BRUNO HENRIQUE MIRANDA DOS SANTOS

Orientador: Marco Antonio Moreira de Carvalho

UM MÉTODO HEURÍSTICO APLICADO À PRODUÇÃO NA INDÚSTRIA AUTOMOTÍVA

Ouro Preto

Março de 2016

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

UM MÉTODO HEURÍSTICO APLICADO À PRODUÇÃO NA INDÚSTRIA AUTOMOTÍVA

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

BRUNO HENRIQUE MIRANDA DOS SANTOS

Ouro Preto
Março de 2016

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas
Graduate Program in Computing Science

Monograph presented to the Undergraduate Program in Computing Science of the Universidade Federal de Ouro Preto in partial fulfillment of the requirements for the degree of in Computing Science.

BRUNO HENRIQUE MIRANDA DOS SANTOS

Ouro Preto
March 2016

UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

Um Método Heurístico Aplicado à Produção na Indústria Automotíva

BRUNO HENRIQUE MIRANDA DOS SANTOS

Monografia defendida e aprovada pela banca examinadora constituída por:

Dr. Marco Antonio Moreira de Carvalho – Orientador
Universidade Federal de Ouro Preto

Dr. Alan Robert Resende de Freitas
Universidade Federal de Ouro Preto

Dr. Tiago Garcia de Senna Carneiro
Universidade Federal de Ouro Preto

Ouro Preto, Março de 2016

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Henry Ford developed a new way of producing cars in 1903, known as "Fordism", where each worker would have a very specific job to perform. This innovation reduced greatly the production time and cost taken per each unit to be produced. With the reduced cost and production time, it was possible to begin the mass production of items. Later on, it led to mixed production lines where a single line could produce more than one type of product, for instance, an automobile industry could build various types of cars from the same car body. According to Golle (2011), mass customization was created after that, leading to highly differentiate products. For instance, in terms of customization, Mercedes produced 1.1 million of cars in the plant Rasstatt in Germany between the years 2003 to 2005. From all those cars, only two were identical.

According to Boysen et al. (2005), at the beginning of each working day a sequence of cars need to be decided. It often requires that the parts needed are in stock and the time of each work station must be taken in consideration. For the parts, to be delivered in the production line, there are two options, they can be brought from the suppliers, which are *Just-in-Time* (JIT) and *Just-in-Sequence* (JIS). JIT brings the parts in the moment they are required, whereas JIS will pre order them, or change the order when they arrive, so that they will be on the order as they are needed on the production line. Another point is that JIS would require temporary storing, either in the factory or in the suppliers, also the time taken to sort the parts could increase the delivery time.

A work station is a worker or a machine that performs a job by installing an optional item. An optional item is something that the buyer selects when he is deciding which car is the best for his needs and what it needs to have. Some examples of optional items are leather seating, sun-roof, and air conditioning. The production line can have the work station in two ways, *physical* and *virtual*. In the *physical* mode, the cars are sequenced in order to follow the working times of the stations. Conversely, in the *virtual* mode the the work stations move along the production line and the cars do not need to be changed from the original sequence.

Events such as lack of parts or broken machines could lead to a scenario where the sequencing of cars need to be changed, or *resequenced*. This could lead to great monetary losses and should be avoided at all cost. Lack of parts can be avoided by creating a better sequence of cars to be produced by taking into consideration the inventory. Machine failure can be prevented by doing regular maintenance and by introducing redundancy in the production line, that is, having more than one machine, whenever possible, that can produce or do the job. This is a line of research known as production forecast, that tries to keep every equipment up and running and taking care of buying parts.

