dc_924_14

SENSITIVITY ANALYSIS AT PRODUCTION PLANNING AND PRODUCTION SCHEDULING MODELS

TAMÁS KOLTAI

PH.D. IN ENGINEERING
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE TITLE OF D.SC. AT HUNGARIAN ACADEMY OF SCIENCES

BUDAPEST, 2015

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The economical operation of production and service systems has always been a major concern of engineers. In 1886, Henry Towne of the Yale and Towne Company in a paper titled "Engineer as an Economist" recommended that a new section should be organized within the American Society of Mechanical Engineers (ASME). This new section could be a forum for those professionals, who are mechanical engineers, but interested in the economic aspects of production (Towne, 1886). The recommendation of Towne initiated the rapid development of a new branch of engineering, which is called *industrial engineering* (IE) (Hicks, 1977). The official definition of industrial engineering formulated by the Institute of Industrial Engineering (IIE) states (Salvendy, 1992) that

> *"Industrial engineering is concerned with the design, improvement and installation of integrated systems of people, materials, information, equipment and energy. It draws upon specialized knowledge and skill in the mathematical, physical and social sciences together with the principles and methods of engineering analysis and design, to specify, predict, and evaluate the results to be obtained from such systems."*

The problems, scientific foundations and information processing infrastructure of IE have changed considerably since the time of Towne. By now, all stages of the lifecycle of production systems − design, implementation, operation, improvement and restructuring − are influenced by industrial engineering. The problems discussed in the dissertation are related to planning and scheduling of production operations. *Production planning and production scheduling* determines the allocation of manufacturing resources to production task on the medium and short run, that is, a plan should be prepared a priori, to determine how much and when to produce of the different parts/products, and what amount of resources should be applied during production. In the course of production, the difference of the plan and the actual operation must be compared, analyzed, and appropriate control actions are to be implemented if required.

The solution of the complicated problems of production planning and control is supported by *operations research* methods. Frequently, optimization models are formulated to determine the best possible production plan and production schedule. By the time of implementation or during operation several parameters − which were used to obtain the implemented solution − may change. Some of these changes are important and require actions on behalf of the decision maker. Other changes, however, may not influence the implemented decision, although influence the result of operation. Consequently, the robustness of the plan, that is, the sensitivity of the results with respect to some model parameters is important information for management decision-making (see for example, Little, 1970; Ragsdale, 2007; Monostori et al., 2010).

Sensitivity analysis methods can be applied to get information about the effect of parameter changes on an optimal or on a heuristic solution. The objective of sensitivity analysis is to determine the effect of the change of parameters or conditions assumed in the planning phase on some performance measures important for decision-making. The method used for generating sensitivity information depends on the planning model, on the changing factor (parameter, condition) and on the performance measure.

Sometimes, calculating the value of the objective function with the original and with the changed values of a parameter, and analyzing the difference may lead to general sensitivity conclusions. One of the first, widely documented sensitivity analysis applied in industrial engineering has been the examination of the robustness of the classical economic order quantity (EOQ) formula in inventory management, which dates back to 1913 (Harris, 1913).

This analysis is based on the examination of the total cost function when some parameters (order cost, inventory holding rate) change, and the results are discussed in most of the basic production management related university textbooks (see for example, Nahmias, 1993; Anderson, 1994; Waters, 1996).

The simple analysis of the change of the objective function value, however, may not always be sufficient to get proper sensitivity information. Sometimes, studying the structure of the problem may lead to analytical description of robustness and sensitivity. In linear programming, for example, the simplex method provides information about the sensitivity of the objective function and about the validity range of this sensitivity. This information is provided by the simplex table of the optimal solution (see for example Hillier and Lieberman, 1996; Prékopa, 1968). In case of discrete event simulation, perturbation analysis can be used to obtain sensitivity information related to a performance measure from a single sample path (Ho and Cao, 1991).

Frequently, in complex models, numerical examination of the behavior of an objective function in a predefined parameter range must be performed. Such analysis, however, requires extensive computations and advanced information processing environment.

No matter which technique is applied, information about sensitivity is important for the decision maker due to the following three reasons:

− Some model parameters *may change* despite of the intention of the decision maker. For example, a customer demand may change, an operation cost may increase or production capacity may decrease by the time a production plan is implemented. The operation manager must know whether the change of operation is required or the change of the parameters does not influence the plan, consequently, the change of operation is not necessary.

− The decision maker may *have the possibility* to change some parameters. For example, a selling price can be changed, a production capacity can be increased by overtime or a new production route of a part can be implemented. Analysis of the possible effects of these parameter changes is required before decision on implementation is made.

− Sometimes model parameters may change but the *change of operation is not possible* even if it were required. In these cases, the analysis of the consequences of the change helps to determine how far the actual operation is from the optimum, and what measures should be taken to avoid the unfavorable effects in the future.

The proper presentation of sensitivity information for management decision-making is also a very important question. Generally, sensitivity results consist of a large amount of data. The *range* information of a linear programming solution contains the sensitivity range of several thousand objective function coefficients and shadow prices. Filtering and clustering this information is necessary for efficient decision-making. Graphical presentation of these data might considerably direct the attention of the decision maker to critical points, and highlights the most efficient intervention areas (Eschenbach, 1992).

Consequently, sensitivity analysis is important from theoretical and from practical points of view as well, and a strong emphasis is made to develop both theory and technique to obtain sensitivity information in several production related fields (see for example Wagner, 1995; Saltelli, Tarantola and Campolongo, 2000; Higle and Wallace, 2003 or Hall and Posner, 2004; Kövesi, 2011).

The application of sensitivity information to improve decision-making is not just a possibility, but also a *necessity*. The development of the theory and technology of data mining has a strong impact on production related decisions as well (Jackson, 2002). The information obtained by the processing of a large amount of data can be a source of competitive advantage according to Davenport (2006). If these data are efficiently collected (about customer behavior, about operation, about environmental conditions, etc.), if proper (statistical, operational research) methods are used to process these data, and the results are adequately channeled into the decision-making process, then the competitors can be outperformed. The emphasis on the collection of a large amount of data and on the efficient processing of the collected information forms the basis of a new paradigm in management decision-making which is called *competing on analytics* (Davenport, 2006; Davenport and Harris, 2007; Koltai, 2007).

The support of competition with the results of the processing of a large amount of data is possible as a consequence of the development of information technology. Data can be collected automatically about the progress of a production or a service process at reasonably low cost and with acceptable speed. The collected data can be processed with efficient and easy to use statistical and operation research software even on an ordinary computer. Consequently, *competing on analytics* and *big data management* is becoming a general approach when decision support systems are designed (Davenport 2013).

On the one hand, the availability of a large amount of data about actual operation, and the possibilities of advanced data processing environment provide excellent opportunities for generating sensitivity information. On the other hand, sensitivity information is constantly demanded in a system which strives for continuous improvement of its operation.

The dissertation discusses the theoretical and practical aspects of sensitivity analysis results related to production planning and scheduling problems. In some cases, the derivation and analysis of sensitivity results of existing models are the main objective of the research. In other cases, new models are formulated and sensitivity analysis supports the application of the models. The following research problems are discussed in details in the dissertation:

1) Linear Programming (LP) is frequently applied to solve production planning problems. The sensitivity information of the optimal plan with respect to the model parameters is important information for capacity extension, operation improvement and customer related decisions. In case of degenerate optimal solution, however, sensitivity information generated by the traditional LP solvers can be misleading. I have investigated the reason of the misleading sensitivity results and the ways of correcting this information.

2) In flexible manufacturing systems (FMS), products/parts can be manufactured by visiting different machines, that is, the same part may follow several different routes in the manufacturing process. Routing influences the available capacity of the system. Frequently, however, capacity related decisions must be made before parts are assigned to the specific routes. I have investigated how capacity of FMSs can be determined before the routing information is available, and I have analyzed the sensitivity of the results with respect to some basic model parameters.

3) In case of assembly lines, frequently, 0-1 mathematical programming models are used to allocate tasks to workstations. One important shortcoming of assembly line balancing (ALB) models is that the models do not take into consideration several real life conditions. One important group of such conditions is related to workforce skills. I have developed a general framework to formulate workforce skill constraints. I have also investigated the effect of the change of production quantity on the optimal solution.

4) Scheduling problems of production systems are frequently solved with the help of discrete event simulation. The sensitivity of the scheduling criteria to the change of some model parameters is important information for the decision maker. I have analyzed the sensitivity of the throughput time and the sensitivity of waiting time with respect to some operation times, and I have determined the validity of these gradient information using perturbation analysis.

5) Frequently, the objective of scheduling is the minimization of inventory holding cost. There are, however, several ways to determine the value of inventory holding cost. I have investigated how the optimal schedule of a single resource scheduling problem is influenced by the method of inventory holding cost calculation. I have also analyzed the robustness of

the optimal schedule using analytical and numerical methods as well.

Some remarks must be made about the structure and the content of this work.

− Each chapter of the dissertation is related to sensitivity analysis, however, the problems examined, the techniques used for modeling, and the generation of sensitivity information is different in each chapter. Consequently, a different method of notation is required in each chapter. To facilitate the reading of the text, each main chapter has a separate list of notation.

− The results of the main chapters have already been published in relevant scientific journals of the related areas. The content of the main chapters of this work is an edited and integrated version of the corresponding papers. Since these papers were written in the last 20 years, new results, algorithms, software and computing technology may have appeared. Those chapters which are based on earlier papers may not reflect the most up-to date technology, but the scientific results of the chapters are independent of the changes of the technological conditions and of the change of information technology.

2 THE DIFFERENCE BETWEEN THE MANAGERIAL AND MATHEMATICAL INTERPRETATION OF SENSITIVITY ANALYSIS RESULTS IN LINEAR PROGRAMMING

One important problem of production planning is the allocation of production resources to production tasks. This problem is frequently solved by mathematical programming models. Linear programming (LP) has a special role within mathematical programming, because resource allocation problems can be easily described or efficiently approximated by linear relationships. The theory and practice of linear programming is well established and several software are available to support real life applications. The main result an LP problem is the *optimal solution*. In a production planning context the optimal solution determines the optimal allocation of production resources to production tasks. Further results of an LP problem are related to *the sensitivity analysis* of the optimal solution. In some cases LP sensitivity analysis results of the currently available LP solvers provide misleading information. In this chapter the problems of LP sensitivity information are explained, a new categorization of sensitivity information is provided and a calculation framework is suggested. The results of this chapter are based on the papers of Koltai and Terlaky (2000), Koltai and Tatay (2008) and Koltai and Tatay (2011).

2.1 Introduction

Linear programming (LP) is one of the most extensively used operations research technique in production and operations management (Johnson and Montgomery, 1974; Cane and Parker, 1996). As a result of the development of computer technology and the rapid evolution of user friendly LP software every operation manager can run an LP software easily and quickly on a laptop computer. Although to solve LP models is now accessible for everybody, the interpretation of the results requires a lot of skill. Most of the management science and operational research textbooks pay a special attention to sensitivity analysis, and to the problems of degeneracy, but sensitivity analysis under degeneracy is rarely discussed. Commercially available software do not give enough information to the user about the existence and about the consequences of these, very common, special cases. In practice, managers very frequently misinterpret the LP results which may lead to erroneous decisions and to important financial and/or strategic disadvantages.

Several papers have addressed this issue. Evans and Baker (1982) draw the attention to the consequences of the misinterpretation of sensitivity analysis results in decision-making. They illustrate their point with a simple example and list some published cases in which the erroneous interpretation of sensitivity analysis results is obvious. Aucamp and Steinberg (1982) also warn that shadow price analysis is incorrect in many textbooks, and that the shadow price is not equal to the optimal solution of the dual problem when the obtained optimal solution is degenerate. They present some examples of shadow price calculations by commercial packages. Akgül (1984) refines the shadow price definition of Aucamp and Steinberg, and introduces the negative and positive shadow prices for the increase and for the decrease of the right-hand side (RHS) elements. Greenberg (1986) shows that very frequently practical LP models have a netform structure; and netform structures are always degenerate. He illustrates sensitivity analysis of netform type models by one of the Midterm Energy Market Model of the U.S. Department of Energy. Gal (1986) summarizes most of the critics concerning sensitivity analysis of LP models and highlights some important research directions. Rubin and Wagner (1990) illustrate the traps of the interpretation of LP results by using the industry cost curve model in a tutorial type paper written for managers and instructors. Jansen, et al. (1997) explains the effect of degeneracy on sensitivity analysis by using a transportation model, and presents the shortcomings of the most frequently used LP packages. Wendell also pays special attention to correct and practically useful calculation of sensitivity information (see for example Wendell, 1985 and Wendell, 1992). The problem is not that operations researchers are unaware of the difficulties of sensitivity analysis. This issue is discussed thoroughly in the scientific literature, (see for example Gal, 1979; and Wendell, 1992) and a complete, mathematically correct treatment of sensitivity analysis is presented e.g. in Jansen et al. (1997), and in Roos at al. (1997). Practice, however, shows that the problem is not widely known among LP users, and available commercial software packages are not helping to recognize the difficulties.

The main objective of this chapter is to explain the difference between the managerial questions and the traditional mathematical interpretation of sensitivity analysis. In the first part of this chapter the basic definitions are introduced, the most important types of sensitivity information are classified, and degenerate LP solutions are illustrated graphically. Next, a production planning problem is used to demonstrate the consequences of incorrect interpretations of the provided sensitivity information. In the second part of the chapter a practice oriented framework for calculating sensitivity information is provided and sensitivity information for management decision-making are presented for a degenerate production planning problems. Finally, some recommendations are formulated both for decision makers and for software developers. All notations used in this chapter are summarized in Table 2.1.

