



Nash equilibrium as a tool for the Car Sequencing Problem 4.0

Sara Bysko¹ · Jolanta Krystek¹ · Andrzej Świerniak¹

Received: 30 July 2020 / Accepted: 12 January 2023 / Published online: 26 February 2023
© The Author(s) 2023

Abstract

This paper introduces a new concept to solve car sequencing problem called the Car Sequencing Problem 4.0, focuses the paint shop. The problem of effective car sequencing in the paint shop is caused by the specifics of the production process itself and the structure of the production line. Sequencing of cars as required by the painting process is justified economically. The main goal is to minimize the number of costly changeovers of the painting guns because of color changes and to synchronize those with periodic cleanings, forced by technological requirements. For this purpose, a buffer located in the paint shop is applied. In this paper a game theoretic framework is presented to analyze the problem. Three games are introduced: Buffer Slot Assignment Game–Buffer-OutShuttle Game called the BSAG-BOSG, In–Out Shuttle Game and its modification called modified In–Out Shuttle Game. Based on the simulations performed the efficiency of the algorithms is verified using several datasets.

Keywords Car production · Sequencing · Car sequencing problem · Game theory · Nash equilibrium

Introduction

The production model *Make to Order* is widely used in modern manufacturing plants. Its use is most often motivated by a variety of customer requirements. In such plants, production lines are flexible and produce a variety of products produced at many stages of production. Due to the fact that the processing time of different types of products varies, storage is organized between each stage. This is to guarantee both continuity of production and the possibility of changing the order of production.

This paper considers a special case; car production. It consists of four main stages (Fig. 1).

Car production begins at the press shop, where, first of all, sheet metal is cut, followed by pressing of the individual components of car bodies. The body parts obtained in this way are connected in the correct order at the body shop. Complete

bodies are then transported to the paint shop, where the painting process is carried out. After its completion, the finished bodies are dried and transferred to the assembly line. This line comprises of several dozen stands on which all parts of the car equipment are assembled, including seats, dashboards, etc. It can be pointed out that the problem of sequencing has already appeared at the paint shop stage. Several layers of paint are applied to the car body at many subsequent painting stations, see Fig. 2: vehicle bodies go through several production stages before they are actually painted in a specific color.

The paint shop conveyor system transfers car bodies between these stages of production, but also includes buffers in which bodies can be temporarily stored; buffering is particularly important from the perspective of ensuring the continuity of the production process. In the event of downtime in the paint shop due to lack of paint, the vehicles are stored in the buffer until it is completely filled. Furthermore, if a failure occurs in the body shop, car bodies stored in the buffer are successively transported to an output line; preventing downtime in the painting process.

In some paint stations, e.g. the primary and base paint shops, changing to a different color results in costs associated with a change in configuration, which depends on the sequence; if subsequent vehicles in the conveyor system do

✉ Sara Bysko
sara.bysko@polsl.pl

¹ Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland

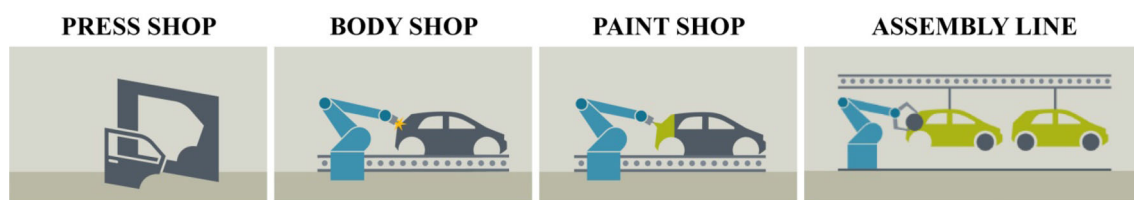
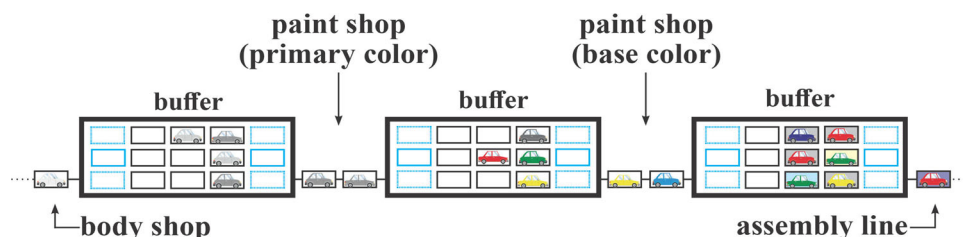


Fig. 1 The structure of the car production line

Fig. 2 The structure of the car production line



not have the same color, it is necessary to remove the old paint from the gun and clean the gun head solvent before applying a new color. This process not only wastes time and money, but is also not environmentally friendly. The painting process is the primary source of air emissions for regulated chemicals, including volatile organic compounds (VOCs) and hazardous air pollutants (HAPs). Solid and hazardous waste is also produced by the painting process, e.g. wasted paint through overspray, chemicals used to clean the paint lines and application equipment (Geffen & Rothenberg, 2000).

The color of the car is usually already determined when it enters the paint shop. In addition, often the previous system (body shop) does not supply cars in a color sequence. That is why in many paint shops significant optimization potential lies in setting the batch before entering a given painting station. The car sequencing problems currently discussed in the literature focuses on creating the longest possible blocks of cars that must be painted in the same color at a given painting station, by changing the body sequence using a buffer system. This problem has many names, including the Car Sequencing Problem (CSP) (Solnon et al., 2008), the Color Batching Problem (CBP) (Spieckermann et al., 2004) and the Paint Shop Problem for Words (PPW) (Epping et al., 2004). The problems considered do not include the periodic cleaning of paint guns occurring within the actual production system.

Periodic cleaning is necessary due to the need to maintain a good quality of painting process; if the guns are not cleaned regularly, the remaining paint will agglutinate, which results in poor quality of the paint coat. One of the parameters of periodic cleaning is its interval, which determines the number of painted car bodies after which this cleaning occurs. From the point of view of optimization of the painting process, both the gun cleaning due to color changes and the periodic cleaning should be considered.

In this paper, the problem of sequencing car bodies directed to the painting station intended for painting in a base color is considered. The sequence is changed by using a buffer with a specific structure and dimensions. Because it is necessary to minimize solvent consumption and unnecessary paint loss the process optimization aims to both minimize the number of color changes and synchronize these changes with periodic cleanings. Furthermore the decision-making process is carried out online in conditions of incomplete information and a correlation between the loading and unloading sides of the buffer is ensured. The specificity of this problem results from the observation of the paint shop's demand for solutions taking into account the above-mentioned assumptions. This problem has been named Car Sequencing Problem 4.0 (CSP 4.0), because it is a part of larger project called Paint Shop 4.0. This concept is still developed in cooperation with ProPoint S.A. as a response for the need to automate workstations located in the paint shop, using tools typical for Industry 4.0, like Virtual Engineering and Virtual Commissioning (Bysko & Krystek, 2020). The project is being commissioned for one of the foreign car manufacturer. The actual data provided by the company were used for development of sequencing algorithms. Due to protection of undisclosed information, the paper does not specify the location of the project, details of used data and the proposed algorithms implementations.

One of the goals of Industry 4.0 is to introduce smart factories with systems enabling intelligent management in order to predict and react effectively and quickly to events and changes in state of the production system (Giannetti & Essien, 2022; Oluyisola et al., 2022). It is crucial especially in the optimization of flexible production systems, an example of which is the paint shop presented in this paper. The initial stage of work on the Paint Shop 4.0 project assumed the development of a system for managing the buffer located in the paint shop. The designed smart buffer control system

(Bysko & Krystek, 2020) is a control system for buffer transport devices integrated with a sequencing algorithm. The smart system was designed through the development of a functional model of the buffer, and the building of Virtual Commissioning environment. From the point of view of the CSP 4.0, the implemented algorithm plays the main role of the system.

So far, as part of the work conducted on this problem, approaches such as using priority and follow-up algorithms have been reviewed (Bysko & Krystek, 2019; Krystek & Bysko, 2019). This paper focuses on algorithms which are based on game theory, i.e. the BSAG-BOSG concept (Buffer Slot Assignment Game – Buffer-OutShuttle Game), the IOSG concept (Input–Output Shuttle Game) and its modified version, the mIOSG concept (modified Input–Output Shuttle Game). The proposed approaches are presented and discussed in detail. Test results from these approaches are compared.

The article is organized as follows. “Literature review” section presents similar sequencing problems considered in the literature and the methods used to solve them. “The car sequencing problem 4.0” section describes CSP 4.0 proposed by the authors. “Game theory approach” section introduces approaches to solve CSP 4.0 based on game theory and describes the practical implementation of proposed algorithms. “Conclusions” section presents the experimental research and discussion of the results obtained. The paper is concluded in the final Section.

Literature review

A literature review was carried out for the car sequencing problem in view of the requirements and limitations of various production departments, in particular the paint shop and assembly line. This section presents the most commonly considered sequencing problems and developed methods of solving them.

The car sequencing problem

Research on scheduling car production processes focused mainly on the problem of car sequencing. This problem was presented and described by Parello et al. (1986) for the first time. The Car Sequencing Problem is concerned with the sequencing of cars in the assembly shop, in which various options (e.g. sunroof, air conditioning) are to be installed in cars by appropriate work stations located along the assembly line. Each workstation is designed to handle a certain numbers of cars passing along an assembly line. To prevent overloading the workstation, cars requiring the same options have to be spaced out in the processing sequence. It was realized from 2005 onwards that it was necessary to expand CSP

to include an assembly line as well as a paint shop. Thus, additional parameters and constraints associated with the car painting process were introduced to the primary definition of CSP. The modified problem became the main subject in the ROADEF’2005 Challenge, organized by the French Society of Operations Research and Decision Analysis (Solnon et al., 2008).

