Minimizing the weighted completion time on a single machine with periodic maintenance

KRIM Hanane

University of Valenciennes and Hainaut-Cambrésis

LAMIH UMR CNRS 8201

1st year Phd Student

February 12, 2016

Advisors:

Dr. David DUVIVIER

Dr. Rachid BENMANSOUR

Plan

1. Introduction
2. $1/pm/\sum_{i=1}^{n} W_i C_i$ Problem
3. Resolution
4. Lower bound
5. Computational results
6. Conclusion

Plan

1. Introduction
2. $1/pm/\sum_{i=1}^{n} W_i C_i$ Problem
3. Resolution
4. Lower bound
5. Computational results
6. Conclusion

Plan

1. Introduction
2. $1/pm/\sum_{i=1}^{n} W_iC_i$ Problem
3. Resolution
4. Lower bound
5. Computational results
6. Conclusion

Plan

1. Introduction
2. $1/pm/\sum_{i=1}^{n} W_i C_i$ Problem
3. Resolution
4. Lower bound
5. Computational results
6. Conclusion

Plan

1. Introduction
2. $1/pm/\sum_{i=1}^{n} W_i C_i$ Problem
3. Resolution
4. Lower bound
5. Computational results
6. Conclusion

Plan

1. Introduction
2. $1/pm/\sum_{i=1}^{n} W_iC_i$ Problem
3. Resolution
4. Lower bound
5. Computational results
6. Conclusion

Plan

Introduction

Maintenance Activity

Maintenance: Actions necessary for retaining or restoring a piece of equipment, machine, or system to the specified operable condition to achieve its maximum useful life.

Preventive maintenance: Systematic inspection, detection, correction, and prevention of incipient failures, before they become actual or major failures.

Corrective maintenance: Activities undertaken to detect, isolate, and rectify a fault so that the failed equipment, machine, or system can be restored to its normal operable state

Plan

Why/When/Where: Industrial process with single critical machine in the shop. [1]

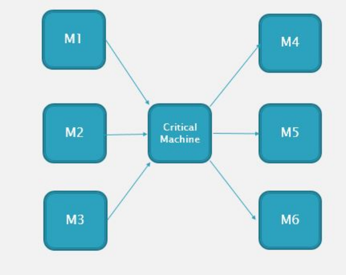


Figure 1 : Example 1

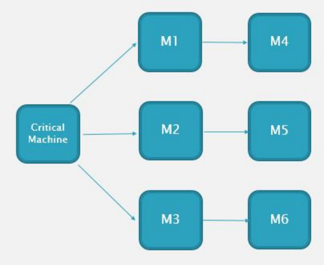


Figure 2 : Example 2

Why/When/Where: Optimize the sum of weighted completion time $\sum_{i=1}^{n} W_i C_i$.

In industry, the $\sum_{i=1}^{n} W_i C_i$ criteria allows to estimate the cost of the stocks.

[1] $\rightarrow$: Flow of semi-finished products

Why/When/Where: Industrial process with single critical machine in the shop. [1]

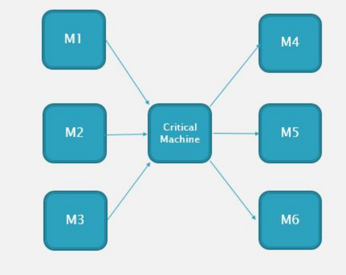


Figure 1 : Example 1

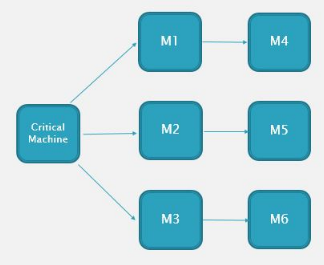


Figure 2 : Example 2

Why/When/Where: Optimize the sum of weighted completion time $\sum_{i=1}^{n} W_i C_i$.

In industry, the $\sum_{i=1}^{n} W_i C_i$ criteria allows to estimate the cost of the stocks.

