10.1515/acsc-2015-0023

Archives of Control Sciences
Volume 25(LXI), 2015
No. 3*, pages 355–365*

A Two-stage approach for an optimum solution of the car assembly scheduling problem Part 1. Problem statement, solution outline and tutorial example

MICHAŁ MAZUR and ANTONI NIEDERLIŃSKI

A new approach to solving realistic car assembly scheduling problems for mixed model assembly line is presented. It is proposed to decompose the problem into two subproblems: 1) a sequencing problem that generates admissible car sequences fulfilling capacity constraints for all car models in the production plan, 2) a scheduling problem that determines an admissible car sequence with shortest makespan. The details of this approach are illustrated by a simple numerical example.

Key words: car sequencing, car assembly scheduling, workstation capacity constraints.

1. State of the Art

The modern car assembly line is usually attributed to Henry Ford, who perfected some earlier attempts by installing (in 1913) a driven conveyor belt that enabled the production of the famous *Model T* in 93 minutes. As Henry Ford himself confessed, "*The idea came in a general way from the overhead trolley that the Chicago packers use in dressing beef*", (see [4]). A gradual, long, common sense and technology driven development lead directly from the black coloured Model T assembly line to the contemporary highly sophisticated, extremely combinatorial-complex multi-coloured multi-option assembly lines. A large number of simplified academic problem has been distilled over the years (see e.g. [5] and [1]) from the practice of assembly car scheduling, which was and still is - as a rule - covered by clouds of industrial secrecy and the obvious fact that car industry is for profit, not for disseminating and refining knowledge. Hence the scarcity of academic publications dealing with real-world car assembly problems. Some of the existing literature reflects the heavy dependence of the scheduling practice upon various types of heuristics, present in internal factory regulations, see e.g. [7]. Others are

---

M. Mazur is with General Motors Manufacturing Poland, Gliwice. A. Niederliński, the corresponding author, is with University of Economics in Katowice, Poland. e-mail: antoni.niederlinski@ae.katowice.pl

Received 28.06.2015.

attempts at more precise scheduling solutions, usually at the expense of relaxing some nasty constraints which makes the problem manageable by conventional optimization techniques, see e.g. also [7]. A notable development has however to be noticed: it is the emergency and development of *Constraint Logic Programming* (CLP) techniques, with its promise to deal with any constraints in a declarative manner with due respect to the limitations posed by combinatorial explosion, see e.g. [3] and [6], as well as *Constraint Programming* (CP) techniques, see [2]. The solution of the car assembly scheduling problem is illustrated by a simple tutorial example in Part 1 of this paper. It is presented with technical details in Part 2 of this paper for a real-world example using proprietary CLP tools.

2. Introduction

This paper proposes a method of determining the optimal car assembly line schedule in automotive industries using *mixed model assembly lines*. A car assembly line is a manufacturing process in which parts are added to car bodies as they are transported from workstation to workstation until the completed cars are leaving the last workstation.

It is assumed that the assembly line, defined as sequence of workstations corresponding to the approved assembly technology, is given. That means it specifies what can be done on any workstation and how long it will take. So the sequence of stations in the assembly line corresponds to the sequence in which assembly operations should be performed. It is assumed that the physical length of all works stations is the same. In a car assembly line, car bodies are moving on conveyors through different workstations, each specialized for a particular job, such as installing the engine, installing the power seats, installing wheels etc. Car bodies can't change their position in the sequence after feed into the line, so the FIFO rule holds while generating admissible sequences. For each car body entering a workstation, a crew of operators from that station moves with the car body while performing their jobs. The *shift poduction plan* consists of a set of m car orders specifying all options for the cars to be produced on this shift. The conteporary motor car industry aimes at satisfying a broad range of customer requirements: automakers rely on a built-to-order production system, instead of a built-to-inventory perspective. So nowadays it may happen that each car on a shift is produced with a different set of options, although more often the cars may be grouped into sets S_i $(i = 1,..,v)$ having the same options O_i, like navigation system, rear-view video camera, glass roof, four wheel drive. So for each set of cars to be produced according to the production plan, an individual assembly technology is needed. The technology specifies operations performed for the car on any work station as well as the *Assembly Times* needed to perform those operations for any car body and any workstation. This proves to be a challenge to the entire manufacturing process including the assembly line.