2.2 Basic definitions and concepts

Every LP problem can be written in the following standard form,

$$\min_{\underline{\mathbf{x}}}\left\{\underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}} \,\middle|\, \mathbf{A}\underline{\mathbf{x}}=\underline{\mathbf{b}};\underline{\mathbf{x}}\geq\underline{\mathbf{0}}\right\} \tag{2.1}$$

where $\mathbf{A}$ is a given J x I matrix with full row rank and where the column vector $\underline{\mathbf{b}}$ represents the right-hand side (RHS) terms and the row vector $\underline{\mathbf{c}}^{\mathrm{T}}$ represents the objective function coefficients. Problem (2.1) is called the *primal problem* and a vector $\underline{\mathbf{x}}\geq 0$ satisfying $\mathbf{A}\underline{\mathbf{x}}=\underline{\mathbf{b}}$ is called a *primal feasible solution*. The objective is to determine those values of the vector $\underline{\mathbf{x}}$ which minimize the objective function. To every primal problem (1) the following problem is associated,

$$\max_{\underline{\mathbf{y}}}\left\{\underline{\mathbf{b}}^{\mathrm{T}}\underline{\mathbf{y}} \,\middle|\, \mathbf{A}^{\mathrm{T}}\underline{\mathbf{y}}\leq\underline{\mathbf{c}}\right\} \tag{2.2}$$

Problem (2.2) is called the *dual problem* and a vector $\underline{\mathbf{y}}$ satisfying $\mathbf{A}^{\mathrm{T}}\underline{\mathbf{y}}\leq\underline{\mathbf{c}}$ is called a *dual feasible solution*. For every primal feasible $\underline{\mathbf{x}}$ and dual feasible $\underline{\mathbf{y}}$ it holds that $\underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}}\geq\underline{\mathbf{b}}^{\mathrm{T}}\underline{\mathbf{y}}$ and the two respective objective function values are equal if and only if both solutions are optimal (see for example Hillier and Liberman, 1995).

Most computer programs to solve linear programming problems are based on a version of the simplex method. Modern, hi-performance packages are furnished with interior point solvers as well; however, the implemented sensitivity analysis is based always on the simplex method. The simplex procedure selects a basis of the matrix $\underline{\mathbf{A}}$ in every step The selected *basis solution* is calculated and the optimality criteria are checked. To define the optimal basis solution some preparation is needed.

Let B be a set of m indices, and $\mathbf{A}_B$ be the matrix obtained by taking only those columns of $\mathbf{A}$ whose indices are in B. If $\mathbf{A}_B$ is a nonsingular matrix then by using the vector $\underline{\mathbf{x}}_B=\mathbf{A}_B^{-1}\underline{\mathbf{b}}$ a basis solution can be defined as

$$x_i=\begin{cases}(\underline{\mathbf{x}}_B)_i & \text{if } i\in B,\\ 0 & \text{otherwise.}\end{cases} \tag{2.3}$$

Table 2.1 Summary of notation of Chapter 2

Subscripts:

i	−	index of the variables of a primal LP problem (i=1,…,I),
j	−	index of the variables of a dual LP problem (j=1,…,J),
t	−	index of the time period in the production planning example (t=1,…,T)
n	−	index of the products in the production planning example (n=1, …,N).
Parameters:		
$\mathbf{A}$	–	coefficient matrix with elements a_{ji},
$\mathbf{A}_\mathrm{B}$	−	coefficient matrix containing only the columns of $\mathbf{A}$ in the basis,
$\underline{\mathbf{b}}$	–	right-hand side vector with elements b_j,
$\underline{\mathbf{c}}$	–	objective function coefficient vector with elements c_i,
$\underline{\mathbf{c}}_\mathrm{B}$	−	objective function coefficients belonging to the variables in the basis,
$\underline{\mathbf{e}}_i$	–	unit vector with I elements, and with e_i=1 and e_k=0 for all $k \neq i$,
$\underline{\mathbf{e}}_j$	–	unit vector with J elements, and with e_j=1 and e_k=0 for all $k \neq j$,
δ	–	perturbation of a right-hand side parameter
n_t	–	number of working days in month t
D_{nt}	–	demand of product n in period t,
p_{nt}	–	unit production cost of product n in period t,
h_{nt}	–	unit inventory holding cost of product n in period t,
C_t	–	production capacity in period t,
W_t	–	warehouse capacity in period t.
Sets:		
B	−	index set containing the indices of the basis variables.
Variables:		
$\underline{\mathbf{x}}$	–	variable vector of the primal problem with elements x_i,
$\underline{\mathbf{x}}_\mathrm{B}$	–	vector of the basis variables with elements $(\underline{\mathbf{x}}_B)_i$,
$\underline{\mathbf{x}}^*$	–	optimal solution of the primal problem with elements x_i^*,
$\underline{\mathbf{y}}$	–	variable vector of the dual problem with elements y_j,
$\underline{\mathbf{y}}^*$	–	optimal solution of the dual problem with elements y_j^* ,
$\underline{\mathbf{s}}$	−	slack variable vector with elements s_j,
OF^*	–	optimal value of the objective function,
y_j^-	–	the left shadow price of right-hand side element b_j (δ<0),
y_j^+	–	the right shadow price of right-hand side element b_j (δ>0),
γ_i	–	change of objective function coefficient c_i,
γ_i^-	–	feasible decrease of objective function coefficient c_i,
γ_i^+	–	feasible increase of objective function coefficient c_i,
ξ_j	–	change of right-hand side element b_j,
ξ_j^-	–	feasible decrease of right-hand side element b_j,
ξ_j^+	–	feasible increase of right-hand side element b_j,
$\mathrm{n}\xi_j^-$	–	feasible decrease of b_j belonging to the left shadow price,
$\mathrm{n}\xi_j^+$	–	feasible increase of b_j belonging to the left shadow price,
$\mathrm{p}\xi_j^-$	–	feasible decrease of b_j belonging to the right shadow price,
$\mathrm{p}\xi_j^+$	–	feasible increase of b_j belonging to the right shadow price,
x_{nt}	–	production quantity of product n in period t,
I_{nt}	–	inventory of product n in period t.

If in addition $\underline{\mathbf{x}}_B \geq 0$ holds then $\underline{\mathbf{x}}$ is called a *primal feasible basic solution*. The variables with their index in B are the *basic variables*; the others are the *non-basic variables*. Dual variables can be associated to any basis $\mathbf{A}_B$ as follows:

$$\underline{\mathbf{y}} = \left(\mathbf{A}_B^{-1}\right)^{\mathrm{T}} \underline{\mathbf{c}}_B \tag{2.4}$$

If $\underline{\mathbf{c}} - \mathbf{A}^{\mathrm{T}}\underline{\mathbf{y}} \geq 0$ then $\underline{\mathbf{y}}$ is a feasible solution for the dual problem, and $\underline{\mathbf{y}}$ is called *dual feasible basic solution*. If the basis $\mathbf{A}_B$ is both primal and dual feasible, then $\mathbf{A}_B$ is an *optimal basis*, and the corresponding basic solutions $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ are *optimal basis solutions* for the primal and for the dual problems respectively. It might happen that a basis gives an optimal primal solution, but the related dual basis solution is dual infeasible. Such a basis is called *primal optimal*. Analogously, when a basis gives a dual optimal solution, but the related primal solution is infeasible, then the basis is called *dual optimal*.

Sometimes the optimal basis is not unique, more than one basis may yield an optimal solution either for the primal or for the dual problem or for both. This is called degeneracy and occurs very frequently in practice. Formally, a basis is called *primal degenerate* when there are variables with zero value among the basis variables and it is called *dual degenerate* when some dual slack variables $s_i = c_i - (\mathbf{A}^{\mathrm{T}}\underline{\mathbf{y}})_i$, not belonging to the basis indices B, are zero. In general, if a basis is either primal, or dual, or from both sides degenerate then we simply say that it is *degenerate*. In case of degeneracy many optimal solutions exists that are not basic solutions.

Very frequently the main parameters of an LP model changes (e.g. cost coefficients, resource capacities, etc.) and it would be important to know if any action on behalf of the decision maker is required as a consequence of these changes. Sensitivity analysis can help to answer this question if it is applied correctly. The objective of sensitivity analysis is to analyze the effect of the change of the objective function coefficients (OFC) and the effect of the change of the right-hand side (RHS) elements on the optimal value of the objective function, furthermore, the validity ranges of these effects. Depending on how this analysis is performed three types of sensitivities can be defined (Koltai and Terlaky, 1999, 2000):

− *Type I sensitivity*: Type I sensitivity determines those values of some model parameters for which a given *optimal basis* remains optimal. Sensitivity analysis of the optimal basis for the OFC elements determines within which range of an OFC the current optimal basis remains optimal and what is the rate of change (directional derivative) of the optimal objective function value when the OFC changes within this range. In case of the RHS elements the question is, within which range a RHS element can change so that the current optimal basis stays optimal, and what is the rate of change (shadow price) of the optimal objective function value within the determined interval.

Type I sensitivity analysis is implemented in almost all commercial software packages. In case of primal degeneracy, however, several bases may belong to the same optimal solution yielding different ranges and rate of changes for the same parameter to different optimal basis. In case of dual degeneracy many primal optimal solutions, and therefore, many different optimal basis may exist resulting in different intervals and rates of changes. From mathematical point of view the provided information is correct, because the question is the sensitivity of the given optimal basis, but, can be misleading for decision makers, if the given information is not interpreted correctly.

− *Type II sensitivity*: Type II sensitivity determines those values of some model parameters for which *the positive variables in a given primal and dual optimal solution* remain positive, and the zero variables remain zero, i.e. the same activities remain active. More accurately, we have an optimal solution (not necessarily basis solution) $\underline{\mathbf{x}}$ with its support set $\mathrm{supp}(\underline{\mathbf{x}}) = \{\, i \mid x_i > 0 \}$. We are looking for those model parameters, for which an optimal solution (basis or not basis) exists with precisely the same support set. Sensitivity analysis of a given optimal solution for an OFC determines within which range of the OFC an optimal solution with the same support set exists and what is the rate of change (directional derivative) of the optimal objective function value when the OFC changes within this range. In case of the RHS elements the question is, within which range a RHS element can change without the change of the support set of the optimal solution, and what is the rate of change (shadow price) of the optimal objective function value within the determined interval.

Contrary to Type I sensitivity, Type II sensitivity depends on the produced optimal solution, but not on which basis − if any − represents the given optimal solution.

− *Type III sensitivity*: Type III sensitivity determines those values of some model parameters for which *the rate of change of the optimal value function* is the same. Roughly speaking sensitivity (and range) analysis means the analysis of the effects of the change of some problem data, in particular an objective coefficient c_j or right-hand side element b_j. Let us assume that either $c_i+\gamma_i$ or $b_j+\xi_j$ is the perturbation. It is known that the optimal value function is a piecewise linear function of the parameter change (see for example Gal, 1979, Jansen et al., 1997 or Roos, Terlaky and Vial, 1997). In performing Type III sensitivity analysis one wants to determine the rate of the change of the optimal value function and the intervals within which the optimal value function changes linearly.

Type III sensitivity information is independent of the solution obtained, it depends only on the problem data and on which OFC or RHS element is changing.

The calculation and importance of the three different types of sensitivity information depends on the optimum solution produced by the LP solver. Most of the LP solvers used for small and medium size problems are based on some versions of the simplex method and they provide an *optimal basis solution*. Other solvers, typically used for (very) large scale problems are based on interior point methods and they provide an interior (i.e. strictly complementary) optimal solution. To distinguish among the three types of sensitivities is necessary because of the existence of degeneracy. The following cases can be observed:

− When the optimal solution is *neither primal nor dual degenerate*, then all the three types of sensitivities are the same, since there is a unique optimal solution with a unique optimal basis. In this case, the sensitivity analysis output of the available LP solvers provides reliable, useful information for decision-making.

− When the optimal solution is *only primal degenerate* then a unique primal optimal solution exists. Moreover, several optimal bases belong to the same, unique primal optimal solution. In this case Type I and Type II sensitivities may be different since there are different Type I sensitivity information for all the optimal basis. One important case is when the increase and the decrease of a RHS parameter results in different rate of changes, i.e. the optimal value function at the current point is not differentiable. Due to this fact the introduction of the right side and left side shadow prices and their respective sensitivities (Aucamp and Steinberg, 1982) was needed. Type II and Type III sensitivity information for the RHS elements are split into two parts: the left and right side sensitivities. The left and right linearity intervals of the optimal value function provide the Type III information. When the left and right side shadow prices are identical, then only one interval is given. Type II and Type III sensitivity information for an OFC are identical in the case when the solution is only primal degenerate.

− When the optimal solution is *only dual degenerate*, then several different primal optimal basis and non-basis solutions may exist with different support sets, while the dual optimal solution is unique. In this case Type I and Type II sensitivities at each alternative primal optimal basic solution are identical, but Type II sensitivities can be calculated from non-basic solutions as well. Type II sensitivity is interested only in the optimal solutions belonging to the same support sets, therefore, Type III sensitivity may be different from the Type II sensitivities of each optimal solution.

− When the optimum is *both primal and dual degenerate*, then all the three types of sensitivities may be different. In this case each optimal basis of each optimal basis solution may have a different Type I sensitivity information. Optimal solutions with different support sets may have different Type II sensitivities and can be examined at non-basis solutions as well. As it is known, Type III sensitivity information is uniquely determined; it is independent of the optimal solution obtained. Typically, the intervals provided by Type I and Type II sensitivities are subintervals of the Type III sensitivity intervals. The rates of changes produced by Type I and Type II either coincide with Type III information or are useless, their validity (as a sub-differential) is restricted to the current point only.

− In case of large models, solvers based on interior point methods (IPMs) are frequently used. IPM solvers generally provide strictly complementary optimal solutions. In this case Type I sensitivity cannot be asked because, in case of degeneracy, the produced optimal solution is not a basis solution. When one is interested in obtaining an optimal basis solution, a *basis identification* procedure might be applied to produce an optimal basis. Such procedures are implemented in many software packages. Type II and Type III sensitivity information are identical in this case, because the change of the support set of a strictly complementary optimal solution is in one to one correspondence with the linearity intervals of the optimal value function (Roos, Terlaky and Vial, 1997).

An important question is, when the difference between Type II and Type III sensitivities is important for the decision maker. When the decision maker implements an optimal solution then, in many situations, the important information is the sensitivity of the *implemented optimal solution* (Type II sensitivity). For example if an optimal production plan, determined by LP, is already running, then the important question is how the change of certain costs, or the change of a machine capacity influences the *implemented plan*. When the question is, how much a RHS element can be increased with the same consequences, and independently of the possible change of an optimal solution, then it is a Type III sensitivity question. For example if a machine capacity can be increased economically at the calculated shadow price, the decision maker should know how much the capacity can be increased economically in total. It is possible that different production plans (different optimal solutions, especially when optimal basis solutions are implemented) belong to different amount of capacity increases, but all capacity extensions are made at the same marginal benefits.

