applied sciences

MDPI

Article

Sales Kit Automated Production: An Integrated Procedure for Setup Reduction in Case of High Products Variety

Matteo Bottin [1], Maurizio Faccio [2,*], Riccardo Minto [2] and Giulio Rosati [1]

[1] Department of Industrial Engineering, University of Padova, 35131 Padova, Italy; matteo.bottin@unipd.it (M.B.); giulio.rosati@unipd.it (G.R.)

[2] Department of Management and Engineering, University of Padova, 36100 Vicenza, Italy; riccardo.minto@unipd.it

* Correspondence: maurizio.faccio@unipd.it

Abstract: Nowadays, the demand for personalized goods is increasing, with small batches of customized products. Companies are asked to cope with this need to be more competitive; thus, it is crucial to optimally set up the production process to cope with the demand. Among the products composed of several parts, sales kits occupy a significant role. Sales kits are sets of different components, supplied as a single unit. Automation is usually used to reduce the unit direct production costs for assembled products, and this is especially true for kitting since it has minimum added value, compared to other technologies. The wide range of components leads to frequent reconfiguration of the feeding devices between different products, increasing the setup time. The aim of this work is to provide a model that minimizes the setup times of automatic kitting systems when working with a high variety of products. To reach this goal, we propose to apply the traveling salesman problem (TSP) to the production process by integrating a multi-clustering model, thus increasing its appeal to several applications. This method is applied both to simulations and to a real case study, and proves to be able to provide a good solution in a reasonable time.

Keywords: sales kit; clustering; kitting lines; traveling salesman problem; feeder

check for updates

Citation: Bottin, M.; Faccio, M.; Minto, R.; Rosati, G. Sales Kit Automated Production: An Integrated Procedure for Setup Reduction in Case of High Products Variety. *Appl. Sci.* **2021**, *11*, 10110. https://doi.org/10.3390/app112110110

Academic Editor: Emanuele Carpanzano

Received: 14 September 2021
Accepted: 26 October 2021
Published: 28 October 2021

1. Introduction

Nowadays, it is fundamental for many manufacturing companies to provide a wide range of products with minimum lead time. To reach this goal, a production system must be flexible, i.e., manage changes in product and volumes [1]. Despite manual assembly systems (MAS) achieving a high degree of flexibility, several drawbacks limit their usage: the accuracy and repeatability is inferior to that achievable with automatic systems, and ergonomic problems affects the achievable throughput [2]. Therefore, automation is a viable option, thanks to their improved throughput and product quality with reduced production costs in comparison to MAS.

Kitting is one of the most critical in terms of throughput and flexibility. Kitting is the process where all the parts required to assemble a product are collected in a kit container, instead of storing the parts along the line-side storage locations, as in line-side stocking [3]. However, this work focuses on kits as the end product of the production line, i.e., the set of the assembled product packed together with its accessories and supplied to the customers as a single product, and which are called sales kits [4]. Different from other assembly technologies, kitting is based on operations with minimum added value to the end product [5], even if from the logistic point of view it creates added value in the form of improved quality and reduced search time [6]; thus, it is critical to increase throughput. The required flexibility often leads to the adoption of manual systems, which creates a range of problems as seen with MAS [7], but also, it is labor intensive. Thus, the adoption of automatic kitting systems is fundamental, but to reach this goal, it is necessary to have flexible systems to minimize the lead times.

For an automatic system to be competitive, it should [8] be able to adapt to part and volume changes, be capable of processing multiple parts and models simultaneously, and carry out model changeovers quickly and easily. To improve the flexibility of automatic systems, flexible assembly systems (FAS) have been developed. Flexibility during the fixing phase is achieved by using a programmable robotic manipulator [9,10]. However, for a FAS to be truly flexible, it is important to ensure that the feeder system provides multiple and different parts quickly. Indeed, the fundamental tasks of partitioning the parts, i.e., the separation and presentation in a certain amount and orientation at a picking location, are carried out by the feeders [11]. This process is critical for the flexibility of the assembly system [12] and should be accordingly studied.

Indeed, traditional automatic feeding systems [11] are mainly passive orientating systems, and therefore, they present limited flexibility, i.e., they can provide a limited number of component types at the same time. However, their low cost is appealing for kitting companies. Hence, the workcell may require as many feeders as the number of parts; changing between products requires the set up of usually more than one feeder, thus increasing the setup time. Therefore, it is important to reduce the number of feeder reconfigurations between changeovers to reduce the overall setup time.

This work aims to provide a model that minimizes the setup times of automatic kitting systems by adopting a suitable schedule that minimizes the number of retoolings of its feeders. Indeed, the productivity of a logistics system depends on the efficiency of the schedule: if different products are to be made on a single line, it is important to reduce the setup time between each changeover [11]. To achieve this goal, this work adopts a traveling salesman problem (TSP) approach, which aims to minimize the reconfiguration times. TSP is already adopted in the literature to minimize the cost of setup. However, when the number of parts is significant, which is reasonable for sales kitting since it is characterized by high product variety, the computational time may be too large to adopt this method in flexible systems. Hence, the novelty of this work is to adopt a multi-clustering approach to decompose the problem into subproblems, which can be managed with the TSP algorithm and provide a solution in low computational time. This faster approach can be used in flexible manufacturing systems in order to quickly respond to an unexpected event and obtain a new schedule faster than with traditional methods.

This work is structured as follows: Section 2 provides a literature review, showing the novelty and the importance of the study; Section 3 presents the mathematical model. Section 4 applies the model to simulations to identify its capabilities in terms of final changeover cost and computational time, and Section 5 applies the proposed model to an industrial case study. Lastly, Section 6 concludes the work.

2. Literature Review

An analysis of the state of the art should be carried out to identify the different approaches adopted in the literature to reduce the setup time, with a major focus on kitting processes. As seen in Table 1, a major part of the literature regarding the reduction of setup times is focused on job shop floor scenarios.

Table 1. Summary of the literature review on setup reduction methods in the last decade.

