This chapter deals with automation of machining lines, sometimes called transfer lines, which are serial machining systems dedicated to the production of large series. They are composed of a set of workstations and an automatic handling system. Each workstation carries out one identical set of operations every cycle time. The design of transfer lines is comprised of several steps: product analysis, process planning, line configuration, transport system design, and line implementation. In this chapter, we deal with line configuration. Its design performance is crucial for companies to compete in the market. The main problem at this step is to assign the operations necessary to manufacture a product to different workstations while respecting all constraints (i.e., the line balancing problem). The aim is to minimize the cost of this line while ensuring a desired production rate. After a review of the existing types of automated machining lines, an illustration of a developed methodology for line configuration is given using an industrial case study of a flexible and reconfigurable transfer line.

Manufacturers are increasingly interested in the optimization of their production systems. The objective is to optimize some criteria such as total investment cost, floor area, number of workstations, production rate, etc.

The automatic serial line, often called a transfer line, is a widely used production system in machining environments [35.1–6]. Transfer lines also exist in the assembly industry. Their properties are defined, for example, by Nof et al. [35.7]. In such a line, a repeatable set of operations is executed each cycle. The line is composed of sequentially arranged workstations and a transport system which ensures a constant flow of parts along the workstations. This automatic handling system is generally composed of conveyors fixed on rails that transfer the part from one station to the next with holder robots for part loading and unloading at stations. The transfer machining lines produce large series of identical or similar items.

Automation of a machining line for a given product family (or reconfiguration of an existing line for a new product family) is a significant investment, and requires a long period for its design (often 18 months). Manufacturers have to invest heavily when installing these lines or for their reconfiguration. This investment influences to a large extent the cost of the finished products over the lifetime of the line. Therefore, profitability depends directly on the success of the line design or reconfiguration. Investment cost should be minimized and the configuration obtained should be as efficient as possible. Thus, optimization is a crucial issue at the transfer line design or reconfiguration stage.

The design of transfer lines is comprised of several steps: product analysis, process planning, line config-
In this chapter, we deal with line configuration. Its design performance is crucial for companies to compete in the market.

As a rule, the configuration of a transfer line involves two principle steps:

1. choice of line type
2. logical synthesis of the manufacturing process, which consists of grouping the operations into stations (i.e., line balancing).

In this chapter, we focus on the second step of this procedure, because the decisions made there define the principal characteristics of the line. An error at this time is too costly to rectify. A brief description of this problem is in order.

Automated machining lines are composed of a set of serial workstations. The stations are visited in a given order. The line investment cost depends on the number of stations and equipment of each station. Both are defined via an assignment of operations to workstations. Usually, each task is characterized by: (1) its time, (2) a set of operations which must be assigned before (precedence constraints), (3) a set of operations which must be executed on the same workstation (inclusion constraints), and (4) a set of operations which cannot be executed on the same workstation (exclusion constraints). Of course, in actual industrial problems, various additional specific constraints may have to be taken into account as well. Thus, at the line configuration stage, it is necessary to solve the line balancing problem, which consists of assigning the operations to workstations, minimizing the line investment cost while respecting the objective production rate as well as the aforementioned constraints.

This chapter is organized as follows. In Sect. 35.1, the fundamental assumptions and existing types of automated machining lines are introduced. In Sect. 35.2, some challenges and a general methodology for design and reconfiguration of these lines are explained. In Sect. 35.3, the role and importance of line balancing at the design or reconfiguration stage are presented. Section 35.4 illustrates our approach and models on an industrial case study. Moreover, in this section, a novel and promising exact resolution method is suggested for balancing of machining lines with parallel machines and setup times.

### 35.1 Machining Lines

A machining line is a production system composed of several sequential workstations; each workstation contains various machining equipment. A given set of operations is performed at each station to obtain the final product. The most frequent machining operations are:

- **Drilling**, to fabricate holes in parts
- **Milling** of shapes or removal of material with various milling cutters to form concave or convex grooves, etc.
- **Tapping**, which involves cutting internal screw threads in holes
- **Boring** to enlarge a hole that has already been drilled to precise dimensions.

A combination of these operations is usually needed to manufacture complex parts such as cylinder heads, cylinder blocks; see, for example, Fig. 35.1.

The machining process has numerous specific properties that directly influence the organization of the automated machining lines. Because of the complexity involved, special studies and decision-aid tools are required for competitive design and reconfiguration of these lines.

There are three principal types of automated machining lines for large series, namely, dedicated, flexible, and reconfigurable transfer lines. Each of these has its own characteristics and assumptions, which will be briefly detailed in the following.

#### 35.1.1 Dedicated Transfer Lines

A dedicated transfer line (DTL) is the most economic form of machining systems with a large productivity and profitability if there is enough volume. DTLs are used for the production of a single type of product (or close variants) in large series; a large quantity of identical products is manufactured with the same sequences of operations on stations. The stations are arranged serially. Each station is equipped with multispool heads. Each multispool head executes several operations simultaneously. Depending on the architecture of the line, spindle heads can be activated at each station in parallel.
or in sequence. An example of such a multispindle head is shown in Fig. 35.2.

When customer demand is significant and stable for a number of years, this type of machining line is the most profitable solution.

One of the design principles for the dedicated lines is the reduction of the cycle time and minimizing the amount of equipment (machines, spindle heads, tools, etc.) which, as mentioned above, has a direct influence on the reduction of the unit production cost.

Principal advantages of dedicated transfer lines are:

- High precision: these lines are designed to maximize the accuracy when machining of the part.
- Quality: there are no tool changes; therefore, once quality is established, it is stable.
- Mass production: the annual production can be in the millions.

