Resorting Vehicles in an Automotive Manufacturing Environment

Jens Feller[1], Bernhard Mauersberg[1], Wolfgang Mergenthaler[1], Yvonne Feller[2] and Lutz Müller[3]
[1]*FCE Frankfurt Consulting Engineers GmbH, Altmünsterstraße 2d, D-65439 Flörsheim, Germany*
[2]*Heinz-Nixdorf-Berufskolleg, Dahnstraße 50, 45144 Essen, Germany*
[3]*Porsche Leipzig GmbH, Porschestraße 1,D-04158 Leipzig, Germany*

Keywords: Combinatorial Optimization, Dynamic Optimization, Sequencing, Sorting, Set Theory.

Abstract: One of the most important concepts in an automotive production process is that of a manufacturing sequence. Sequencing has a vital influence on a series of important performance indicators such as load balance, setup time, setup cost, timeliness, flow of material etc. Generating appropriate sequences has thus become a key task for the automotive production planner. Ensuring the stability of a sequence, once it has been determined, by sorting its actual version back into the form intended originally, however, with systematic means, is a new issue gaining recently more importance. This problem represents the main topic of the present paper. Instruments for physically resorting vehicles are sorting channels and parking spaces. Both instruments are closely related to one another, as will be shown.

1 INTRODUCTION

Problems and solution methods in the sequencing domain have been widely studied in the past, both from the theoretical as well as the applicational point of view, viz. (Chrètienne et al., 1995; Mergenthaler et al., 1995; Mergenthaler et al., 1994).

Since combinatorial optimization is at the mathematical core of a sequencing problem, an optimal sequence can be found by using heuristics and local search techniques, viz. (Reeves, 1996). Once an optimal sequence has been found for a set of vehicles, for the sake of generality called objects below, it is by no means guaranteed in practice, that this sequence will remain stable throughout the production process. A perennial source of disturbance is the paint shop, for example, where rework is frequent.

Assume n objects numbered $1,2,3,\dots,n$ arrive at a certain point in an arrival queue according to a sequence, or permutation $\rho = \{\rho^1,\dots,\rho^n\}$, ρ^1 arriving first, ρ^n arriving last and set $N = \{1,\dots,n\}$. The objects need to undergo a certain process starting at this point in the sequence $\tilde{\rho} = \{1,\dots,n\}$, however, and must therefore be resorted correspondingly. There are two instruments to implement the resorting task: sorting channels and parking spaces. Once an object is assigned to a sorting channel it can only be removed from the bottom of the channel at the end of the assignment process. Alternatively it can be parked on a parking space until an assignment to a channel becomes possible. The quantity of parked objects should be minimized because of cost considerations. The sorting channels, in this paper, are assumed to have unlimited capacities, the parking space is limited.

An example sorting process with 7 objects to resort and a factory layout of three channels and four parking spaces is given in figure 1. The pipeline of objects is considered to be processed from right to left. The arrows show one possible assignment strategy. The process starts by moving object 7 to a parking space. All other objects are directly assigned to a channel marked by the arrows. Finally object 7 is moved from the parking space to the first channel. Based on this assignment the original order $\{1,\dots,7\}$ can be restored by pulling the objects from the bottoms of the channels.

Both sorting channels and parking spaces require investments which naturally the factory planner wants to minimize. One question of interest in this paper will be, how many sorting channels and parking spaces should be installed. Once the factory layout has been fixed, only the channel and parking space assignment can be controlled by the plant manager. It is straightforward to ask for the optimal assignment strategy which minimizes the number of objects to be parked at any time.

This paper is organized as follows: Chapter 2

Feller J., Mauersberg B., Mergenthaler W., Feller Y. and Müller L..
Resorting Vehicles in an Automotive Manufacturing Environment.
DOI: 10.5220/0005087601840191
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 184-191
ISBN: 978-989-758-040-6

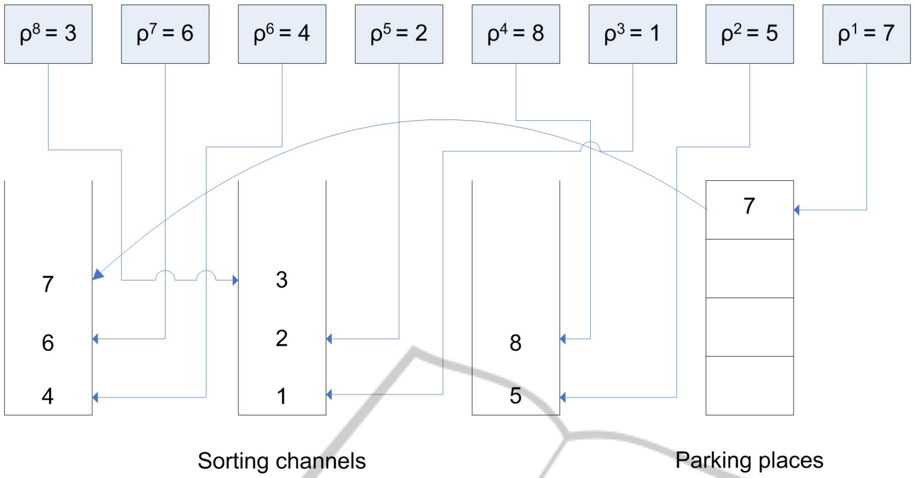

Figure 1: Sorting channels and parking spaces.