Type I sensitivity analysis is the classical sensitivity analysis, provided by most LP solvers. Type III sensitivity analysis examines the sensitivity of the decision criteria, and provides the *widest range* for the possible change of the parameter. Type II sensitivity analysis examines the sensitivity of an important property of the optimal operation. In this case the sensitivity range provided by Type III sensitivity is narrowed down with constraints expressing the required property of the optimal solution.

As a summary it can be stated that in case of degeneracy commercial packages do not provide the sensitivity information useful for the decision maker (note that in practice l problems are very frequently degenerate). They give answer to a less ambitious question. They provide information about the interval of a parameter value within which the current *optimal basis* remains optimal, and at what rate the change of the parameter varies the optimal objective function value in that interval (Type I sensitivity). This answer is intimately related to the optimal basis obtained by the simplex solver. In case of degeneracy many different optimal bases exist, thus many different ranges and rates of changes might be obtained. To obtain the true Type III sensitivity information about the change of the value of the OFC and RHS elements one needs extra effort. In fact one has to solve some subsidiary LP=s for determining linearity intervals, and left and right derivatives of the optimal value function (see Chapter 2.4).

2.3 Graphical illustration of the problem of sensitivity analysis

When the LP problem has no more than two variables then the solution space and all the information concerning the optimum and its sensitivity, can be represented in a two dimensional space. The following problem will be our prototype problem (Koltai and Terlaky, 2000),

$$\begin{array}{lcccccc} \max & 12x_1 & + & 10x_2 & & \\ \hline \text{C1} & x_1 & + & x_2 & \leq & 600 \\ \text{C2} & 2x_1 & + & x_2 & \leq & 1000 \\ \text{L1} & x_1 & & & \leq & 400 \\ \text{L2} & & & x_2 & \leq & 500 \\ & \multicolumn{5}{l}{x_1 \geq 0,\ x_2 \geq 0} \end{array} \tag{2.5}$$

The feasible set and the solution of problem (2.5) can be seen on Figure 2.1.

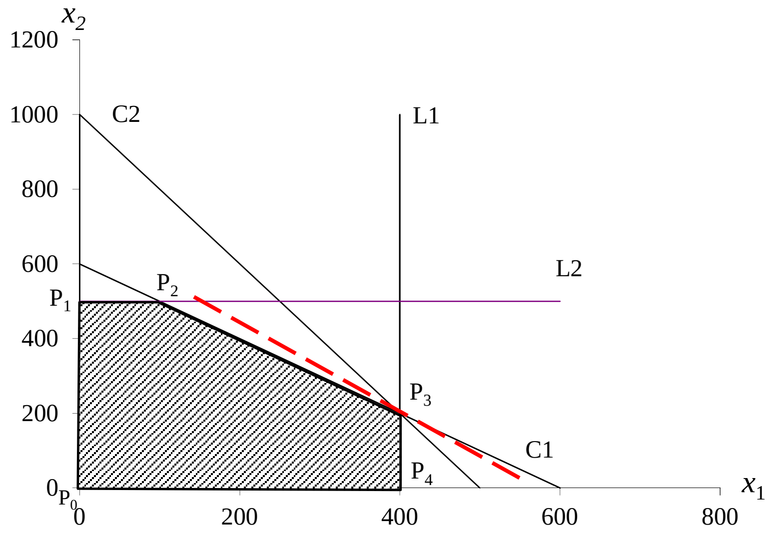

Figure 2.1 Graphical illustration of the prototype problem

The two constraints (C1 and C2) and the upper bounds on x_1 and x_2 (L1 and L2) are represented as half spaces. The boundary of these spaces with the corresponding labels is depicted on the figure. The intersection of these half spaces is represented as a shaded area, which contains all the primal feasible solutions. The objective function (iso-profit line) is drawn as a straight dashed line. The objective function touches the shaded area at point P_3, therefore the unique optimal solution is at x_1=400 and x_2=200.

In order to transform problem (2.5) into the standard form, indicated by problem (2.1), *slack variables* (denoted by s_i, i=1,...,4) are introduced for all the constraints, and the objective function is changed to have a minimization problem. The problem in standard form is as follows,

$$
\begin{array}{llllllllll}
\min & -12x_1 & - & 10x_2 & & & & & & \\
\hline
\text{C1:} & x_1 & + & x_2 & + s_1 & & & & = & 600 \\
\text{C2:} & 2x_1 & + & x_2 & & + s_2 & & & = & 1000 \\
\text{L1:} & x_1 & & & & & + s_3 & & = & 400 \\
\text{L2:} & & & x_2 & & & & + s_4 & = & 500 \\
\multicolumn{10}{l}{x_1 \geq 0, x_2 \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0, s_4 \geq 0}
\end{array}
\tag{2.6}
$$

Problem (2.6) shows that **A** is a 4x6 matrix with rank equal to 4. The values of the slack variables at P_3 are the following,

s_1=0; s_2=0; s_3=0; s_4=300.

Since at P3 there are three nonzero variables (x_1, x_2 and s_4) and the rank of matrix **A** is 4, the optimal solution is degenerate. This can be seen in Figure 2.1. The point P3 is the intersection of three lines (C1, C2 and L1). Two lines would be enough to determine the location of a point in a two dimensional space, therefore P3 is over determined. Even if we remove any one of C2, or L1, the point P3 remains the only optimal solution. This over determination of the optimal point is a graphical illustration of primal degeneracy.

Let us see the consequences of degeneracy on sensitivity analysis. The shadow prices and the corresponding validity ranges for the optimal solution, calculated with the help of Figure 2.1, are given in Table 2.2. The change of a RHS element is represented by a parallel shift of the corresponding line in Figure 2.1.

Table 2.2 Shadow prices and validity ranges of the optimal values

Dual variable	Current RHS value	Left side shadow price	Validity range		Right side shadow price	Validity range	
			LL	UL		LL	UL
y_{C1}	600	10	400	600	8	600	750
y_{C2}	1000	2	700	1000	0	1000	∞
y_{L1}	400	2	100	400	0	400	∞
y_{L2}	500	0	200	500	0	500	∞

If the RHS of any of these constraints are decreased, then the left side shadow prices are obtained for each constraint respectively (column three of Table 2.2). The optimal point P_3 is at the intersection of constraints C1, C2 and L1. The decrease of any of the RHS of these constraints results in the move of the optimum point, P_3, which consequently changes the objective function value as well. Since the change of the RHS of L2 does not affect the location of P_3 its shadow price is zero. C1 can be moved to P_4, C2 and L1 can be moved to P_2 with the same shadow price value. L2 can be moved to P_3 without affecting the objective function value. The corresponding lower limits (LL) are given in the fourth column of Table 2.2. In case of left side shadow prices the upper limits (UL) are equal to the current values of the RHS elements (fifth column of Table 2.2).

If the RHS of any of these constraints are increased, then the right side shadow prices are obtained for each constraint, respectively (column six of Table 2.2). In case of constraints C2 and L2 the increase of the right-hand side values do not affect the location of the optimum point, because C1 and either C2 or L1 fixes its place. Therefore the corresponding right side shadow prices are equal to zero. When the right-hand side of C1 is increased, then the optimum point will stay at the intersection of C1 and C2 and the shadow price will be equal to 8. Since L2 does not affect the location of P_3 its shadow price is also zero. In case of right side shadow prices the lower limits (LL) are equal to the current values of the RHS elements, while the upper limits (UL) are determined by the geometrical properties of the solution space. When C1 is moved upward the intersection of C1 and C2 (P_3) moves upward as well. When P_3 reaches L2, then the move of C1 does not affect the location of P_3, and the shadow price turns into zero. The RHS value at this point is the UL of the sensitivity range, and it is equal to 750. The UL of all the other constrains are equal to infinity.

Table 2.3 shows the shadow prices and their validity ranges found by the STORM computer package (Emmons at al, 2001) at the optimal basis B_1={1, 2, 3, 6}. It can be seen that at this basis the left side shadow prices and validity ranges were provided for constraints C1 and L1, and the right side shadow price and validity range was found for constraint C2.

Table 2.3 Shadow prices and validity ranges at the optimal bases B_1

Dual variable	Current RHS value	Left side shadow price	Validity range	
			LL	UL
y_{C1}	600	10	400	600
y_{C2}	1000	0	1000	∞
y_{L1}	400	2	100	400
y_{L2}	500	0	200	∞

Table 2.4 contains the shadow prices and their validity ranges found at the optimal basis B_2={1, 2, 5, 6} . At this basis the right side shadow prices and validity ranges were provided for constraints C1 and L1, and the left side shadow price and validity range was found for constraint C2. The left and right side shadow prices for constraint L1 are identical, and its correct value and validity range was found in both optimal bases as the last rows of Table 2.3 and 2.4 shows.

Table 2.4 Shadow prices and validity ranges at the optimal bases B_2

Dual variable	Current RHS value	Left side shadow price	Validity range	
			LL	UL
y_{C1}	600	8	600	750
y_{C2}	1000	2	700	1000
y_{L1}	400	0	400	∞
y_{L2}	500	0	200	∞

The reason of the differences of Table 2.2, 2.3 and 2.4 can be explained if we look at the mathematical interpretation of degeneracy. Every corner point of the shaded area of Figure 2.1 can be represented by one or more basis. The corner point which is over determined, i.e. defined by the intersection of more than two lines, represents more than one basis. Depending on which two lines are taken to define this point different basis is considered, that is, different sets of B in (2.3) may lead to the same basis solution. This is the case at P_3, where Table 2.3 was calculated with the help of a basis containing columns 1, 2, 4 and 6, and Table 2.4 was calculated with the help of a basis containing columns 1, 2, 5 and 6 of problem (2.6).

The main problem of RHS sensitivities in the prototype problem is that in case of a degenerate primal optimal solution the dual problem has no unique solution. Different basis belonging to the same optimal solution provide different shadow prices and validity ranges. Table 2.3 and Table 2.4 show that the results provided by the two optimal basis are mixtures of the left side, right side and full shadow prices and validity ranges. The complete Type III information, similar to Table 2.2, is not given at any of the basis. It depends on the computer code at which basis, among the many optimum ones, the program stops. Different commercially available software may report different RHS sensitivities for the same problem (Jensen et al., 1997). All these results are correct mathematically, because they describe the validity of an *optimal basis* (Type I sensitivity), but not useful for decision-making, because these are not reflecting the validity of the *positivity status of the decision variables at optimality* (Type II sensitivity), or not characterizes the *validity range of the left/right marginal values* (Type III sensitivity). The correct RHS information, which refers to the rate of change of the optimal objective value, and the range where these rates are valid are given in Table 2.2.

It can be seen in Table 2.2 that most of the right side shadow prices are zero. An interesting question is how the optimal objective function value can be increased by the simultaneous increase of those RHS elements which have a zero shadow price. This question is equivalent to the problem of increasing the capacity of bottleneck resources of production systems. Figure 2.1 shows that the RHS of C1 can be increased alone, but the RHS of C2 and L1 need to be increased simultaneously. This information is summarized in Table 2.5.

Table 2.5 Increase of the objective function by a unit of the increment of RHS elements

RHS elements	Rate of change of the objective function	Validity range
C1	10	$400 \leq \Delta b_{C1} \leq 600$
C1, L1	2	$400 \leq \Delta b_{L1} \leq 600$ $\Delta b_{C2=}\Delta b_{L1}$

The optimum value of the objective function increases by 10 if the RHS of C1 is increased by one unit. This is true within the interval [400, 600]. When the RHS of C2 and L1 are simultaneously increased by one unit, the change of the objective function value is 2 and the validity range is a line segment in a two dimensional space, given in the last window of Table 2.5.

Since the objective function coefficient sensitivity of the primal problem is the same as the RHS sensitivity of the dual problem, all what was said for the RHS is valid for the objective function coefficients as well. Graphically, the change of an OFC can be represented by the change of the slope of the line of the objective function. In Figure 2.1 the optimal solution of problem (2.6) is P_3 as long as the objective function line stays between L1 and C1. The corresponding OFC sensitivities are given in Table 2.6. These data coincide with the sensitivities provided by the STORM computer package when the optimum was calculated at the basis B_1.

Table 2.6 Objective function coefficient sensitivities and rate of changes at the optimal bases B_1

Dual variable	Current RHS value	Rate of changes	Validity range	
			LL	UL
c_1	12	400	10	∞
c_2	10	200	0	12

The results provided at the basis B_2 are given in Table 2.7. The intervals obtained in this case are subsets of the correct sensitivity ranges. The last columns of Table 2.6 and 2.7 show the rate of changes of the optimum value function. The identical rate of changes of the respective coefficients in both optimal basis B_1 and B_2 indicate that the optimal solution is not dual degenerate. This is also clear from Figure 2.1 since the optimal solution is unique.

Table 2.7 Objective function coefficient sensitivities and rate of changes at the optimal bases B_2

Dual variable	Current RHS value	Rate of changes	Validity range	
			LL	UL
c_1	12	400	10	20
c_2	10	200	6	12

Since the optimum at P_3 is not dual degenerate Type II and Type III sensitivities for the OFC are the same, and are given in Table 2.6. The Type III sensitivity of the RHS elements are given in Table 2.2, in which for y_{C1}, y_{C2}, y_{L1}, the left and right side sensitivities are Type III information for two different linearity intervals. For y_{L2}, the Type II and Type III sensitivity information are identical.

Figure 2.2 illustrates a slight modification of the sample problem. A new constraint (C3: $x_1-2x_2\leq200$) is added to the problem and the objective function is also modified ($min[-12x_1-0x_2]$).

In this case the optimal objective function coincides with constraint L1, and all the points in the interval [P_3, P_4] are optimal. Consequently, all bases at P_3 and the basis at P_4 are optimal and the optimal solution is both primal and dual degenerate, and we expect different Type I, Type II and Type III sensitivities.

