

# Datasets in "Heuristic solutions to minimise makespan in a hybrid flow shop scheduling environment with energy consumption constraints in steel making"

Brighton Miyelani Baloi, 14202469

April 19, 2024

## 0.1 Main algorithm

### **Main Algorithm**

```
//Comment: Step 1 - use small number of runs for all problem sizes
Use the general processing time matrix and all LLHs as inputs
For every LLH
    Allocate machines based on the current processing rule
    Create the actual processing time based on the current LLH rule
    For every job sequencing rule
        Sequence job using the current sequencing rule
        Select the sequence with the best makespan
        Reoptimise for energy constraint
        Save the best makespan and energy level
    endFor
endFor
Return the best LLH

//Comment: Step 2 - use large number of runs
Use the selected LLH and as input
For each job sequencing rule (IHNEH/GA) for all problem sizes
    Input selected LLH
    Schedule jobs using the current sequencing rule
    Find the optimum makespan sequence
    Improve best sequence using neighbourhood search
    Re-optimize for energy constraint
    Record best improved makespan sequence and energy level
endFor

//Comment: Step 3 - solution performance evaluation
Solve small sized problems with the branch and bond algorithm
For each sequencing algorithm (IHNEH and IHGA)
    Compare the solution quality of small sized problems to branch and bound solution
    Compare the time behaviour of proposed algorithms for medium and large problem sizes
    Compare the solution quality of proposed algorithms for medium and large problem sizes
endFor
```

Figure 1: The pseudocode for the main algorithm.

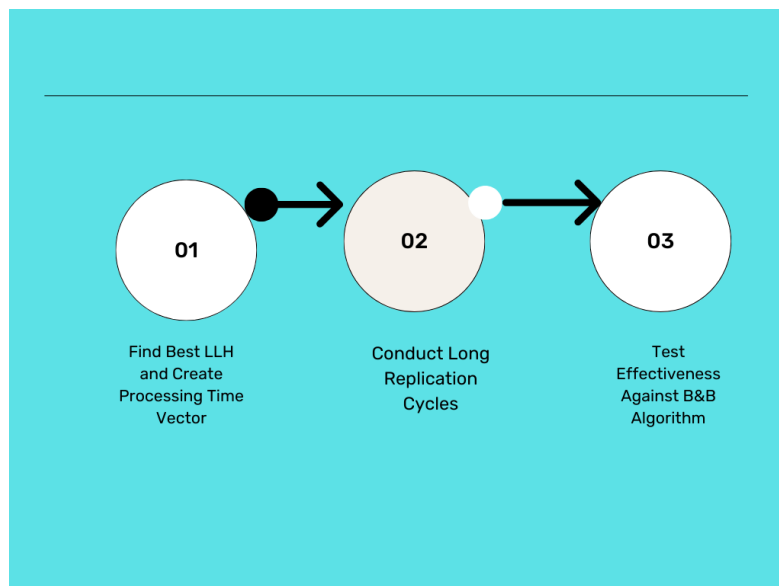


Figure 2: The three main steps of the main algorithm

## 0.2 The Hyper Heuristic

Property	LLH 1	LLH2	LLH3	LLH4	LLH5	LLH6
position of jobs	Y	N	N	Y	Y	Y
processing time	N	Y	Y	Y	N	Y
attack time	N	Y	Y	N	N	N
speed scaling	N	Y	Y	Y	N	Y

Table 1: Design structure for the LLHs.

<b><u>LLH1:</u></b> Assign first set half of jobs to the slower and the rest to the faster machine
<b><u>LLH2:</u></b> Assign job, first to machine with fast attack time if it is available Else, assign job to the machine with the slow attack time, if it is available Else wait for the first machine to be available  When half of the jobs have been processed If the 50 percent of the energy threshold has been exceeded Reduce processing speed (divide by 2) Increase processing time (multiply by 2) Continue processing with the new speed and processing time Else Continue processing with the original speed and processing time endif
<b><u>LLH3:</u></b> Assign job, first to machine with slow attack time if it is available Else, assign job to the machine with the fast attack time, if it is available Else wait for the first machine to be available  When half of the jobs have been processed If the 50 percent of the energy threshold has been exceeded Reduce processing speed (divide by 2) Increase processing time (multiply by 2) Continue processing with the new speed and processing time Else Continue processing with the original speed and processing time endif
<b><u>LLH4</u></b> Assign first set half of jobs to the slower and the rest to the faster machine  When half of the jobs have been processed If the 50 percent of the energy threshold has been exceeded Reduce processing speed (divide by 2) Increase processing time (multiply by 2) Continue processing with the new speed and processing time Else Continue processing with the original speed and processing time endif
<b><u>LLH5</u></b> Assign first set half of jobs to the faster and the rest to the slower machine
<b><u>LLH6</u></b> Assign first set half of jobs to the faster and the rest to the slower machine  When half of the jobs have been processed If the 50 percent of the energy threshold has been exceeded Reduce processing speed (divide by 2) Increase processing time (multiply by 2) Continue processing with the new speed and processing time

Figure 3: The algorithms of the LLHs.

### 0.3 NEH algorithm

```
For every job
    Calculate the sum of the processing times over all machines
endFor
For every job
    Order the jobs in a descending order of total processing time
endFor

Form a partial sequence using the first two jobs
Find the permutation with the smallest makespan
For job = 3 to the last job
    Insert job j in all possible locations in the current partial sequence
    Calculate makespan for each position of insertion
    Return the partial sequence with the minimum makespan
endFor
```

Figure 4: NEH pseudocode.

### 0.4 The genetic algorithm

```
Initialise all simulation parameters
    nIteration, propRetained, numMutation, popSize, chromPopulation,
    optMakespan, optChrom

Evaluate population fitness of chromosomes and rank them
    Calculate makespan for each chromosome
    Rank chromosomes in ascending order of makespan

Repeat for iteration 2 until nIteration
    Retain top chromosomes for reproduction (use propRetained)
    Repeat until new population is created
        Select two chromosomes from retained chromosomes
        use roulette wheel rank-based selection
        Create offspring through crossover and put in new population
        Repair offspring from gene duplication and omission
        Mutate offspring (use numMutation)
    endRepeat

    Evaluate fitness of new chromosome population and rank them
        Calculate makespan for each chromosome
        Rank chromosomes in ascending order of makespan

    Update optMakespan and the optChrom
endRepeat
```

Figure 5: GA pseudocode.

#### 0.4.1 The branch and bound procedure

```

Set all jobs as unscheduled

While there is still an unscheduled job
    While not yet at the leaf node
        If optimality condition fails
            Compute makespan Lower Bound (LB) for all jobs
            Select the job with the minimum LB and branch
            Remove the job selected from unscheduled jobs
        Else
            Bubble up from node
        endif
    endwhile

    If job is leaf node
        Calculate makespan
        Calculate energy consumed
        If energy consumed is below threshold
            Update optimal makespan and energy level
        Else
            Return last scheduled job to unscheduled list
            Bubble up from node
        endif
    endif
endwhile

```

Figure 6: The modified B&B pseudocode.