shows that the original order can be restored by dividing the arbitrary queue into ascending subsequences and discusses the tradeoff between sorting channels and parking spaces. Furthermore it shows that permutations can be subdivided into a set of contiguous subsequences such that statements about maximum parking space can be reduced to statements restricted to those subsets. An important result says that permutations tend towards contiguous subsets as the size of the permutations tends to infinity. Based on these results in Chapter 3 a dynamic optimization model is formulated in order to calculate an optimal assignment strategy for the factory planner. Chapter 4 describes an appropriate solution heuristic. Chapter 5 concludes this paper with an example.

2 MAIN RESULTS

2.1 Definitions

Let m be the number of sorting channels and set $M := \{1,\dots,m\}$. Let $x_i \in M$ be the channel assigned to object $\rho^i, i \in N$. Let P be the set of all permutations of the objects in N. The set of all possible channel assignments is defined as $X := M^n$. Let $S_t(x,\rho)$ be the set of objects parked right after ρ^t has been drawn from the pipeline, with $x \in X, \rho \in P, t \in N$.

2.2 The Importance of Ascending Subsequences

In this section the importance of ascending subsequences in each channel with respect to generating the original sequence will be discussed.

Lemma 1. *Let $T_\mu \subseteq N$, $\mu \in M$ be m ordered ascending subsets of N such that*

$$T_\mu = \{\rho_\mu^1,\dots,\rho_\mu^{n_\mu}\},$$
$$n_\mu = card(T_\mu),$$
$$\rho_\mu^1 \le \rho_\mu^2 \dots \le \rho_\mu^{n_\mu}$$
$$T_\mu \cap T_\nu = \emptyset, \nu \in M - \{\mu\},$$
$$\bigcup_{\mu \in M} T_\mu = N \quad \text{and}$$
$$\forall \mu \in M, l \in \{1,\dots,n_\mu\}\ \exists w(l,\mu) \in N : \rho_\mu^l = \rho^{w(l,\mu)}$$

Then there is a sequence $\zeta = \{\zeta_1,\dots,\zeta_n\}$ where $\zeta_i \in M, \forall i \in \{1,\dots,n\}$ *such that* $\rho_{\zeta_1}^1 = 1$, $\rho_{\zeta_2}^{k(2,\zeta_2)} = 2, \dots, \rho_{\zeta_n}^{k(n,\zeta_n)} = n$, $k(\nu,\zeta_\nu) = \sum_{j=1}^{\nu-1} \delta_{\zeta_\nu \zeta_j} + 1, \nu \in N$ *and* $\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & else \end{cases}$ *is the Kronecker symbol.*

Proof. The proof is by induction. Assume that, after $l \in N$ steps the objects

$$\rho_{\zeta_1}^1 = 1, \rho_{\zeta_2}^{k(2,\zeta_2)} = 2, \dots, \rho_{\zeta_l}^{k(l,\zeta_l)} = l \qquad (1)$$

have been drawn from the respective channels $z_1,\dots,z_l$. This is clearly the case for $l = 1$. If $\rho_\nu^1 \ne 1\, \forall \nu \in M$ then, since there must exist one channel $\nu^* \in M$ containing element 1, i.e. $p_{\nu^*}^{y(\nu^*)} = 1$ for some $y(\nu^*)$ with $2 \le y(\nu^*) \le n_{\nu^*}(l)$ one obtains

$$\rho_{\nu^*}^1 > 1 \text{ and } \rho_{\nu^*}^{y(\nu^*)} = 1 \qquad (2)$$

violating the assumption of ascending subsequences in each channel. $n_\nu(l), \nu \in M$ denotes the number of objects in channel $\nu \in M$ after $l \in N$ objects have been assigned.

If, in general, 1 holds for some arbitrary l with $1 \leq l \leq n$, then there must be some channel ζ_{l+1} with $\rho_{\zeta_{l+1}}^{k(l+1,\zeta_{l+1})} = l+1$ because, otherwise, the assumption of ascending subsequences would be violated again □

Verbally, lemma 1 says that when an arbitrary sequence, i.e. a permutation of the numbers 1 through n, can be partitioned into a series of ascending subsequences, then the original sequence can be restored by pulling consecutively the bottoms from the subsequences. On the other hand, the existence of ascending subsequences is also necessary for the existence of the ordered sequence $\tilde{\rho}$. If at least one subsequence is not ordered, then obviously $\tilde{\rho}$ cannot be restored by pulling the respective bottoms from the sequences. This is a basic result for the optimization model where the number of possible assignments is reduced to assignments which must create ascending subsets in the respective channels.