Let us consider now the shadow price and sensitivity range of the RHS of constraint L1. It can be seen that as long as L1 increases or decreases the shaded area the shadow price is equal to 12. This is true between points P_0 and P' and corresponds to the RHS values of L1 in the interval [0, 440], which is the Type III sensitivity information for the RHS of L1. If, however, the problem is solved by a computer code of the simplex method, then depending on the basis found by the program, the following intervals can be obtained: [100, 400], [200, 440], [400, 400], [400, 440], that is, there are four different Type I sensitivities. In this modified example the left and right shadow prices are equal. In the case of the optimal solution at P_3 the Type II sensitivity range is [400, 400], and the Type II sensitivity rang at P_4 is [200, 440].

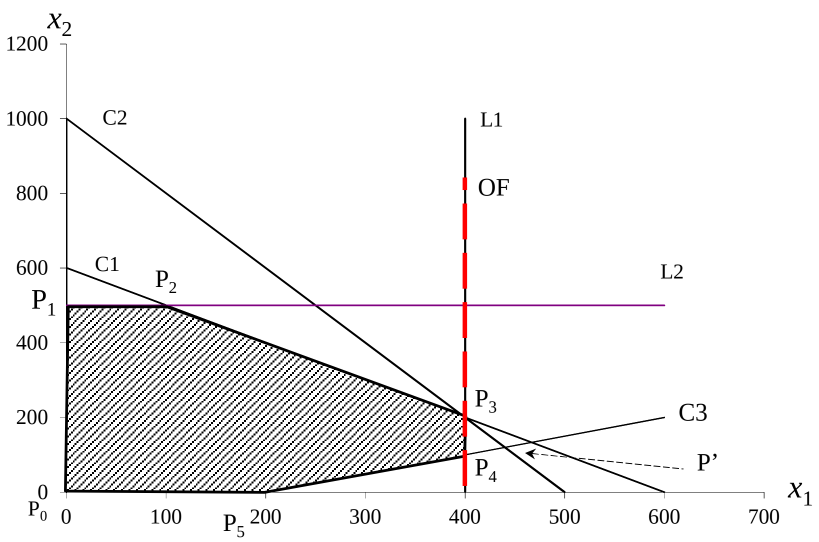

Figure 2.2 Graphical illustration of the modified prototype problem

As a conclusion, it can be said that the sensitivity results based on an optimal basis characterize correctly the optimality of that basis. The graphical representation, however, shows that several results are either incomplete or irrelevant from the point of view of the information required by a decision maker. The next chapter shows, how Type III sensitivity analysis results can be obtained by solving several additional LP problems.

2.4 A practical approach to sensitivity analysis under degeneracy

To get the Type III validity ranges for the primal and for the dual optima additional LP problems must be solved. These additional LP problems are summarized in Table 2.8 (Koltai and Tatay, 2011).

Table 2.8 Summary of additional LP problems for sensitivity analysis

	Maximal decrease	Maximal increase
Sensitivity analysis of objective function coefficients (OFC)	$\begin{aligned} &\mathbf{A}^{\mathrm{T}}\underline{\mathbf{y}} \geq \underline{\mathbf{c}} + \gamma_i \underline{\mathbf{e}}_i \\ &\underline{\mathbf{b}}^{\mathrm{T}}\underline{\mathbf{y}} = OF^* + \gamma_i x_i^* \\ &\gamma_i \leq 0 \\ &\text{Min}\ (\gamma_i); \end{aligned} \quad (2.7)$ Optimal solution: γ_i^-	$\begin{aligned} &\mathbf{A}^{\mathrm{T}}\underline{\mathbf{y}} \geq \underline{\mathbf{c}} + \gamma_i \underline{\mathbf{e}}_i \\ &\underline{\mathbf{b}}^{\mathrm{T}}\underline{\mathbf{y}} = OF^* + \gamma_i x_i^* \\ &\gamma_i \geq 0 \\ &\text{Max}\ (\gamma_i); \end{aligned} \quad (2.8)$ Optimal solution: γ_i^+
Sensitivity analysis of the left shadow price ($\delta<0$) (y_j^-)	$\begin{aligned} &\mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}} + \delta\underline{\mathbf{e}}_j + \xi_j \underline{\mathbf{e}}_j \\ &\underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}} = OF^* + \xi_j y_j^* \\ &\xi_j \leq 0 \\ &\text{Min}(\xi_j) \end{aligned} \quad (2.9)$ Optimal solution: $n\xi_j^-$	$\begin{aligned} &\mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}} + \delta\underline{\mathbf{e}}_j + \xi_j \underline{\mathbf{e}}_j \\ &\underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}} = OF^* + \xi_j y_j^* \\ &\xi_j \geq 0 \\ &\text{Max}(\xi_j) \end{aligned} \quad (2.10)$ Optimal solution: $n\xi_j^+$
Sensitivity analysis of the right shadow price ($\delta>0$) (y_j^+)	$\begin{aligned} &\mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}} + \delta\underline{\mathbf{e}}_j + \xi_j \underline{\mathbf{e}}_j \\ &\underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}} = OF^* + \xi_j y_j^* \\ &\xi_j \leq 0 \\ &\text{Min}(\xi_j) \end{aligned} \quad (2.11)$ Optimal solution: $p\xi_j^-$	$\begin{aligned} &\mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}} + \delta\underline{\mathbf{e}}_j + \xi_j \underline{\mathbf{e}}_j \\ &\underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}} = OF^* + \xi_j y_j^* \\ &\xi_j \geq 0 \\ &\text{Max}(\xi_j) \end{aligned} \quad (2.12)$ Optimal solution: $p\xi_j^+$

Ranges can be expressed by determining the maximal decrease and the maximal increase of a parameter. The maximal decreases are determined by the LP problems of the second column of Table 2.8, while the maximal increases are determined by the third column of Table 2.8. When validity ranges of the OFCs of the primal problem are determined the additional LP problems are based on the *dual* problem (see the first row of Table 2.8). The OFCs of the original primal problem are the RHS elements of the dual problem. If the validity range of the OFC of a decision variable is examined a γ_i variable can be used to express the change of c_i, that is, the new OFC is equal to $c_i+\gamma_i$. It is assumed that within the validity range the optimal solution of the primal problem is always the same. Therefore, the original optimal objective function value (OF^*) is changed exclusively by the change of c_i. This change is equal to $\gamma_i x_i^*$, where x_i^* is the original optimal value of x_i. Adding this condition to the dual conditions, and minimizing a non-positive γ_i we get LP problem (2.7) for finding the maximal decrease of c_i. Adding this condition to the dual conditions, and maximizing a non-negative γ_i we get LP problem (2.8) for finding the maximal increase of c_i. For I variables $2I$ additional

LP problems must be solved to get the proper validity ranges for all OFCs.

When validity ranges of the RHS elements of the primal problem are determined the additional LP problems are based on the *primal* problem (see the second and third rows of Table 2.8). In this case a ξ_j variable is used to express the change of b_j, that is, the new RHS value is equal to $b_j + \xi_j$. It is assumed that within the validity range the optimal solution of the dual problem is always the same. Therefore, the original optimal objective function value (OF^*) is changed exclusively by the change of b_j. This change is equal to $\xi_j y_j^*$, where y_i^* is the original optimal value of y_i. Adding this condition to the primal conditions, and minimizing a non-positive ξ_j we get an LP problem for finding the maximal decrease of b_j. Adding this condition to the primal conditions, and maximizing a non-negative ξ_j we get an LP problem for finding the maximal increase of b_j.

Under degeneracy the effect of increase and the effect of decrease of the RHS elements can be different. Since information about the marginal increase and about the marginal decrease of each RHS element are necessary, a $\delta>0$ perturbation is used to get information about the increase, and a $\delta<0$ perturbation is used to get information about the decrease. That is, the original problem must be solved with a positive and with a negative perturbation as well, and with each perturbation a maximal decrease (LP problems (2.9) and (2.11)) and a maximal increase (LP problems (2.10) and (2.12)) must be determined. In these problems the new value of a RHS element is equal to $b_j+\delta+\xi_j$. For J RHS elements $6J$ additional LP problems must be solved to get the proper validity ranges for all RHS parameters.

Altogether, in case of I variables and J constraints $2I+6J$ additional LP problems must be solved if range information for each OFC and RHS element of the original problem is required. A possible implementation of the suggested calculations is illustrated in Figure 2.3. The LP problems are solved with the Lingo mathematical programming software (Schrage, 2003). The successive solution of the additional LP problems is controlled by Visual Basic Application (VBA) implemented in Excel. The data and the results are stored and presented in Excel.

First the basic LP problem is solved, and the primal optimum ($\mathbf{x}^*$) and dual optimum ($\mathbf{y}^*$), furthermore the optimal value of the objective function (OF^*) is stored. Next, two FOR cycles must be run. The first cycle (i=1,…,I) is used for solving LP problems (2.7) and (2.8) to get the validity ranges of OFCs, that is, to obtain the γ_i^-, and γ_i^+ values. The second cycle (j=1,…, J) is used to get the left and right shadow prices (y_j^- and y_j^+), and the corresponding sensitivity ranges ($n\xi_j^-$, $n\xi_j^+$, $p\xi_j^-$, $n\xi_j^+$). In those cases, when the left and right shadow prices are equal (y_j^- $=y_j^+$), it is not necessary to solve the extended dual problems with perturbations, that is, $\delta=0$. The left and right shadow prices are equal, if neither the maximal decrease (ξ_j^-) nor the maximal increase (ξ_j^+) is equal to zero in the sensitivity report originally provided by Lingo. In this case the validity range embraces the original value of the corresponding RHS element. Therefore, first the existence of two sided shadow price is checked ($\xi_j^-=0$ or $\xi_j^+=0$), and next LP problems (2.9), (2.10), (2.11) and (2.12) are solved with the proper value of δ. If for K number of RHS elements $y_j^-=y_j^+$ then the total number of LP problems to solve is reduced by $4K$.

The presented method determines the Type III sensitivity ranges. The method, however, can be used for calculating Type II sensitivity ranges as well, but in this case the original problem must be completed with the conditions expressing the required property of the optimal solution.

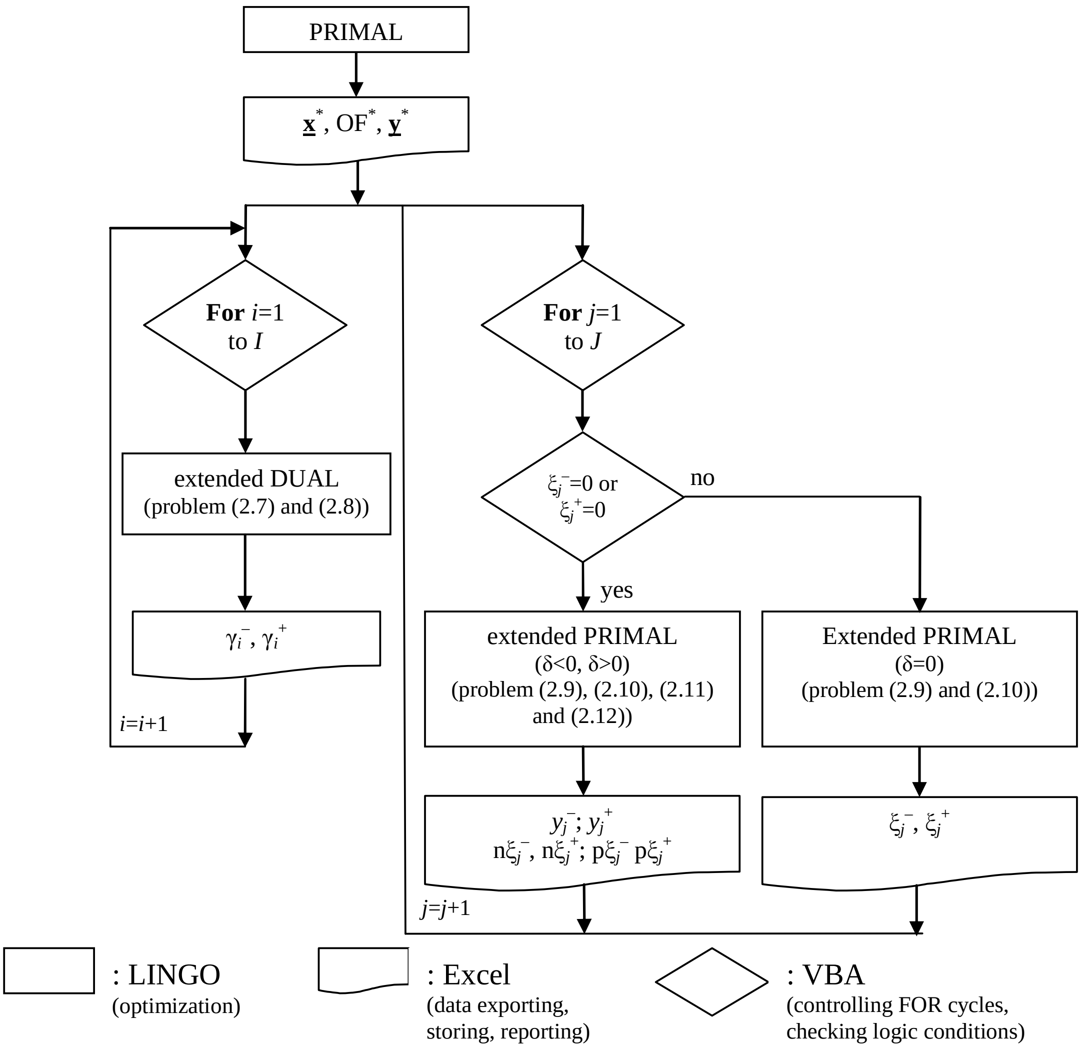

Figure 2.3 Implementation of sensitivity analysis

2.5 Calculation of Type III sensitivity results of a production planning example

In this chapter a production planning example is used to illustrate the misleading results of traditional sensitivity analysis results. On the one hand, the selected sample problem is small enough to get the correct sensitivity results based on simple reasoning. On the other hand the sample problem have more than two variables, consequently graphical solution (similar to the sample problem in Chapter 2.3) cannot be possible (Koltai and Terlaky, 2009).

