THE INTELLIGENT SCENARIO SELECTION IN DYNAMIC HOIST SCHEDULING PROBLEM: THE REAL-LIFE ELECTROPLATING LINE CASE

Krzysztof Kujawski, Jerzy Świątek
Institute of Informatics, Wrocław University of Technology Wybrzeże Wyspiańskiego
27, 50-370 Wrocław, Poland

ABSTRACT

The paper presents a scheduling system which was designed for a electroplating production line. The system allows flexible production, in the real time environment. The number of transport hoists is routed as well as chemical and material treatment processing stations are scheduled in order to create a functional production line. The expert knowledge is used for preparation of production scenarios. The system manages to create schedules in the real time because of using machine-learning methods to identify production system state and support the decisive process of scenario selection. The Intelligent Scenario Selection Method is used on the real-life production line located in Wrocław, Poland. The paper presents issues that occur on analyzed line and the results achieved by using The Intelligent Scenario Selection Method.

Keywords: hoist scheduling, real-time production systems, flexible manufacturing systems, machine learning

HOIST SCHEDULING PROBLEM

In electroplating industry a HSP – the Hoist Scheduling Problem occurs. The problem lies in creating a proper schedule for machines working on a production line. The schedule is used during production by the processing machinery. When machinery works accordingly to the schedule, the processing is performed. On electroplating lines items are chemically processed. The items are transported from a workstation to a workstation by automated hoists. An electroplating line can produce multiple item types. In case of electroplating industry, usually metal elements are coated with noble metals e.g. nickel, chrome. The Hoist Scheduling Problem also occurs in other industries, production of printed circuit boards and food processing. Each produced item type has its own sequence of visiting workstations, processing intervals, etc. In practice, production lines produce multiple item types because either a line was designed to perform multiple technological processes or within single technological process (e.g. chroming), items vary in size or other properties and require different sequence or times of processing.

Depending on the used scheduling system, the production can be organized cyclically (the Cyclic Hoist Scheduling Problem – CHSP) and dynamic (the Dynamic Hoist Scheduling Problem - DHSP) - order driven. In the cyclic production a static schedule, called a cyclogram, is repeated a certain number of times in order to produce items. Such organization has its advantages, like simplicity and predictability, but lack in flexibility. In case line is supposed to process multiple item types, low flexibility leads to resource waste and may make the whole system less profitable. In case of DHSP, a schedule is created and is adapted constantly during production. Produced items vary both in type and in time of introduction to line. We present the scheduling system, which divides the problem to the Local Problem (Lamothe 1996) and the real-time part. Real-time part reacts on new item orders, verifies the feasibility of schedules and implements schedules to line automatons.

DYNAMIC HOIST SCHEDULING PROBLEM

Electroplating lines cover processed items with thin material coatings using chemical reactions – usually by galvanization. They are automated production systems, which use hoists as transportation. Production line is made of series of workstations. Each workstation is capable of performing different elementary chemical reaction or material processing operation. Production is based on subjecting the input products to several reactions in specific sequence.

Production line consists of: *Baths (tanks)* and *Hoists*. Baths are workstations that perform a certain stage of the chemical or material processing. Groups of uniform workstations are also considered. Some workstations can perform more than one stage in processing – they are called multifunctional workstation. Baths are arranged in line, and they create an axis of hoist movement. Hoists – automatons controlled by schedule. Hoists are capable of transferring products between workstations. Hoists move only in production line axis. Hoists cannot pass each other. A hoist can pick up and put down products to workstation. Hoists cannot pass products between each other. The processing starts when a hoist plunge item to a bath. This happens because of the chemical nature of the processing. When a hoist plunge an item to workstation we refer to it as the operation of putting down of an item. The processing stops, when the hoist picks item out of a bath. In case of chemical processing, instead of time of processing at a certain workstation, we have quality constrains. It means that an item is immersed in the workstation for at least given minimum time and no longer than given maximum time.

In order to increase flexibility of the production system, some production lines are designed to produce many item types. This is done by composing workstations, which are required to produce the certain type of item. Scheduling item types differ in sequence of visited workstations and times of processing at those workstations.

The electroplating production lines scheduling was described in many publications. One of the first papers about HSP Phillips 1976 describes the simple application, where the production line has only one hoist and simple sequence of production – always to next workstation. More recent publications Varnier 1997. Yan 2008, Liu 2002 expanded the problem to multiple hoists, Leung 2003, Mak 2002 an arbitrary production sequence, groups of workstations, multifunctional baths Mak 2002. Mentioned papers describe creating of cyclograms in the cyclically organized production and present solutions for solving CHSP. The latest papers that introduce solutions for solving DHSP also vary in details. Hindi 2004 limits the problem to a single hoist. Jegou 2006 does not consider workstation groups. However, all papers propose some non-exact methods - heuristics. This is understandable, because Lei 1989 proved that HSP is NP-Hard type of decisive problems and calculation effort is too big for the real-time calculations in DHSP.

