Job Scheduling Using Neural Network in Environment Inspection

Chenqi Cao
Abstract

Job Scheduling Using Neural Network in Environment Inspection

Chenqi Cao

Environment inspection is becoming more and more important now. Many qualified institutes provide professional inspecting services for general companies. One problem with environment inspection is that the number of equipment in the institute is limited compared to the number of projects.

A hybrid scheduler designed for the problem of environment inspection job scheduling. It uses tabu search techniques and embeds neural network in the scheduler. After training the neural network, the hybrid scheduler uses trained network to narrow the search range of the tabu search which makes the scheduler faster. The comparison between tabu search scheduler and hybrid scheduler indicates that the latter runs much faster.
Acknowledgments

My professors and friends gave me great help during the study and writing of the thesis report. Thank you all.
## Contents

1 Introduction ................................................................. 11  
1.1 Setting ................................................................. 11  
1.2 Purpose ................................................................. 11  
1.3 Scope ................................................................. 11  
1.4 Structure of the Report ............................................... 12  

2 Background ............................................................... 13  
2.1 Environment Inspection ............................................... 13  
2.2 Tabu Search ........................................................... 13  
2.3 Neural Network ....................................................... 14  
2.3.1 Neuron ............................................................. 14  
2.4 Multilayer Perceptron ............................................... 14  
2.4.1 Activation Function ............................................ 15  
2.4.2 Backpropagation ............................................... 15  

3 Environment Inspection Job Scheduling Problem .................. 19  
3.1 Purpose ................................................................. 19  
3.2 Definition .............................................................. 19  

4 Job Scheduling using Neural Network ............................... 21  
4.1 Approach ............................................................... 21  
4.2 Input ................................................................. 22  
4.3 Tabu Search Part ..................................................... 24  
4.3.1 Initial setup ...................................................... 24  
4.3.2 Initialize the best solution .................................... 24  
4.3.3 Check the stopping conditions ............................... 25  
4.3.4 Get the neighborhood $N_S$ by the neighbor function .... 25  
4.3.5 Find the local optimal solution $LS$ in $N_S$ .............. 26  
4.3.6 Return to stop conditions checking ......................... 27  
4.4 Neural Network Part ................................................ 28  
4.4.1 Structure ........................................................ 28  
4.4.2 Input Layer ...................................................... 28  
4.4.3 Hidden Layer .................................................... 30  
4.4.4 Output Layer .................................................... 30  
4.4.5 Activation Function ........................................... 32  
4.5 Hybrid Scheduler .................................................... 33  

5 Environment Inspection Management System ..................... 35  
5.1 System Structure ....................................................... 35  
5.2 Project Creation Module ............................................ 35  
5.3 Site Sampling Module ............................................... 35  
5.4 Test Experiment Module ............................................ 36  
5.5 Environment Evaluation Report Module ......................... 36  
5.6 Progress Monitoring Module ..................................... 37  
5.7 Common Function Module ......................................... 37  
5.8 System Management Module ..................................... 37  
5.9 Information Used by Scheduler ................................... 38
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Results</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>6.1 Experiment Setup</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>6.2 Comparison</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>6.3 Conclusion</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>Discussion and Conclusions</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>7.1 Model Improvement</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>7.2 Other Approaches</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>7.3 Conclusions</td>
<td>42</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>43</td>
</tr>
</tbody>
</table>
List of Tables

Table 4.1: Inventory information: equipment type and number of equipment in the inventory. 22
Table 4.2: Job basic information: job, earliest start time, latest end time, sampling duration. 22
Table 4.3: Job equipment request: equipment request of each job for each equipment type. 23
Table 4.4: Features for instance of 20 jobs and 6 equipment types. 29
Table 4.5: Normalized Features for instance of 20 jobs and 6 equipment types. 29
Table 4.6: Correctness of NN scheduler with different number of hidden layers and number of neurons. 30
Table 4.7: Best number of neurons in the hidden layer for different problem instances. 31
Table 4.8: Schedule indexes of jobs. 31
Table 4.9: Index mapping: scheduling index and sequence class. 31
Table 4.10: Output mapping: sequence class and desired output. 32
Table 6.1: Running time and number of successfully scheduled jobs for tabu scheduler and hybrid scheduler. 40
List of Figures

Figure 2.1: Neuron Model ........................................................................................................14
Figure 2.2: Multilayer Perceptron ..........................................................................................15
Figure 2.3: Step Function ......................................................................................................16
Figure 2.4: Logistic Function ................................................................................................16
Figure 2.5: Hyperbolic tangent function \( f(x) = \tanh(x) \) ......................................................16
Figure 2.6: Update weights for output layer. \( j \) is neuron of the output layer. \( i \) is neuron of 'left' layer in the MLP structure. .................................................. 17
Figure 2.7: Update weights for hidden layer. \( j \) is neuron of the hidden layer. \( k \) is neuron of the layer updated in the previous step. \( i \) is neuron of 'left' layer in the MLP structure. ........ 18
Figure 4.1: Hyperbolic tangent function \( f(x) = 1.7159 \tanh(\frac{2}{3}x) \) ..................................32
Figure 4.2: Work flow of hybrid scheduler ............................................................................33
Figure 5.1: Structure of Environment Inspection Management System ................................36
1. Introduction

1.1 Setting

The work environment of a company is very influential to its employees. A bad environment can cause occupational diseases. Nationally certified environment inspection institutes provide testing services to general companies. (ex. Shanghai Environmental Monitoring Center\(^1\)) According to Law of the People’s Republic of China on Prevention and Control of Occupational Diseases, the company needs an environmental report to carry out production work.

In the daily work of the environment inspection institute, various testing equipment is used. However, due to procurement and maintenance cost, the number of equipment is relatively small compared to the number of equipment required by all projects. And the required equipment is different in different projects. Meanwhile, the time periods of the site sampling should also be negotiated with the inspected companies. This is because some companies need to submit the final reports before their deadlines and they also need time to prepare for the inspection, for example, to arrange employees to help on-site sampling.

All these requirements of the real environment inspection work makes the job scheduling complicated, which is very difficult to do by hand.

In addition, the basic information such as the equipment inventory, the equipment required by projects are essential data for scheduling. Therefore, an information management system should be established to collect and manage all these data.

1.2 Purpose

The purpose of this thesis project is to create an efficient way to schedule jobs of the environment inspection institute. A hybrid scheduler is designed. It uses tabu search (TS) \([1, 2]\) to search for the best scheduling solution, and neural network \([3]\) is used to boost the scheduling process by providing advice on job selection in the event of a conflict.

This thesis project uses the scheduler to arrange the site sampling jobs automatically for the environment inspection institute.

Testing of the scheduler uses random inventory setting and random projects as input data. Results and performance are evaluated.

1.3 Scope

The thesis project focuses on design, training of the neural network (NN) and combines it with a tabu search. Finally a hybrid scheduler is implemented to complete job scheduling for the environment inspection institutes.

\(^1\)Shanghai Environmental Monitoring Center English website: http://www.semco.cn/home/english.aspx
The information needed for scheduling is offered by the Environment Inspection Management System. The brief introduction of the system is included while design details of the management system is not included.

