*The format for the result file is explained in this document. The file name indicates the name of the instance. File names starting with* ***Results\_la*** *refer to Lawrence instances and file names beginning with* ***Results\_ta*** *refer to Taillard instances. In the descriptions below* **M***(resp.* **J***)**is the number of machines (resp. jobs) in the problem instance.*

*The result document contains 5 sections, namely* Makespan, Start, End, OPERATIONS SEQUENCE *and* MACHINE PREDECESSOR INFO: OPERATION, MACHINE PREDECESSOR. *In sections* Start, End, OPERATIONS SEQUENCE, we *essentially provide a number for all the operations \ nodes in the alternative graph* G(F ∪ S) *(refer Section 3 in the paper).* *We explain the content of each section separately below. The general template in which the results are presented is shown below. It is worthwhile to mention that the format in which the results are presented aligns closely with descriptions in our paper. Whenever example data is provided in this document, the data is taken from* ***Results\_la02.txt.***

Makespan: <Val>

*// In* ***Results\_la02.txt****, there is an entry* Makespan: 900. *This entry implies that the best makespan we obtained for the la02 instance using our local search algorithm is 900.*

Start: 0

// 0 *is the number* *assigned to the fictitious node* θ *(refer Section 3).*

End: **J**(**M** + 1) + 1

// **J**(**M** + 1) + 1 *is the number* *assigned to the fictitious node* Λ *(refer Section 3).*

*// In* ***Results\_la02.txt****, notice that the document contains the following entry ``*End:61*’’. This is because the la02 instance is a problem instance with 10 jobs and 5 machines, and so the formula mentioned adds up.*

OPERATIONS SEQUENCE:

1, 2, …, **M**+1 // Job -1’s nodes

**M**+2, …, 2(**M**+1) // Job -2’s nodes

………………………

(**J**-1)(**M**+1) + 1, … , **J** (**M**+1) // Job -**J**’s nodes

*// The data under* OPERATIONS SEQUENCE b*asically provides a number for all the nodes in the problem. We will explain the numbering using the data provided in the first row under* OPERATIONS SEQUENCE in ***Results\_la02.txt***:

1, 2, 3, 5, 6

*// The list above enumerates all the nodes belonging to Job 1 in its sequence. The last node* i.e. 6, is *a dummy node (refer Section 3 in the paper). So, each row under* OPERATIONS SEQUENCE *contains 1 node more than the number of machines in the problem. The processing duration of the dummy node is 0, and the processing duration for the remaining nodes can be obtained from the problem description file. The problem description files can be accessed from https://github.com/tamy0612/JSPLIB/tree/master/instances, but other websites have also shared the data. If the reader refers to the problem description file of la02 from the link posted, then the data corresponding to Job 1 is printed as shown below:*

0 20 3 87 1 31 4 76 2 17

*The above data implies, node 1 (i.e. the first node belonging to Job 1) is to be serviced on machine 0 and requires a processing duration of 20-time units. Similarly, node 2 (i.e. the second node belonging to job 1) needs to be serviced on machine 3 and requires 87-time units. If you refer to the problem description file, then no information is provided about the dummy node and rightly so. The dummy node is introduced only for the purposes of explanations in the paper.*

*Now let us move to the second row under* OPERATIONS SEQUENCE *in* ***Results\_la02.txt***.

7,8,9,10,11,12

*The data corresponding to Job 2 in the problem description file (i.e. la02) is as follows:*

4 25 2 32 0 24 1 18 3 81

*The above data implies that node 7* (i.e. the first node of Job -2)*needs to be serviced on machine 4 and requires 25-time units. Likewise, node 8* *(i.e.* *second node of Job-2) needs to be serviced on machine 2 and requires 32-time units.*

*The nodes in the remaining sequences needs to also be interpreted in a similar way.*

MACHINE PREDECESSOR INFO: OPERATION, MACHINE PREDECESSOR

Oper : Machine Predecessor(Oper)

*In this section, we encode the solution outputted by our local search algorithm whose makespan value was listed earlier. We specify the solution by listing the operation and it’s machine predecessor in that solution. So, for instance, if you refer to the first row under* MACHINE PREDECESSOR INFO: OPERATION, MACHINE PREDECESSOR *from* ***Results\_la02.txt****, then you would observe it says:*

1, 59

*It means that the machine predecessor of node 1 is node 59. The reader can verify that both nodes 1 and 59 (the penultimate node of Job 10) require machine 0 to be serviced. So, in the solution that we report, node 59 is serviced before node 1 on machine 0.*

*There are also entries in this section where only one entry is provided. So, for example in* ***Results\_la02.txt****, there are entries without the machine predecessor specified. For example, these are some entries such as those shown as follows:*

*6*

*8*

*Notice from the node sequence of Job 1, 6 is a dummy node. Since dummy nodes are not associated with any machine, we do not specify a machine predecessor for dummy nodes.*

*Node 8 is the second node belonging to Job 2, and it requires machine 2 for servicing. Since a machine predecessor is not specified, it means that 8 is the first node that is serviced on machine 2 among all those nodes that also require machine 2.*

*All the entries under the* MACHINE PREDECESSOR INFO: OPERATION, MACHINE PREDECESSOR ca*n be interpreted in a similar fashion as described previously. Using all that information, the complete solution outputted by our local search algorithm can be constructed.*