HOSTED BY
ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Design and Engineering 2 (2015) 105–112
www.elsevier.com/locate/jcde

A comprehensive approach for managing feasible solutions in production planning by an interacting network of Zero-Suppressed Binary Decision Diagrams

Keita Takahashi*, Masahiko Onosato, Fumiki Tanaka

Graduate School of Information Science and Technology, Hokkaido University, Japan

Received 26 November 2014; received in revised form 5 December 2014; accepted 9 December 2014
Available online 7 January 2015

Abstract

Product Lifecycle Management (PLM) ranges from design concepts of products to disposal. In this paper, we focus on the production planning phase in PLM, which is related to process planning and production scheduling and so on. In this study, key decisions for the creation of production plans are defined as production-planning attributes. Production-planning attributes correlate complexly in production-planning problems. Traditionally, the production-planning problem splits sub-problems based on experiences, because of the complexity. In addition, the orders in which to solve each sub-problem are determined by priorities between sub-problems. However, such approaches make solution space over-restricted and make it difficult to find a better solution. We have proposed a representation of combinations of alternatives in production-planning attributes by using Zero-Suppressed Binary Decision Diagrams. The ZDD represents only feasible combinations of alternatives that satisfy constraints in the production planning. Moreover, we have developed a solution search method that solves production-planning problems with ZDDs. In this paper, we propose an approach for managing solution candidates by ZDDs' network for addressing larger production-planning problems. The network can be created by linkages of ZDDs that express constraints in individual sub-problems and between sub-problems. The benefit of this approach is that it represents solution space, satisfying whole constraints in the production planning. This case study shows that the validity of the proposed approach.

Keywords: Production planning; Production-planning attributes; ZDD

1. Introduction

Product Lifecycle Management (PLM) is known as one of the most effective approaches for product development and management [9]. Production planning is one of the phases in PLM. Production planning can be categorized into several domains, such as process planning, machine layout, production scheduling and so on [2,3,10]. In this research, key decisions in each domain for the creation of production plans are defined as production-planning attributes [4]. Since production-planning attributes correlate complexly in production-planning problems, production-planning problems have traditionally split sub-problems and have been solved by each sub-problem. If a solution in a sub-problem is infeasible in terms of the entirety of production planning, the sub-problem is resolved. Moreover, the orders in which to solve each sub-problem are determined by priorities between sub-problems, such as the first decision being process plans, the second decision being machine layout and so on. Such approaches make solution space over-restricted and make it difficult to find a better solution.

In this research, solution space in production planning is defined as comprehensive. We have proposed a representation of combinations of alternatives in production-planning attributes by using Zero-Suppressed Binary Decision Diagrams (ZDD) [6], which is are special type of Binary Decision Diagram (BDD) [1] used to represent a binary decision tree in graph form and are suitable for representing and processing combinatorial set data [4]. The ZDD was used to represent a

---

*Corresponding author.
E-mail address: ktakahashi@dse.ssi.ist.hokudai.ac.jp (K. Takahashi).

http://dx.doi.org/10.1016/j.jcde.2014.12.005

set of solution candidates that satisfy constraints in production planning [4]. As a result, the ZDD represented only feasible combinations of alternatives that satisfy constraints in production-planning attributes. Moreover, we have developed a solution search method that solves production-planning problems with ZDDs [5].

In this paper, we proposed an approach for managing solution candidates by ZDDs' network for addressing larger production-planning problems. The network can be created by linkages of ZDDs that express constraints in an individual sub-problem and between sub-problems. The benefit of this approach is to represent solution space, satisfying the entirety of constraints in production planning. Experimental results demonstrate that the interacting network of ZDDs is used for representing feasible solutions in production planning.

2. Production-planning domains and attributes

In this research, we categorized production planning into four domains [4].

- Process planning: creation of product information including operation processes.
- Resource planning: arrangement of equipment and workers for manufacturing execution.
- Execution planning: assignment of jobs to equipment and creation of production schedule.
- Order management: decision of boundary conditions in production processes.

Each domain has production-planning attributes. Table 1 shows the production-planning attributes. As shown in Table 1, 23 production-planning attributes are defined. Each production planning-attribute has alternatives. By combining alternatives in production-planning attributes, a production plan can be created. Each production-planning attribute has relationships to other attributes. A change of alternative may affect other alternatives in other attributes. Such influence may spread to the entirety of production planning. Therefore, the solution space of production planning is defined as comprehensive in this research [4]. For the representation of comprehensive solution space, we have used Zero-Suppressed Binary Decision Diagrams (ZDDs) [6]. The ZDD is described in detail in the next section.