In the literature the CSP was solved using different approaches: constraint programming (Codognet & Diaz, 1996), integer programming (Gravel et al., 2005; Jähren & Achá, 2018) and branch and bound (Valdondo & Gude, 2007). Among the approximate methods two stand out: the local search (Neveu et al., 2004) and the very fast local search method (Estellon et al., 2004) which won the ROADEF’2005 Challenge. Other methods were the beam search procedure (Bautista et al., 2008), tabu search (Zufferey et al., 2006), simulated annealing (Chew et al., 1992; Briant et al., 2008), genetic algorithm (Cheng et al., 1999), ant colony optimization (Moya et al., 2019; Solnon, 2000; Stutzle & Hoos, 2000) and hybrid algorithms (Thiruvady et al., 2011, 2014; Zhang et al., 2018).

The color batching problem

The research, which is closer to the one considered in the paper, concerned the Color Batching Problem. It is based on the use of a buffer system to adapt the sequence of cars from the body shop so that the resulting sequence best suits the needs of the paint shop. In particular, the goal is to minimize the number of color changes (or equivalent, maximizing the average size of color blocks) in the output sequence.

The CBP was formulated as a Sequential Ordering Problem by Spieckermann et al. (2004). To solve the problem they proposed to use the Branch and Bound (B&B) algorithm. Moon et al. (2005) developed a color rescheduling storage, which was used in an automotive factory, and proposed some simple rules to operate the buffer. Two ant colony optimization algorithms were considered by Hartmann and Runkler (2008). The methods were tested for two online re-sequencing stages; filling and releasing the buffer. Two heuristics: arraying in the filling stage and shuffling applied to unloading cars from the buffer, were developed by Sun et al. (2015). Such an approach allowed for quick and effective solutions to CBP. As a result of conducted research the authors stated that the developed heuristics can cooperate with each other in order to obtain competitive solutions for CBP. In addition the computational time was short. CBP was also analyzed in the context of the application in M-to-1 conveyor systems (Ko et al., 2016). The motivation of their research was the problem of re-sequencing a problem occurring for a Korean car manufacturer. The authors proposed a mixed-integer linear programming model for CBP and used dynamic programming for a special case of the

problem under consideration, i.e. 2-to-1 conveyor system. They considered two genetic algorithms to find near-optimal solutions for the general case.

The paint shop problem for words

The methods listed in Sect. 2.2 were based on the use of a buffer for physical car re-sequencing, but in some literature studies the problem of color grouping was analyzed based on the virtual re-sequencing strategy. Such an approach is also known as the Paint Shop Problem for Words. This issue was formulated by Epping et al. (2004). In this case, the car positions in the sequence remain unchanged, but vehicle colors are assigned again to cars that have identical attributes.

Four heuristics to solve the PPW were described by Xu and Zhou (2016). They found the beam search algorithm to be the best of the methods that were tested. In the literature (Amini et al., 2010; Andres & Hochstättler, 2011), a special case of PPW was also looked at, i.e. PPW(2,1), known as the binary problem of PPW, which involves only two colors. Such a problem occurs where each type of car appears twice and must be painted in a different color. Virtual and physical re-sequencing approaches were considered by Sun and Han (2017). Based on the research conducted they showed that integration of these two kinds of re-sequencing methods allows better color grouping results to be obtained compared with conventional physical re-sequencing.

Verification of assumptions for theoretical problems

Based on the analysis of the problems reviewed in the literature, in the context of the paint shop, it can be stated that in reality some assumptions make application of the developed solutions difficult. Among them are:

1. Access to complete information about the initial sequence of vehicles in which the order is to be changed. A production plan is known in the actual production system, but the sequence of vehicles at each stage of production is unknown. This means that the decision-making process is carried out online in conditions of incomplete information, for example, the color of the car body can be read immediately before entering the buffer, therefore the decision is made on the basis of information about one current car body that appears on the buffer input. The color sequence of subsequent bodies on the line preceding the buffer is unknown.
2. Only one optimization goal; minimization the number of color changes.
Considering the requirements of industrial plants, it is necessary to minimize solvent consumption and unnecessary paint loss. This can be achieved if the number of

color changes is minimized and, moreover, the aim is to synchronize these changes with periodic cleaning. Consider the following example:

- the periodic cleaning interval is 3,
- the production plan assumes painting 3 cars to gray and 2 cars to red.

Sample sequences are presented in Fig. 3. Sequence A generates three color changes, sequences B and C two, sequences D and E only one. If we consider only one optimization criterion, which is minimizing the number of color changes (maximizing the length of a block of cars in one color), the D and E sequences are optimal. Between the groups of car bodies in two colors the minimum number of color changes is 1.

Analyzing these sequences from the perspective of two optimization criteria, it can be seen that for sequences B and C, periodic cleaning will take place between bodies of the same color. As a result, the paint remaining in the guns will be unnecessarily lost because after the cleaning process the gun will be reloaded with paint of the same color. Therefore, such sequences are particularly disadvantageous from the perspective of minimizing paint and solvent consumption. Sequence A is an example where one color change is overlapped with periodic cleaning, but in addition there are two additional color changes. For sequences D and E, the only difference is swapping the order of the red and gray car blocks, but the benefit is significant; sequence E requires only one gun to be cleaned, because the color change occurs at the same time as the periodic cleaning.

Based on the examples presented, it can be concluded that the answer to industry requirements is the search for sequences that not only minimize the number of color changes, but also synchronize these changes with periodic cleaning of the paint guns.

3. Most of the developed methods were either intended only for sequencing at the buffer output (e.g. Spieckermann et al., 2004), or did not take into account the need for correlation of heuristics used at the loading and unloading stage of the buffer (e.g. Sun et al., 2015).

The car sequencing problem 4.0

Cars that are transported between the welding and paint shop create the sequence in real time. At the moment when the car reaches the entry position of the buffer, a video camera reads a QR on the car containing information on what color the car body should be painted. This means that the buffer input sequence is not known in advance, only the production plan specified for the specified time horizon is available.

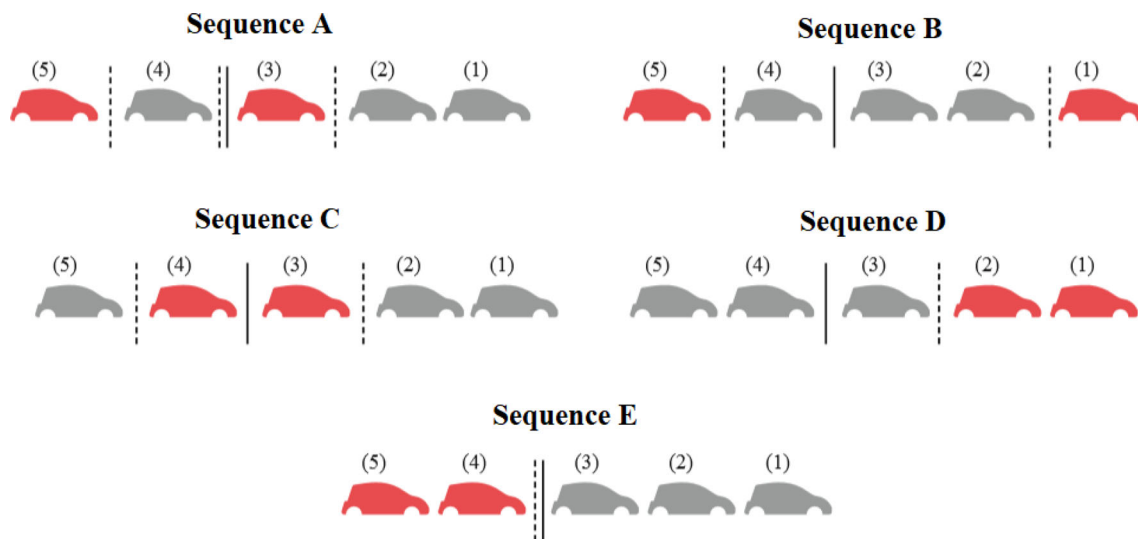
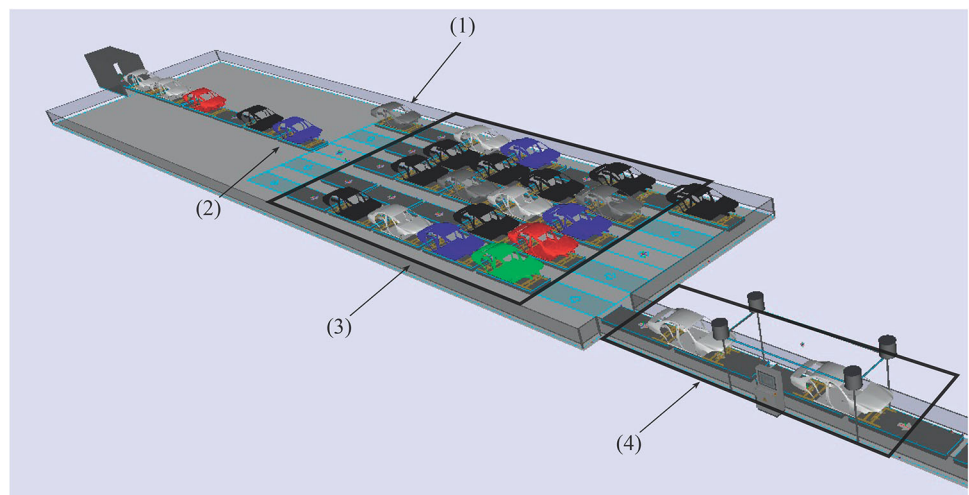


Fig. 3 Impact of sequence selection on the number of color changes

Fig. 4 Illustration of information available in the sequencing process



Thus, input decisions are made based on limited information (Fig. 4), i.e. only the following data is known:

- The color of the car located on the loading shuttle described as cIn (1).
- The color of the car located on the buffer input, described as $cNext$ (2).
- The color of all cars located in the buffer (3).
- The colors of all cars that left the buffer, the last car which left the buffer is described as $cOut$ (4).

