Automating fabrication sequencing for industrial construction

Di Hu *, Yasser Mohamed, and Simaan M. AbouRizk

Department of Civil and Environmental Engineering, University of Alberta, Canada,
** Corresponding author (dhu@ualberta.ca)*

Purpose Industrial construction projects are heavily dependent on pre-fabrication of piping components. Unlike traditional manufacturing, many pipe spools have a unique design and need to be custom fabricated due to the one-of-a-kind characteristic of each industrial project. This is reflected in the fact that fabrication sequences vary greatly among pipe spools. Planning these sequences has considerable impact on the fabrication performance. However, it is currently mostly done in the form of human manual input. Personal experience and judgment are the major grounds on which sequencing decisions are based. Given the enormous number of pipe spools and the fast-tracking nature of industrial projects, the efficiency and quality of such decisions cannot be guaranteed. Automating this decision-making process has the potential for overall performance enhancement, but has not yet been sufficiently investigated. **Method** We explore two different problem solving techniques, mainly artificial intelligence (AI) planning and dynamic programming (DP). A number of experiments have been conducted to evaluate their effectiveness. **Results & Discussion** The results show that AI-planning–a sophisticate planning technique–has difficulty parsing fabrication logic that is prerequisite for AI-planners to result in a solution. DP, on the other hand, shows greater flexibility in incorporating this logic and a higher efficiency of discovering the optimal sequence. Future research will be aimed at incorporating the DP-algorithm with a discrete event simulation model so that fabrication sequences can be dynamically generated and adjusted to address changing project conditions.

***Keywords**: automated planning, industrial construction, artificial intelligence planning, dynamic programming*

INTRODUCTION

Industrial construction projects refer to a wide range of facilities for oil/gas mining and production, power plant, food, and pharmaceutical purposes. Many industrial projects are mega projects (especially in the oil and gas industry), meaning that they are large scale, require intensive capital, and involve complex technologies. Piping is an indispensable component of industrial construction projects which is used to connect various processing units and equipment and to convey processed gas and fluid. Piping is always the largest single job (Kim and Ibbs 1995)[1] and a critical and costly process in an industrial construction project (BRT 1982)[2]. To avoid trade stacking and congestion on the construction site, a piping system is broken down into pipe spools which are usually fabricated in off-site shops. Fabrication shops are controlled environments and they normally enjoy higher productivity than on-site activities.

Pipe spool fabrication is critical in the entire piping process, since it produces the pipe spools which are building blocks for downstream processes (e.g. module assembly and site installation). However, in reality, it is usually not pipe spool fabrication that drives the whole piping process. Due to the fact that many industrial construction projects are executed in a fast track manner, all piping stages could be more or less overlapped, which means pipe spool fabrication and installation could start even before the design is complete. It is often the availability of pipe spool drawings and site installation that drive the pipe spool fabrication. Pipe spool fabrication strives to fit its work to the sequence of the site installation given the availability of ISO drawings and raw materials. This leads to a lot of problems later in the shop operation. For example, pipe spool fabricators often contend with out-of-sequence or late supply of ISO drawings. They also suffer from the rush order or change order from the site installation. It is not uncommon for pipe spool fabrication shops to experience productivity loss, missed due dates, and incurred extra costs.

The sequence of pipe spool fabrication has a significant impact over the shop performance. Di and Mohamed (2011)[3] conduct a simulation experiment to test if different fabrication sequences can lead to different shop performance. The results show that by varying the fabrication sequence for 22 pipe spools, the total cycle time can be reduced by 10.09%, and the number of handlings (i.e. considered as non-value-adding activity) can be decreased by 16.88%. The capability to identify the optimal fabrication sequence for pipe spools is therefore critical for fabricators to maintain their performance in a project environment full of uncertainties.

In reality, the fabrication sequences of pipe spools are often decided by shop foremen in a very heuristic

manner (i.e. primarily based on personal experience). Given the enormous number of pipe spools involved in an industrial project and the fast track nature of project execution, it is rather challenging for human planners to determine fabrication sequence with both efficiency and quality. There is a potential for performance improvement if this manual decision making process can be automated.