The French Society of Operations Research and Decision Analysis, ROADEF, is an international society that, amongst other attributions, organizes an international competition, bi yearly, focused on interacting the academy with companies in order to make the improvements made by the academy appear in companies for better production results. They also formalize the problem description and create sets of instances along the competing year. It also develops solution checkers so that the participants can see the final cost of their solution, whether it is in the right format or not and also validate constraints. This society often offers prizes, in the form of money for its participants, varying up to eleven thousand euros given in ROADEF 2016 conference. ROADEF began developing competitions on 1999 and repeast this process every two years. Each competition is focused on a different type of industry:

- ROADEF'99: Inventory Management Problem;
- ROADEF'01: Frequency Assignment Problem with Polarization constraints;
- ROADEF'03: Management of the Missions of the Earth Observation Satellites;
- ROADEF'05: Car Sequencing problem;
- ROADEF'07: Technicians and Interventions Scheduling for Telecommunications;
- ROADEF'09: Disruption Management for Commercial Aviation;
- ROADEF'10: A Large-Scale Energy Management Problem with varied constraints;
- ROADEF'12: Machine Reassignment;
- ROADEF'14: Rolling Stock Unit Management on Railways Sites; and
- ROADEF'16: Inventory Routing Problem.

This study will focus especifically on the ROADEF'05 with the subject Car Sequencing Problem.

Estellon and Gardi (2013) and Kis (2004) state that the Car Sequencing Problem (CS) is a problem where the company has to produce a determined amount of cars in a working day and the cars of this group have options to be added (e.g. radio, sun-roof, air-conditioning, etc.). When dealing with this kind of problem, the goal is to reduce the workload of each working station. The workload of these stations are often defined by the ratio of cars that they can install the option within the total amount of cars. For instance, a given station could install a particular option on three out of seven cars. It means that no sequence of cars can have more than three cars with that option to be installed, in a total amount of seven cars. Therefore, the cost of CS is usually associated with how many times this type of constraint is broken.

The objective of the Car Sequencing Problem is to determine the sequence of cars to be produced in order to reduce the costs of painting, as each change of colors costs a different amount of money. According to Boysen et al. (2012), it can vary from $27 on black up to $122 on amber. Another concern is that each customization can be done to only a few cars for each batch. Note that in this case some breaks might be required in order to remove the paint that have been accumulating in the nozzle tip.

In the case of ROADEF'05 there are three soft constraints, described in Chapter 3. ROADEF'05 established a partnership with car manufacturer company Renault to get the data from real factories in order to generate a more realistic approach.

Estellon and Gardi (2013) and Kis (2004) show that the Car Sequencing Problem is a NP-hard problem. This problem is being researched over the last twenty years in order to cut costs in the automobile industry. This study will devise a heuristic in order to try to outperform the best known results of the literature. The following 9 methods, amongst others, have been used to solve this problem in the past:

- Constraint Programming;
- Greedy;
- Integer Programming;
- Backtracking;
- Branch and Bound;
- Large Neighbourhood Search;
- Local Search;
- Genetic algorithms; and
- Ant Colony Optimization.

As can be seen below, the number of articles published and registered on web of science[1] have been increasing since this problem was proposed firstly by Parrello et al. (1986) as a

[1] https://apps.webofknowledge.com/ accessed on 26/01/2016

job-shop schedule problem. The peak on 2008 could be due to the fact that the ROADEF challenge on 2005 have drawn the academic community towards it.