2.3 Tradeoff Between Channel Size and Parking Spaces

Recall that the channel assignment $x \in X$ is defined such that

$$x_\nu = \mu \Longleftrightarrow \exists l \in \{1,\dots,n_\mu\} : \rho_\mu^l = \rho^\nu, \nu \in N, \mu \in M$$

One can now show that the following holds:

Lemma 2. *The set of objects parked at time t under policy x and for an incoming sequence* ρ *is equal to*

$$S_t(x,\rho) = \bigcup_{i \in \{1,\dots,t\}, \prod_{j=t+1,x_j=x_i}^{n} 1_{\{\rho^j > \rho^i\}} = 0}^{t} \rho^i \qquad (3)$$

Proof. After step t all those objects ρ^i, $i \in \{1,\dots,t\}$ must be parked,who have at least one successor within $\{\rho^{t+1},\dots,\rho^n\}$ smaller than ρ^i, but are assigned to the same channel. Thus an object must be parked, if

$$\prod_{j=t+1,x_j=x_i}^{n} 1_{\{\rho^j > \rho^i\}} = 0 \qquad (4)$$

Equation 4 is identical to the filtering condition in equation 3 □

Two special cases may illustrate lemma 2: Special case 1: Let $\rho = \{1,\dots,n\}$. In this case it is obvious that

$$S_t(x,\rho) = \emptyset, \quad \forall t = 1,\dots,n \qquad (5)$$

if, for arbitrary $\mu \in Mx_1 = \mu,\dots,x_n = \mu$. for instance. There may be other assignments also yielding 5.

Special case 2: Let $\rho = \{n,\dots,1\}$. Then, with m channels given, exactly $\lceil \frac{n}{m} \rceil - 1$ parking spaces are needed. This can be seen by noting that a possible assignment policy would be given by $r = \left\lceil \frac{n}{m} \right\rceil$ and

$$\begin{aligned} x_1 = 1, \quad &\dots \quad ,x_r = 1 \\ x_{r+1} = 2, \quad &\dots \quad ,x_{2r} = 2 \\ &\dots \\ x_{(m-1)r+1} = m, \quad &\dots \quad ,x_n = m. \end{aligned}$$

In this case channel 1 receives objects ρ^1 through ρ^r, requiring exactly $r-1$ parking spaces. The same holds for channels 2 through $m-1$. For channel m exactly $(n \, mod \, r) - 1$ parking spaces are needed, if $(n \, mod \, r) > 0$, $r-1$ else.

Let $Z_t(x,\rho) := card(S_t(x,\rho))$. It then follows immediately from lemma 2 that

$$Z_t(x,\rho) = \sum_{i=1}^{t} (1 - \prod_{j=t+1,x_j=x_i}^{n} 1_{\{\rho^j > \rho^i\}})$$

Two worst case approximations on the number of parking spaces given the number of channels, or - the number of channels, given the number of parking spaces - are readily obained.

Corollary 1. *In order to generate m ascending subsequences from an arbitrary permutation* ρ *the number of required parking spaces* $A(n,m)$ *satisfies*

$$A(n,m) \leq \left\lceil \frac{n}{m} \right\rceil - 1 \qquad (6)$$

Proof. Assign the first $\left\lceil \frac{n}{m} \right\rceil$ objects to channel 1, the second $\left\lceil \frac{n}{m} \right\rceil$ objects to channel 2, etc. Then, in the worst case one deals with special case 2 and the corollary is proven □

Corollary 2. *Assume there are r parking spaces available. Then for any permutation* ρ *there is an assignment policy such that the number of required channels* $B(n,r)$ *satisfies*

$$B(n,r) \leq \left\lceil \frac{n}{r} \right\rceil \qquad (7)$$

Proof. From 6 it follows that $r \leq \lceil \frac{n}{B(n,r)} \rceil - 1$ must hold.Therefore

$B(n,r)r \leq B(n,r) \lceil \frac{n}{B(n,r)} \rceil - B(n,r)$. On the other hand $\lceil \frac{n}{B(n,r)} \rceil \leq \frac{n}{B(n,r)} + 1$.

Hence $B(n,r)r \leq B(n,r)(\frac{n}{B(n,r)} + 1) - B(n,r) = n$ and therefore $B(n,r) \leq \frac{n}{r}$. Since $B(n,r)$ is an integer, 7 is correct □

In the special case, where there is no parking space available, there is a simple sufficient criterion for the necessity of at least m sorting channels.

