Using Online Bees Algorithm for Real-time Permutation Flow Shop Problem in Car Disassembly Line

murat sahin (msahin@roketsan.com.tr)
Roketsan Roket Sanayii ve Ticaret AS https://orcid.org/0000-0002-3659-3528
Wang Chao
Mechanical and Electronic Products Changshu

Research Article

Keywords: Online Bees Algorithm, Disassembly, Permutation Flow Shop Problem

Posted Date: March 15th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-286326/v1

Abstract

The automobile disassembly line is a typical permutation flow shop problem (PFSP). In the automobile disassembly line, the goal of the research is to minimize the completion time of scheduling with optimizing the disassembly order of vehicles. However, disassembly time was uncertain at the initial stage as the consuming time of cars' specific sequence only could be roughly estimated, which make the PFSP in automobile disassembly line unlike the traditional ones. In this study, the real-time PFSP problem in automobile disassembly line was defined and well solved with the proposed online Bees Algorithm (O-BA). The algorithm has been prepared in Matlab/Simulink to work in real-time. Time consumed by each component of each vehicle was roughly estimated based on engineer's experience. First optimisation was carried out to decide the disassembly order of vehicles to be disassembled. When each component was disassembled, the real consuming time value was updated with the detecting system which was realized in Simulink. Then the O-BA was activated and created the new solution for the disassembly order of vehicles which were still not entering the disassembly line. The proposed O-BA algorithm has an online structure and conducted the optimisation with distinguishing whether the vehicles to be disassembled were entering the disassembly line. The result shows the O-BA with the detecting system was succeeded in realizing real-time PFSP in the disassembly line. Moreover, the method proposed in this study was suitable for solving the same kind of real-time PFSP in the assembly line or disassembly line.

1. Introduction

The automobile disassembly line is a typical permutation flow shop problem (PFSP). The PFSP in one of the most studied optimisation problems in the literature. It could be understood as a single flow shop with consistent process which n jobs must be processed on a shop of m machines following the same order. With finding the most appropriate sequence of n jobs, the problem of optimisation could be resolved. One of the most popular optimisations of PFSP was minimization of makespan [1] and initially proposed by Johnson [2]. Many literature also set the criteria as minimization of total flow time [3], or minimization of total tardiness [4]. In the past decades, one-time optimisation of sequence was acknowledged in most literature on optimisation of PFSP. However, the "real-time" means rescheduling the sequence of jobs became more reverent and meaningful at the age of Industry-4.0 [5]. Rahman et al. defined the real-time PFSP as customer orders are placed at random time intervals and re-optimisation as each new order arrives has important implications [6]. Ghaled and Taghipour et al. concluded the real-time scheduling (RTS) has the concept that refers to optimisation models which can utilize real-time data to update affected schedules without disrupting the operations of the manufacturing system [5]. In the automobile disassembly line, the real processing time for each vehicle's removing part could not be accurately measured, which will lead to the imbalance of overall production and low production efficiency. In the serial-flow car disassembly line [6], one-time optimisation of sequence has no help for actual production. Generally, each car's time consuming of removing part could only be roughly estimated and sometimes it has great changes depending on aging degree. Thus, rescheduling should be conducted after each car's part was removed and updated the information in the system. In this study, the real-time PFSP pay attention to the serial-flow car disassembly line which has the characteristics of uncertain operation time and could be expanded to similar assembly line or disassembly line.

Disassembly processes are discrete optimisation problems and graphical and matrix methods have been preferred for many years in their solution. Nowadays, metaheuristic algorithms are generally preferred [7]. Some prominent studies when looking at the applications in recent years; disassembly sequence planning with a hybrid method integrating fuzzy simulation and artificial bee colony [8], flexible process planning with a genetic algorithm for Product Recovery Optimization [9], a disassembly sequence planning with multiobjective ant colony algorithm for CNC machine tools [10], a tensorial memetic algorithm for disassembly sequence planning in product refurbishment [11], planning for disassembly using discrete artificial bee colony [12], a disassembly sequence planning with genetic algorithm for hydropower station [7], robotic disassembly sequence planning using discrete bees algorithm [13], an improved multi-objective discrete bees algorithm for robotic disassembly [14], collaborative optimisation of robotic disassembly sequence planning using improved discrete bees algorithm [15].

It is seen that the Bees Algorithm is used successfully in disassembly process optimisation. The Bees Algorithm has successful applications against other combinatorial problems. Some of the highlights from the few applications; single machine scheduling problem [16], combinational circuit design [17], PCB assembly optimisation [18], vehicle routing problem [19], examination

timetabling problems [20], a multi-objective supply chain optimisation [21, 22], disassembly sequence planning of human-robot collaboration in remanufacturing [15, 23]. In this study, the Bees Algorithm was designed within the scope of car disassembly process planning. The disassembly process, unlike the assembly process, includes many uncertain situations. Due to uncertain situations, some operations may take longer than expected during disassembly. In this case, better process time can be determined by changing the sequence by re-optimisation [16]. For this reason, two different Bees Algorithms have been designed within the scope of the study. The first algorithm makes the first optimisation for the initial values. The second algorithm performs online optimisation while the process is continuing if there is a change in the processes.