Published articles per year

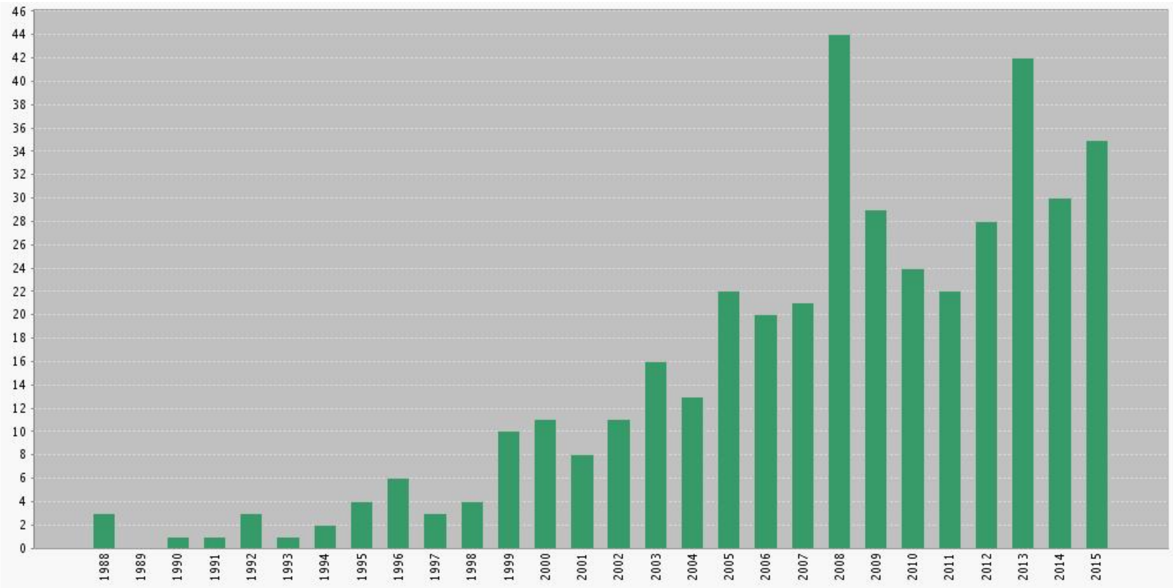

Figure 1.1: Published articles on car sequencing subject after 1987.

1.1 Motivation

There are three main motivations for this work. First, it is a NP-Hard problem. Second, it has been the subject of research by a great Society, ROADEF. Third, it has a big impact on the economy of Minas Gerais state and Brazil.

As mentioned before, it has been shown by Kis (2004) and Estellon and Gardi (2013) that this problem belongs to the class of NP-Hard Problems.

As stated before, on 2005 the French Society of Operations Research and Decision Analysis alongside with car maker Renault, selected this subject as its competition subject on 2005. ROADEF paid up to seven thousand euros as prize for the contestants. It counted with participants from numerous countries such as France, Brazil, Netherlands, Canada, Germany, Switzerland, Poland and Bosnia and Herzegovina.

According to ANFAVEA[2] the automotive industry participation on the overall brazilian gross domestic product was 5% and 23% on the particular industrial area on 2013. It generated around R$ 178,5 billions in taxes. Brazil is currently seventh largest car producer in the world.

[2]http://www.anfavea.com.br/50anos/60.pdf Acessed on 09/12/2015

1.2 Goals

- Review the literature on car sequencing;
- Devise a heuristic optimization method that fulfills all problem requirements;
- Implement the previously mentioned method;
- Research for polishing techniques;
- Evaluate the method with real data and publicly available test sets; and
- Compare the method with state of the art methods from the literature.

1.3 Work Organization

The rest of this work is organized as follows. Chapter 2 presents the literature review from years 1986 to 2012. Chapter 3 presents the basis of the problem with a mathematical description. Chapter 4 presents a small description of how the input and output works as well as planning for the algorithm development. Finally, chapter 5 draws the conclusions regarding this first version and points the future work.

Chapter 2

Literature Review

In this chapter the work done through the years 1986 to 2012 will be described. Parrello et al. (1986) were the first to describe the car sequencing problem on 1986. On 2005 ROADEF used the car sequencing as the problem of its biannual challenge, constituting a milestone on the problem history.

Parrello et al. (1986) described it as a case study of the job-shop scheduling problem. [REVER They described step by step how to reduce the it to the job-shop showing what is the schedule, the sequence of cars, and the attributes, the constraints for each option. They also showed that it might not have a solution, (i.e., more cars requiring the option than the factory can handle).] For these cases the authors created the idea of penalties to change the cost of a solution. Their main goal was to develop an environment to allow other researchers to tackle this problem using automated reasoning, where they state a set of rules and the software would decide the sequence of cars.

For instance, consider that an option has the ratio constraint 1 : 3 (i.e., for each 3 cars within a sequence, only one can have the option installed), and that the input requires 2 cars with that option. Table 2.1 shows, on the left hand side that regardless of the sequence the penalty would be 1. Another kind of penalty, applied when there are constraint violating cars close to each other, is shown on third and fourth lines of the right hand side of the table. The table represents a part of a car sequence, where Y means that a car has the option and N otherwise.

