REVISTA ELECTRO, Vol. 45 pp. 178-183, Oct 2023, Chihuahua, Chih. México
https://itchihuahua.mx/revista_electro
ISSN 1405-2172

# Optimization of a Multi-skilling Workforce Scheduling Problem with Answer Set Program

Carmen Leticia García-Mata, Alma Gabriela Rivera Parra,
Pedro Rafael Márquez-Gutiérrez and Isidro Robledo Vega
Tecnológico Nacional de México, campus Chihuahua
Chihuahua, Chih., México
{carmen.gm, pedro.mg, isidro.rv}@chihuahua.tecnm.mx
gabrielaripa18@gmail.com

## ABSTRACT.

**This paper presents an optimized scheduling solution to the multi-skilling workforce allocation problem for a real-world industry with multi-stage flow-shop production lines. In this industry, the production line in the study mainly includes manual assembly operations. The performance of the production process is highly influenced by the assignment of skilled workers to different operations executed in each workstation of the production lines. There is a set of skill requirements for each workstation of the line. The workers have developed a series of scored and certified skills. The solution is based on a single objective with preferences related to the best team for each production line. The problem was modeled and solved with the knowledge representation language Answer Set Programming and the Clingo solver. The solution is of practical utility for the industry because a better-skilled team generates products of better quality with fewer customer complaints and, consequently, minor production costs.**
**.**

**Keywords: Scheduling Optimization, Multi-skilling Workforce Scheduling Problem, Answer Set Programming.**

## 1. INTRODUCTION

As customer requirements tend to be more personalized, manufacturing production needs to change the prevailing production paradigm from a few products and high volumes to one with low-volume and high-mix products. This new production approach requires adaptability to reconfigure the production lines and reallocate the workforce to them. Besides, the industry needs to adjust its systems to Industry 4.0, integrating physical and decision aspects and human resources (Dolgui et al., 2022; Rinaldi et al., 2022). Automating the logistic process involved in the production must be considered to achieve previous goals. One of the most relevant aspects of manufacturing logistics is scheduling. Therefore, since many years ago, many efforts have been developed to solve scheduling problems, both in the academy and in practical applications (Fuchigama & Rangel, 2018). Initially, research focused on the machine scheduling problem, but it was quickly understood that human resources are essential to the production equation. The problem of how to assign workers to task execution over periods is known as the Workforce Scheduling Problem (WSP), which is a variant of the Resource-constrained Project Scheduling Problem (RCPSP). WSP differs from the workforce assignment problem in one aspect: each task-work assignment is specified in time for its execution. In some WSPs, workers are rotated at a particular time, usually daily. WSP is a complex problem due to the many constraints that must be considered.

Some typical constraints in modeling the WSP are industrial regulations, relevant agreements, personnel time availability, and their preferences. Common optimization objectives in WSP are minimizing workforce tardiness, minimizing costs, and maximizing personal satisfaction.
WSP has been studied for over fifty years, but the worker's skills were only sometimes included in the WSP modeling. When this was the case, the workers were considered specific machines, not workers with different capabilities and skills (Kletzander & Mulsliu, 2020). In other cases, human resources are treated as skilled or unskilled instead of workers with different skill levels (Bellenguez & Neron, 2007; Avramidis et al., 2010).

The probable reason that which level skills are only sometimes considered in workforce scheduling is due to the direct impact of skilled workers on performance, which remained an elusive research topic for a long time. The difficulty in proving a quantitative relation among these concepts is widely due to the inexistence of a clear, unambiguous definition for both terms. For example, in the skill concept, many dimensions are involved: expertise, education, cognitive abilities, competence, and experience. A unique definition for the performance system is also complex because the level at which performance is defined can vary widely (Gruglis & Stoyonova, 2011~~).~~

Until recently, with the adoption of more adaptive and flexible production systems, it is evident that an appropriate selection of personnel based not only on time availability but also on personnel skills and capabilities can positively impact the performance of the production systems (Liu & Liu, 2019; Cakirgirl et al., 2020). Even more, a new term has been adopted by the research community to describe the workforce scheduling problems with work skills, and it is a multi-skilling workforce scheduling problem (MSWSP) (Behrouz, 2021).