This paper explores two candidate solutions to automatically identify optimal fabrication sequence for pipe spools. The first solution uses Artificial Intelligence (AI) planning technique, while the second adopts the dynamic programming (DP) technique. A number of experiments have been done in both solutions and the results show that AI planning is constrained by its numerical assignment and calculations capability and it could require special preprocessing of the input that could be computationally prohibitive for complex pipe spools. DP technique, on the other hand, offers a great flexibility of accommodating both numerical calculation and logical complexity (e.g. various fabrication rules). It displays quite satisfactory computation efficiency. In addition, DP also guarantees a global optimal solution. A conclusion is thus drawn that the DP technique is more suitable than AI planning to solve the pipe spool fabrication sequencing problem.

Pipe Spool Fabrication

Pipe spools are fabricated in fabrication shops from a group of raw materials, i.e. raw pipes and pipe fittings (elbows, flanges, nozzles, and etc.). The fabrication process consists of three major steps: (1) pipe cutting, (2) fitting (temporary connection), and (3) welding (permanent connection). Usually some of the pipe spool components are fitted and welded first before it is fitted with other components. The back-and-forth between fitting table and welding stations will continue until the all pipe spool components are fabricated.

Two types of fitting and welding are exercised in the fabrication shop—roll and position (in Figure 1). The selection is made by whether the arm length is longer than the clearance limit of rolling machines or not. If it is within the clearance limit, roll fitting and welding can be performed and the main pipe run can be rolled by a rolling machine, whereas if it exceeds the limit, position fitting and welding is the option and the fitter or the welder has to move around the main pipe run to accomplish fitting or welding. Since position fitting and welding involves much more manual effort, it usually takes a much longer time than roll fitting and welding. One major objective of pipe spool fabrication sequencing is to minimize the number of position fitting and welding processes.

The fabrication sequence defines the process of how a pipe spool will be fabricated gradually from raw materials (e.g. pipes and fittings), to intermediate sub-assemblies, and eventually to the final product. Usually, a pipe spool can be fabricated through a number of alternative sequences. However, in reality, sequence is determined by shop foremen in a very heuristic manner and these alternative sequences seldom have a chance to be compared and evaluated. As a result, opportunities of productivity improvement slip away.

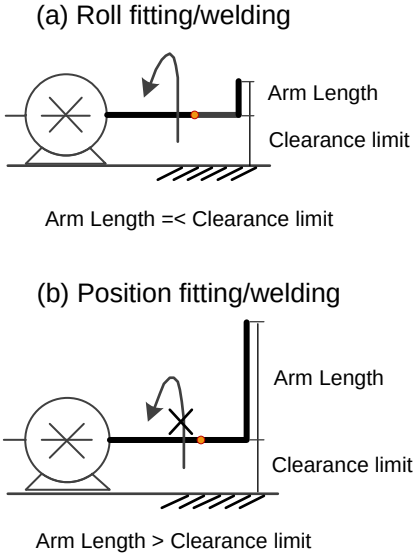

Fig. 1 Roll welding vs. position welding (Di and Mohamed 2012)[3]

Previous Research on Construction Sequencing

It should be noted that in this paper sequencing is more related to logic dependency between processes by considering the geometric and technological requirements, than the sequencing or prioritizing processes (i.e. those that compete for the same type of resource) given limited resource availability. A review of relevant research reveals two major topics: (1) identify underlying construction sequencing rationales, and (2) develop planning systems to automate the sequencing process. Inter-relationships between building components are often used as rationales to derive construction sequence. Researchers (Gray 1986[4], Kartam and Levitt 1990[5], Navinchandra et al. 1988[6], Jin et al. 1992[7]) identified various inter-component relationships, such as “covered by,” “weather protected by,” “supported by,” “enclosed by,” “connected to,” as well as “damaged by.” Echeverry et al (1991)[8] enriched the body of knowledge and categorized reasons for precedence relationship into four groups, namely “physical relationships among building components,” “trade interaction,” “path interference,” and “code regulations.” Under each category, they comprehensively enumerated more specific sequencing rationales. Other researchers have attempted to develop AI planning systems that capitalize on the existing construction sequencing rationales and automatically generate sequential dependencies between construction activ-

