# Problem Description File Format

Jayanth Krishna Mogali

December 14, 2019

## 1 File Specification At a Glance

At a high level, a typical problem description file has the structure shown below. <string> and <size_t> (unsigned int) simply refers to the C++ type of the corresponding entry.

NAME : <string>
VERTICES : <size_t>
ROBOTS : <size_t>
NODE_TYPE : V IV
EDGE_DATA_FORMAT : ADJ_LIST
NODE_DESCRIPTION_SECTION:
<NODE_TYPE> <NODE ID> <X_COORD> <Y_COORD> <TIME> (If NODE_TYPE is V)
<NODE_TYPE> <NODE ID> <TIME> (If NODE_TYPE is IV)
EDGE_DESCRIPTION_SECTION:
ROBOT: <ROBOT INDEX>
<NODE ID_1> <INTERMEDIATE NODE_ID> <NODE ID_2>
COLLISION_DESCRIPTION_SECTION:
<NODE ID_1> <ROBOT INDEX_1> <NODE ID_2> <ROBOT INDEX_2>
ENABLER_DESCRIPTION_SECTION:
<NODE ID> <NODE ID_1> <NODE ID_2> ... <NODE ID_K>
START_END_DESCRIPTION_SECTION:
<ROBOT INDEX> <NODE ID_1> <NODE ID_2>

## 2 Sections In Detail

As in the paper, let the problem consist of $\mathbf{M}$ robots and $\mathbf{N}$ task locations to be serviced. Basically the input file is going to provide a description of the routing graph $\mathbf{G}(\mathbf{V}, \mathbf{A})$ (refer Page 2 in the journal paper).

### 2.1 Header Information

NAME : <string>
VERTICES : <size_t>
ROBOTS : <size_t>
NODE_TYPE : V IV
EDGE_DATA_FORMAT : ADJ_LIST

1. NAME : <string>
   Name of the file.

2. VERTICES : (2*$\mathbf{M}$) + $\mathbf{N}$
   You should expect to see $|\mathbf{O}|+|\mathbf{D}|+|\mathbf{H}| = (2*\mathbf{M})+\mathbf{N}$ number of start-terminal locations (denoted by $\mathbf{O}, \mathbf{D}$ in the paper) and task locations (denoted by $\mathbf{H}$ in the paper) in this file, where $|\cdot|$ refers to cardinality.

3. ROBOTS : $\mathbf{M}$
   $\mathbf{M}$ specifies the number of robots in the file.

4. NODE_TYPE : V IV
   Tells you that all the nodes encountered in the input file are either classified as V or IV. More on V, IV in Section 2.2.

5. EDGE_DATA_FORMAT: ADJ_LIST

## 2.2 NODE_DESCRIPTION_SECTION:

This is perhaps the most important section. This section in the file begins with NODE_DESCRIPTION_SECTION. We need to introduce several concepts before this can make sense.

### Nodes

Nodes in the routing graph $\mathbf{G}(\mathbf{V}, \mathbf{A})$ are classified as either V or IV (Intermediate Nodes in the paper) in the problem description. Nodes corresponding to start, terminal locations and task locations (i.e. nodes in the set $\mathbf{O}, \mathbf{D}, \mathbf{H}$) are classified as V. Any intermediate node (i.e. nodes in the set $\mathbf{I}$) are classified as type IV in the problem description. Each node in $\mathbf{G}(\mathbf{V}, \mathbf{A})$ is uniquely labelled by a non-negative integer. Under the NODE_DESCRIPTION_SECTION, all the nodes of type V are first listed, following which all the nodes of type IV are listed.
The robots in the problem description file are also uniquely labelled by a non-negative integer. The robots are numbered by $0, 1, \ldots, \mathbf{M}-1$.
We now describe the data provided under the NODE_DESCRIPTION_SECTION. We also provide some additional information which can be helpful to verify if your parser has been implemented correctly.

### Node Type: V

- The general format in which a node of type V is described is given by the following:
  <V> <NODE ID> <X_COORD> <Y_COORD> <TIME>
  NODE ID is the label of the node.
  Recall no node of type V belongs to the set $\mathbf{I}$, so they do not represent the movement of the robot. As a result we can associate some physical coordinates to each node. (X_COORD , Y_COORD) are the physical coordinates associated with the node.
  TIME is the processing time of the node i.e. the time it takes for any robot to process that node.

  (X_COORD , Y_COORD) is not required for constructing $\mathbf{G}(\mathbf{V}, \mathbf{A})$, but we included it in case the reader wants to visualize the problem, or any solution technique he/she develops. In the description of the MRSBE heuristic described in the paper, X_COORD, Y_COORD are not used in any way.