The basic data of the production planning example are summarized in Table 2.9. The production quantity of two products (P1 and P2) in two production periods (T1 and T2) should be determined. The demand for P1 is zero in the first period and 200 units in the second period. The demand for P2 is 100 units in both periods. The production cost is the same ($10 per unit) for both products in T1, and $25 per unit for P1, and $20 per unit for P2 in T2. The inventory holding cost is the same in all periods for all products ($5 per unit). There is capacity to produce 300 units in the first period, and to produce 200 units in the second period. The inventory cannot exceed 200 units in any of the two periods.

Table 2.9 Data of the sample production planning model

Model parameters		Period 1 (T1)	Period 1 (T2)
Demand (units/period) (D_{it})	Prod 1 (P1)	0	200
	Prod 2 (P2)	100	100
Production cost (\$/units) (p_{it})	Prod 1 (P1)	10	25
	Prod 2 (P2)	10	20
Inventory cost (\$/units) (h_{it})	Prod 1 (P1)	5	5
	Prod 2 (P2)	5	5
Production capacity (units/period) (K_t)		300	200
Inventory capacity (units/period) (INV_t)		200	200

Using the data of Table 2.9 a production planning model with eight variables, four equalities, and four inequalities are obtained. Completing this model with four slack variables the following LP model, written in the standard form, is obtained:

$$
\begin{array}{rcccccccccc}
\text{Min:} & 10x_{1,1} & +10x_{2,1} & +25x_{1,2} & +20x_{2,2} & +5I_{1,1} & +5I_{2,1} & +5I_{1,2} & +5I_{2,2} & & \\
\hline
\text{Dem(P1\_T1):} & x_{1,1} & & & & -I_{1,1} & & & & = & 0 \\
\text{Dem(P2\_T1):} & & x_{2,1} & & & & -I_{2,1} & & & = & 100 \\
\text{Dem(P1\_T2):} & & & x_{1,2} & & +I_{1,1} & & -I_{1,2} & & = & 200 \\
\text{Dem(P2\_T2):} & & & & x_{2,2} & & +I_{2,1} & & -I_{2,2} & = & 100 \\
\hdashline
\text{Prod(T1):} & x_{1,1} & +x_{2,1} & & & & & & & \leq & 300 \\
\text{Prod(T2):} & & & x_{1,2} & +x_{2,2} & & & & & \leq & 200 \\
\hdashline
\text{Inv(T1):} & & & & & I_{1,1} & +I_{2,1} & & & \leq & 200 \\
\text{Inv(T2):} & & & & & & & I_{1,2} & +I_{2,2} & \leq & 200 \\
& \multicolumn{8}{l}{x_{1,1},\ x_{2,1},\ x_{1,2},\ x_{2,2},\ I_{1,1},\ I_{2,1},\ I_{1,2},\ I_{2,2} \geq 0} & &
\end{array}
\tag{2.13}
$$

This small size problem can be solved with any LP software, but the optimal solution can be found easily by simple reasoning as well. The data show that there is a considerable difference between the production costs in T1 and in T2. It would be cheaper to produce all the products in T1. The products demanded in T1 are produced first. If there is free capacity, products demanded in T2 can be produced in T1 as well. After producing 100 units of P2, there is free production capacity, therefore the production of P1 demanded in T2 is scheduled for T1 as well. The planned 200 units of P1 and 100 units of P2 are exactly equal to the production capacity of the first period. There is also enough inventory capacity to store the 200 units of P1 until the second period. Since there is no more free production capacity, the second period demand of P2 cannot be produced in T1, although it would be advantageous financially. The optimal solution, therefore, is the following,

$x_{1,1}$=200; $x_{2,1}$=100; $x_{1,2}$=0; $x_{2,2}$=100;
$I_{1,1}$=200; $I_{2,1}$=0; $I_{1,2}$=0; $I_{2,2}$=0;
s_1=0; s_2=100; s_3=0; s_1=200.

There are six non-zero values and the rank of the matrix of problem (2.13) is 8, therefore the solution is primal degenerate, and since there is no alternative optimum, the solution is not dual degenerate. In this case Type I and Type II (which is in this case is equivalent to Type III) sensitivities are the relevant information.

Solving the model with the STORM computer program, the same solution is obtained. However, the Type I sensitivity results depend on which basis is found by the software. Koltai and Terlaky (2000) provided a detailed analysis of the results obtained in two different bases. In the following, Type I sensitivity results obtained with STORM, and Type II (Type III) results obtained by the suggested method in the previous section will be compared.

2.5.1 Sensitivity analysis of the right-hand side (RHS) elements

Table 2.10 summarizes the results of sensitivity analysis of the RHS elements. The first part of the table contains the shadow prices and the ranges provided by the linear programing solver of the STORM software. The second part of the table contains the left and right shadow prices and the corresponding linearity intervals calculated by the suggested method. The linearity interval is defined by the maximal decrease (dec.) and the maximal increase (inc.) of the original RHS value. When the right and left shadow prices are identical, the single shadow price is given in the y_j^- (y_j) column.

Table 2.10 Sensitivity analysis of the RHS elements

RHS parameter	j	Original RHS value	STORM			Suggested method					
			y_j	dec.	inc.	y_j^-(y_j)	nξ_j^-	nξ_j^+	y_j^+	pξ_j^-	pξ_j^+
$D_{1,1}$	1	0	15	0	0	10	-200	0	20	0	100
$D_{2,1}$	2	100	15	0	0	10	-100	0	20	0	100
$D_{1,2}$	3	200	20	-100	0	20	-100	0	25	0	100
$D_{2,2}$	4	100	20	-100	100	20	-100	100	-	-	-
C_1	5	300	-5	0	0	-10	-100	0	0	0	∞
C_2	6	200	0	-100	∞	0	-100	∞	-	-	-
W_1	7	200	0	0	∞	-10	-100	0	0	0	∞
W_2	8	200	0	-200	∞	0	-200	∞	-	-	-

Note that a shadow price is negative if the direction of change of the objective function is opposite to the direction of change of the respective RHS element. For example, if $D_{2,1}$ *decreases* by one unit, the optimal production cost *decreases* by \$10 (y_2^-=10). If C_1 *decreases* by one unit, the cost of the new optimal production plan *increases* by \$10 (y_5^-=−10).

The results show that STORM provided the correct shadow prices and ranges for $D_{2,2}$, C_2, and W_2. For $D_{1,1}$, $D_{2,1}$ and C_1 STORM provides a shadow price which has no managerial significance, but the suggested method provided the correct left and right shadow prices. Finally, for $D_{1,2}$, only the left shadow price and for W_1 only the right shadow price is found by STORM.

In the following some of the differences between the two different sensitivity information are analyzed and explained.

− *The analysis of the shadow price of* $D_{1,1}$:

The production cost of P1 in T1 is \$10. If one extra unit should be produced in T1, then the production of another unit, which was originally produced in T1, but demanded in T2, has to be produced in T2 because of production capacity limitations. The shift of the production of this one unit from T1 to T2 will increase production cost from \$10 to \$25, and at the same time eliminates the \$5 inventory holding cost. The total cost of the shift is therefore \$10 for every unit (25-10-5). The result is the sum of the production cost of the new product (\$10) and the cost of shift (\$10), which yields a \$20 increase of the objective function for every unit of new P1 produced for T1. There are 100 units free capacity in T2 to reschedule P1. Therefore, the \$20 shadow price is valid as long as the new demand for P1 in the first period is less than 100 units.

The sensitivity information provided by STORM for $D_{1,1}$ is incorrect. The \$15 shadow price, shown in the second row of Table 2.10, is not the deduced value. The validity range shows that we are at a break point of the piecewise linear optimal value function. \$15 is just a sub-differential, the true right and left derivatives might be different from \$15.

− *The analysis of the shadow price of* $D_{1,2}$:

Producing one unit less from P1 for T2 will result in the savings of \$10 production cost, and in the savings of $5 inventory holding cost because all P1 is produced in T1. Since there will be free capacity in T1, one unit of P2 produced for T2 can be shifted to T1, saving by this way another $5 (20-10-5). This is altogether $20 per unit. Since 100 units of P1 can be substituted by 100 units of P2, this $20 left side shadow price is valid as long as the demand decreases from 200 units to 100 units.

When the demand for P1 increases by one unit in T2, this extra quantity should be produced in T2 because in T1 there is no free production and inventory capacity. This will result in a $25 increase of the objective function for every unit (right side shadow price). Production can be increased up to 100 units as a consequence of the 100 units free production capacity in T2.

STORM has found the left side shadow price and validity range (see Table 2.10).

− *The analysis of production capacity increase in T1 (right shadow price of* C_1):

Since the 200 units of P1 are produced for T2, these products must be kept in the warehouse. Inventory constraints indicate that there is no more space to store, therefore the demand of P2 in the second period cannot be produced earlier, although, financially it would be advantageous. Therefore, no matter how much the production capacity is increased in T1, it will not influence the objective function; the shadow price is zero.

The zero shadow price is not found by STORM (see Table 2.10).

− *The analysis of production capacity decrease in T1 (left shadow price of* C_1):

Since production capacity is fully utilized in T1, the lost capacity will decrease the production of P1. If P1 is produced in T2 production cost increases by $15 (from $10 to $25) but inventory cost disappears ($5). The objective function therefore increases by $10 per unit. No more than 100 units of production can be shifted to T2 because of capacity limitations; therefore the $10 is valid when production capacity does not decrease below 200 units.

The shadow price found by STORM is incorrect (see Table 2.10). The validity range indicates that this is true just in the very near neighborhood of the current capacity, but from practical point of view this information is irrelevant.

2.5.1 Sensitivity analysis of the objective function coefficients (OFCs)

Table 2.11 summarizes the results of sensitivity analysis of the OFCs. The first part of the table contains the OFC ranges provided by STORM. The second part of the table contains the OFC ranges calculated by the suggested method. The ranges are defined by the maximal decrease (dec.) and the maximal increase (inc.) of the original OFC value. It can be seen that the correct ranges are found by STORM only for $p_{1,2}$ and for $h_{2,2}$. In all other cases the range found by the suggested method is larger.

Table 2.11 Sensitivity analysis of the OFCs

OFC	i	Original OFC	STORM decrease	STORM increase	Suggested method γ_i^-	Suggested method γ_i^+
$p_{1,1}$	1	10	-25	5	$-\infty$	5
$p_{2,1}$	2	10	-5	5	-5	∞
$p_{1,2}$	3	25	-5	∞	-5	∞
$p_{2,2}$	4	20	-5	5	-25	5
$h_{1,1}$	5	5	-25	5	$-\infty$	5
$h_{2,1}$	6	5	-5	5	-5	∞
$h_{1,2}$	7	5	-25	∞	-30	∞
$h_{2,2}$	8	5	-25	∞	-25	∞

In the following some of the differences between the two different sensitivity information are analyzed and explained.

− *The analysis of the decrease of* $p_{2,2}$:

At the current production cost it would be better to move the production of P2 to T1, but the production capacity is fully utilized. If production cost of P2 decreases in T2 then the possible benefit by producing P2 in T1 decreases as well. When production cost drops to $15, the production cost in T2 will be equal to the production plus inventory cost in T1, therefore it is not advantageous any more shifting the production to T1. Since the production was not moved to T1 because of the capacity constraints, this $15 is just a symbolic value. This value indicates that if we could change the plan it would be advantageous to do it as long as the production cost is higher than $15. But the correct answer to the question is, that no matter how much the production cost of P2 in T2 decreases, the production plan will stay optimal.

Based on the information of STORM we may conclude that the production plan should be changed when the cost decreases below $15, because the fifth row of Table 2.11 indicates a $15 ($20–$5) lower limit for the validity of the optimal production plan.

− *The analysis of the increase of* $h_{2,1}$:

Since only the demand of P2 in T1 is scheduled for production in T1, there is no inventory of P2 in the optimal production plan. It means that no matter how much the inventory holding cost of P2 increases it will not influence the optimal plan. It is true, however, that reaching $10 has a symbolic importance. Above this level it will not be worth to move the production of all the P2 to T1 even if it were possible, because the high production cost in the second period will still be better than the low production cost in T1 plus the increased inventory cost.

Based on the information provided by STORM we may conclude that the production plan should be changed when the cost increases above $10 ($5+$5), because the seventh row of Table 2.11 indicates a $10 upper limit for the validity of the optimal production plan.

2.6 Decreasing the number of additional LP problems

According to Figure 2.3, for an LP problem with I variables and J constraints $2I+6J$ additional LP problems must be solved. This can be a very high number in case of large problems. Some of the additional LP problems, however, are unnecessary to solve. There are mathematical and managerial possibilities for the reduction of the number of LP problems.

When the optimal solution is analyzed *mathematically*, we may conclude, that for some RHS elements the right and left shadow prices are identical. In these cases the perturbation of the RHS elements is not necessary; therefore instead of 6 additional LP problems only 2 must be solved (for the maximal decrease and for the maximal increase). To filter those RHS elements, for which perturbation is not necessary, the sensitivity results referring to the optimality of a basis – and provided automatically by the LP solvers – must be analyzed. If neither end of the validity range of a shadow price is 0; that is, the maximal increase and the maximal decrease of the RHS element is non-zero; then the left and right shadow prices are identical. In Table 2.10 it can be seen that for 3 RHS elements perturbations are not necessary. This way the solution of 12 additional LP problems can be saved.

When the optimal solution is analyzed from *managerial* point of view the solution of several additional LP problems can be ignored. The decision-making situation strongly determines which OFCs and RHS elements must be analyzed in details, therefore, general rules cannot be given. Here are some examples for managerial filtering of the sensitivity analysis information:

– Frequently, managers are interested only in the shadow price of some critical bottleneck resources. In this case additional LP problems must be solved just for the capacity constraints of these resources.

– Sometimes either the increase or the decrease of an RHS element is not a feasible alternative. For example, for technological reasons a manufacturing capacity cannot be increased, or a production limit cannot be decreased. In this case either the positive or the negative perturbation can be omitted.

– Frequently we use such variables in the model formulations for which the OFC sensitivity results have no practical significance. In these cases the solution of the additional LP problems can also be saved.