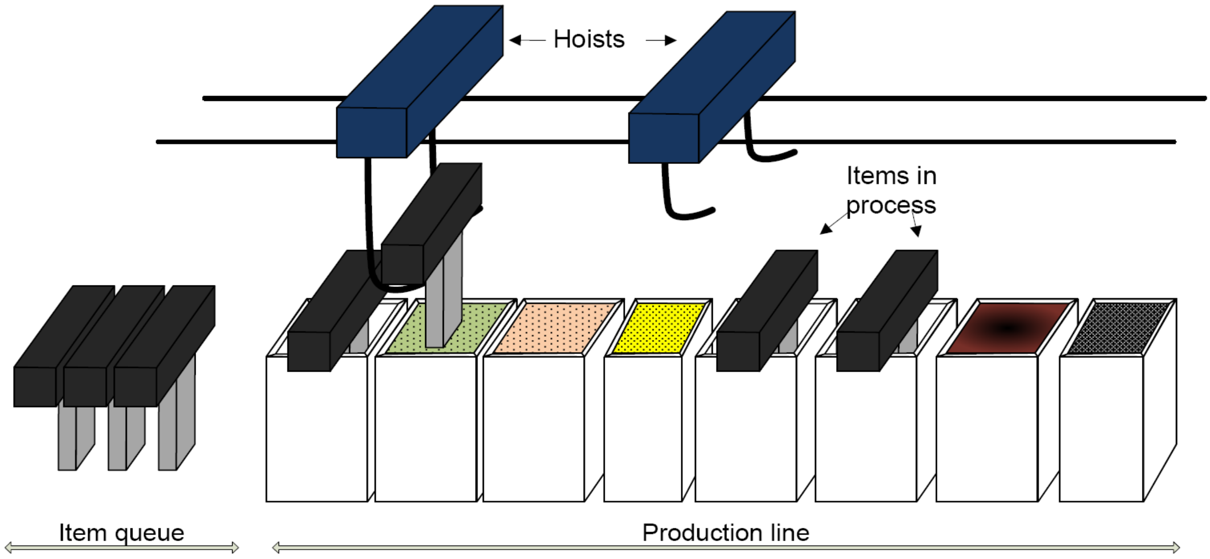

Fig. I The Electroplating Line Overview

The Fig.I presents an overview of typical electroplating lines. There are two hoists, three products waiting to be processed, and four items in-process. The hoist on the left is picking the item from the second workstation in line. There are eight workstations in total.

In the cyclic production, an item must be available in a loading station every given interval of time. You cannot immediately change the item type. Due to the cyclic production, a line gradually is loaded with items and after a few cycles the line is full and produces one item each cycle. If we want to change the produced item type, the line must be unloaded - no item is introduced at the loading station and after the same number of cycles the line is empty, and we can start producing new item type. If we are interested in producing many item types together, we have to accept such a costly line loading unloading phase. We may also switch the production to the order driven.

The DHSP occurs when the production is performed as described above, but new to-be-produced items availability is not known in advance. We need to maintain a schedule, which includes currently processed items and newly ordered items. Since the items are not known in advance, schedule cannot be prepared earlier. Scheduling is parallel to production. Usually, it is assumed that production continues with the schedule, which was used before a new order was specified as long as new schedule is created. This organization of production is convenient for middle term planning in a company. It may allow achieving the higher utilization rate of resources as well as higher performance of a production line. Advantages of order driven production come with cost. A real-time scheduling system is required. A significant computation power may be required by such a system in order to provide schedules on time and avoid defective items otherwise.

The Scheduling System

The goal of a scheduling system in an order driven electroplating production line is to create constantly a valid schedule for all items ordered through all production time. Creation of schedules is a real-time problem because we cannot halt the production overall at the time when a new order is specified. If we stopped, the items that are currently processed would break the quality constraints and be defective. Therefore, we propose the scheduling system (Fig. II), which bases on the old schedule until it finds a feasible update.

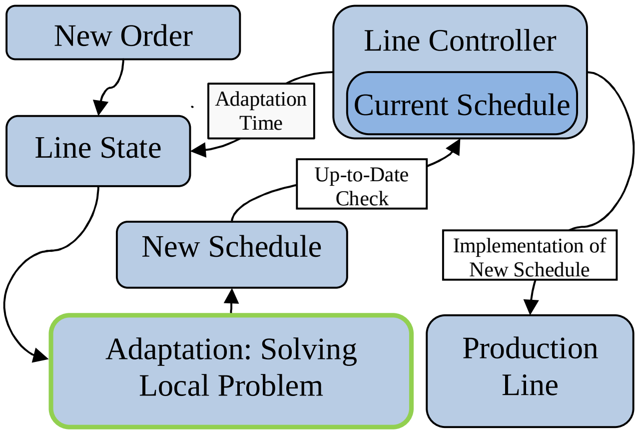

Fig. II The Scheduling System

When a new order is stated at a moment τ, the line controller decides about the adaptation moment t – time in the future, from which the schedule changes will be carried on. $t = \tau + \Delta,\ \Delta > 0$. Line controller provides also information about the line state at the adaptation time according to current schedule. Line state is a parameter to the scheduling algorithm, which generates a feasible schedule. Creating of schedule lasts σ of the cyclogram unfolding computation time and produces an updated schedule. Besides other constraints, the updated schedule is the same as the current schedule up to time t. At this point, the new schedule is checked against its applicability $(\sigma < \Delta)$. In case the updated schedule is infeasible, the scheduling system figures out a new adaptation time and a new Local Problem is solved. When the adaptation is successful, the line controller implements the new schedule to the production line. The line controller converts the schedule to some automaton language e.g. STEP7.