[1] $\rightarrow$: Flow of semi-finished products

Review

Qi et al. invistigate the $1/pm/\sum_{i=1}^{n} C_i$ problem and prooved that this problem is NP-Hard.(1999)

C.J Liao et al. study the 1/pm/T_{max} problem. The preventive maintenance should be done after each T period of time. (2001)

Min Ji et al. consider the same problem than Liao, but minimize the makespan.(2005)

WJ. Chen et al. consider the same problem with the objective of minimizing the total flow time.(2005)

Ying Ma et al. made a survey of scheduling with deterministic machine availability constraints.(2010)

Review

Qi et al. invistigate the $1/pm/\sum_{i=1}^{n} C_i$ problem and prooved that this problem is NP-Hard.(1999)

C.J Liao et al. study the 1/pm/T_{max} problem. The preventive maintenance should be done after each T period of time. (2001)

Min Ji et al. consider the same problem than Liao, but minimize the makespan.(2005)

WJ. Chen et al. consider the same problem with the objective of minimizing the total flow time.(2005)

Ying Ma et al. made a survey of scheduling with deterministic machine availability constraints.(2010)

Review

Qi et al. invistigate the $1/pm/\sum_{i=1}^{n} C_i$ problem and prooved that this problem is NP-Hard.(1999)

C.J Liao et al. study the 1/pm/T_{max} problem. The preventive maintenance should be done after each T period of time. (2001)

Min Ji et al. consider the same problem than Liao, but minimize the makespan.(2005)

WJ. Chen et al. consider the same problem with the objective of minimizing the total flow time.(2005)

Ying Ma et al. made a survey of scheduling with deterministic machine availability constraints.(2010)

Review

Qi et al. invistigate the $1/pm/\sum_{i=1}^{n} C_i$ problem and prooved that this problem is NP-Hard.(1999)

C.J Liao et al. study the 1/pm/T_{max} problem. The preventive maintenance should be done after each T period of time. (2001)

Min Ji et al. consider the same problem than Liao, but minimize the makespan.(2005)

WJ. Chen et al. consider the same problem with the objective of minimizing the total flow time.(2005)

Ying Ma et al. made a survey of scheduling with deterministic machine availability constraints.(2010)

Review

Qi et al. invistigate the $1/pm/\sum_{i=1}^{n} C_i$ problem and prooved that this problem is NP-Hard.(1999)

C.J Liao et al. study the 1/pm/T_{max} problem. The preventive maintenance should be done after each T period of time. (2001)

Min Ji et al. consider the same problem than Liao, but minimize the makespan.(2005)

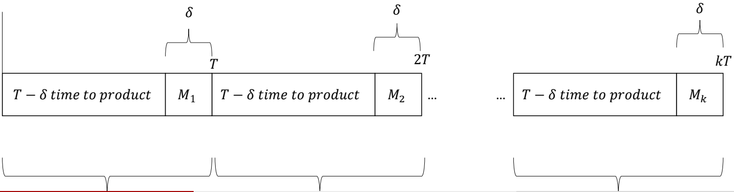
WJ. Chen et al. consider the same problem with the objective of minimizing the total flow time.(2005)

Ying Ma et al. made a survey of scheduling with deterministic machine availability constraints.(2010)

Problem definition

The problem considered can be stated as follows:

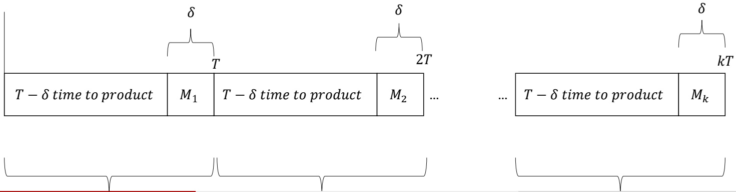
- A set $N = \{1, 2, \ldots, n\}$ of n independent jobs
- P_i, S_i, C_i, W_i represent, respectively, the processing time of job i, its starting time, its completion time and its weight.
- Each job, has to be processed without preemption on a single machine that can handle only one job at a time.
- The machine has to undergo a preventive maintenance each T units of time.
- δ is the duration of the maintenance.
- The periodic maintenances defines a batch. We can have at most n batches.



Problem definition

The problem considered can be stated as follows:

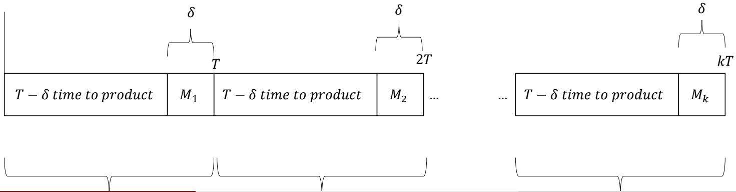
- A set $N = \{1, 2, \ldots, n\}$ of n independent jobs
- P_i, S_i, C_i, W_i represent, respectively, the processing time of job i, its starting time, its completion time and its weight.
- Each job, has to be processed without preemption on a single machine that can handle only one job at a time.
- The machine has to undergo a preventive maintenance each T units of time.
- δ is the duration of the maintenance.
- The periodic maintenances defines a batch. We can have at most n batches.