To train the NN, a TS based scheduler is implemented. It generates the example dataset for training, validation and testing. After removing non-typical data, dealing with imbalanced data and other pretreatment of the given dataset, data is used as input to the NN and the training process is started. The NN tries to learn the inner relationship between the features of a job and its scheduling class.

After training, the NN is used as a classifier which can offer the scheduling priority of a job according its class.

Finally, a hybrid scheduler which uses tabu search to search the possible solutions and utilizes NN to reduce the searching scope is implemented.

1.4 Structure of the Report

This report consists of seven sections.

Section 1 introduces the basic setting of this thesis project, the purpose of the project and the scope of work included in the report.

Section 2 offers the background of the environment inspection and related techniques (tabu search, neural network and multilayer perceptron) used in the thesis project to solve the problem.

Section 3 provides the definition of the job scheduling problem in environment inspection. All variables used in following algorithm are defined in this section.

Section 4 discusses the approach of solving the problem in details. It includes the tabu search part, the neural network part and the final hybrid scheduler.

Section 5 gives brief description of the management system implemented to control and manage various information created and used by the environment inspection institutes. The system works as a data source for the schedule algorithm.

Section 6 shows the comparison between the tabu scheduler and the hybrid scheduler. And a conclusion of the scheduler is also included.

Section 7 gives several possible improvements and introduces other techniques which can also be used to solve the problem.
2. Background

2.1 Environment Inspection

The quality of the work environment is a big problem for today’s company. Harsh environment can cause occupational diseases due to continuously exposition to chemical hazardous substances ($CO_2$, $CO$, $NO$, etc.) or physical agents (noise, radiation, heat, etc.). In order to protect the workers and standardize the daily environmental management of the company, the law on the prevention and control of occupational diseases was promulgated. The inspection work is authorized for qualified environment inspection institutes. This is because most of the hazardous substances are not visible and if there is no professional equipment and techniques, it is impossible to carry out quantitative measurements.

2.2 Tabu Search

Tabu search (TS) [1, 2] was crated by Fred W. Glover, it is an optimized local search (LS) [4] algorithm.

The key idea of a LS is to design a neighbor function ($N$), which maps every solution $S$ to a subset of given problems’ search space ($N(S)$), called neighborhood. Every solution in the neighborhood is called neighbor of $S$. The local search algorithm starts with an initial solution and tries to find a better solution in its neighborhood. LS then repeats this search process by setting local best solution and searching in the new neighborhood. LS stops searching when certain stop criterion is satisfied.

The problem of LS is that the searching often stagnates at a local minimum or a ‘plateau’ with the same cost. To solve the problem of LS, TS loose the rule by accepting a ‘bad’ move which causes deterioration of solution when there is no improving move.

In TS, tabu restrictions are used to reject moves that will return to the previously visited state. And TS introduces three different strategies of tabu restrictions[1]:

- **Short term:**
  A tabu list is maintained during the search. Solutions found recently are stored. If a new solution is in the list, then it is considered as a tabu and not taken into account this round.

- **Intermediate term:**
  Intermediate term TS records good solutions during the search. It focuses on finding the best solution based on them. It compares features between those good solutions and tries to find common features. Then it judges the quality of solution according to the number of good features it exhibits.

- **Long term:**
  Long term TS tries to break the local minimum by finding a new start point which can lead search to the unexplored region and thus has the opportunity to find a better solutions. It achieves this goal by avoiding common features which are exhibited in the previous solutions.
2.3 Neural Network

Neural Network (NN) or Artificial Neural Network (ANN) [3] is one of the Machine Learning (ML) [5] techniques which simulates information treatment process of the real biological neural network [6]. NN is a computing model which consists of lots of neurons and connections between them. Each neuron stands for a function called activation function. And the connection stands for weight of information transferred. NN depends on the weights to memorized patterns or classifications. So different structures of the network, different weights and different activation functions results in different NNs. NN is usually used to solve the classification and function approximation problem.

![Figure 2.1. Neuron Model](image)

2.3.1 Neuron

Neuron is the basic part of the NN. If the input data of one neuron exceed its threshold, then it is activated. Meanwhile, neuron is connected to other neurons. When one neuron is activated, it will stimulate connected neurons by setting its output value and therefore probably activates other neurons.

The model of neuron is designed by McCulloch and Pitts [7] as shown in Figure 2.1. The neuron receives inputs ($x_i$) from $n$ other neurons. These inputs are transferred to the neuron through connections with weights ($w_i$). After the neuron has received all inputs, the weighted summary are compared to the threshold ($\sum_{i=1}^{n} x_i w_i - \theta$) where $\theta$ is the threshold. Finally, the neuron utilizes a activation function (discussed in Section 2.4.1) to setup output ($y = f(\sum_{i=1}^{n} x_i w_i - \theta)$).

2.4 Multilayer Perceptron

Multilayer Perceptron (MLP) [8] is a feedforward NN which contains an input layer, some hidden layers and an output layer. It accepts group of input data and maps them to the output data. Feedforward means that the connections among nodes don’t form a cycle like recurrent neural networks [9].

The simplest perceptron consists of one input layer and one output layer. The input layer accepts inputs from outer environment and send them to the output layer. The weights can be simply learned by Equations (2.1) and (2.2).

\[
\begin{align*}
    w_i &= w_i + \Delta w_i \\
    \Delta w_i &= \eta(d - y)x_i
\end{align*}
\]
where $\eta$ is called learning rate which controls the learning speed from an error. So the neuron can update its weights according to the difference between desired output and its current one. If they are the same, weight is kept unchanged.

However, the learning ability of perceptron with only one input layer and one output layer is limited. Since the input layer just takes the inputs and only output layer has 'functional neuron' inside, it can be proved that this kind of perceptron can only solve linearly separable problem [10]. To solve problem which is not linearly separable, multilayer perceptron (MLP) is introduced. As shown in Figure 2.2. MLP have additional layers called hidden layers which consists of functional neurons with activation functions. In MLP, each neuron of one layer are connected to all neurons of next layer. Neurons in the same layer are not connected to each other and there is no cross-layer connection.

MLP has a much more powerful learning ability than the simple perceptron. Therefore the Equations (2.1) and (2.2) cannot fulfill the more complicated learning task. MLP uses backpropagation (BP) as its training methods [11] which utilizes the error between desired and actual output to update weights backwards from output layer to input layer. The backpropagation is explained in Section 2.4.2.

2.4.1 Activation Function

The simple activation function is the step function shown in the Figure 2.3. It maps inputs to 0 or 1. 1 stands for activated and 0 stands for unactivated. However, the step function is not smooth and not continuous. So it cannot offer a smooth transition while the input changes. The most common activated functions are sigmoid functions like hyperbolic tangent function ($f(x) = \tanh(x)$) and logistic function ($f(x) = \frac{1}{1+e^{-x}}$) as shown in Figures 2.4 and 2.5.