The sequencing problem described can be classified as an online problem (Fiat & Woeginger, 1998). It consists of making two decisions in real time (Fig. 5):

1. Entry decision: when a car appears on the buffer input, the line to which the car is directed is determined.

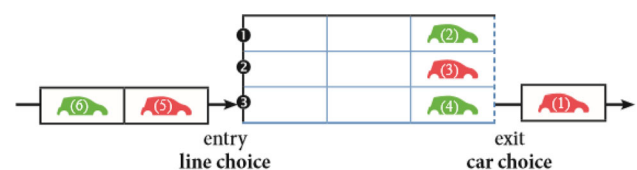


Fig. 5 Decision making process

2. Exit decision: on the unloading side of the buffer, it is necessary to decide which car should be directed to the paint station.

The decision must be completed within one cycle of the painting operation, which again confirms that the requirements of the car industry must be met in real time. One cycle is equal to the time needed at a given station to paint one car body. In the automotive industry, cycle times usually range from 30 s to 3–4 min (Spieckermann et al., 2004).



Fig. 6 Possible solution for the example from Fig. 5

Problem formulation

The problem presented is called the Car Sequencing Problem 4.0, because it is a part of a larger project: Paint Shop 4.0 (Bysko & Krystek, 2020). An instance of the problem considered is defined as a tuple $(V, C, NRowBuff, NColBuff, NPerClean)$, based on production plan and technical parameters, where:

- $V = \{v_1, \dots, v_N\}$ —set of vehicles to be produced,
- $C = \{c_1, \dots, c_D\}$ —set of available colors and function $c: V \rightarrow C$, that associates color c_i to each vehicle v_i ,
- $NRowBuff, NColBuff$ —buffer size defined by number of buffer rows and number of buffer columns,
- $NPerClean$ —periodic cleaning interval.

The solution of the CSP 4.0 is a sequence $\delta = (\delta_n)_{n \in \{1, \dots, N\}}$ which defines an order in which cars are delivered to the paint station. The sequence is generated in real time, therefore the solution is known only after the current production plan is completed. This means that it is difficult to predict the consequences of a decision made on an ongoing basis. For the example shown in Fig. 5, one of the possible sequences is the sequence $\delta = \delta_1, \delta_3, \delta_5, \delta_4, \delta_2, \delta_6$ (Fig. 6).

Quality indices

In order to evaluate the solution of Car Sequencing Problem 4.0, two quality indices are proposed (1,2).

- Number of Changeovers (NCs): defines the number of color changes excluding the changes occurring during periodic cleaning.

$$NCs = \sum_{\substack{1 \leq n \leq N-1 \\ n \bmod NPerClean \neq 0}} isCC_{n,n+1} \quad (1)$$

where: N —number of vehicles to be produced,
 $isCC_{n,n+1} = \begin{cases} 0, & c(v_n) = c(v_{n+1}) \\ 1, & \text{else} \end{cases}$ —the variable determining whether there is a color change between the next two cars, $c(v_n)$ —the color of n -th vehicle.

- Effectiveness of Synchronization (ES): determines the total number of color changes occurring between two sub-sequences in relation to the number of all color changes.

$$ES = \frac{\sum_{\substack{1 \leq n \leq N-1 \\ n \bmod NPerClean = 0}} isCC_{n,n+1}}{\left\lceil \frac{N-1}{NPerClean} \right\rceil} \times 100\% \quad (2)$$

Hierarchical optimization

The optimization problem is to find $NPerClean$ -element sub-sequences for which the number of color changes requiring gun changeovers is minimal. Consequently, the number of changeovers for the whole sequence, which is a solution of the problem, is also minimal. For economic reasons, the most favorable situation is when the gun changeover coincides with periodic cleaning. It should be noted that in the search for a problem, a hierarchical approach is used. The primary optimization goal is to minimize the number of color changes, while the secondary optimization goal is to maximize the value of the ES indicator. For the example shown in Fig. 7, two solutions are presented. Figure 8 presents an optimal solution in terms of the primary optimization goal and Fig. 9 shows the sequence obtained when the secondary goal is also taken into account.

Game theory approach

The Car Sequencing Problem 4.0 discussed is making two real-time decisions regarding the choice of transport line for a car entering the buffer and the choice of a car directed to the paint shop. Each of these decisions can be made as a result of playing the proposed games once (Bysko & Krystek, 2018). This section introduces and describes in detail three concepts, which are based on game theory. There are: *Buffer Slot Assignment Game–Buffer-Out Shuttle Game* called the BSAG-BOSG concept, *In-Out Shuttle Game* (IOSG) and its modification called modified *In-Out Shuttle Game* (mIOSG).

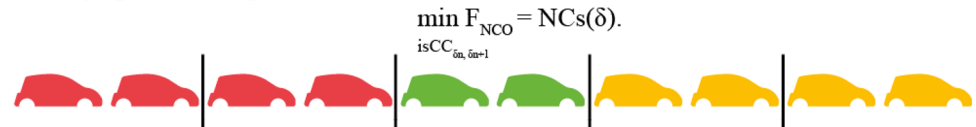
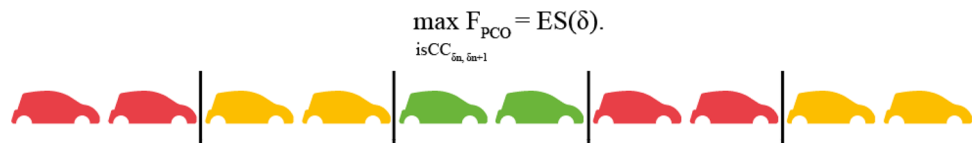
John von Neumann presented in 1928 the basic ideas of game theory, which were originated from the problems of maximum and minimum (Leonard, 1995). In 1944, mathematician J. von Neumann and economist Oskar Morgenstern published a work in which they formulated previously developed theories (von Neumann & Morgenstern, 1944). A general model of sequential games in an extensive form was formulated, such a game is characterized by the finite number of players, the set of possible strategies or choices for the players and the payoff function (cost function). The payoff function determines what is the cost to each player based on a given strategy profile. J. von Neumann and O. Morgenstern described this game structure as a normal form, players choose strategies simultaneously. A strategy is the

Fig. 7 Example input sequence

Input sequence:

**Fig. 8** Example output sequence obtained after taking into account the primary optimization goal

Primary optimization goal:

**Fig. 9** Example output sequence after taking into account two optimization goals

complete instruction of actions which player can take in a game. A payoff function, which assigns a certain payoff to each player depending on his strategy and the strategy of the other players. In this article, it is proposed to use the weighted criteria method to determine the players payoff function. This method consists in reducing a multi-criteria problem to a single-criteria problem as a result of introducing a substitute criterion, which is a weighted sum of the criteria. If the number of players is limited to two and if their sets of strategies consist of only a few elements, the outcome of the payoff function can be represented in the so-called payoff matrix.

Due to specific properties, there are many types of strategy games. Game theory has evolved from finite to infinite, from two players to many players or from certainty problems to random problems. The subject of this research are non-cooperative, 2-person non-zero-sum games played once with complete information and the finite number of strategies. In a such type of game, the players need to choose their strategies before make any binding decisions. A two player game is called a zero-sum game if the sum of the payoffs to each player is constant for all possible outcomes of the game. Complete information means that the players' strategies and payoffs are common knowledge to all players.

One of the fundamental concept in game theory is Nash equilibrium (Nash, 1950). It defines a situation where each player's strategy is optimal given the strategies of all other players. It also allows predicting the decisions of the players if they are making decisions at the same time and the decision of one player takes into account the decisions of other players. A game may include multiple Nash equilibria or none of them. In this paper the Nash equilibrium as the standard desired strategy is proposed to model the individual choices of players in a game.

BSAG-BOSG concept

In the case of the BSAG-BOSG concept the sequencing problem is solved by playing two independent games; the BSAG game is played on the buffer entry side and the BOSG game is played on the buffer exit side.

Buffer slot assignment game (BSAG)

On the buffer loading side, the decision problem can be defined analogously to the concept presented by Ayala et al. (2011). In general, it can be considered that the buffer is a set of parking spaces (slots) for cars that are on the side of the entrance. Each of the cars entering can be treated as an independent decision-making agent. Such a formulation of the car sequencing problem, as an allocation of assigning a parking space, allows us to consider CSP 4.0 as a game. It is assumed that vehicles entering the paint shop are players and that they compete for parking spaces and want to maximize profit at the same time. Their payment depends on their own choices and choices of other players. Taking into account incomplete access to information and the considered structure of the buffer, the *Buffer Slot Assignment Game* can be defined as follows:

- the set of players: $P^{(\text{BSAG})} = \{v_I, v_{II}\}$ (v_I – vehicle located on the loading conveyor, v_{II} —vehicle located on the buffer input),
- the set of available strategies: $S = \{s_1, s_2, s_3, s_4, s_5\}$ (transport lines),
- the payoff function $F^{V(\text{BSAG})}(s_i, cX)$ is defined as a weighted objective function (3), where:

$$cX = \begin{cases} cIn & \text{for player } V_I \\ cNext & \text{for player } V_{II} \end{cases}.$$

$$\begin{aligned}
F^{V(BSAG)}(s_i, cX) = & w_{LOcc}^{V(BSAG)} F_{LOcc}^{V(BSAG)}(s_i) \\
& + w_{CDiv}^{V(BSAG)} F_{CDiv}^{V(BSAG)}(s_i) \\
& + w_{LPrio}^{V(BSAG)} F_{LPrio}^{V(BSAG)}(s_i) \\
& + w_{BL}^{V(BSAG)} F_{BL}^{V(BSAG)}(s_i, cX) \\
& + w_{LBC}^{V(BSAG)} F_{LBC}^{V(BSAG)}(s_i, cX). \quad (3)
\end{aligned}$$

where: $F_{LOcc}^{V(BSAG)}(s_i)$ —the line occupation criterion (lines with the smallest occupation are preferred):

$$\begin{aligned}
& F_{LOcc}^{V(BSAG)}(s_i) \\
& = \begin{cases} \frac{NColBuff - LOcc(s_i)}{NColBuff} \text{ for } NColBuff > LOcc(s_i) \geq 0 \\ \text{and Limit} = 0 \\ -2, \text{ for } LOcc(s_i) = NColBuff \text{ or Limit} = 1 \end{cases},
\end{aligned}$$