Problem definition

The problem considered can be stated as follows:

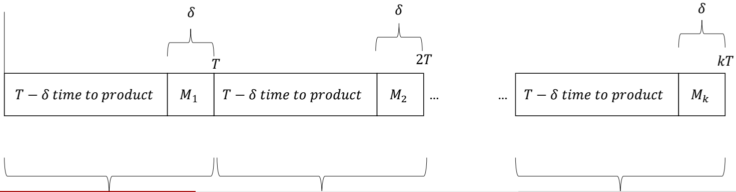
- A set $N = \{1, 2, \ldots, n\}$ of n independent jobs
- P_i, S_i, C_i, W_i represent, respectively, the processing time of job i, its starting time, its completion time and its weight.
- Each job, has to be processed without preemption on a single machine that can handle only one job at a time.
- The machine has to undergo a preventive maintenance each T units of time.
- δ is the duration of the maintenance.
- The periodic maintenances defines a batch. We can have at most n batches.



Problem definition

The problem considered can be stated as follows:

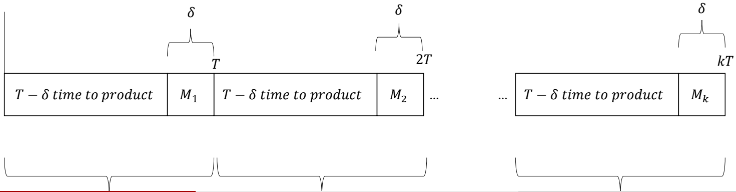
- A set $N = \{1, 2, \ldots, n\}$ of n independent jobs
- P_i, S_i, C_i, W_i represent, respectively, the processing time of job i, its starting time, its completion time and its weight.
- Each job, has to be processed without preemption on a single machine that can handle only one job at a time.
- The machine has to undergo a preventive maintenance each T units of time.
- δ is the duration of the maintenance.
- The periodic maintenances defines a batch. We can have at most n batches.



Problem definition

The problem considered can be stated as follows:

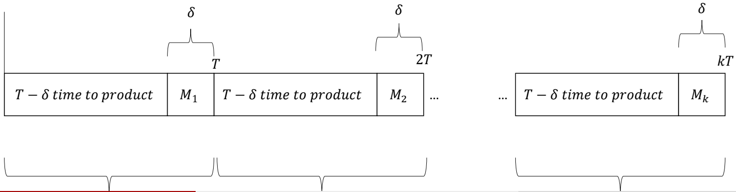
- A set $N = \{1, 2, \ldots, n\}$ of n independent jobs
- P_i, S_i, C_i, W_i represent, respectively, the processing time of job i, its starting time, its completion time and its weight.
- Each job, has to be processed without preemption on a single machine that can handle only one job at a time.
- The machine has to undergo a preventive maintenance each T units of time.
- δ is the duration of the maintenance.
- The periodic maintenances defines a batch. We can have at most n batches.



Problem definition

The problem considered can be stated as follows:

- A set $N = \{1, 2, \ldots, n\}$ of n independent jobs
- P_i, S_i, C_i, W_i represent, respectively, the processing time of job i, its starting time, its completion time and its weight.
- Each job, has to be processed without preemption on a single machine that can handle only one job at a time.
- The machine has to undergo a preventive maintenance each T units of time.
- δ is the duration of the maintenance.
- The periodic maintenances defines a batch. We can have at most n batches.



Plan

MIP formulation

We propose a MIP formulation based on precedence variables.

$$x_{ik} = \begin{cases} 1 & \text{if job } i \text{ is scheduled before job } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if job } i \text{ is in batch } j \\ 0 & \text{otherwise} \end{cases}$$

- The set of jobs scheduled between two periodic maintenances defines a batch. We can have at most n batches.
- We define also R as a big positive integer.

