# Heuristics for Scheduling Pipe-laying Support Vessels: An Identical Parallel Machine Scheduling Approach

Victor Abu-Marrul[1], Davi Mecler[1], Rafael Martinelli[1], Silvio Hamacher[1] and Irina Gribkovskaia[2]

[1] Industrial Engineering Department, Pontifical Catholic University of Rio de Janeiro, Brazil
{victor.cunha,davizm}@tecgraf.puc-rio.br, {martinelli,hamacher}@puc-rio.br
[2] Molde University College - Specialized University in Logistics, Norway
irina.gribkovskaia@himolde.no

**Keywords:** Ship Scheduling, Offshore Logistics, Heuristic, Parallel Machine Scheduling.

## 1 Introduction and Problem Description

We address a problem that emerges in oil & gas offshore logistics where a company needs to schedule its fleet of pipe laying support vessels (PLSV), responsible for connecting sub-sea oil wells to production platforms. We model it as an identical parallel machine scheduling problem with non-anticipatory family setup times. Here, the vessels are machines, jobs are wells, and each job includes a number of connecting operations. The problem consists of scheduling a set $\mathcal{O}$ of operations from a set $\mathcal{F}$ of families in a set $\mathcal{M}$ of machines. Each operation $i \in \mathcal{O}$ belongs to a family $(f_i)$, has a release date $(r_i)$, a processing time $(p_i)$, a load occupancy $(l_i)$, can be assigned to an eligible subset of machines $(\mathcal{M}_i)$, and is associated with a subset of jobs $(\mathcal{N}_i)$. Each job $j \in \mathcal{N}$ has a weight $(w_j)$ according to the related well production rate, and a subset $\mathcal{O}_j$ of the operations associated with it. Each machine $k \in \mathcal{M}$ has a release date $(r_k)$ and a capacity $(q_k)$. A non-anticipatory family setup time $(s_f)$ is incurred before the first operation on each machine, whenever a machine changes the execution of operations between families, and when the capacity of a machine is reached. A set of operations that shares the same setup time is called a batch. Setups are non-anticipatory since they can only start when all operations scheduled inside a batch are released. The sum of the load occupancy of the operations assigned to a batch must respect the machine capacity. The objective is to minimize the total weighted completion time of jobs $(\sum_{j \in \mathcal{N}} w_j C_j)$, where the completion time of job $j$ $(C_j)$ is the maximum completion time of the associated operations $(C_j = \max_{i \in \mathcal{O}_j} C_i)$. This objective aims to complete the connections of the most productive oil wells as soon as possible. An example of a PLSV schedule is depicted in Figure 1 with 15 operations, 5 jobs, and 4 machines. The associated jobs are shown in the boxes, and their completion times are marked with dotted lines.

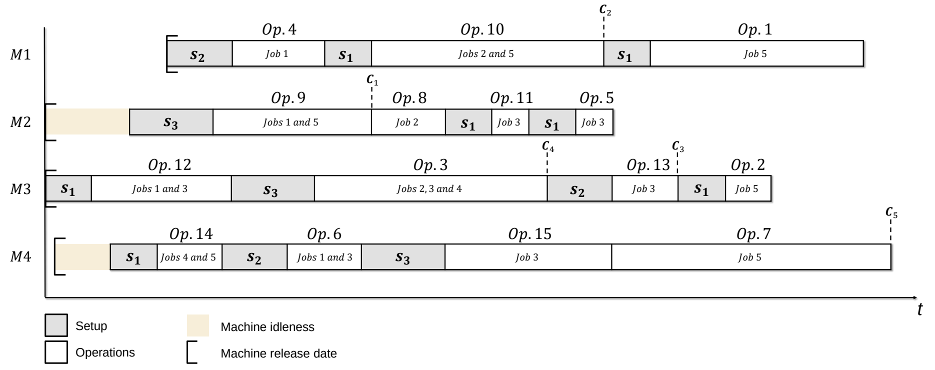

Fig. 1: PLSV Scheduling example with 15 operations, 5 Jobs and 4 machines.

## 2 Solution Methods

We present several constructive heuristics to solve the PLSV scheduling problem, using dispatching rules, and defining how to do the machine assignment and to construct batches. The schedule construction procedure used in all heuristics is presented in Algorithm 1. To describe the procedure, we introduce variables and sets: $S_k$ (Start time of the current batch on machine $k$), $L_k$ (Cumulative load of the current batch on machine $k$), $F_k$ (Family of the current batch on machine $k$), $C_k$ (Completion time of machine $k$), *same* (Boolean variable that defines if the chosen operation will be assigned to the last position in the current batch or to a new batch), $\mathcal{B}_k$ (Set of operations assigned to the current batch on machine $k$) and $\mathcal{U}$ (Set of unscheduled operations). The procedure returns a list of schedules $\sigma = (\sigma_1, \ldots, \sigma_k)$ for each machine $k$ containing operations and families (representing the setup times).