Disadvantages of these lines are:

- The dedicated transfer lines demand large investments and must have a long lifetime to be profitable, the ramp up of the production is relatively long (2–4 weeks).
- Taking into account specific aspects of the product during the line design stage is possible, but once the line is defined it is very difficult to modify (line reconfiguration is costly).
- Breakdowns are a crucial problem; when a breakdown occurs in a single station, the entire line is stopped (in addition, if the corresponding operation did not end on time because of this breakdown, the product is automatically defective).

Note that for these lines the criterion to be optimized is easy to identify and calculate: minimizing the investment cost. Moreover, the interest in studying DTLs lies in the fact that their structure represents a basic form of organization for other machining systems. Indeed, all the problems that appear during the optimization of a dedicated transfer line are present in the design of other automated machining lines.

### 35.1.2 Flexible Transfer Lines

The flexible transfer line (FTL) is a special case of a flexible manufacturing system (FMS). The flexibility of a FMS is ensured thanks to the utilization of computer numerical control (CNC) machines (machining centers), automated transport, and warehousing systems with sophisticated control software [35.8]. An exam-
ple of a flexible machining center with the devices for change of tools is shown in Fig. 35.3.

FMS can produce several types of products, belonging to a broad family. By family we mean products having comparable dimensions and similar geometric characteristics, as well as the same tolerances. These related products can be manufactured by the same equipment. Software takes care of possible changes by reprogramming the machining or rescheduling the products to be manufactured.

There are three basic types of FMS [35.7]:

- **Flexible lines**: These consist generally of sequentially arranged workstations with programmable CNC machines (machining centers) and are used especially for products with several product variations and a short lifetime for each variation.

- **Flexible cells**: Such a system is composed of disconnected programmable cells, where each cell consists of one or several machining centers and carries out processes that comprise complete or almost complete tasks. The number of distinct parts in such a cell is often restricted, from eight to ten. This is due to a limited capacity of the cells.

- **Flexible systems**: These are composed of linked flexible cells. There are two types of linkage: (1) with a rigid sequence of linking (cells are connected in a given invariable order); (2) with linkages that can be adapted to any particular production and/or assembly process.

The main objective of flexible transfer lines (FTLs) is to be able to produce several variations of the same product in large series. These lines assure a quick passage from one variation to another. FTLs are also able to change production volumes, if necessary, within a given range. The ramp up of production is short (1–2 days). However, they present a certain number of drawbacks:

1. These systems are very expensive mainly because they are composed of CNC machines. These machines are designed for a forecasted family of parts and produced without optimal process planning for each actual part. At the line design stage, the machining specifications are not accurately known. Therefore, the designer tends to insert more functions than necessary. Obviously, this increases the cost.

2. The development of software for the line control system is also very expensive because this flexible equipment requires sophisticated rules of management for each machine as well as for the entire line.

3. Contrary to dedicated machines, which contain multisindle heads with fixed tools, CNC machines use single spindle heads with frequent tool changes. Therefore, it is often difficult to maintain the level of precision in machining operations equal to that of dedicated lines.

4. Owing to rapid technological advances, these sophisticated and costly machines are quickly subject to obsolescence.

5. Because of their complexity, these flexible transfer lines tend to be less reliable.

### 35.1.3 Reconfigurable Transfer Lines

The concept of reconfigurable manufacturing systems (RMS) was introduced in Koren et al. [35.9]. The authors highlight the industry’s new requirements for machining systems given the increasingly shorter product runs and the need for more customization. From the beginning, an RMS is designed to be able to make...
changes in its physical configuration to answer market
fluctuations in both volume and type of product.

For RMS, and especially for reconfigurable transfer
lines (RTL), the principal characteristics are:

- Modularity: in a reconfigurable manufacturing sys-
tem, all the major components are modular (system,
software, control, machines, and process). Selection
of basic modules and the way they can be connected
provide systems that can be easily integrated, diag-
nosed, customized, and converted.

- Integrability: to aid in designing reconfigurable
systems, a set of system configurations and their in-
tegration rules must be established. Initially, such
rules were developed for configurable comput-
ing [35.10]. In the machining domain, these rules
should allow designers to relate clusters of part fea-
tures and their corresponding machining operations
to workstations and machine modules, thereby en-
abling product–process integration.

- Customization: this characteristic distinguishes
RMS from FMS and DTL, and can reduce system
and machine costs. This type of system provides
customized flexibility for a particular part family,
and is open ended.

- Convertibility: rapid changeover between members
of the existing part family and quick system adapt-
ability for future products.

35.2 Machining Line Design

We will now consider the preliminary design of
transfer lines: corresponding challenges and general
methodology.

35.2.1 Challenges

Usually for this type of project, the procedure is as
follows: a company (client) contacts the transfer line
manufacturer. The client gives the parts properties (part
plans, characteristics, etc.) and the required output
(production rate). Then comes the critical phase: the
manufacturer should quickly offer a complete prelimi-
nary design solution for the corresponding line in terms
of line architecture, number of machines, etc., and an
approximate line cost. The acceptance of this solution
by the client, and consequently the continuation of the
negotiation and further development of the project, de-
pends on the quality of this early solution. The temporal

Fig. 35.4 Trade-off between production rate and frequency of mar-
ket changes

- Diagnosability: detects machine failure and identi-
fies causes of unacceptable part quality.

RTLs are usually conceived when there is lit-
tle or no knowledge of future production volume
or product changes. In this sense, they can be
viewed as a compromise between the DTL and FTL
(see Fig. 35.4). On the other hand, as they allow
hardware reconfiguration in addition to software re-
configuration of FTL, some authors judge them more
flexible.
solution at the lowest possible cost within a very short time period.