Table 2.1: Both types of penalties.

car	Y	N	Y	car	Y	N	Y
penalty	0	0	1	penalty	0	0	1
car	N	Y	Y	car	N	Y	Y
penalty	0	0	1	penalty	0	0	3

2.1 90's: Constraint Programming

In the early 90's the effort taken to solve the car sequencing problem focused mainly on Constraint Satisfaction methods. Hententyck et al. (1991), Hentenryck et al. (1992), A. Davenport and Zhu (1994), Lee et al. (1998) and Dincbas et al. (1999) developed new algorithms and techniques to improve the current constraint satisfaction solvers, such as the Constraint Handling in Prolog (CHIP). Also, operators to speed up the programming were developed. Due to hardware constraints of that time, most of the experiments conducted considered only small instances up to 400 cars, usually containing only 100 of them, and it would take up to 10 hours to process. The instances used were all randomly generated. Regin and Puget (1997) proposed preprocessing operations in order to remove possible bad values among the data.

After that, Smith (1996) started developing more advanced constraint satisfaction techniques in order to solve the car sequencing problem, but realised that it was not such a good idea. Nevertheless, this work posed interesting ideas:

- Cars with the same optionals would be counted as a class of cars, instead of a single car. If the number of optionals would grow this idea would be infeasible, because the number of possible cars grows exponentially to the number of options (2^O, where O is the number of options);
- The order of cars in the input could be changed so that the backtracking done by the solver would be less frequent and less heavy; and
- A technique for removing possible bad values before assigning them to a variable. A bad value is a value that might lead to a high cost solution.

Instances with up to five options were considered in the reported experiments, but as mentioned above, the results were discouraging.

2.2 00's: Metaheuristc, Heuristic and Exact Methods

Warwick and Tsang (1995) were the first to use a genetic algorithm with hill climbing (GAwHC). The GAwHC was compared to a Tabu Search and the reported results showed that the GAwHC performed better. However, the GAwHC presented some limitations, such as the maximum size of the input and the time taken to solve an instance.

In the early 00's much more diversity of methods was employed to address the car sequencing problem. Bergen et al. (2001) used multiple types of greedy, local search, backtracking, branch and bound methods, also combining them. Gottlieb et al. (2003) mixed various methods as well, but included the Ant Colony Optmization (ACO) metaheuristic. Perron and Shaw (2004) started using Large Neighbourhood Search to solve it. Neveu et al. (2004) developed IDWALK a local search that would do some changes to the solution and see which solution generated after that was the best one. Drexl et al. (2006) combined branch and bound with tabu search. It is not possible to compare them as they used different test cases of the problem.

On 2005 ROADEF used car sequencing as its biannual contest. Solnon et al. (2007) wrote an article to summarize the contest and the results. ROADEF made an explicit list of what was the problem, how a solution would be evaluated. It has also shown the problem difficulty, it's complexity, as well as some methods the participants used to solve it. Explained how the results were calculated and on the website [1] there is the possibility of sending new results for evaluation and comparison with old ones.

After 2005 much effort was taken in using ACO. The book Dorigo and Stützle (2005) shows various ways of implementing it as well as problems in which it has been used in the past. Some authors such as Solnon (2008) used random greedy initial solution and after that he used ACO to improve the results.

Gravel et al. (2005) compared three methods for solving the car sequencing problem, Constraint Satisfaction programming, Integer programming, and ACO. He used CSPLib test case and proposed a new way of calculating the penalties to remove double counting of cars. However this new calculation has not been found in any other paper.

Boysen et al. (2005) used dynamic programming and simulated annealing.

In recent years, 2009 and after, there was a slight difference in the formulation of the problem. Authors started using more than one line of production or removing the color change costs from the optimization, as it is a separated problem and sometimes it could be solved separately. Boysen et al. (2009), Golle et al. (2010) and Yazgan et al. (2011) all followed this line of research.

Boysen et al. (2012) surveyed various past publications that addressed the car sequencing problem and pointed what he thought to be the future research agenda of this area. This agenda involves tackling multi-objective optimization, resequencing the production line.

As sugested by Boysen et al. (2012) articles found after 2012 showed a much more specific and local definition of the car sequecing problem. Such as Mazur and Niederliński (2015a) and Mazur and Niederliński (2015b) where they take in account the makespam taken in order to produce the veihicles, or Yazgan et al. (2011) were they optimize more than one production line at the same time.