Objective function

$$\sum_{i=1}^{n} W_i \,(S_i + P_i)$$

Do not exceed available time per batch

$$\sum_{i=1}^{n} P_i \, y_{ij} \leq T - \delta \quad \forall j \in N$$

Perform all jobs

$$\sum_{j=1}^{n} y_{ij} = 1 \quad \forall i \in N$$

Compute starting times while avoiding overlaps(disjunctive constraints)

$$S_i \geq (j-1)(T)y_{ij} \quad \forall i \in N, \forall j \geq 2$$

$$S_i + p_i \leq S_k + R(1 - x_{ik}) \quad \forall i \in 1..n-1, k \in i+1,..n$$

$$S_k + P_k \leq S_i + Rx_{ik} \quad \forall i \in 1..n-1, k \in i+1,..n$$

Define domain for the variables

$$S_i \geq 0 \quad \forall i \in N$$

$$x_{ik} \in \{0,1\} \quad \forall (i,k) \in N^2$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in N^2$$

Objective function

$$\sum_{i=1}^{n} W_i \, (S_i + P_i)$$

Do not exceed available time per batch

$$\sum_{i=1}^{n} P_i \, y_{ij} \leq T - \delta \quad \forall j \in N$$

Perform all jobs

$$\sum_{j=1}^{n} y_{ij} = 1 \quad \forall i \in N$$

Compute starting times while avoiding overlaps(disjunctive constraints)

$$S_i \geq (j-1)(T)y_{ij} \quad \forall i \in N, \forall j \geq 2$$

$$S_i + p_i \leq S_k + R(1 - x_{ik}) \quad \forall i \in 1..n-1, k \in i+1, ..n$$

$$S_k + P_k \leq S_i + Rx_{ik} \quad \forall i \in 1..n-1, k \in i+1, ..n$$

Define domain for the variables

$$S_i \geq 0 \quad \forall i \in N$$

$$x_{ik} \in \{0, 1\} \quad \forall (i, k) \in N^2$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in N^2$$

Objective function

$$\sum_{i=1}^{n} W_i\,(S_i + P_i)$$

Do not exceed available time per batch

$$\sum_{i=1}^{n} P_i\,y_{ij} \leq T - \delta \quad \forall j \in N$$

Perform all jobs

$$\sum_{j=1}^{n} y_{ij} = 1 \quad \forall i \in N$$

Compute starting times while avoiding overlaps(disjunctive constraints)

$$S_i \geq (j-1)(T)y_{ij} \quad \forall i \in N, \forall j \geq 2$$

$$S_i + p_i \leq S_k + R(1 - x_{ik}) \quad \forall i \in 1..n-1, k \in i+1,..n$$

$$S_k + P_k \leq S_i + Rx_{ik} \quad \forall i \in 1..n-1, k \in i+1,..n$$

Define domain for the variables

$$S_i \geq 0 \quad \forall i \in N$$

$$x_{ik} \in \{0,1\} \quad \forall (i,k) \in N^2$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in N^2$$

Objective function

$$\sum_{i=1}^{n} W_i \left(S_i + P_i\right)$$

Do not exceed available time per batch

$$\sum_{i=1}^{n} P_i \, y_{ij} \leq T - \delta \quad \forall j \in N$$

Perform all jobs

$$\sum_{j=1}^{n} y_{ij} = 1 \quad \forall i \in N$$

Compute starting times while avoiding overlaps(disjunctive constraints)

$$S_i \geq (j-1)(T) y_{ij} \quad \forall i \in N, \forall j \geq 2$$

$$S_i + p_i \leq S_k + R(1 - x_{ik}) \quad \forall i \in 1..n-1, k \in i+1,..n$$

$$S_k + P_k \leq S_i + R x_{ik} \quad \forall i \in 1..n-1, k \in i+1,..n$$

Define domain for the variables

$$S_i \geq 0 \quad \forall i \in N$$

$$x_{ik} \in \{0,1\} \quad \forall (i,k) \in N^2$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in N^2$$

Objective function

$$\sum_{i=1}^{n} W_i \left(S_i + P_i\right)$$

Do not exceed available time per batch

$$\sum_{i=1}^{n} P_i \, y_{ij} \leq T - \delta \quad \forall j \in N$$

Perform all jobs

$$\sum_{j=1}^{n} y_{ij} = 1 \quad \forall i \in N$$

Compute starting times while avoiding overlaps(disjunctive constraints)

$$S_i \geq (j-1)(T)y_{ij} \quad \forall i \in N, \forall j \geq 2$$

$$S_i + p_i \leq S_k + R(1 - x_{ik}) \quad \forall i \in 1..n-1, k \in i+1, ..n$$

$$S_k + P_k \leq S_i + Rx_{ik} \quad \forall i \in 1..n-1, k \in i+1, ..n$$

Define domain for the variables

$$S_i \geq 0 \quad \forall i \in N$$

$$x_{ik} \in \{0, 1\} \quad \forall (i, k) \in N^2$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in N^2$$