Table 1
Production-planning domains and attributes [4].

Production-planning domains	Production-planning attributes
Process planning	Product redesign Grouping of process plan Manufacturing method Production process Process sequence Machine type Tool type Tool approach direction
Resource planning	Quantity of resource Available time Resource allocation Control rule
Execution planning	Job sequence Job routing Lot splitting Objective function Dispatching rule Constraint Inventory Placement Quantity of inventory
Order management	Due date Order cancellation Outsourcing

3. Representation of feasible solutions in production planning

3.1. Zero-Suppressed Binary Decision Diagrams

A ZDD is a directed graph representation of a Boolean function and can efficiently represent a set of combinations [6]. ZDDs have two terminal nodes, called 0-terminal node and 1-terminal node, and many decision nodes with two edges, called 0-edge and 1-edge. In order to represent a Boolean function efficiently, the following reduction rules are usually applied [6].

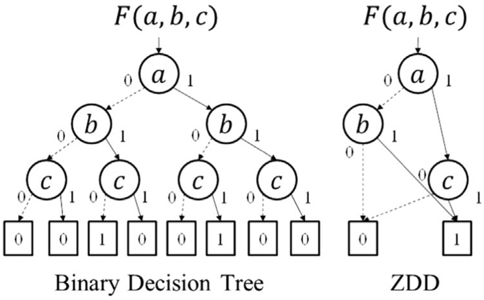

Fig. 1. Binary Decision Tree and ZDD.

1. Share equivalent nodes.
2. Delete all nodes of which 1-edge directly points to the 0-terminal node, and jump through to the 0-edge's destination.

Fig. 1 shows a ZDD representing Boolean function $F = a\bar{b}c \vee \bar{a}b\bar{c}$, which can be equivalently represented by using a set of combinations $\{ac\}, \{b\}$.

If a variable never appears within any elements in a set of combinations, a node representing the variable is removed from the ZDD.

3.2. VSOP

The Valued-Sum-Of-Products (VSOP) [8] program is for calculating combinatorial item set data specified by symbolic expressions based on the ZDD techniques [7]. The VSOP can efficiently handle large-scale sum-of-products expressions with a number of item symbols. Numerical arithmetic operations based on VSOP, such as addition, subtraction multiplication, division, numerical comparison and so on, are implemented. In this paper, we use the following operations for production-planning problems [8].

- Addition: $F+G$ means the union of F and G.
  (Ex) when $F=\{a,b\}$ and $G=\{c\}$, $F+G=\{a,b,c\}$
- Subtraction: F-G means the difference between $F$and G.
  (Ex) when $F=\{a,b,c\}$ and $G=\{a,c\}$, $F+G=\{b\}$
- Multiplication: $F\times G$ generates all possible concatenations of two items in respective F and G.
  (Ex) when $F = \{\, ab,\ b,\ c\}$ and $G = \{ab,\ 1\}$
  $$F\times\ G=(ab\times ab)+(ab\times 1)+\ (b\times ab)+\ (b\times 1)\\ +\ (c\times ab)+(c\times 1)=\{ab,abc,b,c\}$$
  "1" in G includes only a "*null*" item.
- Division: F/v (quotient) is to extract items that include variable v and remove v from the extracted items.
  Ex) when $F\ =\ \{ab,\ b,\ c\}$ and $v\ =\ \{b\}$, $F/v\ =\ \{a,1\}$
- Modulo: $F\%v$ (remainder) is to extract items that do not include variable v.
  (Ex) when $F=\{ab,b,c\}$ and $v\ =\{b\}$, $F\ \%\ v\ =\{c\}$
  Therefore, $F=v\times(F/v)\ +\ (F\%v)$
  Moreover, we use the *Restrict*, *Permitsym* and $If-Then-Else$ operation. These operations are described as follows.
- *Restrict* operation: $F.Restrict(G)$ extracts the product terms from F such that the item combination is a superset of at least one item combination in G.
  (Ex) when $F\ =\ \{b,ac,\ cd,\ abc,\ bcd\}$ and $G\ =\ \{a,bc\}$, $F.\ Restrict(G)=\{ac,abc,bcd\}$
- *Permitsym* operation: $F.Permitsym(n)$ filters the product terms in F, each of which consists of less than or equal to n items.
  (Ex) when $F=\{b,ac,cd,abc,bcd\}$ and $n=1$, $F.Permitsym(n)=\{b\}$
- $If-Then-Else$ operation: $F?G:H$ extracts the product terms from G such that the item combinations are included in F, and also extracts the terms from H for the item combinations not included in F.