2.4.2 Backpropagation

The backpropagation (BP) is the most common learning algorithm for MLP. In BP, the error function (Equation (2.3)) is used to computer the error between desired output and actually output.
Figure 2.3. Step Function

Figure 2.4. Logistic Function

Figure 2.5. Hyperbolic tangent function \( f(x) = \tanh(x) \)
\[ E = \frac{1}{2} (d - y)^2 \]  

(2.3)

where \(d\) stands for desired output and \(y\) stands for the actual output. Similar to Equations (2.1) and (2.2), weight is updated according to its contribution to the error:

\[ w_i = w_i + \Delta w_i \]  

(2.4)

\[ \Delta w_i = -\eta \frac{\partial E}{\partial w_i} \]  

(2.5)

Let \(x_0 = -1\) and \(w_0 = \theta\), then the input of the activation function can be expressed by:

\[ S = \sum_{i=0}^{n} w_i x_i \]  

(2.6)

and output \(y = f(S)\).

Then \(\Delta w_i\) can be expressed by \(\eta \delta x_i\) by the steps shown below:

\[
\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial S} \cdot \frac{\partial S}{\partial w_i}
\]

where

\[
\frac{\partial E}{\partial y} = -(d - y)
\]

\[
\frac{\partial y}{\partial S} = f'(S)
\]

\[
\frac{\partial S}{\partial w_i} = \frac{\partial \sum_{i=0}^{n} w_i x_i}{\partial w_i} = x_i
\]

therefore

\[
\frac{\partial E}{\partial w_i} = -(d - y) \cdot f'(S) \cdot x_i
\]

\[
\Delta w_i = -\eta \frac{\partial E}{\partial w_i} = \eta (d - y) \cdot f'(S) \cdot x_i
\]

Assume logistic function is used as the activation function, then \(\Delta w_i = \eta \delta x_i\) where \(\delta = f'(S)(d - y) = y(1 - y)(d - y)\). This is called delta rule and used by BP to update the weights.

In BP, weights is updated from output layer to hidden layer. As shown in the Figure 2.6, for the output layer, the delta rule is used directly:

\[
\Delta w_{ij} = \eta \delta_j x_i
\]

where

\[
\delta_j = f'(S_j)(d_j - y_j) = y_j(1 - y_j)(d_j - y_j)
\]

\[ w_{ij} \]  

Figure 2.6. Update weights for output layer. \(j\) is neuron of the output layer. \(i\) is neuron of 'left' layer in the MLP structure.
As shown in the Figure 2.7, for the hidden layer, weights are updated by:

\[ \Delta w_{ij} = \eta \delta_j x_i \]

where

\[ \delta_j = f'(S_j) \sum_k w_{jk} \delta_k = y_j(1 - y_j) \sum_k w_{jk} \delta_k \]

*Figure 2.7. Update weights for hidden layer. j is neuron of the hidden layer. k is neuron of the layer updated in the previous step. i is neuron of ‘left’ layer in the MLP structure.*

The error above is computed for one training instance. In batch learning, accumulated error backpropagation is used. All \( \Delta w \) for all instances are accumulated and then weights are updated. And it is called one epoch.

**Early Stopping**

One common problem during the training is overfitting. The network learn too much about the particular instances and response with a bad classification result when dealing with new instance. The early stopping technique can be utilized to solve this problem. The dataset is divided into training set and validation set. Training set is used to train the network to learn the weights. And the validation set is used to estimate the error. If the error of training set is decreasing while the error of validation set is increasing, the training is stopped.
3. Environment Inspection Job Scheduling Problem

3.1 Purpose

Environment inspection institutes usually carry out multiple environment inspection projects at the same time. One project cannot be interrupted by other projects. And site sampling requires various types of sampling equipment. However, the number of equipment owned by the institute is limited. If the inspection periods are not well scheduled, it may cause equipment conflict.

The purpose of the job scheduling in environment inspection is as following:

First, each project requires a variety of equipment. Second, there is a inspection period negotiated between environment inspection institute and the customer company for each project.

Under these two constraints, the scheduling algorithm should give a time plan for all projects’ site sampling jobs and avoid equipment conflicts as well.

3.2 Definition

In the environment inspection job scheduling problem, there is a set $T = \{t_1, \ldots, t_m\}$ of $m$ different types of sampling equipment. The inventory of each type $t_j$ equipment is $inv_j$. The remaining inventory of type $t_j$ equipment on day $x$ is $rn_{xj}$. So $rn_{xj} \leq inv_j$.

There is a set $J = \{j_1, \ldots, j_n\}$ of $n$ site sampling jobs. For each job $j_i$, there is a set $N_i = \{n_{i1}, n_{i2}, \ldots, n_{im}\}$ where $n_{ij}$ is the request number of type $t_j$ equipment. So if certain job requests type $t_j$ equipment, then $n_{ij} > 0$. Otherwise, $n_{ij} = 0$.

The plan sampling time period which is negotiated between the institute and customer company is $[b_i, e_i]$ where $b_i$ is the earliest begin day and $e_i$ is the latest end day for job $j_i$. The sampling duration which is the number of days to finish the site sampling is $l_i$. If there is a set $S = \{s_i\}$ where $s_i$ is the real start day of the site sampling, then time period constraint for job $j_i$ is:

$$s_i \geq b_i \land s_i + l_i - 1 \leq e_i \quad (3.1)$$

To focus on jobs scheduled on day $x$, there is a set $J_x = \{j_i \mid s_i \leq x \leq s_i + l_i - 1\}$. For any day $x$ and any equipment type $t_j$, the equipment requests of all jobs in $J_x$ should not greater than the total inventory of the institute, that is:

$$\forall x, \forall t_j \in T, \sum_{j_i \in J_x} n_{ij} \leq inv_j \quad (3.2)$$

The purpose of the scheduling algorithm is to make as many jobs as possible satisfy time period constraint $(3.1)$ under the premise of equipment constraint $(3.2)$. And set $S$ is the solution of the environment inspection job scheduling problem.

For a solution $S$, the satisfaction index $cn$ is the number of jobs that satisfy the time period constraint $(3.1)$:
\[ c_{nS} = \sum_{j_i \in J} f(j_i) \quad (3.3) \]

\[
 f(j_i) = \begin{cases} 
 1 & \text{if } s_i \geq b_i \land s_i + l_i - 1 \leq e_i \\
 0 & \text{otherwise} 
\end{cases} \quad (3.4)
\]

If set \( S \) satisfies the condition that \( c_{nS} \geq c_{nS'} \) where \( S' \) is any other solution of the certain problem instance, then \( S \) is optimal and recorded as \( BS \).
4. Job Scheduling using Neural Network

4.1 Approach

In this thesis project, neural network is used to solve the environment inspection job scheduling problem. To utilize the power of NN, there is several topics to be discussed:

1. The input and output of NN.
   NN will be used as a classifier. It takes features of job which are defined in the section 4.4 as input. And return its class used as scheduling priority in the scheduling, that is, job with higher priority will be scheduled first when there is a conflict.

2. Training data for NN.
   As a NN classifier, it needs training data to learn the relationship between features and classes. To provide the training data, a tabu search based scheduler is designed and launched first before NN. By run the TS scheduler for several times, enough training data can be collected.

3. The Way to use NN.
   Instead of scheduling the jobs directly, the trained NN will be utilized as classifier in the neighbor function to reduced the neighborhood size so that boost searching speed. So the final scheduler is a hybrid scheduler of TS and NN.