**Algorithm 1:** Schedule Construction Procedure

```
 1: $C_k \leftarrow r_k, S_k \leftarrow r_k,\ L_k \leftarrow 0,\ F_k \leftarrow 0,\ \mathcal{B}_k \leftarrow \emptyset, \sigma_k \leftarrow \emptyset : \ \forall k \in \mathcal{M}$
 2: $C_i \leftarrow \infty : \ \forall i \in \mathcal{O}$
 3: $\mathcal{U} \leftarrow \mathcal{O}$
 4: while there exists operations not assigned do
 5:    Select operation $i^* \in \mathcal{U}$ and machine $k^* \in \mathcal{M}_{i^*}$ according to a chosen heuristic, defining
       same as true or false
 6:    if same = true then
 7:       $S_{k^*} \leftarrow \max(r_{i^*}, S_{k^*}),\ C_{k^*} \leftarrow C_{k^*} + \max(0, r_{i^*} - S_{k^*}) + p_{i^*}$
 8:       $L_{k^*} \leftarrow L_{k^*} + l_{i^*},\ \mathcal{B}_{k^*} \leftarrow \mathcal{B}_{k^*} \cup \{i^*\}$
 9:    else
10:       $S_{k^*} \leftarrow \max(r_{i^*}, C_{k^*}),\ C_{k^*} \leftarrow \max(r_{i^*}, C_{k^*}) + s_{f_{i^*}} + p_{i^*}$
11:       $L_{k^*} \leftarrow l_{i^*},\ \mathcal{B}_{k^*} \leftarrow \{i^*\},\ \sigma_{k^*} \leftarrow \sigma_{k^*} \cup \{f_{i^*}\}$
12:    end if
13:    $\sigma_{k^*} \leftarrow \sigma_{k^*} \cup \{i^*\},\ F_{k^*} \leftarrow f_{i^*},\ C_{i^*} \leftarrow C_{k^*},\ \mathcal{U} \leftarrow \mathcal{U} \setminus \{i^*\}$
14: end while
15: return $\sigma$
```

The main part of the method is defined at line 5, the decision of the next operation, machine, and batch composition. We consider two approaches at this step. In both, when we evaluate the insertion of an operation to an existing batch, we consider the delay on the start time of this batch that may occur due to the inserted operation release date and the non-anticipatory setup consideration. This delay changes the completion time of all operations scheduled in the batch, and it is penalized using the operations weights. We develop rules to estimate weights for the operations ($w_i$), since weights in our problem are related to jobs. Five rules for estimating weights are considered:

- `MAX`: Maximum weight of associated wells, computed as $w_i = \max_{j \in \mathcal{N}_i} w_j$
- `SUM`: Sum of the weights of associated wells, computed as $w_i = \sum_{j \in \mathcal{N}_i} w_j$
- `AVG`: Average weight of associated wells, computed as $w_i = \sum_{j \in \mathcal{N}_i} w_j / |\mathcal{N}_i|$
- `WAVG`: Weighted average weight of associated wells, computed as $w_i = \sum_{j \in \mathcal{N}_i} w_j / |\mathcal{O}_j|$
- `WAVGA`: `WAVG` adjusted at each iteration by the set of unscheduled operations (i.e., it replaces subsets $\mathcal{O}_j$ by subsets $\mathcal{U}_j$ of unscheduled operations associated to a job $j$).

In the first approach, we initially select the next operation based on a dispatching rule, and then assign an eligible machine to this operation according to the minimum weighted completion time, as described in Algorithm 2. New variables and sets are defined: $\Delta_{ik}$ (Delay at the start time of the current batch on machine $k$ due to the insertion of operation $i$), $C_{ik}^{CB}$ (Completion time of operation $i$ if inserted in the current batch on machine $k$), $C_{ik}^{NB}$ (Completion time of operation $i$ if inserted in a new batch on machine $k$), $\mathcal{CB}$ (Set of feasible assignments $cb_{ik}$ of operation $i$ into the current batch on machine $k$) and $\mathcal{NB}$ (Set of feasible assignments $nb_{ik}$ of operation $i$ into a new batch on machine $k$).