ities; for example, CONSTRUCTION-PLANEX (Hendrickson1987[9]), GHOST (Navinchandra et al. 1988[6]) , and OARPLAN (Darwiche et al. 1989[10]). Aalami et al. (1998)[11] propose a system named Construction Method Modeler, and intend to capture both process-based features (e.g. GHOST) and component-based features (e.g. OARPLAN). Recent work by Koo et al. (2007)[12] pointed out that much research on domain specific AI planning systems is focused more on identifying a correct construction sequence, than on discovering a number of possible sequence alternatives. They introduced a prototype system named "constraint-loaded CPM (CLCPM)", that makes use of constraint ontology, and a classification mechanism to automatically assign "role" and "status" to relevant activities (i.e. to a target activity that needs re-sequencing) in CPM.

Most research on previous construction sequencing is focused on building construction. Most sequencing rationales are derived from the physical relationships between building components. At the same time, many automatic planning systems are domain-specific which means they can only be applied to building construction projects. This makes it difficult to apply existing body of knowledge or planning systems to industrial construction projects, i.e. where the building blocks are pipe spools, equipment, and modules which most likely need to be pre-fabricated or pre-assembled before the final installation on site. The sequence constraints between these components are significantly different from those applied in building construction. For example, the relationship between pipe spool components is being welded (or connected). This relationship does not stipulate any natural sequence of actions that are applied on these components. For the pipe spool (shown in Experiment 1 in Figure 3), it is possible either to weld component1 and 2 first and then weld with component 3 or to weld component 2 and 3 first and then with component 1. In other words, being welded together does not mean there is any specific sequence between component 1, 2 and 3. But this is not the case between building components. For instance, columns have to be installed before beams can be constructed. Therefore, actions that are acting on columns have to precede those that are applied on beams.

Domain-Independent AI planning

Planning has been one of the major AI research areas since 1960s (Newell and Simon 1972[13]). Planning refers to selecting a sequence of actions that leads to a system (i.e. on which actions are applied) transforming from its initial state to a specific goal state. State is usually expressed in the form of a set of propositions. The state of a system is usually depicted by the states of all its constituent objects at a certain point of time. A number of actions are available to choose from and to apply on the system (or some of its constituent objects). Each action has its own precondition(s). An action is only applicable when the current state (of the system or of constituent objects) matches its entire precondition(s). Each action also has an explicit effect, which is the resulting state of the system (or the objects) after performing the action. Based on all this description, a planner carries out a state-space search or a plan-space search and identifies a sequence of actions to achieve the goal. If evaluation criteria are provided, some AI planning systems are able to identify optimal sequences of actions. This type of planning is sometimes also referred to as general-purpose planning. Planning Domain Description Language (PDDL) is one of the domain-independent AI planning languages (such as STRIPS or ADL). It was first developed by Drew McDermott (1998)[14]. Since then, it has evolved and refined through several versions. Domain-independent AI planning languages have been successfully implemented in many domains such as robot navigation, manufacturability of machined parts, and emergency evacuation (Ghallab et al. 2004[15]).

Darwiche et al. (1989)[10] argued that domain-independent AI planning is not a suitable approach to plan and sequence building construction projects. The major reason is that domain-independent AI planning fails to take advantage of domain knowledge (i.e. building construction) to reduce its search space and consequently has to carry out an extensive search. This leads to a few negative consequences such as computation inefficiency and sometimes inability to find an optimal plan. However, it should be noted that the context of this argument is fully limited to building construction. As mentioned before, pipe spool fabrication is a domain that fundamentally differs from building construction (i.e. building blocks, logic, inter-relationships). In fact, pipe spool fabrication has a number of distinguished features that make it a good candidate problem to be solved by domain-independent AI planning technique.