In addition, after the preliminary design, almost always the product to be manufactured undergoes some modifications during the stage of detailed design of the line. The line manufacturer must continuously take into account these modifications. Furthermore, modifications of the design solution are difficult and time consuming. Therefore, decision-aid models are eminently useful for the preliminary design and to take into account the modifications during the detailed design. We will now present the methodology as applied to one such decision-making process.

### 35.2.2 General Methodology

Independent of the type of transfer line considered (dedicated, flexible or reconfigurable), its design demands an overall approach requiring the resolution of several interconnected problems [35.2]. Ideally, decisions relating to all these problems must be considered simultaneously. However, the total problem is very complex. Therefore, it is necessary to decompose this problem into several subproblems, each engendering less complex decisions [35.3].

Note that only the preliminary design stage is considered in this chapter, i.e., when all principal decisions are made concerning line architecture and its elements. Usually, this is followed by a detailed design (specifications for mechanical elements, tools, spindle heads, etc.), which is outside the scope of this chapter (see [35.4, 5] for a presentation).

The following general steps can summarize the preliminary design process. Note that the importance of each step depends on the type of transfer line considered. Some steps can be omitted.

- **Product analysis**: this gives a complete description of the operations that have to be executed for the future products.
- **Process planning**: covers the selection of processes required to transform raw parts into finished products. Here, technological constraints are defined. For instance, during process planning, partial order between operations, inclusion and exclusion constraints are established. This requires an accurate understanding of the functional specifications for the products and technological conditions for the operations.
- **Configuration design and balancing problem**: selection of the type of machining line and the resolution of the balancing problem, i.e., the allocation of operations to workstations in order to obtain the necessary production rate meeting demand while achieving the quality required. It is imperative to consider here all the constraints, particularly those of precedence.
- **Dynamic flow analysis and transport system design**: simulation is used to study the flow of products taking into account random events as well as variability in production. The objective is to analyze the dynamic flows and choose the material handling system as well as optimize the facilities layout, i.e., placement of machines. The decisions must be coherent with those defined at the previous steps.
- **Detailed design and implementation of the line.**

In addition, for flexible lines, a scheduling step also has to be considered [35.11]. After the implementation, if product and/or volume change, a similar analysis...
should be performed for the optimal reconfiguration of the transfer line (note that this is rarely considered for dedicated lines of mass production; more precisely, a reconfiguration of a dedicated line deals with specific engineering approaches). As illustrated in Fig. 35.6, these steps are executed sequentially. Of course, the designer can return to the previous steps as often as necessary (i.e., the decision-making process is iterative).

Such a methodology was already implemented in a decision-aid software tool for the preliminary design of dedicated transfer lines [35.12]. The developed software includes a database of parameterized features for product analysis. The product analysis provides a set of features which will be used at the process planning step. The process planning generates several process plans with the best one chosen for each feature. A set of operations and constraints are obtained. Then, a type of transfer line is selected by considering the process plans, part dimensions, required productiveness, cost of equipment, variability and longevity of market demand, etc. For the obtained process plans and production system type, the corresponding line balancing problem is solved. Finally, an estimation of the cost of the production system is made. If the solution or the cost is unsatisfactory, the designer can modify the data and constraints and restart the procedure.

35.3 Line Balancing

As aforementioned, the line balancing (assignment of operation to workstations) is the key problem in the design of transfer lines.

Historically, the line balancing problem was first stated for assembly lines. As far as we know, the earliest publication on assembly line balancing (ALBP) was presented by Salveson [35.13]. Furthermore, exhaustive studies were made by several researchers in the last 50 years, with many interesting applications covered. One comprehensive state of the art has been presented in a special issue [35.14]. Several articles provide broad surveys of this problem; see, for example [35.15–20]. To summarize, the ALBP is NP-hard; see, for example [35.21]. Much research has been generated to solve the problem by developing approximate or exact methods [35.22–32].

The problem of machining line balancing is rather recent. This problem was mentioned in [35.33]. In Dolgui et al. [35.34], it was defined for dedicated transfer lines and first called transfer line balancing problem (TLBP).

Industry favors solving TLBP because the machining lines become too expensive otherwise. The TLBP consists of answering the following questions:

1. Which machining units are to be chosen to execute the required operations?
2. How many workstations are necessary?
3. How should the machining units be assigned to the stations?

These questions can be answered by an intelligent assignment of operations and machining units to workstations, minimizing the line cost while satisfying the objective production rate as well as respecting all other constraints.
Several exact and approximate (or heuristic) methods for TLBP have been proposed. Exact methods are useful to better understand the problem, however for large-scale problems they require excessive computation time. Contrarily, approximate methods can provide quicker results but do not guarantee the optimality of solutions. Additionally, a heuristic algorithm is often easier to develop than optimal procedures.

The most significant methods for an exact resolution of the TLBP are:

- Linear programming in mixed variables: the problem is modeled as a mixed integer program and solved with an optimization tool such as ILOG Cplex [35.35–37].
- Dynamic programming: a recursive method used for the resolution of problems having an additive objective function. Examples of this approach for TLBP are given in [35.33, 34, 38, 39], where the initial problems were transformed into constrained shortest-path problems and solved with appropriate algorithms.
- Branch and bound: an implicit enumerative procedure which avoids verifying all solutions. Several works use this approach for the resolution of the TLBP; see, for example, [35.40, 41].
- Column generation method can be used for TLBP. Indeed, it was already successfully used for assembly line balancing; see, for example, [35.42].

For large-scale problems, or when the allocated computing time is severely limited (e.g., for flexible transfer lines), several approximate methods have been designed. We classify these methods into two categories:

1. Heuristics based on priority rules derived from the methods for ALBP. There are several heuristic algorithms, which differ in the rule(s) used:
   - Ranked positioned weight (RPW) [35.22]: based on the weights of the operations calculated from their execution time and the operational times of their successors [35.43].
   - Computer method of sequencing operations for assembly lines (COMSOAL) [35.24]: solutions are generated by assigning operations randomly to the stations [35.44–47].