Formulations for production planning in ZDDs are defined by the above operations. Other operations refer to other literatures [7,8] for more details.

3.3. Comprehensive representation of solution candidates of alternatives

In our previous research, we have used ZDDs to represent solution candidates that satisfy constraints in process planning and resource planning [4]. Four attributes are considered, namely process sequence, manufacturing method, machine type and quantity of resource. The results showed that ZDDs could represent only feasible combinations in production planning. Fig. 2 shows the approach for the ZDD that represents all solution candidates. In Fig. 2, circles of a solid line express production-planning attributes. Variables, such as O^1_{11}, mean alternatives. Circles of a broken line represent a set of solution candidates between production-planning attributes in the circle. The proposed approach is that all solution candidates are contained in a ZDD. Consider three production-planning attributes (A,B,C). First, ZDD X representing combinations of alternatives in A is created, and then combinations that satisfy constraints in A are extracted. Next, the ZDD X is extended, considering constraints between A and B. The extended ZDD X contains combinations of alternatives in A and B that satisfy constraints between them. Such manipulation is repeated until ZDD X contains all attributes. Finally, the ZDD X is extended, considering constraints between A, B and C. As a result, the ZDD X represents solution candidates in production planning. In the declaration of variables in the proposed approach, three types of variables are defined, which are O^s_{wr}, M^{swr}_i, and M_{ij}. O^s_{wr} is a variable of operation type w performed on the rth sequence for job s. M^{swr}_i is a variable of machine type i for operation type w performed on the rth sequence for job s. M_{ij} is a variable of instance j for machine type i. Variables of operation types in the process planning were declared based on jobs and sequences, and variables of machine types in the process planning were declared based on jobs, sequences and operation types. Such a declaration makes the number of variables increase if a problem size in production planning is large. Moreover, this will cause the increase of loads on a calculator and computational time. Therefore, the proposed method may not be suitable for large-sized problems in production planning. We proposed a comprehensive approach to represent solution candidates of such problems by an interacting network of ZDDs.

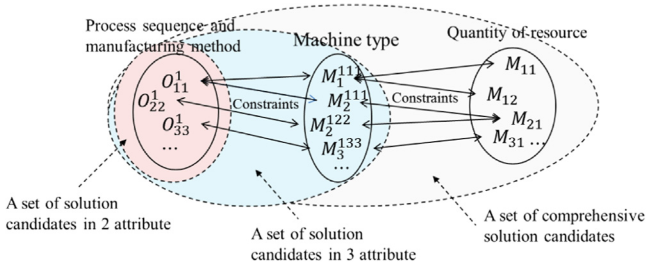

Fig. 2. Comprehensive representation of solution candidates.

4. Interacting network of ZDDs

4.1. Approach of interacting network of ZDDs

The previous approach was that feasible solutions were represented as a ZDD. In this paper, we propose an interacting network of ZDDs to represent feasible solutions in production planning. The feasible solutions are represented by the linkage of ZDDs. Fig. 3 shows the approach of the proposal method. In this approach, a ZDD represents feasible combinations of

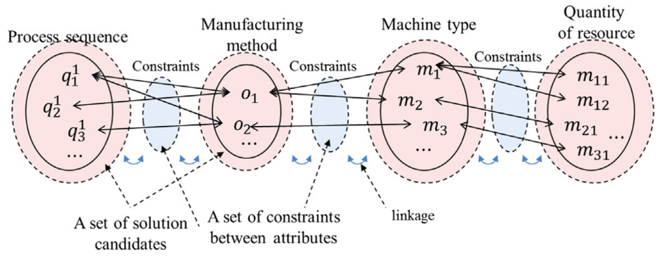

Fig. 3. Interacting network of ZDDs.

alternatives in a production-planning attribute or constraints between production-planning attributes. By linking such ZDDs, the network is created. This approach mainly can be categorized into three phases as follows.