The presented scheduling system does not guarantee that a valid solution is found, unless the adaptation time t can be eventually estimated correctly $(\sigma < \Delta)$ after a number of iterations. Therefore, the system quality depends on both adaptation time estimating algorithm and scheduling algorithm.

There are no assumptions made on the new orders. It can be a single item of a certain type or a sequence of multiple items of many types.

The Intelligent Scenario Selection In Dynamic Hoist Scheduling Problem

In order to describe the scheduling system in more detail a notion of the Local Problem and the Intelligent Scenario Selection scheduling method is going to be described first.

The Local Problem

The problem of finding the schedule for specified line, queue of items, and line state can be formulated as optimization problem. The problem parameters are: the properties of the line (workstations locations, sizes, workstation counts), the hoists count, the hoists speed. Other parameters concern produced product types: the sequence of workstation for each product type, minimum and maximum times of processing in visited workstations. The parameters describing the line state in adaptation time: hoist positions, processing stages of products processed, a queue of to be processed items are also considered. A result schedule can be described by decisive variables of this problem: routes of hoists, assignments of pickup operations to hoists, assignments of put down operations to workstations in groups, times of pickup and put down operations. The optimization criterion is minimizing the time of a last put down operation. Such criterion maximizes the performance of the production line for a given product order. Similarly, to other scheduling problems, DHSP has many constraints. Schedule must fulfil process requirements – processing sequence must be correct, processing times must not be shorter than specified minimum times and longer than specified maximum times. A hoist can carry only one item at the time, a workstation can process only one item at the time. Hoists cannot move outside their physical capabilities, and they cannot collide. It is usually assumed that no collision is present when at any time the positions of hoists are not close enough. We provide only relevant symbols that are necessary to describe the cyclogram unfolding method.

N - number of product types, $n \in \hat{N} = \{1,...,N\}$. $Z = \{z(1),...,z(K)\} = \{s(1),...,s(KC)\} \cup \{\delta(1),...,\delta(KQ)\}$ - queue of items to be produced, $s(k) \in \hat{N}$ - in process products types upon rescheduling, $\delta(k) \in \hat{N}$ - products in queue. $z(k) \in \hat{N}$, KQ - number of items in queue. KC - number of items processed on line during rescheduling. $K = KC + KQ$. L - number of workstation groups present on line, g_l - number of workstations of type l, $l \in \hat{L} = \{1,...,L\}$. $O_n = \{w_1,...,w_{I(n)}\}$ - sequence of workstation group types necessary to manufacture product of type n. $w_{i(n)} \in \hat{L}$ - workstation group type of i -th product processing stage of type n. $i(n) \in \hat{I}(n) = \{1,...,I(n)\}$ where $I(n)$ is number of steps in of product type n processing sequence. H - number of hoists present on line, $h \in \hat{H} = \{1,...,H\}$. $\mu_{n,l}$, $\eta_{n,l}$ - minimum and maximum amount of time needed for product of type n should be processed in l -th workstation group.

Parameters are derived from physical properties of production line and line state in moment of rescheduling. A schedule can be created basing on values of following optimization problem decisive variables: routes of hoists have to be found: $U(h,\lambda)$, $\lambda \in \{1,...,\Upsilon\}$ - the routes of hoists, represented as position of hoist h in production time λ. Υ - time of production cease. The variables $\bar{t}_{k,i(z(k))}$, $\underline{t}_{k,i(z(k))}$ - the moments of picks and puts of products in i - th step workstation have to be established as well. The activities of picking up and putting down a product from/to workstation have to be assigned to the specific hoists $\overline{h}_{k,i(z(k))} \in \widehat{H}$, $\underline{h}_{k,i(z(k))} \in \widehat{H}$. Additionally, for given step of product processing, a specific station from group has to be chosen: $a_{k,i(z(k))}$. Note that not all decisive variables are required to be found. Pickup and putdowns of items already in process are already calculated up to the time of adaptation $t = \tau + \Delta$. Initial hoists positions are in fact hoists positions in adaptation time $U(h,0) = U(h,t)$.

An optimization criterion used to reach a maximum manufacturing performance is $Q = \max\left(\underline{t}_{k,I(z(k))}\right)$. The optimal solution is found when criterion reaches a minimum. The criterion is minimized when last item in sequence leaves production line as soon it is possible.

Finding any feasible solution is very hard because some decisive variables value permutations are to be analyzed. This makes the full space search possible only for very small problems and cannot be scaled to non-trivial problem sizes. Heuristic method of the Intelligent Scenario Selection is used to solve the Local Problem instances. The scheduling method uses few basic operations: segmenting, cyclogram unfolding, and shifting.