First, pipe spool fabrication does not have to deal with a wide range of objects and actions. A pipe spool usually comprises a number of pipes and fittings. However, they all can be viewed as one object class—pipe spool component—when it comes to welding. What matters is NOT the type of the component but rather the minimum and maximum coordinates (i.e. X, Y and Z) of the component. Only two actions are available to change the state of a pipe spool or pipe spool component—roll-welding or position-welding. This differs greatly from building construction where a wide variety of objects (e.g. footings, columns, beams, walls, doors, windows, etc.) are involved and hundreds of actions are required to build these components. It is also difficult to enumer-

ate all possible actions available to apply to the building components (Darwiche et al. 1989)[10].

Second, in the pipe spool fabrication world, every pipe spool component has only two different states—welded or not welded. The preconditions for welding action are quite simple—two pieces of sub-assemblies (or components) and one unfinished welding point (i.e. the welding point connects these two sub-assemblies). Its effect is a new sub-assembly (could also be the final pipe spool). However, it is difficult to abstract the state of building components as well as to define the preconditions and effects for building actions (Darwiche, 1989)[10].

Application of PDDL to pipe spool fabrication sequencing problem

Using PDDL to solve a planning problem entails a modeling process which results in two pieces of description. The first one is called domain file and provides a general description of the system (i.e. pipe spool fabrication system). Basically it describes what types of objects are involved in the system, what states are possible for each type of object, what actions are available, and what the pre-conditions and post-conditions are to execute these actions. Another description is called problem file which provides a specific description of a particular problem for this system. For example, it includes a number of instances of objects, each with a specific initial state and a goal state. Figure2 shows example domain and problem files that are customized for a pipe spool fabrication problem.

Fig.2 Example domain and problem definition files for pipe spool fabrication (Di and Mohamed 2012)[19]

A number of experiments are conducted to test the capability of AI planners to solve the pipe spool fabrication sequencing problem. These experiments are designed to be of incremental complexity. Figure3 shows the configurations of pipe spools used in these experiments. It should be noted that although the pipe spool in experiment3 seems to have a simpler configuration than the one in experiment 2, experiment 3 involves the major challenge of consideration of dimensions of components. The ability to calculate and track changes in the dimensions of spool components (or sub-assemblies) throughout the fabrication process is essential to distinguish roll welding and position welding. The domain file for pipe spool fabrication system is formulated and problem files for all pipe spools are also specified. Meanwhile, three popular AI planners (domain-independent) are selected: (1) Metric-FF (Hoffmann 2002[16]); (2) LPRPG (Coles et al. 2008[17]); (3) LPG (Gerevini and Serina 2002[18]). These planners are tested in all experiments and show a varied level of competence.

The results of all experiments are discussed in detail by Di and Mohamed (2012)[19]. It is indicated that none of the AI planners are fully adequate to solve real-life pipe spool fabrication sequencing problems. Specifically, Metric-FF has no problem handling all kinds of pipe spool fabrication logic but does not well support numerical assignments and calculations (i.e. in experiment 3). An illogical fabrication sequence is returned in experiment 3. LPG, on the other hand, contends with conditional effects. In order to release the conditions from the effects of actions, a grounding process must be completed. This requires making actions more specific and enumerating all possible situations (i.e. the domain file needs to define more specific actions). The result from experiment 3 shows that after the grounding process, LPG is able to return a logical fabrication sequence for the pipe spool. However, there is a side effect of the grounding process: the number of actions defined in the domain file grows exponentially with the number of welds in the pipe spool. For example, a pipe spool has N welds and then 2^{N-1} actions need to be explicitly formulated in the domain file (e.g. a pipe spool with 13 welds requires 4096 actions defined). This poses a huge challenge (i.e. could be computation prohibitive) to solve some extremely complex pipe spool problems. This finding led to a search for other problem-solving techniques that have the potential to tackle the pipe spool fabrication sequencing problem. Dynamic programming has been found to be a good candidate.