The assembly line is moving with constant speed. The quotient of workstation length by assembly line speed is known as *Tact Time* of the assembly line. It is the time interval between the instant any car body is entering a workstation and the instant it is leaving it, provided no line stoppage occured in between. Practically Tact Times are different for different automakers and usually may correspond to values from 70 to 120 seconds. For particular workstations and car bodies the Assembly Times may be longer than the Tact Time, which may lead to line stoppages. The stoppages are time (i.e. money) looses that unfortunately cannot always be eliminated, even for optimum schedules.

The production plan has to be transformed into a sequence of car bodies fed into the assembly line. For m car bodies there exist $m!$ such a sequences. This may be quite a very large number, but fortunately, the sequences must fulfill a number of capacity constraints discussed below. Sequences fulfilling these constraints will be referred to as *admissible sequences*. The number of admissible sequences is considerably smaller than $m!$ although still rather large.

The main contribution of the paper consists in decomposing the generation of the optimal car assembly schedule into two consecutively solved subproblems:

1. Generating admissible sequences that fulfill work station capacity constraints for all car orders from the shift production plan. The admissible sequences generated so far result as a rule in different makespan times.

2. Minimizing the overall makespan by determining an admissible sequence with the shortest makespan. Minimizing the makespan is the most reasonable performance index from the manpower/equipment utilization point of view.

3. Fulfilling workstation capacity constraints

A mixed-model assembly line is designed to service a limited number of car bodies with different options. The speed of the assembly line is such as to allow the crews to finish their jobs while the car bodies are in their stations. E.g. if the installation of power seats takes 16 minutes and a new car body enters the assembly line every 4 minutes, then (assuming that each car needs a power seats), the station for power seats installation needs a capacity to handle $16/4 = 4$ car bodies, i.e. it has to be staffed by 4 power seats handling crews. However, because not each car requires a power seat, in order to save instrumentation and labour, the capacity of the power seats station may be smaller, e.g. the station may have only 3 crews to handle power seats. That means the station can cope with no more than 3 cars requiring power seats out of any sequence of 4 cars. In shorthand - the power seats station has a *capacity constraint* $3/4$. Now its up to the assembly line scheduling program to assure that the entire sequence of car bodies feed into the assembly line has no 4-bodies subsequences with more than 3 bodies requiring power seats.

Generally, any workstation of a mixed-model assembly line is designed to service at most p_i cars with option O_i in a subsequence of q_i cars. To characterised this feature the term *workstation capacity constraint* dedicated for option O_i is defined as:

$$WCCO_i = \frac{p_i}{q_i}$$

So if for some particular option $WCCO_i = 3/4$, any subsequence of `4` car bodies in any admissible sequence may include no more than `3` car bodies with that option. The number of options follows from the production plan. Each option O_i is associated with its dedicated capacity constraint $\frac{p_i}{q_i}$. The solution of car sequencing problem is any sequence of cars feed into assembly line with all constraints fulfilled. Any sequence fulfilling all workstation capacity constraints is called *admissible sequence* and stored for further processing. The remaining inadmissible sequences are rejected.

4. Minimizing the makespan

Admissible sequences generated above usually correspond to different *makespans* defined as the overall times from starting the assembly at the first station to finishing the assembly at the last station, for any car. The makespan is the most important performance index of the assembly process. Makespan minimization is an objective of the second stage of finding optimal assembly schedule. Scheduling involves the simulation of all admissible sequences of car bodies on assembly line in order to get their makespans. An optimal schedule is the admissible sequence with the minimum makespan.