2. Online Disassembly Process Planning

In this study, online planning of the autonomous vehicle disassembly line was realized. The disassembly line consists of 14 stations, each with different disassembly jobs. Details of the process are given below:

- The disassembly process of a vehicle starts from the first station and travels through all stations in turn. The disassembly process is completed at the 14th station.
- After the first vehicle leaves the first station, the second vehicle enters the first station. Likewise, the vehicles are taken to the disassembly line one after the other, and when its process finished they switch to the other station.
- As soon as a vehicle enters the line, it has to enter all stations sequentially, it cannot switch between lines that are not included in the order.
- The initial disassembly times was decided by the engineer’s experience. In the process that will start with a large number of tools, the first planning will be carried out to this first known information.
- Disassembly periods can be completed at different times than the first determined times depending on the conditions of the vehicles (In cases where the disassembly of the connection materials takes longer or shorter due to rust, breakage, etc.).
- After the vehicles exit each station, the disassembly time of that vehicle and station is updated.
- After each update at the stations, online re-planning is made for vehicles that were still waiting for entering the disassembly line. Even the order of the vehicles starting the process is not changed.

Information on the Autonomous vehicle disassembly line is in Fig. 1 and the initial information on the process is given in Table 1.

3. The Online Bees Algorithm

The Bees Algorithm was inspired by the honey bee swarm searching for flower groups in nature by Pham et al. [24]. Scout bees search randomly in nature and when they find efficient resources, they come to the hive and transfer the information to other bees with a special dance (waggle dance). Guidance with dance is at the basis of the Bees Algorithm. The local search section is divided into two parts which are elite sites and selected sites. In these sections, many searches are made in the neighborhood of optimum solutions, similar to doing guidance with real bees dance. In addition, random searches continue in the global search section, thus avoiding local optimum values [25]. The problem in this study is in combinatorial structure. In optimisation algorithms prepared for combinatorial problems, search sections and neighborhood definitions are different from continuous problems. In general, the positions of the members in the problem are changed according to certain rules, and new sequences are tried to be found with less cost. For the Bees Algorithm, Ismail et al. [26] suggested three different operators in the local search section, namely, Swap, Insertion, and Inversion.

The pseudo code of the Bees Algorithm and the pseudo code of the process optimisation section is given below together as shown in Fig. 2. As can be seen, there are 3 different operator selections within the general Bees Algorithm structure. In each iteration, one of these operators is applied randomly. Within the Bees Algorithm structure, more searches are made in elite sites. (Sections 21-36 are the same as sections 5-20, nsp come instead of nep.)

In the swap process, the order of two different vehicle groups chosen at random is changed with each other [26]. As an example in Fig. 3, changing the order of vehicle groups numbered 2 and 4 is shown. In the insertion process, the order of a randomly selected vehicle group is inserted to a different rank, also randomly selected [16]. It is shown in Fig. 3. As can be seen, the location

of the number 2 vehicle group on the first plan is inserted into the section after the number 4 group. In the reversion process, the order between two randomly determined vehicle groups is reversed [16]. It is shown in Fig. 3. The order between vehicle groups number 2 and 5 on the first route has been reversed.

The algorithm parameters used in this study are given in Table 1.

Table 1 Parameter of the algorithms

Parameter	Value
Number of iterations	100
Number of scout bees (n)	40
Number of elite sites (e)	3
Number of selected sites (m)	9
Number of elite bees (nep)	60
Number of selected bees (nsp)	20

The general model consists of two parts. In the first part, the process plan is determined by optimising the starting vehicle and station values and the process begins. In the second part, online optimisation is done. When changes in station times are detected, a re-planning is made for vehicles that are not available on the line, and this process is repeated every time it is updated. A general view of the model is given in Fig. 4. The model consists of 4 main parts.

1. Getting the starting data of the process.
2. First optimisation.
3. The section that follows the time updates at the stations.
4. Online optimisation

The last column of the “Disassembly Times of Vehicles” matrix shows the vehicle numbers. After the data in the matrix is taken, the algorithm given in Fig. 2 is run and the first optimisation is made. The processes start according to the process sequence determined by this optimisation. The order of the first optimisation is shown with green blocks in Fig. 4. The first one shows the order of the vehicles, and the second shows the total duration times in the stations. After each operation in the stations, the times for the relevant station are updated in the “Detect of Changing Parameters” block. When updated, this block triggers the “Online Optimisation” block. In the online optimisation section, cars that are not on-line are determined first. The planning of these cars is re-optimised. Since the number of non-line cars is variable, the number of local search zones will also change dynamically. A section has been added to the first part of the algorithm to identify offline vehicles. By controlling the processing time, the place of the last online tool on the line is determined in the table. The flow chart of the online algorithm is given below.

The algorithm is developed in Matlab/Simulink. Information on the Simulink model is given below.

<u>Model Configuration</u>

Solver type : Fixed step

Solver : ode 3 (Bogacki-Shampine)

Fixed-step size : 1 [s]

Initial information on the process is given in Table 2.