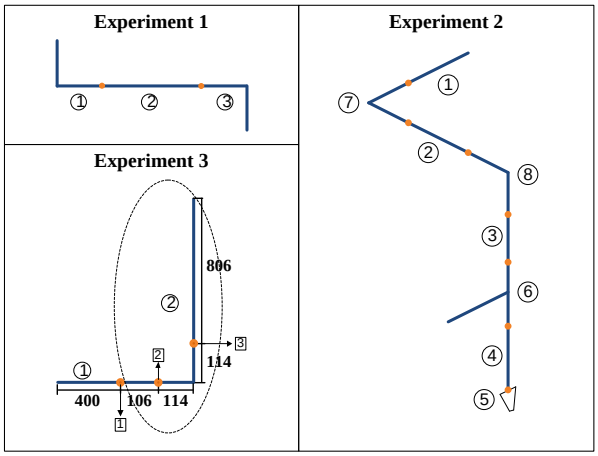

Fig.3. Pipe spools used in PDDL experiments (Di and Mohamed 2012)[19]

Dynamic Programming

Dynamic programming (DP) is a generic, efficient approach to solve optimization problems that involves a succession of decision making processes. More specifically, these problems should possess two features. First, the problem can be broken down into sub-problems whose solutions will be reused a number of times. This is called *overlapping sub-problems*. Another feature is that the optimal solution of a problem can be constructed from the optimal solutions of its sub-problems. This is referred to as *optimal substructure*. Compared to brute force approaches, DP improves the computation efficiency and guarantees a global optimal solution. It should be noted that DP differs from many other optimization algorithms which contain a universal algorithm that applies to every problem. To use DP, every problem needs to be custom formulated and involves innovative thinking.

Analysis of Pipe Spool Fabrication Sequencing Problem from DP Perspective

DP solves a problem through recursively decomposing the problem into sub-problems until their solutions are trivial and then constructing the optimal solution in a bottom-up manner by continuously selecting the optimal solutions for sub-problems which later become the basis for the optimal solution of the immediate parent sub-problems. This solution construction process stops when the sub-problem becomes the original problem itself. Following this same idea, the pipe spool fabrication sequencing problem is analyzed. Before going into detailed analysis, it is assumed that a roll-welding costs 1 while a position-welding costs 2. The optimization problem is now converted to find the minimum cost sequence to fabricate a pipe spool.

First, the optimal sequence to fabricate a pipe spool depends on the optimal sequence to fabricate its sub-assemblies. Figure4 shows that the pipe spool could be decomposed at each welding point (i.e. stage1 level). Each decomposition option results in a pair of sub-assemblies (could be pipe spool components as well). The cost to fabricate the pipe spool can be expressed in following equation:

$Cost_{pipe\ spool}$ =

$Cost_{sub\text{-}assembly1} + Cost_{sub\text{-}assembly2} + Cost_{To\ weld\ sub\text{-}assembly1\ and\ sub\text{-}assembly2}$

To determine which option is superior to fabricate the pipe spool, it is necessary to know the minimum cost to fabricate all these sub-assemblies, which in turn depends on the minimum cost to fabricate their children sub-assemblies. The decomposition stage2 in Figure 5 shows how sub-assembly 6 and 7 are further decomposed. It is noted that all of their children sub-assemblies are already atomic components (i.e. pipes or fittings). Therefore, it cost nothing to weld them. The minimum cost for sub-assembly6 is determined as following.

$Cost_{component1} = Cost_{component2} = 0$

$Cost_{sub\text{-}assembly6} = Cost_{component1} + Cost_{component2} + Cost_{To\ weld\ component1\ and\ component2}$

$= Cost_{To\ weld\ component1\ and\ component2}$

It is found that the minimum cost to fabricate the sub-assembly6 is equal to the cost to weld component 1 and 2, which makes it simple to determine whether it is a roll-welding or a position-welding. Given that the rolling axis is around the component2, the arm length is H2 (0.5m) and less than the rolling clearance (1.5m). It is then a roll welding. The minimum cost to fabricate sub-assembly 6 is 1.