2.3 Problem Difficulty

Regarding the difficulty of the problem, Gent (1998) showed that it is NP-complete by reducing it to the Halmitonian Path. Kis (2004) and Estellon and Gardi (2013) later on showed it is a NP-hard problem.

[1] http://roadef.proj.info-ufr.univ-montp2.fr/2005/en/revival2005.php?Instance=All&Instances=All Asseced on 27/01/2016

Chapter 3

Theoretical Foundation

The Car Sequencing Problem is usually defined by a tuple (C, O, p, q, r) where C is the set of cars to be produced, O is the set of different optionals that can be added in the current factory, p limits the maximum amount of cars that can have a specific optional item installed within a sequence of size q (referred to as the ratio constraint, or $p : q$ constraint, from now on). The last element r determines for whether or not a specific car has an option installed:

$$r(c_i, o_j) = \begin{cases} 1 \text{ if car } i \text{ has option } j \\ 0 \text{ otherwise} \end{cases}$$

The ROADEF proposed a formula for calculating the cost of a solution which is a simplified version of the one proposed by Parrello et al. (1986). For each option $o_i \in O$ every subsequence of size q_i is going to be checked to see if there are more than p_i cars in that sub sequence that requires option o_i:

$$\sum_{i=1}^{O} \sum_{j=1}^{N} \begin{cases} (\text{count}(j, j+q_i) - p_i) * w_i \text{ if count}(j, j+q_i) - p_i > 0 \\ 0 \text{ otherwise} \end{cases} \tag{3.1}$$

Where:

- O is the number of options;
- N is the number of cars in the sequence;
- w_i is the given weight associated with option o_i;
- q_i is size of the sub sequence to be analyzed and p_i is the maximum numbers of cars that can be in that sequence without having penalties; and
- *count* counts how many cars in the sub sequence j to $j + q_i$ have option i installed.

$$\sum_{j=1}^{N} \begin{cases} w_c \text{ if color}(j) \neq \text{color}(j-1) \\ 0 \text{ otherwise} \end{cases} \tag{3.2}$$

Where w_c is the weight associated with changing colors.

Then the final cost of the solution then is given by:

$$\text{cost} = (3.1) + (3.2) \tag{3.3}$$

3.1 Input and Output Examples

Table 3.1 exemplifies how an input could be. In the example, four types of cars need to be produced and may have up to five optional items. The number of cars required of each type is in the bottom of the table. Each option has its *p:q* constraint, that limits the maximum amount of cars, p, that can have the option in any given sequence of size q. The colors were added from the original table in order to be more similar to the ROADEF'05 problem and only two colors were added. A car type is defined by which optionals it has and its color. This could be the cars solicited to be built in a day in a factory. In this example the cost of violating an constraint is assumed to be 10 and the cost of changing color is 1.

Example Input

Opt.	Car types				Constr.
	1	2	3	4	p : q
1	✓			✓	3:5
2		✓			1:3
3	✓	✓			2:5
4			✓	✓	2:3
5			✓		1:4
Color	Black	Blue	Black	Blue	
Req.#	2	2	2	2	

Table 3.1: Adapted from Kis (2004)

The solution for this problem is a sequence of cars that fulfills the constraints and has the lowest cost of changing colors, or in case it is not possible to fulfill the constraints, has the lowest cost associated to this fault. According to Kis (2004), there are three possible solutions for this problem, which are (2; 3; 1; 4; 4; 2; 3; 1), (1; 3; 2; 4; 4; 1; 3; 2), and (1; 3; 2; 4; 4; 2; 3; 1) which would have costs of 3, 3 and 2 respectively. This similarity in the cost is due to the fact that all solutions do not violate the constraints, therefore only the cost of changing colors was applied. The last solution is optimal since there are two colors at least one change would be required, however the last constraint has the ratio 1 : 4 and the car type 3 must have more than three cars between each production in order to not violate this constraint.

Violating the constraint to maintain only one change of color would have a much higher value, at least eleven, since violating constraints are worth ten times a change of colors.