Corollary 3. *A sufficient criterion for the necessity of at least m sorting channels, if there is no parking space available, is the occurrence of a contiguous, descending subsequence of length m.*

Proof. Let, for some t with $1 \leq t \leq n-m$

$$\rho^t \geq \rho^{t+1} \geq \ldots \geq \rho^{t+m}$$

Then, even in the best case, where all the objects $\rho^1, \ldots, \rho^{t-1}$ have been accommodated to their respective channels due to the fact that

$$\prod_{j=t, x_j=x_i}^{n} 1_{\{\rho^j > \rho^i\}} = 1, \quad 1 \leq i \leq t-1$$

all of the objects $\rho^t, \rho^{t+1}, \ldots, \rho^{t+m-1}$ can safely be stored in the channels numbered $1, \ldots, m$ and must be distributed across them. Therefore ρ^{t+m} will only find channels covered with objects larger than itself. Hence, a new channel is required. □

In the dynamic optimization model in 3 either the channel quantity or the size of the parking space is given. The other value is calculated based on Corollary 1 or 2.

2.4 The Impact of Contiguous Subsequences

Assume that the set N can be partitioned into p subsets N_α, $1 \leq \alpha \leq p$ such that

$$N_\alpha \cap N_\beta = \emptyset, \forall \alpha, \beta \in \{1, \ldots, p\}, \alpha \neq \beta$$

$$\bigcup_{\alpha=1}^{p} N_\alpha = N$$

$$N_\alpha = \{\rho^l(\alpha), \ldots, \rho^u(\alpha)\}, \alpha \in \{1, \ldots, p\}$$

where each subset N_α satisfies the following conditions:

$$min\, N_\alpha = \begin{cases} 1 + \max N_{\alpha-1}, & \alpha > 1 \\ 1, & \alpha = 1 \end{cases} \tag{8}$$

$$\forall \rho \in \{\min N_\alpha, \min N_\alpha + 1, \ldots, \max N_\alpha\} : \rho \in N_\alpha \tag{9}$$

and, where for each $A \subset N_\alpha$ subset A does not satisfy (8 - 9) and where finally $l(\alpha)$ and $u(\alpha)$ denote the lower and the upper limit of the index set N delimiting subset N_α.

This way of partitioning N ensures that the subsets can be arranged such that

1. Any subset contains elements strictly greater than all the elements in the previous subsets
2. Any subset contains all the elements between its minimum and its maximum
3. No subset of a contiguous subset is a contiguous subset

With this definition an important result can be derived.

Lemma 3. *For each $t \in N$ let $\alpha(t)$ be defined through $\rho^t \in N_{\alpha(t)}$. Then*

$$S_t(x, \rho) = \bigcup_{i \in N_{\alpha(t)}, i \leq t, \prod_{j=t+1, x_j=x_i}^{n} 1_{\{\rho^j > \rho^i\}} = 0} \rho^i$$

Proof. Starting from 3, consider what happens if $i \in N_1 \cup N_2 \cup \cdots \cup N_{\alpha(t)-1}$. In this case one knows that, by definition,

$$\rho^i < min_{j \in N_{\alpha(t)} \cup N_{\alpha(t)+1} \cup \cdots \cup N_p}\, \rho^j = min_{j \in \{t+1, \ldots, n\}} \rho^j$$

with the consequence that

$$\prod_{j=t+1}^{n} 1_{\{\rho^j > \rho^i\}} = 1$$

and $\prod_{j=t+1, x_j=x_i}^{n} 1_{\{\rho^j > \rho^i\}} = 1$.

Therefore, the filtering condition in the union index in 3 reduces to the one shown above, proving lemma 3. □

Verbally lemma 3 says that, in order to determine the set of objects in the parking space at any time, only those objects qualify, which belong to the same contiguous subset as the actual object.

In the rest of the section contiguous subsets will be analyzed and, in particular, the chance to find them in random permutations. Therefore the following definitions are needed:

- Let $\beta(n)$ be the number of permutations of the set $\{1, \ldots, n\}$, which form a contiguous subset and therefore contains no other proper contiguous subset.
- Let $\gamma(k, n)$ be the number of permutations of the set $\{1, \ldots, n\}$, which contain at least one contiguous subset of length $1 \leq k \leq n$.

The following result holds:

Lemma 4.

$$\beta(n) = n! - \sum_{k=1}^{n-1} \beta(k)(n-k)! \tag{10}$$

Proof. It is obvious that

$$\gamma(k,n) = \beta(k)(n-k)!$$

because the subset $\{1,\ldots,k\}$ is contiguous, therefore can be organized in $\beta(k)$ ways and the remaining elements can be ordered into $(n-k)!$ ways. Also, if $\{1,\ldots,n\}$ forms a contiguous subset and therefore contains no contiguous subset of smaller size, from the $n!$ permutations of the set $\{1,\ldots,n\}$ those must be subtracted, which contain a subset of length $1,\ldots,n-1$, i.e.

$$\beta(n) = n! - \sum_{k=1}^{n-1} \gamma(k,n) \tag{11}$$

Inserting 2.4 into 11 proves the lemma. □

10 generates the following sequence, as is straightforward to verify:

$\beta(1) = 1, \beta(2) = 2, \beta(3) = 3, \beta(4) = 13, \beta(5) = 71, \ldots$.

