColumn* and *IfcColumnStandardCase*. *IfcColumnStandardCase* is used for all columns that are profile defined and swept along a directrix. *IfcColumn* is used for other columns with changing profile sizes along the extrusion. Location and dimension of a column are stored in *IfcColumnStandardCase* or *IfcColumn*. Fig.2 shows how dimension is stored in EXPRESS data structure.

Fig. 2. Illustration of column EXPRESS data structure

2.1.2 Slab IFC entities [13]

Slab is a component of the construction that normally encloses a space vertically. The slab may provide the lower support (floor) or upper construction (roof slab) in any space in a building. Three entities can be used to represent slab occurrences: *IfcSlabStandardCase*, *IfcSlabElementedCase*, and *IfcSlab*. *IfcSlabStandardCase* represents slabs geometrically by a SweptSolid geometry (or by a Clipping geometry based in SweptSolid), if a 3D geometric representation is assigned. *IfcSlabElementedCase* is used slabs that are aggregated from subordinate elements. *IfcSlab* is used for any other slabs, especially for slabs with changing thickness. EXPRESS data structure of slab is similar to column.

2.1.3 Wall IFC entities [14]

The wall represents a vertical construction that bounds or subdivides spaces. Some may work as a bearing structure (e.g., shear wall). The IFC specification provides three entities for wall occurrences:

IfcWallStandardCase, *IfcWallElementedCase*, and *IfcWall*. *IfcWallStandardCase* represents walls that have a non-changing thickness along the wall path. *IfcWallElementedCase* is used for walls that are aggregated from subordinate elements, following specific decomposition rules. *IfcWall* is used for any other walls, especially for walls with changing thickness along the wall path. EXPRESS data structure of a wall is presented in Fig.3.

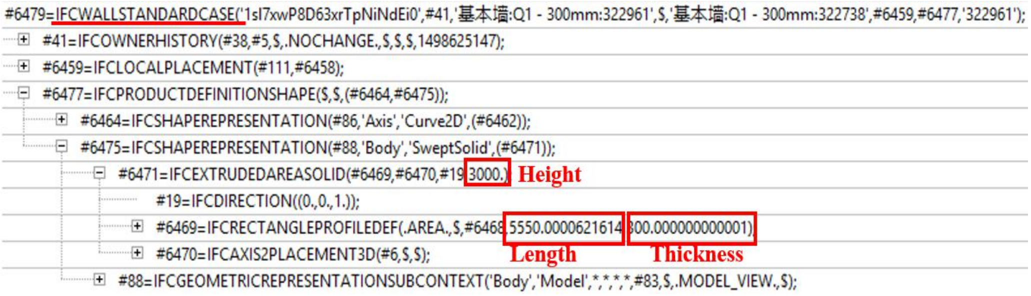


Fig. 3. Illustration of wall EXPRESS data structure

2.1.4 Beam IFC entities [15]

An *IfcBeam* is a horizontal, or nearly horizontal, structural member that is capable of withstanding load primarily by resisting bending. It represents such a member from an architectural point of view. IFC specification provides two entities for beam occurrences: *IfcBeamStandardCase* and *IfcBeam*, which is similar to IFC entities of column. EXPRESS data structure of a beam storage in IFC format can resort to *IfcWall*.

After studying the EXPRESS data structures of all considered building elements, an application using Java programming is developed to extracted dimensions and quantities of all columns, slabs, walls, and beams by levels. The extracted information is then stored in an Excel file and will be used to optimize formwork planning, which is introduced in the next section.