Table 2 Initial information on the process

	Rough inspected disassembly cost of vehicle parts at 14 workstations (min)													
	WS1	WS2	WS3	WS4	WS5	WS6	WS7	WS8	WS9	WS10	WS11	WS12	WS13	WS14
Car1	8	8	8	8	2	8	4	2	8	8	8	6	8	8
Car2	4	4	8	8	4	8	4	6	8	6	4	8	6	6
Car3	4	8	8	8	8	8	4	8	4	8	4	6	8	8
Car4	8	8	4	8	4	8	8	8	8	8	8	8	8	8
Car5	4	8	8	8	8	8	4	8	4	8	4	6	8	8
Car6	8	8	4	8	4	8	8	8	8	8	8	8	8	8
Car7	4	2	8	8	8	8	4	2	2	8	4	6	2	8
Car8	8	8	4	8	4	8	6	8	8	8	8	8	8	8
Car9	8	2	4	4	4	4	8	4	8	8	6	8	8	8
Car10	6	4	8	8	8	4	4	8	4	6	8	8	8	8
Car11	8	8	2	8	8	8	8	4	2	2	4	2	2	8
Car12	4	8	6	6	8	8	4	8	4	8	4	6	8	8
Car13	6	4	8	8	6	4	4	8	4	6	8	8	8	8
Car14	6	4	8	8	8	4	4	8	6	8	8	8	8	8
Car15	8	8	8	8	8	8	8	4	8	2	4	8	8	8
Car16	8	8	4	4	4	4	8	4	4	4	8	8	8	8
Car17	8	2	4	4	4	4	8	4	8	8	8	8	8	8
Car18	6	4	8	8	8	4	4	8	4	8	8	8	8	8
Car19	8	8	8	8	8	8	8	4	8	2	4	8	8	8
Car20	8	8	4	4	8	4	8	4	2	8	8	8	6	8

4. Results Of The Proposed Case Study

The results of the simulations made with the prepared model are shown as Fig. 6. As a result of the first optimisation, the ranking of the vehicles was found as follows.

7 - 13 - 12 - 16 - 10 - 18 - 9 - 5 - 17 - 2 - 3 - 11 - 15 - 20 - 14 - 6 - 19 - 8 - 1 - 4

As a result of the first optimisation, the total duration of the process was calculated as 230+8=238 minute. As shown in Fig. 6. b., the start time of the last process is 230 and the last 4th vehicle enters the process. At the 36th minute of the process, the time of the 5th vehicle (Vehicle number is 10) in the 3rd station is updated by +10 minutes. The process plan of this moment of the process and the table containing the total times in the stations are given below.

As can be seen from the Fig. 7, the first 7 cars even started the process when there was an update. Therefore, planning will be done for later vehicles. As can be seen from the table, with the updated plan, a plan that will give a shorter process time for the first 7 vehicles was created. As a result of the optimisation, the new ranking of the vehicles was as follows.

7 - 13 - 12 - 16 - 10 - 18 - 9 **- 2 - 5 - 1 - 20- 11 - 14 - 17 - 8 - 6 - 4 - 3 - 19 - 15**

After online optimisation, the total time for the new order was found to be 238+8=246. Note that despite a 10-minute delay, a ranking with a delay of 8 minutes compared to the previous process was found with the new optimisation. The order has been

changed for the minimum process time according to the processing times of the vehicles in the stations.

5. Conclusion

This paper has analyzed the real time permutation flow shop problem in car disassembly line. As the initial of disassembly times were rough inspected depends on engineers' experience. Hence, one time optimisation doesn't make sense help to decrease the total working time. Bees Algortithm and simulink model were both well adopted in solving the problem in this study. In the future, control hardware will be ready to design with applying O-BA algorithm and test the effectiveness of online optimisation in the factory.

(1) Swap, insertion, and inversion were performed with the local searching. Comparing to the case without real-time scheduling, the processing time was reduced by 2 mins. Although only one processing time of a vehicle's part was updated to the effectiveness of O-BA algorithm, a huge potential on minimization of makespan in actual production line was observed.

(2) The proposed O-BA algorithm has an online structure and conducted the optimisation with distinguishing whether the vehicles to be disassembled were entering the disassembly line. In the actual production line, sensors can be used to achieve controlling and scheduling for online optimisation.

Declarations

Funding: This work was funded by Open Fund Project of Jiangsu Key Laboratory of Recycling Technology for Mechanical and Electrical Products [grant no. RRME201906].

Conflicts of interest/Competing interests: We know of no conflicts of interest associated with this publication, and there has been no significant financial support for this work that could have influenced its outcome. As Corresponding Author, I confirm that the manuscript has been read and approved for submission by all the named authors.

Availability of data and material: There are some data in the Simulink model (Supplementary material).

Code availability: Algorithm codes and Simulink Model have been uploaded into the system (Supplementary material).

Ethics approval: We declare that this manuscript is original, has not been published before, and is not currently being considered for publication elsewhere.

Consent to participate: Not applicable.

Consent for publication: Not applicable.

