An Automatic Piping Algorithm Including Elbows and Bends

Dept. of Marine System Engineering, Kyushu University
Hajime Kimura
Dept. of Marine System Engineering, Kyushu University
Yuto Ando
Outline

1. Background and Purpose
   - Previous Research

2. Routing Algorithm Including Bends
   - Approach
   - Dijkstra’s Method
   - Outline of “Bends”
   - Experiments

3. Conclusion and Challenges
Pipe Arrangement requires …

◆ keeping to regulations
ex. 
- not to set fuel oil pipelines near to electrical lines

http://www.cadpipe.com/industrial3D.html
Pipe Arrangement requires …

- keeping to regulations
- meeting demands
  ex.
  - to shorten the total length
  - to set along with the ship hull

http://www.cadpipe.com/industrial3D.html
Background

Pipe Arrangement requires …

- keeping to regulations
- meeting demands
- originality by each ship

http://www.cadpipe.com/industrial3D.html
Pipe Arrangement requires ...

- keeping to regulations
- meeting demands
- originality by each ship

Experiences of skilled designers

http://www.cadpipe.com/industrial3D.html
Pipe Arrangement requires ...

- keeping to regulations
- meeting demands
- originality by each ship

Experiences of skilled designers

Automatic Design System

http://www.cadpipe.com/industrial3D.html
Purpose

Previous Researches

Problems are ...

✓ optimization of piping routes
✓ searching of piping routes
✓ constraints

Our Proposal

We try for ...

◆ solving these problems
◆ high performance system
◆ full automatic design
Previous Research

Approach by Ikehira and Kimura

- taking into account of “valve operationality”
- dividing the pipe arrangement problem into two challenges
  - equipments layout problem
  - routing problem

solved by “pattern match method” which joins primitive pipe parts

Disadvantages are…

- uncertainty of optimal routing
- impossibility of solving a complicated routing problems
Previous Research

Approach by Asmara and Nienhuis

- looking on the pipe arrangement problem as a routing problem in a directed and weighted graph
  - solved by “Dijkstra’s method”

Disadvantage is …
- the mesh size is restricted to be larger than a pipe’s diameter
  - especially in large pipe’s diameter

Strong Constraint!
Outline

1. Background and Purpose
   - Previous Research

2. Routing Algorithm Including Bends
   - Approach
   - Dijkstra’s Method
   - Outline of “Bends”
   - Experiments

3. Conclusion and Challenges
Approach

Problems of Previous Researches

- uncertainty of the route with minimum costs
- demanding of the mesh size on the diameter

Our Approach

◆ using “Dijkstra’s method”
◆ improvement the routing algorithm
◆ using not only elbows but “bends”
Approach

Problems of Previous Researches

- uncertainty of the route with minimum costs
- demanding of the mesh size on the diameter

Our Approach

◆ using “Dijkstra’s method”
◆ improvement the routing algorithm
◆ using not only elbows but “bends”
Approach

Problems of Previous Researches

- uncertainty of the route with minimum costs
- demanding of the mesh size on the diameter

Our Approach

◆ using “Dijkstra’s method”
◆ improvement the routing algorithm
◆ using not only elbows but “bends”
Approach

Problems of Previous Researches

- uncertainty of the route with minimum costs
- demanding of the mesh size on the diameter

Our Approach

- using "Dijkstra's method"
- improvement of the routing algorithm
- using not only elbows but "bends"
**Approach**

- **Design Space**: box for pipe arrangement
- **Start and Goal**: coordinates and vectors
- **Target Pipeline**: pipes not including any branches
Design Space: box for pipe arrangement

Start and Goal: coordinates and vectors

Target Pipeline: pipes not including any branches
◆ Design Space : box for pipe arrangement
◆ Start and Goal : coordinates and vectors
◆ Target Pipeline : pipes not including any branches
Design Space
- box for pipe arrangement

Target Pipeline
- pipes not including any branches

Start and Goal
- coordinates and vectors

Subject of this research

Approach

Pipe Arrangement

Routing Problem

Equipments Layout Problem

Target Pipeline
- pipes not including any branches
◆ **Obstacle** : structures and equipments in ships
◆ **Aisle Space** : space for passages
◆ **Pipe-rack Area** : space for pipelines
Approach

◆ **Obstacle**: structures and equipments in ships

◆ **Aisle Space**: space for passages

◆ **Pipe-rack Area**: space for pipelines
◆ Obstacle: structures and equipments in ships
◆ Aisle Space: space for passages
◆ Pipe-rack Area: space for pipelines
Dijkstra’s Method

This method can …

- find the shortest path in a directed and weighted graph
- guarantee a path with minimum costs

Where is the path with minimum costs between C1 and C2?