– We saw in Tables 2.10 and 2.11 that the sensitivity ranges provided by STORM for OFCs and for RHS elements are frequently narrower, than the ranges given by the suggested method. A narrower range, however, can be large enough for management decision-making. In this case it is not necessary to solve the additional LP problems.

Finally, it is important to stress the advantage of sensitivity analysis over parametric analysis. In case of parametric analysis the change of the objective function is calculated for several different values of a parameter by solving the LP problem repeatedly with different parameter values. In case of sensitivity analysis the change of the objective function is known for all parameter values within the validity range. That is, if the exact value of a parameter change is not known yet, sensitivity analysis provides more information about the effect of the possible changes than parametric analysis.

2.7 Conclusions of Chapter 2

The main objective of Chapter 2 is to show that sensitivity analysis results provided by the generally used LP solvers and sensitivity analysis results required for decision-making are different. The sensitivity information given by the simplex based LP software tell the user in what range some basic parameters can vary to keep the obtained optimal basis optimal, and how the current optimal basis solution changes as a function of these parameters. When the optimal solution of an LP model is degenerate then there are several optimal bases providing the same optimal value, and possibly all optimal bases provide different sensitivity results. These results are mathematically correct, but their information content either incomplete or irrelevant from management decision point of view. Management wants to know either the sensitivity information concerning activities in an optimal solution (Type II sensitivity), or the sensitivity information concerning the objective function (Type III sensitivity).

Both the graphical solution of the small LP model and the logical solution of the production planning model have illustrated the existence of the three types of sensitivities. Consequently, users should be careful when sensitivity results of an LP software are used for management decisions. Almost all practical size problems are degenerate, and the sensitivity information depends on the basis found by the computer program. Different software may give different result to the same model. Sometimes the goodness of the sensitivity output can be checked by simple logic, but in most of the cases there is no direct way of evaluating the results.

Linear programming will probably stay one of the most popular operations research tool used in practice. The development of computer technology makes it possible to solve linear production planning problems routinely by inexperienced users as well. The interpretation of the sensitivity output of the currently available solvers is difficult and contains several traps. The proposed definition of the *three types of sensitivities* may help the analyst to place the proper questions about sensitivity, and the suggested *computation method* may help to provide the correct answers to these questions.

As a summary, based on Chapter 2, the following scientific results can be formulated:

Result 1/1

I have defined the following three different types of sensitivity information for the sensitivity analysis of the optimal solution of linear programming problems:

Type I sensitivity: Type I sensitivity determines those values of some model parameters for which a given *optimal basis* remains optimal.

Type II sensitivity: Type II sensitivity determines those values of some model parameters for which the positive variables in a given primal and dual optimal solution remain positive, and the zero variables remain zero, i.e. the *same activities* remain active.

Type III sensitivity: Type III sensitivity determines those values of some model parameters, for which, *the rate of change* of the optimal objective value function is unchanged.

Result 1/2

To obtain *Type III* sensitivity information of the optimal solution of a linear programming problem I have developed an algorithm which is based on the LP models summarized in Table 2.8. With these models sensitivity information related to the objective function coefficients (OFC) and to the right-hand side (RHS) parameters can be determined.

The definition and detailed explanation of the three sensitivity types can be found in Koltai and Terlaky (1999, 2000). The algorithm for calculating the Type III sensitivity information is published in Koltai and Tatay (2008a, 2008b, 2011). The interpretation of the different sensitivity types in case of linear production planning models are discussed in Koltai (1995, 2006), Koltai, Romhányi and Tatay (2009), and Koltai and Tatay (2008a).

3 ROUTE-INDEPENDENT ANALYSIS OF AVAILABLE CAPACITY IN FLEXIBLE MANUFACTURING SYSTEMS

One of the objectives of production planning is the optimal allocation of production tasks to production resources. In conventional manufacturing systems, generally, production planning models allocate parts/products directly to the machines. In flexible manufacturing systems a wide range of operations can be performed by the machines. In these systems parts/product can be prepared along several routes, consequently, instead of the classical product mix problem, the best possible routing mix must be determined. This chapter discusses some important questions of the analysis of routing. The requirement of a new way of aggregation in the planning stage is explained and justified. Capacity analysis of flexible manufacturing systems based on the suggested operation type aggregation concept is explained, and sensitivity of the optimal capacity allocation with respect to machine capacity changes and to operation time changes is analyzed. The results of this chapter are based on the papers of Guerrero et al. (1999), Koltai et al., (2000) and Koltai and Stecke (2008).

3.1 Introduction

A Flexible Manufacturing System (FMS) is an automated manufacturing system consisting of a set of numerically controlled machine tools with automatic tool interchange capabilities, linked together by an automated material handling system. One of the most important features of an FMS is the capacity to efficiently produce a great variety of part types in variable quantities. The aim of FMS is to achieve the efficiency of automated mass production, while conserving the ability of a job shop to simultaneously machine several part types. However, managing the production of an FMS is more difficult than managing production lines or job shops because the additional, flexibility-related degrees of freedom greatly increase the number of decision variables.

There are several production management problems which must be solved simultaneously or hierarchically in the operation planning phase of an FMS. Stecke (1986) defined the following problems:

a) *Part type selection*: from a set of part types a subset must be determined which contains those parts, which will be simultaneously processed. This can also be called batching.

b) *Machine grouping*: The machine tools of each type must be partitioned into groups. In each groups the machine tools are identically tooled and can perform the same operations.

c) *Production ratios*: the calculation of the ratios of those part types which are selected in problem a).

d) *Resource allocation*: The limited number of pallets and fixtures of each fixture type must be allocated to the selected part types.

e) *Loading*: The operations and the associated cutting tools of the selected set of part types must be allocated to the selected machine groups subject to technological and capacity constraints of an FMS. This problem includes the scheduling and routing information as well.

Several models are developed in the literature which solves a set of the above problems simultaneously or hierarchically (see the literature review in Chapter 3.3). The exact evaluation of the capacity of an FMS can be determined, only if all the above problems are solved. Often, however, operations managers need a route-independent answer to production planning questions. For example “How much can be produced of a certain part type and when” are important capacity questions in business negotiations, when the detailed routing and scheduling is not yet of interest or cannot be known.

The objective of this chapter is to provide an aggregate approach to a route-independent capacity analysis for FMS production planning. The chapter is organized as follows. In Chapter 3.2 the problem of capacity analysis in FMS is illustrated with a simple example. In Chapter 3.3, the relevant literature is reviewed. In Chapter 3.4, some preliminary research which provided the basis of the introduced aggregation concept is discussed. In Chapter 3.5 the concepts of operation type and available capacity range are introduced and the basic definitions and notation are explained. Next, the mathematical formulation of the capacity constraints and its application in production planning models are presented and the sensitivity analysis of the feasible capacity range is described in Chapters 3.6, 3.7 and 3.8. Finally, Chapter 3.9 provides some general conclusions.

3.2 Illustration of the problem

Manufacturing systems produce parts to meet demand, which is either forecasted and/or is an actual quantity. When developing a production plan, an initial question is whether there is enough capacity of the system for the different operations needed. Production planning for conventional manufacturing systems is more straightforward than in flexible manufacturing systems. In some conventional systems, the capacity available for production can be determined directly from the available capacities of the different single-purpose machines, as they can usually perform only a small variety of operations. A system with multi-purpose computer numerical control (CNC) machines provides additional opportunities to increase system utilization through machine flexibility, since each machine can be used for a variety of operations. In this case, however, the capacity of the system is related to the routing of the parts. The problem of route dependence of manufacturing capacity is illustrated with the help of Figure 3.1.

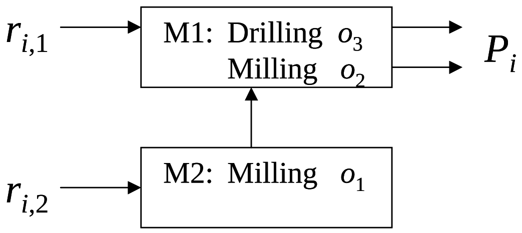

Figure 3.1 Illustration of the routing of part type i

For example, consider two identical flexible machines. In a current configuration, M1 is tooled for drilling and milling operations, while M2 is tooled just for the milling operations. The milling operations of part type *i* can be done on either M1 or M2 while the drilling operations can be performed only on M1. In this case, the capacity of the system to produce part type *i* depends on the quantity produced on the two different routes ($r_{i,1}$, $r_{i,2}$) indicated in Figure 3.1. Therefore, an operations manager's questions on the production quantity of this part type cannot be answered without solving the routing problem. In practice, however, managers frequently face situations where the production quantity should be determined prior to deciding the routing of parts. It would be desirable to be able to determine the available capacity independent of the future routing.

To illustrate this problem, consider a part manufacturer which produces two different part types on four CNC machines (M1, M2, M3 and M4)[1]. In the current setup configuration, there

[1] This example is based on the manufacturing of two simple parts in the Machine Division of GE Lighting Tungsram in Budapest, Hungary.

are several ways to perform the required operations, which are summarized in Table 3.1. P1, for example, can be manufactured on a route visiting all machines, and/or on a route visiting just machines M2 and M4, as well as other routes. P2 can only be manufactured on two different routes visiting either machines M1 and M4 or machines M2 and M4. Production quantity related questions therefore can't be answered without knowing the routing mix of the parts. A production planner at times would like to know the answers to the following questions:

1) Is there enough capacity to produce the required quantities of the different part types?
2) A design change of a part requires an increase in the processing time of one of its operations. Can we complete the orders on time, despite the change in this operation time?
3) Maintenance of some machines is scheduled for a given period. Can we complete the orders of that period, despite the decrease in capacity?
4) Is it worthwhile or necessary or possible to schedule overtime in case of a lack of capacity?

In many situations, a route-independent answer is desired to these questions. If, for example, we are negotiating with a customer about some order (whether to accept the order or not; how to set a due date), we would like to have a global view of available capacity. We want to know if some orders can be completed, independent of how the parts of that order will be routed. If maintenance of some machines is planned, we want to know whether the orders can still be completed with less capacity on some machines. In these cases, the routing and scheduling of the parts is not known yet. We only need to know if the system is capable of manufacturing the required parts both with the current available capacity and also when there is less capacity during planned maintenance. If the answer is yes, and decision is made on the maintenance, then the subsequent detailed planning process can determine the routing of the parts. Route-independent answers to the above questions are given in Chapters 3.7, and 3.8.

Table 3.1 Basic data of the sample problem
(operation times in capacity units)

<table>
<tr><th>Part Type</th><th>Operation</th><th>M1</th><th>M2</th><th>M3</th><th>M4</th><th>Operation type</th></tr>
<tr><td rowspan="4">P1</td><td>Turning</td><td>0.0062</td><td>0.0062</td><td></td><td></td><td>$\boldsymbol{ot_1}$</td></tr>
<tr><td>Drilling 1</td><td></td><td>0.0069</td><td></td><td>0.0069</td><td>$\boldsymbol{ot_2}$</td></tr>
<tr><td>Drilling 2</td><td></td><td></td><td>0.0074</td><td>0.0074</td><td rowspan="2">$\boldsymbol{ot_3}$</td></tr>
<tr><td>Screw cutting</td><td></td><td></td><td>0.0049</td><td>0.0049</td></tr>
<tr><td rowspan="2">P2</td><td>Turning</td><td>0.0106</td><td>0.0106</td><td></td><td></td><td>$\boldsymbol{ot_1}$</td></tr>
<tr><td>Milling</td><td></td><td></td><td></td><td>0.0543</td><td>$\boldsymbol{ot_4}$</td></tr>
</table>

Table 3.2 Summary of literature review

Objective function	Loading	Grouping and loading	Batching and loading	Loading and scheduling	Tool allocation and routing
Workload balancing	Berrada and Stecke (1986) Wilson (1992) Kim and Yano (1993) Kirkavak and Dincer(1993) Kim and Yano (1994)	Stecke (1983) Stecke (1986) Stecke and Raman(1994)	Bastos (1988) Shanker and Sirinivasulu (1989) Stecke and Kim (1991) Moreno and Ding (1993) Solomon, Millen and Afentakis (1995)	Sanker and Tzen (1985) Sawik (1990)	Arbib, Lucertini and Nicolo (1990) Sodhi, Agnetis and Askin (1994)
Cost optimization	Sarin and Chen (1987) Ram, Sarin and Chen (1990) Kouvelis and Lee (1991) Basnet (1996)		Liang and Dutta (1993)		Liang and Dutta (1990) Sodhi, Askin and Sen (1994)
Part movement minimization	Shanker and Rajamarthandan (1989) Wilson (1989)	Stecke (1983) Stecke (1986)			Arbib, Lucertini and Nicolo (1990) D'Alfonso and Ventura (1995)
Sum of part type priorities maximization			Bastos (1988) Hwang and Shogan (1989) Liang and Dutta (1993) Srivastava and Chen (1993) Mohamed (1996)		
Tool changeovers minimization	De Werra and Widmer (1990)			Sawik (1990)	
Makespan minimization			Chen and Chung (1996) Liang and Dutta (1993)	Greene and Sadowski (1986) Sherali, Sarin and Desai (1990) Chen and Chung (1996)	Chen and Chung (1991) Liang and Dutta (1990)
Total processing time minimization	Chakravarty and Shtub (1984) De Werra and Widmer (1990)				
Part type lateness minimization			Moreno and Ding (1993)	Shanker and Tzen (1985) Green and Sadowski (1986) Sawik (1990)	
Other	Lashkari, Bopari, and Paulo (1987) De Werra and Widmer (1990)	Stecke (1983) Stecke (1986)	Shanker and Sirinivasulu (1989)	Greene and Sadowski (1986)	Chen and Chung (1991) Atan and Pandit (1996)

3.3 Literature review

The loading problem has been extensively studied due to its importance in FMS production planning. The available capacity for production in an FMS is generally examined together with the loading and routing problem. The loading problem has been considered separately as well as together with other related problems such as machine grouping, part type selection and scheduling. A variety of loading objectives have been considered. Also, instead of loading, some researchers consider only one of the two issues that make up the loading problem: tool allocation and routing. Table 3.2 provides a summary of the approaches categorized by problem type and objective function. Since some researchers use more than one objective function, the same item may appear more than once in the same column. Note the following:

– Overall, the most common objective function is workload balancing, followed by the more traditional cost minimization approach.