$Cost_{sub\text{-}assembly6} = Cost_{To\ weld\ component1\ and\ component2} = 1$

The minimum cost for all the other sub-assemblies can be determined in a similar way. The process is always to decompose the pipe spool (or sub-assembly) to atomic component level where the fabrication cost is easy to determine.

From this point, the solution can be constructed in a bottom-up manner. For example, connecting component 1 and 2 is the minimum cost and the only way to fabricate the sub-assembly6. The same is for the sub-assembly 7. Then if connecting sub-assembly 6 and 7 is also the minimum cost sequence to fabricate the whole pipe spool (i.e. in fact it is), then the optimum sequence is formed.

Stage1: Component1 + Component2 → Sub-assembly6;

Component3 + Component4 → Sub-assembly7;

Stage2: Sub-assembly6 + Sub-assembly7 → The Pipe spool;

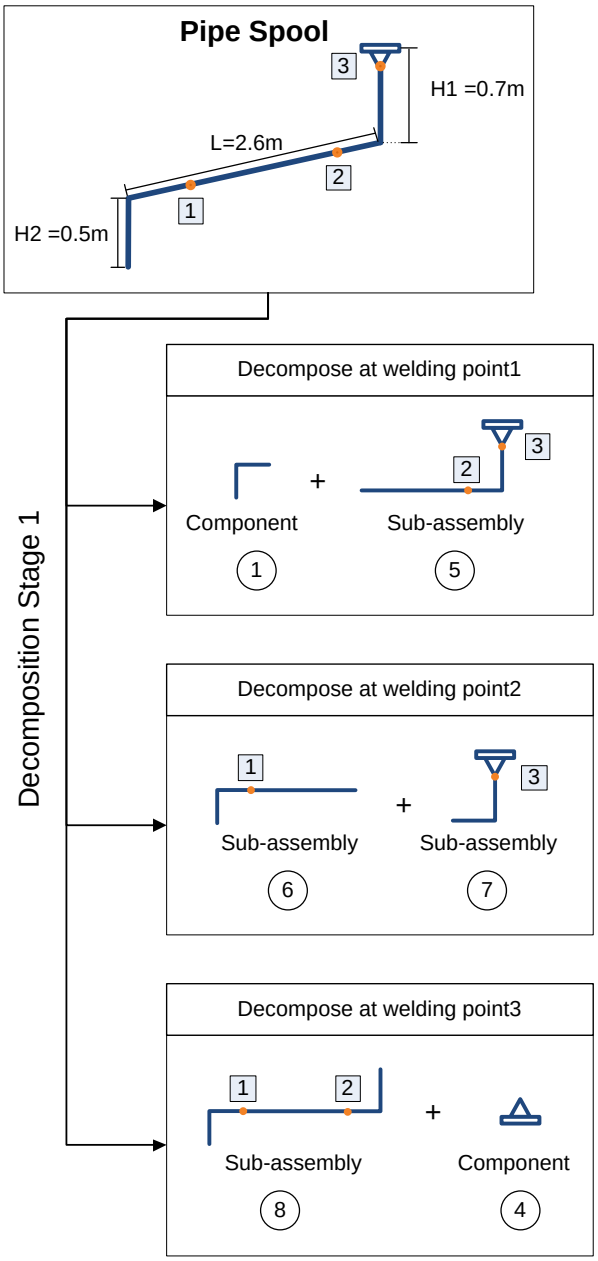

Fig. 4 Decomposition of an example pipe spool (Stage1)