Example Output

Opt.	Car								Constr.
	1	3	2	4	4	2	3	1	p : q
1	✓			✓	✓			✓	3:5
2			✓			✓			1:3
3	✓		✓			✓		✓	2:5
4		✓		✓	✓		✓		2:3
5		✓					✓		1:4
Color	Black	Black	Blue	Blue	Blue	Blue	Black	Black	
Req.#	2	2	2	2	1	1	1	1	

Table 3.2: Optimal detailed result previously cited

As can be seen on table 3.2 the sub-sequene (1; 3; 2; 4; 4) follows the rules for the first constraint. In this sub-sequence of size 5, there is at maximum three cars that require the first optional item(1; 4; 4). The same can be seen for all other sub-sequences.

Chapter 4

Methodology

The input is composed of four files:

optimization_objectives contains how are the priorities for this instance, for example, if the cost of changing colors is higher than the cost of not respecting the $p : q$ constraints;

paint_batch_limit is how many cars can be painted at the maximum before a color change is required;

ratios is the definition of $p : q$ constraints

vehicles is the description of each vehicle required in the current planning period

Constraints of the type $p : q$ are divided into two groups: *priority* constraints and *non priority* constraints. The difference between them is that the penalty given when more than p cars are in a subsequence of size q the weight will always be higher on a priority constraint. The number of color changes also result in a penalty.

Regarding the penalties for this problem, ROADEF'05 defined three categories of penalties, which are priority constraints, non priority constraints and color change. The value associated with a penalty is defined in the input and represented by a permutation of $\{1, 10^3, 10^6\}$.

In this work, we model the problem using graphs, where each vertex v represents a car and an edge connects v to w if the car w could be sequenced after car v.

As can be seen on 4.1, it is a graph where every car can be sequenced after the others (i.e., there is no $p : q$ constraint of 1 in 2 or more).

To solve the problem an Hamiltonian path would be required, since every car needs to be on the sequence. However, the associated decision problem is NP-Complete. Adding weights to the edges models the problem more realistically. For example it can represent the cost of color changing between cars. Notwithstanding, it defines the Traveling Salesman Problem, also a NP-Hard problem.

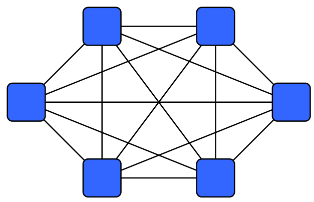

Figure 4.1: Example graph.

A quicker way to find a solution would be doing a Depth-First Search (DFS), and at the end of the procedure the order of the vertices visited would be the cars sequenced. This does not grant to have a good, or even feasible solution, since the edges only take in consideration only if the next car can be sequenced after the current. However, it can be a good start for a heuristic method, giving good insights on the solution.

The idea to use graphs to solve this problem comes from the literature review, since surprisingly no other author has used it like to solve this problem.

4.1 Future Work

The future work done will follow the following plan.

Table 4.1: Future tasks

Project Steps				
	Months			
Task	1	2	3	4
1. Implement the Heuristic	✓			
1.1 Refine the Heuristic	✓			
2. Revise the literature review	✓			
3. Run experiments		✓		
3.1 Improve the Heuristic		✓		
4. Compare with other methods			✓	
5. Submit to ROADEF'05				✓

Task 1 is the implementation of the code itself. Task 2 is to revise to see if any new paper has been published in the area. Task 3 is to run the experiments and make any improvement that is observed from the experiments ran. Task 4 is to see how well the heuristic is doing. Task 5 is to submit the results to ROADEF'05 in order to be acknowledged of our work done.

Chapter 5

Conclusions

The car sequencing problem showed to be an interesting problem, in terms of difficulty and what can be developed. It was concluded that the knowledge required to devise an heuristic and build an algorithm to solve the car sequencing problem.

Future work will be done as follows.

- Devise an heuristic;
- Implement the before mentioned heuristic; and
- Compare the aforementioned heuristic with other literature methods.

Bibliography

C. Wang A. Davenport, E. Tsang and K. Zhu. Genet: a connectionist architecture for solving constraint satisfaction problems by iterative improvement. *In Proceedings of AAAI'94*, pages 325–330, 1994.

M.E. Bergen, P. van Beek, and T. Carchrae. Constraint-based vehicle assembly line sequencing. *Proceedings of the 14th Canadian Conference on Artificial Intelligence*, 2056:88–99, 2001.