It is a fact that WSPs with skills play an increasingly important role in all types of applications, especially in the industry but also in other systems such as transport, services and health care. Also, it is expected that WSP is called in different ways depending on the application areas. In industry, WSP is also called *workforce or labor scheduling* problem. WSP for the transportation area is called *crew scheduling* or *rostering*. Applications of WSP in health care are known as *nurse scheduling* (Wongwien, 2017).

Since it is known that WSP class complexity is already an NP-hard optimization problem, the MSWSP is also NP-hard in the strong sense. Due to the high complexity of WSP, the most used approaches to solve this problem are heuristic or meta-heuristic (Musliu, 2022; Karimi-Mamaghan, 2022); on the contrary, the exact methods are less used, given the difficulty of finding an optimal solution for this kind of problems (Sivasundari et al., 2019). Different versions of the WSP have been solved using the previously mentioned methods. In some cases, even optimal solutions have been found. However, the solution of WSP continues to be a challenging task. Additionally, it is known that some techniques are highly dependent on the data, and problems exist where solutions cannot be found in reasonable time for some instances. Therefore, exploring how a general-purpose approach such as ASP can solve problems where usually dedicated approaches are used is also enjoyable.

This paper reports an optimized solution for the Multi-skilling Workforce Scheduling Problem (MSWSP) in a real-world industry. The problem is solved with Answer Set Programming (ASP) and the Clingo solver 5.6.1 running on Windows 10 in a computer with 16 Gb of RAM. ASP is a declarative language for knowledge representation and reasoning as well as for solving complex combinatorial problems (Gelfond & Lifschitz, 1988; Ostrowski, 2018). ASP has been successfully used in applications in many areas, including manufacturing, and at least one related problem to WSP is the Shift Design Problem (Abseher et al., 2015).

## 2. THE MULTI-SKILLING WORKFORCE SCHEDULING PROBLEM.

This section is dedicated to the multi-skilling workforce scheduling problem definition.

### 2.1 Problem Definition

- For this experiment, it is assumed that two production lines are running two different products—one per line.

- The set of workers is defined by *P={p1, p2,…,p20}*.
- Two parallel lines are defined by the set *L={l1, l2}*.
- The set of all skills is defined by *Sk={sk1, sk1,…,sk17}.*
    - Each line comprises eight workstations defined by *W={w1,w2,..w8}*. The production model is a flow-shop. Therefore, the production must be executed starting at time *t1* in the first workstation (1) and finishing at the last workstation (8).
    - Work execution time for each workstation and line is around 10 minutes, with slight variations of a few seconds. For problem simplification, the work execution time is rounded to 10 minutes, and each 10 minutes is represented as one unity. This simplification has the purpose of reducing the combinatorial explosion.
    - For each workstation and each line, a set of worker skills is defined, e.g., for line 1 is defined *L1S1H ={ws1, ws2,…., ws4}*, where each element is a tuple. Moreover, each *tuple={l*, st, sk} defines l as the number line, *st* as the workstation, *sk* as the skilled worker required for *st*, and similarly for line 2. For both lines, the skills required per workstation are different and can vary from 2 to 5. SL1 is the set of each of these sets of workstation skills for line 1. *SL1={l1ws1, l1ws2, …l1ws8}*. Similarly, a set SL2 is defined for line 2.
    - The skill workers are acquired through scored certification. The value scores are the natural numbers 1..10. The set of worker scores for each skill is defined by the set *SSW={{t1},{t2}…{t340}}*, where each element is a tuple. Furthermore, each *tuple={p, sk, q}*, defines *p* as the worker identifier, *sk* as the skill, and *q* as the score of *p* for *sk*. For this reason, that set of workers is 20, the set of skills is 17, and each skill has a score. *SSW* has 340 elements, one per worker, skill, and skill qualification. Then *SSW={{p1,sk1,q1}, {p1,sk2,q2}, …{p1, sk17, q17}, {p2,sk1,q1}, {p2,sk2,q2}, …{p2, sk17, q17}, … {p20,sk1,q1}, {p20,sk2,q2}, …{p20, sk17, q17}}.*
    - The possible solutions are represented by a set of tuples defined by *TW={L, S, W, J, T, Q}*, where *L* is the *line*, *S* is the *workstation*, *W* is the *worker*, *J* is the *job*, *T* is the start time in the workstation, and *Q* is the *sum of skills scores per worker at each workstation*.
    - The optimization objective is to obtain the best sets of workers per workstation line. Optimized sets are chosen based on the Q values that were previously defined.