2.2 Items calculation module

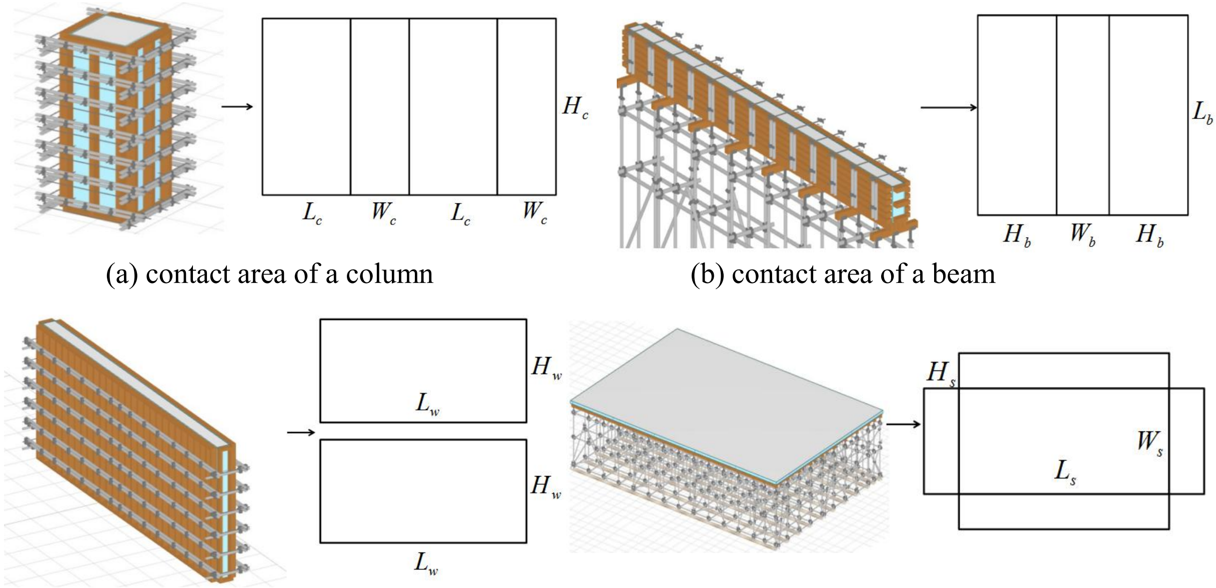
In the concrete work, items are erected around the elements of buildings to mold the concrete to the desired size and shape. However, a building always consists of large amounts of structure elements, causing the difficult calculation, manpower waste, and some errors. In addition, the dimensions of the objects obtained are diverse according to different manufacturers, form materials, and related norms. There are some assumptions for the rule calculating items. First, items are in regular shapes of the rectangle or square. Second, the length of items is always used to describe the side parallel to the horizontal plane, while the width is the side parallel to the vertical plane. Third, the thickness of objects is not taken into consideration. Forth, the length of an object is always longer than its width. Above all, this section sets three main constraints for the automatic calculation rule, including dimension, seam number, and seam layout constraints.

- *Dimension constraints* are imposed to guarantee the items cut from objects, and the dimension of items should not to exceed that of objects on both horizontal and vertical directions.
- *Seam number constraints* are imposed to meet the concrete work construction process requirement that seams should be at least minimized to the possible extent.
- *Seam layout constraints* are imposed to avoid the situation in which seams are on the same horizontal or vertical plane of the adjacent surface of a single element. If not, the quality of shaped elements may get worse due to the local stress concentration.

2.2.1 Element contact areas

There is one more point to notice that the dimensions of items are related to the contact areas between items and elements. Therefore, the contact areas of main four building elements are presented respectively as follows.

The contact area of a column consists of four rectangles (see Fig.4(a)) and the opposite rectangles are symmetric. Therefore, in the rule for calculating column items, only two surfaces need to be taken into consideration. For a wall, which consists of two symmetric rectangles, an alternative one is used to calculate the items (see Fig.4(c)). As illustrated in Fig.4(b) and (d), the contact area of a beam consists of three rectangles, while there are five in a slab. However, they have one point in common that the middle one is asymmetric, while the others opposite are symmetric. So, in the rule, there is no need to calculate all the members in the contact area of an element. With the illustration of the contact area, rule process for four main elements is presented in the next sub-section.



(a) contact area of a column (b) contact area of a beam

(c) contact area of a wall (d) contact area of a slab

Fig. 4. Illustration of the contact area of four main building elements

2.2.2 Rule-calculating approach

This paper has improved the method mentioned in [7] by considering the constraints and contact areas simultaneously. The rule process consists of the following steps:

Step 1 Dimension judgment: The planner may not accurately obtain the length and the width of an object in advance, so the *dimension judgment* takes on even more importance to generate a formwork plan. If the dimensions of contact areas are less than or equal to the dimension of the object, the length and width of items equal to that of the contact area respectively and the total number of items is easy to obtain (e.g., only four items for a column). If not, go to Step 2.