Unfortunately one can show that $\beta(n)$ tends to $n!$ for n going towards infinity, meaning that less and less permutations contain contiguous subsets, thus destroying the hope of being able to apply lemma 3 to large permutations in a systematic way.

Corollary 4.

$$lim_{n\to\infty} \frac{\beta(n)}{n!} = 1 \tag{12}$$

Proof. 10 yields, after some rearrangements of terms

$$\sum_{k=1}^{n} \frac{\beta(k)}{k!} \frac{k!(n-k)!}{n!} = 1 \tag{13}$$

or, explicitly

$$\frac{\beta(1)}{1!}\frac{1!(n-1)!}{n!} + \frac{\beta(2)}{2!}\frac{2!(n-2)!}{n!} + \cdots + \frac{\beta(n-1)}{(n-1)!}\frac{(n-1)!1!}{n!} + \frac{\beta(n)}{n!}1 = 1$$

Abbreviate

$$C(n) := \frac{1!(n-1)!}{n!} + \frac{2!(n-2)!}{n!} + \cdots + \frac{(n-1)!1!}{n!} = \sum_{k=1}^{n-1} \frac{1}{\binom{n}{k}}$$

If it can be proven that

$$lim_{n\to\infty} C(n) = 0 \tag{14}$$

then one obtains

$$lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{\beta(k)}{k!} \frac{k!(n-k)!}{n!} = 0$$

because $\beta(k) < k!$ and, therefore $\frac{\beta(k)}{k!} < 1$ yielding 12 upon using 13. In order to prove 14 one proceeds along the following lines:

$$\begin{aligned}
C(n+1) &= \sum_{k=1}^{n} \frac{1}{\binom{n+1}{k}} \\
&= \sum_{k=1}^{n-1} \frac{1}{\binom{n+1}{k}} + \frac{1}{n+1} \\
&= \sum_{k=1}^{n-1} \frac{k!(n+1-k)!}{(n+1)!} + \frac{1}{n+1} \\
&= \sum_{k=1}^{n-1} \frac{k!(n-k)!}{n!}\frac{(n+1-k)}{(n+1)} + \frac{1}{n+1} \\
&= \sum_{k=1}^{n-1} \frac{k!(n-k)!}{n!} - \sum_{k=1}^{n-1} \frac{k!(n-k)!k}{n!(n+1)} + \\
&\quad \frac{1}{n+1} \\
&= C(n) - C(n+1) + \frac{C(n)}{n+1} + \frac{2}{n+1}
\end{aligned}$$

after some lengthy transformations. Therefore

$$C(n+1) = \frac{(n+2)}{2(n+1)} C(n) + \frac{1}{n+1} \tag{15}$$

Set

$$C_\infty = \lim_{n\to\infty} C(n) \tag{16}$$

15 and 16 together produce $C_\infty = \frac{1}{2} C_\infty$ which can only be true for $C_\infty = 0$ which is equivalent to 14 and therefore proves the lemma. □

Corollary 4 tells us that, as n tends towards infinity, almost every permutation of the set $\{1,\ldots,n\}$ is a contiguous subset. So, lemma 3, which allows for permutations with limited parking space, will be less and less applicable for large n.

3 DYNAMIC OPTIMIZATION MODEL

The main objective of this chapter is to find an optimal assignment strategy for the factory planner which minimizes the number of parked objects for a given factory layout at any time. Based on the results of chapter 2 the integer programming problem is represented as a dynamic optimization model.

As in (Neumann and Morlock, 1993) the following dynamic optimization model is defined.

$$\begin{array}{ll} \text{Minimize} & \sum_{t=1}^{T} c_t(z_t,u_t) \\ \text{Subject to} & z_{t+1} = f_t(z_t,u_t), \quad t=1,\dots,T \\ & z_1 = z_a \\ & z_{t+1} \in Z_{t+1} \\ & u_t \in U_t(z_t), \quad t=1,\dots,T \end{array}$$

Definitions:

1. In every state t either an object is assigned to a channel or parked in it. In case of an assignment the object can be the next object in the queue or an object which is parked. Let $N' \subset N$ be the set of objects parked ever in a parking space and $n' = |N'|$. Then $T = n + n'$ is the total number of states and defines the horizon of the optimization problem.
2. The number of available parking spaces or the number of available channels should be given. The other variable can be calculated based on corollary 1 and 2. Let p_{max} be the maximum number of available parking spaces and m the number of channels.
3. $z_t = (\lambda_0(t), p(t), (\lambda_1(t),\dots,\lambda_{p(t)}(t)), (\rho_1^{\eta_1}(t),\dots,\rho_m^{\eta_m}(t))$ where $\lambda_0(t)$ is the next object in the queue, $p(t)$ the number of occupied parking spaces, $\lambda_i(t) \in N' \subset N, \quad i=1,\dots p(t)$ the objects on a parking space, $\rho_\mu^{\eta_\mu(t)} \in N$ the top-most objects of channel $\mu \in M$. If no object is assigned $\rho_\mu^{\eta_\mu(t)} = 0$.
4. $z_a = (\lambda_0(1), 0, (0,\dots,0))$ with $\lambda_0(1) = \rho^1$ the first element in the queue, $p(1) = 0$ occupied parking spaces and m empty channels.
5. $\Lambda(t) = \cup_{i=0}^{p(t)} \{\lambda_i(t)\}$ are candidates waiting for an assignment at time t.
6. The decision space is described by the union of two subsets $U_t^i(z_t), i=1,2$. The first set $U_t^1(z_t)$ describes the assignment of an object to a channel. Possible candidates are elements from $\Lambda(t)$. The second set $U_t^2(z_t)$ describes the alternative that the next object in the queue is parked.
   In detail the subsets are defined as follows:
   $$\begin{array}{lcl} U_t^1(z_t) & = & \{(j_t,\tilde{\mu}_t) : j_t \in \Lambda(t) \wedge \tilde{\mu}_t \in M\} \\ U_t^2(z_t) & = & \text{park}\, \lambda_0(t) \end{array}$$
7. Let κ be the index of the actual element in the queue starting with $\kappa = 1$ in $t = 1$.
8. Depending on $u_t \in U_t$ the following states $z_{t+1} = f_t(z_t,u_t)$ are obtained as follows:
   $u_t = (j_t, \tilde{\mu}_{j_t}) \in U_t^1(z_t)$:
   $j_t = 0$:
   $\kappa = \kappa + 1$
   $\lambda_0(t+1) = \rho^\kappa$
   $\lambda_i(t+1) = \lambda_i(t), i = 1,\dots,p(t)$
   $p(t+1) = p(t)$
   $j_t > 0$:
   $\lambda_i(t+1) = \lambda_i(t), i = 0,\dots,j_t - 1$
   $\lambda_i(t+1) = \lambda_{i+1}(t), i = j_t,\dots,p(t)-1$
   $p(t+1) = p(t) - 1$
   $\forall j_t \geq 0$:
   $\rho_\mu^{\eta_\mu}(t+1) = \rho_\mu^{\eta_\mu}(t), \mu \in M \setminus \{\tilde{\mu}_{j_t}\}$
   $\rho_{\tilde{\mu}_{j_t}}^{\eta_{\tilde{\mu}_{j_t}}}(t+1) = \lambda_j(t)$

   $u_t \in U_t^2(z_t)$:
   $\kappa = \kappa + 1$
   $\lambda_0(t+1) = \rho^\kappa$
   $\lambda_i(t+1) = \lambda_i(t); i = 1,\dots,p(t)$
   $\lambda_{p(t)+1}(t) = \lambda_0(t)$
   $p(t+1) = p(t) + 1$
   $\rho_\mu^{\eta_\mu}(t+1) = \rho_\mu^{\eta_\mu}(t), \forall \mu \in M$
9. Lemma 1 says that the existence of ordered subsequences is necessary for the existence of the ordered sequence $\tilde{\rho}$. Therefore the set of feasible assignment candidates in t is $\hat{\Lambda}(t) = \{j_t \in \Lambda(t) \wedge \exists \mu \in M : j_t > \rho_\mu^{\eta_\mu(t)}\}$. In order to make sure that the optimization problem chooses only feasible assignments, infinitely high penalty cost for assigning $j_t \in \Lambda(t) \setminus \hat{\Lambda}(t)$ are defined. Also, exceeding the maximal number of parking spaces leads to infinitely high cost. Parking an object before it is assigned leads to setup cost c_p. The overall cost in t with $t = 1,\dots,T$ can be defined as:
   $$c_t(z_t,u_t) = \max(p(t+1) - p(t), 0)c_p + \delta_t$$
   with δ_t defined as:
   $$\delta_t = \begin{cases} \infty, \text{if } j_t \in \Lambda(t) \setminus \hat{\Lambda}(t) \\ \infty, \text{if } p(t) = p_{max} \wedge u_t \in U_t^2 \\ 0, \quad else \end{cases}$$

4 HEURISTIC

4.1 Basic Ideas

Based on the definitions in 3 in every step the following question has to be answered:

1. Should the first object in the channel or one of the objects in the parking spaces be assigned to a channel (U^1) or
2. should the first object in the queue be parked (U^2).