So the whole process is:

1. Use tabu search scheduler generating training data.
2. Train the neural network.
3. Use the hybrid scheduler to schedule all future instances.
4.2 Input

The data source of input is the inventory information, job basic information and the equipment request. Tables 4.1 to 4.3 show input data of an example instance with 20 jobs and 6 different equipment types.

<table>
<thead>
<tr>
<th>Type($t_j$)</th>
<th>Inventory($inv_j$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>8</td>
</tr>
<tr>
<td>$t_2$</td>
<td>6</td>
</tr>
<tr>
<td>$t_3$</td>
<td>6</td>
</tr>
<tr>
<td>$t_4$</td>
<td>9</td>
</tr>
<tr>
<td>$t_5$</td>
<td>8</td>
</tr>
<tr>
<td>$t_6$</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.1. Inventory information: equipment type and number of equipment in the inventory.

<table>
<thead>
<tr>
<th>Job($j_i$)</th>
<th>Start($b_i$)</th>
<th>End($e_i$)</th>
<th>Duration($l_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j_1$</td>
<td>19</td>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>$j_2$</td>
<td>35</td>
<td>38</td>
<td>3</td>
</tr>
<tr>
<td>$j_3$</td>
<td>13</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>$j_4$</td>
<td>32</td>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td>$j_5$</td>
<td>10</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>$j_6$</td>
<td>6</td>
<td>19</td>
<td>5</td>
</tr>
<tr>
<td>$j_7$</td>
<td>1</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>$j_8$</td>
<td>3</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td>$j_9$</td>
<td>14</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>$j_{10}$</td>
<td>14</td>
<td>34</td>
<td>1</td>
</tr>
<tr>
<td>$j_{11}$</td>
<td>35</td>
<td>38</td>
<td>4</td>
</tr>
<tr>
<td>$j_{12}$</td>
<td>16</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>$j_{13}$</td>
<td>22</td>
<td>34</td>
<td>4</td>
</tr>
<tr>
<td>$j_{14}$</td>
<td>8</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>$j_{15}$</td>
<td>31</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>$j_{16}$</td>
<td>35</td>
<td>38</td>
<td>4</td>
</tr>
<tr>
<td>$j_{17}$</td>
<td>9</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>$j_{18}$</td>
<td>35</td>
<td>39</td>
<td>5</td>
</tr>
<tr>
<td>$j_{19}$</td>
<td>34</td>
<td>39</td>
<td>4</td>
</tr>
<tr>
<td>$j_{20}$</td>
<td>13</td>
<td>29</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.2. Job basic information: job, earliest start time, latest end time, sampling duration.
<table>
<thead>
<tr>
<th>Job( (j_i) ) / Type( (t_i) )</th>
<th>( t_1 )</th>
<th>( t_2 )</th>
<th>( t_3 )</th>
<th>( t_4 )</th>
<th>( t_5 )</th>
<th>( t_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( j_1 )</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>( j_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>( j_3 )</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>( j_4 )</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>( j_5 )</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>( j_6 )</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( j_7 )</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( j_8 )</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>( j_9 )</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>( j_{10} )</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>9</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>( j_{11} )</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>( j_{12} )</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>( j_{13} )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>( j_{14} )</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>( j_{15} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( j_{16} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( j_{17} )</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( j_{18} )</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( j_{19} )</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>( j_{20} )</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 4.3.** Job equipment request: equipment request of each job for each equipment type.
4.3 Tabu Search Part

To generate the training data, a tabu search based scheduler is designed. It is an optimized local search algorithm. It starts from an initial solution and find the optimal solution by improve the current best solution. The main process of tabu search scheduler for environment inspection problem is shown below:

1. Initial setup.
2. Initialize the best solution.
3. Check the stopping conditions.
4. Get the neighborhood $NS$ by the neighbor function.
5. Find the local optimal solution $LS$ in $NS$.

4.3.1 Initial setup

In the real project, the $b_i$, $e_i$ and $s_i$ are all real date like May 1st. To make the schedule algorithm get rid of date computation during the searching process, the date is transformed to the day index.

First, a start date $sd$ is selected (which is typically the first day of next month). Second, the schedule duration $d$ is specified (which is typically the number of days in next two month). Then $b_i$, $e_i$ and $s_i$ is transformed to the difference between the start date and themselves plus 1 (so that the start day index is 1). Finally, notice that these three values may less than the $sd$ or greater than the $sd + d - 1$. So a new set $J^* = \{j^*_i \mid j^*_i \in J \land (1 \leq b_i \leq d \lor 1 \leq e_i \leq d)\}$ is defined where $j^*_i$ is a job to be included in this schedule. For simplicity, $b_i$, $e_i$ are trimmed according Equations (4.1) and (4.2) and $J^* = \{j^*_i\}$ is referred as $J = \{j_i\}$ in the following sections.

$$
\begin{align*}
b_i &= \begin{cases} 
1 & \text{if } b_i < 1 \\
b_i & \text{otherwise}
\end{cases} \\
e_i &= \begin{cases} 
d & \text{if } e_i > d \\
e_i & \text{otherwise}
\end{cases}
\end{align*}
$$

Meanwhile, not improved counter $(ni)$ and loop counter $(lc)$ is initialized to 0. Their functionalities is explained in the Section 4.3.3.

4.3.2 Initialize the best solution

The best solution is initialized with a random solution. This is achieved by creating a random schedule sequence and transform it to a solution.

Schedule sequence is a sequence: $j_{k_1}, j_{k_2}, \ldots, j_{k_q}, \ldots, j_{k_n}$ where all $j_{k_q} \in J$. So it stands for a possible arrangement of schedule for jobs. For example, if these is a job set $J = \{j_1, j_2, j_3, j_4, j_5\}$, its one possible schedule sequence is $j_3, j_2, j_1, j_4, j_5$.

Following steps are used to transform a schedule sequence($A$) to a solution($S$):

1. Fetch next unsettled job $j_{k_q}$ from $A$.
2. Set $s_{k_q} = b_{k_q}$
3. Check the remaining inventory:

$$\forall x \in [s_{kq}, s_{kq} + l_{kq} - 1], \forall t_j \in T \quad n_{kqj} \leq r_{n_{kqj}}$$  \hspace{1cm} (4.3)

where as described in the Section 3.2, \(n_{kqj}\) is the request number of type \(t_j\) equipment for job \(j_{kq}\), while \(r_{n_{kqj}}\) is the remaining inventory of type \(t_j\) equipment on day \(x\).

If constraint 4.3 is satisfied, goto step 4 directly. Otherwise, update \(s_{kq}\) according to 4.4 and go back to step 3.

$$s_{kq} = \begin{cases} 
    s_{kq} + 1 & \text{if } b_{kq} \leq s_{kq} < d \\
    1 & \text{if } s_{kq} \geq d \\
    -1 & \text{if } s_{kq} = b_{kq} - 1
\end{cases} \hspace{1cm} (4.4)$$

If \(s_{kq} = -1\) which indicates job \(j_i\) has tried every possible day slots and still cannot satisfy both equipment and time period constraints. So go to step 5.