The answer is …
Design Objectives

- to minimize the total length of pipes
- to minimize the number of elbows and bends
- to avoid passing aisles as possible
- to pass through pipe-rack areas as possible
Routing Algorithm

Pipe’s Diameter < Mesh Size
Searching of Straight Pipes

Pipe’s Diameter > Mesh Size
Searching of Straight Pipes

Step 1

Diameter: R

Temporary node

Current node
Searching of Straight Pipes

Interference Check!

Temporary node

Current node
Searching of Straight Pipes

Step 3

Recording Costs

Next node

Current node
Searching of Elbows

Pipe’s Diameter $>\$ Mesh Size
Searching of Elbows

Step 1: Searching of Elbows

- Node X
- \( d_1 \geq R/2 \)
- Diameter: \( R \)
- Current node
Searching of Elbows

Step 2

Node X

$d_1 \geq R/2$

Current node

Diameter: $R$

Temporary node

$d_2 \geq R/2$
Searching of Elbows

Step 3

Current node

Interference Check!

Temporary node
Searching of Elbows

Step 4

Current node

Recording Costs

Next node
Outline of Bends

“Bends” are …

◆ pipe parts to take the form of gentle S-shape
◆ connectors for gaps within the pipe’s diameter

from “NAMURA TECHNICAL REVIEW No.13, 2010”
Outline of Bends

Total length: 19
Outline of Bends

Total length: 22
Outline of Bends

Total length: $22 \rightarrow 18.7$
Searching of Bends

System user sets a value of $\alpha_1$, $\alpha_2$, $d$

ex. $\alpha_1 = 5$, $\alpha_2 = 0$, $d < R$

$$\beta = \arcsin \left( \frac{d - 2\alpha_1 R}{\sqrt{(2\alpha_1)^2 + (\alpha_2)^2}} \right) + \arcsin \left( \frac{2\alpha_1}{\sqrt{(2\alpha_1)^2 + (\alpha_2)^2}} \right)$$

$$L = (2\alpha_1 R \sin \beta) + (\alpha_2 R \cos \beta)$$

$L$ : the minimum vertical length of the bend
Searching of Bends

Pipe’s Diameter $\geq$ Mesh Size

Current node

Next node
Step 1

Searching of Bends

- Diameter: $R$
- Current node
- Temporary node
- $d_1 < R$
- $d_2 \geq L$
- $L$ : the minimum vertical length
Searching of Bends

Step 3

Recording Costs

Current node

Next node
Experiments

Purpose

◆ to verify the useful of the algorithm

Test Case Setting

◆ Design Area : X 16.75m, Y 3m, Z 3m
◆ Mesh Size : X 0.25m, Y 0.25m, Z 0.25m
◆ Start Point : ( 0.5m, 1.75m, 1.5m), x+
◆ Goal Point : ( 16.5m, 1.5m, 1.5m), x-
◆ 10 boxes as obstacles
Experiments

Cost of a Straight Pipe: 1 per 1m
Cost of a Elbow : \( d_1 + d_2 + 0.1 \)
Cost of a Bend: \( d_1 + d_2 + 0.3 \)
Experiments

Design Objectives for This Test

◆ to minimize the total length of pipes
◆ to minimize the number of elbows and bends
◆ to avoid passing aisles as possible
◆ to pass through pipe-rack areas as possible
Results \((R=0.2\text{m})\)