U^1 includes $(p(t)+1)m$ different actions which lead to $((p(t)+1)m)^{n'}$ possible states. In total

$((p(t)+1)m+1)^{n'}$ states are possible. In this chapter a heuristic is developed which, in every step, selects only those states, which are locally likely to produce a good solution.

Assume that the channels after every assignment are renumbered so that $\rho_1^{\eta_1}(t) \geq \rho_2^{\eta_2}(t) \geq \cdots \geq \rho_m^{\eta_m}(t)$.

Then the state

$$z_t^* = (\lambda_0(t), p(t), (\lambda_1(t), \ldots, \lambda_{p(t)}(t)), (\rho_1^{\eta_1 *}(t), \ldots, \rho_m^{\eta_m *}(t))$$

dominates

$$z_t = (\lambda_0(t), p(t), (\lambda_1(t), \ldots, \lambda_{p(t)}(t)), (\rho_1^{\eta_1}(t), \ldots, \rho_m^{\eta_m}(t))$$

if

$$\begin{aligned} \forall j \in M: \quad & \rho_j^{\eta_j *}(t) \leq \rho_j^{\eta_j}(t) \\ \exists j \in M: \quad & \rho_j^{\eta_j *}(t) < \rho_j^{\eta_j}(t) \end{aligned}$$

because z_t^* allows for at least one more assignment, such as for instance of the element $\rho_j^{\eta_j *}(t)+1$.

It is now shown how actions can be chosen which automatically lead to a state that dominates all other possible states.

Lemma 5. *For every $\lambda_j(t) \in \hat{\Lambda}(t)$ a channel μ_j^* exists which is defined by the following equation:*

$$\begin{aligned} d_j^{\mu}(t) &= \lambda_j(t) - \rho_{\mu}^{\eta_{\mu}(t)} \\ \tilde{M} &= \{\tilde{\mu} \in M | d_j(t) \geq 0\} \\ d_j^{\tilde{\mu}_1}(t) &\leq d_j^{\tilde{\mu}_2}(t) \leq \cdots \leq d_j^{\tilde{\mu}_{\tilde{m}}}(t), \tilde{m} = |\tilde{M}| \\ \mu_j^* &= \tilde{\mu}_1 \end{aligned} \tag{17}$$

Proof. If λ_j is assigned to another possible channel $\sigma \in M$ with $\sigma \neq \mu_j^*$ and $\lambda_j - \sigma > 0$ in t all top most elements in $t+1$ are the same as in t except the topmost element of channel μ_i which changes from $\rho_{\sigma}^{\eta_{\sigma}}$ to λ_j and the top most element from μ_j^* stays $\rho_{\mu_j^*}^{\eta_{\mu_j^*}}$.

When μ_j^* is selected $\rho_{\mu_j}^{\eta_{\mu_j}}$ changes to λ_j and $\rho_{\sigma}^{\eta_{\sigma}}$ stays the same.

The states in $t+1$ will only be different in the top most elements $\rho_{\sigma}^{\eta_{\sigma}}$ and $\rho_{\mu_j^*}^{\eta_{\mu_j^*}}$. All other elements remain the same.

Based on (17) the following holds:

$$\lambda_j - \rho_{\sigma}^{\eta_{\sigma}} \leq \lambda_j - \rho_{\mu_j^*}^{\eta_{\mu_j^*}} \Leftrightarrow \rho_{\mu_j^*}^{\eta_{\mu_j^*}} \leq \rho_{\sigma}^{\eta_{\sigma}}$$

which contradicts the required dominating state of z_t^*. □

In the worst case we still have to assign one of $p(t)+1$ elements in every state.

With

$$\begin{aligned} e_j^{\lambda}(t) &= \lambda_j(t) - \rho_{\mu}^{\eta_{\mu}(t)} \\ \hat{\Lambda}(t) &= \{j_t \in \Lambda(t) \wedge \exists \mu \in M: \ j_t > \rho_{\mu}^{\eta_{\mu}(t)}\} \\ e_j^{\hat{\lambda}_1}(t) &\leq e_j^{\hat{\lambda}_2}(t) \leq \cdots \leq e_j^{\hat{\lambda}_l}(t), \quad l = |\hat{\Lambda}| \\ \lambda_j^* &= e_j^{\hat{\lambda}_1}(t) \end{aligned} \tag{18}$$

under the assumption of unlimited parking space analog to Lemma 5 λ_j^* leads automatically to a dominate state.

4.2 Implemented Rules

Based on these ideas the following heuristic is implemented.