Let $S = \left\{Z, P, \bigcup_{t=0,h=1}^{t=\Upsilon,h=H} \{U(h,t)\}, \bigcup_{k=0,i(z(k))=0}^{k=K,i(z(k))=I(z(k))} \left\{\overline{t}_{k,i(z(k))}\right\}, \bigcup_{k=0,i(z(k))=0}^{k=K,i(z(k))=I(z(k))} \left\{\underline{t}_{k,i(z(k))}\right\}, \bigcup_{k=0,i(z(k))=0}^{k=K,i(z(k))=I(z(k))} \left\{\overline{h}_{k,i(z(k))}\right\}, \bigcup_{k=0,i(z(k))=0}^{k=K,i(z(k))=I(z(k))} \left\{\underline{h}_{k,i(z(k))}\right\}\right\}$ be the schedule for queue Z. Let us mark $\Psi(S,\tau)$ as the shifting operation. It shifts the schedule S by τ seconds. The operation changes the schedule variables as defined: $\overline{t}_{k,i(z(k))} = \overline{t}_{k,i(z(k))} + \tau$, $\underline{t}_{k,i(z(k))} = \underline{t}_{k,i(z(k))} + \tau$, $U(h,t) = \begin{cases} U(h,t-\tau), t-\tau > 0 \\ U(h,0), t-\tau \le 0 \end{cases}$.

Segmenting operation $\Xi(Z)$, divides products from orders to segments of items of the same type, such that: $\{\delta_1,...,\delta_{KQ}\} = \{s_{1,1},...,s_{1,S_1}\} \cup ... \cup \{s_{SEG,1},...,s_{SEG,S_{SEG}}\}$ where $\sum_{i=1,...,SEG} S_i = KQ$ and $s_{i,j} = s_{i,m} \forall j, m = 1,...,S_i, i = 1,...,SEG$. Segmenting divides the new order to sub-sequences of items of the same time. I.e. Z=$\{1,1,3,1,2,2,3,3\}$ means that following segments are created: $\{1,1\}$, $\{3\}$, $\{1\}$, $\{2,2\}$, $\{3,3\}$.

It is important to introduce the idea of cyclogram, which is a cornerstone of cyclic production. Cyclogram is a specific type of schedule used in electroplating lines. In [3] authors claim that the cyclic production causes schedule to be periodic. Cyclogram is a schedule that represents one period, a cycle of periodic schedule. Cyclogram contains a constant number of hoist operations. Repeated over and over again, it allows production of any number of items. Cyclograms are built this way, that one item is introduced and completed during one cycle. Main features of cyclogram are a capacity and a cycle time. The cycle time is a length of cyclogram in a time domain. The capacity is a number of cycles needed to be performed in order to load fully a production line and produce a first item. Afterwards, a line produces one new item each cycle. There can be many feasible cyclograms for one item type. Let us assume that there is a number cyclograms for each item type $n \in \{1,...,N\}$ referred as $V_1,...,V_N$, $v_n \in \{1,...,V_n\}$. $T(v_n)$– cycle-time of cyclogram v_n, $G(v_n)$ – capacity of cyclogram v_n. Cyclogram can be defined similarly as schedules in DHSP – routes, times of operations and assignment of operations to hoists. $U(v_n,h,\Lambda)$, $\Lambda \in \{0,...,T(v_n)\}$ – routes of hoists, represented as position of the hoist h in production time Λ for cyclogram v_n. $\bar{t}(v_n,i(n)),\underline{t}(v_n,i(n))$– moments of pick and put of i-th stage of product n-th in cyclogram v_n, $\bar{h}(v_n,i(n)),\underline{h}(v_n,i(n))$– indexes of hoists which, perform a i-th pick and put of product n-th in cyclogram v_n.

Let us mark $C(x,v_n)$ as the cyclogram unfold operation, for x items of type n using cyclogram v_n. As a result of $C(x,v_n)$ we get a schedule based on cyclogram v_n, which produces x items. Cyclogram operations are repeated a number of times until x items leave the line. A schedule length $|C(x,v_n)| = (\mathrm{G}(v_n)+x)\cdot T(v_n) - \mathrm{T}_{\mathrm{load}}[v_n] + \mathrm{T}_{\mathrm{unload}}[v_n]$. $C(x,v_n)$ creates a solution for Z=$\{z_1,...,z_x\}$, where $z_k = n$ by assignment: $U(h,t)=U(v_n,h,t \bmod T(v_n))$, $t \in \{0,...,|C(x,v_n)|\}$, $\bar{t}_{k,i(n)} = \bar{t}(v_n,i(n)) + (k+\theta(i(n)))\cdot T(v_n)$, $\theta(i(n)) = \sum_{a=1}^{a=i(n)} \begin{cases} 1-\bar{t}(v_n,a) < \underline{t}(v_n,a) \\ 0-\bar{t}(v_n,a) > \underline{t}(v_n,a) \end{cases}$, $\underline{t}_{k,i(n)} = \underline{t}(v_n,i(n)) + (k+\theta(i(n)))\cdot T(v_n)$, where θ is the counter of passing product between two cycles. $\bar{h}_{k,i(z(k))} = \bar{h}(v_n,i(n))$, $\bar{h}_{k,i(z(k))} = \bar{h}(v_n,i(n))$.