4. Update the inventory.

For all \(t_j \in T\), update remaining inventory between date \([s_{kq}, s_{kq} + l_{kq} - 1]\) by \(r_{n_{xj}} = r_{n_{xj}} - n_{kqj}\).

5. \(S = S \cup \{s_{kq}\}\). If \(|S| < n\), then return step 1.

By creating a random \(A\), a random solution \(S\) can be created. And the best solution \(BS\) is initialized by \(S\).

4.3.3 Check the stopping conditions

If one of the following stopping conditions is satisfied, stop the searching process.

- \(cn_{BS} = n\).
  If all jobs satisfy both the equipment and time period constraints, then the \(BS\) is optimal for sure.
- \(ni\) reaches threshold.
  Not improved threshold is used to prevent the algorithm process from searching a local searching space without improvement. The update of \(ni\) is described in Section 4.3.5.
- \(lc\) reaches threshold.
  Loop threshold is used to controlled the overall search time. For a complicated problem instance, it may cost too much time to get the optimal solution. So it is a trade-off between quality and speed. By tuning the threshold, an acceptable result is achieved with a reasonable running time.

4.3.4 Get the neighborhood \(NS\) by the neighbor function

The neighbor function takes a solution as input and outputs its neighborhood which contains several neighbor solutions created by modify the input solution.

As mentioned in the Section 4.3.2, scheduling sequence is used to generate solutions. So actually the neighbor function takes the \(A\) of \(S\), then creating neighborhood of scheduling sequence (NA) and finally transform NA to \(NS\).

The neighbor function works as following:

1. Get the corresponding scheduling sequence \(A\).
2. Find all jobs violates the time periods. So they form a set $J' = \{ j_i \mid s_i < b_i \lor s_i + l_i - 1 > e_i \}$.

3. Take one job $j_i$ from $J'$, that is, $J' = J' \setminus j_i$ and creating new sequences through 2 methods shown below.

**Method 1:** Bring $j_i$ forward in $A$. For example, if there is a scheduling sequence $A$: $j_{k_1}, j_{k_2}, j_{k_3}, \ldots, j_{k_q}, \ldots, j_{k_n}$, where $k_q = i$, then $q - 1$ new neighbor scheduling sequences are created:

$$j_{k_q}, j_{k_1}, j_{k_2}, j_{k_3}, \ldots, j_{k_q-1}, j_{k_{q+1}}, \ldots, j_{k_n}$$

$$j_{k_1}, j_{k_q}, j_{k_2}, j_{k_3}, \ldots, j_{k_q-1}, j_{k_{q+1}}, \ldots, j_{k_n}$$

$$\ldots$$

$$j_{k_1}, j_{k_2}, j_{k_3}, \ldots, j_{k_q}, j_{k_{q-1}}, j_{k_{q+1}}, \ldots, j_{k_n}$$

**Method 2:** Rearrange jobs before $j_i$ in $A$. For example, if there is a scheduling sequence $A$: $j_{k_1}, j_{k_2}, j_{k_3}, \ldots, j_{k_q}, \ldots, j_{k_n}$, where $k_q = i$, then $(q - 1)! - 1$ new neighbor scheduling sequences are created:

$$j_{k_2}, j_{k_1}, j_{k_3}, \ldots, j_{k_q-1}, j_{k_q}, \ldots, j_{k_n}$$

$$j_{k_1}, j_{k_2}, j_{k_3}, \ldots, j_{k_q-1}, j_{k_q}, \ldots, j_{k_n}$$

$$\ldots$$

$$j_{k_{q-1}}, j_{k_{q-2}}, j_{k_q-3}, \ldots, j_{k_1}, j_{k_q}, \ldots, j_{k_n}$$

Method 1 intends to make job $j_i$ satisfied its time period constraint by scheduling it first. And method 2 intends to achieve this by rearrange the other jobs so that a better schedule which gives job $j_i$ more free time slots is created. However, the permutation in method 2 may produce too many instances, so an upper limit number of crated sequences is used.

4. Add all sequences created to $NA$. If $|J'| = 0$, then transform $NA$ to $NS$ by method described in the Section 4.3.2. Otherwise, return step 3.

4.3.5 Find the local optimal solution $LS$ in $NS$

1. Traverse the $NS$ set, ignore all solutions which are already in the tabu list and set the $LS$ to the solution with maximum $cn_S$.

2. Add $LS$ to tabu list.

3. Update best solution.

   If $cn_{LS} > cn_{BS}$, set $BS = LS$.

4. Update $ni$.

   $$ni = \begin{cases} 
   0 & \text{if } BS = LS \\
   ni + 1 & \text{Otherwise}
   \end{cases}$$ (4.5)

$LS$ doesn’t have to be better than the $BS$, tabu search allows this kind of ‘bad’ move.
4.3.6 Return to stop conditions checking

The LS is set as the next current solution and restart the searching process by checking stop conditions, get the new neighborhood and find the new LS until one of the stop conditions is satisfied.
4.4 Neural Network Part

In this thesis project, multilayer perceptron is utilized as a classifier. It is a feedforward neural network. It creates the mapping between inputs and given desired outputs by training.

It extracts features from job basic data and their equipment requests. These features are used as inputs for the MLP. Jobs are classified into several classes which stand for scheduling priorities when conflicts occur.

4.4.1 Structure

The MLP designed for environment inspection scheduling problem consists of one input layer, one hidden layer and one output layer. Each layer has several nodes and each node of one layer are connected to all nodes of next layer.

4.4.2 Input Layer

Five features are extracted from inventory, job basic information and equipment requests. As shown in the table 4.4, all features are between [0, 1]. After that, normalization is used to speed up the training and non-typical jobs are removed from the training set. Finally, features are used as the input data. The input layer has 5 nodes. Each node accepts one feature.

Feature Definition

- Earliest start time ($F_b$) and latest end time ($F_e$).
  $b_i$ and $e_i$ affects the schedule of jobs by setting the time range of job in which it can be adjusted. These two features are computed by:
  \[
  F_b = \frac{b_i}{d} \quad (4.6)
  \]
  \[
  F_e = \frac{e_i}{d} \quad (4.7)
  \]

- Sampling duration ($F_l$).
  If $l_i$ is relatively shorter, then it is more possible to find a feasible time slot to satisfy the constraints and vice versa. This feature is computed by:
  \[
  F_l = \frac{l_i}{e_i - b_i + 1} \quad (4.8)
  \]

- Maximum number of equipment request for a certain type ($F_n$).
  Since all the requests of one job should be satisfied at the same time, the maximum number determines whether it is easy to find a possible time slot. This feature is computed by:
  \[
  F_n = \text{Maximum}(\frac{n_{ij}}{inv_j}) \quad (4.9)
  \]

- Number of types requested ($F_t$).
  More different types requested by a job, more chances to conflict with other jobs. This feature is computed by:
  \[
  F_t = \frac{\sum_{t \in T} g(n_{ij})}{m} \quad (4.10)
  \]
  \[
  g(n_{ij}) = \begin{cases} 
  1 & \text{if } n_{ij} > 0 \\
  0 & \text{otherwise}
  \end{cases} \quad (4.11)
  \]
Normalization

The input date is shifted from $[0, 1]$ to $[-1, 1]$ so that the average of every data is 0. The zero mean helps to speed up the convergence [12]. This can be explained by a example whose input data only consists of positive numbers. A weight vector of a certain neuron can only be updated in one direction. And when the weight vector wants to change direction, it can only achieve the goal by zigzagging which is inefficient [12]. And for other cases which mean is not zero, it will also lead to a bias of a particular direction. The normalized features are shown in Table 4.5.