References

1. Fernandez-Viagas V, Framinan JM (2014) On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. Computers & Operations Research 45:60–67.
2. Pan Q-K, Ruiz R (2013) A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. Computers & Operations Research 40:117–128.
3. Fernandez-Viagas V, Framinan JM (2017) A beam-search-based constructive heuristic for the PFSP to minimise total flowtime. Computers & Operations Research 81:167–177.
4. Khare A, Agrawal S (2020) Effective heuristics and metaheuristics to minimise total tardiness for the distributed permutation flowshop scheduling problem. International Journal of Production Research 1–17.
5. Ghaleb M, Zolfagharinia H, Taghipour S (2020) Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns. Computers & Operations Research 123:105031.
6. Rahman HF, Sarker R, Essam D (2015) A real-time order acceptance and scheduling approach for permutation flow shop problems. European Journal of Operational Research 247:488–503.

7. Li B, Li C, Cui X, et al (2020) A Disassembly Sequence Planning Method With Team-Based Genetic Algorithm for Equipment Maintenance in Hydropower Station. IEEE Access 8:47538–47555.
8. Tian G, Zhou M, Li P (2017) Disassembly sequence planning considering fuzzy component quality and varying operational cost. IEEE Transactions on Automation Science and Engineering 15:748–760.
9. Feng Y, Gao Y, Tian G, et al (2018) Flexible process planning and end-of-life decision-making for product recovery optimization based on hybrid disassembly. IEEE Transactions on Automation Science and Engineering 16:311–326.
10. Feng Y, Zhou M, Tian G, et al (2018) Target disassembly sequencing and scheme evaluation for CNC machine tools using improved multiobjective ant colony algorithm and fuzzy integral. IEEE Transactions on Systems, Man, and Cybernetics: Systems 49:2438–2451.
11. Pistolesi F, Lazzerini B (2019) TeMA: a tensorial memetic algorithm for many-objective parallel disassembly sequence planning in product refurbishment. IEEE Transactions on Industrial Informatics 15:3743–3753.
12. Tian G, Ren Y, Feng Y, et al (2018) Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony. IEEE Transactions on Industrial Informatics 15:2456–2468.
13. Liu J, Zhou Z, Pham DT, et al (2018) Robotic disassembly sequence planning using enhanced discrete bees algorithm in remanufacturing. International Journal of Production Research 56:3134–3151.
14. Liu J, Zhou Z, Pham DT, et al (2018) An improved multi-objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing. The International Journal of Advanced Manufacturing Technology 97:3937–3962.
15. Liu J, Zhou Z, Pham DT, et al (2020) Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete Bees algorithm in remanufacturing. Robotics and Computer-Integrated Manufacturing 61:101829.
16. Packianather MS, Yuce B, Mastrocinque E, et al (2014) Novel Genetic Bees Algorithm applied to single machine scheduling problem. IEEE, pp 906–911.
17. Mollabakhshi N, Eshghi M (2013) Combinational circuit design using bees algorithm. IEEE, pp 1–4.
18. Mei CA, Pham DT, Anthony JS, Kok WN (2010) PCB assembly optimisation using the Bees Algorithm enhanced with TRIZ operators. IEEE, pp 2708–2713.
19. Alzaqebah M, Jawarneh S, Sarim HM, Abdullah S (2018) Bees algorithm for vehicle routing problems with time windows. International Journal of Machine Learning and Computing 8:234–240.
20. Abdullah S, Alzaqebah M (2013) A hybrid self-adaptive bees algorithm for examination timetabling problems. Applied Soft Computing 13:3608–3620.
21. Yuce B, Mastrocinque E, Lambiase A, et al (2014) A multi-objective supply chain optimisation using enhanced Bees Algorithm with adaptive neighbourhood search and site abandonment strategy. Swarm and Evolutionary Computation 18:71–82.
22. Lambiase A, Iannone R, Miranda S, et al (2016) Bees algorithm for effective supply chains configuration. International Journal of Engineering Business Management 8:1847979016675301.
23. Xu W, Tang Q, Liu J, et al (2020) Disassembly sequence planning using discrete Bees algorithm for human-robot collaboration in remanufacturing. Robotics and Computer-Integrated Manufacturing 62:101860.
24. Castellani M, Otri S, Pham DT (2019) Printed circuit board assembly time minimisation using a novel Bees Algorithm. Computers & Industrial Engineering 133:186–194.
25. Zhou Z, Xie Y, Pham D, et al (2016) Bees Algorithm for multimodal function optimisation. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 230:867–884.
26. Ismail AH, Hartono N, Zeybek S, Pham DT (2020) Using the Bees Algorithm to solve combinatorial optimisation problems for TSPLIB. IOP Publishing, p 012027.