## 3. IMPLEMENTATION OF THE MS-WSP IN ASP

This section describes how ASP is used to model the MS-WSP problem. The program was designed following the standard methodology to solve ASP problems, consisting of the Generate, Define, and Test sections.

Previously, to define the generated rule, some other rules were designed to establish the workstations' processing sequence. These rules are

```
(1) next_time(T,T+D):-time(T), duration (D).

(2) next_station(S,S+1):- station(S).

(3) shift_length(N,L-BL..L+AL)
    :- shift_length(N,L,AL,BL).

(4) shift_start((T+AT),L,BT+AT)
      :- shift_start(N,T,AT,BT),
      shift_length(N,L), 0 < L.

(5) shift_start(T1,L,C-1)
      :- shift_start(T2,L,C),
      next_time(T1,T2), 0 < C.

(6) shift_start(T,L)
      :- shift_start(T,L,C).
```

The rules `next_time, next_station, shift_length, and shift_start` are used in the rule `forward` for the sequential ordering of the workstations in time. At the same time, the rule `forward` is called from each of eight rules called `forwardS1..forwardS8`. Both rules are designed as a series of recursive rules, as shown in the next paragraph.

```
(7) forward(S0,T,S0,T)
      :- shift_start(T,L), station(S0).

(8) forward(S0,T,S2,C+1)
      :- forward(S0,T,S1,C),
      next_station(S1,S2), 1 < C.

(9) forward(S0,T,S2,C+1)
      :- forward(S0,T,S2,C),
      next_time(T1,T2), C = T2.

(10) forward(S1,C)
      :- forward(S0,T,S1,C).

(11) forwardS1(S1,C)
       :- forward(S,C), forward(S1,C1),
       S=S1, C > C1, S1=1.

(12) forwardS1L(S,C)
       :- forward(S,C), not forwardS1(S,C),
       time(T), C<=T.

(13) forwardS2(S2,C1)
      :- forwardS1L(S1,C), S1=1,
      forwardS1L(S2,C1), C1 != C+1, S2=2.

(14) forwardS2L(S,C)
      :- forwardS1L(S,C),
      not forwardS2(S,C).

(15) forwardS3(S3,C2)
      :- forwardS2L(S2,C1), S2=2,
      forwardS2L(S3,C2), C2 != C1+1, S3=3.

(16) forwardS3L(S,C)
      :- forwardS2L(S,C),
      not forwardS3(S,C).

(17) forwardS4(S4,C3)
      :- forwardS3L(S3,C2), S3=3,
      forwardS3L(S4,C3),  C3 != C2+1, S4=4.

(18) forwardS4L(S,C)
      :- forwardS3L(S,C), not
      forwardS4(S,C).

(19) forwardS5(S5,C4)
      :- forwardS4L(S4,C3), S4=4,
      forwardS4L(S5,C4),  C4 != C3+1, S5=5.

(20) forwardS5L(S,C)
      :- forwardS4L(S,C),
      not forwardS5(S,C).

(21) forwardS6(S6,C5)
      :- forwardS5L(S5,C4), S5=5,
      forwardS5L(S6,C5),  C5 != C4+1, S6=6.

(22) forwardS6L(S,C)
      :- forwardS5L(S,C),
      not forwardS6(S,C).

(23) forwardS7(S7,C6)
      :- forwardS6L(S6,C5), S6=6,
      forwardS6L(S7,C6),  C6 != C5+1, S7=7.

(24) forwardS7L(S,C)
      :- forwardS6L(S,C),
      not forwardS7(S,C).

(25) forwardS8(S8,C7)
      :- forwardS7L(S7,C6), S7=7,
      forwardS7L(S8,C7),  C7 != C6+1, S8=8.

(26) forwardS8L(S,C)
      :- forwardS7L(S,C),
      not forwardS8(S,C).
```

The Generate part was designed using the next rule:

```
(27) 1{work(L,S,W,C):workers(W)}1
      :-line(L),job(J),forwardS8L(S,C), S<=8.
```

The rule defined in (27) creates the required combination of *lines (L)*, *stations (S)*, *workers (W)*, *job (J)*, *stationOrder (C)*. The workstations are sequentially ordered previously to the generation step with the rule *forwards8L(S, C)*.