Assembly line is moving with constant speed. On each workstation car bodies are assembled throughout the Tact Time, and sometimes longer. If until end of Tact Time any assembly operation is not finished, then the line is stopped for a time necessary to terminate the operation. It should be noticed that eliminating line stoppages by increasing Tact Time does not necessarily lead to shorter maksepans: therefore minimizing makespans must be the dominant performance index.

5. A tutorial example

The complexity of the proposed methodology and the intricacy of its details is difficult to grasp from generalities only. It has to be supported by some tutorial example, small enough to have the advantage of being intuitively understandable and giving an insight into all relevant concepts. The example initial data is limited to a small production plan, a table of workstation capacity constraints for all options (`WCCO`) and a table of Assembly Times for all workstations and all car types.

5.1. Determining admissible car body sequences

The investigated production plan consists of `4` cars to be produced in `4` groups of cars (`A, B, C and D`), with two different options (Option O_1, Option O_2). A production plan for 4 cars (if all cars had different options) results in 4! = 24 car body sequences, which is a number small enough for understanding all relevant concepts, listening all admissible sequences, comparing them and understanding the optimization problem. It is assumed that capacity constraints for both options are given by Tab. 5.

Table 5: Workstation capacity constraints for tutorial example

`Car type`	`A`	`B`	`C`	`D`	$WCCO_i$
Option O_1	√	√	-	-	1/2
Option O_2	-	√	√	√	2/3
Number of cars in production plan	1	1	1	1	

The meaning of this table is as follows: e.g. a single `B` car should be produced with option O_1 for which $WCCO_1 = 1/2$, and option O_2 which $WCCO_2 = 2/3$.

5.2. Sequencing results

The extremely small (by car industry standards) size of this example allows to generate all `24 = 4!` sequences and test each of them for the fulfilling of the capacity constraints. The results are presented in Tab. 6.

All data in Tab. 6 with the exception of the *Makespan* column data has been generated using data from Tab. 5 only. The meaning of this table is as follows: e.g. the sequence `21` is an admissible sequence, the order of car bodies on assembly line (1,2,3,4 for `C`, `A`, `D`, `B`) means that the first car body fed into the line is `C`, the next one `A` followed by car body `D` and the last one being `B`, from the left end of the sequence to the right end.

Obviously, for any sequence to be admissible, the car bodies `A` and `B` should not be adjacent in any order ($WCCO_1$ constraint) and the car bodies `B`, `C` and `D` should neither be adjacent in any order ($WCCO_2$ constraint).

5.3. Scheduling constraints

The size of this example makes the makespan optimization problem solvable by inspection. However, some additional data is needed: it is assumed that there are `3` workstations on the assembly line. Any car body of the sequence is moving through workstations `1`, `2`, and `3` in this order: this is the approved assembly technology that satisfies all the remaining technological constraints different from capacity constraints. To each car body and each workstation a value of Assembly Time, measured in *Time Units* (TU), has been attributed as shown in Tab. 7.

Table 6: All possible sequences with makespans and $WCCO_i$ data.