Author	Year	Industry			Objective					Method					Results
		Job Shop	Assembly	Kitting	Tardiness	Flowtime	Makespan	Setup Tme	Other	Search Algorithm	GA	Heuristic	SMED	Other	
[13]	2018		X		X	X				X					Improved calculation time
[14]	2018	X					X				X				Similar results to exact method
[15]	2017	X					X			X					Tabu search allows to solve more complex scenarios
[16]	2017			X					X					X	Kitting requires floor space and picking time
[17]	2014	X				X				X					Improvement on traditional VNS
[18]	2014		X						X		X				Average assembly time reduction of 2.5%
[19]	2013	X	X				X			X					Different results when machining or assembly are bottleneck.
[20]	2013		X					X					X		Setup time reduction up to 79%
[21]	2013	X							X		X				Genetic algorithm preferred for large scale problems
[22]	2013	X			X									X	Reduction of at least 17% in tardiness
[23]	2012	X						X					X		Changeover from 399 min to 120 min
[24]	2012	X			X							X			Better solutions than traditional ACO
[25]	2012	X					X						X		Setup time reduction from 85 min to 50 min
[1]	2011	X					X		X			X			Reduction in makespan and increase in machine utilization
[26]	2011	X					X				X	X			GA more performing if setup time is 20% of makespan; heuristic reduces more setup times

Indeed, the literature on kitting processes is mainly focused on comparison with other feeding policies [27], such as line-side stocking [6,28], focusing more on product kits [4]. Moreover, the kitting processes studied are mainly manual [29], preferred due to the flexibility of the operator, whereas automatic systems are more focused on providing technical solutions [30–32], providing cost evaluating models [33] or estimating the performance [34]. However, automating the kitting process is fundamental since kitting is labor intensive; hence, it is considered a valid solution to increase productivity [16]. Despite automated storage and retrieval systems having been adopted to improve logistics since 1950s, traditional systems based on stacker cranes [35] are not suitable for kitting processes in small areas; hence, mobile robotic systems are studied in the literature [36]. One of the main problems when adopting mobile robots is optimally performing all the tasks to minimize the schedule length or duration [37] to minimize the setup time of the workcells.

From our review, it is possible to distinguish two main methods. The first approach is based on the reduction in the setup time by acting on the setup operations, i.e., the single minute exchange of die (SMED) methodology. Applied since the 1950s [25], this approach is based on the identification of two types of setup operations [38]:

- Internal setup activities, which can be performed only when the machine is stopped;
- External setup activities, which can be performed even when the machine is operating.

Through this, it is possible to focus only on and to improve the internal activities, as seen in [23]. The authors developed a new clamping system; by applying the SMED approach, it was possible to reduce the changeover time from 399 min to 120 min.

However, when working on a complex system, SMED may not be sufficient to obtain successful solutions [20]. Indeed, manufacturing systems are complex, as they are comprised of a set of different machines connected by automated transfer systems, and their relationship is not included in the SMED approach. As shown by the author, considering also a balancing approach improves the achievable setup time reduction, from 40% to 79%.

The flexible job shop scheduling problem, which consists of scheduling a set of operations on a set of flexible machines, is known to be NP-hard [39]. Hence, to solve large-scale problems, which are more akin to real industrial scenarios, meta-heuristics are usually employed [21]. Genetic algorithms (GA) have been considered. Jalilvand-Nejad and Fattahi [21] proposed a GA based on different strategies for initial population, crossover, and mutation operations to improve its efficiency. Shokohui [14] presented a multi-objective GA, which aims to minimize the makespan, machine load, and total load. Luo et al. [26] proposed a comparison between a GA and a heuristic to solve hybrid flow shop scheduling with a setup time characteristic for each family. Their results showed that the GA, despite being more high performing with a lower makespan, requires more changeover than the heuristic approach, thus increasing the setup time, with an average increase in the setup time of 43.2%.

GA have also been adopted to solve the TSP problem for mobile robots[40]. In [37], such a GA was proposed to reduce the travel path of a mobile robot that performs logistics-related tasks (picking and delivery). The algorithm was compared to a traditional greedy algorithm, i.e., a planner that picks nearest tasks, showing a reduction in the travel path of at least 40%. A GA was also adopted in [41], where it minimizes the energy usage by minimizing the traveling time. The system was also applied to an industrial application, where the robot was successfully capable of continuously picking and placing SLCs from the warehouse. Lastly, [42] compared the performance of a GA-based heuristic with a mixed-integer programming (MIP) model. Indeed, the MIP was capable of finding optimal solutions, but proved to be too time consuming. On the other hand, the approximated GA approach was significantly faster, with differences in the results of only 3%.

Considering our review, the majority of works have focused on the search algorithm, e.g., tabu search [15]. Sabouni et al. [13] proposed a custom search algorithm, CFIM2, which improves the traditional tabu search. Whereas the two algorithms achieve similar performance, tabu search generates a larger number of solutions, thus requiring a higher computational time. Adibi et al. [17] combined a variable neighborhood search with a clustering approach. The authors introduced a clustering analysis to improve the efficiency of the algorithm by grouping a set of jobs on the basis of the processing time. This approach leads to better solutions, increasing the improvement percentage as the mean time job arrivals increases, thereby improving the schedule as the problem complexity increases.

As a result of our literature review, we can identify that previous studies agree that real industrial scheduling problems are complex, and meta-heuristics or approaches that decompose the problem into subproblems should be considered. Despite previous works having presented algorithms to reduce the setup times, little focus has been directed toward the reduction in setup time in sales kitting. Moreover, the literature regarding automatic kitting processes focuses, to the authors' knowledge, on technical solutions and performance evaluation models. Approaches to minimize kitting process times have been developed regarding mobile robots, where heuristics have been adopted to solve the TSP. However, these works do not focus on the minimization of the retooling of feeders, which is another important problem in automatic kitting processes. Therefore, this work aims to reduce the setup time in the sales kitting industry, adopting a multi-clustering approach to reduce the complexity of the problem due to the high number of products and components characteristic of this process.

3. Mathematical Model

3.1. Model Description

Let us consider an enterprise in the logistics field. Such an enterprise is able to provide to the customers multiple products, and each product is made of several parts. Different products are built with different parts, but different products may share some parts; clearly, different products must not be built with the same parts; otherwise, they would be considered coincident. Indeed, since this work focuses on the feeding process, the assembly phase is not considered, which may result in different final products. This approach is perfectly applicable to a kitting line, in which the parts are grouped together to form kits (i.e., products).

Products and parts are summarized in the bill of materials (BOM) matrix. The BOM is a logical $N_p \times N_c$ matrix (M), where N_p is the number of products and N_c is the number of parts. The M elements m_{ik} are as follows:

$$M = [m_{ik}] \quad , \quad m_{ik} = \begin{cases} 1 & \text{if product } i \text{ is made with part } k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

As stated before, for each part to be fed, a feeder is required. Thus, with a great number of parts, it is important to reduce the setup time between each changeover, i.e., between two kits. This setup time depends on the number of feeders to be reconfigured; thus, it is important to evaluate its extent.