The rules (28) and (29) are used to select the best worker for the workstation. Selection is based on their scores for the skills required by each workstation line. It is crucial to clarify that the skills required for each workstation are different, and the number of skills can also differ. Also, in these rules, the specific skill scores of each worker per workstation are added up. The head of the rule is defined by `p_workL2(L, S, W, J, C, Q)`, the parameters *(L, S, W, J, C)* stand the same meaning as those defined in rule (27), and the extra parameter, *Q*, is the sum of skills scores per workstation. The body of rules (28) and (29) is built by joining each combination of `work(L,` S, W, C) and the set of facts related to the skills required per workstation line. These facts are named `s_skill(L, S, H)`, where the parameters mean *lines(L), station (S), and skill(H)* and were defined previously in the domain problem. Similar rules to the (28) and (29) are designed per workstation-line.

If the same workers are selected for multiple workstations in the same line, those solutions must be deleted because each worker can only be at one station simultaneously (rules 30 and 31). It is essential to recall that the production model is a flow shop, and consequently, more products will be assigned to each line during the work shift.

```
(28) p_workL1(L,S,W,J,C,Q)
    :- work(L,S,W,J,C), L==1, S==5,
    s_skill(L,S,H1),H1==10,
    w_skill(W,H1,G1), G1>=LI, G1<=LS,
    s_skill(L,S,H2),
    H2==15,w_skill(W,H2,G2), G2>=LI, G2<=LS,
    s_skill(L,S,H3),H3==16, w_skill(W,H3,G3),
    G3>=LI, G3<=LS, Q =G1+G2+G3.
```

```
(29)  p_workL2(L,S,W,J,C,T):-
    work(L,S,W,J,C), L==2, S==1,
    s_skill(L,S,H1), H1==8,
    w_skill(W,H1,G1), G1>=50, G1<=100,
    s_skill(L,S,H2), H2==12,
    w_skill(W,H2,G2),G2>=50, G2<=100,
    T =G1+G2.
```

```
(30) :-p_workL1(L,S,W,J,C,Q),
    p_workL2(L1,S1,W1,J1,C1,Q1), C<C1, W==W1.
```

```
(31) :-p_workL2(L,S,W,J,C,T),
    p_workL2(L1,S1,W1,J1,C1,T1), C<C1, W==W1.
```

The rules `f_workR_L1` (32) and `f_workR_L1` (33) and the constraints related (rules 34 and 35) were designed to delete those solutions having less than eight workstations per line.

```
(32) f_workR_L1(L,S,W,J,C,T):-
    p_workL1(L,S,W,J,C,T).
```

```
(33) f_workR_L2(L,S,W,J,C,T):-
    p_workL2(L,S,W,J,C,T).
```

```
(34) :-f_workR_L1(L,S,W,J,C,T)}7.
```

```
(35) :-{f_workR_L2(L,S,W,J,C,T)}7.
```

Remanent solutions per lines 1 and 2 that meet all the constraints previously detailed are joined under a unique name called `team_work`, rules (36), (37).

```
(36) team_work(L,S,W,J,T,Q)
    :- p_workR_L1(L,S,W,J,T,Q).
```

```
(37) team_work(L,S,W,J,C,T):-
    p_workR_L2(L,S,W,J,C,T).
```

When both lines are joined under a single name, it is required to delete the solutions selecting the same worker per different workstations in the same line or the same worker in different workstations and lines.

```
(38) :- team_work(L,S,W,J,T,Q),
    team_work(L1,S1,W1,J1,T1,Q1),Q<Q1,
    W==W1.
```

The objective function maximizes the work teams with the best sum of skills score (Q). The rule is:

```
#maximize {1,Q : team_work(L,S,W,J,T,Q)}.
```