`Sequence` of cars	Order of cars 1	2	3	4	Makespan	$WCCO_1$ fulfilled?	$WCCO_2$ fullfilled?
1, inadmissible	`A`	`B`	`C`	`D`	15	no	no
2, inadmissible	`A`	`B`	`D`	`C`	17	no	no
3, inadmissible	`A`	`D`	`B`	`C`	15	**yes**	no
4, inadmissible	`A`	`D`	`C`	`B`	17	**yes**	no
5, inadmissible	`A`	`C`	`D`	`B`	16	**yes**	no
6, inadmissible	`A`	`C`	`B`	`D`	16	**yes**	no
7, admissible	`B`	`C`	`A`	`D`	15	**yes**	**yes**
8, inadmissible	`B`	`C`	`D`	`A`	15	**yes**	no
9, admissible	`B`	`D`	`A`	`C`	16	**yes**	**yes**
10, inadmissible	`B`	`D`	`C`	`A`	17	**yes**	no
11, inadmissible	`B`	`A`	`C`	`D`	16	no	**yes**
12, inadmissible	`B`	`A`	`D`	`C`	17	no	**yes**
13, admissible	`D`	`A`	`C`	`B`	16	**yes**	**yes**
14, inadmissible	`D`	`A`	`B`	`C`	15	no	**yes**
15, inadmissible	`D`	`C`	`A`	`B`	17	no	**yes**
16, inadmissible	`D`	`C`	`B`	`A`	15	**yes**	no
17, inadmissible	`D`	`B`	`A`	`C`	16	no	**yes**
18, inadmissible	`D`	`B`	`C`	`A`	15	**yes**	no
19, inadmissible	`C`	`D`	`A`	`B`	17	no	**yes**
20, inadmissible	`C`	`D`	`B`	`A`	17	no	no
21, admissible	`C`	`A`	`D`	`B`	17	**yes**	**yes**
22, inadmissible	`C`	`A`	`B`	`D`	17	no	**yes**
23, inadmissible	`C`	`B`	`A`	`D`	17	no	**yes**
24, inadmissible	`C`	`B`	`D`	`A`	17	**yes**	no

The sequences generated are illustrated by Gantt charts, which are graphical representations of car bodies allocation over Time Units for concurrently performed operations on all workstations. The general outline of Gantt charts presenting assembly line operations is shown in Fig. 1. It differs from Gantt charts used to present job-shop activities because it presents movements of car bodies not only in time, but across workstations as well.

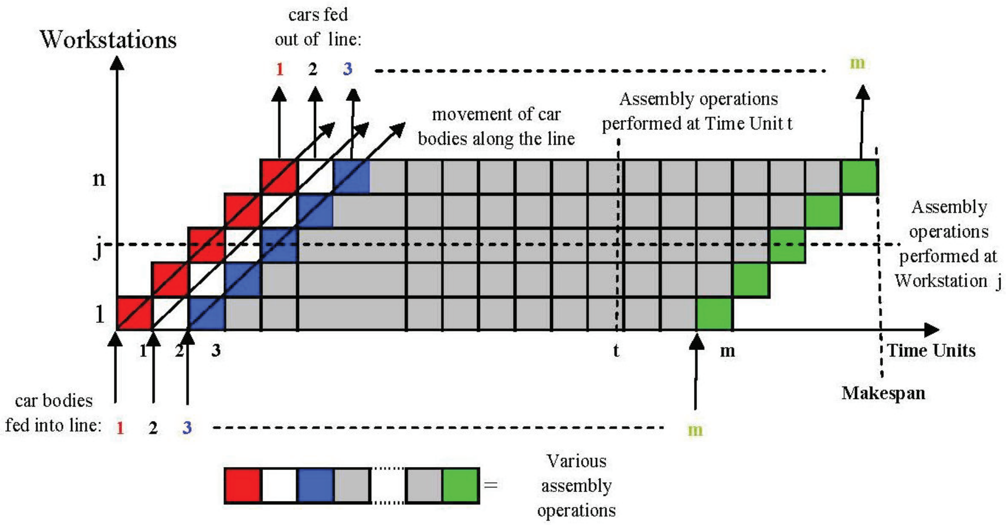

Figure 1: Basic structure of Gantt charts for assembly line operations

The Tact Time is assumed to be 2 TU, i.e. it is smaller than the maximum Assembly Time for some operation (equal 4 TU). Therefore in order to perform all assembly operations line stoppages are unavoidable. For this Tact Time and data from Tab. 7, thanks to the small size of the problem all 24 Gantt charts for all sequences are generated, as well as their corresponding makespan may be calculated. Results are shown in Figs 2 and 3. The makespans were additionally displayed in Tab. 6.

Table 7: Assembly Times (in TU) for example.