Benchmarks

First benchmark: $\delta = 0$ and fixed T

Why/When/Where:

- Preliminary tests
- Negligible maintenance duration vs T
- Fixed timetable-based maintenance

Second benchmark: $\delta = 2\% T$ and Fixed T

Why/When/Where:

- More realistic tests
- Significant maintenance duration vs T (*Approximately 2 days spent in maintenance every 3 months*)
- Fixed timetable-based maintenance

No breakdowns

Why/When/Where:

- Preliminary tests
- Perfect maintenance/machine (theoretically)

Benchmarks

First benchmark: $\delta = 0$ and fixed T

Why/When/Where:

- Preliminary tests
- Negligible maintenance duration vs T
- Fixed timetable-based maintenance

Second benchmark: $\delta = 2\% T$ and Fixed T

Why/When/Where:

- More realistic tests
- Significant maintenance duration vs T

  (*Approximately 2 days spent in maintenance every 3 months*)
- Fixed timetable-based maintenance

No breakdowns

Why/When/Where:

- Preliminary tests
- Perfect maintenance/machine (theoretically)

Benchmarks

First benchmark: $\delta = 0$ and fixed T

Why/When/Where:

- Preliminary tests
- Negligible maintenance duration vs T
- Fixed timetable-based maintenance

Second benchmark: $\delta = 2\% T$ and Fixed T

Why/When/Where:

- More realistic tests
- Significant maintenance duration vs T (*Approximately 2 days spent in maintenance every 3 months*)
- Fixed timetable-based maintenance

No breakdowns

Why/When/Where:

- Preliminary tests
- Perfect maintenance/machine (theoretically)

15 instances for $n = 2$ to 20 jobs with $P_i \in [25, 50]$, $W_i \in [1, n]$ and $T = 150$ are randomly generated .

		Time				%CPU			
	Size	average	std,dev	max	min	average	std,dev	max	min
delta = 0	10	2,44	2,55	9,07	0,50	591,20	91,35	724,00	422,00
	11	29,10	34,73	100,52	0,73	684,07	93,65	758,00	464,00
	12	100,50	189,21	731,34	4,42	721,07	34,85	766,00	665,00
	13	351,39	513,26	1200,00	5,79	718,13	29,54	759,00	654,00
	14	638,63	485,58	1200,08	27,86	735,60	28,17	770,00	696,00
	15	1084,72	265,32	1200,06	251,37	717,13	20,94	767,00	701,00
	16	999,35	403,90	1200,07	73,33	717,73	22,66	766,00	667,00
	17	1200,06	0,01	1200,07	1200,05	708,20	18,36	725,00	650,00
	18	1200,04	0,01	1200,06	1200,00	713,93	12,34	736,00	695,00
	19	1200,04	0,01	1200,06	1200,00	710,80	16,85	731,00	672,00
	20	1200,04	0,01	1200,08	1200,02	698,60	16,62	721,00	653,00
delta = 2%T	10	1,97	1,86	6,64	0,56	582,40	77,77	704,00	460,00
	11	19,23	23,65	87,99	0,66	673,33	100,69	755,00	465,00
	12	58,33	84,15	251,78	3,85	718,73	37,23	768,00	653,00
	13	338,45	520,01	1200,08	4,75	713,73	28,60	754,00	642,00
	14	596,76	521,72	1200,09	15,47	731,27	25,22	765,00	691,00
	15	977,32	382,88	1200,07	123,85	716,80	25,99	764,00	670,00
	16	1008,31	400,81	1200,07	53,36	709,20	26,16	761,00	648,00
	17	1141,74	161,61	1200,06	589,16	712,93	15,62	740,00	682,00
	18	1200,05	0,01	1200,10	1200,04	707,87	15,86	733,00	671,00
	19	1200,05	0,02	1200,09	1200,00	708,07	14,31	729,00	680,00
	20	1200,04	0,01	1200,07	1200,02	699,87	13,53	723,00	682,00

Figure 3 : Computational results of the MIP for some instances

Some properties

Property

The jobs in each batch are scheduled by Smith's rule. (smith, 1956).

