International Conference on Computer Applications in Shipbuilding (ICCAS-2009) Shanghai, China Vol.2, pp.121--126.

AUTOMATIC DESIGN FOR PIPE ARRANGEMENT CONSIDERING VALVE OPERATIONALITY

H Kimura, Kyushu University, Japan
S Ikehira, Kyushu University, Japan

SUMMARY
We propose a novel evaluation method of valve operationality for pipe-arrangement design, and multi-objective genetic algorithms (GAs) suitable for the pipe-arrangement problems are developed. The pipe-arrangement design in shipbuilding requires not only to arrange all pipes without interference but also to make space from pathways to valves so that crew can access the valves. For this reason, it needs sophisticated skill, and is considered that numerical evaluation for pipe-arrangement designs is difficult. To evaluate pipe-operationality, we propose a recursive-fill algorithm to judge whether the arranged valves are accessible by crew or not. In the proposed genetic algorithms, only the parameters of location and directions of valves are encoded, and the other parameters are generated by a modification operators that are similar to local-search algorithms. The proposed approach is demonstrated through computational experiments that contain five valves, one pump and five branches in the diagram, and remarks are provided for applying this methodology to practical pipe-arrangement design problems.
~~~~~~
~~~~~~

1. INTRODUCTION

Pipe arrangement design in shipbuilding needs highly sophisticated skill, and it is considered that numerical evaluation or automatic arrangement are quite difficult. One reason is that considered items are plural; Material costs, assembly costs, convenience of valve operation or pipe-maintenance, etc. The other reason is that many considered items are ambiguous or defined without numerical evaluation. Therefore, automatic pipe arrangement will be possible by means of

- Defining numerical evaluation for all considered items , and
- Formulating the pipe-arrangement problem into a multi-objective optimization problem.

Based on this idea, we propose a new approach to solve the problem.
The pipe-arrangement design in shipbuilding requires not only to arrange all pipes without interference but also to make space from pathways to valves (or some other equipment) so that crew can access the valves. This accounts largely for requiring sophisticated design skills.
Since the designing know-how is implicit, the evaluation of the pipe-arrangement design is ambiguous and it is considered that numerically sound evaluation is difficult.
In our previous works, we proposed a formulation of pipe-arrangement problems including valves and pipe-branches to develop an automatic designing system that generates rough arrangement of pipes and equipments for CAD operators[1][2][3]. In this paper, we propose a novel numerical evaluation algorithm of valve operationality, and formulate a pipes-and-valves-arrangement problem as a multi-objective optimization problem. Since it includes not only numerical but combinatorial optimization, we developed multi-objective genetic algorithms (GAs) suitable for the pipe-arrangement problems. The proposed approach is demonstrated through computational experiments that contain five valves, one pump and five branches in the diagram, and remarks are provided for applying this methodology to practical pipe-arrangement design problems.

2. AN EVALUATION ALGORITHM FOR VALVE OPERATIONALITY

2.1 VALVE OPERATIONALITY

Valves are essential equipment manipulated by crew. Since almost valves are worked by hand, the pipes and valves must be arranged to make space from pathways to the valves so that the crew can access and operate the valves. Here we refer to the evaluation of the enough space from pathways to the valves as "valve operationality", and defined as below.

2.1 (a) Accessibility

We define that a valve is "accessible" in the case of

- Crew can move to a position where the valve can be operated by hand, or
- Crew can move to a position where the valve cannot be operated by hand, but can be operated by some tools (rods, etc.).

The arrangement plan must be accessible for all valves.

2.1 (b) Possibility of Valve Handling

We define "possibility of valve handling" as the valve can be operated by hand. In the two cases above, it is obvious that the arrangement plan of the former is superior than the latter because the former does not need any tools for the valve operation.

2.2 EVALUATION ALGORITHM

In this section we describe an algorithm for calculate the accessibility and the convenience of valve handling.

International Conference on Computer Applications in Shipbuilding (ICCAS-2009) Shanghai, China
Vol.2, pp.121--126.