Step 2 Regional division: In order to deal with the situation mentioned in Step 1 and satisfy the constraints simultaneously, the dimension of an object is used as the reference standard to divide the contact area. For example, if the length of a contact area is greater than that of an object and the width is less than the object (e.g., a column or a beam), the area should be divided into two sub-areas. In the first sub-area, there are several items in the same dimension with one side equal to the width of the contact area, and other side equal to the length of the object. However, the second sub-area contains only one item with the same width as the former and the length equal to the rest length of the contact area. With this *regional division* procedure, the dimensions of items could be calculated for arbitrary dimensions of contact areas.

Step 3 Category integration: According to the Step 2, all the required items have been calculated. Due to the amount of elements with different dimensions, there may be some items in the same dimensions. In order to execute the next module easily, this procedure aims to integrate them into a category.

Based on the item calculation rule, the dimensions and quantities of each category of items have been obtained. The next problem to be solved is how to cut these items from the objects, and it turns out to be a 2-dimensional cutting stock problem as mentioned in section 1.3.

2.3 Objects plan module

As mentioned in section 2.2, the dimensions and quantities of items erected around elements have been automatically calculated and integrated for the objects plan. In addition to the information of items, the concrete work schedule is also important to generate the reuse plan. Usually, several floors have the same structure or elements in a building, namely standard floor, among which the formwork could be reused for several times. With the schedule of concrete work and the information about the items, the algorithm integrating First Fit Decreasing (FFD) with Coordinate Bottom Left (CBL), is used to generate the formwork plan. More details of the generated process will be discussed in the following sub-sections.

2.3.1 Reuse plan generation

In the AEC industry, the formwork along with its hardware and accessories, are used over and over again through over their life time to maximize the productivity and minimize the cost. In terms of the concrete work schedule, the formwork reuse plan has been generated given the items reuse in standard floors. Hence, in the framework, the MS Project has been used to generate the schedule, including task name, duration, start and finish time of each work, and the time to purchase and process the objects. In addition, the files from MS Project could be read by BIM tools automatically, making the formwork plan process efficient.

2.3.2 Algorithm for formwork planning

Once the items information and reuse plan are prepared completely, the formwork planning can be generated by combining the FFD algorithm with CBL algorithm. Due to the Module 1 and 2, a set of items has been given, each category with length l_i, width w_i and demand d_i. An unlimited number of objects are also given, with the identical length L and width W. And the objective is to minimize the number of objects used to place all the items. Before the algorithm executing, the object could be set in the first quadrant of Cartesian coordinates system. A category of items is represented by a vector $V_i = (x_i, y_i, l_i, w_i, d_i)$, in which (x_i, y_i) is the left-bottom coordinate of the item after being set in the system (see Fig.5(a)).The detailed procedure of the integrated algorithm will be explained as follow and shown in Fig.6.