1. Creation of all combinations of alternatives in each production-planning attribute.
2. Creation of constraints in each production-planning attribute and extraction of combinations that satisfy the constraints.
3. Creation of constraints between production-planning attributes and extraction of combinations that satisfy the constraints by linking ZDDs.

If the number of combinations of alternatives in a ZDD is changed, this network interacts and the number of combinations in other ZDDs may be changed by linkages of ZDDs that represent constraints. The range of use for the ZDD and each phase above is described from the following subsections.

4.2. Range of use in production planning

In this section, the combinatorial problem of production planning is formulated by the ZDD to represent only feasible combinations that satisfy constraints. The problem is formulated under the same assumptions as previous research as follows:

1. Four production-planning attributes in two production-planning domains are considered as the range of use for the ZDD: manufacturing method, process sequence, machine type and quantity of resource.
2. Each machine type has its instances with the same performance.
3. In a factory, the maximum number of installable machine instances is defined as J with respect to each machine type.
4. The sum of installable machine instances is less than or equal to F.

The formulation for the proposal approach is described in the following subsections.

4.3. Combinations in production-planning attributes

In this approach, each production-planning attribute has a ZDD that represents a set of combinations of alternatives. Moreover, constraints between production-planning attributes are also represented by a ZDD. We define four types of variables in ZDDs. Each variable represents an alternative in a production-planning attribute. The variables are defined as follows.

q_r^s: rth sequence of job s (process sequence)
o_w: operation type w (manufacturing method)
m_i: machine type i (machine type)
m_{ij}: instance j of machine type i (quantity of resource)

The first step is to create a set of all combinations of alternatives in each production-planning attribute. If a variable in a production-planning attribute is defined as $a_1, a_2, ..., a_x$, a ZDD X representing all combinations of alternatives is defined as follows.

$$X = (a_1 + 1) \times (a_2 + 1) \times \ldots \times (a_x + 1) \tag{1}$$

In Eq. (1), ZDD X represents all combinations including *null*. For example, if there are three alternatives in a production-planning attribute, X is $\{a_1a_2a_3, a_1a_2, a_1a_3, a_2a_3, a_1, a_2, a_3, 1\}$. The second phase is to create constraints in each production-planning attribute and to extract combinations that satisfy these constraints.

4.4. Constraints in production-planning attributes

Each production-planning attribute has its own constraints. In this problem, process sequence and quantity of resource have such constraints.

In the process sequence, combinations must include variables that express the first sequence. Therefore, combinations that satisfy this constraint include q_1^s. Moreover, the variables in combinations must appear in order of sequences. When job s has two feasible process sequences that consist of r and $r+1$ operations, this constraint is defined as follows.

$$C_s = \left(q_1^s \times q_2^s \times \ldots \times q_{r-1}^s \times q_r^s\right) + \left(q_1^s \times q_2^s \times \ldots \times q_r^s \times q_{r+1}^s\right) \tag{2}$$

When the number of jobs is S, combinations of feasible process sequences of jobs are defined as follows.

$$C_{all} = C_1 \times C_2 \times \ldots \times C_S \tag{3}$$

When A_s is a set of all combinations of alternatives in the process sequence, combinations of the alternatives that satisfy this constraint can be obtained by using $If-Then-Else$ operator as follows.

$$A_s \leftarrow A_s ? C_{all} : 0 \tag{4}$$

A_s is a set of combinations that satisfy constraints in the attribute of the process sequence.

In the quantity of resource, instances of a machine type have the same performance. For example, consider combinations that consist of two variables in this attribute. When machine type 2 has three instances and two instances are installed, combinations of instances can be represented as $\{m_{21}m_{22}, m_{21}m_{23}, m_{22}m_{23}\}$ in

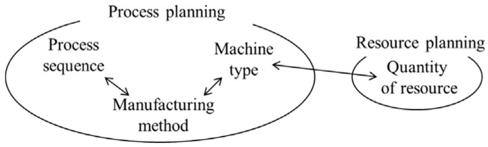

Fig. 4. Relationships between production-planning attributes.

ZDDs. Since instances of a machine type have the same performance, these combinations are redundant. Therefore, redundant combinations of machine instances should be removed from the ZDD. When A_r is a set of all combinations of alternatives in quantity of resource and the number of machine types is I, the removal algorithm is as follows.