2. Metaheuristics, i.e., solving strategies applicable to a wide range of combinatorial optimization problems:
   - A multistart decomposition approach was suggested in [35.43, 48].

An example of machining line balancing via simulation can be found in [35.49].

Note that most of these methods were developed for dedicated transfer lines. In the next section, we will show how this approach can be applied to flexible and reconfigurable transfer lines. To illustrate, an industrial case study will be presented with a mixed integer programming model.

### 35.4 Industrial Case Study

#### 35.4.1 Description of the Case Study

In Fig. 35.7, the machining line considered in this case study is presented. This line is designed to manufacture automotive cylinder heads. It is equipped with CNC machines (machining centers) for the output of 1250 parts per day. All the machines are identical (line modularity principle), with some exceptions. In contrast to dedicated transfer lines with multispool machines, here, each machine contains one spindle and a magazine for tools. For each machine, to pass from one operation to the next it is necessary to consider an additional time due to tool changes and displacements or/and the rotation of the part (setup time). Taking into account the fact that a part is held at a machine with some fixtures in a given position (part fixing and clamping), some faces and elements of the part are not accessible for machining even after part displacement or rotation. Whatever positioning and clamping are chosen some areas on the part will be hidden or covered. Therefore, the choice of a part position for part fixing should be also considered in the optimization procedure.

In Fig. 35.7, lines (1) represent the transport system composed of conveyors. Robots are used for part loading and unloading. The boxes (2) represent the CNC machines. Machines in a group aligned vertically represent a workstation. Then a workstation can comprise more than one machine; in this case, the same operations are duplicated and executed on different machines. With the parallel machines at each station, the line is easily reconfigurable. The line cycle time can be modified, if necessary, and even be shorter than the
The boxes (3) represent dedicated stations for specific operations such as assembly or washing.

To help the designer of this line, we developed a model for line balancing. The input data used were:

- **Cycle time** (*takt* time) imposed by the objective production rate: one part is produced at each cycle.
- **Precedence constraints**: relations of order between operations. These relations define feasible sequences of operations.
- **Inclusion constraints**: the need to carry out fixed groups of operations on the same workstation.
- **Exclusion constraints**: the impossibility of carrying out certain subsets of operations at the same workstation.
- **Accessibility constraints**: these are related to the positioning of the part; indeed, for a position some part sides are not accessible, and thus operations on these sides cannot be carried out without repositioning. In the considered machining line, only one part fixing position is defined for each workstation (part repositioning occurs between two stations).
- **Sequence-dependent setup times**: the time required for the execution of two sequential operations is not equal to the sum of their times but also depends on the order in which they are done, because the time needed for the displacement/change of tool and part rotation are not negligible.
- **Parallel machines**: at each workstation several identical CNC machines are installed. Thus, the local cycle time of the workstation is equal to the number of parallel machines multiplied by the line cycle time (*takt* time). The machines of the same workstation execute the same operations (in parallel on different product units).

Hence, here, we have a special case of line balancing with a sequential execution of operations, setup times, parallel machines, as well as accessibility, exclusion, and inclusions constraints.

The line of the case study can be regarded as reconfigurable. Indeed, while designed for the production of a single product, if there are changes on the product characteristics, the reconfiguration of this line is possible and easy thanks to:

- The use of standard and identical CNC machining centers, which simplifies the reallocation of operations to the workstations.
- At each station, machining centers can be added or eliminated as needed thanks to this modularity.
Now, we present a mixed integer programming (MIP) model for the design of this line for a given product. Furthermore, at the end of this section, we will give an extension of this model which can be used when reconfiguring the line for another product.

### 35.4.2 Mixed Integer Programming (MIP)

To summarize the optimization problem, we will enumerate its main assumptions.

The set of all operations $N$ to be executed at the line is determined by the process plans for the product for which the line is designed. A part to be machined will pass through a sequence of workstations in the order of their installation. Each workstation is provided with at least one machine which carries out operations during the line cycle time. In the case where workload time of a workstation exceeds the line cycle time, parallel and identical machines are installed. In this case, the local cycle time is equal to the number of parallel machines multiplied by the line cycle time. All machines of the same station execute the same operations.

There are four types of additional constraints on the assignment of the operations (as detailed earlier), namely:

- Precedence constraints
- Exclusion constraints
- Inclusion constraints
- Accessibility constraints.

The time required for the execution of two operations is not equal to the sum of their times but depends on the sequence in which they are executed (Fig. 35.8).

The optimization problem consists of assigning operations to workstations to minimize the total number of machines on the line while respecting the given constraints.

### Mathematical Model

We will introduce the following notations.

- $i, j$ for operations
- $q$ for the place (order) of an operation in the sequence of assigned operations
- $n$ for the number of parallel machines at a workstation
- $k$ for the workstations
- $a$ for the part fixing positions

#### Indexes:

- $i, j$: operations
- $q$: place (order)
- $n$: number of parallel machines
- $k$: workstations
- $a$: part fixing positions

#### Parameters:

- $N$: set of operations (1, ..., $|N|$)
- $A$: set of part positions (1, ..., $|A|$)
- $t_i$: operational time for operation $i$
- $t_j$: setup time for operation $j$
- $t_{0l}$: maximum number of operations authorized to be assigned to a workstation
- $n_0$: maximum number of machines
- $m_0$: maximum number of workstations
- $q_0$: maximum number of possible assignments (places)
- $t_{0l}$: operational time for operation $i$
- $t_{0l}$: setup time for operation $j$
- $T_0$: objective line cycle time
- $P_i$: set of direct predecessors
- $P^*_i$: set of all predecessors
- $F^*_i$: set of all successors
- $E$, collection of subsets $e$: operations which must be assigned to the same workstation
- $E$: set of pairs of operations $(i, j)$ which cannot be assigned to the same workstation
- $A(i)$: set of possible part fixing positions
- $S(k)$: set of possible places for operations at workstation $k$
- $K(i)$: set of workstations on which operation $i$ can be processed