2.2 (a) Overview of The Algorithm

In order to evaluate valve operationality, we need to recognize whether the space from the pathway to the valves is enough or not for the crew. Therefore, we partition the design space into regular grids, and distinguish free cells from ones occupied by obstacles. Thereafter, the accessibility and the possibility of valve handling are checked in the cellular space as below. We refer to the cells that include or intersect obstacles (i.e., pipes, valves, or other equipments) as "obstacle segments", and the cells that are put in pathways as "aisle segment". We define "worker-segment matrix" as aggregated cells imitating shape of the crew (worker). The cells that are swept by the worker-segment matrix without intersecting obstacle segments starting from an aisle segment are recognized as "accessible segments". Then, both the accessibility and possibility of valve handling for an arbitrary valve are easily evaluated by counting the grids between the cell containing the corresponding valve and the nearest accessible segments. Figure 1 shows an example of the calculation procedure.

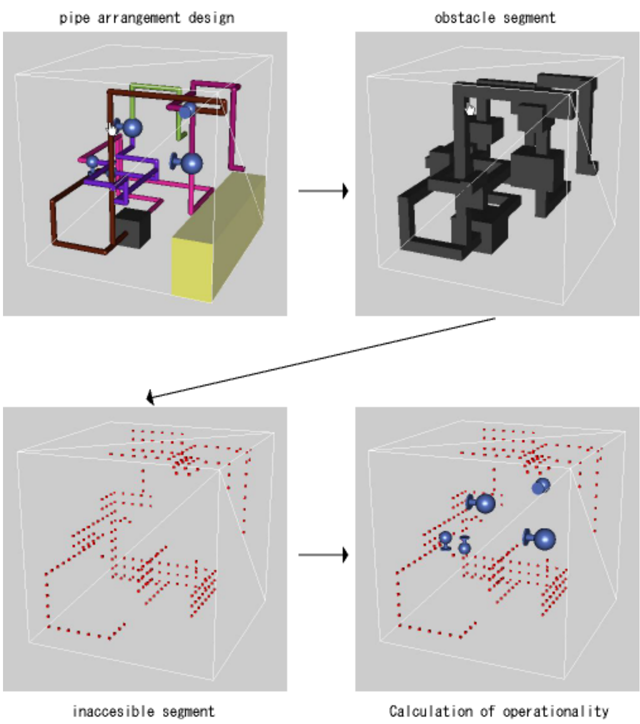


Figure 1: An example of the calculation procedure of the valve operationality

2.2 (b) The Detailed Explanation of the Algorithm

1. Partition the design space into grids at regular intervals of some distance k.
2. Find obstacle segments that include or intersect obstacles.
3. Find aisle segments that are put in pathways.
4. Define worker-segment matrix in the size of $\ell \times m \times n$.
5. Recognize accessible segments using Recursive Fill algorithm.
6. Find minimum distance from each valve to accessible segments that are located in the direction of the axis of the valve's handle or four directions perpendicular to that axis.
7. Sum up the minimum distance over all valves.

Figure 2 shows the detail of the calculation procedure. Recursive Fill algorithm shown in the step 5 is described in the next section.

Figure 2: Valve operationality evaluation algorithm

2.2 (c) Recursive Fill Algorithm

This algorithm is used to recognize accessible segments. It is a kind of 3D-sweeping algorithm that the worker-segment matrix is swept in the design grid-space without intersecting obstacle segments. The outline of the procedure is described below.
Select one cellular segment where worker-segment matrix can be put without intersecting obstacle segments in aisle segments. Expressing the coordinate of the selected segment as (x_0, y_0, z_0), then execute a method $fill(x_0, y_0, z_0)$. In the method of $fill(x, y, z)$, the corresponding cellular segment (x, y, z) is checked whether it is already labelled as accessible or not. When the segment is not labelled, it is checked whether the worker-segment matrix can be put without intersecting obstacle segments at this segment. When it is possible, the method $fill()$ is recursively applied to neighbour six cellular segments (up, down, back, forth, right and left). Using C like language expression, the method $fill(x, y, z)$ is described as Figure 3. The function WorkerSegmentMatrix_intersects_with_ObstacleSegment_at(segment[x][y][z]) returns false when the worker-segment matrix containing segment[x][y][z] can be put without intersecting obstacle segments.