The number of reconfigurations can be calculated directly by comparing the list of parts to be used for two specific products (Table 2): if a product i uses a certain part k and another product j does not, then a reconfiguration is required. Such reconfiguration could be avoided if a very high number of feeders is installed in the workcell, but such a solution is impractical if N_c is high, due to the cost and encumbrance.

Table 2. Relationship between two different products in terms of feeder changes for a generic part k. Feeder resetting is to be done if only one of the products uses the specific part (and the other does not).

Part k		Product j, m_{jk}	
		1	0
Product i, m_{ik}	1	0	1
	0	1	0

The aforementioned comparison is equal to a XOR operation of two different rows of matrix M. The resulting vector D_{ij} shows the parts for which the feeder reconfiguration is required. The total number of reconfigurations between two products i and j, S_{ij} is as follows:

$$S_{ij} = \sum_{k=0}^{N_c} D_{ij,k} \tag{2}$$

where $D_{ij,k}$ is the k-th element of vector D_{ij}. It is clear that $S_{ij} \in [1, N_c]$ since there is at least one part that differs between the two products (lower bound).

If the reconfiguration time of one feeder for a specific part is $t_{set,k}$, the overall cell setup time between two products i and j is as follows:

$$t_{set,ij} = \sum_{k=0}^{N_c} D_{ij,k} t_{set,k} \tag{3}$$

In Equation (3), the setup time $t_{set,k}$ can be modified, including similarities between the parts, i.e., the fact that in some cases, it is possible to reconfigure a feeder by changing only a subset of traps, reducing the overall setup time [43]. For the sake of brevity, $t_{set,k}$ is considered a mean time among all the possible setup times for the specific part.

From Equation (3), it is possible to define a $N_p \times N_p$ cost matrix C. Such a matrix contains all the reconfiguration times:

$$C = [c_{ij}] \quad , \quad c_{ij} = t_{set,ij} \tag{4}$$

To optimize the logistics system, C can be used: several algorithms are suitable (such as genetic algorithms [14,21] or tabu search algorithms [15]), but in our case, we choose to use the traveling salesman problem (TSP) [44] to solve the problem, due to its ease of implementation and its non-stochastic properties.

TSP is a problem which aims at finding the best path that connects a certain number of "cities" with the minimum cost, starting from one city, visiting all the other cities just once and coming back to the first one. This results in a closed loop in which the starting point can be chosen arbitrarily. In our scenario, the cities are the products, and the path cost is made up of the feeder reconfiguration. Reducing the cost of the entire path means reducing the overall reconfiguration time. Our work aims at reducing the complexity of the TSP in computational terms, dividing the products into clusters, and thus, imposing additional constraints to the problem that allow the finding of a solution to the TSP in a lower computational time. Clusters can be further divided into clusters; thus, it is possible to apply a multi-clustering approach, increasing the flexibility of our approach.

3.2. The Traveling Salesman Problem

Let us consider x_{ij} to be the connection between products i and j; x_{ij} is an integer, and its value is defined by the status of the path:

$$x_{ij} = \begin{cases} 1 & \text{if } loc_i \text{ and } loc_j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

In a kitting process, x_{ij} describes the actual reconfiguration process between two kits. Its cost is described by c_{ij} of Equation (4).

The TSP finds $X = [x_{ij}]$ that solves the function as follows:

$$\min \left(\sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij} \right) \tag{6}$$

While TSP can be used with the full C matrix, the computational time may greatly increase if a large number of products is to be considered [45]. Therefore, it is important to reduce the computational time to obtain a good solution in a reasonable time.

To do so, clustering is of great importance: not only is the algorithm that solves the TSP constrained to evaluate a smaller number of solutions, but the intrinsic characteristic of TSP makes it well suited for the clustering of cities [44]. In fact, clustering the cities in the TSP means that the solution "enters" the cluster, moves between all the cities of the cluster and then moves to another cluster. This means that for each cluster S, containing n_s positions, the TSP defines a non-closed sub-tour containing only the positions of the cluster. This constraint can be defined as follows [44]:

$$\sum_{i \neq j, i \in S, j \in S} x_{ij} = n_s - 1 \tag{7}$$

since the last connection that would close the tour has to be removed to connect the cluster to the other positions. The number of connections between the clusters must be equal to the number of clusters to obtain a closed path. The clustering can be performed multiple times: Equation (7) can be used to group different clusters or group different products.

By applying constraints to the TSP, the final solution can result in a cost which differs from the optimal solution, but can be obtained in lower computational times.

3.3. Clustering

Grouping the products into clusters is a problem that has been studied in the past years [46]. Sometimes, some parts are widely used among products, while in other cases, very few parts require a high setup time $t_{set,k}$. Combinations of the two can also be found.

Similar parts can be grouped into "families". As an example, screws with different thread lengths but the same diameter can be grouped within the same family. In mathematical terms, the membership of a part k to a family F is defined via the integer f_k:

$$f_k = \begin{cases} 1 & \text{if part } k \text{ is in family } F \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

As a result, it is possible to evaluate the mean setup time of each family t_F as follows:

$$t_F = \frac{\sum_{k=1}^{N_c} t_{set,k} f_k}{\sum_{k=1}^{N_c} f_k} \tag{9}$$

Different families are described by different mean setup times (t_F). As a result, some families have a greater impact on the overall setup time with respect to others. Due to the Pareto principle [47], "roughly 80% of consequences come from 20% of the causes", which translates to the fact that only a few families are responsible for the majority of the setup time.

By following the Pareto principle, it is possible to cluster the parts into the respective families. The steps of the clustering process can be summarized as follows (Figure 1):

1. Parts are divided into families (i.e., the parts that are used by a specific machine can be grouped into a family).
2. The families are listed in descending order based on t_F (i.e., at the top of the list are placed the families with higher t_F).
3. The cumulative sum of t_F is performed.
4. Once a certain threshold is reached (e.g., 80% for the Pareto principle), the list of the summed families is retrieved. These families are used for the multi-clustering process.
5. Starting from the first family, the products are divided based on the different parts of the families (thanks to the value of m_{ik}, Equation (1)) to form clusters. Starting from

the second family, products with the same part may be placed in different clusters based the previous clustering (Figure 2).