**Algorithm 2:** Operation and Machine Disjunctive Selection Procedure

```
1: Select operation $i^* \in \mathcal{U}$ according to a chosen dispatching rule
2: $\Delta_{i^*k} \leftarrow \max(0, r_{i^*} - S_k) : \ \forall k \in \mathcal{M}_{i^*}$
3: $C_{i^*k}^{CB} \leftarrow C_k + \Delta_{i^*k} + p_{i^*} : \ \forall k \in \mathcal{M}_{i^*}$
4: $C_{i^*k}^{NB} \leftarrow \max(r_{i^*}, C_k) + s_{f_{i^*}} + p_{i^*} : \ \forall k \in \mathcal{M}_{i^*}$
5: $\mathcal{CB} \leftarrow \left\{ cb_{i^*k} = w_{i^*} C_{i^*k}^{CB} + \sum_{i \in \mathcal{B}_k} w_i \Delta_{i^*k} \mid k \in \mathcal{M}_{i^*},\ F_k = f_{i^*},\ L_k + l_{i^*} \leq q_k \right\}$
6: $\mathcal{NB} \leftarrow \left\{ nb_{i^*k} = w_{i^*} C_{i^*k}^{NB} \mid k \in \mathcal{M}_{i^*} \right\}$
7: $b_{min} \leftarrow \min\{b : b \in (\mathcal{CB} \cup \mathcal{NB})\}$
8: Select $k^*$ corresponding to $b_{min}$
9: if $b_{min} \in \mathcal{CB}$ then $same \leftarrow$ true
10: return $i^*, k^*, same$
```

We consider six dispatching rules for this approach. The priority value $\pi_i$ indicates the next operation to schedule. At each iteration, the operation $i$ with the largest $\pi_i$ value is selected (Đurasević and Jakobović 2018). We adapt some rules by adding family setup times, and assume that every operation will be assigned to a new batch. $T_i$ is the minimum completion time among the eligible vessels for each operation $i$, and is computed as $T_i = min_{k \in \mathcal{M}_i} C_k$. The following dispatching rules are considered:

- `ERD`: Earliest Release Date $\pi_i = 1/r_i$.
- `SPT`: Shortest Processing Time $\pi_i = 1/p_i$.
- `LPT`: Longest Processing Time $\pi_i = p_i$.
- `MCT`: Minimum Completion Time $\pi_i = \max(T_i, r_i) + p_i + s_{f_i}$.
- `WSPT`: Weighted Shortest Processing Time $\pi_i = w_i/p_i$.
- `WMCT`: Weighted Minimum Completion Time $\pi_i = [\max(T_i, r_i) + p_i + s_{f_i}]/w_i$.

The second approach extends one of the heuristics from Weng et al.(2001), by considering the PLSV scheduling properties, such as the release dates of operations and machines, the family setup times and the batch composition, to decide at each iteration the next pair operation/machine simultaneously (Algorithm 3). We call it `WMCT-Pair`.

**Algorithm 3:** Operation and Machine Simultaneous Selection Procedure

```
1: $\Delta_{ik} \leftarrow \max(0, r_i - S_k) : \ \forall i \in \mathcal{U},\ k \in \mathcal{M}_i$
2: $C_{ik}^{CB} \leftarrow C_k + \Delta_{ik} + p_i : \ \forall i \in \mathcal{U}, k \in \mathcal{M}_i$
3: $C_{ik}^{NB} \leftarrow \max(r_i, C_k) + s_{f\,i} + p_i : \ \forall i \in \mathcal{U}, k \in \mathcal{M}_i$
4: $\mathcal{CB} \leftarrow \left\{ cb_{ik} = \frac{C_{ik}^{CB}}{w_i} + \sum_{\hat{i} \in \mathcal{B}_k} \frac{\Delta_{ik}}{w_i} \mid i \in \mathcal{U},\ k \in \mathcal{M}_i,\ F_k = f_i,\ L_k + l_i \leq q_k \right\}$
5: $\mathcal{NB} \leftarrow \left\{ nb_{ik} = \frac{C_{ik}^{NB}}{w_i} \mid i \in \mathcal{U},\ k \in \mathcal{M}_i \right\}$
6: $b_{min} \leftarrow \min\{b : b \in (\mathcal{CB} \cup \mathcal{NB})\}$
7: Select $i^*$ and $k^*$ corresponding to $b_{min}$
8: if $b_{min} \in \mathcal{CB}$ then $same \leftarrow$ true
9: return $i^*, k^*, same$
```

## 3 Computational Experiments

We introduce in total 19 heuristics, where 4 do not consider weights and 15 combine the dispatching rules and the ways of estimating the operations' weights. We tested all of them on a set of 72 PLSV instances[3] with $|\mathcal{M}| = \{2, 4\}$, and $|\mathcal{O}| = \{15, 25, 50\}$. We performed

[3] available at https://doi.org/10.17771/PUCRio.ResearchData.45799

the experiments on a computer with 64 GB of RAM and Intel Core i7-8700K CPU of 3.70GHz, using C++ for coding the heuristics and running Linux. The results of tests, in terms of the average relative deviations from the best generated solutions, are presented in Table 1. Each instance group, defined by the number of operations and machines, contains 12 instances. The relative deviation is computed as $RD^h_{inst} = TWC^h_{inst}/TWC^{best}_{inst}$, where $TWC^h_{inst}$ is the total weighted completion time of heuristic $h \in \mathcal{H}$ applied to instance $inst \in \mathcal{I}$, and $TWC^{best}_{inst}$ is the best solution obtained for a given instance. The best result for each instance group is shown in bold. All heuristics run in less than 0.1 seconds. Last column ($\#BKS$) accounts how many times each heuristic yields the best solution.