International Conference on Computer Applications in Shipbuilding (ICCAS-2009) Shanghai, China
Vol.2, pp.121--126.

```
fill( int x , int y , int z){
    if(segment[x][y][z]==NotLabeled){
       if( WorkerSegmentMatrix_intersects_with_Ob
    stacleSegment_at(   segment[x][y][z] ) ==
    false ){
            segment[x][y][z] = AccessibleSegment ;
            fill( x+1 , y , z ) ;
            fill( x-1 , y , z ) ;
            fill( x , y+1 , z ) ;
            fill( x , y-1 , z ) ;
            fill( x , y , z+1) ;
            fill( x , y , z-1) ;
         }else{
            segment[x][y][z] = InaccessibleSegment ;
        }
      }else{
         return;
      }
}
```

Figure 3: Recursive Fill algorithm: fill(x,y,z)

3. GENERATING CANTIDATES FOR THE SOLUTION OF THE PIPE-ARRANGEMENT PROBLEM

3.1 FORMULATION AS A MULTI-OBJECTIVE OPTIMIZATION PROBREM

In our previous works, we proposed a formulation of pipe-arrangement problems including valves and pipe-branches as an single-objective optimization problem. In this paper, we consider two-objective functions; One is the valve operationality introduced in the previous section. The other is a material cost defined as follows:

$$f_{\_material} = \sum_{k=1}^{n_p} W_k L_k D_k$$, where

k : Index of the corresponding pipe,

W_k : Material cost per unit weight of the pipe k,

L_k : Length of the pipe k,

D_k : Diameter of the pipe k.

In this section, we explain our optimization algorithm to solve the pipe arrangement design problems.

3.2 NONDOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)

In multi-objective optimization, since plural objective functions that have trade-off relations each other must be considered, selection strategy for generation alternation is very important for multi-objective genetic algorithms (MOGA). Deb's NSGA-II [4] and Zitzler's SPEA-II [5] are representative methods in MOGA. In this paper, we adopt NSGA-II for the selection strategy.

3.3 DESIGNING GENE CODING

In our pipe-arrangement problems, since the design parameters are locations and directions of valves, bending patterns of path of pipes and length of the straight parts of the pipes, it is a combination of numerical optimization and combinational optimization. Therefore, conventional coding methods such as binary coding, gray coding, real-coding, etc. cannot be applied without modification. Putting all design parameters into a matrix as genes is the most simple approach, but it is quite wasteful because many design parameters are strongly related each other or depend on the other parameters. For example, if the location of valves are changed, the parameters of the pipes that are connected the removed valves would not be feasible any more. We consider that the locations and directions of the valves are dominant to ones of the pipes. For this reason, we encode only the parameters of the locations and directions of the valves as the genes for the MOGA. Let (x_i, y_i, z_i) be a location of a valve i, where each component is continuous variable as shown in Figure 4, and let θ_i be the direction of the handle of the valve i, where θ_i takes one of twelve discrete values as shown in Figure 5.

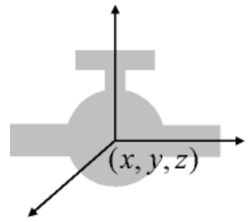


Figure 4: Location of Valve

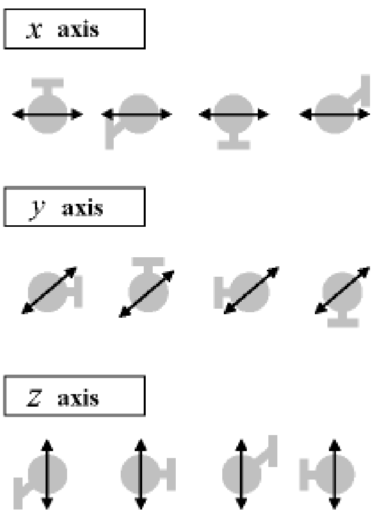


Figure 5: Direction of Valve

In this paper, locations and directions of all valves are expressed by the following matrix as the genome of the valves:

International Conference on Computer Applications in Shipbuilding (ICCAS-2009) Shanghai, China
Vol.2, pp.121--126.

$$\begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_n \\ x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{bmatrix}.$$

This coding does not inhere the information of the pipes. Pipes are arranged by the other search algorithm [3] after the locations and directions of the valves are given.