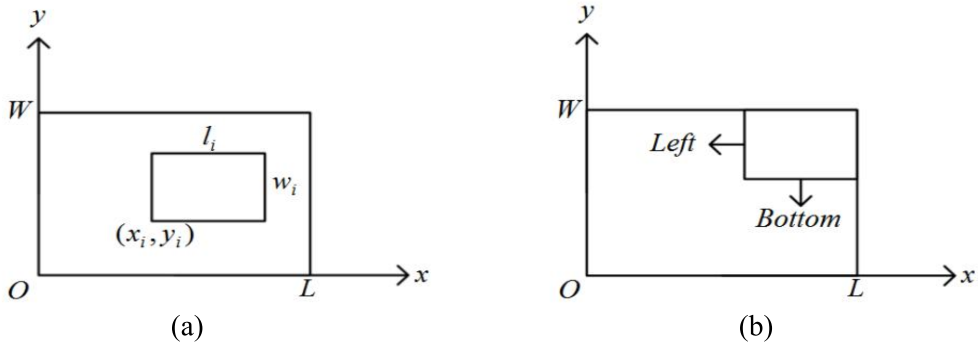


Fig.5. Illustration of items location

(1) *Initialization*. The algorithm starts by initializing parameters. First, the *binCounter* is initialized with zero value, which represents the number of objects opened. Second, owing to the initialized value of *binCounter*, the *currentArea* is also initialized with zero value, which records the remaining area of an object. The value of *currentArea* equals the area of the object illustrating that there is no item located, while zero value means the items located have covered all of the object. Third, the *itemNum* representing the remaining number of items is initialized with the total number of the items (Σd_i).
(2) *Items location*. In order to maximize the utility of an object, the items should be sorted decreasingly with their areas, and each item will be set in the coordinate system in this order. First to compare the area of the setting item with the *currentArea* in the order of objects opened, if the *currentArea* is less, go to step (4). Second set the item in the coordinate system and let its top-right corner land on that of the object (see Fig.5(b)). Third, try to move the item down parallel to the vertical axis until the bottom of the item has reached that of the object or the top of other items. And then, try to move the item left parallel to the horizontal axis until the left has reached that of the object or the right of other items. If there is no more space for moving and all areas of the item are in the object, locate the item and go to step (3). If not, go on moving or go to step (4).
(3) *Updating the currentArea*. As it is previously stated, if there are no items located in the object, *currentArea* is equal to that of the object. After step (2) executed, *currentArea* should minus the areas of the located items and the areas cannot be occupied by any other items. Then go back to step (2).
(4) *Updating the binCounter*. If *currentArea* is less than the area of the setting item, or all the area of the item cannot be contained in the object, the new one should be opened, and go back to step (2). If *itemNum* equals to zero value, the algorithm ends up. It is worth noting that the previous objects will not be closed until *currentArea* equals to zero value or algorithm ends up, meaning that the smaller item is

also permitted to be located in the first object if possible.

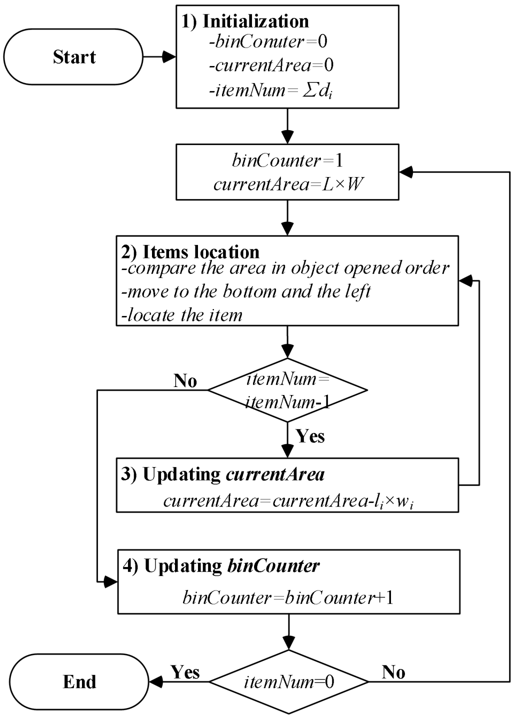


Fig.6. The algorithm for formwork planning

3. ILLUSTRATIVE EXAMPLE

An example is herein presented to illustrate the proposed framework. In the example, a 10-store building using the frame-shear wall structure is originally represented in the format of Revit (see Fig.7). The four main elements column, beam, wall and slab are all included in this model. According to an experienced planner on site, some parameters and conditions should be set here to simulate the real construction site. The objects are made of plywood, namely plyform, each object with length *L=2440mm* and width *W=1220mm*, which is the common category of formwork on construction sites. Actually, in the proposed framework, the dimension of objects is open for users to make it more efficient and automated. The plyform with its typical feature can only be used for no more than five times. The example is illustrated by the proposed framework in the following sub-sections.