<table>
<thead>
<tr>
<th>Bends</th>
<th>O</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of Elbows</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>17.9</td>
<td>17.9</td>
</tr>
<tr>
<td>Time [s]</td>
<td>1285</td>
<td>1260</td>
</tr>
</tbody>
</table>
## Results (R=0.3m)

<table>
<thead>
<tr>
<th>Bends</th>
<th>○</th>
<th>×</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of Elbows</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>19.0</td>
<td>19.4</td>
</tr>
<tr>
<td>Time [s]</td>
<td>447</td>
<td>306</td>
</tr>
</tbody>
</table>
## Results (R=0.4m)

<table>
<thead>
<tr>
<th>Bends</th>
<th>O</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of Elbows</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>19.5</td>
<td>19.9</td>
</tr>
<tr>
<td>Time [s]</td>
<td>387</td>
<td>298</td>
</tr>
</tbody>
</table>
Results (R=0.5m)

<table>
<thead>
<tr>
<th></th>
<th>O</th>
<th>×</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of Elbows</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>19.6</td>
<td>19.9</td>
</tr>
<tr>
<td>Time[s]</td>
<td>373</td>
<td>288</td>
</tr>
</tbody>
</table>
## Results \((R=0.6\text{m})\)

<table>
<thead>
<tr>
<th></th>
<th>O</th>
<th>×</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Num. of Elbows</strong></td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td><strong>Num. of Bends</strong></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>Total Costs</strong></td>
<td>21.5</td>
<td>22.3</td>
</tr>
<tr>
<td><strong>Time [s]</strong></td>
<td>80</td>
<td>68</td>
</tr>
</tbody>
</table>
## Results \((R=0.7m)\)

<table>
<thead>
<tr>
<th></th>
<th>✓</th>
<th>×</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bends</td>
<td>O</td>
<td>×</td>
</tr>
<tr>
<td>Num. of Elbows</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>22.4</td>
<td>22.4</td>
</tr>
<tr>
<td>Time([s])</td>
<td>68</td>
<td>53</td>
</tr>
</tbody>
</table>
Results \( (R=0.8\text{m}) \)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bends</td>
<td>(\bigcirc)</td>
<td>(\times)</td>
</tr>
<tr>
<td>Num. of Elbows</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>22.4</td>
<td>22.4</td>
</tr>
<tr>
<td>Time[s]</td>
<td>63</td>
<td>50</td>
</tr>
</tbody>
</table>
## Results (R=0.9m)

<table>
<thead>
<tr>
<th></th>
<th>O</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of Elbows</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>Num. of Bends</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Total Costs</td>
<td>26.7</td>
<td>27.5</td>
</tr>
<tr>
<td>Time [s]</td>
<td>45</td>
<td>44</td>
</tr>
</tbody>
</table>
The algorithm practical design including bends.
The algorithm generated optimized solutions by each case.
Discussion

- It took more time to search a routing with smaller diameter.

Why?

The decrease of diameter causes an increase of the searching space.
Discussion

As a result in other test case…

Especially in very narrow space

◆ An obtained route interfered with itself!
Discussion

Node 1

Node 2

Node 3

Node 4

Node 5

Coordinates: same
Directions: different
Outline

1. Background and Purpose
   - Previous Research

2. Routing Algorithm Including Bends
   - Approach
   - Dijkstra’s Method
   - Outline of “Bends”
   - Experiments

3. Conclusion and Challenges
Advantages of the algorithm

- The mesh size is free.
- The algorithm generates practical designs with bends.
- The algorithm generates routes with minimum costs.
Conclusions

Advantages of the algorithm

◆ The mesh size is free.
◆ The algorithm generate practical designs with bends.
◆ The algorithm generate routes with minimum costs.
Advantages of the algorithm

- The mesh size is **free**.
- The algorithm generate practical designs with **bends**.
- The algorithm generate routes with **optimum costs**. **guaranteed!**
Future Works

We need to …

◆ improve the routing algorithm
◆ consider about pipe-rack and aisle areas
◆ make better the interference check algorithm
◆ create more maintainable system by the use of XML input / output
◆ associate the routing algorithm with the equipments layout algorithm

This system will be opened for free at

http://sysplan.nams.kyushu-u.ac.jp/gen/index.html