$F_{CDiv}^{V(BSAG)}(s_i)$ —the criterion of line color variety (lines with the smallest variety of colors are preferred):

$$F_{CDiv}^{V(BSAG)}(s_i) = \frac{C - CDiv(s_i)}{C},$$

$F_{LPrio}^{V(BSAG)}(s_i)$ —the line priority criteria (lines with the lowest priority are preferred):

$$F_{LPrio}^{V(BSAG)}(s_i) = \begin{cases} \frac{1}{LPrio(s_i) \cdot NPP}, \text{ for } LOcc(s_i) > 0 \\ 1, \text{ for } LOcc(s_i) = 0 \end{cases},$$

$F_{BL}^{V(BSAG)}(s_i, cX)$ —the criterion of length of an unblocked block in the color cX (lines with the longest block length are preferred):

$$F_{BL}^{V(BSAG)}(s_i, cX) = \frac{BL(s_i, cX)}{NColBuff},$$

$F_{LBC}^{V(BSAG)}(s_i, cX)$ —the criterion of block in cX color blocked on the left side (lines without such blocks are preferred):

$$F_{LBC}^{V(BSAG)}(s_i, cX) = \begin{cases} 0 & \text{if on the transport line } s_i \\ & \text{appears a block of cars in } cX \text{ color} \\ & \text{which is blocked from the left side} \\ 1 & \text{else} \end{cases},$$

$Limit = \lfloor NRowBuff/2 \rfloor$ —requirement imposed by the paint shop, $LOcc(s_i)$ —Line Occupancy, determines the number of cars located on the transport line s_i , C —maximum number of available colors, $CDiv(s_i)$ —Color Diversity, determines the number of different colors appearing on the transport line s_i , $LPrio(s_i)$ —Line Priority, determines the priority of the

transport line s_i , which is the same as the priority of the color to which the line ends $Prio(cX)$:

$$Prio(cX) = \frac{NPP(cX) - NP(cX) - NB(cX)}{NPP},$$

- NPP —the number cars determined by the production plan,
- $NPP(cX)$ —the number of cars in the color cX determined by the production plan,
- $NP(cX)$ —number of produced cars in the color cX ,
- $NB(cX)$ —the number of cars in the color cX located in the buffer,

$BL(s_i, cX)$ —Block Length in given color, determines the number of cars in cX color, which form on the transport line s_i a block unblocked on the left side, where $cX \in \{cIn, cNext\}$. $w_{LOcc}^{V(BSAG)}$, $w_{CDiv}^{V(BSAG)}$, $w_{LPrio}^{V(BSAG)}$, $w_{BL}^{V(BSAG)}$, $w_{LBC}^{V(BSAG)}$ —weights, it is assumed, that the sum of all weights is 1.

Buffer-OutShuttle game (BOSG) The decision problem on the buffer output is considered as a game played between the buffer and the unloading shuttle. The proposed approach is motivated by an attempt to make the decisions made on the input and output side of the buffer. For this purpose, the buffer as a player seeks to obtain a fill state, which is most advantageous from the perspective of the entry situation. This is the case when the buffer wants to remove a car from a line that is completely full, which ends with a car of the same color as the color of the car located on the loading shuttle. In turn, the purpose of the unloading shuttle is to optimize the proposed quality indicators. The proposed *Buffer-OutShuttle Game* can be defined as follows:

- the set of players: $P^{(BOSG)} = \{B, OS\}$, where B —buffer, OS —unloading shuttle,
- the set of available strategies: $S = \{s_1, s_2, s_3, s_4, s_5\}$ (transport lines),
- the payoff function for player I is determined by the $F^{B(BOSG)}(s_i, cIn, cNext)$ function and the payoff function for player II is determined by the $F^{OS(BOSG)}(s_i, cX)$ function.

The payoff function $F^{B(BOSG)}(s_i, cIn, cNext)$ is defined as a weighted objective function (4).

$$\begin{aligned}
 F^{B(BOSG)}(s_i, cIn, cNext) = & w_{LOcc}^{B(BOSG)} F_{LOcc}^{B(BOSG)}(s_i) \\
 & + w_{CDiv}^{B(BOSG)} F_{CDiv}^{B(BOSG)}(s_i) \\
 & + w_{LPrio}^{B(BOSG)} F_{LPrio}^{B(BOSG)}(s_i) \\
 & + w_{FSCcIn}^{B(BOSG)} F_{FSCcIn}^{B(BOSG)}(s_i, cIn) \\
 & + w_{FSCcNext}^{B(BOSG)} F_{FSCcNext}^{B(BOSG)}(s_i, cNext)
 \end{aligned} \quad (4)$$

where:

$$\begin{aligned}
 F_{LOcc}^{B(BOSG)}(s_i) \\
 = \frac{LOcc(s_i)}{NColBuff},
 \end{aligned}$$

$$F_{CDiv}^{B(BOSG)}(s_i) = \frac{CDiv(s_i)}{C},$$

$$F_{LPrio}^{B(BOSG)}(s_i) = \begin{cases} \frac{1}{LPrio(s_i) \cdot NPP}, & \text{for } LOcc(s_i) > 0 \\ 1, & \text{for } LOcc(s_i) = 0 \end{cases},$$

$F_{FSCcIn}^{B(BOSG)}(s_i, cIn)$ —the criterion of free space for car in the color cIn (lines that free up space for the longest blocks in color cIn are preferred):

$$\begin{aligned}
 F_{FSCcIn}^{B(BOSG)}(s_i, cIn) \\
 = \begin{cases} \frac{BL(cIn, s_i)}{NColBuff}, & \text{for } LOcc(s_i) = NColBuff \\ 0, & \text{for } LOcc(s_i) < NColBuff \end{cases},
 \end{aligned}$$

$F_{FSCcNext}^{B(BOSG)}(s_i, cNext)$ —the criterion of free space for car in the color $cNext$ (lines that free up space for the longest blocks in color $cNext$ are preferred):

$$\begin{aligned}
 F_{FSCcNext}^{B(BOSG)}(s_i, cNext) \\
 = \begin{cases} \frac{BL(cNext, s_i)}{NColBuff}, & \text{for } LOcc(s_i) = NColBuff \\ 0, & \text{for } LOcc(s_i) < NColBuff \end{cases},
 \end{aligned}$$

$w_{LOcc}^{B(BOSG)}$, $w_{CDiv}^{B(BOSG)}$, $w_{LPrio}^{B(BOSG)}$, $w_{FSCcIn}^{B(BOSG)}$, $w_{FSCcNext}^{B(BOSG)}$ —weights, it is assumed, that the sum of all weights is 1.

The payoff function $F^{OS(BOSG)}(s_i, cX)$ is defined as a weighted objective function (5).

$$\begin{aligned}
 F^{OS(BOSG)}(s_i, cX) = & w_{CCComp}^{OS(BOSG)} F_{CCComp}^{OS(BOSG)}(s_i) \\
 & + w_{ISComp}^{OS(BOSG)} F_{ISComp}^{OS(BOSG)}(s_i, cX) \\
 & + w_{CCPerClean}^{OS(BOSG)} F_{CCPerClean}^{OS(BOSG)}(s_i) \\
 & + w_{CCCompUnCol}^{OS(BOSG)} F_{CCCompUnCol}^{OS(BOSG)}(s_i).
 \end{aligned} \quad (5)$$

where: $F_{CCComp}^{OS(BOSG)}(s_i)$ —the color compatibility criterion, lines ending in color $cOut$ are preferred:

$$\begin{aligned}
 F_{CCComp}^{OS(BOSG)}(s_i) \\
 = \begin{cases} 1, & \text{if the color at the beginning of the line } s_i \\ & \text{is the same as } cOut \\ 0, & \text{else} \end{cases},
 \end{aligned}$$

$F_{ISComp}^{OS(BOSG)}(s_i, cX)$ —the block length matching criterion (line for which $F_{ISComp}^{OS(BOSG)}(s_i, cX) = 1$ are preferred):

$$\begin{aligned}
 F_{ISComp}^{OS(BOSG)}(s_i, cX) \\
 = \begin{cases} 1 - \frac{\text{mod} \left(\frac{BL(cX, s_i)}{SeqCompletion} \right)}{SeqCompletion}, & \text{for } BL(cX, s_i) > SeqCompletion \\ \frac{BL(cX, s_i)}{SeqCompletion}, & \text{for } BL(cX, s_i) \leq SeqCompletion \end{cases},
 \end{aligned}$$

$F_{CCPerClean}^{OS(BOSG)}(s_i)$ —the color variation criterion:

$$\begin{aligned}
 F_{CCPerClean}^{OS(BOSG)}(s_i) \\
 = \begin{cases} 1, & \text{if periodic cleaning appears} \\ & \text{and the color of the beginning of the line } s_i \\ & \text{is different from the color } cOut \\ 0, & \text{else} \end{cases},
 \end{aligned}$$

$F_{CCCompUnCol}^{OS(BOSG)}(s_i)$ —the criterion of color match in the column (lines that end in the same color are preferred):

$$F_{CCCompUnCol}^{OS(BOSG)}(s_i) = \frac{CCCompUnCol(s_i)}{NRowBuff},$$

$SeqCompletion$ —Sequence Completion, determines the number of cars that are currently missing to obtain a $NPerClean$ -element sequence, $CCCompUnCol(s_i)$ —Color Compatibility in Unload Column, determined if there is a color match within the unloading column. $w_{CCComp}^{OS(BOSG)}$, $w_{ISComp}^{OS(BOSG)}$, $w_{CCPerClean}^{OS(BOSG)}$, $w_{CCCompUnCol}^{OS(BOSG)}$ —weights, it is assumed, that the sum of all weights is 1.