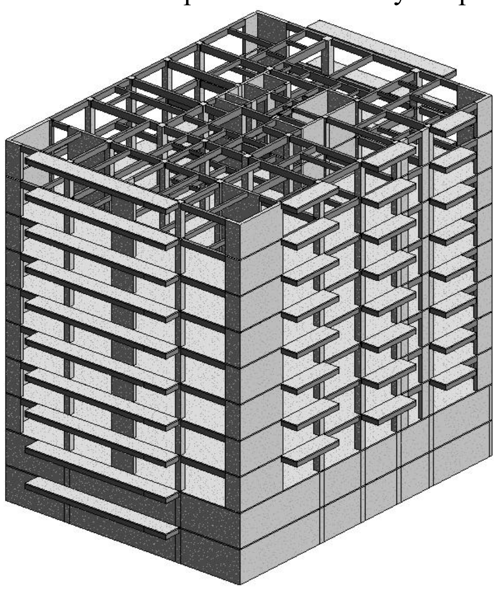
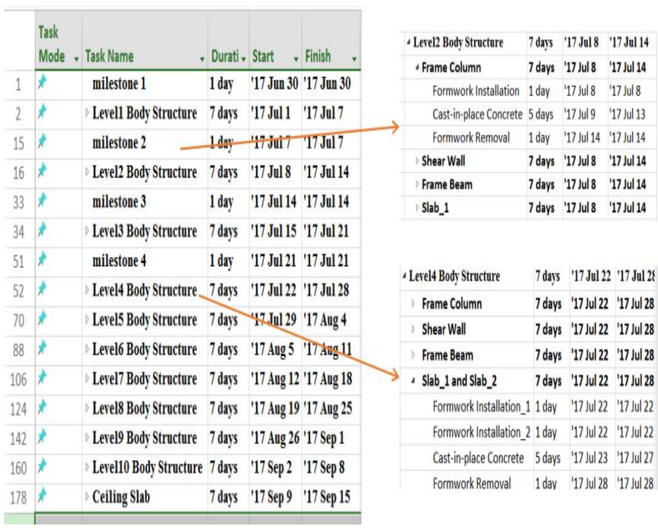


Fig.7. Overview of the BIM model



	Task Mode	Task Name	Durati	Start	Finish
1		milestone 1	1 day	'17 Jun 30	'17 Jun 30
2		Level1 Body Structure	7 days	'17 Jul 1	'17 Jul 7
15		milestone 2	1 day	'17 Jul 7	'17 Jul 7
16		Level2 Body Structure	7 days	'17 Jul 8	'17 Jul 14
33		milestone 3	1 day	'17 Jul 14	'17 Jul 14
34		Level3 Body Structure	7 days	'17 Jul 15	'17 Jul 21
51		milestone 4	1 day	'17 Jul 21	'17 Jul 21
52		Level4 Body Structure	7 days	'17 Jul 22	'17 Jul 28
70		Level5 Body Structure	7 days	'17 Jul 29	'17 Aug 4
88		Level6 Body Structure	7 days	'17 Aug 5	'17 Aug 11
106		Level7 Body Structure	7 days	'17 Aug 12	'17 Aug 18
124		Level8 Body Structure	7 days	'17 Aug 19	'17 Aug 25
142		Level9 Body Structure	7 days	'17 Aug 26	'17 Sep 1
160		Level10 Body Structure	7 days	'17 Sep 2	'17 Sep 8
178		Ceiling Slab	7 days	'17 Sep 9	'17 Sep 15

Level2 Body Structure	7 days	'17 Jul 8	'17 Jul 14
Frame Column	7 days	'17 Jul 8	'17 Jul 14
Formwork Installation	1 day	'17 Jul 8	'17 Jul 8
Cast-in-place Concrete	5 days	'17 Jul 9	'17 Jul 13
Formwork Removal	1 day	'17 Jul 14	'17 Jul 14
Shear Wall	7 days	'17 Jul 8	'17 Jul 14
Frame Beam	7 days	'17 Jul 8	'17 Jul 14
Slab_1	7 days	'17 Jul 8	'17 Jul 14

Level4 Body Structure	7 days	'17 Jul 22	'17 Jul 28
Frame Column	7 days	'17 Jul 22	'17 Jul 28
Shear Wall	7 days	'17 Jul 22	'17 Jul 28
Frame Beam	7 days	'17 Jul 22	'17 Jul 28
Slab_1 and Slab_2	7 days	'17 Jul 22	'17 Jul 28
Formwork Installation_1	1 day	'17 Jul 22	'17 Jul 22
Formwork Installation_2	1 day	'17 Jul 22	'17 Jul 22
Cast-in-place Concrete	5 days	'17 Jul 23	'17 Jul 27
Formwork Removal	1 day	'17 Jul 28	'17 Jul 28

Fig.8. Schedule of the concrete work

3.1 Elements information extraction

According to section2.1, the algorithm was developed by using Java to extract the information of elements in the building model. The partial outputs are shown in Fig.9. The extracted information has been stored in an Excel file.