– When not alone, loading has been solved most often together with part type selection and with scheduling. In the first case, the usual objective function is the maximization of the sum of part type priorities. In the second case, traditional scheduling performance measures such as makespan and lateness are used.

– Many researchers have either proposed alternative objective functions or have tried to harmonize more than one objective.

Two assumptions of most existing approaches can be observed. The first general assumption is that each part type is required to follow only one of its alternative production routes. This makes only a partial use of the routing flexibility of the FMS. The second assumption is that the models consider only one copy of each tool type to be assigned to a machine when the tool is required. However, multiple copies may be beneficial, or even necessary, if they are used heavily or have short lives. It is thus desirable to allow for duplicate copies of tools to be loaded in tool magazines. This increases the length of time until the system is stopped to change tools and also augments the amount of processing that can be shared among machines.

Note that Table 3.2 summarizes the literature available in the period when this research was conducted between 1991 and 1997 (Guerrero et al., 1999).

Finally, it can be concluded that, although many mathematical programming models have been proposed to solve the FMS loading problem, none considers explicitly the concept of production route, which is the major source of operational flexibility of an FMS, and has a crucial role in capacity planning.

3.4 The requirement of an aggregation concept

Since routing of parts is one of the major sources of complexity in the planning phase, any effort which results in acceptable simplification of planning models may have scientific and practical significance. Two of my earlier research must be mentioned, which lead to the suggested aggregation concept.

One possible way to describe routing possibilities in production planning models is the introduction of the routing parameter θ_{ir} which expresses the proportion of parts *i* following route *r* in the system. In a paper written with Guerrero et al. (1999) we have formulated a production planning model which determined optimal batching of parts together with optimal routing and tooling. The objective function in this model expressed deviation from a balanced workload, which was minimized. The recommended model has limited practical significance as a consequence of the high number of variables, but the proposed introduction of the routing parameter (θ_{ij}) gained wide acceptance in the scientific literature (See for example Lashkari, Bopari and Paulo, 2004; Magarjuna, Mahesh and Rajagopal, 2006; Sujono and Lashkari, 2007; Navala and Awari, 2011; Arikan and Erol, 2012).

Flexible manufacturing systems have high fixed cost which makes product costing very complicated. Costing in high fixed cost production and service system can be efficiently performed with activity based costing (Johnson, 1991). Fixed costs must be assigned to the products according to the resource consumption during manufacturing. Parts following different routes, however, might have different resource consumption characteristics, and consequently several different unit manufacturing cost may belong to a product if not all units follow the same route. To illustrate this problem, I have developed a costing model based on activity based costing (ABC) which is illustrated in Figure 3.2.

According to the suggested model, production overhead is divided into five activity centers. Obviously each activity center represents several activities, but the applied cost drivers must correctly estimate the resource consumption of the pooled activities. The "other activities" can be further split based on the specific characteristics of an FMS. Three out of the five overhead allocation bases are calculated from the results of a production planning model and the other two are calculated from the output of the real or simulated performance of an FMS. Detailed numerical results generated by the proposed costing system can be found in Koltai et al. (2000).

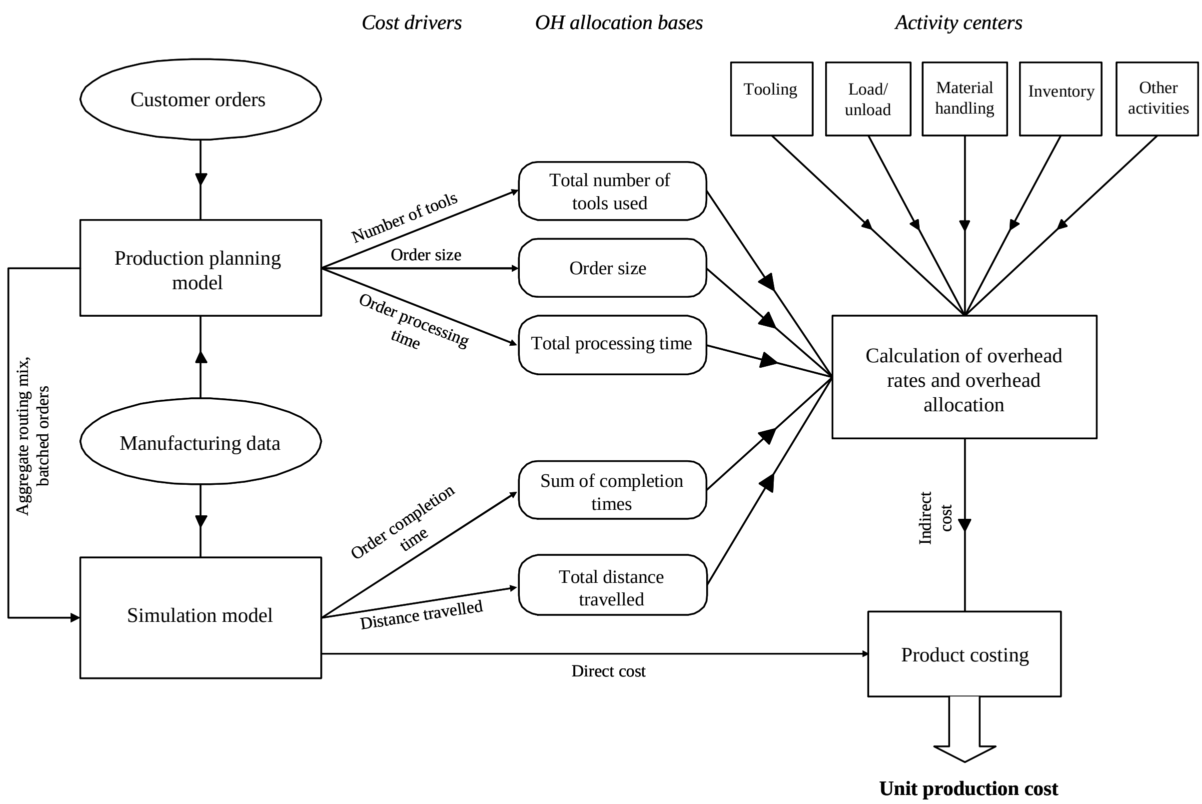

Figure 3.2 Activity based costing in a FMS

The analysis of the performance of the suggested flexible costing system shows the complexity of the overhead allocation process. Production planning selects different orders under different conditions, which influences the batch makespan, the overhead allocation bases and rates and the overall resource consumption. The interaction of these four elements makes unit production costs unpredictable unless a sophisticated costing system similar to the suggested one is available. The several different unit costs obtained for the same product do not imply that selling price of the product should constantly be updated. But monitoring the constantly changing manufacturing cost of a product in a longer period can help to examine whether an FMS is really using its potential flexibility, and whether the products produced in different product mixes are really requiring the available flexibility.

The complexity of modeling of routing in the production planning phase and the difficulty of production cost evaluation indicate that the elimination of routing with some kind of aggregation technique may help to ease the complexity problem.

Aggregation is a widely used tool in production planning. Among the reasons for aggregation, the following three are especially important.

– First, when some elements of a production system are aggregated, simpler models can be applied for capacity, inventory, and production planning. When more detailed information is required, then the aggregated elements are disaggregated and/or a more detailed model is applied. In traditional production planning models, products and/or facilities are aggregated (Thomas and McClain 1993). Products using the same setup of a production process are aggregated into product families and/or products with similar resource consumption are aggregated into product types. When a feasable aggregate production plan and capacity utilization is determined, a detailed production program can be prepared, in which product types and/or families are disaggregated into products (see, for example, Johnson and Montgomery 1974, Hax and Candea 1984). Facility-level aggregation consolidates several different production resources, such as machines, workforce, and materials, into a single resource or facility (see, for example, Holt et al. 1960).

– Second, in situations where the production environment changes constantly (e.g., changing demand, machine breakdowns, online control decisions), capacity planning should be insensitive to such changes. Robust planning methods are required, which provide results for a large variety of possible scenarios (Váncza and Kovács, 2004; Taal and Wortmann, 1997; Tolio, Urgo and Váncza, 2011). Aggregation of products and/or resources helps to decrease the consequences of changes in the production environment (Nam and Logendran 1992; Vollman, Berry and Whybark, 1997).

– Third, when not all information is available for planning (e.g., scheduling decisions are not made yet), a rough estimate of the required workforce and machine capacities can be done at an aggregate level. There are several rough-cut capacity planning methods based on the aggregation of several elements of a production system (see, for example, Vollman, Berry and Whybark, 1997). These methods are built into most of the standard software packages prepared for production and capacity planning (Wortman et al., 1996).

In most decision-making-contexts aggregate models serve as a link between strategic and tactical decisions (Singhal and Singhal 2007). These models can be used as rough cut planning tools, which provide aggregate information for strategic decisions without the unnecessary (and frequently unknown) details of operation. Furthermore, their results provide a planning framework for operational decisions as well.

What should be aggregated is an important question of aggregate planning. A special-purpose machine performs just a small set of technologically different operations. In this case, aggregation of machines is approximately equivalent to the aggregation of operations. In FMSs, machines and operations have to be treated separately, since an operations manager decides the set of operations a machine can perform. This was recognized by Niess (1980), who aggregated similar operations into operation types. Niess developed an algorithm to determine a series of sets of orders for several production periods. The generated production plan provides balanced capacity utilization. Niess applied this method to conventional production systems consisting of several single- and multi-purpose machines.

Bertrand and Wortmann (1981) formulated capacity constraints that consider the alternative use of several resources. These constraints were first formulated for machine operators. If several operators can tend several machines, the situation is similar to an FMS, in which several machines can perform several operations. The formulated capacity constraints were extended from operators to machines, but used for detailed planning, and not for aggregate planning.

It can be concluded that methods that reduce the complexity of FMS capacity analysis, and provide aggregate, rough-cut, route-independent estimates of available capacity can be useful for operations managers of these systems (Koltai and Stecke, 2008).

3.5 Basic definitions and concepts of the aggregation based on operation types

The notations used in Chapter 3.5 are summarized in Table 3.3. A flexible manufacturing system (FMS) is a collection of machines, linked by an automated materials handling system and directed by a central computer. Different part types are produced in the system and each part type has a finite number of operations.

An *operation*, o_j, is defined by its processing time on a machine and by the set of cutting tools required. In FMSs, generally more than one machine can perform certain operations. In Figure 3.1, for example, each machine can perform milling operations. This provides routing flexibility.

The set of all operations, which can be performed on any machine in a particular group of machines, is called *operation type*, ot_h. An operation type is an aggregated set of operations. In the example of Figure 3.1, the drilling operations can only be performed on machine M1. Therefore this operation is also an operation type (ot_1). Milling can be done on machines M1 and M2. Therefore the milling operations of these two machines can be aggregated into another operation type (ot_2).

To analyze the capacity of an FMS, the available capacity of every combination of the operation types has to be known. A specific combination of different operation types is called an *operation type set*, S_k.

The tooling of a machine determines the operations that the machine can perform. Operations are aggregated into operation types. Therefore a machine can perform a set of operation types. An *operation type set assignment parameter,* z_{km}, specifies the operation type sets assigned to machine m. If z_{km}=1, then machine m can perform all operations belonging to operation type set k.

The calculation of the capacity of each operation type set is based on machine capacity. The *capacity of a machine*, c_m, is expressed in capacity units (CUs) over a period of, for example, a shift or two, or a day, or a week. The capacity unit is a normalized measure of the available capacity for the period examined. For example, 1 CU is equal to 8 hours, if the production capacity of one 8 hour long shift is to be examined.

An *upper capacity bound* of a particular operation type set k, u_k, is the *maximum amount of capacity available* for operation type set k. It is calculated as the sum of the CUs of those machines that are capable of performing *any and all* operations belonging to that operation type set, that is,

$$u_k = \sum_{\{k''|S_{k''} \in S''_k\}} \sum_{m=1}^{M} c_m \cdot z_{k''m} \quad k = 1, \ldots, K \tag{3.1}$$

In equation (3.1), S''_k is the set of all operation type sets that contain any of the operation types of operation type set S_k. For each operation type set k, it is checked to see if the corresponding S''_k sets can be found on any of the machines. If they can, then the corresponding machine capacities are considered at the calculation.

In each column of the matrix defined by z_{km}, only one element is equal to 1, since only one operation type set can be assigned to a machine m. Therefore, the capacity of each machine can be considered only once in the summation in (3.1). But the capacity of machine

m is considered in the summation only if the operation type of that machine is an element of the set S''_k, that is, for which $S_{k''} \in S''_k$ and $z_{k''m} = 1$.

Table 3.3 Summary of notation of Chapter 3

Subscripts:

i	–	index of part type (1, ...,I),
h	–	index of operation type (1, ...,H),
k	–	index of a set of operation types (1, ...,K),
k'	–	index of a subset of a set of operation types (1, ...,K'),
k''	–	index of a subset of the set of all operation types (1, ...,K''),
m	–	index of machines (1, ...,M).
Parameters:		
o_j	–	operation j,
ot_h	–	operation type h,
S_k	–	operation type set k,
S'_k	–	set of operation type sets that contain only operation types belonging to S_k,
S''_k	–	set of operation type sets that contain any of the operation types of operation type set S_k,
c_m	–	production capacity of machine m,
z_{km}	–	operation type set assignment parameter. It is equal to 1 if operation type set k is assigned to machine m, and it is equal to 0 otherwise,
u_k	–	upper capacity bound of operation type set k,
l_k	–	lower capacity bound of operation type set k,
α	–	acceptable ratio of capacity under-utilization,
β	–	acceptable ratio of capacity over-utilization,
p_{ji}	–	processing time of operation j of a part of type i,
pt_{hi}	–	processing time of all operations of operation type h of type i,
ps_{ki}	–	processing time of all operations of operation type set k of a part of type i,
rt_h	–	capacity requirement of operation type h,
rs_k	–	capacity requirement of operation type set k,
w_i	–	weight of part type i,
Δrt_h^-	–	feasible decrease of the capacity requirement of operation type h,
Δrt_h^+	–	feasible increase of the capacity requirement of operation type h,
Δc_m^-	–	feasible decrease of the capacity of machine m,
Δc_m^+	–	feasible increase of the capacity of machine m,
w_i	–	weight of a part type in the objective function.
Variables:		
x_i	–	production requirements of part type i.