3.4 DESIGNING CROSSOVER OPERATOR

We apply one-point crossover to the genome. It selects a single crossover point randomly on both parent's organism strings, and all data beyond that point in either organism string is swapped between the two-parent organisms. The resulting organism are children as shown in Figure 6.

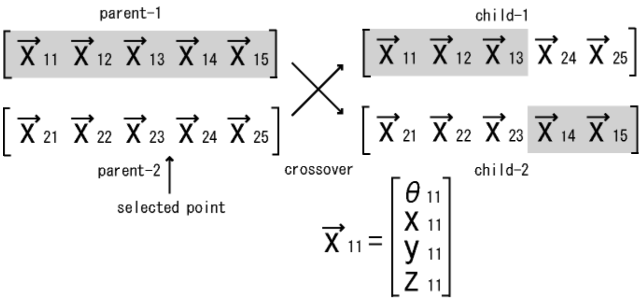


Figure 6: One-Point Crossover

3.4 DESIGNING MUTATION OPERATOR

The mutation is applied to each generation genome at some constant probability referred as a mutation rate. The genome to which the mutation is applied is selected at random following constant probability.

3.5 DESIGNING MODIFICATION OPERATOR

In this paper, a solution candidate of the arrangement of valves is regarded as infeasible when the valves interfere with each other. It is too difficult to find feasible solutions using only naïve genetic operators because the probability of finding feasible solutions exponentially decreases as the number of valves increases when the candidates are generated at random. Therefore, the infeasible solution candidates of the valve arrangement are modified to feasible candidates. We refer to this operation as modification operator for valve (MOV).

4. EXPERIMENTS AND RESULTS

4.1 A SAMPLE PROBLEM FORMULATION

To confirm the effectiveness of the proposed method, it was applied to a pipe arrangement design problem that have five valves, one pump, and five branch of pipes as shown in Figure 7. Its equipment tables are shown in Figure 8, and the pipeline list shown in Figure 9 specifies the connection between the pipes and equipments.

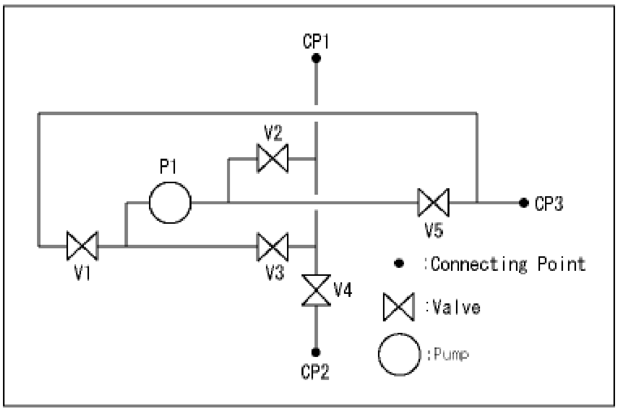
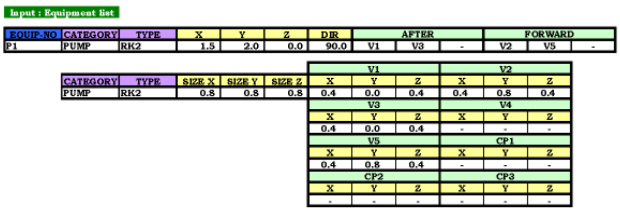