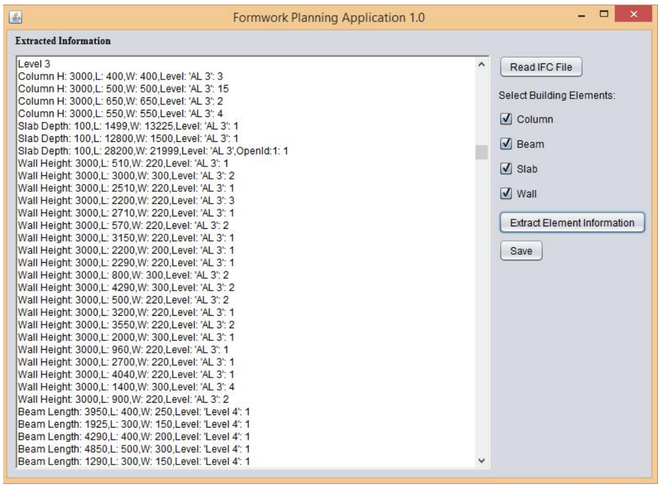


Fig.9. User interface of information extraction

3.2 Items calculation implementation

The rule proposed for items calculation has also been applied in the Java program. According to the rule, the dimensions of the contact areas of elements in the Excel are first judged. Then, the dimension of the object was used as a reference standard to divide the contact areas into a proper one. Dimensions and quantities of the divided items have been represented as vectors, which could be used in next module directly. Finally, the items with the same dimensions are integrated to simplify the computing complexity.

3.3 Objects plan generation

As shown in Fig.8, the schedule of concrete work is generated. In the schedule, every floor consists of three steps to shape the structure, including erecting items, casting concrete in place, and removing items. The task named *milestone* is used to emphasize the time to purchase and process the formwork. As a result, the reuse plan is created, and then the output of the proposed framework is generated in Table 1.

Table 1. The generated formwork plan

<table>
<tr><th>No.</th><th>Elements</th><th>Level</th><th>Purchase and Process</th><th>Objects Number</th><th>Reuse Times</th></tr>
<tr><td>1</td><td>Column, Wall, Beam</td><td>Level 1-Level 2</td><td>milestone 1</td><td>416</td><td>2</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Column, Wall, Beam</td><td>Level 3-Level 7</td><td rowspan="2">milestone 3</td><td>305</td><td>5</td></tr>
<tr><td>Level 8-Level 10</td><td>305</td><td>3</td></tr>
<tr><td rowspan="3">3</td><td rowspan="3">Slab_1</td><td>Level 2-Level 6</td><td rowspan="3">milestone 2</td><td>228</td><td>5</td></tr>
<tr><td>Level 7-Level 10</td><td rowspan="2">228</td><td rowspan="2">5</td></tr>
<tr><td>Ceiling</td></tr>
<tr><td rowspan="3">4</td><td rowspan="3">Slab_2</td><td>Level 4-Level 8</td><td rowspan="3">milestone 4</td><td>8</td><td>5</td></tr>
<tr><td>Level 9-Level 10</td><td rowspan="2">8</td><td rowspan="2">3</td></tr>
<tr><td>Ceiling</td></tr>
<tr><td colspan="2">Total Number of Objects:</td><td colspan="4">1498</td></tr>
</table>

In line with the drawings and experience, the construction planners should first identify the formwork demand for each floor and readjust the number of the purchase and process formwork next time in accordance with the reality, but this practice may make it difficult to calculate the number of needed construction formwork and add more workloads to planners. As it is shown in the case, the automated framework could precisely calculate the number of all required construction formwork for pouring and reuse them if necessary. In addition, the automated framework could also provide information about purchase, process and reuse which could reduce the uncertainties for formwork calculation as well as the repetitive workloads of planners.