INTELLIGENT SCENARIO SELECTION METHOD

The basic idea of the scheduling method is to enhance the schedules, which are created in cyclic production. In cyclic production, in order to change the produced item type, the line had to be emptied and then the new production was started. New production was performed using some cyclogram prepared for newly produced item type. There are two fallacies in such organization. We wait until the line is emptied in order to change the produced item type, and that we do not allow parallel production of two item types. The Intelligent Scenario Selection Method improves the scheduling in those two areas. Any new order can be divided to same type segments of items by segmenting operation $\Xi(Z)$. Each segment items can be produced using one of the v_n cyclograms by using the unfolding algorithm $C(x,v_n)$. Unfolding algorithm creates schedule parts, which need to be adapted to each other in order to fulfil constraints and create a feasible solution. Adaptation is done by shifting procedure and routing algorithm presented in Kujawski 2007.

One of the important aspects of production of the heuristic is that it is adapting its knowledge about creating high performance schedules by using machine learning methods. When a new segment is about to be adapted, one of the decisions that is being done is choosing a cyclogram from prepared set for unfolding process. A classifier does this decision. For each product type $n \in \{1,...,N\}$, we construct a classifier $\Im_n$. It selects a scenario used for unfolding and shifting operations during scheduling. A number of problem features are calculated to reflect the state of a line and the characteristic of the product order. Features include utilization of workstations, utilization of hoists, a number of products in segment, a length of unfolded segment, a resource collision count. Each $\Im_n$ classifies the current line state to the production scenario, which is going to be used in further processing. The scenario is, in fact, a cyclogram used for unfolding, so the classifier decides which cyclogram is going to maximize the performance.

The algorithm pseudo-code:

```
Scenario Selection
 1) Calculate $\Xi$ for queue Z
 2) Result = CURRENT_SCHEDULE
 3) for x = 1 to SEG in
      $\{s_{1,1},...,s_{1,S_1}\}\cup...\cup\{s_{SEG,1},...,s_{SEG,S_{SEG}}\}$
 3a)  Select $v_{s_{x,1}}$ using $\Im_{s_{x,1}}$ and Result.
      If classification is
      inconclusive select arbitrary
      $v_{s_{x,1}}$ from $\{1,...,V_{s_{x,1}}\}$
 3b)  Calculate S(x)=$C(S_x,v_{s_{x,1}})$.
 3c)  Find smallest $\tau$ that there
      exists collision-less routing   for Result$\cup\Psi(S(x),\tau)$.
 3d)  Result=Result$\cup\Psi(S(x),\tau)$
```

The Intelligent Scenario Selection Method allows improving the scheduling system. We can see that every iteration in point 3), of the algorithm, updates the result. We can consider that each segment is, in fact, separate order. If we assume so, time of the calculation is independent of the size of the order. That increases the responsiveness of the real-time system. We are updating the schedule more times but each update is calculated much faster than calculating the whole order. Each update pushes the time horizon of the outdated schedule further into future. Additional advantage of such a solution is that it is easier to estimate the calculation time of single segment scheduling calculation. It makes the estimation of the closest adaptation time easier. Another advantage is that when the τ is greater than the time of the last activity from the previous segment then the schedule is automatically feasible, providing that there is some time for hoists to drive to their initial position. The schedule in such a case is always valid because after all the items from the line are unloaded then the line is empty, so our segment is the new production starting and no resource collisions can occur. Another advantage of cyclogram unfolding is that process of finding τ is easier, because most of the problem constraints are always satisfied. Schedule for segment, generated by unfolding procedure has a number of constraints satisfied: sequence of picking the item out of bath and putting the item to bath, times of processing in baths are satisfied, etc... Shifting of the whole segment can break only some of the constraints, while others are still satisfied. The cost of calculating each segment separately is a loss of generality that could lead to find locally optimal schedules. In Fig. III the more detailed description of the scheduling system is presented. The schedule is updated for each segment separately.