IOSG concept

Decisions on the entry and exit of the buffer are made based on the result of the game played between two shuttles, i.e. loading and unloading. The IS player strives to find the best transport line for the input car body, while taking into account the current buffer status. In turn, the goal of the player OS is to optimize the proposed quality indices. Due to the fact that the buffer is loaded at a different time than unloading, the game is played every time a decision is needed on the loading or unloading side of the buffer, with the decision being used only for one side of the buffer. The proposed *In-Out Shuttle Game* can be defined as follows:

- the set of players: $P^{(IOSG)} = \{IS, OS\}$, where IS—input shuttle, OS—output shuttle,
- the set of available strategies: $S = \{s_1, s_2, s_3, s_4, s_5\}$ (transport lines),
- the payoff function for player IS is determined by the $F^{IS(IOSG)}(s_i, cIn, cNext)$ function and the payoff function for player OS is determined by the $F^{OS(IOSG)}(s_i, cX)$ function.

The payoff function $F^{IS(IOSG)}(s_i, cIn, cNext)$ is defined as a weighted objective function (6).

$$F^{IS(IOSG)}(s_i, cIn, cNext) = w_{Locc}^{IS(IOSG)} F_{Locc}^{IS(IOSG)}(s_i) + w_{CDiv}^{IS(IOSG)} F_{CDiv}^{IS(IOSG)}(s_i) + w_{LPrio}^{IS(IOSG)} F_{LPrio}^{IS(IOSG)}(s_i) + w_{FSCcIn}^{IS(IOSG)} F_{FSCcIn}^{IS(IOSG)}(s_i, cIn) + w_{FSCcNext}^{IS(IOSG)} F_{FSCcNext}^{IS(IOSG)}(s_i, cNext) \quad (6)$$

where:

$$F_{Locc}^{IS(IOSG)}(s_i) = F_{Locc}^{B(BOSG)}(s_i)$$

$$F_{CDiv}^{IS(IOSG)}(s_i) = F_{CDiv}^{B(BOSG)}(s_i)$$

$$F_{LPrio}^{IS(IOSG)}(s_i) = F_{LPrio}^{B(BOSG)}(s_i)$$

$$F_{FSCcIn}^{IS(IOSG)}(s_i, cIn) = F_{FSCcIn}^{B(BOSG)}(s_i, cIn)$$

$$F_{FSCcNext}^{IS(IOSG)}(s_i, cNext) = F_{FSCcNext}^{B(BOSG)}(s_i, cNext)$$

$w_{Locc}^{IS(IOSG)}$, $w_{CDiv}^{IS(IOSG)}$, $w_{LPrio}^{IS(IOSG)}$, $w_{FSCcIn}^{IS(IOSG)}$, $w_{FSCcNext}^{IS(IOSG)}$ —weights, it is assumed, that the sum of all weights is 1.

The payoff function $F^{OS(IOSG)}(s_i, cX)$ is defined as a weighted objective function (7).

$$F^{OS(IOSG)}(s_i, cX) = w_{CCComp}^{OS(IOSG)} F_{CCComp}^{OS(IOSG)}(s_i) + w_{ISComp}^{OS(IOSG)} F_{ISComp}^{OS(IOSG)}(s_i) + w_{CCPerClean}^{OS(IOSG)} F_{CCPerClean}^{OS(IOSG)}(s_i, cX) + w_{CCCompUnCol}^{OS(IOSG)} F_{CCCompUnCol}^{OS(IOSG)}(s_i) \quad (7)$$

where:

$$F_{CCComp}^{OS(IOSG)}(s_i) = F_{CCComp}^{OS(BOSG)}(s_i)$$

$$F_{ISComp}^{OS(IOSG)}(s_i, cX) = F_{ISComp}^{OS(BOSG)}(s_i, cX)$$

$$F_{CCPerClean}^{OS(IOSG)}(s_i) = F_{CCPerClean}^{OS(BOSG)}(s_i)$$

$w_{CCComp}^{OS(IOSG)}$, $w_{ISComp}^{OS(IOSG)}$, $w_{CCPerClean}^{OS(IOSG)}$, $w_{CCCompUnCol}^{OS(IOSG)}$ —weights, it is assumed, that the sum of all weights is 1.

mIOSG concept

We propose also a modification of the IOSG game. The proposed *modified In–Out Shuttle Game* can be defined as follows:

- the set of players: $P^{(mIOSG)} = \{IS, OS\}$, where IS—input shuttle, OS—output shuttle,
- the set of available strategies: $S = \{s_1, s_2, s_3, s_4, s_5\}$ (transport lines),
- the payoff function for player IS is determined by the $F^{IS(mIOSG)}(s_i, cIn, cNext)$ function and the payoff function for player OS is determined by the $F^{OS(mIOSG)}(s_i, cX)$ function.

The proposed change was to take into account the buffer status when making decisions regarding the selection of the body to be unloaded and concerned the extension of the payout function of the OS player (8) to the following form:

$$F^{OS(mIOSG)}(s_i, cX) = w_{CCComp}^{OS(mIOSG)} F_{CCComp}^{OS(mIOSG)}(s_i) + w_{ISComp}^{OS(mIOSG)} F_{ISComp}^{OS(mIOSG)}(s_i) + w_{CCPerClean}^{OS(mIOSG)} F_{CCPerClean}^{OS(mIOSG)}(s_i, cX) + w_{CCCompUnCol}^{OS(mIOSG)} F_{CCCompUnCol}^{OS(mIOSG)}(s_i) + w_{CDiv}^{OS(mIOSG)} F_{CDiv}^{OS(mIOSG)}(s_i) + w_{LPrio}^{OS(mIOSG)} F_{LPrio}^{OS(mIOSG)}(s_i) \quad (8)$$

where:

$$F_{CCComp}^{OS(mIOSG)}(s_i) = F_{CCComp}^{OS(IOSG)}(s_i)$$

$$F_{ISComp}^{OS(mIOSG)}(s_i) = F_{ISComp}^{OS(IOSG)}(s_i)$$

$$F_{CCPerClean}^{OS(mIOSG)}(s_i, cX) = F_{CCPerClean}^{OS(IOSG)}(s_i, cX)$$

$$F_{CCCompUnCol}^{OS(mIOSG)}(s_i) = F_{CCCompUnCol}^{OS(IOSG)}(s_i)$$

$$F_{CDiv}^{OS(mIOSG)}(s_i) = F_{CDiv}^{B(BOSG)}(s_i)$$

$$F_{LPrio}^{OS(mIOSG)}(s_i) = F_{LPrio}^{B(BOSG)}(s_i)$$

$w_{CCComp}^{OS(mIOSG)}$, $w_{ISComp}^{OS(mIOSG)}$, $w_{CCPerClean}^{OS(mIOSG)}$, $w_{CCCompUnCol}^{OS(mIOSG)}$, $w_{CDiv}^{OS(mIOSG)}$, $w_{LPrio}^{OS(mIOSG)}$ —weights, it is assumed, that the sum of all weights is 1.

The payout function of the IS player (9) is given by a formula analogous to the formula (6):

$$F_{IS(mIOSG)}^{IS(mIOSG)}(s_i, cIn, cNext) = w_{LOcc}^{IS(mIOSG)} F_{LOcc}^{IS(mIOSG)}(s_i) + w_{CDiv}^{IS(mIOSG)} F_{CDiv}^{IS(mIOSG)}(s_i) + w_{LPrio}^{IS(mIOSG)} F_{LPrio}^{IS(mIOSG)}(s_i) + w_{FSCcIn}^{IS(mIOSG)} F_{FSCcIn}^{IS(mIOSG)}(s_i, cIn) + w_{FSCcNext}^{IS(mIOSG)} F_{FSCcNext}^{IS(mIOSG)}(s_i, cNext) \quad (9)$$

where:

$$F_{LOcc}^{IS(mIOSG)}(s_i) = F_{LOcc}^{IS(IOUSG)}(s_i)$$

$$F_{CDiv}^{IS(mIOSG)}(s_i) = F_{CDiv}^{IS(IOUSG)}(s_i)$$

$$F_{LPrio}^{IS(mIOSG)}(s_i) = F_{LPrio}^{IS(IOUSG)}(s_i)$$

$$F_{FSCcIn}^{IS(mIOSG)}(s_i, cIn) = F_{FSCcIn}^{IS(IOUSG)}(s_i, cIn)$$

$$F_{FSCcNext}^{IS(mIOSG)}(s_i, cNext) = F_{FSCcNext}^{IS(IOUSG)}(s_i, cNext)$$

$w_{LOcc}^{IS(mIOSG)}$, $w_{CDiv}^{IS(mIOSG)}$, $w_{LPrio}^{IS(mIOSG)}$, $w_{FSCcIn}^{IS(mIOSG)}$, $w_{FSCcNext}^{IS(mIOSG)}$ —weights, it is assumed, that the sum of all weights is 1.

Implementation of game theory-based algorithms

The classic game theory approach to solving a non-cooperative game is the Nash equilibrium. It is proposed to search for the Nash equilibrium in pure strategies. The presented theory-based concepts have been implemented in C# and Python. In order to find the Nash equilibrium points of the considered games, the free Nashpy library and the Lemke-Howson algorithm implemented in it were used.