From Point 5 it is possible to apply TSP to calculate the optimal sequence of products. To do so, the appropriate constraints have to be applied (Equation (7)).

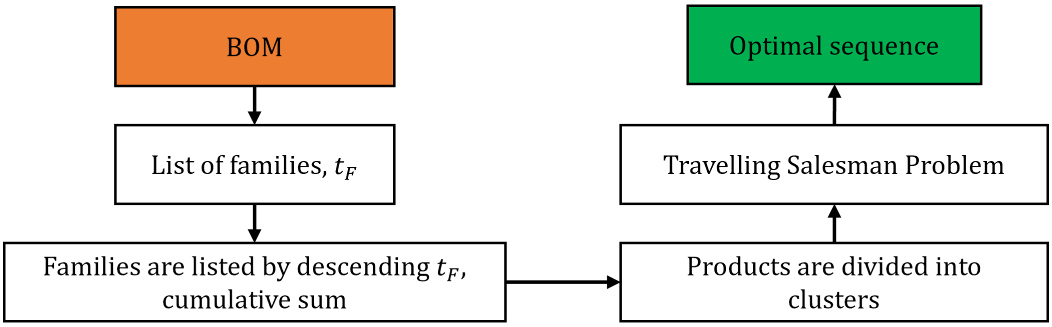

BOM
List of families, t_F
Families are listed by descending t_F, cumulative sum
Products are divided into clusters
Travelling Salesman Problem
Optimal sequence

Figure 1. Optimization process. Starting from the matrix *BOM* as an input, it is possible to obtain the optimal sequence to be applied to the system.

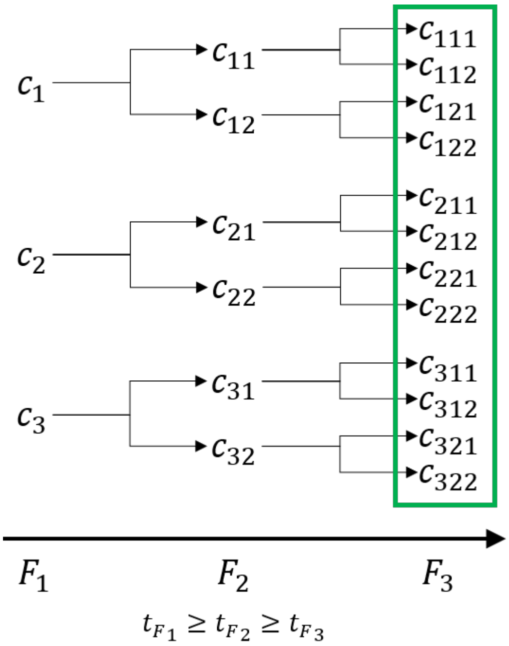


Figure 2. An example of the multi-clustering process: starting from the first family F_1 (with 3 parts), multiple clusters are created by dividing the first clusters based on the families F_2 and F_3 (each with 2 parts). Clusters in the green box are used for the TSP.

4. Simulations

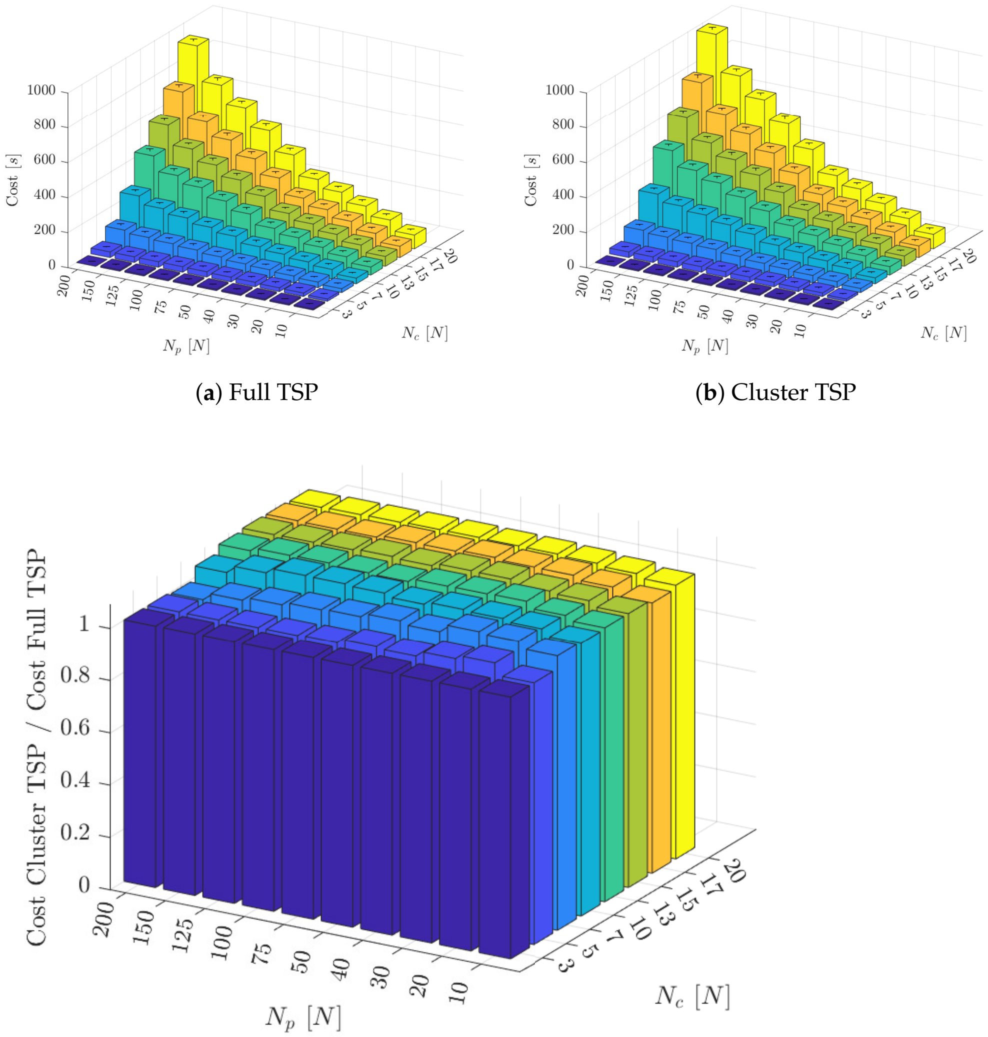
This section aims at comparing the performance of the standard TSP to the TSP with clusters (which, from now on, we refer to as “cluster TSP”). To achieve this result, the mathematical model described in Section 3 is simulated in Matlab by varying the ruling parameters:

- The number of parts N_c.
- The number of products N_p.