Figures

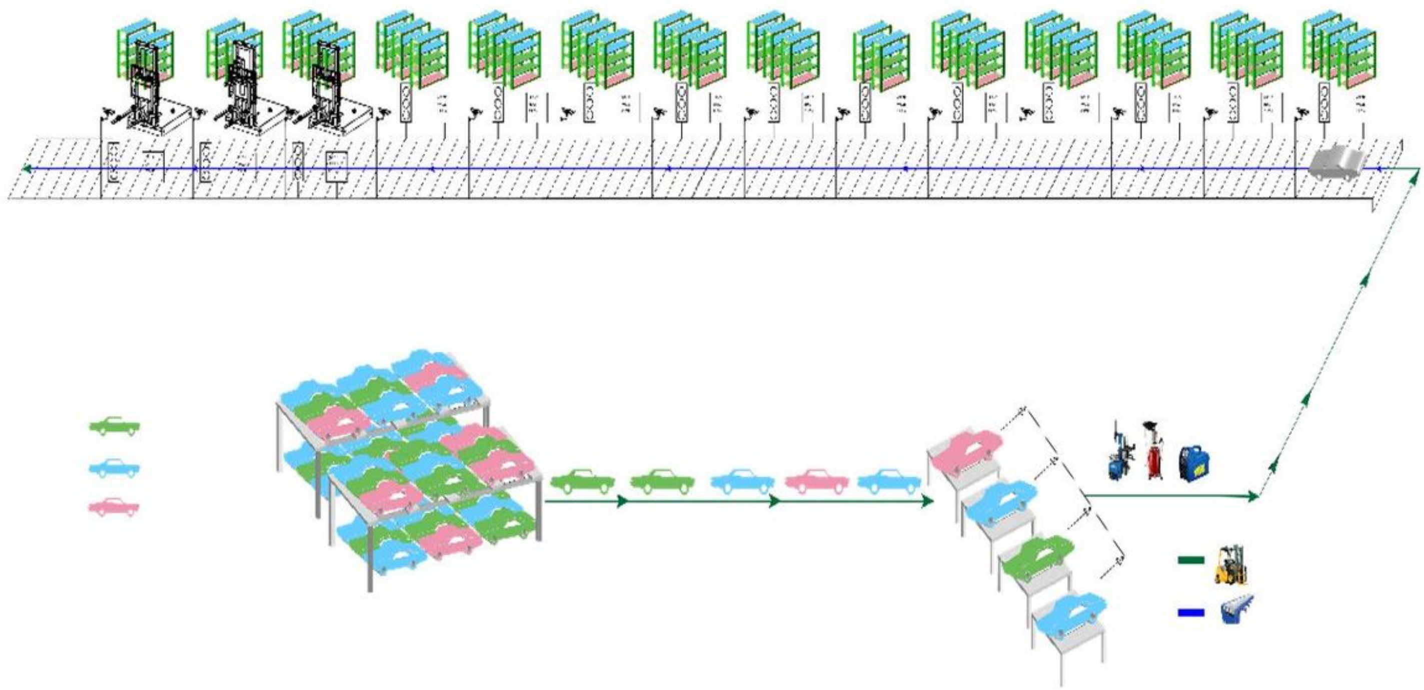

Figure 1

Autonomous vehicle disassembly line

```
1: Determine algorithm parameters
2: Generate the first population
3: Sort the population
4: while (if stop criteria are not met) do
// Elite site
5:    for i=1:e
6:        for j=1:nep
7:            neighborhood search for n(i)
8:            Select n(i)'s best neighbor
9:        end
10:       if best neighbor < n(i)
11:           n(i)=best neighbor
12:       end
13:   end
// Selected Site
14:   for i=e+1:m+e
15:       for j=1:nsp
16:           neighborhood search for n(i)
17:           Select n(i)'s best neighbor
18:       end
19:       if best neighbor < n(i)
20:           n(i)=best neighbor
21:       end
22:   end
// Global search
23:   for i=e+m+1:n
24:       Generate random new members
25:   end
26:   Sort population
27: end while
```

```
1: Determine algorithm parameters
2: Generate the first population
3: Sort the population
4: while (if stop criteria are not met) do
// Elite site
5:    for i=1:e
6:        for j=1:nep
7:            Generate random number s (1-3)
8:            if s==1
9:                Swap(n(i))
10:           elseif s==2
11:               Insertion(n(i))
12:           else
13:               Reversion(n(i))
14:           end
15:       Select n(i)'s best neighbor
16:       end
17:       if best neighbor < n(i)
18:           n(i)=best neighbor
19:       end
20:   end
// Selected Site
21:   for i=e+1:e+m
..................................
..................................
..................................
36:   end
// Global search
33:   for i=e+m+1:n
34:       Generate random new plans
35:   end
36:   Sort population
37: end while
```