Figure 7: Pipe diagram in the experiments



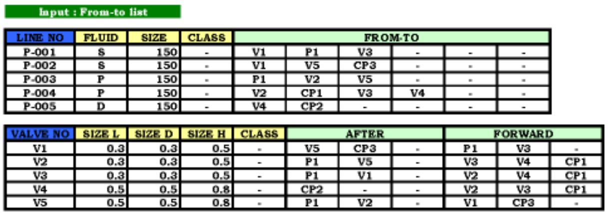
Input : Equipment list

EQUIP-NO	CATEGORY	TYPE	X	Y	Z	DIR	AFTER			FORWARD		
P1	PUMP	RK2	1.5	2.0	0.0	90.0	V1	V3	-	V2	V5	-

CATEGORY	TYPE	SIZE X	SIZE Y	SIZE Z
PUMP	RK2	0.8	0.8	0.8

V1			V2		
X	Y	Z	X	Y	Z
0.4	0.0	0.4	0.4	0.8	0.4
V3			**V4**		
X	Y	Z	X	Y	Z
0.4	0.0	0.4	-	-	-
V5			**CP1**		
X	Y	Z	X	Y	Z
0.4	0.8	0.4	-	-	-
CP2			**CP3**		
X	Y	Z	X	Y	Z
-	-	-	-	-	-

Figure 8: Equipment arrangement list



Input : From-to list

LINE NO	FLUID	SIZE	CLASS	FROM-TO					
P-001	S	150	-	V1	P1	V3	-	-	-
P-002	S	150	-	V1	V5	CP3	-	-	-
P-003	P	150	-	P1	V2	V5	-	-	-
P-004	P	150	-	V2	CP1	V3	V4	-	-
P-005	D	150	-	V4	CP2	-	-	-	-

VALVE NO	SIZE L	SIZE D	SIZE H	CLASS	AFTER			FORWARD		
V1	0.3	0.3	0.5	-	V5	CP3	-	P1	V3	-
V2	0.3	0.3	0.5	-	P1	V5	-	V3	V4	CP1
V3	0.3	0.3	0.5	-	P1	V1	-	V2	V4	CP1
V4	0.5	0.5	0.8	-	CP2	-	-	V2	V3	CP1
V5	0.5	0.5	0.8	-	P1	V2	-	V1	CP3	-

Figure 9: From-To list (Pipeline list)

4.2 RESULTS

The proposed method was compared with a simpler method in which a random search is substituted for the crossover operator. It is not affected to the framework of the NSGA-II. Size of the population is set to 10, the parameter α of modification parameter for pipes (MOP)[3] is 10, and the rate for the mutation operator is 0.1. All timings are reported for a Pentium IV 2.4GHz processor with 512MB RAM running Java program codes on Microsoft WindowsXP. Figure 10, 11, 12 are the candidates of the solution at the initial generation,, 400th generations and 450th generations respectively. The horizontal axis denotes the valve operationality (f_operationality), and the vertical axis denotes the materials cost. The nearer candidates to the origin are the better. In the 400th generation, 4.0×10^3 individuals are evaluated and it takes about 60 minutes for the calculation. In the 450th generation, 4.5×10^3

International Conference on Computer Applications in Shipbuilding (ICCAS-2009) Shanghai, China
Vol.2, pp.121--126.

individuals are evaluated and it takes about 70 minutes. Figure 13 and 14 show the 3-D models generated by the proposed algorithm using one-point crossover in the 450th generations. The white wireframe denotes the design space, the transparent cuboid is a pathway, and the black cuboid is a pump. The pipes with the same colours means the same pipeline.