Car type	A	B	C	D
Workstation 1	1	1	2	1
Workstation 2	2	2	4	1
Workstation 3	3	4	2	1

Thus, for admissible sequence 21, the assembly starts with body C entering Workstation 1 and remaining there for 2 TU. It is next moved to Workstation 2, where it is dealt with for 4 TU, with the necessary line stoppage for the additional 2 TU. At the beginning of TU=3, car body A is entering Workstation 1, where it is processed for a singly TU. At the beginning of `TU=6`, car body `C` is moving to Workstation `3`, at the same time car body `A` is entering Workstation `2`, and the new car body `D` is entering the line at Workstation `1`. None of the operation performed at any workstation over `TU=6` and `7` takes longer than the Tact Time. At the beginning of `TU=8` the assembling of car body `C` is finished and car `C` leaves the assembly line, while care body `A` moves to Workstation `3` where it has to stay for `3` Time Units, thus causing another line stoppage. At the same time car body `D` moves from Workstation `1` to Workstation `2` and a new car body `B` is entering Workstation 1. The Assembly Times for `D` and `B` are well below the time that must be allocated for processing care body `A` in Workstation `3`. At the beginning of `TU=11` the assembling for car body `A` is finished, the car is moved out of Workstation `3`, and car bodies B and D move to their next workstations. For both car body `B` in Workstation `2` and car body `D` in Workstation `3` no line stoppages are needed. The assembling of car body `C` is finished at the begining of `TU=8`, and the car is moved out of Workstation `3`. However, with car body `B` moving to Workstation `3`, another stoppage occurs because the Assembly Time for car body `B` in Workstation `3` is equal `4` Time Units. The assembling of last car body `B` ends at the beginning of `TU=17` with moving this body out of Workstation 3, making the makespan equal to `17`.

The beginnings of underscored Time Units are corresponding to time instances at which car bodies move to their next workstations. Visual inspection of Gantt charts from Figs 2 and 3 shows that:

1. All sequences are plagued by line stoppages. However, increasing the Tact Time to avoid them does not necessarily lead to shorter makespans.
2. There is a single optimum sequence `7` (`B`,`C`,`A`,`D`) with makespan equal `15`.

6. Summary

A two stage decomposition for determining minimum makespan schedules for mixed model car assembly lines was proposed. The first stage consists of determining all admissible car sequences, i.e. car sequences fulfilling the workstation capacity constraints. The second stage consists in determinig the admissible car sequence with minimum makespan. The results of its application for a simple assembling line were presented by a tutorial example using tables and Gantt charts. The details of its solution using a *Constraint Logic Programming* approach will be shown in Part 2 of this paper.