## 4. EXPERIMENTS AND DISCUSSION OF RESULTS

The solution was tested with data from the real-world manufacturing industry. It was impossible to obtain optimal solutions using the workers' actual skill scores because the personnel was not qualified enough in all the required skills for each workstation. Then, it was necessary to use artificially modified skills scores where the workers are correctly qualified to test if it is possible to get an optimized solution for the problem. When the score skills improve, the optimal solution is obtained in 256 microseconds, consisting of 8 optimal Answer Sets. The first Answer Set corresponds to the teamwork for both lines and is represented by the predicate `team_work(L, S, W, J, T, Q)`, where `L`=line, `S`=station, `W`=worker, `J`=job, `T`=start-time of `J` in `W` and `L`; and `Q` is the sum of score skills for worker `W` in the specified workstation-line. The first optimal solution is shown as displayed by the solver Clingo.

```
Answer: 1
team_work(1,1,18,1,0,380)
team_work(1,2,16,1,1,200)
team_work(1,3,1,1,2,400)
team_work(1,4,2,1,3,300)
team_work(1,5,19,1,4,300)
team_work(1,6,12,1,5,200)
team_work(1,7,11,1,6,200)
team_work(1,8,20,1,7,300)
```

```
team_work(2,1,17,1,0,120)
team_work(2,2,4,1,1,150)
team_work(2,3,3,1,2,190)
team_work(2,4,8,1,3,130)
team_work(2,5,5,1,4,200)
team_work(2,6,13,1,5,50)
team_work(2,7,9,1,6,150)
team_work(2,8,7,1,7,200)
```

For comparative questions, the following tables comprise six of the optimal solutions obtained by the solver. For space considerations, separated tables were created for the three first Answer Sets, one table per line. Then, the sum of Answer Sets 1 to 3 scores is shown in the following table. The respective tables for the Answers Sets 4 to 6, and their sums are displayed ahead of the ones corresponding to optimal solutions 1 to 3.

| **Line: 1, Job: 1** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Answer: 1** | | **Answer: 2** | | **Answer: 3** | |
| **S** | **Ti** | **W** | **Tot** | **W** | **Tot** | **W** | **Tot** |
| 1 | 0 | 18 | 380 | 18 | 380 | 1 | 400 |
| 2 | 10 | 16 | 200 | 16 | 200 | 16 | 200 |
| 3 | 20 | 1 | 400 | 1 | 400 | 18 | 370 |
| 4 | 30 | 2 | 300 | 2 | 300 | 2 | 300 |
| 5 | 40 | 19 | 300 | 20 | 300 | 12 | 300 |
| 6 | 50 | 12 | 200 | 12 | 200 | 20 | 180 |
| 7 | 60 | 11 | 200 | 11 | 200 | 11 | 200 |
| 8 | 70 | 20 | 300 | 19 | 280 | 19 | 280 |
| SUM: | | | 2280 | | 2260 | | 2230 |

Table 1. Optimal solutions for line 1 in Answer Sets 1 to 3 related score skills per workstation-line for the worker (W) selected. See columns named **Tot**.

| **Line: 2, Job: 1** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Answer: 1** | | **Answer: 2** | | **Answer: 3** | |
| **S** | **Ti** | **W** | **Tot** | **W** | **Tot** | **W** | **Tot** |
| 1 | 0 | 17 | 120 | 17 | 120 | 17 | 120 |
| 2 | 10 | 4 | 150 | 4 | 150 | 4 | 150 |
| 3 | 20 | 3 | 190 | 3 | 190 | 3 | 190 |
| 4 | 30 | 8 | 130 | 8 | 130 | 8 | 130 |
| 5 | 40 | 5 | 200 | 5 | 200 | 5 | 200 |
| 6 | 50 | 13 | 50 | 13 | 50 | 13 | 50 |
| 7 | 60 | 9 | 150 | 9 | 150 | 9 | 150 |
| 8 | 70 | 7 | 200 | 7 | 200 | 7 | 200 |
| | | | 1190 | | 1190 | | 1190 |

Table 2. Optimal solutions for line 2 in Answer Sets 1 to 3 related score skills per workstation-line for the worker (W) selected. See columns named **Tot**.

| | **Answer: 1** | **Answer: 2** | **Answer: 3** |
|---|---|---|---|
| **L1, Total Score** | 2280 | 2260 | 2230 |
| **L2, Total Score** | 1190 | 1190 | 1190 |
| **L1+L2** | **3470** | **3450** | **3420** |

Table 3. Optimal solutions for lines 1 and 2 regarding the Answer Sets 1 to 3 related score skills per workstation-line for the worker selected.