4. DISSCUSIONS AND CONCLUSIONS

This paper presents a BIM-based automated framework for formwork planning on construction sites. The major contribution of the proposed framework is to automatically plan and reuse the formwork on construction sites. In the framework, the geometric information of elements in a building is extracted in IFC, which has been exported by the BIM model. The extracted information of elements is then used to determine the contact areas between elements and forms. According to the contact areas, the forms, named items in a CSP, can be calculated by the proposed rule for elements. The previous output is the total number of forms without formwork reusing plan. So, with the concrete schedule from MS Project, the utility of the calculated forms can be maximized. The algorithm integrating the FFD with CBL has been used to generate the formwork plan considering the reusing plan. Finally, an illustrative example has been presented to demonstrate the utility of the proposed framework. In the example, the quantities and reuse plan of the formwork required in distinct construction phases and different structures are obtained and generated automatically by implementing the framework proposed in this paper. With the proposed automated framework, potentially, more accurate and detailed formwork plan can be generated in the future and reduce the error-prone manual efforts eventually more quickly.

However, some limitations still exist in the framework. First, the proposed framework relies on the assumption that shapes of the elements and forms are square or rectangular. Despite the framework can be used in most of residential buildings, for other irregular shapes, it may not work. In addition, as shown in Table 1, in the illustrative example, some items were not reused adequately, for there are no elements in the same dimensions to reuse the items in this building. Future research may focus on developing more comprehensive rules that are not limited by the assumption in this research. Accordingly, more instances, such as several similar buildings in a project or projects, are needed to implement the proposed framework to realize its full potential.

ACKNOWLEGEMENTS

The support of the Graduate Research Innovation Project of Chongqing Jiaotong University (No.2017B0101) is gratefully acknowledged.

REFERENCES

1. ACI 347, 2004. *Guide to Formwork for Concrete*. American Concrete Institute.
2. Huang RY, Chen JJ, Sun KS, 2004. Planning gang formwork operations for building construction using simulations. *Automation in Construction* 13, pp.765-779. doi:10.1016/j.autcon.2004.05.001.
3. CPWR Small Study Report, 2014. *Use and Re-use of Formwork: Safety Risks and Reliability Assessment*. Oregon State University.
4. Nemati KM, 2005. *Introduction to Concrete Formwork and Vertical Formwork Design*. Tokyo Institute of Technology.
5. Krawczyńska-Piechna A, 2017. Comprehensive Approach to Efficient Planning of Formwork Utilization on the Construction Site. *Procedia Engineering* 182, pp.366-372. doi:10.1016/j.proeng.2017.03.114.
6. Zhang H, Rasmussen KJR, Ellingwood BR, 2012. Reliability assessment of steel scaffold shoring structures for concrete formwork. *Engineering Structures* 36, pp.81-89. doi:10.1016/j.engstruct.2011.11.027.
7. Wei H, Zheng S, Zhao L, Huang R, 2017. BIM-based method calculation of auxiliary materials required in housing construction. *Automation in Construction* 78, pp.62-82. doi:10.1016/j.autcon.2017.01.022.
8. Liu H, Al-Hussein M, Lu M, 2015. BIM-based integrated approach for detailed construction scheduling under resource constraints. *Automation in Construction* 53, pp.29-43. doi:10.1016/j.autcon.2015.03.008.
9. Brandão F, Pedroso JP, 2014. Fast pattern-based algorithms for cutting stock. *Computers & Operations Research* 48, pp.69-80. doi:10.1016/j.cor.2014.03.003.

10. Xu Y, Yang G, Pan C, 2013. A Heuristic Based on PSO for Irregular Cutting Stock Problem. *IFAC Proceedings* Volumes 46, pp.473-477. doi:10.3182/20130708-3-cn-2036.00094.
11. Industry Foundation Classes Release 4. May]; Available from: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm.
12. buildingSMART. *IfcColumn*. [cited 2017 July 25]; Available from: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcsharedbldgelements/lexical/ifccolumn.htm.
13. buildingSMART. *IfcSlab*. [cited 2017 July 25]; Available from: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcsharedbldgelements/lexical/ifcslab.htm.
14. buildingSMART. *IfcWall*. [cited 2017 July 25]; Available from: http://www.buildingsmart-tech.org/ifc/IFC4x1/old_html/schema/ifcsharedbldgelements/lexical/ifcwall.htm.
15. buildingSMART. *IfcBeam*. [cited 2017 July 25]; Available from: http://www.buildingsmart-tech.org/ifc/IFC4x1/old_html/schema/ifcsharedbldgelements/lexical/ifcbeam.htm.