Table 1: Average deviations from the best solutions.

| **Heuristic** | **Instance Group ($\|\mathcal{O}\| - \|\mathcal{M}\|$)** | | | | | | **All Instances** | **#BKS** |
|---|---|---|---|---|---|---|---|---|
| | **15-4** | **15-8** | **25-4** | **25-8** | **50-4** | **50-8** | | |
| ERD | 1.212 | 1.198 | 1.245 | 1.190 | 1.261 | 1.250 | 1.226 | 2 |
| SPT | 1.278 | 1.217 | 1.363 | 1.296 | 1.442 | 1.392 | 1.331 | 1 |
| LPT | 1.347 | 1.153 | 1.412 | 1.265 | 1.471 | 1.398 | 1.341 | 0 |
| MCT | 1.226 | 1.165 | 1.268 | 1.248 | 1.254 | 1.295 | 1.243 | 0 |
| WSPT-MAX | 1.161 | 1.079 | 1.224 | 1.173 | 1.299 | 1.255 | 1.198 | 1 |
| WSPT-SUM | 1.156 | 1.066 | 1.181 | 1.141 | 1.280 | 1.241 | 1.178 | 0 |
| WSPT-AVG | 1.181 | 1.085 | 1.266 | 1.187 | 1.327 | 1.294 | 1.223 | 1 |
| WSPT-WAVG | 1.124 | 1.076 | 1.184 | 1.124 | 1.234 | 1.196 | 1.156 | 0 |
| WSPT-WAVGA | 1.085 | 1.041 | 1.112 | 1.067 | 1.138 | 1.114 | 1.093 | 3 |
| WMCT-MAX | 1.084 | 1.040 | 1.070 | 1.088 | 1.046 | 1.096 | 1.071 | 7 |
| WMCT-SUM | 1.063 | 1.046 | 1.093 | 1.082 | 1.158 | 1.132 | 1.096 | 3 |
| WMCT-AVG | 1.101 | 1.027 | 1.123 | 1.118 | 1.150 | 1.147 | 1.111 | 4 |
| WMCT-WAVG | 1.065 | 1.036 | 1.041 | 1.059 | 1.111 | 1.084 | 1.066 | 5 |
| WMCT-WAVGA | **1.023** | **1.021** | **1.013** | **1.016** | 1.027 | **1.003** | **1.017** | 34 |
| WMCT-Pair-MAX | 1.086 | 1.043 | 1.063 | 1.082 | 1.036 | 1.091 | 1.067 | 9 |
| WMCT-Pair-SUM | 1.063 | 1.043 | 1.087 | 1.071 | 1.151 | 1.124 | 1.090 | 4 |
| WMCT-Pair-AVG | 1.101 | 1.027 | 1.123 | 1.107 | 1.136 | 1.136 | 1.105 | 2 |
| WMCT-Pair-WAVG | 1.060 | 1.035 | 1.045 | 1.050 | 1.100 | 1.076 | 1.061 | 7 |
| WMCT-Pair-WAVGA | 1.029 | 1.023 | 1.024 | 1.018 | **1.026** | 1.005 | 1.021 | 19 |

Note that among the heuristics, `WCMT-WAVGA` generated the best average solutions for 5 of 6 groups with the best average deviation of 1.003, achieved on group 50-8. This heuristic also found the highest number of best solutions, on 34 of 72 instances.

## 4 Conclusions

We studied an identical parallel machine scheduling problem with non-anticipatory family setup times, derived from a ship scheduling problem in the offshore oil & gas logistics. Tests of the 19 heuristics presented on all instances show that the heuristic `WCMT-WAVGA` performs better, with an average deviation of 1.017 from the best solutions. For future work, local searches and meta-heuristics will be developed.

## References

Đurasević, M; Jakobović, D. "A survey of dispatching rules for the dynamic unrelated machines environment", *Expert Systems with Applications*, Vol. 113, pp. 555-569, 2018.

Weng, M.; Lu, J.; Ren, H. "Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective", *International journal of production economics*, Vol. 70, pp. 215-226, 2001.