In game theory a payoff matrix is a table in which strategies of one player are listed in rows and those of the other player in columns. The cells show payoffs to each player such that the payoff of the row player is listed first. Depending on the value in the payoff matrix, there may be one Nash equilibrium, many permissible Nash equilibria, or a Nash equilibrium in pure strategies may not exist. In a situation where the Nash equilibrium in pure strategies does not exist or there are many Nash equilibrium points, the following procedure of selecting one solution was proposed:

1. Select from the found Nash equilibria (or in the absence of any – from among all elements of the payout matrix) those for which the player's payout:
 - v_I for the BSAG game,
 - OS for the BOSG game,
 - IS for the IOSG and mIOSG games played at the input of the buffer,
 - OS for the IOSG and mIOSG games played at the buffer output,
 is the greatest.
2. If more than one acceptable solution is found, select among them solutions for which a player's payout:
 - v_{II} for the BSAG game,
 - B for the BOSG game,
 - OS for the IOSG and mIOSG games played at the input of the buffer,
 - IS for the IOSG and mIOSG games played at the buffer output,
 is the greatest.
3. If more than one acceptable solution is found, choose the first one.

The payoff matrix row where the Nash equilibrium is located specifies the buffer line to which the car should be directed (on the input side of the buffer) or from which line the car should be directed to the paint station (on the output side of the buffer), depending on whether an entry or exit decision is made.

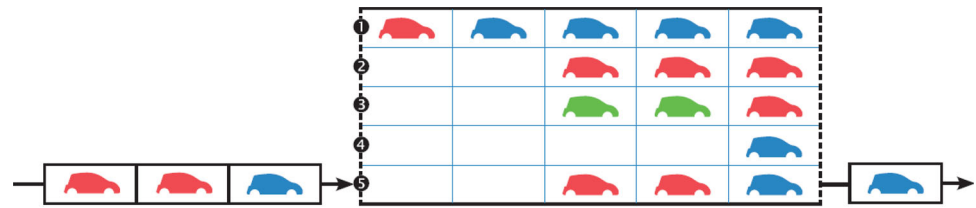
When constructing a payoff matrix, attention is paid to how its value is determined diagonal:

1. BSAG-BOSG model – in the case of a game played at the buffer input, the value of the payout for the second player (the car at the buffer input) is calculated as if on a given transport line there was a car from the loading shuttle; in the case of a game played at the buffer output, the value of the payout for player B is calculated as if the first car was unloaded from a given transport line.
2. IOSG and mIOSG model – the value of the payout for an IS player is calculated as if the first car was unloaded from a given transport line; payout value for the OS player is calculated as if it were on a given transport line car from the input conveyor.

Case study

This chapter explains the application of the selected game theory approach to an example CSP 4.0 with the initial buffer state shown in Fig. 10.

Fig. 10 Example initial buffer state



For the purposes of this example, the following assumptions were made:

Instance	Production plan	Initial buffer state	Production state
$V = \{v_1, \dots, v_{150}\}$	$NPP(blue) = 56$	$NB(blue) = 5$	$cIn = cOut = blue$
$C = \{c_1, \dots, c_6\}$	$NPP(orange) = 21$	$NB(orange) = 1$	$cNext = red$
$NRowBuff = 5$	$NPP(red) = 12$	$NB(red) = 6$	$NP(blue) = 1$, so only one car body was painted
$NColBuff = 5$	$NPP(green) = 17$	$NB(green) = 2$	
$NPerClean = 7$	$NPP(black) = 25$	$NB(black) = 0$	
	$NPP(yellow) = 19$	$NB(yellow) = 0$	

The BSAG-BOSG concept is used for the proposed problem. It is assumed that the decision is made first on the entry side of the buffer. Therefore, in order to determine the buffer line on which the vehicle from the loading conveyor is to be directed, it is necessary to play the game BSAG. The following weights are assumed for the components of the payoff functions $F^{V(BSAG)}(s_i, cX)$: $w_{LOcc}^{V(BSAG)} = 0, 2$, $w_{CDiv}^{V(BSAG)} = 0, 1$, $w_{LPrio}^{V(BSAG)} = 0, 1$, $w_{BL}^{V(BSAG)} = 0, 4$, $w_{LBC}^{V(BSAG)} = 0, 2$.

After calculating the payouts of all players, the payoff matrix for the BSAG game is as follows:

	v_{II}				
v_I	− 0.13; − 0.05	− 0.13; 0.62	− 0.13; 0.16	− 0.13; 0.45	− 0.13; 0.51
0.38; − 0.05	0.38; 0.13	0.38; 0.16	0.38; 0.45	0.38; 0.51	
0.37; − 0.05	0.37; 0.62	0.37; 0.11	0.37; 0.45	0.37; 0.51	
0.85; − 0.05	0.85 ; 0.62	0.85; 0.16	0.85; 0.21	0.85; 0.51	
0.35; − 0.05	0.35; 0.62	0.35; 0.16	0.35; 0.45	0.35; 0.09	

Nash equilibrium points are determined for the presented payoff matrix. There is only one such point: (0.85; 0.62). As

the car body in color cIn is first directed to the buffer, a decision is made to direct the vehicle on the loading conveyor to the line 4 (according to the choice of this car body). The state of the buffer is changing (Fig. 11). Then, a decision is made on the output side of the buffer. The BOSG game is played to define the line from which the car body should be directed to be painted. The following weights are assumed for the components of the payout function $F^{B(BOSG)}(s_i, cIn, cNext)$: $w_{LOcc}^{B(BOSG)} = 0.35$, $w_{CDiv}^{B(BOSG)} = 0.15$, $w_{LPrio}^{B(BOSG)} = 0.1$, $w_{FSCcIn}^{B(BOSG)} = 0.25$, $w_{FSCcNext}^{B(BOSG)} = 0.15$ and for the components of the payout function $F^{OS(BOSG)}(s_i, cX)$: $w_{CCComp}^{OS(BOSG)} = 0.35$, $w_{ISComp}^{OS(BOSG)} = 0.15$, $w_{CCPerClean}^{OS(BOSG)} = 0.35$, $w_{CCCompUnCol}^{OS(BOSG)} = 0.15$.

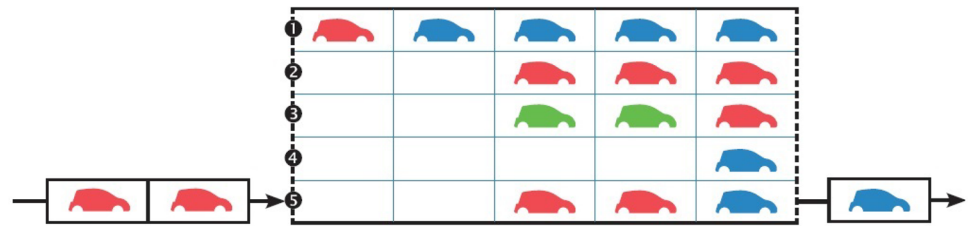
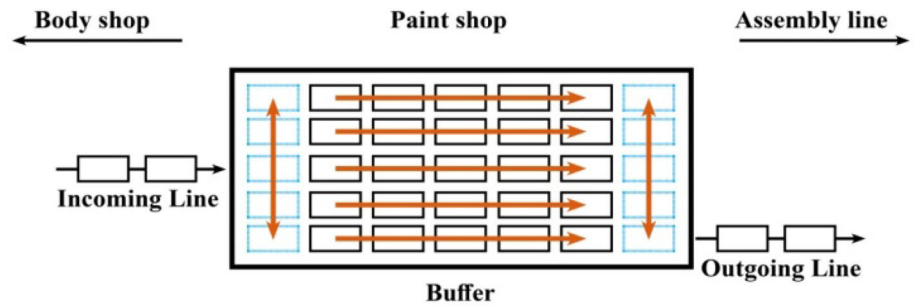
After calculating the payouts of all players, the payoff matrix for the BOSG game is as follows:

	OS				
B	0.33; 0.38	0.48; 0.1	0.48; 0	0.48 ; 0.43	0.48; 0.38
0.26; 0.38	0.18; 0.1	0.26; 0	0.26; 0.43	0.26; 0.38	
0.28; 0.38	0.28; 0.1	0.17; 0	0.28; 0.43	0.28; 0.38	
0.17; 0.38	0.17; 0.1	0.17; 0	0; 0.43	0.17; 0.38	
0.26; 0.38	0.26; 0.1	0.26; 0	0.26; 0.43	0.18; 0.38	

Nash equilibrium points are determined for the presented payoff matrix. There is only one such point: (0.48; 0.43). Due to the optimization of the quality indicators, it is assumed that the choice of the OS player is more important. On this basis, a decision is made to direct the vehicle from line 4 to be painted. Playing games continues until the end of production.

Computational experiments and results

The buffer model shown in Fig. 12 was used for the computational experiments. It consisted of 25 positions (5×5), intended for car body buffering, and 2 shuttles used to transport the car bodies to the correct transport line (loading shuttle) and to the painting station from the buffer (unloading shuttle).

Fig. 11 Buffer state after making a decision**Fig. 12** The structure of the buffer considered

For the purpose of the research, 2 sets of real and experimental data were used. The first set consisted of 5 real samples, each with 100 car bodies and the second consisted of 5 experimental samples, each with 1000 car bodies. Samples with 1000 cars are marked with an additional *. The cars were painted in one of 6 possible colors. The individual samples differed only in the order in which the car bodies were transported to the buffer. Each instance had the same set of parameters:

- $V = \{v_1, \dots, v_{100}\}$,
- $C = \{c_1, \dots, c_6\}$,
- $NRowBuff = 5, NColBuff = 5$,
- $NPerClean = 7$.

The color distribution in each set was the same and was: C1: 6%, C2: 38%, C3: 29%, C4: 14%, C5: 10%, C6: 3%.

The aim of the research was to compare the quality of the CSP 4.0 solutions between three proposed algorithms which are based on game theory. For this purpose proposed quality indices NC and ES were calculated.