The BOM matrix M is randomly generated, and for each part, $t_{set,k}$ is randomly assigned. As a result, the total number of reconfigurations between two products (S_{ij}) is calculated by means of Equation (2). For each set of N_c and N_p, the simulations are run 20 times for statistical relevance.

The results are shown in Figures 3 and 4. As expected, in general, the overall cost provided by the solution of the cluster TSP (Figure 3b) is higher than the one of the full TSP (Figure 3a). This result is perfectly reasonable since the full TSP aims at finding the best solution to the problem, while the cluster TSP applies some constraints on the connections that force the final solution to diverge from the optimal one. In this sense, the cluster TSP provides a sub-optimal solution which satisfies the constraints provided during the definition of the problem.

However, the final cost of the cluster TSP is generally very close to the cost of the full TSP (Figure 3c). In this sense, an enterprise could apply some constraints to the problem (maybe not directly related to the setup time) without the worry of drastically reducing the performance of the system.

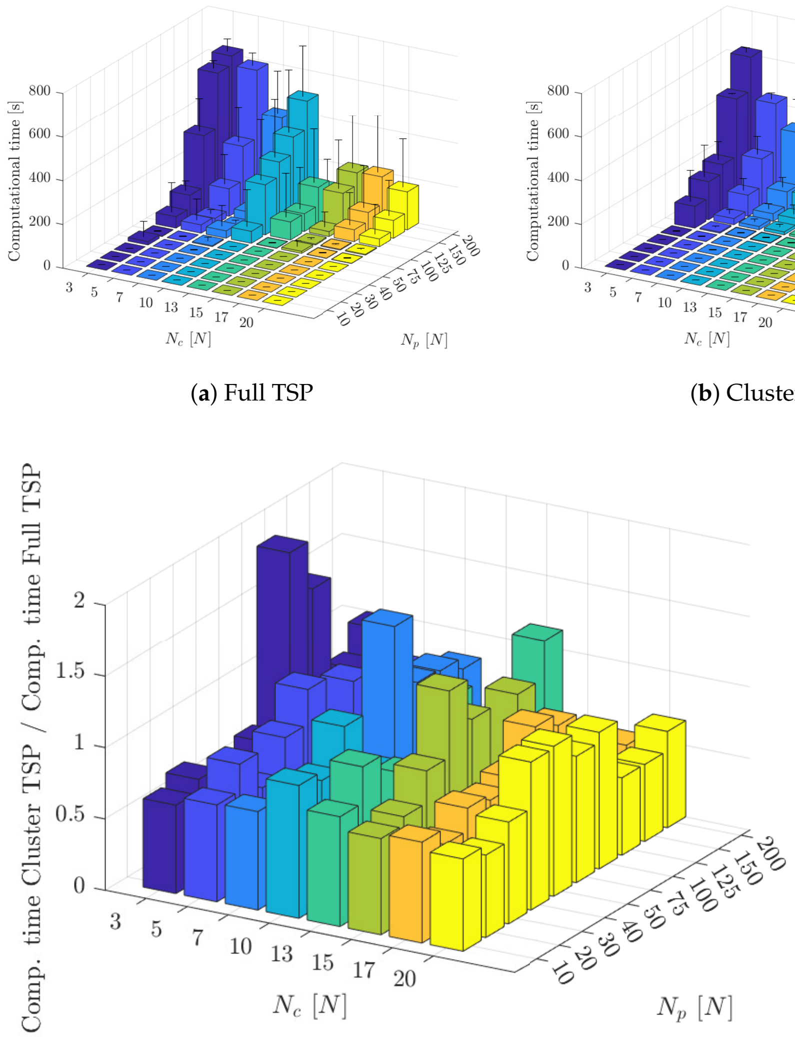


(**a**) Full TSP

(**b**) Cluster TSP

(**c**) Normalized values

Figure 3. Simulation results: overall cost. (**a**) The cost of the solution of the Full TSP; (**b**) The cost of the solution of the Cluster TSP; (**c**) The cost of the solution of Cluster TSP divided per the corresponding cost of the solution of Full TSP.



(**a**) Full TSP

(**b**) Cluster TSP

(**c**) Normalized values

Figure 4. Simulation results: computational time. (**a**) The computational time of the Full TSP; (**b**) The computational time of the Cluster TSP; (**c**) The computational time of Cluster TSP divided per the corresponding computational time of Full TSP.

On the other hand, the computational time (Figure 4) is where generally the cluster TSP outperforms the full TSP. In fact, the performance of the solver depends on the implementation itself, but we have found that for most of the tests, the cluster TSP requires less computational time than that of the full TSP. This result is also highly influenced by the punctual load of the PC, and it can be found sometimes (Figure 3c) that the full TSP is faster than the cluster TSP. Furthermore, this influence is even more relevant with lower N_c and N_p, where the computational time of the two TSP is generally lower than a tenth of a second. In this scenario, the TSP implementation is of no use, and the SMED approach should be used.

By analyzing Figure 3c, it is worth noting that the cluster TSP is more appealing if the number of products to be handled is high. In this way, it is possible to obtain a sub-optimal solution in a reasonable time.

Figure 3a,b shows how, in general, with equal N_p, the computational time decreases with N_c. This is due to the number of constraints applied to the problem: if the number of parts is low, the clusters usually contain a higher number of products, thus applying more constraints. Moreover, with a high number of products and a low number of parts, it is more likely to have very similar S_{ij} (Equation (2)), so it is more difficult for the algorithm to find an optimal solution. This issue can be solved by improving the algorithm that solves the TSP [48].

It is worth noting that the TSP itself is crucial in reducing the overall cost. Figure 5 shows the maximum, median and minimum cost of all the possible combinations of products, for various values of N_p. For simplicity, the number of components is kept constant ($N_c = 10$), and the simulations are run 20 times for statistical relevance. The results show how the TSP (blue line) greatly reduces the overall cost, even considering the median value (black line).