<table>
<thead>
<tr>
<th>Job($j_i$)</th>
<th>$F_b$</th>
<th>$F_c$</th>
<th>$F_f$</th>
<th>$F_n$</th>
<th>$F_l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j_1$</td>
<td>0.750</td>
<td>0.800</td>
<td>0.333</td>
<td>0.158</td>
<td>0.875</td>
</tr>
<tr>
<td>$j_2$</td>
<td>0.048</td>
<td>0.833</td>
<td>0.500</td>
<td>0.789</td>
<td>0.475</td>
</tr>
<tr>
<td>$j_3$</td>
<td>0.192</td>
<td>1.000</td>
<td>0.667</td>
<td>0.895</td>
<td>0.325</td>
</tr>
<tr>
<td>$j_4$</td>
<td>0.357</td>
<td>0.333</td>
<td>0.333</td>
<td>0.474</td>
<td>0.150</td>
</tr>
<tr>
<td>$j_5$</td>
<td>0.833</td>
<td>1.000</td>
<td>0.500</td>
<td>0.474</td>
<td>0.800</td>
</tr>
<tr>
<td>$j_6$</td>
<td>0.400</td>
<td>0.875</td>
<td>0.667</td>
<td>0.526</td>
<td>0.350</td>
</tr>
<tr>
<td>$j_7$</td>
<td>0.118</td>
<td>1.000</td>
<td>0.667</td>
<td>0.632</td>
<td>0.400</td>
</tr>
<tr>
<td>$j_8$</td>
<td>0.417</td>
<td>1.000</td>
<td>0.667</td>
<td>0.474</td>
<td>0.075</td>
</tr>
<tr>
<td>$j_9$</td>
<td>0.333</td>
<td>1.000</td>
<td>0.500</td>
<td>0.526</td>
<td>0.250</td>
</tr>
<tr>
<td>$j_{10}$</td>
<td>1.000</td>
<td>1.000</td>
<td>0.500</td>
<td>0.368</td>
<td>0.875</td>
</tr>
<tr>
<td>$j_{11}$</td>
<td>0.625</td>
<td>0.125</td>
<td>0.167</td>
<td>0.263</td>
<td>0.775</td>
</tr>
<tr>
<td>$j_{12}$</td>
<td>0.308</td>
<td>0.889</td>
<td>0.667</td>
<td>0.579</td>
<td>0.550</td>
</tr>
<tr>
<td>$j_{13}$</td>
<td>0.032</td>
<td>0.667</td>
<td>0.333</td>
<td>0.526</td>
<td>0.025</td>
</tr>
<tr>
<td>$j_{14}$</td>
<td>1.000</td>
<td>0.778</td>
<td>0.167</td>
<td>0.158</td>
<td>0.875</td>
</tr>
<tr>
<td>$j_{15}$</td>
<td>0.036</td>
<td>0.889</td>
<td>0.500</td>
<td>0.895</td>
<td>0.225</td>
</tr>
<tr>
<td>$j_{16}$</td>
<td>0.087</td>
<td>1.000</td>
<td>0.500</td>
<td>0.526</td>
<td>0.200</td>
</tr>
<tr>
<td>$j_{17}$</td>
<td>0.048</td>
<td>1.000</td>
<td>0.500</td>
<td>0.737</td>
<td>0.350</td>
</tr>
<tr>
<td>$j_{18}$</td>
<td>0.353</td>
<td>0.500</td>
<td>0.500</td>
<td>0.526</td>
<td>0.325</td>
</tr>
<tr>
<td>$j_{19}$</td>
<td>0.667</td>
<td>0.600</td>
<td>0.667</td>
<td>0.526</td>
<td>0.850</td>
</tr>
<tr>
<td>$j_{20}$</td>
<td>1.000</td>
<td>0.889</td>
<td>0.500</td>
<td>0.368</td>
<td>0.875</td>
</tr>
</tbody>
</table>

Table 4.5. Normalized Features for instance of 20 jobs and 6 equipment types.
Non-typical Jobs

Some jobs can fit to any sequence classes. They are jobs without typical features. To exclude such kind of jobs, the same instance is sent to the tabu search scheduler several times and variance of job’s class indexes is computed. If the variance is less than the predefined threshold then the job has no significant preference for particular sequence class and it is discarded. Otherwise the most common class is returned as the desired class.

Oversampling

As mentioned above, jobs are classified according to most common position in several runs of the same instance. So the distribution of the jobs for each class is not even and the training data is imbalanced. Imbalanced data leads to bias on the major class with more job instances and thus affects the correctness of classification in a bad way [13]. Oversampling is used in this thesis project to deal with the problem. The minority class is re-sampled so that all classes have the same quantity.

4.4.3 Hidden Layer

In this thesis, MLP contains one hidden layer and there are ten neurons in it. This choice is mainly base on the experiments.

Different number of layers and number of neurons are tested. Correctness is computed by number of correctly classified jobs by total number of jobs. And the data shown in Table 4.6 is the average of 10 results.

It turns out that more neurons will improve the correctness until the number reaches 10. And more layers do not improve the correctness because of over-fitting and they also slow down the training.

<table>
<thead>
<tr>
<th># Hidden Layers</th>
<th># neurons</th>
<th>Training Time(s)</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>580.51</td>
<td>0.64</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>772.47</td>
<td>0.69</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>893.05</td>
<td>0.72</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>956.78</td>
<td>0.76</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>1016.78</td>
<td>0.79</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>1219.37</td>
<td>0.78</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>1322.98</td>
<td>0.77</td>
</tr>
<tr>
<td>2</td>
<td>3,2</td>
<td>667.96</td>
<td>0.56</td>
</tr>
<tr>
<td>2</td>
<td>5,4</td>
<td>950.98</td>
<td>0.66</td>
</tr>
<tr>
<td>2</td>
<td>7,5</td>
<td>1298.31</td>
<td>0.70</td>
</tr>
<tr>
<td>2</td>
<td>10,7</td>
<td>1443.22</td>
<td>0.79</td>
</tr>
<tr>
<td>2</td>
<td>12,10</td>
<td>1944.75</td>
<td>0.72</td>
</tr>
<tr>
<td>2</td>
<td>15,12</td>
<td>2275.73</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 4.6. Correctness of NN scheduler with different number of hidden layers and number of neurons.

As shown in the Table 4.7, different types of problem instances doesn’t affect the best structure of NN scheduler so much.

4.4.4 Output Layer

As shown in the Table 4.8, the schedule indexes of jobs are used as the desired output for training. They are taken from the scheduling sequence which is created by the tabu search search scheduler.
<table>
<thead>
<tr>
<th># jobs</th>
<th># equipment</th>
<th>best # neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 4.7. Best number of neurons in the hidden layer for different problem instances.