Figure 2

Pseudo code of the Bees Algorithm & Process Optimisation

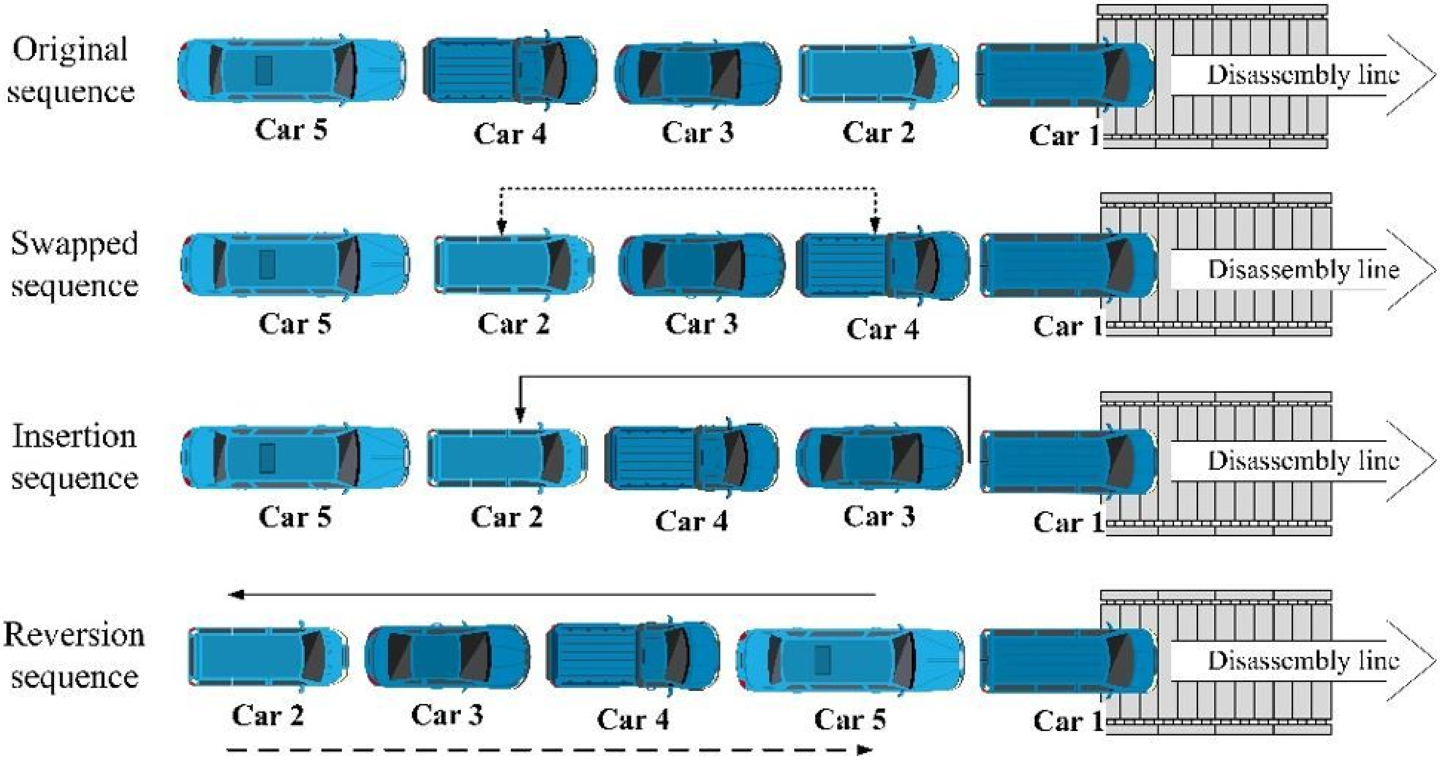

Figure 3

Three different operators in the local search section

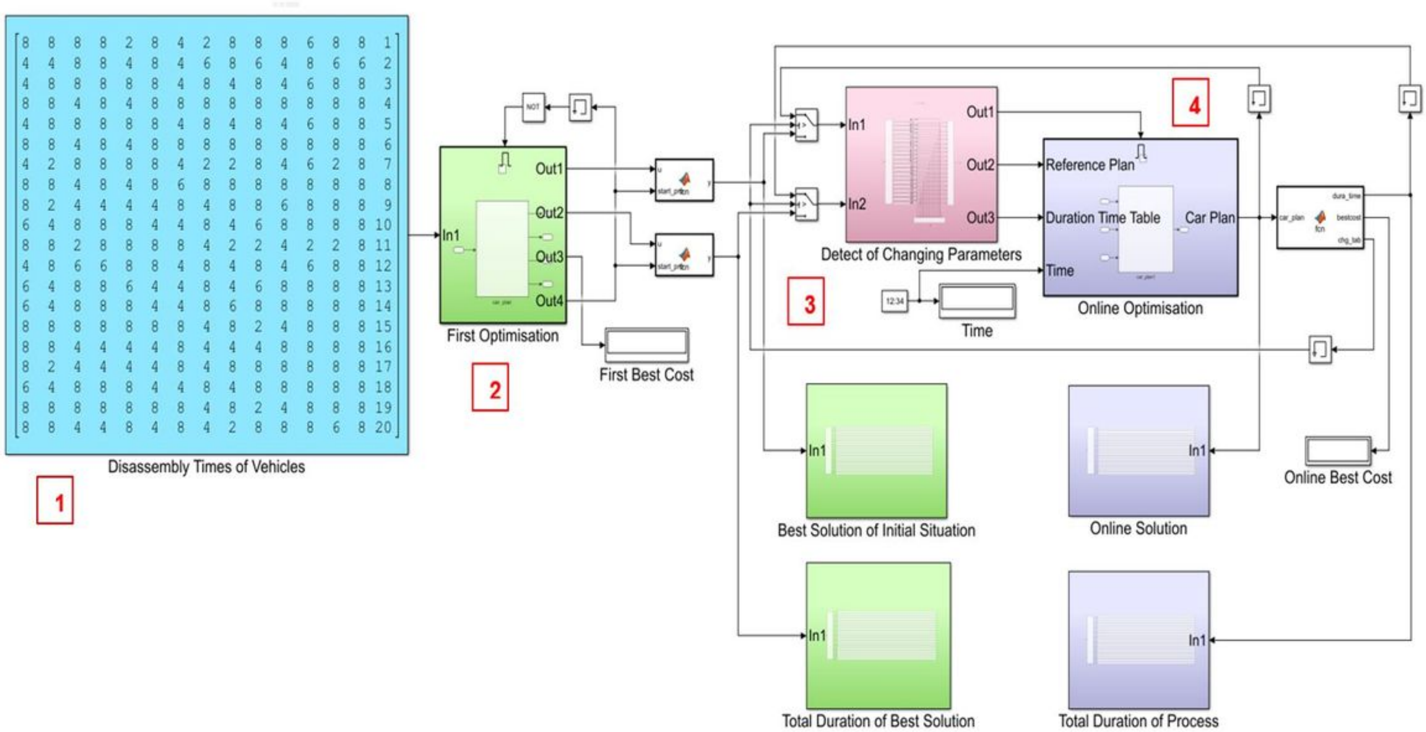