**Fig. 35.8** Sequence-dependent setup times
• $Q(i)$, the set of possible places for operation $i$ in the sequence of all operations: $Q(i) \subseteq \{1, 2, \ldots, k_{\text{max}}\}$
• $N(k)$, the set of operations which can be processed at workstation $k$
• $M(q)$, the set of operations which can be assigned to the place $q$ in the sequence
• $E_i$, the earliest workstation to which operation $i$ can be assigned
• $L_i$, the last workstation to which operation $i$ can be assigned

Variables:
• $x_{iq} = 1$, if operation $i$ is in $q$th place ($q$ is its order in the overall assignment sequence), otherwise $x_{iq} = 0$;
• $\tau_q$, the setup time required between operations assigned to the same workstation in place $q$ and $q + 1$ (Eq. 35.9);
• $y_{nk} = 1$, if there are $n$ parallel machines at the workstation $k$, 0 otherwise;
• $z_{ka} = 1$, if for the part of the workstation $k$ the fixing position $a$ is used, 0 otherwise.

Note that, if an operation is assigned to place $q$, it is the $q - \lceil q/l_0 \rceil \cdot l_0$th operation of the workstation $[q/l_0]$.

The optimization model is as follows:

• The objective function (35.1) minimizes the total number of machines

\[
\text{Minimize} \quad \sum_{k=1}^{m_0} \sum_{n=1}^{n_0} n \cdot y_{nk}. \tag{35.1}
\]

• Equation (35.2) verifies that there is only one value for the number of parallel machines on each workstation

\[
\sum_{n=1}^{n_0} y_{nk} \leq 1, \forall k = 1, 2, \ldots, m_0. \tag{35.2}
\]

Fig. 35.9 Definition of the parameter $\tau_q$

• Equation (35.3) assures that a workstation is open only if the preceding workstation is also open

\[
\sum_{n=1}^{n_0} y_{nk} \geq \sum_{n=1}^{n_0} y_{n(k+1)}, \quad \forall k = 1, 2, \ldots, m_0 - 1. \tag{35.3}
\]

• Equation (35.4) assures that each operation $i$ is assigned once and only once

\[
\sum_{q \in Q(i)} x_{iq} = 1, \quad \forall i \in N. \tag{35.4}
\]

• The constraints (35.5) assure that a place in the sequence is occupied by only one operation

\[
\sum_{i \in M(q)} x_{iq} \leq 1, \quad \forall q = 1, 2, \ldots, q_n. \tag{35.5}
\]

• Equation (35.6) assures that an operation is assigned to a place only if another operation is assigned to the preceding place of the sequence (there is no empty place in the sequence of assigned operations)

\[
\sum_{i \in M(q-1)} x_{iq-1} \geq \sum_{i \in M(q)} x_{iq}, \quad \forall q \in S(k) \setminus \{S(k)\}, \quad \forall k = 1, 2, \ldots, m_0. \tag{35.6}
\]

• Equation (35.7) verifies that only one part fixing position is chosen for each workstation

\[
\sum_{a \in A} z_{ka} \leq 1, \quad \forall k = 1, 2, \ldots, m_0. \tag{35.7}
\]

• Equation (35.8) assures that accessibility constraints are respected (the part fixing position chosen for a workstation authorizes the execution of every operation assigned to this station)

\[
\sum_{q \in S(k)} x_{iq} \leq \sum_{a \in A(i)} z_{ka}, \quad \forall k = 1, 2, \ldots, m_0, \forall i \in N. \tag{35.8}
\]

• Equation (35.9) calculates the additional time between operation $i$ and operation $j$ when operation $j$ is processed directly after operation $i$ at the same workstation

\[
\tau_q \geq t_{ij} \cdot (x_{iq} + x_{j(q+1)} - 1), \quad \forall i \in M(q), \quad \forall j \in M(q + 1), \quad \forall q \in S(k) \setminus \{S(k)\}, \quad \forall k = 1, 2, \ldots, m_0. \tag{35.9}
\]
● Equation (35.10) assures that the workload time of every workstation does not exceed the local cycle time, which corresponds to the number of installed parallel machines at this workstation multiplied by the objective cycle time of the line

\[ \sum_{q \in S(k)} \tau_q + \sum_{i \in N(k)} \sum_{q \in S(k)} t_i \cdot x_{iq} \leq T_0 \sum_{n=1}^{n_0} n \cdot y_{nk}, \quad \forall k = 1, 2, \ldots, m_0. \]  

(35.10)

● Equation (35.11) defines the precedence constraints between operations

\[ \sum_{q \in Q(j)} q \cdot x_{jq} \leq \sum_{q \in Q(i)} q \cdot x_{iq}, \forall i \in N, \quad \forall j \in P_i. \]  

(35.11)

● Equation (35.12) represents the inclusion constraints

\[ \sum_{q \in S(k) \cap Q(i)} x_{iq} = \sum_{q \in S(k) \cap Q(j)} x_{jq}, \quad \forall i, j \in e, \quad \forall e \in ES, \quad \forall k \in (i). \]  

(35.12)

● Equation (35.13) represents the exclusion constraints

\[ \sum_{q \in S(k)} (x_{iq} + x_{jq}) \leq 1, \quad \forall (i, j) \in ES, \quad \forall k \in (i) \cap (j). \]  

(35.13)