<table>
<thead>
<tr>
<th>Job($j_i$)</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j_1$</td>
<td>9</td>
</tr>
<tr>
<td>$j_2$</td>
<td>3</td>
</tr>
<tr>
<td>$j_3$</td>
<td>11</td>
</tr>
<tr>
<td>$j_4$</td>
<td>4</td>
</tr>
<tr>
<td>$j_5$</td>
<td>2</td>
</tr>
<tr>
<td>$j_6$</td>
<td>18</td>
</tr>
<tr>
<td>$j_7$</td>
<td>5</td>
</tr>
<tr>
<td>$j_8$</td>
<td>17</td>
</tr>
<tr>
<td>$j_9$</td>
<td>6</td>
</tr>
<tr>
<td>$j_{10}$</td>
<td>1</td>
</tr>
<tr>
<td>$j_{11}$</td>
<td>7</td>
</tr>
<tr>
<td>$j_{12}$</td>
<td>12</td>
</tr>
<tr>
<td>$j_{13}$</td>
<td>19</td>
</tr>
<tr>
<td>$j_{14}$</td>
<td>20</td>
</tr>
<tr>
<td>$j_{15}$</td>
<td>15</td>
</tr>
<tr>
<td>$j_{16}$</td>
<td>10</td>
</tr>
<tr>
<td>$j_{17}$</td>
<td>13</td>
</tr>
<tr>
<td>$j_{18}$</td>
<td>16</td>
</tr>
<tr>
<td>$j_{19}$</td>
<td>14</td>
</tr>
<tr>
<td>$j_{20}$</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4.8. Schedule indexes of jobs.

However, try to learn the relationship between features and the indexes is difficult. The optimal solution for single instance of the problem is not unique. Therefore the scheduling index for the same job varies. If we try to map the index to ranges of indexes called sequence class, then more reasonable relationship can be found. In this thesis project, jobs are classified into four classes. For a instance which has 20 jobs, the mapping is shown in table 4.9.

<table>
<thead>
<tr>
<th>Scheduling Index</th>
<th>Sequence Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 5</td>
<td>1</td>
</tr>
<tr>
<td>6 - 10</td>
<td>2</td>
</tr>
<tr>
<td>11 - 15</td>
<td>3</td>
</tr>
<tr>
<td>16 - 20</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.9. Index mapping: scheduling index and sequence class.

The output layer has 4 nodes. Each node represents one of four classes. As shown in the Table 4.10, corresponding node is set to 1 and other nodes are set to $-1$. 
<table>
<thead>
<tr>
<th>Class</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.10. Output mapping: sequence class and desired output.

Figure 4.1. Hyperbolic tangent function \( f(x) = 1.7159 \tanh(\frac{2}{3}x) \)

4.4.5 Activation Function

As shown in the Figure 4.1, a modified hyperbolic tangent function \((f(x) = 1.7159 \tanh(\frac{2}{3}x))\) is chosen as the activation function. According to LeCun[12], it keeps target values (-1 or 1) away from its asymptotes so that it can prevent weights from becoming too large.
4.5 Hybrid Scheduler

As shown in the Figure 4.2, the whole work flow of the hybrid scheduler includes the training phase explained in the previous Sections 4.3 and 4.4. Tabu searcher scheduler takes instances and returns scheduled instances. By extracting features from basic inventory and job information, input data is created. By transforming scheduling index to sequence class, the desired output is created. These data is then used to train the neural network.

After training, the NN scheduler can classify jobs to sequence classes by their features. The class reflects job’s most potential range of positions in the scheduling sequence. However, only the sequence class is not enough to do the scheduling. While the tabu search scheduler can solve the problem but suffers from the slow searching speed caused by the large neighborhood. So the trained NN scheduler is utilized in the tabu search scheduler which results in a hybrid scheduler.

The NN scheduler is used in the step 4 described in the Section 4.3.4. The sequence class is used as the scheduling priority which indicates which job should be scheduled first when there is a conflict.

To deal with the non-typical jobs, if the output of NN has no significant preference for particular sequence class, then its class is set to a value greater than any class number. This value setting is mainly to simplified implement of following strategies.

In the method 1: Current job only moves before another job satisfied both constraints:

1. Its position is before current job in the scheduling sequence.
2. Its class number ≥ current job’s class number.

So the current job is moved to the jobs which belong to the same position range or ranges after it. And all non-typical jobs are also included.

In the method 2: Only jobs satisfied both constraints are rearranged:

1. Its position is before current job in the scheduling sequence.

2. Its class number < current job’s class number.

So only jobs belong to the position ranges before current job are considered. And the non-typical jobs are excluded. It is mainly used to reduce the large amount of neighbors created by permutation.

Through this approach, the hybrid scheduler only considers jobs that has high possibility to fulfill the intention of these two methods explained in the Section 4.3.4.
5. Environment Inspection Management System

5.1 System Structure

As shown in the Figure 5.1, the environment inspection project management system consists of project creation module, site sampling module, test experiment module, environment evaluation report module, progress monitoring module, common function module and system management module.

The management of the project process is provided by the project creation module, site sampling module, test experiment module and environment evaluation report module.

Progress monitoring module provides monitoring of the progress of the project and can view the real-time status of the project.

Common function module mainly provides users with the commonly used functions in the system, such as project flow control, test reports, personal centers, and document management.

System management module manages information shared by all other modules, such as permissions, customers, departments, employees and equipment information.

For example, in a real environment inspection project, the employee creates the project and follows the process of project management. Different employees complete their duties by working on their own tasks within the scope of the corresponding permissions. The sampling jobs are arranged by scheduling algorithm. And when the project is finalized, all project documents are recorded in the system for archiving. Employees with the appropriate permissions can view and manage departments, employees, customers, permissions and equipment.

5.2 Project Creation Module

The project creation module covers the work flow of project which involves all departments. It is responsible for recoding test items and the initial plan of the environment inspection tasks into the system. The following functions are included: project creation, project review, project planning and customer data upload.

- Project creation: Enter basic information about the project, contract-related information and project type.
- Project review: Each department reviews the content of the project and its feasibility.
- Project planning: Departments plan their own tasks to be completed in the project.
- Customer data upload: Archive the project related customer data.

5.3 Site Sampling Module

The site sampling module mainly deals with the work of the evaluation department, the quality control department and the inspection department. It is responsible for evaluating plan and the site sampling. It includes the following functions: evaluation program review, test notification, sample plan and job schedule timetable.
5.4 Test Experiment Module

The test module mainly involves the work of the testing department and the laboratory. It is responsible for the relevant tasks of the test report, including the following functions: send & receive samples and test report.

- **Record of send & receive samples**: The samples’ information transferred between the inspection department and the laboratory and their storage status are recorded.
- **Testing report**: The final results of the tests are recorded in the test report for the final environment assessment.

5.5 Environment Evaluation Report Module

The environment evaluation report module mainly deals with the work of the evaluation department and the quality control department. It is responsible for the development and review of
the report, including the following functions: evaluation report calculation and evaluation report review.