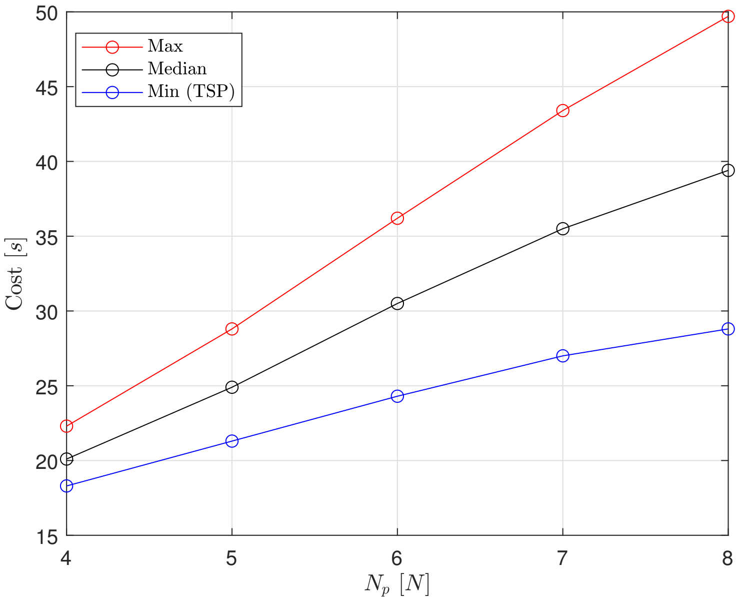


Figure 5. Maximum, median and minimum cost of all the possible combinations of products N_p ($N_c = 10$), simulated 20 times. The TSP solutions lie on the blue curve.

5. Case Study

To assess the validity of our method, it is applied to a real case application. This application is defined by a kitting machine in which small parts are placed within a plastic film to be grouped and a label is applied to the film.

The small parts (e.g., screws) are serialized via bowl feeders and hoppers, while the film and the label are installed directly in the kitting machine. The small parts fall through the hoppers inside the film to form a bag, and then the label is applied to the film. Different products require different numbers of feeder retooling. The higher this number, the higher the number of retooling, and thus, the higher the setup time.

In this sense, the film and the label changes are the leading parts in terms of setup time; thus, it is important to reduce the number of film and label changeovers.

As a result, our multi-clustering approach is applied to both the film and the label. Firstly, the products are clustered based on the film (F_1 of Figure 2). Then, the resulting clusters are clustered based on different labels (F_2 of Figure 2). The total number of possible clusters is 24. The matrix C contains the number of feeder retooling between two kits, considering all the retooling times to be equal.

The company that owns the kitting machine is able to handle hundreds of different products on the same device. Since not every product has to be produced in a certain timespan, the company is interested in optimizing the retooling of the machine with a small number of products. For example, the company may be interested in managing a week of production in which 100 different products have to be produced, based on costumers needs. As a result, our method is applied to different N_p values.

The results are depicted in Figures 6 and 7. The overall cost provided by the cluster TSP (blue line of Figure 7) is always higher than the cost of the standard TSP (red line), which is an expected result: the clustering applies some constraints that drift the final solution away from the optimal one provided by the standard TSP. However, the clustering, in the same way as shown previously in Section 4, is generally more computationally efficient. Figure 6 shows how the standard TSP generally requires higher computational time, due to the increased difficulty in reaching the optimal solution via the optimizer. Moreover, in some cases, the standard TSP even fails at providing a solution within the constrained maximum computational time.

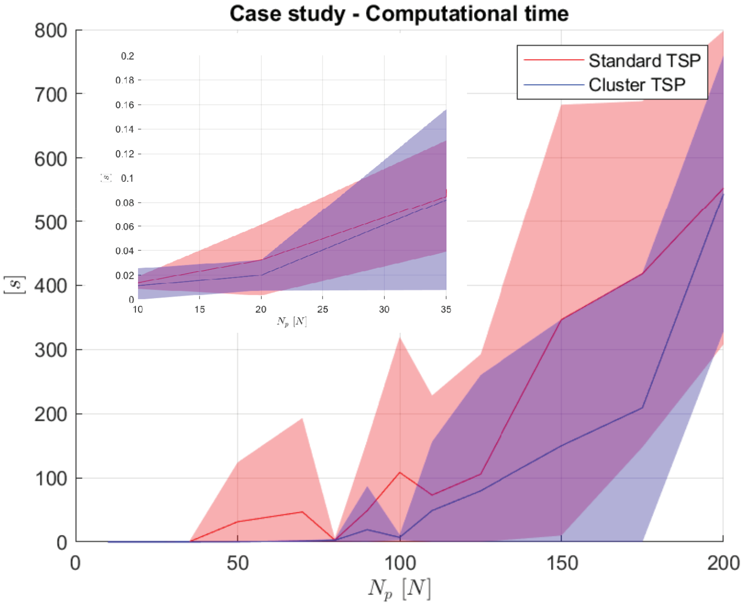


Figure 6. Computational time of the case study with different N_p. On the upper left corner, a zoom of the lowest N_p values.

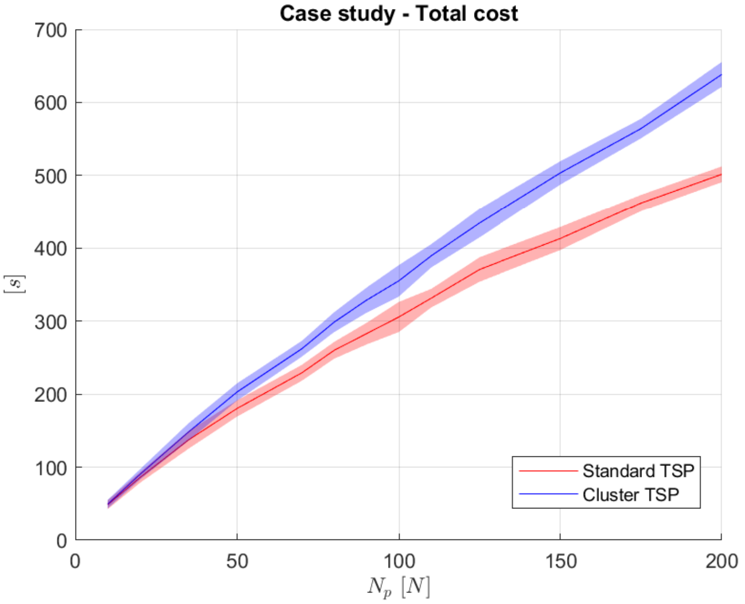


Figure 7. Overall cost of the case study with different N_p, where the cost identifies the total number of retooling to process all the products.