N. Boysen, M. Fliedne, and A. Scholl. Level scheduling for batched jit supply. 2005.

N. Boysen, M. Fliedne, and A. Scholl. Sequencing mixed-model assembly lines: Survey, classification and model critique. *Elsevier*, 2009.

N. Boysen, A. Scholl, and N. Wopperer. Resequencing of mixed-model assembly lines: Survey and research agenda. *Elsevier*, pages 594–604, 2012.

M. Dincbas, H. Simonis, and P. Van Hentenryc. Solving constraint satisfaction sequencing problems. 1999.

M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2005.

A. Drexl, A. Kimms, and L. Matthießen. Algorithms for the car sequencing and the level scheduling problem. *Springer Science*, pages 153–176, 2006.

B. Estellon and F. Gardi. Car sequencing is np-hard: a short proof. *Journal of the Operational Research Society*, 64(10):1503–1504, 2013.

I.P. Gent. Two results on car-sequencing problems. *APES*, 1998.

Uli Golle. *On the Car Sequencing Problem: Analysis and Solution Methods*. Phd thesis, Johannes Gutenberg-Universität Mainz, Fachbereichs Rechts- und Wirtschaftswissenschaften, 2011.

Uli Golle, Nils Boysen, and Franz Rothlauf. Analysis and design of sequencing rules for car sequencing. *Elsevier*, 2010.

J. Gottlieb, M. Puchta, and C. Solnon. A study of greedy, local search and ant colony optimization approaches for car sequencing problems. *Applications of evolutionary computing*, 2611:246–257, 2003.

M. Gravel, C. GagnÃ©, and W. L. Price. Review and comparison of three methods for the solution of the car sequencing problem. *Operational Research Society*, 2005.

P. V. Hentenryck, H. Simonis, and M. Dincbas. Performance of a comprehensive and efficient constraint library using local search. *Artificial Intelligence 58(1-3)*, pages 113–159, 1992.

P. V. Hententyck, V. Saraswat, and Y. Deville. The cardinality operator: a new logical connective and its application to constraint logic programming. *ICLP-91*, 1991.

Tam'as Kis. On the complexityof the car sequencing problem. *IEEE Transaction on Consumer Electronics*, 53(3):1186–1194, August 2004.

J.H.M. Lee, H.F. Leung, and H.W. Won. Performance of a comprehensive and efficient constraint library using local search. *11th Australian JCAI, LNAI. Springer Verlag*, 1998.

M. Mazur and A. Niederliński. A two-stage approach for an optimum solution of the car assembly scheduling problem part 1. problem statement, solution outline and tutorial example. *Archives of Control Sciences*, 25:355–365, 2015a.

M. Mazur and A. Niederliński. A two-stage approach for an optimum solution of the car assembly scheduling problem. part 2. clp solution and real-world example. *Archives of Control Sciences*, 25:367–375, 2015b.

B. Neveu, G. Trombettoni, and F. Glover. Id walk: A candidate list strategy with a simple diversification device. *Springer Science*, 2004.

Bruce D Parrello, Waldo C. Kabat, and L. Wos. Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem. *Journal of Automated Reasoning*, 2 (10):1–42, 1986.

L. Perron and P. Shaw. Combining forces to solve the car sequencing problem. *Proceedings of CP-AI-OR'2004*, 3011:225–239, 2004.

J.-C. Regin and J.-F. Puget. A filtering algorithm for global sequencing constraints. *LNCS*, 1330(3):32–46, 1997.

B. Smith. Succeed-first or fail-first: A case study in variable and value ordering heuristics. *third Conference on the Practical Applications of Constraint Technology PACT'97*, pages 321–330, 1996.

Christine Solnon. Combining two pheromone structures for solving the car sequencing problem with ant colony optimization, 2008.

Christine Solnon, Van Dat Cung, Alain Nguyen, and Christian Artigues. 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 (EJOR)*, 2007.

T. Warwick and E. Tsang. Tackling car sequencing problems using a genetic algorithm. *Journal of Evolutionary Computation - MIT Press*, 3:267–298, 1995.

H. R. Yazgan, I. Beypinar, S. Boran, and C. Ocak. A new algorithm and multi-response taguchi method to solve line balancing problem in an automotive industry. *Springer*, 2011.