Experimental setup

Experimental studies were conducted for each of the proposed game theory concepts, assuming empirically selected values of payout function weights:

1. BSAG-BOSG concept:

- weights for $F^{V(BSAG)}(s_i, cX)$ function:
 - $w_{LOcc}^{V(BSAG)} = 0.2$,

- $w_{CDiv}^{V(BSAG)} = 0.1$,
- $w_{LPrio}^{V(BSAG)} = 0.1$,
- $w_{BL}^{V(BSAG)} = 0.4$,
- $w_{LBC}^{V(BSAG)} = 0.2$;

– weights for $F^{B(BOSG)}(s_i, cIn, cNext)$ function:

- $w_{LOcc}^{B(BOSG)} = 0.35$,
- $w_{CDiv}^{B(BOSG)} = 0.15$,
- $w_{LPrio}^{B(BOSG)} = 0.1$,
- $w_{FSCcIn}^{B(BOSG)} = 0.25$,
- $w_{FSCcNext}^{B(BOSG)} = 0.15$;

– weights for $F^{OS(BOSG)}(s_i, cX)$ function:

- $w_{CComp}^{OS(BOSG)} = 0.35$,
- $w_{ISComp}^{OS(BOSG)} = 0.15$,
- $w_{CCPerClean}^{OS(BOSG)} = 0.35$,
- $w_{CCCompUnCol}^{OS(BOSG)} = 0.15$.

2. IOSG concept:

– weights for $F^{IS(IOSG)}(s_i, cIn, cNext)$ function:

- $w_{LOcc}^{IS(IOSG)} = 0.2$,
- $w_{CDiv}^{IS(IOSG)} = 0.1$,
- $w_{LPrio}^{IS(IOSG)} = 0.1$,
- $w_{FSCcIn}^{IS(IOSG)} = 0.4$,
- $w_{wFSCcNext}^{IS(IOSG)} = 0.2$;

– weights for $F^{OS(IOSG)}(s_i, cX)$ function:

- $w_{CComp}^{OS(IOSG)} = 0.35$,
- $w_{ISComp}^{OS(IOSG)} = 0.15$,
- $w_{CCPerClean}^{OS(IOSG)} = 0.35$,
- $w_{CCCompUnCol}^{OS(IOSG)} = 0.15$.

Table 1 NC quality index values for 100-element instances

Dataset No	NCs		
	BSAG-BOSG	IOSG	mIOSG
Data_01	16	22	17
Data_02	15	37	20
Data_03	14	18	25
Data_04	21	25	24
Data_05	15	27	18

Table 2 ES quality index values for 100-element instances

Dataset No	ES		
	BSAG-BOSG (%)	IOSG (%)	mIOSG (%)
Data_01	71	79	71
Data_02	71	86	86
Data_03	79	79	86
Data_04	86	79	50
Data_05	64	64	79

3. mIOSG concept:

- weights for $F^{IS(mIOSG)}(s_i, cIn, cNext)$ function:

- $w_{LOcc}^{IS(mIOSG)} = 0.2$,
- $w_{CDiv}^{IS(mIOSG)} = 0.1$,
- $w_{LPrio}^{IS(mIOSG)} = 0.1$,
- $w_{FSCcIn}^{IS(mIOSG)} = 0.4$,
- $w_{FSCcNext}^{IS(mIOSG)} = 0.2$;

- weights for $F^{OS(mIOSG)}(s_i, cX)$ function:

- $w_{CComp}^{OS(mIOSG)} = 0.3$,
- $w_{ISComp}^{OS(mIOSG)} = 0.1$,
- $w_{CCPerClean}^{OS(mIOSG)} = 0.3$,
- $w_{CCompUnCol}^{OS(mIOSG)} = 0.1$,
- $w_{CDiv}^{OS(mIOSG)} = 0.1$,
- $w_{LPrio}^{OS(mIOSG)} = 0.1$.

Evaluation of solutions

The values of NC and ES indices assessing the results obtained using the proposed game theory approaches, for the set of 100-element instances, are summarized in Tables 1 and 2, respectively.

The analysis of the NC values shows that modification of the IOSG approach (extending the OS player payout with an additional two criteria, i.e. the criteria of color diversity

Table 3 NC quality index values for 1000-element instances

Dataset No	NCs		
	BSAG-BOSG	IOSG	mIOSG
Data_01*	149	153	169
Data_02*	165	156	160
Data_03*	168	164	168
Data_04*	171	165	154
Data_05*	174	175	164

Table 4 ES quality index values for 1000-element instances

Dataset No	ES		
	BSAG-BOSG (%)	IOSG (%)	mIOSG (%)
Data_01*	73	71	71
Data_02*	75	75	70
Data_03*	75	73	76
Data_04*	67	73	71
Data_05*	70	70	65

and line priority), in the case of the most tested instances, allows better results to be obtained in comparison with the unmodified model. The number of paint gun changeovers for the BSAG-BOSG algorithm is the smallest among all tested models. Based on the data contained in Table 2, it cannot be clearly determined which of the tested models is the best in terms of the ES indicator.

The values of NC (Table 3) and ES (Table 4) indicators for 1000-element instances are comparable for all tested models.

Evaluation of Nash equilibrium

In order to select the best algorithm from the proposed concepts, an additional analysis of occurrences of the following situations was performed: one Nash equilibrium (1-RN), multiple admissible Nash equilibria (N-RN, where $N \in \mathbb{C}$, $N > 1$) or no Nash equilibria (0-RN). Table 5 and Table 6 present the results of this analysis for 100-car and 1000-car samples.

Comparing all of the game theory approaches tested, it can be noted that the BSAG-BOSG algorithm is characterized by the largest number of decisions based on the occurrence of one Nash equilibrium in pure strategies, and this applies only to the game played at the buffer input (BSAG). No Nash equilibria in pure strategies occurs most often for the game played on the buffer output (BOSG). The mIOSG algorithm was considered the best of the existing game theory concepts because it has a similar number of decisions made based on

Table 5 The analysis of occurrences of situation 1-RN, N-RN and 0-RN for samples consisting of 100 cars

Dataset No	BSAG-BOSG						IOSG			mIOSG		
	BSAG			BOSG								
	1-RN (%)	N-RN (%)	0-RN (%)	1-RN (%)	N-RN (%)	0-RN (%)	1-RN (%)	N-RN (%)	0-RN (%)	1-RN (%)	N-RN (%)	0-RN (%)
Data_01	92	8	0	49	46	5	70	30	0	88	11	2
Data_02	93	7	0	69	28	2	45	55	0	78	21	2
Data_03	89	11	0	59	38	3	63	35	2	78	21	1
Data_04	88	12	0	59	37	4	71	29	0	75	25	1
Data_05	90	10	0	58	38	4	50	50	1	76	23	1

Table 6 The analysis of occurrences of the situation 1-RN, N-RN and 0-RN for samples consisting of 1000 cars

Dataset No	BSAG-BOSG						IOSG			mIOSG		
	BSAG			BOSG								
	1-RN (%)	N-RN (%)	0-RN (%)	1-RN (%)	N-RN (%)	0-RN (%)	1-RN (%)	N-RN (%)	0-RN (%)	1-RN (%)	N-RN (%)	0-RN (%)
Data_01*	96	4	0	56	39	39	5	68	0	83	17	1
Data_02*	97	4	0	56	37	37	7	67	0	83	17	0
Data_03*	98	2	0	56	37	37	7	67	0	82	17	1
Data_04*	93	7	0	54	40	40	6	64	0	82	17	1
Data_05*	90	10	0	55	38	38	7	65	0	82	17	1

one Nash equilibrium in pure strategies (similar to the BSAG model), while no Nash equilibria occurred in only 1–2% of decisions (much less than in the BOSG model, not much more than for the IOSG model).

The no Nash equilibrium in pure strategies occurred most often for the BSAG-BOSG model, and this applies only to the game played at the buffer output (BOSG). However, in the case of the buffer entry game (BSAG), the vast majority of decisions were made based on one Nash equilibrium in pure strategies, which is the most desirable solution. The best solution is the mIOSG model, because there were few situations where the Nash equilibrium in pure strategies was not found, and compared to the IOSG model, more decisions were made based on the one Nash equilibrium.

Based on the conclusions presented, it was found that it is not possible to clearly indicate which of the proposed game theory concepts is the best, taking into account the NC and ES indices. However, the mIOSG algorithm was chosen due to the largest number of 1-RN situations for the tested data.

Conclusions

In this paper the problem of production sequencing was analyzed; this is part of the production planning process that appears in the car production industry and is important both economically and ecologically. In order to take into account the demands on the industry, this problem has again been formulated as a multi-criteria optimization model. In this model, one goal reflects the traditional goal of minimizing the number of color changes associated with paint gun changeovers, while the other, very practical goal, related to synchronizing gun changeovers with necessary periodic cleanings, results from the principle of the paint systems used in car factories.

The game theory was proposed to solve this novel problem, which was significantly different from the ones considered in the literature to date. Three approaches, BSAG-BOSG, IOSG, and mIOSG were developed and they allowed for the incorporation of several important features. Among the most important ones there is the need to develop a method that allows for unloading and loading the buffer. However, it is important that there should be some kind of information feedback between the input and output algorithm so that input decisions are made taking into account the exit situation and vice versa. The algorithm analyzes the effects that the buffer

entry decision will have on its exit situation, and analogously, buffer exit decisions are made taking into account the entry situation. As a result, the decisions made are aimed at optimizing both indicators determining the quality of the output sequence as well as the buffer status. In the algorithm, this is implemented, e.g. by the criterion of free space for *cIn* and *cNext* cars. At this point, another important feature can be distinguished; the buffer entry decision is made taking into account information about both the car on the loading conveyor belt and the car on the position preceding this conveyor belt. This allows the elimination of the situation of blocking the favorable position from the perspective of the *cNext* car. The last of the important features of the algorithms developed is the inclusion of a production plan from the parent system; as a result the algorithms have the property of being able to keep up with the currently changing state of production. A production plan is created for a limited time period; usually 3–4 h. In the algorithm this was accomplished using the parameter, which is the priority of the line.