- For $i \in \{0, 1, \ldots, \mathbf{M}-1\}$, a node labelled $2*i$ corresponds to the start location of robot $i$ and $2*i+1$ is the node corresponding to its terminal location. So labels in the set $\{0, 1, \ldots, (2\mathbf{M}-1)\}$ are reserved for nodes in $\mathbf{O} \cup \mathbf{D}$.

- The nodes with labels in the range $(2*\mathbf{M}) - (2*\mathbf{M}+\mathbf{N}-1)$ correspond to task locations i.e. nodes in $\mathbf{H}$.

- The following is an example from a 2 robot, 20 hole case. The entry below is to be interpreted as:
  V 1 0 0 0

"1" refers to the terminal node of robot 0, (0,0) is its physical location and 0 at the end is the processing time of the terminal node for robot 1.
Another entry for the same 2 robot 20 hole problem:
V 12 1.96667 4 4000
The above entry corresponds to a task location whose corresponding node is labelled 12, its physical location is (1.96667, 4), and any robot that can reach this task location would require 4000 time units to service it.

**Node Type: IV**
Nodes belonging to the set $\mathbf{I}$ (refer paper), are classified as type IV in the problem description file.
We briefly remind the reader what an intermediate node (nodes in $\mathbf{I}$) is. Given some robot $i$, let $o^i, d^i$ denote the start and terminal location of robot $i$. Let $j \in \mathbf{H_i} \cup o^i$, $k \in \mathbf{H_i} \cup d^i$ (where $\mathbf{H_i}$ refers to the task location nodes reachable by robot $i$). To model the motion from $j$ to $k$ by robot $i$, we introduce an intermediate node, and label it as $l$ say. To transition from $j$ to $k$, robot $i$ must first traverse from to $j$ to $l$, and then from $l$ to $k$. Then observe that: $l$ is unique to the robot and direction in which it traverses. Secondly, we point out that $\mathbf{H_i}$ is not explicitly defined in the problem description for each robot. Instead it should be inferred from the IVs enumerated in the problem description.
Convention and important info for node type IV entries:

- The general format in which a node of type IV is described is given by the following:
  <IV> <NODE ID> <TIME>
  As before, NODE ID is the label of the node and TIME corresponds to the processing time of the node (i.e travel time for the robot between the pair of task locations nodes associated with the intermediate node). Note that, we do not print the pair of task location nodes and robot associated with the intermediate node in the current section, it is printed in the EDGE_DESCRIPTION_SECTION (i.e. Section 2.3).

- Observe that the smallest label corresponding to an intermediate node is $2 * \mathbf{M} + \mathbf{N}$.

- In the 4 and 6 robot test cases, we mentioned in the paper that all task locations are reachable by all robots. So, the reader may verify that there should be $2 \times \binom{\mathbf{N}}{2} \times \mathbf{M}$ IV nodes corresponding to traversals between task locations by robots. The reader may verify that the total number of intermediate nodes listed in the file are: $2 \times \binom{\mathbf{N}}{2} \times \mathbf{M} + (2 \times \mathbf{MN})$, where the $2 \times \mathbf{MN}$ corresponds to the motion from the start location to each task location, and from each task location to the terminal location.

## 2.3 EDGE_DESCRIPTION_SECTION:

This section in the file begins with the string EDGE_DESCRIPTION_SECTION. In this section we will fill in the missing details for the intermediate nodes that was omitted earlier. The general format for this section is:

EDGE_DESCRIPTION_SECTION:
ROBOT: <ROBOT INDEX>
<NODE ID_1> <INTERMEDIATE NODE ID> <NODE ID_2>

NODE ID_1, NODE ID_2 are labels corresponding to nodes of type V. I will explain entries in this section with an example. Below I am representing a few entries from the edge section corresponding to a 2 robot, 20 hole problem, which will be interpreted for helping the user understand the entries described better.

EDGE_DESCRIPTION_SECTION:
ROBOT: 0
8 75 12
8 68 4
8 76 13
ROBOT: 1
8 112 12

10 166 17
10 159 9

The entry "8 75 12" under robot 0 means that, intermediate node with label 75 corresponds to the robot motion from node labelled 8 to node labelled 12 for robot 0. Likewise the entry "10 166 17" under robot 1 means that, intermediate node 166 corresponds to the motion from node labelled 10 to node labelled 17 for robot 1. Under robot 1, notice we have an entry "8 112 12", contrast it with "8 75 12" under robot 0 to understand that different nodes were required for different robots to traverse between the same pair of type V nodes.