Figure 4

The general view of the model

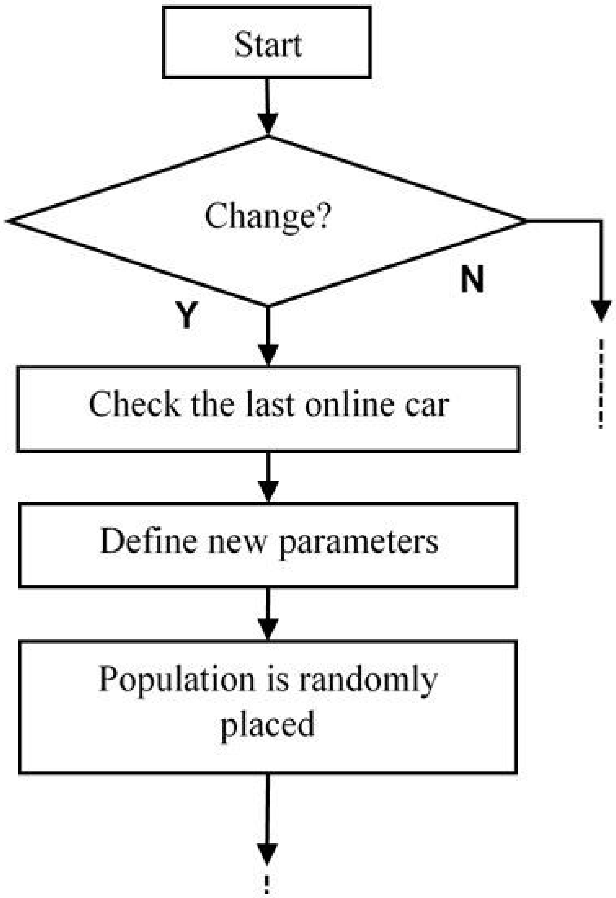

Figure 5

Online optimisation scheme

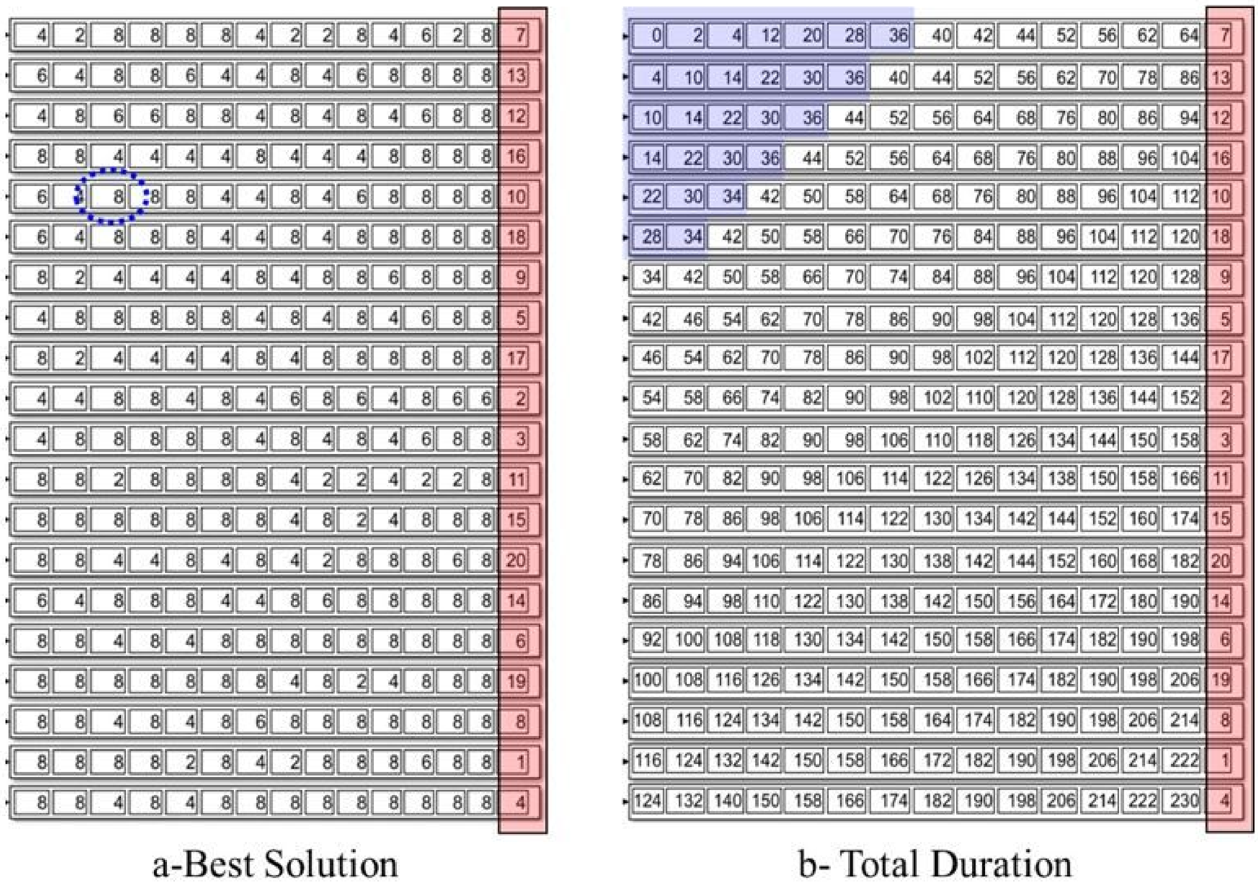

Figure 6

Best Solution of Initial Situation