● Equations (35.14)–(35.17) provide additional constraints on the possible values of variables

\[ \tau_q \geq 0, \quad \forall q = 1, 2, \ldots, q_0, \]  

(35.14)

\[ x_{iq} \in [0, 1], \quad \forall i \in N, \quad \forall q \in Q(i), \]  

(35.15)

\[ y_{nk} \in [0, 1], \quad \forall n = 1, 2, \ldots, n_0, \quad \forall k = 1, 2, \ldots, m_0, \]  

(35.16)

\[ z_{ka} \in [0, 1], \quad \forall k = 1, 2, \ldots, m_0, \quad \forall a \in A. \]  

(35.17)

### 35.4.3 Computing Ranges for Variables

The model (35.1–35.17) can be solved using a standard operational research solver, for example, ILOG Cplex. Nevertheless, the calculation time is prohibitive. The resolution time for the model (35.1–35.17) can be greatly decreased using efficient techniques to reduce the number of variables (the size of the model) and consequently to accelerate the search for an optimal solution.

We propose a technique for calculating bounds for the possible indexes for the variables of the mathematical model. This can simplify the problem and thus reduce the calculation time.

Taking into account the different constraints between operations, we can calculate the sets \( K(i), N(k), S(k), Q(i), \) and \( M(q) \) more precisely. Note that these sets give intervals of possible values for the corresponding indexes.

The following additional notations can be defined:

- \( E_i[r] \) is a recursive variable for the step by step calculation of the value of \( E_i \) taking into account setup times between operations, \( r = 0, 1 \).
- \( L_i[r] \) is a recursive variable for the step by step calculation of the value of \( L_i \) taking into account setup times between operations, \( r = 0, 1 \).

With \( P_i^s \), which is the set of all predecessors of operation \( i \), and \( F_i^s \), which is the set of all successors of operation \( i \), we can also introduce:

- \( S_{P_i}[r] \): the sum of the \( |P_i^s| - E_i[r] + 1 \) shortest setup times between the operations of the set \( P_i^s \cup \{i\} \) composed of operation \( i \) and all its predecessors, \( i \in N \)
- \( S_{F_i}[r] \): the sum of the \( |F_i^s| - m_0 + L_i[r] \) shortest setup times between the operations of the set \( F_i^s \cup \{i\} \) composed of operation \( i \) and all its successors, \( i \in N \)
- \( d[i, j] \): a parameter (distance) which has the following property: if \( (i, j) \) or \( (j, i) \in ES \), then \( d[i, j] = 1 \), else \( d[i, j] = 0 \).

The total operational time \( T_{sum} \) without considering the setup times between operations is calculated as follows

\[ T_{sum} = \sum_{i \in N} t_i. \]

A lower bound on the number of workstations can be calculated by supposing that each workstation contains \( n_0 \) machines. Therefore, the local cycle time of each workstation is equal to \( (T_0 \cdot n_0) \). The line becomes a serial line composed of identical workstations with a cycle time which is equal to \( (T_0 \cdot n_0) \). Then, a lower bound on the number of workstations \( LB_{ws} \) can be calculated as follows

\[ LB_{ws} = \left\lceil T_{sum}/(T_0 \cdot n_0) \right\rceil, \]
where the notation $\lceil x \rceil$ indicates the lowest integer value higher than or equal to $x$.

In the same way, a lower bound on the number of machines in the line ($LB_m$) can be determined by the following expression

$$LB_m = \left\lceil \frac{T_{sum}}{T_0} \right\rceil .$$

Thus, the following procedure calculates the sets $K(i)$, $Q(i)$, $M(q)$, $N(k)$, and $S(k)$. Note that the operations are numbered in order of precedence graph ranks (in topological order). Some lines are annotated with comments. The symbol "//" is used to mark the beginning and the end of these comments.

**Algorithm**

**Step 1** // step-by-step calculation of $E_i$ and $L_i$, taking into account precedence constraints and setup times

```plaintext
for all $i \in N$
begin
  // calculate the earliest workstation $E_i[0]$ on which operation $i$ can be processed taking into account the precedence constraints; note that an operation cannot be processed before its predecessors
  $E_i[0] \leftarrow \left\lfloor \frac{(t_i + \sum_{j \in P_i^+} t_j)}{(n_0 \cdot T_0)} \right\rfloor + 1; \quad // calculate the latest workstation $L_i[0]$ on which operation $i$ can be processed considering the precedence constraints, note that an operation cannot be processed after its successors
  \begin{align*}
    L_i[0] &\leftarrow m_0 - \left\lfloor \frac{(t_i + \sum_{j \in P_i} t_j)}{(n_0 \cdot T_0)} \right\rfloor + 1; \\
    \text{// calculate $E_i[1]$, which are new values of $E_i$ obtained by taking into account in addition setup times between operations}
  \end{align*}
  $E_i[1] \leftarrow \left\lfloor \frac{(t_i + S_{pi}[i]) + \sum_{j \in P_i^+} t_j)}{(n_0 \cdot T_0)} \right\rfloor; \\
  \text{// calculate $L_i[1]$, which are new values of $L_i$ obtained by taking into account in addition setup times between operations}
  L_i[1] \leftarrow m_0 - \left\lfloor \frac{(t_i + S_{pi}[i]) + \sum_{j \in P_i} t_j)}{(n_0 \cdot T_0)} \right\rfloor + 1; \\
  \text{// updating the values of $E_i$}
  \begin{align*}
    \text{if } E_i[1] \neq E_i[0] \text{ then } E_i &\leftarrow \max\left\{E_i[0] + 1, \\
    &\left\lfloor \frac{(t_i + S_{pi}[i]) + \sum_{j \in P_i} t_j)}{(n_0 \cdot T_0)} \right\rfloor \right\}; \\
    \text{else } E_i &\leftarrow E_i[1]; \\
    \text{// updating the values of $L_i$}
  \end{align*}
  \begin{align*}
    \text{if } L_i[1] \neq L_i[0] \text{ then } L_i &\leftarrow \min\left\{L_i[0] - 1, \\
    m_0 - \left\lfloor \frac{(t_i + S_{pi}[i]) + \sum_{j \in P_i} t_j)}{(n_0 \cdot T_0)} \right\rfloor + 1 \right\}; \\
    \text{else } L_i &\leftarrow L_i[1];
  \end{align*}
end
```