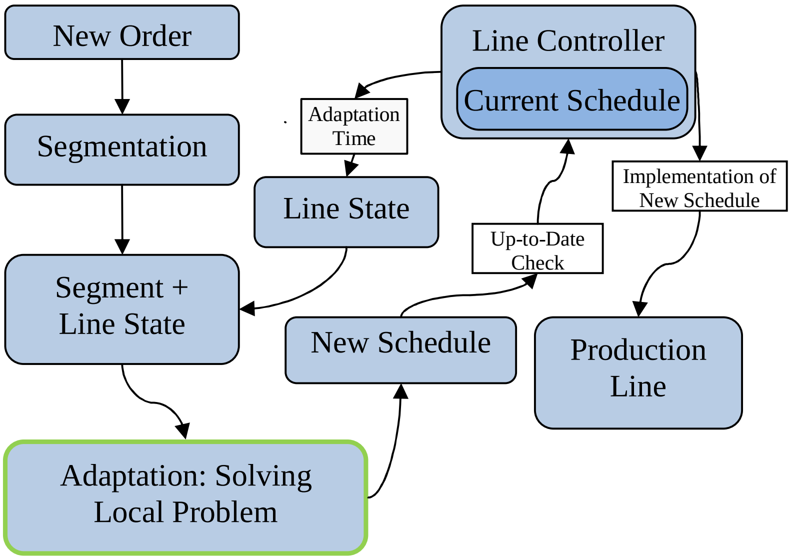

Fig. III. The Detailed View of The Scheduling System

ADAPTATION TIME ESTIMATION

The adaptation time t is estimated during run time. In fact, the adaptation time is calculated from $t = \tau + \Delta,\ \Delta > 0$ so the Δ is estimated. At the beginning of production Δ is empirically set to value equal to expected value of scheduling time $E(\sigma)$. At each estimation, firstly it is checked whether $\tau + \Delta$ is sooner than the time of the last item from the previous segment introduction to line time. If it is then the Δ is set to time of introduction of the last item from the previous segment. This causes that items are processed in same sequence as ordered. In case last introduction time is sooner than $\tau + \Delta$ then Δ is set to $E(\sigma)$. After each update the expected value of σ is updated. If at any time the up-to-date check fails the next estimation of Δ is doubled.

EXPERIMENTAL CASE ANALYSIS

The production line in Wrocław, Poland is a typical small electroplating line. It is used in production of metal furniture elements. The line consists of 16 workstation groups composed into one column. One group has 3 workstations, and the other groups are, in fact, a singular workstation. Two hoists are available. Hoists maximum speed is 0.7 m/s. Hoist collision zone is 2 meters - two hoists centers cannot be closer than 2 meters or the collision occurs.

We will consider two item types that are frequently ordered together. Let us mark the technology process of technical chroming as a "type A" and process of nickeling as a "type B". Table I and Table II define the processes sequence and quality constraints for process, A and B respectfully. All pick up and put down times are 8 seconds.

Table I. The Technical Chroming Process Definition

No. Step	Step name	Min. time	Max. time	Group No.
1	Loading station	-	-	1
2	Chem. degreasing	300	420	10
3	Dripping	30	90	11
4	Electrochemical. degreasing .	60	180	12
5	Warm rinse	30	90	13
6	Rinse I	20	80	14
7	Cascade rinse I	20	80	15
8	Cascade rinse II	20	80	16
9	Anode etching	150	210	9
10	Chroming	1200	1300	8(3)
11	Salvaging rinse	20	80	7
12	Rinse II	20	80	6
13	Rinse with chrome reduction	20	80	5
14	Rinse III	20	80	4
15	Timed rinse.	20	80	3
16	Blow in bath	30	90	2
17	Unloading station	-	-	1

The step number 10, "Chroming" lasts extensively longer than other steps. For that reason, the chroming bath is multiplied to three workstations. All three workstations can be used in the processing.

Table II The Nickeling Process Definition

No. Step	Step name	Min. time	Max. time	Group No.
1	Loading station	-	-	1
2	Timed rinse.	300	400	3
3	Salvaging rinse I	30	90	5
4	Salvaging rinse II	30	90	7
5	Degreasing I	60	180	10
6	Electrochemical. degreasing.	20	80	12
7	Rinse I	20	80	14
8	Cascade rinse I	20	80	16
9	Cascade rinse II	20	80	15
10	Warm rinse	150	210	13
11	Nickeling	200	400	11
12	Degreasing II	20	80	10
13	Rinse II	20	80	6
14	Rinse III	20	80	4
15	Blow in bath	30	90	2
16	Unloading station	-	-	1

To calculate the time required to move a hoist from the workstation, i to j the distance between workstation centers is divided by the maximum hoist speed and rounded up to whole seconds. In order to transport an item from a group hoist need to go to the group center, pick the item up (8 seconds), move to the destination bath and put the item down (8 seconds). In order to create as usable schedules as it is possible the movement is approximated by widely used 3-5-7 degree polynomial.

Table III: The Workstation Centers Positions

Group No.	1	2	3	4	5	6
Position (m)	0.3	0.94	1.56	2.18	2.8	3.43
Group No.	7	8(1)	8(2)	8(3)	9	10
Position (m)	4.08	4.85	5.75	6.65	7.46	8.14
Group No.	11	12	13	14	15	16
Position (m)	8.76	9.42	10.08	10.7	11.27	11.79

Production

The organization on the line is currently cyclical. The owners want to switch to the dynamic production, because they need to produce many item types together. As a data to research, we will use some historical runs and typical order ratios. For historical runs the schedules are available but without the real-time aspect. We can do it offline as we know the whole order before production is started. Such test is also valuable because it allows benchmarking the performance of created schedules.