Input I, J, ZDD A_r

```
Step 1: $i \leftarrow 1, j \leftarrow 1, ZDD\ b, c \leftarrow \{1\} // \{1\}$ is "null"
Step 2: $c \leftarrow c \times m_{ij}$
Step 3: $j++$, return to Step 2 until $j > J$
Step 4: $l \leftarrow J$, ZDD $t \leftarrow c$
Step 5: $c \leftarrow c / m_{il}$
Step 6: $t \leftarrow t + c$
Step 7: $l--$, return to Step 5 until $l < 1$
Step 8: $b \leftarrow b \times (t+1)$
Step 9: $c \leftarrow \{1\}, j \leftarrow 0, i++$, return to Step 2 until $i > I$
Step 10: $A_r \leftarrow A_r ? b : 0$
```

In the second step of this algorithm, a set of combinations that represent J instances is created. For example, if machine type 1 has two instances, this step creates a ZDD that contains $\{m_{11} m_{12}\}$. Through the fifth and sixth step, ZDD t is created and contains $\{m_{11} m_{12}, m_{11}\}$. Finally, combinations in A_r that satisfy this constraint are extracted by the tenth step.

If the number of installable instances of machine type α is β and less than J ($\beta < J$), the removal algorithm is as follows.

Input β, J, ZDD A_r

```
Step 1: $j \leftarrow \beta$, ZDD $ng \leftarrow \emptyset$
Step 2: $ng \leftarrow ng + M_{\alpha j}$
Step 3: $j++$, return to Step 2 until $j > J$
Step 4: $A_r \leftarrow A_r - A_r.Restrict(ng)$
```

In the case of the above example, A_r as the input of this algorithm is $\{m_{11} m_{12}, m_{11}\}$. When the number of installable instances of machine type 1 is 1, A_r becomes $\{m_{11}\}$ through this algorithm.

Next, the constraint of the sum of installable machine instances is considered. If a combination of alternatives in quantity of resource contains more than F variables, such combinations do not meet this constraint. The combinations are removed from the ZDD. The removal operation is defined as follows.

$$A_r \leftarrow A_r.Permitsym(F) \tag{5}$$

Finally, A_r satisfies constraints in the quantity of resource.

Each ZDD in production-planning attributes represents combinations that satisfy constraints in each attribute. The next phase is to create constraints between production-planning attributes and to extract combinations that satisfy the constraints. For the constraint satisfaction, the ZDDs interact with each other.

4.5. Constraints between production-planning attributes

Alternatives in each attribute have relationships to other alternatives in other attributes. For example, attributes of manufacturing methods and machine types are closely related. If an alternative in the manufacturing method is decided upon, some alternatives in machine types may not be able to be selected. Fig. 4 shows such relationships between production-planning attributes. This relationship is analyzed qualitatively. If the number of combinations of alternatives in the process sequence is changed, combinations of alternatives in the manufacturing method may exert influence due to the change. In this paper, the relationships between production-planning attributes are defined as constraints in terms of ZDDs.

For the representation of the constraints, a union of combinations between two production-planning attributes is created as the first step. Consider alternatives $(a_1, a_2, a_3, \ldots, a_x)$ in production-planning attribute A and alternatives $(b_1, b_2, b_3, \ldots, b_y)$ in production-planning attribute B. If a_1 can be matched with b_1 or b_2, such combinations can be represented as $a_1 (b_1 + b_2)$ in ZDDs. The union contains these combinations, which means feasible combinations between two production-planning attributes. Combinations of alternatives in two production-planning attributes that do not satisfy constraints between them can be removed by using the union.

For example, A and B is a set of combinations of alternatives in a production-planning attribute. When the production-planning attributes have relationships with each other, there are constraints between them. When ZDD U_{AB} represents the constraints as the union of feasible combinations of alternatives, the following algorithm is used to extract combinations of alternatives in each production-planning attribute that satisfies constraints between them.