6. Conclusions

This paper presents a multi-clustering approach that is implemented on the traveling salesman problem to reduce computational time and thus, improve feasibility. The approach is applied both to simulations and to a case study. Results show that while the TSP with clusters provides a solution with a cost that is higher than the one of the standard TSP (as expected, due to the cluster constraints), the TSP with clusters is generally more computationally efficient than the standard TSP, especially when the number of products and parts to be held increases.

The proposed method can be applied to many industrial processes with very little computational effort: in fact, it could be implemented in any language without any special requirements. The optimization process is simplified since after its general implementation it requires only the bill of material matrix to find the optimal sequence. Under and industrial perspective, the proposed method could be of great use because it does not require skilled operators.

Future work will address the possibility of using off-line retooling for specific parts (e.g., when the number of feeders is higher than the number of parts that are fed to assemble the current product) and its impact on the final reconfiguration cost.

Author Contributions: Conceptualization, M.F. and G.R.; methodology, G.R.; software, M.B. and R.M.; validation, M.B. and G.R.; formal analysis, M.B. and M.F.; investigation, R.M.; resources, M.F.; data curation, M.B. and R.M.; writing—original draft preparation, M.B. and R.M.; writing—review and editing, M.F. and G.R.; visualization, R.M.; supervision, M.F. and G.R.; project administration, M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

N_p	Number of products
N_c	Number of components
i, j, k	Indices that define a specific element of a matrix
M	Bill of materials (BOM) matrix with elements m_{ik}
S_{ij}	Number of reconfigurations between products i, j
$t_{set,ij}$	Setup time between products i, j
$t_{set,k}$	Reconfiguration time of part k
C	Cost matrix with elements c_{ij}

References

1. Azzi, A.; Faccio, M.; Persona, A.; Sgarbossa, F. Lot splitting scheduling procedure for makespan reduction and machine capacity increase in a hybrid flow shop with batch production. *Int. J. Adv. Manuf. Technol.* **2012**, *59*, 775–786. [CrossRef]
2. Battini, D.; Faccio, M.; Persona, A.; Sgarbossa, F. New methodological framework to improve productivity and ergonomics in assembly system design. *Int. J. Ind. Ergon.* **2011**, *41*, 30–42. [CrossRef]
3. Hu, Z.; Ramaraj, G.; Hu, G. Production planning with a two-stage stochastic programming model in a kitting facility under demand and yield uncertainties. *Int. J. Manag. Sci. Eng. Manag.* **2020**, *15*, 237–246. [CrossRef]
4. Faccio, M.; Gamberi, M.; Pilati, F.; Bortolini, M. Packaging strategy definition for sales kits within an assembly system. *Int. J. Prod. Res.* **2015**, *53*, 3288–3305. [CrossRef]
5. Bozer, Y.A.; McGinnis, L.F. Kitting versus line stocking: A conceptual framework and a descriptive model. *Int. J. Prod. Econ.* **1992**, *28*, 1–19. [CrossRef]
6. Hua, S.Y.; Johnson, D.J. Research issues on factors influencing the choice of kitting versus line stocking. *Int. J. Prod. Res.* **2010**, *48*, 779–800. [CrossRef]
7. Khajavi, S.H.; Baumers, M.; Holmström, J.; Özcan, E.; Atkin, J.; Jackson, W.; Li, W. To kit or not to kit: Analysing the value of model-based kitting for additive manufacturing. *Comput. Ind.* **2018**, *98*, 100–117. [CrossRef]
8. Wolfson, W.; Gordon, S.J. Designing a parts feeding system for maximum flexibility. *Assem. Autom.* **1997**, *17*, 116–121. [CrossRef]