and Duration Time of Best Solution

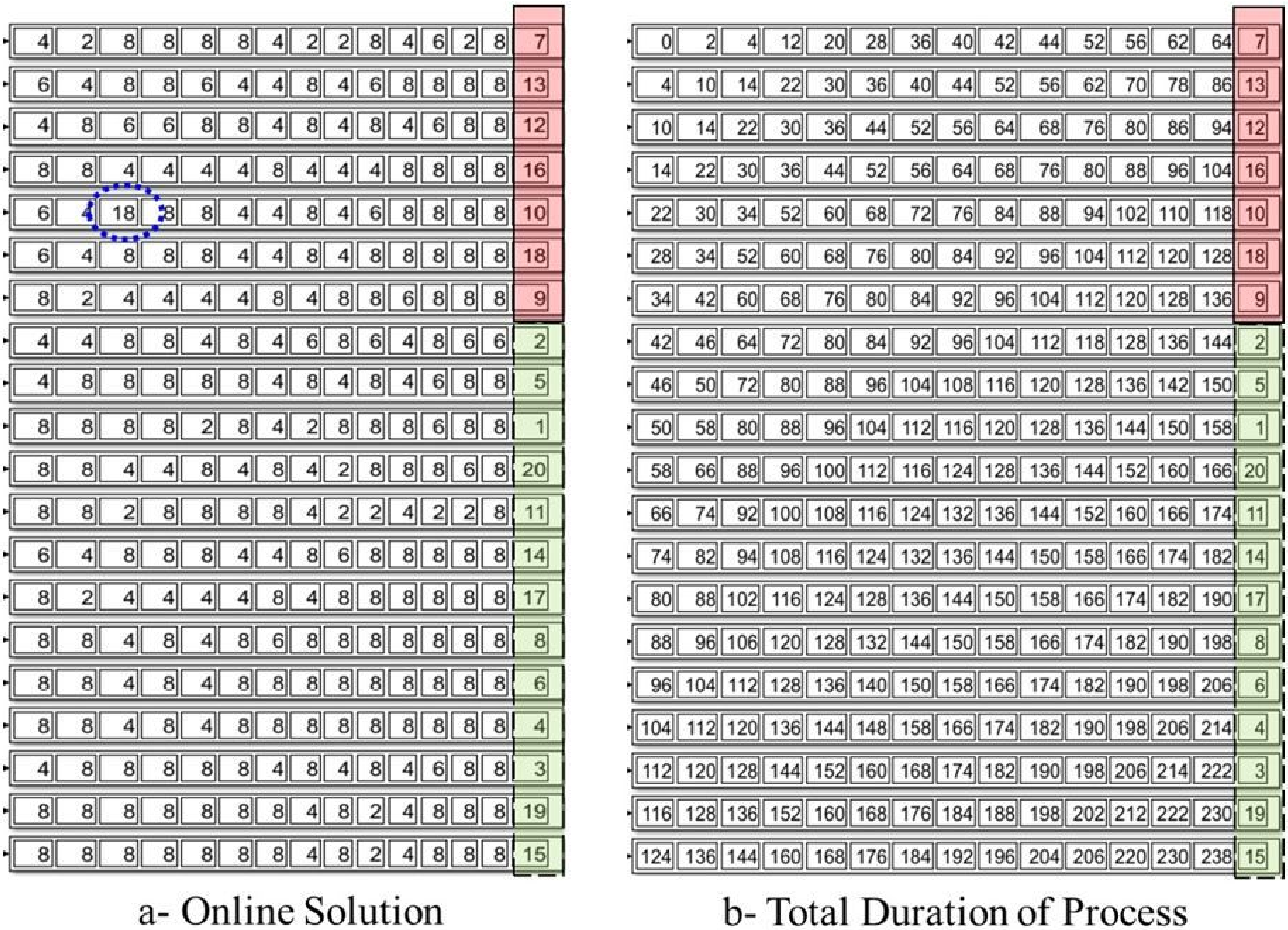

Figure 7

Online Solution and Total Duration of Process

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- onlinebeesPFSP.rar