Tables 4, 5, and 6 are the ones related to the Optimal Solutions of the Answer Sets 3 to 6

| **Line: 1, Job: 1** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Answer: 4** | | **Answer: 5** | | **Answer: 6** | |
| **S** | **Ti** | **W** | **Tot** | **W** | **Tot** | **W** | **Tot** |
| 1 | 0 | 1 | 400 | 19 | 380 | 1 | 400 |
| 2 | 10 | 16 | 200 | 1 | 200 | 5 | 160 |
| 3 | 20 | 19 | 380 | 2 | 400 | 13 | 370 |
| 4 | 30 | 2 | 300 | 20 | 280 | 9 | 230 |
| 5 | 40 | 12 | 300 | 7 | 220 | 7 | 220 |
| 6 | 50 | 18 | 180 | 18 | 180 | 8 | 160 |
| 7 | 60 | 11 | 200 | 11 | 200 | 17 | 170 |
| 8 | 70 | 7 | 230 | 12 | 300 | 20 | 300 |
| | | | 2190 | | 2160 | | 2010 |

Table 4. Optimal solutions for line 1 in Answer Sets 4 to 6 related score skills per workstation-line for the worker (W) selected. See columns named **Tot**.

| **Line: 2, Job: 1** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Answer: 4** | | **Answer: 5** | | **Answer: 6** | |
| **S** | **Ti** | **W** | **Tot** | **W** | **Tot** | **W** | **Tot** |
| 1 | 0 | 17 | 120 | 17 | 120 | 4 | 140 |
| 2 | 10 | 4 | 150 | 4 | 150 | 3 | 120 |
| 3 | 20 | 3 | 190 | 3 | 190 | 18 | 250 |
| 4 | 30 | 8 | 130 | 8 | 130 | 6 | 150 |
| 5 | 40 | 5 | 200 | 5 | 200 | 12 | 250 |
| 6 | 50 | 13 | 50 | 13 | 50 | 10 | 50 |
| 7 | 60 | 9 | 150 | 9 | 150 | 19 | 130 |
| 8 | 70 | 20 | 250 | 16 | 240 | 16 | 240 |
| | | | 1240 | | 1230 | | 1330 |

Table 5. Optimal solutions for line 2 in Answer Sets 4 to 6 related score skills per workstation-line for the worker (W) selected. See columns named **Tot**.

| | **Answer: 4** | **Answer: 5** | **Answer: 6** |
|---|---|---|---|
| **L1, Total Score** | 2190 | 2160 | 2010 |
| **L2, Total Score** | 1240 | 1230 | 1330 |
| **L1+L2** | **3430** | **3390** | **3340** |

Table 6. Optimal solutions for lines 1 and 2 regarding the Answer Sets 4 to 6 related score skills per workstation-line for the selected worker.

Analyzing the optimal results obtained by the solver and shown in the tables, it is possible to notice that the workers selected for lines 1 and 2 of each Answer Set are different, an essential requirement that the solver must meet the resulting plan. Also, as usual, the best solutions are those appearing first. The best results are those of the first solution, and the worst ones are those of solution 6. It is crucial to clarify that there is no need to build worker teams for the rest of the day because the production lines work for some days with the same product. Only when the products being produced are changed it is necessary to reschedule again according to the new production requirements.

## 5. CONCLUSIONS

While it is true that the MS-WSP is an NP-hard problem and commonly heuristics or meta-heuristics approaches are used for its solution, this research has shown that simplified versions of the multi-skilling workforce scheduling problem can be efficiently and optimally solved using Answer Set Programming. The problem simplification related to processing time by workers does not represent an issue for this solution because the processing times differ from each other only in seconds, and the processing time per workstation is around 10 minutes. Then, a difference in seconds is irrelevant, and the processing times per workstation can be rounded to 10 minutes without complications.

As it is in the case study, the personnel are underqualified. Therefore, an optimal solution cannot be found for the actual data because the workers' skill constraints still need to be fulfilled. Instead, the solution can be used for training advice with light modifications.

Additionally, as the MS-WSP problem results relevant to improving productivity and earnings in the industry and many other areas, it is important to continue researching and solving practical problems occurring in the real world. In particular, as future work, the problem indexed in this research can be extended to consider more production lines with different products and, in general, using different data sets.