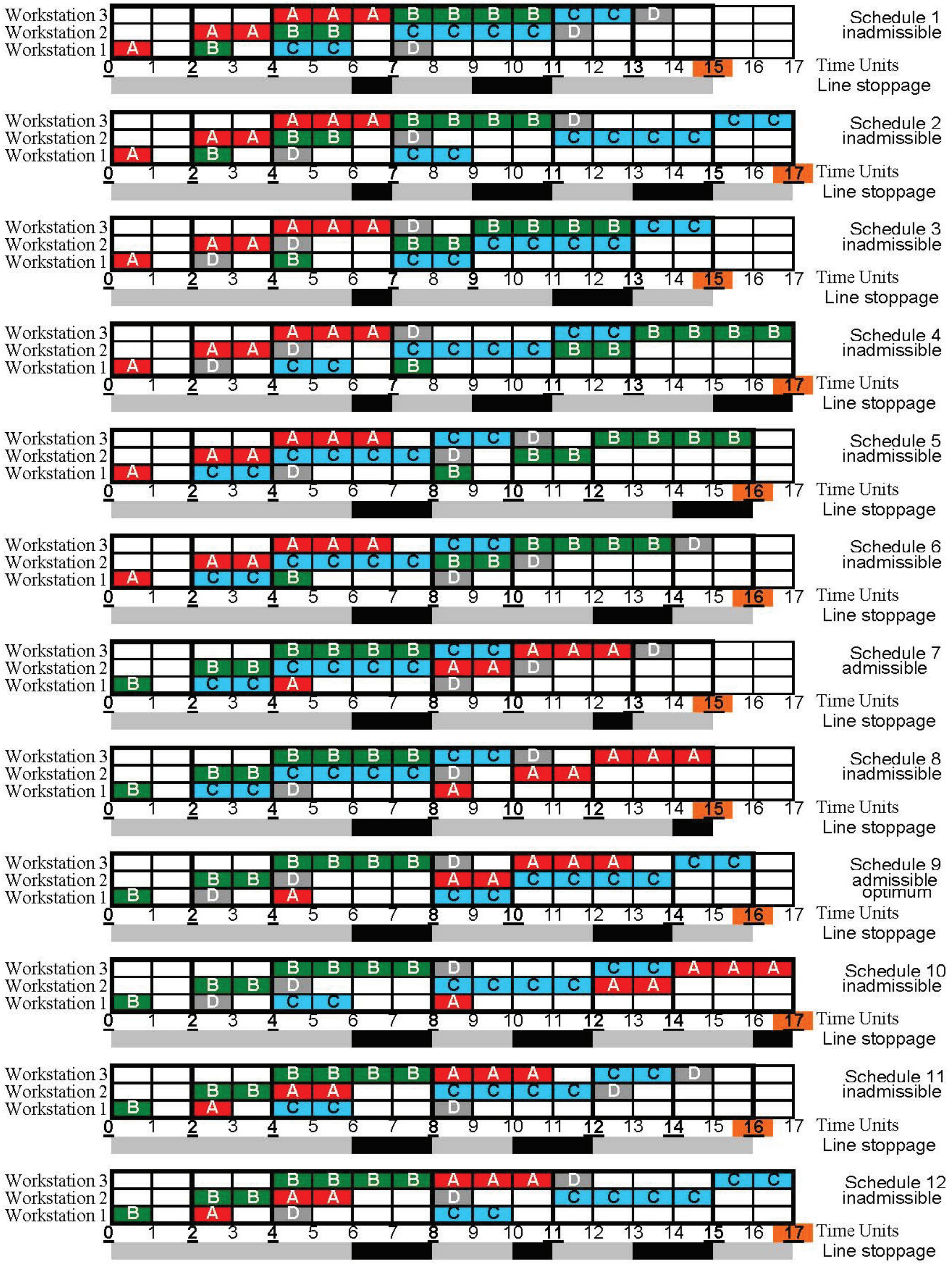

Figure 2: Gantt charts for the first 12 sequences

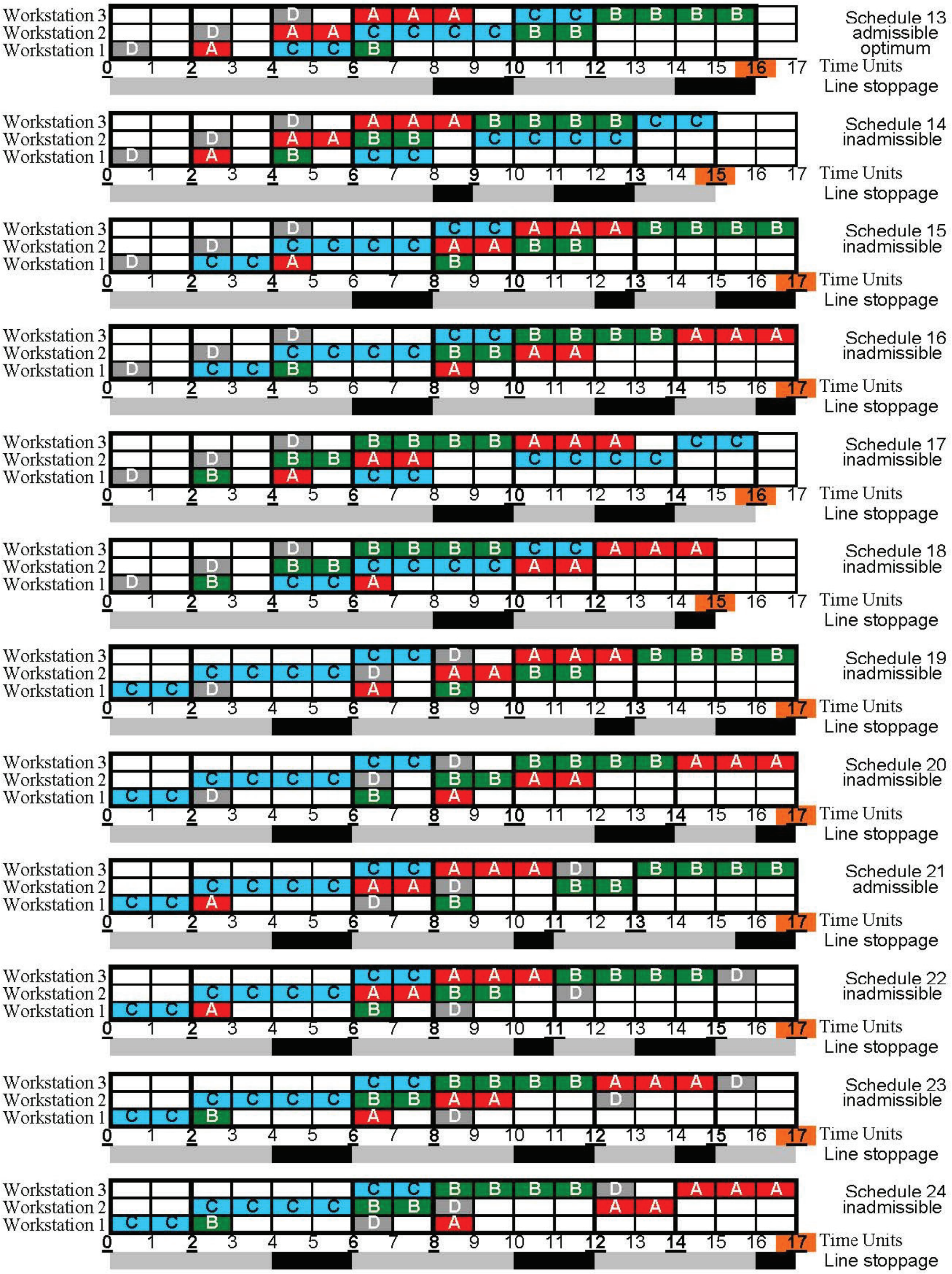

Figure 3: Gantt charts for the last 12 sequences

References

[1] K.R. BAKER and D. TRIETSVCH: Principles of Sequencing and Scheduling. John Wiley and Sons, Hoboken, 2009.

[2] PH. BAPTISTE, C. LE PAPE and W. NUIJTEN: Constraint-Based Scheduling. Kluwer Academic Publishers, Boston, 2001.

[3] M. DINCBAS, H. SIMONIS and P. VAN HENTENRYCK: Solving the car-sequencing problem in constraint logic programming. *Proc. of the ECAI*, (1988), 290-295.

[4] H. FORD and S. CROWTHER: Mein Leben und Werk. Unter Mitwirkung von Samuel Crowther. Leipzig, 1923.

[5] S. FRENCH: Sequencing and Scheduling: An Introduction to the Mathematics of Job-Shop. Ellis Horwood, 1982.

[6] A. JOLY and Y. FREIN: Heuristics for an industrial car sequencing problem considering paint and assembly shop objectives. *Computers & Industrial Engineering*, **55** (2008), 295-310.

[7] C. SOLNON, V. CUNG, A. NGUYEN and C. ARTIGUES: The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. *European J. of Operational Research*, **191** (2008), 912-927.