Input ZDD A, B, U_{AB}, X, Y

```
Step 1: ZDD $t_1 \leftarrow A$, ZDD $t_2 \leftarrow B, x \leftarrow 1, y \leftarrow 1$
Step 2: $t_1 \leftarrow (t_1 / a_x) \times (U_{AB} / a_x) + t_1 \% a_x$
Step 3: $x++$, return to Step 2 until $x > X$
Step 4: $t_2 \leftarrow (t_2 / b_y) \times (U_{AB} / b_y) + t_2 \% b_y$
Step 5: $y++$, return to Step 4 until $y > Y$
Step 6: $A \leftarrow A ? t_1 : 0$
Step 7: $B \leftarrow B ? t_2 : 0$
```

X and Y is the number of alternatives in each production-planning attribute. In the second step of this algorithm, (t_1 / a_x) extracts combinations that include a_x in t_1 and remove a_x from the extracted combinations. (U_{AB} / a_x) extracts combinations that include a_x in U_{AB} and remove a_x from the extracted combinations as well. $t_1 \% a_x$ extracts combinations that do not include a_x

in U_{AB}. The second and fourth steps are the same operation. Through the second and forth steps, combinations of alternatives that satisfy constraints between two production-planning attributes can be created. Finally, the combinations of alternatives in the attributes satisfy constraints between the two attributes by the sixth and seventh step.

If the number of combinations in A or B is changed, this change may influence other attributes. For example, when the number of B is changed through the above algorithm and the production-planning attribute of B has relationships with a production-planning attribute, constraints between them are considered as the next step. The above algorithm is applied until such spreads stop. By using this algorithm, ZDDs that represent combinations of alternatives and ZDDs that represent constraints between production-planning attributes are linked. The algorithms are repeated in accordance with Fig. 4 until all sets of combinations of alternatives in production-planning attributes are not changed.

5. Sample problems

5.1. Experiment conditions and implementation

In this experiment, the ZDDs' network is applied to Problem 1 and Problem 2. In Problem 1, the number of part types is 4, which is $P_1 \sim P_4$ in Fig. 5. A job consists of a part type. Each part type is produced by one part types. The number of operation types is 9, which is $O_1 \sim O_9$. Fig. 5 shows the process sequences for each part type. In Fig. 5, there are three types of nodes, namely, starting node, intermediate node, and ending node. Both the starting node and the ending node are dummy ones that indicate the beginning and the end of the manufacturing method of a part, respectively. An intermediate node represents an operation. The arrow that connects two nodes indicates the precedence relations between the nodes. Moreover, there are two types of brackets. The vertical bars (| |) enclosing nodes mean alternative and the square brackets ([ ]) enclosing nodes express arbitrary sequence order. The alternative means $(O_1 \rightarrow O_2 \rightarrow O_3), (O_3 \rightarrow O_5 \rightarrow O_4)$ or $(O_3 \rightarrow O_1 \rightarrow O_4)$ is selectable in P_1. The arbitrary sequence order means O_4 in P_3 can be the first process or the second process as well as O_7. Table 2 represents information of each operation and machine type in the Problem 1 and 2. The row of "Instance" expresses the number of instances of each machine type. In this problem, the number of machine types is 5, which is $M_1 \sim M_5$. Each machine type has instances. In terms of the resource planning, four machine type instances out of eight can be installed in a factory.

In Problem 2, the number of part types is 10, which is $P_1 \sim P_{10}$ in Fig. 5. Each part type is produced by two part types. Therefore, the number of jobs is 20. The number of operation types is 15, which is $O_1 \sim O_{15}$. The number of machine types is 8, which is $M_1 \sim M_8$. The total number of machine instances is 12. In terms of the resource planning, 4 machine type instances out of 12 can be installed in a factory.

All the algorithms implemented in this paper were developed in VSOP coded on Ruby and run on a workstation with a Xeon E5-2643 3.30 GHz and 256GB memory.

Table 2
Information of operation and machine types for problems.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
Instance	3	1	1	2	1	1	2	1
O_1	*	*					*	
O_2		*				*		
O_3			*	*				
O_4	*			*				
O_5	*		*		*			
O_6		*		*	*			
O_7	*	*				*		
O_8			*					
O_9				*				*
O_{10}						*	*	
O_{11}			*		*			*
O_{12}		*				*	*	
O_{13}	*					*		*
O_{14}				*				*
O_{15}					*		*	