Table IV: Queues to Analyze

Queue	Items sequence
Queue 1	AABBAABB
Queue 2	7xA;5xB;5xA
Queue 3	20xA;15xB
Queue 4	5xA;5xB;5xA;5xB;5xA
Queue 5	12xB;4xA;6xB;2xA
Queue 6	8xB;8xA

More general production requirements, which we have on the Wrocław line, is that statistically orders are composed like 1:2, so for each item of type Two items of the type B are ordered. We are going to analyze the ratio 2:3 and 1:1.87. For each ratio, the orders are generated. We are going to simulate the real-time production. We generate the queue of 40 items for each proportion (16xA:24xB and 18xA:22xB) with random sequence. We calculate the schedule for three cases: whole order is known at the beginning, the next segment is known in 100 seconds before the shortest available adaptation time, and the next segment is known in the time of the shortest adaptation time. The shortest adaptation time is the time where the last item of current schedule is loaded to line, adaptation time must be greater to keep the correct sequence of item introduction. The result is averaged from 100 generated instances.

Algorithm work example

We have prepared a V_1,V_2 cyclograms using [13]. Initially there were 56 and 30 different cyclograms. During preparation of patterns for classifier learning, it happened that only 3 cyclograms of each type are significant. We have continued with cyclograms for item type A: A1 - length 415 sec., capacity 7; A2 - length 424 sec., capacity 6; A3 - length 432 sec., capacity 6; and for type B: B1 - length 240 sec., capacity 6; B2 - length 257 sec., capacity 5; B3 - length 532 sec., capacity 3. We have trained two neural networks, each for the different type of item. For the Wrocław line 39 features are extracted from line state, and queue, so neural networks have 39 inputs, 40 neurons in hidden layers and 3 outputs.

Let us analyze the Scenario Selection method for Queue 1. Let us assume that we are starting the production, so the line and schedule are empty. Ξ operation gives us four segments {AA}, {BB}, {AA}, {BB}. In the first step $\Im_1$ returns A3, S(x) is a production schedule of two items of type A, its length is 2887 seconds. τ=0 as the current schedule is empty, so there cannot be any collisions. We store the result and proceed to a second iteration. $\Im_2$ returns B3, We calculate S(x), its length is 1720 seconds. We analyze $\tau >$ 440 seconds and find τ=893 constructs a feasible schedule for AABB with a length of 2887 seconds. In a third iteration $\Im_1$ returns A2, We calculate S(x) by cyclogram unfolding, its length is 2967 seconds. We analyze $\tau >$ 1433 seconds and find τ=2464. We store the result (5431 sec.) and proceed to the last iteration. $\Im_2$ returns again B3, We calculate S(x), its length is 1720 seconds. We analyze $\tau >$ 2896 seconds and find τ=3424. The schedule for Queue 1 is created and its length is 1:30:31 (5431 sec.). Scheduling takes around 3 seconds. All calculations are performed on Intel Q9300 2.5GHz processor.

RESULTS

The tables summarize the results achieved by using the Intelligent Scenario Selection Method. In order to show how much time can be gained using calculated schedule, we compare their length to a base solution length. To calculate the base solution length, we sum up all the products minimum times, pick up times, putdown times, and transition times required to produce items from the given queue. The utilization ratio is a length of base solution divided by a length of schedule.

Literature Benchmarks

In order to compare work results of proposed hoist scheduling method with results of methods presented in literature, a series of tests on benchmark problems were analyzed. We present the comparison of results from Jegou 2006, Paul 2007 and Lamothe 1996. Both Jegou 2006 and Lamothe 1996 use the simplified linear movement model, where the infinite hoist acceleration and deceleration are assumed. For these tests, we use also the linear movement model.

Table V. Comparison to literature benchmark problems

Type of benchmark	Result from literature	Results of The Intelligent Scenario Selection
Jegou 2006	62 products in 8:00:00 hours	108 products in 8:00:00 hours 62 products in 4:44:00 hours
Paul 2007	27:56:27±2:37:47	24:14:45±1:28:53
Lamothe 1996 (sequence 1,2,3,…,13)	00:04:11 hours	00:03:29 hours
Lamothe 1996 (sequence 1,5,6,…,1)	00:04:55 hours	00:04:02 hours

In Jegou 2006 a local scheduling problem of 3 types of products and 2 hoists is given as the example. A goal was to create a schedule for 8 hours of work and see how many products will be finished in this time.

In Paul 2007 benchmark deals with 4 types of products on single hoist line. A mean of 10 tests with random sequence of 40 products with given quantities of each product is presented as the result. Presented results are for default parameters presented in Paul 2007.

Lamothe 1996 presents a classic benchmark problem known in HSP, introduced by Phillips 1976. Only one type of product is processed. Lamothe 1996 uses this process definition for scheduling with two hoists and queue of items. Since only one type of product is scheduled and new item is always accessible in input buffer, although no cyclic assumption, authors claim that schedule should have cyclic nature in this case.