Property

If batch B_j is before batch B_{j+1} then in the optimal solution we have

$$\sum_{i \in B_j} w_i \geq \sum_{k \in B_{j+1}} w_k$$

Heuristics

The MIP model takes more than 20 minutes to solve optimally instances with more than 12 jobs.

We present here 9 combination of heuristics designed to solve the big instances.

Step 1: Form a sequencing priority list. Rank the job in :

- Decreasing order of processing time P_i.
- Increasing order of processing time P_i.
- Decreasing order of $\frac{P_i}{w_i}$

Step 2: Determine a minimum number of batches. Using:
First fit, Best fit or Next fit strategy.

Step 3: Compute W_j, the sum of the weights of all the jobs T_i who belong to the same Batch j

Step 4: Rank the batches in decreasing order of W_j

Step 5: Rank the jobs in increasing order of $\frac{P_i}{w_i}$ in each batch.

Plan

Proposition

$LB1 = \sum_{j=1}^{n} P_j \sum_{i=j}^{n} W_i$ is a valid lower bound.

Proof.

The WSPT rule can obtain an optimal solution of this relaxed problem[**?**]. Let us consider f^* its optimal objective value.

We suppose that the n jobs are sorted in increasing order of the WSPT rule.

$L = \{P_1, P_2, ..., P_n\}$. So we have:

$f^* = \sum_{i=1}^{n} W_i C_i$

$= W_1 C_1 + W_2 C_2 + ... + W_n C_n$

$= W_1 P_1 + W_2 (P_1 + P_2) + W_3 (P_1 + P_2 + P_3) ... + W_n (P_1 + P_2 + P_3 + ... P_n).$

$= P_1 \sum_{i=1}^{n} W_i + P_2 \sum_{i=2}^{n} W_i + P_3 \sum_{i=3}^{n} W_i + ... + P_{n-1} \sum_{i=n-1}^{n} W_i + P_n W_n x =$
$\sum_{j=1}^{n} P_i \sum_{i=j}^{n} W_i.$ □

Plan

Comparison MIP/Heuristic

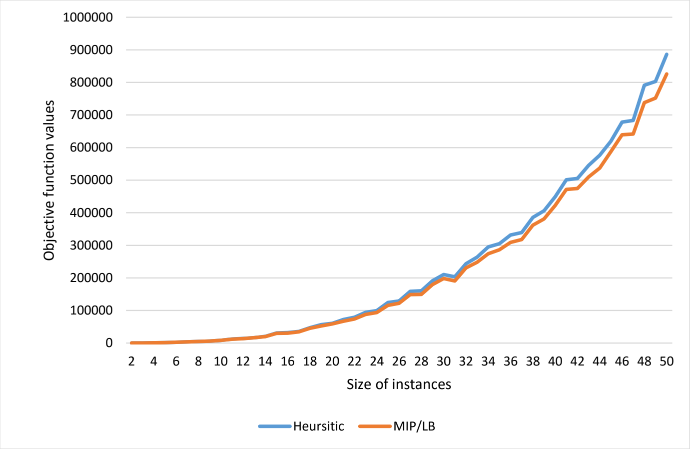


Figure 4 : Comparison of results-heuristics vs MIP

		WCT	Gap % on objective function	Gap %
Size	Seed	Delta=0	WSPTBF	L.B.
10	289	8211	0,60	-5,07
10	1517	7110	1,24	-4,42
10	5601	9518	0,28	-5,06
10	6174	7348	0,00	-3,18
10	10025	9137	0,00	-1,09
10	11606	8442	0,00	-3,51
10	11643	7024	2,51	-3,47
10	13617	6092	0,69	-3,94
10	15093	9015	7,04	-2,42
10	17396	7878	0,00	-3,26
10	20460	6618	2,13	-7,03
10	22374	9511	6,70	-4,36
10	30160	5980	2,39	-2,04
10	30778	9616	0,67	-5,37
10	31882	9452	1,49	-1,82
		Average	1,72	-3,74

Figure 5 : Computational results for 15 instances with $n = 10$

Plan

Conclusion

In this work:

- We investigate a single machine with periodic preventive maintenance
- The objective of minimizing the weighted completion time is adressed and explained
- The mathematical model is efficient on instances with size ≤ 20
- Several heurictics are compared and applicable on large instances
- A lower bound is also proposed.

Perspective

Potential direction of our future research would be :

- Add the constraint of unpredicted situation (failure, order cancellation,etc)
- Consider more than one machine in the shop
- Resolve the problem with metaheuristics

Thank you for your attention