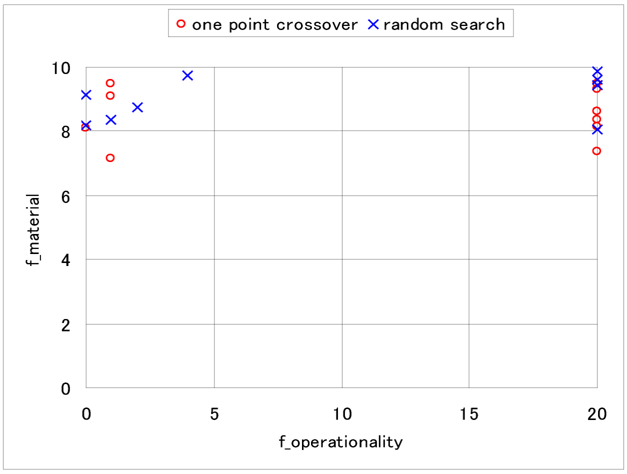


Figure 10: Solution candidates at initial population

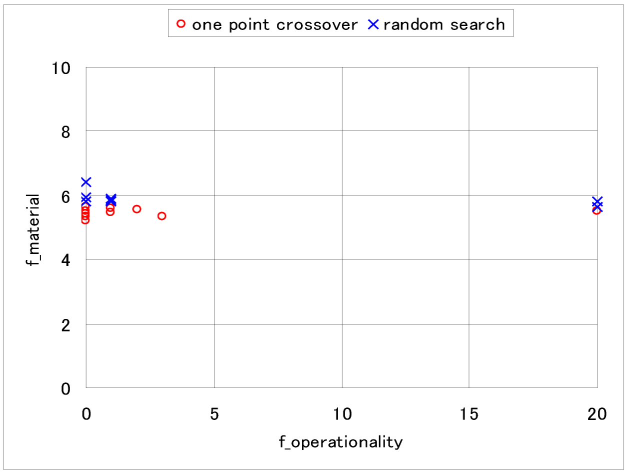


Figure 11: Solution candidates at 400th generation

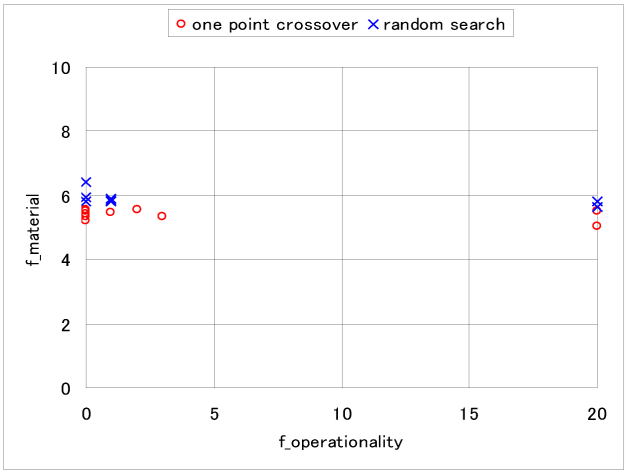


Figure 12: Solution candidates at 450th generation

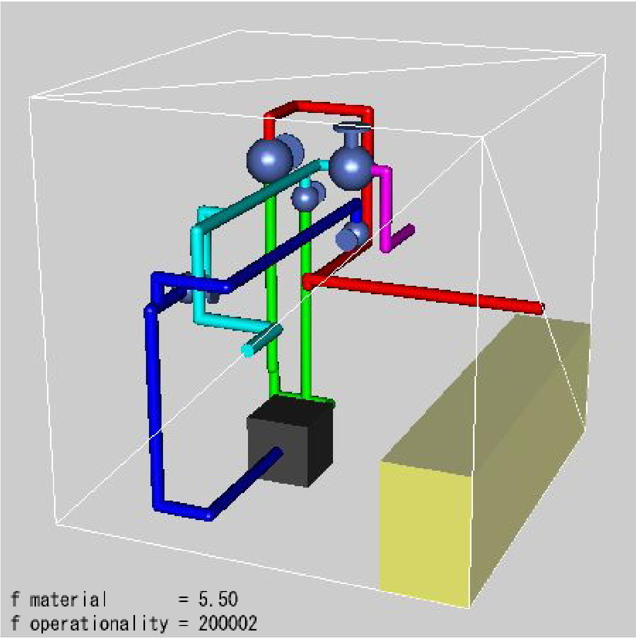


Figure 13: The result of the best design in materials cost

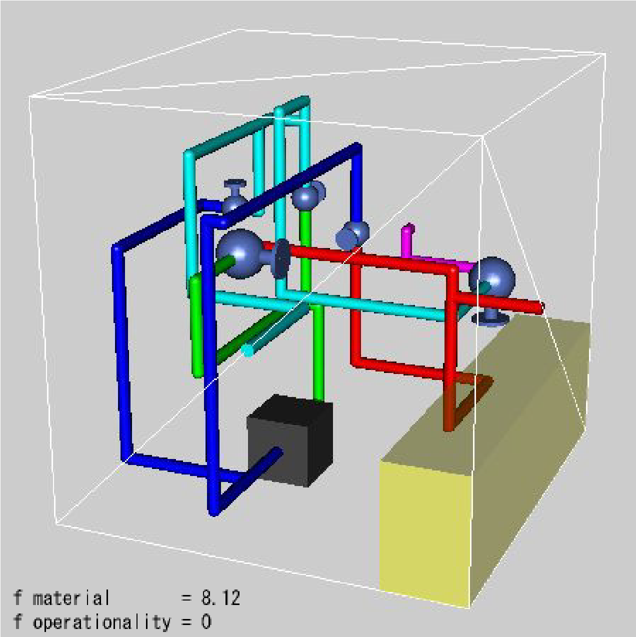


Figure 14: The result of the best design in the valve operationality

5. DISCUSSION

In Figure 10, both the one-point crossover and the random search found good solutions with regard to the valve operationality. In the progress of the search, the materials cost is also improved as shown in Figure 11 and 12. But comparing the one-point crossover with the random search in the same individual, the former found the better solutions with regard to the materials cost. In addition, the one-point crossover generated more various and better solutions than the ones of random search. The reason of this advantage is considered that the one-point crossover generates good candidates from the promising candidates that are referred as parents. Note that the search is converged sufficiently at 400th generations because the improvement of the 450th generations is little. From Figure 13 and 14, we can see that the multi-objective optimization algorithms generate multiple solutions that are superior in the sense of the valve operationality or the materials cost simultaneously.

6. CONCLUSIONS

In this paper, we proposed a criterion for valve operationality in pipe arrangement, and we formulated the pipe arrangement problems as a multi-objective optimization problem. The criterion for valve operationality is given by enumerating accessible valves that is neighbouring accessible space. We also proposed a multi-objective genetic algorithm for the pipe arrangement problem. We encoded only the parameters of the locations and directions of the valves as the genes for the multi-objective genetic