The Wrocław Production Line

Table VI Queues

Queue	Historical Schedule Length	Scenario Selection	Improvement	Utilization Ratio
Queue 1	2:41:36	1:30:31	43.9%	2.44
Queue 2	3:28:48	3:11:32	8.2%	2.81
Queue 3	4:34:51	4:26:05	3.1%	3.97
Queue 4	5:11:12	4:45:21	8.3%	2.59
Queue 5	4:02:48	3:41:36	8.7%	2.47
Queue 6	2:33:24	2:22:14	7.2%	3.12

In all cases the used method as better than the provided schedules.

Table VII Ratio 16:24; The Scenario Selection

Test Type	Schedule Length	Utilization Ratio	Scheduling Length
Entire Order Known	9:42:56± 1:15:03	2.05±0.20	0:14:20± 0:05:36
100 Seconds Before	11:09:50±0:18:04	1.76±0.03	0:00:27 per segment
Adaptation Time	11:08:29±0:12:18	1.76±0.03	0:00:27 per segment

Table VIII Ratio 18:22; The Scenario Selection

Test Type	Avg. Schedule Length	Utilization Ratio	Scheduling Length
Entire Order Known	9:33:57± 1:00:36	2.00±0.18	0:13:57± 0:05:54
100 Seconds Before	10:44:49± 0:12:30	1.77±0.03	0:00:26 per segment
Adaptation Time	10:44:00± 0:12:23	1.77±0.03	0:00:26 per segment

Table IX Ratio 16:24; The Cyclogram Unfolding

Test Type	Schedule Length	Utilization Ratio	Scheduling Length
Entire Order Known	10:47:03± 0:39:12	1.83±0.11	0:20:08± 0:12:37
100 Seconds Before	13:16:11± 0:18:13	1.48±0.03	0:01:00 per segment
Adaptation Time	13:19:27± 0:18:53	1.47±0.03	0:01:00 per segment

Table X Ratio 18:22; The Cyclogram Unfolding

Test Type	Avg. Schedule Length	Utilization Ratio	Scheduling Length
Entire Order Known	10:51:04± 0:37:40	1.76±0.10	0:17:11± 0:11:11
100 Seconds Before	12:58:18± 0:17:45	1.47±0.03	0:00:59 per segment
Adaptation Time	12:59:13± 0:17:11	1.47±0.03	0:00:59 per segment

In all tests in real-time produced schedules were up-to-date and did not require additional iteration of scheduling. We can observe that real-time production has overall worse performance than production performed on prepared schedule. Nevertheless, the relatively high utilization ratio is maintained. The movement model parameters (acceleration, deceleration) were constant for the tests.

The Intelligent Scenario Selection In Dynamic Hoist Scheduling Problem

SUMMARY

The scheduling algorithm is efficient enough for the Wrocław line and possibly for similar sized productions. The presented method is competitive to previously proposed methods from literature. The scheduling system prototype is being tested in real-life electroplating test.

REFERENCES

Hindi, K., Fleszar, K. (2004). A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem, *Computers & Industrial Engineering* 47, 91–101

Jegou, D., Kim, D., Baptiste, P., Lee, K. (2006). A contract net based intelligent agent system for solving the reactive hoist scheduling problem, *Expert Systems with Applications* 30, 156–167

Kujawski, K., Świątek, J. (2009). Using Cyclogram Unfolding in Dynamic Hoist Scheduling Problem, *Proceedings of the International Conference on Systems Engineering ICSE'09, UK, Coventry*, 282– 287

Kujawski, K. (2007). Schedule optimization of certain production line type, Dissertation, Wrocław University of Technology, Poland

Lamothe, J., Thierry, C., Delmas, J. (1996). A multihoist model for the real time hoist scheduling problem, *Proceedings of CESA'96, Lille, France, 9–12 July*, 461–466

Lei, L., Wang, TJ. (1989). A Proof - the Cyclic Hoist Scheduling Problem Is NP-Hard. Working paper 89-0016, Rutgers University

Leung J., Zhang G., (2003). Optimal cyclic scheduling for printed-circuit-board production lines with multiple hoists and general processing sequences, *IEEE Transactions on Robotics and Automation*, 19 (3), 480–484

Liu, J., Jiang, Y., Zhou, Z., (2002). Cyclic scheduling of a single hoist in extended electroplating lines: a comprehensive integer programming solution, *IIETransactions*, 34, 905-914

Mak, R.T., Gupta, S.M., Lam, K., (2002). Modeling of material handling hoist operations in a PCB manufacturing facility, *Journal of Electronics Manufacturing*, 11 (1), 33–50

Paul, H., Bierwirth, C., Kopfer, H. (2007). A heuristic scheduling procedure for multi-item hoist production lines. *International Journal of Production Economics* 105, 54–69

Phillips, L.W., Unger P.S. (1976), Mathematical programming solution of a hoist scheduling problem. *AIIE Transactions* 8 (2), 219–225

Varnier, C., Bachelu, A., Baptiste, P. (1997). Resolution of the cyclic multi-hoists scheduling problem with overlapping partitions, *INFOR*, 35(4), 309–324.

Yan P., Chu C., Che A., Yang N. (2008). Algorithm for optimal cyclic scheduling in a robotic cell with flexible processing times, *IEEE-Proceedings of IEEM'08 Singapore*, 153-157