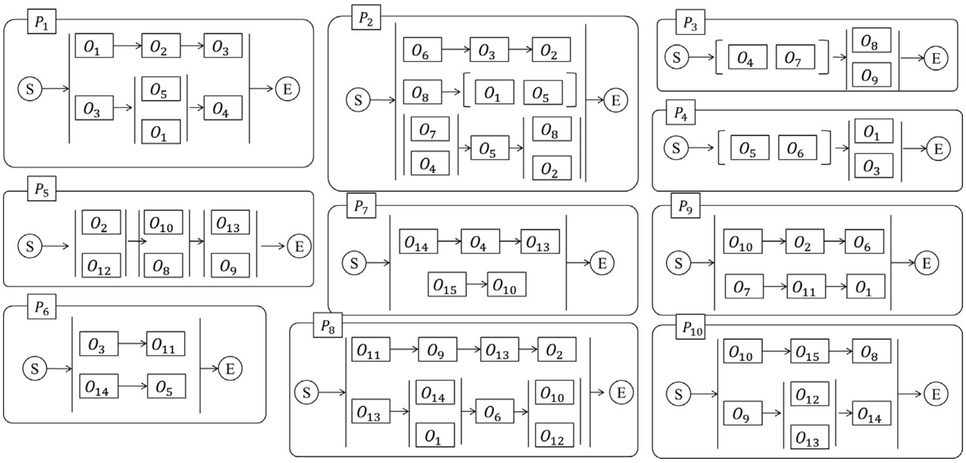

Fig. 5. Process sequence for each part types.

Table 3
Set of combinations of alternatives in production-planning attributes in Problem 1.

Attribute	Phase	Number of nodes	Number of combinations
Process sequence	I	12	4096
	II	12	1
	III	12	1
Manufacturing method	I	9	512
	II	9	512
	III	21	9
Machine type	I	5	32
	II	5	32
	III	11	11
Quantity of resource	I	15	32,768
	II	17	59
	III	20	19

Table 4
Set of combinations of alternatives in production-planning attributes in Problem 2.

Attribute	Phase	Number of nodes	Number of combinations
Process sequence	I	80	Approx. 1.2×10^{24}
	II	60	4
	III	60	4
Manufacturing method	I	15	32,768
	II	15	32,768
	III	142	54
Machine type	I	8	256
	II	8	256
	III	42	23
Quantity of resource	I	24	16,777,216
	II	39	891
	III	104	27

5.2. Results and discussion

Tables 3 and 4 show the results of the initial set of combinations of alternatives in production-planning attributes in Problem 1 and 2, respectively. In the phase column, Phase I means the creation of all combinations of alternatives as initial solution space. Phase II expresses the creation of combinations of alternatives that satisfy constraints in a production-planning attribute. Phase III means the creation of combinations of alternatives that satisfy all constraints between production-planning attributes. The column of the number of nodes means how many nodes are included in a ZDD. Number of combinations is the number of combinations of each production-planning attribute. Since the number of nodes is the same as the number of variables defined in each production-planning attributes in the initial set, the number of combinations in each production-planning attributes is two to the power of the number of nodes. Therefore, the number of variables in Problem 1 and 2 was 41 and 127, respectively.

Table 5
Comparison of previous method and proposed method.

Problem	Method	Number of variables	Computational time[s]
1	Comprehensive representation	991	0.28
	Interacting network	41	0.06
2	Comprehensive representation	10,824	57.10
	Interacting network	127	11.80

In Problem 1, the number of feasible combinations of alternatives in the process sequence was 1, because P_1, P_2, P_3 and P_4 consist of three operations. The combination could be obtained by adding constraints in the process sequence to the ZDD representing all combinations of alternatives in the attribute. Since the manufacturing method and machine type do not have constraints in their own attribute, the number of combinations of alternatives was not changed. However, constraints between production-planning attributes removed the combinations in Phase II. This results means the interacting network of ZDDs worked and combinations that do not satisfy constraints were removed from each ZDD that represents a set of combinations of alternatives in production-planning attributes. The solution space of Problem 2 is larger than Problem 1. The number of initial combinations of alternatives in the process sequence is large. However, the number of combinations was greatly decreased by the constraints in the process sequence. Moreover, feasible combinations in other attributes were extracted by applying the constraints between attributes and Problem 1.

We also applied our previous method [4] to Problem 1 and 2. Table 5 shows the comparison results of the comprehensive representation and the interacting network. As shown in Table 5, the number of variables of the interacting network in Problem 1 and 2 was reduced by about 95.9% and 98.9%, respectively. Moreover, computational time of the interacting network in Problem 1 and 2 was reduced by about 78.6% and 79.3%, respectively. The results show the proposed method could represent feasible combinations in production planning with more efficiency.