algorithms. In this genetic algorithm, the children of the solution inherits only the parent's information of valves , and arrangement of pipes are given by the other heuristics at every new candidate. However, it is inefficient in the search of pipe arrangements. In real pipe-arrangement design, pipe arrangement is determined first from the pipe diagram , and thereafter, valves are set in the arranged pipes. It is largely different from our approaches. Finding a coding method that inherits the arrangement of pipes is the future work.

8. REFERENCES

1. S. Ikehira, H. Kimura, H. Kajiwara : 'Automatic Design for Pipe Arrangement using Multi-objective Genetic Algorithms', 12th International Conference on Computer Applications in Shipbuilding (ICCAS), Vol.2, pp.97-110, 2005.

2. S. Ikehira, Hajime Kimura : 'Multi-objective Genetic Algorithms for Pipe Arrangement Design', Proceedings of the 2006 Genetic and Evolutionary Computation Conference (GECCO 2006), pp.1869-1870, 2006.

3. Satoshi Ikehira, Hajime Kimura and Hiroyuki Kajiwara: 'Automatic Pipe Arrangement Design considering Operationality of Valves', Conference Proceedings of the Japan Society of Naval Architectures and Ocean Engineers, Vol.4, pp.335--338 (2007) in Japanese.

4. K. Deb, S. Agrawal, A. Pratab and T. Meyarivan : 'A fast elitist non-dominated sorting genetic algorithms for multiobjective optimization: NSGA II', KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

5. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: 'Improving the Performance of the Strength Pareto Evolutionary Algorithm', In Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.

9. AUTHORS BIOGRAPHY

Hajime Kimura holds the current position of Associate Professor at Department of Marine Systems Engineering, Kyushu University. His previous experience includes optimization in shipbuilding, etc.

Satoshi Ikehira holds the current position of a candidate of Ph.D. degrees in Department of Maritime Engineering, Graduate School of Engineering, Kyushu University.