**Step 2** // step-by-step calculation of $E_i$, taking into account exclusion and inclusion constraints

```plaintext
j_{min} \leftarrow j_{cur}; \quad j_{cur} \leftarrow |N|; \quad // new values of $E_i$ are calculated by considering exclusion constraints
for $j \leftarrow j_{min} + 1, \ldots, |N|$ do
begin
  $E_j \leftarrow \max\left\{E_i + d[i, j], E_j\right\}; \\
  \text{for each } e \in ES \text{ do}
  \begin{align*}
    E_e &\leftarrow \max(E_j); \\
    \text{for each } j \in ee \text{ if } E_j < E_e \text{ then begin}
  \end{align*}
  \begin{align*}
    \text{// new value of $E_i$ is calculated, now taking into account an inclusion constraint}
    E_i &\leftarrow \max\left\{E_i + d[i, j] \right\}; \\
  \text{end}
  \end{align*}
end
until $j_{cur} = |N|$.
```

**Step 3** // step-by-step calculation of $L_i$, taking into account inclusion and exclusion constraints

```plaintext
j_{max} \leftarrow j_{cur}; \quad j_{cur} \leftarrow 1; \quad // new values of $L_i$ are calculated by considering exclusion constraints
for $j \leftarrow j_{max} - 1, \ldots, 1$ do
begin
  $L_j \leftarrow \min\left\{L_i - d[j, i], L_j\right\}; \\
  \text{for each } e \in ES \text{ do}
  \begin{align*}
    L_e &\leftarrow \min(L_j); \\
    \text{for each } j \in ee \text{ if } L_j > L_e \text{ then begin}
  \end{align*}
  \begin{align*}
    \text{// new values of $L_i$ are again calculated, now taking into account inclusion constraints}
    L_i &\leftarrow L_j; \\
    j_{cur} &\leftarrow \max(j_{cur}, j); \\
  \end{align*}
end
```

Fig. 35.10 An example of initial values for $E_i$ and $L_i$

Fig. 35.11 Modified values of $E_i$ and $L_i$ taking into account setup times

end
end
until $j_{	ext{end}} = 1$.

**Step 4** // calculation of the sets $K(i)$, $N(k)$, $S(k)$, $Q(i)$, and $M(q)$ //
for all $i \in N$ do
   $K(i) \leftarrow \{E_i, L_i\}$;
for $k \leftarrow 1, 2, \ldots, m_0$ do

begin
   $N(k) \leftarrow \{i | i \in N, k \in K(i)\}$;
   $S(k) \leftarrow \left[1 + \sum_{k'=1}^{k-1} |S(k')|, \min (|N(k)|, l_0) + \sum_{k'=1}^{k-1} |S(k')|\right]$;
end
for all $i \in N$ do
   $Q(i) \leftarrow \min \{S(E_i)\}, \max \{S(L_i)\}$;
   for $q \leftarrow 1, 2, \ldots, \max(S(m_0))$ do
      $M(q) \leftarrow \{i | q \in Q(i)\}$;
   End of algorithm.

Some illustrations of the algorithm rules are presented in Figs. 35.10–35.13.

**Numerical Example**

In order to better explain the suggested algorithm, we present a numerical example with ten operations. Figure 35.14 shows the precedence graph and operational times.

The objective line cycle time is: $T_0 = 16$ units of time; the maximum number of stations $m_0 = 6$; the maximum number of machines to be installed on a station $n_0 = 3$; the maximum number of operations to be assigned to a station $l_0 = 8$.

The inclusion constraints are: $ES = \{(2, 4); (8, 9); (5, 6)\}$.

The exclusion constraints are: $ES = \{(2, 7); (3, 4)\}$.

The setup times are reported in Table 35.1. For example, the setup time $t_{4,5} = 3$ corresponds to the time that is required to perform operation 5 immediately after operation 4.

The total operational time, $T_{\text{sum}} = \sum_{i \in N} t_i = 161$ units of time.

**Fig. 35.12** An example of modifications of $E_i$ by considering an exclusion constraint
A lower bound on the number of workstations is:  
\[ LB_{ws} = \left\lceil \frac{T_{sum}}{T_0 \cdot n_0} \right\rceil = \left\lceil \frac{161}{16 \cdot 3} \right\rceil = 4. \]

Thus, the optimal solution cannot have fewer than four workstations.

A lower bound on the number of machines is:  
\[ LB_{m} = \left\lceil \frac{T_{sum}}{T_0} \right\rceil = \left\lceil \frac{161}{16} \right\rceil = 11. \]

Then, the optimal solution cannot have fewer than 11 machines.
Now, the procedure of range calculation for indexes is applied:

- **Step 1** The initial values of $E_i$ and $L_i$ for each operation are calculated considering set-up times between operations and precedence constraints (Fig. 35.15).

- **Step 2** The new values of $E_i$ are obtained by considering the exclusion and inclusion constraints (Fig. 35.16).

- **Step 3** The new values of $L_i$ are calculated by considering the exclusion and inclusion constraints (Fig. 35.17).