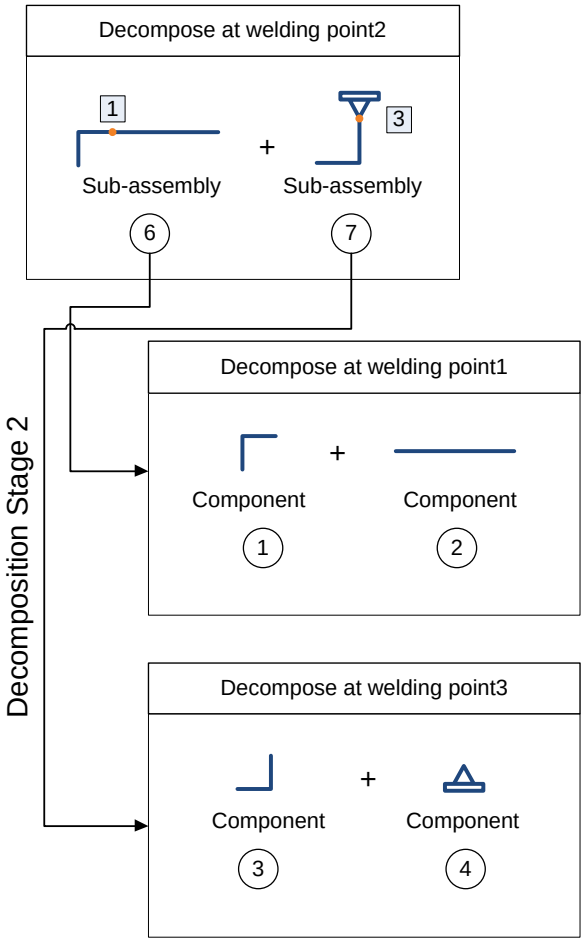

Fig. 5 Decomposition of an example pipe spool (Stage2)

Stage No.	Welding Point	Resulting Sub-assembly
1	4	[2, 6]
2	8	[2, 6, 4]
3	7	[3, 9]
3	5	[2, 6, 4, 8]
3	2	[1, 7]
4	6	[2, 6, 4, 8, 3, 9]
4	1	[1, 7, 11]
4	10	[5, 10]
5	3	[1, 7, 11, 2, 6, 4, 8, 3, 9]
5	11	[5, 10, 12]
6	9	[5, 10, 12, 1, 7, 11, 2, 6, 4, 8, 3, 9]

Table 1. Fabrication sequence returned from DP algorithm

Application of DP Algorithm to Pipe Spool Fabrication Sequencing Problem

Based on the foregoing idea, a DP algorithm is designed to solve the pipe spool sequencing problem. So far, it has been tested with a number of pipe spools. One of pipe spools is considerably complex and its configuration is shown in the Figure 6. The algorithm solves the problem within one and half minutes and returns a fabrication sequence shown in Table 1. It should be noted that because there is more than one optimal sequence in this case, a different sequence could be returned if the algorithm runs several times, but they all incur the same minimum fabrication cost.

CONCLUSION

Fabrication sequence is of significant impact on pipe spool fabrication performance. Current industry practice relies heavily on shop foremen's personal experience. Previous research on construction sequencing is mostly focused on building construction and therefore not readily applicable to pipe spool problems.

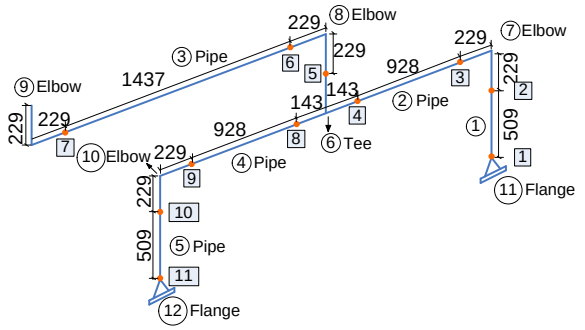

Fig. 6 an example pipe spool solved by Dynamic Programming

Gaps exist in the current body of knowledge related to automated process sequencing on industrial construction. This paper explores two potential problem solving techniques: domain-independent AI planning and Dynamic Programming. Observations from a number of experiments indicate that a DP-based algorithm outperforms AI planners in the capability of handling numerical assignment calculations and it also displays quite satisfactory solutions and computation efficiency. This leads to a conclusion that DP is more suitable for solving the pipe spool sequencing problem than AI planning. Future work will be directed to experiment more pipe spools and use simulation to quantify the performance improvement gained by using the DP algorithm. Another direction is to incorporate the DP algorithm with a simulation model so that fabrication sequences can be dynamically generated.