9. Manzini, R.; Gamberi, M.; Regattieri, A.; Persona, A. Framework for designing a flexible cellular assembly system. *Int. J. Prod. Res.* **2004**, *42*, 3505–3528. [CrossRef]
10. Battini, D.; Faccio, M.; Ferrari, E.; Persona, A.; Sgarbossa, F. Design configuration for a mixed-model assembly system in case of low product demand. *Int. J. Adv. Manuf. Technol.* **2007**, *34*, 188–200. [CrossRef]
11. Rosati, G.; Faccio, M.; Carli, A.; Rossi, A. Fully flexible assembly systems (F-FAS): A new concept in flexible automation. *Assem. Autom.* **2013**, *33*, 8–21. [CrossRef]
12. Tay, M.; Chua, P.S.; Sim, S.; Gao, Y. Development of a flexible and programmable parts feeding system. *Int. J. Prod. Econ.* **2005**, *98*, 227–237. [CrossRef]
13. Sabouni, M.Y.; Logendran, R. Lower bound development in a flow shop electronic assembly problem with carryover sequence-dependent setup time. *Comput. Ind. Eng.* **2018**, *122*, 149–160. [CrossRef]
14. Shokouhi, E. Integrated multi-objective process planning and flexible job shop scheduling considering precedence constraints. *Prod. Manuf. Res.* **2018**, *6*, 61–89. [CrossRef]
15. Shen, L.; Dauzère-Pérès, S.; Neufeld, J.S. Solving the flexible job shop scheduling problem with sequence-dependent setup times. *Eur. J. Oper. Res.* **2018**, *265*, 503–516. [CrossRef]
16. Boudella, M.E.A.; Sahin, E.; Dallery, Y. Kitting optimisation in Just-in-Time mixed-model assembly lines: Assigning parts to pickers in a hybrid robot–operator kitting system. *Int. J. Prod. Res.* **2018**, *56*, 5475–5494. [CrossRef]
17. Adibi, M.A.; Shahrabi, J. A clustering-based modified variable neighborhood search algorithm for a dynamic job shop scheduling problem. *Int. J. Adv. Manuf. Technol.* **2014**, *70*, 1955–1961. [CrossRef]
18. García-Nájera, A.; Brizuela, C.A.; Martínez-Pérez, I.M. An efficient genetic algorithm for setup time minimization in PCB assembly. *Int. J. Adv. Manuf. Technol.* **2015**, *77*, 973–989. [CrossRef]
19. Fattahi, P.; Hosseini, S.M.H.; Jolai, F.; Tavakkoli-Moghaddam, R. A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. *Appl. Math. Model.* **2014**, *38*, 119–134. [CrossRef]
20. Faccio, M. Setup time reduction: SMED-balancing integrated model for manufacturing sys-tems with automated transfer. *Int. J. Eng. Technol.* **2013**, *5*, 4075–4084.
21. Jalilvand-Nejad, A.; Fattahi, P. A mathematical model and genetic algorithm to cyclic flexible job shop scheduling problem. *J. Intell. Manuf.* **2015**, *26*, 1085–1098. [CrossRef]
22. Calleja, G.; Pastor, R. A dispatching algorithm for flexible job-shop scheduling with transfer batches: An industrial application. *Prod. Plan. Control.* **2014**, *25*, 93–109. [CrossRef]
23. Abraham, A.; Ganapathi, K.; Motwani, K. Setup time reduction through SMED technique in a stamping production line. *Sastech J.* **2012**, *11*, 47–52.
24. Huang, R.H.; Yang, C.L.; Cheng, W.C. Flexible job shop scheduling with due window—A two-pheromone ant colony approach. *Int. J. Prod. Econ.* **2013**, *141*, 685–697. [CrossRef]
25. Pellegrini, S.; Shetty, D.; Manzione, L. Study and Implementation of Single Minute Exchange of Die (SMED) Methodology in a Setup Reduction Kaizen. Ph.D. Thesis, University of Hartford, West Hartford, CT, USA, 2012.
26. Luo, H.; Huang, G.Q.; Shi, Y.; Qu, T.; Zhang, Y.F. Hybrid flowshop scheduling with family setup time and inconsistent family formation. *Int. J. Prod. Res.* **2012**, *50*, 1457–1475. [CrossRef]
27. Bortolini, M.; Faccio, M.; Gamberi, M.; Pilati, F. Assembly kits with variable part physical attributes: Warehouse layout design and assignment procedure. *Assem. Autom.* **2020**, *40*, 867–868. [CrossRef]
28. Usta, S.K.; Oksuz, M.K.; Durmusoglu, M.B. Design methodology for a hybrid part feeding system in lean-based assembly lines. *Assem. Autom.* **2017**, *37*, 84–102. [CrossRef]
29. Davarzani, H.; Norrman, A. Toward a relevant agenda for warehousing research: Literature review and practitioners' input. *Logist. Res.* **2015**, *8*, 1–18. [CrossRef]
30. Comand, N.; Bottin, M.; Rosati, G.; Agrawal, S.K. A dynamic model for the optimization of rotatory feeding devices. *Mech. Mach. Theory* **2021**, *166*, 104479. [CrossRef]
31. Domae, Y.; Noda, A.; Nagatani, T.; Wan, W. Robotic General Parts Feeder: Bin-picking, Regrasping, and Kitting. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 5004–5010.
32. Kootbally, Z.; Schlenoff, C.; Antonishek, B.; Proctor, F.; Kramer, T.; Harrison, W.; Downs, A.; Gupta, S. Enabling robot agility in manufacturing kitting applications. *Integr. Comput.-Aided Eng.* **2018**, *25*, 193–212. [CrossRef]
33. Caputo, A.C.; Pelagagge, P.M.; Salini, P. A model for planning and economic comparison of manual and automated kitting systems. *Int. J. Prod. Res.* **2020**, *59*, 1–24. [CrossRef]
34. Boudella, M.E.A.; Sahin, E.; Dallery, Y. A mathematical model to assess the performance of a robotic kitting system in an assembly plant. In Proceedings of the 11th International Conference on Modeling, Optimization and Simulation-MOSIM'16, Montréal, QC, Canada, 22–24 August 2016.
35. D'Antonio, G.; De Maddis, M.; Bedolla, J.S.; Chiabert, P.; Lombardi, F. Analytical models for the evaluation of deep-lane autonomous vehicle storage and retrieval system performance. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 1811–1824. [CrossRef]
36. Wakabayashi, Y.; Kamioka, M.; Kinugawa, J.; Kosuge, K. KitPaDy: Robot co-worker in kitting process: Basic concept and computation of working time. In Proceedings of the 2017 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, China, 20–23 August 2017; pp. 320–327.

37. Harun, S.; Ibrahim, M.F. A genetic algorithm based task scheduling system for logistics service robots. *Bull. Electr. Eng. Inform.* **2019**, *8*, 206–213. [CrossRef]
38. Shingo, S.; Dillon, A.P. *A Study of the Toyota Production System: From an Industrial Engineering Viewpoint*; CRC Press: Boca Raton, FL, USA, 1989.
39. Garey, M.R.; Johnson, D.S.; Sethi, R. The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [CrossRef]
40. Kousi, N.; Koukas, S.; Michalos, G.; Makris, S. Scheduling of smart intra–factory material supply operations using mobile robots. *Int. J. Prod. Res.* **2019**, *57*, 801–814. [CrossRef]
41. Dang, Q.V.; Nielsen, I.; Bøgh, S.; Bocewicz, G. Modelling and scheduling autonomous mobile robot for a real-world industrial application. *IFAC Proc. Vol.* **2013**, *46*, 2098–2103. [CrossRef]
42. Dang, Q.V.; Nielsen, I.; Steger-Jensen, K.; Madsen, O. Scheduling a single mobile robot for part-feeding tasks of production lines. *J. Intell. Manuf.* **2014**, *25*, 1271–1287. [CrossRef]
43. Alkaya, A.F.; Duman, E. Combining and solving sequence dependent traveling salesman and quadratic assignment problems in PCB assembly. *Discret. Appl. Math.* **2015**, *192*, 2–16. [CrossRef]
44. Bottin, M.; Rosati, G.; Boschetti, G. Working Cycle Sequence Optimization for Industrial Robots. *Mech. Mach. Sci.* **2021**, *91*, 228–236. [CrossRef]
45. Bellmore, M.; Nemhauser, G.L. The traveling salesman problem: A survey. *Oper. Res.* **1968**, *16*, 538–558. [CrossRef]
46. Anzanello, M.J.; Fogliatto, F.S. Selecting the best clustering variables for grouping mass-customized products involving workers' learning. *Int. J. Prod. Econ.* **2011**, *130*, 268–276. [CrossRef]
47. Craft, R.C.; Leake, C. The Pareto principle in organizational decision making. *Manag. Decis.* **2002**, *40*, 729–733. [CrossRef]
48. Hoffman, K.L.; Padberg, M.; Rinaldi, G. Traveling salesman problem. In *Encyclopedia of Operations Research and Management Science*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 1573–1578.