ZDDs can also enumerate these combinations. For a creation of a production plan, a combination in each attribute is selected. In Problem 1, the combination of alternatives in the process sequence is determined uniquely, since only one combination is feasible. If combination $\{o_1o_2o_3o_4o_5o_6o_7o_8\}$ is selected in the manufacturing method, this selection influences the machine type. Table 6 shows a change of the number of combinations in each attribute due to the influence of selections. The column of Selection I expresses the change of the number of combinations after the combination $\{o_1o_2o_3o_4o_5o_6o_7o_8\}$ was determined. By this selection, the

Table 6
Influence of selections in problem 1 on ZDDs' network as an example.

Attribute	Original	Selection I	Selection II
Process sequence	1	1	1
Manufacturing process	9	1	1
Machine type	11	5	1
Quantity of resource	19	7	1

machine type and the quantity of resource are influenced and the number of combinations in them is reduced. The column of Selection II means the change of the number of combinations after the combination $\{m_1m_2m_3m_5\}$ was selected. This selection influences the quantity of resource as well. As a result of this influence, only one combination of alternatives in the quantity of resource is left. Finally, the left combination $\{m_{11}m_{21}m_{31}m_{51}\}$ is uniquely determined and a feasible production plan can be created by using alternatives in determined combinations.

6. Conclusion

In this paper, we proposed an interacting network of ZDDs to manage feasible solutions in production planning. By managing combinations of alternatives in each production-planning attribute and constraints in and between them, the network is created. The experimental result shows the network-suppressed combinations of alternatives in each production-planning attribute that do not satisfy constraints in its attribute and between related attributes. As shown in Table 5, the number of variables and computational time is decreased. When a ZDD that represents a set of combinations of alternatives in a production-planning attribute is changed due to a decision of some alternatives, ZDDs of related attributes with the attribute are changed in conjunction with the change ZDD through the proposed algorithm. As a result, the ZDDs' network updates each change and maintains feasible combinations.

Future works will focus on the development of an algorithm to find the quasi-optimal solution from the interacting network of ZDDs.

Conflict of interest

The authors declare no conflict of interest associated with this manuscript.

Acknowledgments

We are grateful to Prof. Shin-ichi Minato at the Graduate School of Information Science and Technology, Hokkaido University, for providing advice as well as for his permission to use VSOP coded on Ruby for this research.

References

[1] Bryant RE. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers* 1986;**vol. C-35**(8)677–91.

[2] Aytug Haldun, Lawley Mark A, Mckay Kenneth, Mohan Shantha. Executing production schedules in the face of uncertainties: a review and some future directions. *Eur. J. Oper. Res.* 2005;**161**(1)86–110.

[3] Ripon Kazi Shah Nawaz, Glette Kyrre, Hovin Mats, Torresen Jim. Dynamic facility layout problem under uncertainty: a Pareto-optimality based multi-objective evolutionary approach. *Central European Journal of Computer Science* 2011;**1**(4)375–86.

[4] Keita Takahashi, Masahiko Onosato, Fumiki Tanaka. Comprehensive representation of feasible combinations of alternatives for dynamic production planning using Zero-Suppressed Binary Decision Diagram. In: Proceedings of the 7th International Conference on Leading Edge Manufacturing in 21st Century; 2013; pp. 559–564.

[5] Keita Takahashi, Masahiko Onosato, Fumiki Tanaka. A solution method for comprehensive solution candidates by Zero-Suppressed Binary Decision Diagram in dynamic production planning. In: Proceedings of the 2014 International Symposium on Flexible Automation; 2014; 2014-66L.

[6] Shin-ichi Minato, Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems. In: Proceedings of the 30th ACM/IEEE Design Automation Conference; 1993; pp. 272–277.

[7] Minato Shin-ichi. Zero-suppressed BDDs and Their applications. *International Journal on Software Tools for Technology Transfer* 2001;**3**(2) 156–70.

[8] Minato Shin-ichi. VSOP (Valued-Sum-of-Products) calculator for knowledge processing based on Zero-Suppressed BDDs. *Lecture Notes in Computer Science* 2006;**3847**:40–58.

[9] Ming XG, Yan JQ, Lu QF, Ma DZ, Song B. Mass production of tooling product families via modular feature-based design to manufacturing collaboration in PLM. *Journal of Intelligent Manufacturing* 2007;**18**(1) 185–95.

[10] Liu Xiao-jun, Hong Yi, Ni Zhong-hua. Application of ant colony optimization algorithm in process planning optimization. *Journal of Intelligent Manufacturing* 2013;**24**(1)1–13.