- **Step 4**
  - Sets $K(i)$ for operations $i \in N$ are obtained (Table 35.2)
  - Sets of operations $N(k)$ for stations $k = 1, 2, \ldots, m_0$ are defined (Table 35.3)
  - Range of places $S(k)$ for operations of station $k$ is calculated, $k = 1, 2, \ldots, m_0$ (Table 35.4)
  - Finally, the range of places $Q(i)$ for operation $i$ is found, for all $i \in N$ (Table 35.5).

### 35.4.4 Reconfiguration of the Line

As indicated at the beginning of this section, the studied line is reconfigurable. After the implementation of the line, if there are changes in the product characteristics or if there is a new product to be machined, the line can be reconfigured. Such a reconfiguration problem consists of reassigning operations to the stations.
Table 35.5 The ranges of places $Q(i)$ for operations

<table>
<thead>
<tr>
<th>Operation $i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q(i)$</td>
<td>(1,23)</td>
<td>(3,23)</td>
<td>(1,15)</td>
<td>(3,23)</td>
<td>(3,29)</td>
<td>(3,29)</td>
<td>(9,29)</td>
<td>(16,29)</td>
<td>(16,29)</td>
<td>(16,30)</td>
</tr>
</tbody>
</table>

while minimizing the number of additional machines and/or those that move from a workstation to another in order to rebalance the line. This problem is similar to the design problem considered in the previous sections. Therefore, the proposed MIP model (35.1–35.17) can be easily adapted for this new problem.

The following modifications are made in the model (35.1–35.17): a new objective function is considered (35.1’) along with additional constraints (35.18) and a new set of variables (35.19) which represents the gap between the number of machines for each station in the line before reconfiguration ($y_{\text{Old}}^{\text{nk}}$), and their number in the line reconfigured.

$$\begin{align*}
\text{Minimize} & \quad \sum_{k=1}^{n_0} \left( \delta^+_k + \delta^-_k \right), \\
\sum_{n=1}^{n_0} n \cdot y_{nk} - \sum_{n=1}^{n_0} n \cdot y_{\text{Old}}^{\text{nk}} & = \delta^+_k - \delta^-_k, \\
\forall k & = 1, 2, \ldots, m_0, \\
\delta^+_k, \delta^-_k & \geq 0, \quad \forall k = 1, 2, \ldots, m_0.
\end{align*}$$ (35.18) (35.19)

35.5 Conclusion and Perspectives

Transfer lines are used in many manufacturing domains, especially in machining systems, to efficiently effectuate high-quality and economical production. In today’s competitive business environment, several manufacturers have opted for transfer lines to benefit from their advantages, namely precision, quality, productivity, reduction of handling cost, etc. However, transfer lines also present some drawbacks, such as requiring a large investment. Normally, transfer lines are highly automated, but the level of automation depends on the type of customer demand. Three types of transfer lines exist: dedicated, flexible, and reconfigurable. Dedicated lines are composed of workstations with multip spindle heads. Flexible transfer lines have several types of CNC machines. Reconfigurable lines offer a mix of different types of machines (special machines, CNC machines, machining units, etc.) and can have different architectures (simple line, U-line, parallel stations, etc.).

With increasing technological progress and development of ever more sophisticated and efficient machining equipment, the problem of automated machining line design is exceptionally pertinent. Indeed, the concepts for machining lines are continuously improved through the development of new types of architectures and machines. Unfortunately, there is a gap between industrial cases and research problems treated. In contrast with assembly systems, in the domain of machining lines, the gap is often due to the lack of collaboration between the industrial and academic worlds.

In this chapter, written jointly by academic (Ecole des Mines de Saint Etienne) and industrial (PCI/SCEMM) partners, a general overview of the automated machining lines is presented. The principal characteristics of these lines and a general methodology for their design are introduced. This methodology is valid independent of the type of the line: dedicated, flexible, or reconfigurable. The goal is to help machining line manufacturers to design efficient lines and become more competitive.

On receipt of a customer demand for a line which includes a part description (plans, characteristics, etc.) and the required output, the machining line manufacturer must be able to propose a complete solution within a very short time interval. This preliminary solution concerns the line architecture, number of machines, and equipment, with a line cost evaluation. A major difficulty deals with line balancing, which is a hard combinatorial optimization problem. All types of transfer lines are concerned. In this chapter, a short survey of general line balancing approaches is given. The methods developed for the balancing of the automated machining lines are enumerated and commented. Then, an industrial case study is presented. It illustrates and highlights the importance of the line balancing problem. Afterwards, a mixed integer program for the considered case is proposed. The presented model and approach are useful from a practical perspective. They generate more appropriate preliminary solutions to customer needs within a very short timeframe. From an academic
point of view, this is a new formulation of the line balancing problem with sequence-dependent setup times and parallel machines.

For future research work, beside the improvement of the models and resolution algorithms, numerous research perspectives have yet to be studied in this field. Among them, the combination of optimization methods and discrete-event simulation seems promising. Simulation is a powerful method to illustrate and study the flow of material on the line and to determine the effect of its architecture on line reliability and performance. Also, the development of interactive and iterative software could provide useful decision-aid systems for industry.

References

35.3 K. Hitomi: Manufacturing System Engineering (Taylor & Francis, London 1996)
35.6 A. Delchambre, J.M. Proth: Les systèmes de production modernes (Hermes-Science, Paris 2006)
35.23 C.L. Moodie, H.H. Young: A heuristic method for assembly line balancing for assumption of constant or variable elements time, J. Ind. Eng. 16, 23–29 (1965)
35.27 A. Scholl: Balancing and Sequencing of Assembly Lines (Physica, Heidelberg 1999)
35.31 J. Bukchin, A. Rubinson: A weighted approach for assembly line design with station parallel-


35.44 B. Finel: Structuration de lignes d’usinage: méthodes exactes et heuristiques. Ph.D. Thesis (Université de Metz, Metz 2004), in French