- Evaluation report calculation: For different types of hazardous substances, it computes the value according to the corresponding national specifications and compared with the national limits.
- Evaluation report review: Upload the evaluation plan, and review the its contents.

5.6 Progress Monitoring Module

The progress monitoring module is responsible for monitoring the current progress of each project.

- Progress monitoring list: List all projects in a list and display the current progress of each project in real time through the progress bar.
- Progress details: Displays detailed progress on the project, providing quick viewing of documents and viewing execution status of the current project step.

5.7 Common Function Module

The common function module provides support for other modules, including project flow control, reports, personal centers, and document management.

- Project flow control: Project flow control works for other modules. It automatically updates the progress of the project. The current tasks for particular employee is collected and shown in the current tasks panel. The employee updates the progress of the project by completing his own job.
- Reports: The report generation function for the various environment inspection files is required in the whole project process. It generates word, excel or pdf documents for each project through report templates.
- Personal center: Personal center is mainly used to modify the employee’s e-mail address, telephone information and other basic information. It also provides the function of password setting.
- Document management: Document management is mainly responsible for adding project-related documents. Files are managed by its project. In the project process, these documents is used as project-related information.

5.8 System Management Module

The system management module is responsible for managing the public information that needs to be accessed in the project process, including permissions, customers, employees, departments, and equipment.

- Permissions management: Permissions management is responsible for the management of each employee’s permissions. They directly affect the scope of the operation particular employee can view and operate. The default permission set of employee is based on the his department. Employee’s permission set can be a subset of the default one.
- Customer management: Customer management is responsible for recording the basic information of the project customers. It avoids duplication of customer information in the project process which helps to avoid data redundancy and inconsistent errors.

- Staff management: Staff management is mainly responsible for recording the basic information of employees. Employee information is the identity of system user. It is the most commonly used information in the project process.

- Department management: Department management is mainly responsible for recording the basic information of the department. Department is mainly used to manage a group of employees. Employee is assigned default permission set according to his department.

- Equipment management: Equipment management is responsible for recording the basic information of the equipment, including the equipment number, model, name, manufacturer, etc. This information is used by the job scheduling algorithm.

5.9 Information Used by Scheduler

The inventory information is managed by the equipment management of system management module. The project basic information is managed by the project creation module. Equipment requests and time period constraints are managed by site sampling module.
6. Results

6.1 Experiment Setup

Experiments were run under Linux Ubuntu 16.04 LTS (64 bit) on an Intel Core i7 6700K of 4.00 GHz with an 1 MB L2 cache and a 32 GB RAM.

2000 instances were used to do the training. Each instance was run on tabu search scheduler for several times so that non-typical jobs could be removed. To utilize the final hybrid scheduler to solve particular type of problem, every problem instances had the same number of equipment types and number of jobs.

Finally, the hybrid scheduler was used to solve new instances and results were compared with the tabu search scheduler.
6.2 Comparison

The tabu search scheduler and the hybrid scheduler are tested with the same instances of different type of problem. As shown in Table 6.1, although hybrid scheduler has no advantage over tabu search scheduler for simple problems with small number of jobs and equipment types. The running time of hybrid scheduler is much shorter than that of tabu search scheduler in most cases. And the number of jobs successfully scheduled is same at most times.

<table>
<thead>
<tr>
<th># jobs</th>
<th># equipment</th>
<th>Running time (s)</th>
<th># Scheduled Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Tabu</td>
<td>Hybrid</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>0.766</td>
<td>0.031</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>56.102</td>
<td>0.033</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>235.714</td>
<td>5.927</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>0.184</td>
<td>0.001</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>76.272</td>
<td>0.160</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>295.295</td>
<td>0.279</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>735.733</td>
<td>8.581</td>
</tr>
</tbody>
</table>

Table 6.1. Running time and number of successfully scheduled jobs for tabu scheduler and hybrid scheduler.

6.3 Conclusion

The hybrid scheduler is trained by certain type of problem first. Then it can be used to schedule instances of the similar problem type. It costs most time at one-time training phase. It needs very short running time in the future scheduling with a good performance.
7. Discussion and Conclusions

In this project, a hybrid scheduler is designed and used to schedule the jobs of the environment inspection institute. The management system collects and provides basic data such as inventory and job information to the scheduler. With the help of the system and the scheduler, the employees can organize and schedule daily work efficiently. However, this thesis report does not contain several topics yet. The following sections discuss possible improvements and future work.

7.1 Model Improvement

The current model of the problem only considers the equipment request of the job. However, in the real environment inspection institute, there is also competition for human resources. It requires the model take two independent resources into consideration which makes the problem more complicated.

The features extracted from the jobs results in about 78% correctness which is not significantly high. One possible improvement is to find new features which can indicates the arrangement jobs more precisely. Another possible improvement can be made by using or creating other kind of neural network.

7.2 Other Approaches

Other than the hybrid scheduler which mainly base on the methodology of tabu search and the classification ability of neural network. Other techniques can also be utilized to solve this problem:

- **Simulated Annealing [14]**
  It accepts worse result with a certain probability, so that it helps the scheduler to escape from the local minimum. The probability to accept suboptimal decreases over time which ensures the stability of the algorithm. This technique can be used in the NN to prevent the influence of local minimum.

- **Genetic Algorithm (GA) [15]**
  GA simulates the problem into a biological evolution process. It produces the next generation of solutions by implying the genetic operations (reproduction, recombination and mutation) on the current generation. It increases solutions with high fitness and discards low fitness ones. So the evolution of N generation is likely to result in a solution with very high fitness. This technique can search the best schedule by evolution from a random initial solution.

- **Particle Swarm Optimization (PSO) [16]**
  PSO is inspired by the movement of birds during feeding process. It searches for the best solution in complex space by cooperation and competition between individuals. Each individual is viewed as one particle of the swarm. The particle moves with its particular speed in the space while it is also affected by the best position found by itself, its neighbor and all particles. By implying this kind of strategy on each particle, the swarm moves to the best solution. This technique can be used to search the best schedule by initializing each particle with similar solution.
7.3 Conclusions

Job scheduling in the environment inspection needs to take both equipment and time period constraints into account. Fulfilling two constraints at the same time makes the job scheduling complicated. Tabu search and artificial neural network are both good approaches to solve the problem. Tabu search is a straightforward method which tests solutions and tries to find better ones using neighbor function. However, it suffers from the heavy time cost of testing solutions one by one in the neighborhood. On the other hand, neural network returns the answer quickly after training. But it needs enough training data for every problem instance. If number of jobs or equipments are changed, new training data will be needed. The hybrid searcher combines those two methods together and makes them complementary to each other. Tabu search creates the training data for the neural network, then the network narrows the searching scope for the tabu search. The time cost of search is transferred from the searching phase to the one-time preparing phase, which makes the hybrid searcher performs better during the searching.

In addition, all data needed in the searching is already collected and maintained by the other modules of the system, which eases the preparation to start the search and ensures the correctness of the result.

Finally, the system enforces the equipment constraint and fulfills the time period constraint as far as possible. It is able to generate correct and efficient results quickly. And with the help from all other modules of system, user can manage jobs in environment inspection institute well.
References