In the first iteration step p_{max} is calculated using 1. Afterwards $\forall z_t \in Z_t$ is done:

1. Determine $\hat{\Lambda}(t) = \{j_t \in \Lambda(t) \wedge \exists \mu \in M: \ j_t > \rho_{\mu}^{\eta_{\mu}(t)}\}$.
2. If $\hat{\Lambda}(t) = \emptyset$ at least one object causes infinitely high cost when it is assigned to a channel and no further analysis of the state is necessary.
3. If $\hat{\Lambda}(t) \neq \emptyset$ two actions are possible.
   (a) Select $\lambda_j^*(t)$ according to 18 and assign it to the channel $\mu_{\lambda_j^*}^*(t)$ determined by 17.
   (b) Move the actual object to a parking space. If $p(t) = p_{max}$ no parking space is available. Then select $\lambda_j^*(t)$ and $\mu_{\lambda_j^*}^*(t)$ as in 3a. If $\lambda_j^*(t) \neq \rho^{\kappa}$ p(t+1) = p(t)-1 and parking is an option in the next step.

Future work will analyze in detail if these rules at least lead to local optimal states in every $t = 1, \ldots, T$.

5 EXAMPLE AND DISCUSSIONS

For illustration the following example is given. Assume 30 objects numbered 1,2,...,30 arrive at a certain point in an arrival queue according to a sequence $\rho = \{5,12,20,26,30,19,29,18,14,8,13,21,2,11,25,15,22,9,17,4,3,24,28,1,6,16,10,27,23,7\}$, 5 arriving first and 7 arriving last. Three sorting channels are available. According to lemma 2 the following 28 parking candidates can be identified $N' = \{5,12,20,26,30,19,29,18,14,8,13,21,2,11,25,15,22,9,17,4,3,24,28,6,16,10,27,23\}$.

Table 1: Assignment strategies (corollary 1 and chapter 4).

ρ	x_c	x_h	ρ	x_c	x_h	ρ	x_c	x_h
5	1	1	13	**2**	**2**	3	**3**	**3**
12	**1**	1	21	**2**	**2**	24	**3**	**2**
20	**1**	1	2	2	2	28	**3**	**2**
26	**1**	1	11	**2**	2	1	3	3
30	**1**	1	25	**2**	2	6	3	3
19	**1**	**2**	15	**2**	2	16	**3**	**3**
29	**1**	**2**	22	**2**	2	10	3	**3**
18	**1**	**2**	9	**2**	**3**	27	**3**	**3**
14	**1**	**2**	17	**2**	**3**	23	**3**	**3**
8	1	**2**	4	2	**3**	7	3	3

With Corollary 1 a limited number of $A(30,3) \leq \left\lceil \frac{30}{3} \right\rceil - 1 = 9$ parking spaces can be determined.

In table 1 x_c is an assignment strategy developed based on the ideas in the proof of corollary 1 and x_h an assignment strategy calculated using the heuristc 4 reducing the quantity of parked objects from 22 to 17. When an object is parked before it is assigned the number of the assignment strategy is marked bold.

Based on Lemma 1 the ordered sequence $\{1,\ldots,30\}$ can be realized by taking the bottoms from the channels and the original problem is solved.

6 CONCLUSIONS

The heuristic helps the factory planner to find a feasible solution for his assignment problem which in general uses a smaller quantity of parked objects than the intuitive assignment strategy of 1.

Ongoing research looks for additional improvements. Furthermore it would be interesting to analyse what a flexible layout (channel and parking space quantity) should look like which supports most of possible object permutations. Another question could be how in practice existing channel and parking space restrictions can be integrated in the model. Instead of the current heuristic considered above, other solution techniques such as Simulated Annealing or Genetic Algorithms could be considered. However, since those techniques usually provide less insight into the "mechanics" of a problem under investigation, the above approach was used.

REFERENCES

Chrètienne, P. et al., editors (1995). *Scheduling Theory and its Applications*. Wiley, Chichester.

Korte, B. and Vygen, J. (1995). *Combinatorial Optimization - Theory and Algorihms*. Springer, Berlin.

Mergenthaler, W., Stadler, W., and Zimmer, N. (OCRI, Ottawa, Ontario, Canada, 1994). Dynamic scheduling and job assignment in an automotive manufacturing process. *The 10th ISPE/IFAC International Conference on CAD/CAM, Robotics and Factories of the Future: CARS & FOF'94, M. Zaremba, pp 478-483*.

Mergenthaler, W., Stadler, W., Zimmer, N., and Wilbertz, H. (1995). Optimizing automotive manufacturing sequences using simulated annealing and genetic algorithms. *Control Eng. Practice, Vol. 3, pp 569-573*.

Neumann, K. and Morlock, M. (1993). *Operations Research*. Hanser, München.

Reeves, C. R. (1996). *Modern Heuristic Techniques, in: V. J. Rayward-Smith, I. H. Osman, C. R. Reeves and G. D. Smith:* Modern Heuristic Search Methods. Wiley, Chichester.