### 2.4 COLLISION_DESCRIPTION_SECTION:

This section partially enumerates all the collision constraints (CCs) in the problem. A typical entry under this section looks as follows:
<NODE ID_1> <ROBOT INDEX 1> <NODE ID_2> <ROBOT INDEX 2>
Suppose robot ROBOT INDEX 1 services node with label NODE ID_1 and robot ROBOT INDEX 2 services node with label NODE ID_2 simultaneously, then a collision occurs. All feasible schedules must avoid such a possibility. NODE ID_1 and NODE ID_2 that are listed under the collision section, both correspond to nodes of type V. Also, for all entries listed under this section, it can be observed that ROBOT INDEX 1 < ROBOT INDEX 2.

As we mentioned in the paper, enumerating all collision constraints is not practical, as it makes the file sizes prohibitively large. So instead, the experiments in the paper were conducted using a function approximation which takes as input a pair of nodes, where the nodes can be of type V or IV, and robots occupying the node, for which the function outputs whether or not the robots will collide. If any of the input nodes is of type IV, then it is assumed that the robot occupying the node can only be the robot under which the IV node was enumerated in the EDGE_DESCRIPTION_SECTION. The function approximation makes use of the collision constraints that have been enumerated in the problem description file under the COLLISION_DESCRIPTION_SECTION, in order to output whether a collision occurs. Although an approximation, it was sufficient to ensure that most schedules that were generated on our real world test cases were collision free.

Before we define our function approximation for the collision function, we define a function $g : IV \rightarrow V \times V$ which takes as input an IV node label, and outputs the pair of V node labels. The pair of V node labels outputted correspond to nodes between which the IV node occurs on the routing graph. For example, for the first entry under the EDGE_DESCRIPTION_SECTION, i.e.

8 75 12

$g(75) = (8, 12)$. For convenience, with a slight abuse of notation let $g(75, 1) = 8$, and $g(75, 2) = 12$.

We define another function $h$, which takes as input 4 entries, namely $node1, r_1, node2, r_2$, where $node1, node2$ are node labels corresponding to nodes of type V, and $r_1, r_2$ are robot labels. $h(node1, r_1, node2, r_2)$ basically evaluates to true, iff either the entry $node1, r_1, node2, r_2$ or the entry $node2, r_2, node1, r_1$ is listed under the COLLISION_DESCRIPTION_SECTION of the problem description file.

So given functions $g, h$, we define our collision approximation function $f$ as follows:

```
bool f(node1, r1, node2, r2)
{
        if ((node1 is not an IV) && (node2 is not an IV))
        {
                return h(node1, r1, node2, r2);
        }
        else if ((node1 is an IV) && (node2 is not an IV))
        {
                v1 = g(node1, 1);
                v2 = g(node1, 2);
```

```
            if(true == h(v1, r1, node2, r2)) return true;
            if(true == h(v2, r1, node2, r2)) return true;
    }
    else if ((node1 is not an IV) && (node2 is an IV))
    {
            v1 = g(node2, 1);
            v2 = g(node2, 2);

            if(true == h(node1, r1, v1, r2)) return true;
            if(true == h(node1, r1, v2, r2)) return true;
    }
    else
    {
            v11 = g(node1, 1);
            v12 = g(node1, 2);

            v21 = g(node2, 1);
            v22 = g(node2, 2);

            if(true == h(v11, r1, v21, r2)) return true;
            if(true == h(v11, r1, v22, r2)) return true;
            if(true == h(v12, r1, v21, r2)) return true;
            if(true == h(v12, r1, v22, r2)) return true;
    }
    return false;
}
```

### 2.5 ENABLER_DESCRIPTION_SECTION:

A typical entry in this section is of the form:
<NODE ID> <NODE ID_1> <NODE ID_2> . . . <NODE ID_K>
All entries NODE ID, NODE ID_1, . . . , NODE ID_K are labels of node type V. The interpretation for the entry above is: NODE ID_1, . . . , NODE ID_K are the K enablers of the node whose label is NODE ID.

### 2.6 START_END_DESCRIPTION_SECTION:

In this section, we basically populate the sets $\mathbf{O}, \mathbf{D}$ (refer paper for Defn.). A typical entry looks like this:
<ROBOT INDEX> <NODE ID_1> <NODE ID_2>
For a 2 robot example we would have the following entries:

0 0 1
1 2 3

The entry "0 0 1" means the start location node for robot 0 is labelled 0 and terminal node is labelled 1. Likewise the start node for robot 1 is 2 and terminal node is 3. This section is somewhat redundant, but included for improving readability of the problem description.