To test the proposed algorithms, a number of experiments were carried out using both real and test data. In addition, to select the best out of the developed approaches, an additional analysis was performed of occurrences of the following situations: one Nash equilibrium, multiple admissible Nash equilibria or no Nash equilibria. Based on the results obtained, the mIOSG algorithm was considered the best choice.

The algorithms proposed in this article, based on theory-based models, can be successfully applied in a real factory, the only problem could be the need to train employees (e.g. maintenance) in the parameterization of the developed approaches. However, these algorithms are characterized by simplicity of implementation, thanks to the possibility of using publicly available programming libraries to solve the problem of searching for Nash equilibria.

Future research will focus on the following aspects. During the first stage, experiments will be carried out verifying the impact of the weighted values present in the objective function on the quality of the solutions obtained. Currently these weights have been chosen arbitrarily. During the next stage an attempt will be made to optimize the operation of algorithms again. During the final stage, modification of the algorithms will be carried out, allowing them to be used for any buffer structure.

Acknowledgements This work has been supported by Silesian University of Technology statutory research funds and by Polish Ministry of Science and Higher Education under internal grants 02/040/BKM20/0006 and 02/040/BK_20/0002 for Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland.

Author contribution SB: manuscript preparation, data acquisition, optimization, and methodology selection, programming, data validation, AS: manuscript preparation and conceptualization, data validation, JK:

manuscript preparation and concept discussion. All authors contributed to the article and approved the submitted version.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Amini, H., Meunier, F., Michel, H., & Mohajeri, A. (2010). Greedy colorings for the binary paintshop problem. *Journal of Discrete Algorithms*, 8, 8–14.
- Andres, S. D., & Hochstättler, W. (2011). Some heuristics for the binary paint shop problem and their expected number of colour changes. *Journal of Discrete Algorithms*, 9, 203–211.
- Ayala, D., Wolfson, O., Xu, B., Dasgupta, B., & Lin, J. (2011). Parking slot assignment games. In *Proc. ACM SIGSPATIAL international conference on advances in geographic information systems (GIS 2011)* (pp. 299–308). ACM Press.
- Bysko, S., & Krystek, J. (2018). A game theory approach for solving the new concept of car sequencing problem. In *Proc. international conference on control, automation and robotics (ICCAR)* (pp. 531–535). <https://doi.org/10.1109/ICCAR.2019.8813372>
- Bysko, S., & Krystek, J. (2019). Follow-up sequencing algorithm for car sequencing problem 4.0. In *Proc. automation 2019. Advances in intelligent systems and computing* (pp. 145–154). Springer.
- Bysko, S., & Krystek, J. (2020). Automotive paint shop 4.0. *Computers & Industrial Engineering*, 139, 105546. <https://doi.org/10.1016/j.cie.2018.11.056>
- Cheng, J., Lu, Y., Puskorius, G., Bergeon, S., & Xiao, J. (1999). Vehicle sequencing based on evolutionary computation. *Evolutionary Computation*, 2, 1207–1214.
- Chew, T. L., David, J. M., Nguyen, A., & Tourbier, Y. (1992). Solving constraint satisfaction problems with simulated annealing: The car sequencing problem revisited. In *Proc. international workshop on expert system & their applications* (pp. 405–416).
- Codognet, Ph., & Diaz, D. (1996). Compiling constraints in clp(FD). *Journal of Logic Programming*, 27(3), 185–226. [https://doi.org/10.1016/0743-1066\(95\)00121-2](https://doi.org/10.1016/0743-1066(95)00121-2)
- Epping, T., Hochstättler, W., & Oertel, P. (2004). Complexity results on a paint shop problem. *Discrete Applied Mathematics*, 136, 217–226.
- Estellon, B., Gardi, F., & Nouioua, K. (2004). Large neighborhood improvements for solving car sequencing problems. *RAIRO Operation Research*, 40, 355–379.
- Fiat, A., & Woeginger, G. (1998). *Online algorithms—The state of the art*. Springer.

- Geffen, C. A., & Rothenberg, S. (2000). Suppliers And Environmental Innovation: The automotive paint process. *International Journal of Operations and Production Management*, 20(20), 166–186.
- Giannetti, C., & Essien, A. (2022). Towards scalable and reusable predictive models for cyber twins in manufacturing systems. *Journal of Intelligent Manufacturing*, 33, 441–455. <https://doi.org/10.1007/s10845-021-01804-0>
- Gravel, M., Gagne, C., & Price, W. L. (2005). Review and comparison of three methods for the solution of the car-sequencing problem. *Journal of the Operational Research Society*, 56(11), 1287–1295. <https://doi.org/10.1057/palgrave.jors.2601955>
- Hartmann, S.A., & Runkler, T.A. (2008). Online optimization of a color sorting assembly buffer using ant colony optimization. In *Operations research proceedings* (pp. 415–420). Springer.
- Jahren, E., & Achá, R. A. (2018). A column generation approach and new bounds for the car sequencing problem. *Annals of Operations Research*, 264(1–2), 193–211. <https://doi.org/10.1007/s10479-017-2663-4>
- Ko, S. S., Han, Y. H., & Choi, J. Y. (2016). Paint batching problem on M-to-1 conveyor systems. *Computers & Operations Research*, 74, 118–126.
- Krystek, J., & Bysko, S. (2019). The follow-up control of the body sequencing process at the paint shop. *Mechanik*, 92(7), 462–464.
- Leonard, R. J. (1995). From parlor games to social science: Von Neumann, Morgenstern, and the creation of game theory 1928–1944. *Journal of Economic Literature*, 33(2), 730–761.
- Moon, D. H., Kim, H. S., & Song, C. (2005). A simulation study for implementing color rescheduling storage in an automotive factory. *SIMULATION*, 81, 625–635.
- Moya, I., Chica, M., & Bautista, J. (2019). Constructive metaheuristics for solving the Car Sequencing Problem under uncertain partial demand. *Computers & Industrial Engineering*, 137, 106048. <https://doi.org/10.1016/j.cie.2019.106048>
- Nash, J. (1950). Equilibrium points in n-person games. In *Proc. of the national academy of sciences* (pp. 48–49), 36(1).
- Neveu, B., Trombettoni, G., & Glover, F. (2004). Id walk: A candidate list strategy with a simple diversification device. In *Proc. CP'2004* (pp. 423–437). Springer. https://doi.org/10.1007/978-3-540-30201-8_32
- Oluyisola, O. E., Bhalla, S., Sgarbossa, F., & Strandhagen, J. O. (2022). Designing and developing smart production planning and control systems in the industry 4.0 era: A methodology and case study. *Journal of Intelligent Manufacturing*, 33, 311–332. <https://doi.org/10.1007/s10845-021-01808-w>
- Parello, B. D., Kabat, W. C., & Wos, L. J. (1986). Job-shop scheduling using automated reasoning: A case study of the car sequencing problem. *Journal of Automated Reasoning*, 2(1), 1–42. <https://doi.org/10.1007/BF00246021>
- Solnon, C., Cung, V. D., Nguyen, A., & Artigues, C. (2008). The car sequencing problem: overview of state-of-the-art methods and industrial case-study of the ROADEF 2005 challenge problem. *European Journal of Operational Research*, 191(3), 912–927. <https://doi.org/10.1016/j.ejor.2007.04.033>
- Solnon, S. (2000). Solving permutation constraint satisfaction problems with artificial ants. In *Proc. ECAI'2000* (pp. 118–122). IOS Press.
- Spieckermann, S., Gutenschwager, K., & Voß, S. (2004). A sequential ordering problem in automotive paint shops. *International Journal of Production Research*, 42, 1865–1878.
- Stutzle, T., & Hoos, H. H. (2000). MAX-MIN ant system. *Future Generation Computer Systems*, 16, 889–914.
- Sun, H., Fan, S., Shao, X., & Zhou, J. (2015). A colour-batching problem using selectivity banks in automobile paint shops. *International Journal of Production Research*, 53, 1124–1142.
- Sun, H., & Han, J. (2017). A study on implementing color-batching with selectivity banks in automotive paint shops. *Journal of Manufacturing Systems*, 44, 42–52.
- Thiruvady, D., Ernst, A. T., & Wallace, M. (2014). A Lagrangian-ACO Matheuristic for Car Sequencing. *EURO Journal on Computational Optimization*, 2(4), 279–296. <https://doi.org/10.1007/s13675-014-0023-6>
- Thiruvady, D., Meyer, B., & Ernst, A. (2011). Car sequencing with constraint-based ACO. In *Proc. of the 13th annual conference on genetic and evolutionary computation* (pp. 163–170). New York, USA, ACM.
- Valdondo, J.B., & Gude, J.P. (2007). Sequencing JIT mixed model assembly lines under station-load and part-usage constraints using lagrangean relaxations. In *Proc. 3rd multidisciplinary international conference on scheduling: Theory and applications (MISTA 2007)* (pp. 550–552).
- von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior*. Princeton University Press.
- Xu Y., & Zhou, J. G. (2016). A virtual resequencing problem in automobile paint shops. In *Proc. 22nd international conference on industrial engineering and engineering management 2015: Core theory and applications of industrial engineering* (pp. 71–80). Atlantis Press.
- Zhang, X., Gao, L., & Wen, L. (2018). A hybrid algorithm based on tabu search and large neighbourhood search for car sequencing problem. *Journal of Central South University*, 25, 315–330. <https://doi.org/10.1007/s11771-018-3739-2>
- Zufferey, N., Studer, M., & Silver, E.A. (2006). Tabu search for a car sequencing problem. In *Proc. 19th International Florida Artificial Intelligence Research Society conference (FLAIRS 2006)* (pp. 457–462). The AAAI Press. https://doi.org/10.1007/978-3-319-23350-5_8

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.