References

1. Kim, J.J., and Ibbs, C.W., "Work-Package-Process Model for Piping Construction", *Journal of Construction Engineering and Management*, 121(4): 381-387, 1995.
2. BRT, *Construction technologies needs and practices*, The Business Roundtable, Construction Industry Cost Effectiveness (CICE) Project Report B-3, New York, N.Y, 1982.
3. Hu, D and Mohamed, Y., “Effect of pipe spool sequence in industrial construction processes”, *Proc., 3rd International/9th Construction Specialty Conf.*, CSCE, Ottawa, Ont., 2011.
4. Gray, C., “Intelligent construction time and cost analysis”, *Journal of Construction Mgmt and Economics*, 4(2), 135–150, 1986.
5. Kartam, N., and Levitt, R. E., “Intelligent planning of construction projects”, *J. Computing in Civ. Engrg.*, ASCE, 4(2), 155–175, 1990.
6. Navinchandra, D., Sriram, D., and Logcher, R. D., "GHOST: Project Network Generator", *J. Computing in Civ. Engrg.*, ASCE, 2(3), 239-254, 1988.
7. Jin, Y., Kunz, J., Levitt, R., and Winstanley, G., "Design of Project Plans from Fundamental Knowledge of Engineered Systems", *Proc. AAAI Fall Symp. Design from Physical Principles*, Cambridge, MA, 149-154, 1992.
8. Echeverry, D., Ibbs, C. W., and Kim, S., “Sequencing knowledge for construction scheduling”, *J. Constr. Engrg. and Mgmt.*, ASCE, 117(1), 118–130, 1991.
9. Hendrickson, C, Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., and Lim, P., "Expert system for construction planning", *J. Comp. in Civ. Engrg.*, ASCE, 1(4), 253-269, 1987.
10. Darwiche, A., Levitt, R., and Hayes-Roth, B., "OARPLAN: Generating Project Plans by Reasoning about Objects, Actions and Resources", *AI EDAM*, 2(3), 169-181, 1989.
11. Aalami, F., Kunz, J., and Fischer, M., “Model-based sequencing mechanisms used to automate activity sequencing”, *Working Paper No. 50*, CIFE, Stanford Univ., Stanford, Calif., 1998.
12. Koo, B., Fischer, M., and Kunz, J., “Formalization of construction sequencing rationale and classification mechanism to support rapid generation of sequencing alternatives”, *J. Computing in Civ. Engrg.*, 21(6), 423–433, 2007.
13. Newell, A., and Simon, H., *Human problem solving*. Prentice Hall, Englewood Cliffs, N.J., 1972.
14. McDermott, D., and the AIPS-98 Planning Competition Committee, "PDDL–the planning domain definition language", *Technical report*, 1998 < http://www.cs.washington.edu/education/courses/cse473/06sp/pddl.pdf > (April, 2012).
15. Ghallab, M., Nau, D., and Traverso, P., *Automated Planning: Theory and Practice*, Elsevier Inc. San Francisco, 2004.
16. Hoffmann, J., "Extending FF to Numerical State Variables", *Proc., 15th European Conf. on Artificial Intelligence*, Lyon, France, July, 2002.
17. Coles, A.I., Fox, M., Long, D. and Smith, A.J., "A Hybrid Relaxed Planning Graph-LP Heuristic for Numeric Planning Domains", *Proc., Eighteenth Int. Conf. on Automated Planning and Scheduling (ICAPS 08)*, Sydney, Australia, September, 2008.
18. Gerevini, A. and Serina, I., "LPG: a Planner based on Local Search for Planning Graphs", *Proc., Sixth Int. Conf. on Artificial Intelligence Planning and Scheduling (AIPS'02)*, AAAI Press, Toulouse, France, 2002.
19. Hu, D., and Mohamed, Y., " Pipe spool fabrication sequencing by automated planning", *Proc., Construction Research Congress*, ASCE, West Lafayette, Indiana, USA, 2012.