## 6. REFERENCES

1. Abseher, M., Gebser, M., Musliu, N., Schaub, T., Woltran, S. (2015) Shift design with answer set programming. In: Calimeri, F., Ianni, G., Truszczynski, M. (eds) Logic Programming and Nonmonotonic Reasoning. LPNMR 2015. Lecture Notes in Computer Science(), vol 9345. Springer, Cham. https://doi.org/10.1007/978-3-319-23264-5_4.
2. Avramidis, N. A., Chan, W., Gendreau, M., L'Ecuyer, P., & Pisacane, O. (2010). Optimizing daily agent scheduling in a multi-skill call center. *European Journal of Operational Research*, *200*(3), 822–832.
3. Behrouz, A.N. (2021). Multi-skilling in scheduling problems: A review on models, methods and applications, Computers & Industrial Engineering, Volume 151,107004, ISSN 0360-8352, https://doi.org/10.1016/j.cie.2020.107004.
4. Bellenguez, M. O., & Neron, E. (2007). A branch-and-bound method for solving multi-skill project scheduling problems. *RAIRO Operations Research*, *41*, 155–170.
5. Cakirgil, S.,_Kuyzu,E.,_Kuyzu,G. (2020) An integrated solution approach for multi-objective, multi-skill workforce scheduling and routing problems. Computers and Operations Research, Vol. 118 (2020), 104908 https://doi.org/10.1016/j.cor.2020.104908.
6. Dolgui, A., Sgarbosa, F., Simonetto, M. (2022). Design and management of assembly systems 4.0: systematic literature review and research agenda. International Journal of Production Research. Vol. 60 (1), págs. 184–210, (2022). doi :10.1080/00207543.2021.1990433.
7. Firat, M. and Hurkens, C.A.J. (2012). An improved MIP-based approach for a multi-skill workforce scheduling problem. Journal of Scheduling, Vol. 15, Issue 3, pages 363–380. DOI: 10.1007/s10951-011-0245-x.
8. Fuchigami, H., Rangel, S. (2018). A survey of case studies in production scheduling: Analisis and perspectives. Journal of Computational Science, Vol. 25 (1). Doi: 10.1016/j.jocs.2017.06.004.
9. Gelfond, M., Lifschitz, V. (1988). The stable model semantics for logic programming. Fifth International Conference on Logic Programming: Available at https://www.researchgate.net/publication/2408055.
10. Grugulis, I., & Stoyanova, D. (2011). Skill and performance. *British Journal of Industrial Relations*, *49*(3), 515–536. https://doi.org/10.1111/j.1467-8543.2010.00779.x.
11. Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, M., Talbi, E.G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. European Journal of Operational Research. Vol.296, Issue 2, Pages 393–422, https://doi.org/10.1016/j.ejor.2021.04.032.
12. Kletzander, L., Mulsliu, N., (2020). Solving the general employee scheduling problem. Computers and Operations Research. Vol. 113. Issue C. https://doi.org/10.1016/j.cor.2019.104794.
13. Liu, Z., Liu, J., Zhuang, C., Wan, F. (2022). Multi-objective complex product assembly scheduling problem considering the parallel team and worker skills. Journal of Manufacturing Systems, Volume 63, Pages 454–470.
14. Musliu, N. Heuristic Methods for Automatic Rotating Workforce Scheduling (2022) International Journal of Computational Intelligence Research, Vol. 2, Issue 4, pages 973-1873. DOI: 10.5019/j.ijcir.2006.69.
15. Ostrowski, M. (2018). Modern constraint answer set solving. Ph.D. thesis, University of Potsdam.
16. Ouelhadj, D. and Petrovic, S.: A Survey of dynamic scheduling in manufacturing systems. Journal of Scheduling, 12(4), 4l7-431 (2009).
17. Prunet, T., Absi, N., Borodin, V., Cattaruzza, D. (2022). Optimization of Human-Aware Manufacturing and Logistics Systems: A Comprehensive Review of Modeling Approaches and Applications. hal-03788101.
18. Rinaldi, M., Fera, M., Bottani, E., Grosse, E.(2022). Workforce scheduling incorporating worker skills and ergonomic constraints. Computers & Industrial Engineering, Vol 168 (2023), 108107. https://doi.org/10.1016/j.cie.2022.108107.
19. Sivasundari, M.,_Rao, K.S., Raju, R. (2019) Production, Capacity and Workforce Planning: A Mathematical Model Approach. Applied Mathematics & Information Sciences - An International Journal. Vol. 13, Issue 3. Pages 368–382. DOI: 10.18576/amis/130309.
20. Wang, H.; Alidaee, B.; Ortiz, J.; Wang, W. (2021) The multi-skilled multi-period workforce assignment problem. International Journal of Production Research. Vol. 59, Issue 18, pages 5477–5494. doi = 10.1080/00207543.2020.1783009.
21. Wongwien. T. (2017) Multi-objective workforce scheduling with combined safety, productivity, and job satisfaction consideration. Ph.D. thesis. International Institute of Technology. Thammasat University.