A *lower capacity bound* of an operation type set k, l_k, is the *minimum amount of unassigned capacity* for operation type set k that is available only for the operations that belong to that operation type set. It is calculated as the sum of the CUs of those machines that are capable of performing *only* those operations belonging to that particular operation type set.

$$l_k = \sum_{\{k' | S_{k'} \in S'_k\}} \sum_{m=1}^{M} c_m \cdot z_{k'm} \quad k = 1, \ldots, K \tag{3.2}$$

In equation (3.2), S'_k is the set of those operation type sets that contain only operation types belonging to S_k. For each operation type set k it is checked to see if the corresponding

S'_k sets can be found on any of the machines. If they can, then the corresponding machine capacities are considered at the calculation. The capacities of those machines need to be summed, which contain any of the operation type sets of S'_k, that is, for which $S_{k'} \in S'_k$ and $z_{k'm} = 1$.

Some details about these bounds are as follows. If each machine that can perform any of the operation types of a specific operation type set k can perform operation types not belonging to operation type set k, then the lower capacity bound of operation type set k is equal to zero. For example, if each machine can perform all operation types, then the lower capacity bound of *all but one* operation type sets are equal to zero. In this case, the only non-zero lower capacity bound will belong to the operation type set that contains all operations types. For this operation type set, the lower and upper capacity bounds always coincide, and are non-zero.

The *available capacity* per period for an operation type set is a range defined by the upper and lower capacity bounds. A necessary condition of capacity availability is that the capacity requirements from all operation type sets must be less than their corresponding upper bounds. When all operations have been assigned to machines and the workload is less than the lower capacity bound of any operation type set, then there is machine idle time.

In real manufacturing systems, production managers may need to work around a certain amount of lack of capacity. To supplement capacity, management may consider overtime, subcontracting, or other possible capacity adjustments. The size of acceptable capacity over-utilization, β, is expressed as a percentage of total capacity. The capacity increased by acceptable over-utilization is called the *extended capacity upper bound*.

Also, production managers are generally resigned to a certain amount of idle capacity. Idle capacity is either planned and serves as buffer capacity to absorb the effect of unexpected events (i.e., machine breakdowns, tool breakages, quality problems, expected or unexpected rush orders) or it is a consequence of scheduling constraints. The size of acceptable capacity under-utilization, α, is expressed as a percentage of total capacity. The capacity decreased by the acceptable idle time is called the *extended capacity lower bound*.

The upper and lower capacity bounds are necessary but not sufficient conditions for the feasibility of a production plan. The role of the extended upper and lower capacity bounds is to provide a capacity reserve for those phenomena not considered in the aggregate planning phase. The extended capacity bounds allow a link between the aggregate and the operational planning and control levels.

The available capacity range and the extended capacity range are information about capacity that can be used for production planning. To analyze the utilization of a production system, the *capacity requirements* of the operation types and operation type sets should also be known. This data is based on the processing times of the individual operations.

The *processing time of operation j* of a part of type i, p_{ij}, is expressed in terms of CUs, rather than in hours or minutes. The ***processing time of operation type*** h of part type i, pt_{hi}, is the sum of the processing times of all of those operations that belong to operation type h, that is,

$$pt_{hi} = \sum_{\{j | o_j \in ot_h\}} p_{ji} \tag{3.3}$$

The *processing time of operation type set k* of one part of type i, ps_{ki}, is the sum of the processing times of all of those operation types that belong to operation type set k, that is,

$$ps_{ki} = \sum_{\{h | ot_h \in S_k\}} pt_{hi} \tag{3.4}$$

If the production requirements of part type i, x_i, are known, then the *capacity requirements of operation type h* can be calculated as the sum of the processing times of all of those operations that belong to operation type h, that is,

$$rt_h = \sum_{i=1}^{I} \sum_{\{j | o_j \in ot_h\}} p_{ij} x_i \tag{3.5}$$

The *capacity requirements of operation type se***t** *k* is the sum of the capacity requirements of all of those operation types that belong to operation type set k, that is,

$$rs_k = \sum_{\{h | ot_h \in S_k\}} rt_h \tag{3.6}$$

Note that neither the calculation of the available capacity of the operation type sets, nor the calculation of the capacity requirements of the operation type sets, require routing information.

Using operation type aggregation, a *machine* capacity problem can be transformed into an *operation type set* capacity problem. In an FMS, the capacity of a machine is relative to its tooling and routing decisions. If a machine is tooled both for milling and drilling as in Figure 3.1, then the available capacity of that machine for drilling depends on how much capacity is required or used for milling. This, however, depends on the routing of the manufactured parts in the whole system. In Figure 3.1, if those routes that require the milling operation of M1 are not used, then we have a lot of capacity for drilling. If, however, many parts are routed to M1 for milling, then less capacity is left for the drilling operation. Consequently, what is important is not the machine capacity, but the operation type capacity, that is, how much capacity the production system has for those operations that can be performed on any machine in a particular group of machines.

3.6 Illustration of the available and extended capacity ranges

The introduced notation and concepts in Chapter 3.5 are illustrated with the help of Table 3.1 and Figures 3.3 and 3.4. The illustration is based on the example used in Chapter 3.2 to introduce some basic concepts. The capacity requirements for manufacturing two part types (P1 and P2) are analyzed. These part types require several operations as indicated in Table 3.1. Some operations can be produced on two different machines. For example, M4 is a machine center that can perform most of the operations of P1. Several milling operations of P2, however, can only be performed on M4. Therefore, in practice, some capacity of M4 may be reserved for the production of P2.

For the sake of illustration and simplicity, it is assumed that the processing times of those operations that can be manufactured on any of several machines are identical on all of the possible machines. This assumption simplifies a general problem of aggregation. If operation times of an operation are not identical on each machine, then alternative uses of the machines results in different operation times. For example, suppose that the operation times of the operations of ot_1 in Figure 3.3 are different on the two different machines (M1 and M2). Then there is no single operation time that can be used when the capacity requirement of ot_1 is evaluated, because the aggregated operation time will depend on how many parts are routed to each machine. There are several possible ways to cope with this problem in practice. Here are some examples.

a) The operations with different operation times on two machines are aggregated into two different operation types. In this case, the number of operation types, and therefore the number of operation type sets, will increase. (Exact solution)

b) A weighted average of the operation times is used. The weights assume a possible routing ratio between the two machines. In this case, the aggregated operation time is approximate.

c) Identical operation times are used, and the actual difference in operation times is reflected in the available capacity.

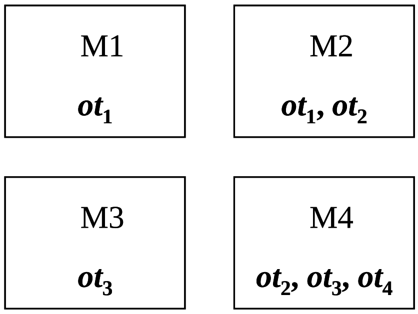

Figure 3.3 Allocation of operation types to machines

The processing times of the operations of the two part types are given in Table 3.1 in capacity units (CUs). In this example, the production system works in one 7.5 hour shift with 0.9 efficiency. Therefore, 0.0062 CUs in Table 3.1 is equivalent to 2.5 minutes (0.0062*7.5*60*0.9).

Since an operation type is an aggregate set of all operations that can be performed on the same group of machines, 4 operation types are identified in Table 3.1. Those operations that can be done on M1 and M2 are aggregated into ot_1. ot_1 is the result of aggregating the turning operations of P1 and P2 into one operation type. The operations that can be done on M2 and M4 are aggregated into ot_2. Those operations that can be done on M3 and M4 are aggregated into ot_3. ot_3 is the result of aggregating two different operations of P1 into one operation type. The operations that require only M4 are ot_4. The machines with their aggregated operations are illustrated in Figure 3.3.

The available capacity ranges for all operation type sets are displayed in Figure 3.4. All operation type sets are placed on the horizontal axis. In this case, there are four operation type sets (S_1, …, S_4) with single operation types, six operation type sets (S_5, …, S_{10}) with two operation types, four operation type sets (S_{11}, …, S_{14}) with three operation types, and one operation type set (S_{15}) with four operation types. The total number of operation type sets, K, can be calculated as $K=2^H-1=2^4-1=15$.

The vertical axis of Figure 3.4 represents the upper capacity bounds, the lower capacity bounds, and the available capacity range of each operation type set. One capacity unit is equal to 405 minutes/day (one 7.5 hour shift, with 0.9 efficiencies).

The upper capacity bounds (u_k) in Figure 3.4 are calculated with equation (3.1). The elements of the S_k'' set in case of the sample problem of Figure 3.3 are listed in Table 3.4. For example, S_5 contains ot_1 and ot_2; therefore S_5'' contains all of those operation type sets that contain either ot_1 or ot_2, as seen in Table 3.4. Since S_1 belongs to M1, S_5 belongs to M2, and S_{13} belongs to M4, and S_1, S_5, and S_{13} are all elements of S_5'', the sum of the capacities of these three machines is calculated according to the left-hand side of equation (3.1).

The lower capacity bounds (l_k) in Figure 3.4 are calculated with equation (3.2). Table 3.4 shows the elements of S_k' for all operation type sets of the sample problem illustrated in Figures 3.3. For example, S_5 contains ot_1 and ot_2; therefore S_5' contains all operation type sets that contain only ot_1, ot_2, or both ot_1 and ot_2, as can be seen in Table 3.4. Since S_1 belongs to

M1, S_5 belongs to M2, and S_1 and S_5 are all elements of S'_5, the sum of the capacities of these two machines is calculated using the left-hand side of equation (3.2).

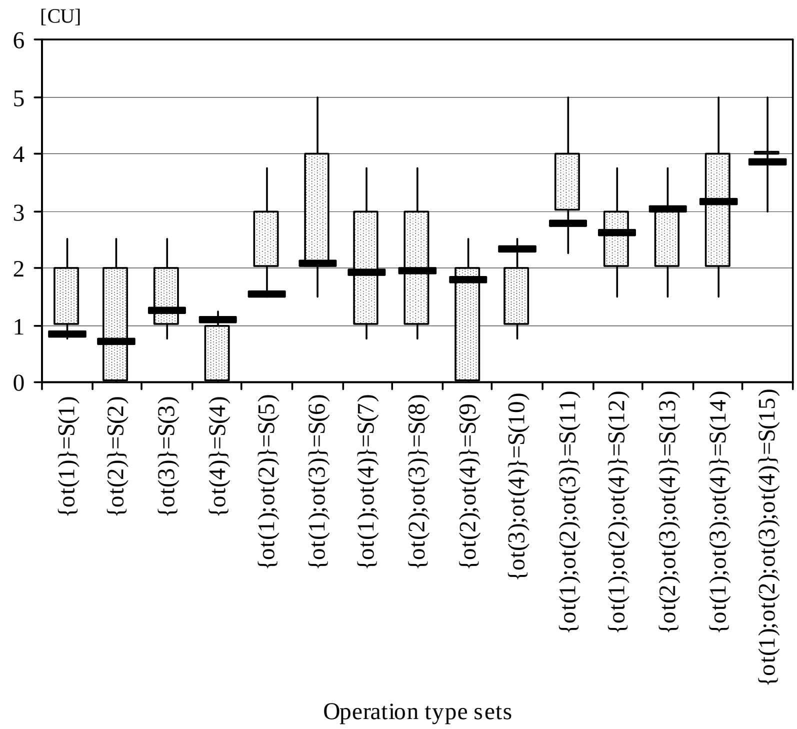

Figure 3.4 Illustration of the ideal available capacity range and the capacity requirements (α=0.25, β=0.25, x_1=100, x_2=20)

The l_k and u_k values are indicated as the lower and upper sides of the grey block at each S_k in Figure 3.4.

For example, for operation type set S_1, the lower bound l_1=1 CU (405 minutes), because M1 is the only machine which exclusively performs the operations of ot_1. The upper bound u_1=2 CUs (810 minutes), since M1 and M2 are both capable of performing the operations of ot_1. For S_6, which is an operation type set containing operation types ot_1 and ot_3, l_6=2 CUs (810 minutes) because M1 can perform ot_1 and M3 can perform ot_3; u_6=4 CUs (1620 minutes), as all four of the machines can do either ot_1 or ot_3.

If the capacity requirements are smaller than the lower bound of any operation type set, then the system is underloaded, and there is idle capacity on one or more machines. For example, if less than 1 CU (405 minutes) from ot_1 is required, then M1 would have idle capacity. If the capacity requirement is greater than the upper bound of any operation type set, then the system is overloaded, i.e., there is not enough capacity available for the required operations. For instance, if more than 2 CUs (810 minutes) are required for ot_1, then M1 and M2 will not have enough capacity to perform the turning operations. When the capacity requirements of each operation type set are within the lower and upper capacity bounds (the gray area in Figure 3.4), then there is neither unutilized nor excess capacity on any of the machines. In the most favorable situation, the capacity requirements of a set of orders of a given period fall into the ranges defined by the lower and the upper bounds for each operation type set.

The capacity requirements for each operation type set are also shown in Figure 3.4 by a bold horizontal line at each operation type set. The capacity requirement data are calculated for a production plan that calls for production of 100 parts of P1 and 20 parts of P2.

Table 3.4 Definition of sets S′ and S″ in the sample problem

k	S_k	S'_k	S''_k
1	$\{ot_1\}$	$\{S_1\}$	$\{S_1; S_5; S_6; S_7; S_{11}; S_{12}; S_{14}; S_{15}\}$
2	$\{ot_2\}$	$\{S_2\}$	$\{S_2; S_5; S_8; S_9; S_{11}; S_{12}; S_{13}; S_{15}\}$
3	$\{ot_3\}$	$\{S_3\}$	$\{S_3; S_6; S_8; S_{10}; S_{11}; S_{13}; S_{14}; S_{15}\}$
4	$\{ot_4\}$	$\{S_4\}$	$\{S_4; S_7; S_9; S_{10}; S_{12}; S_{13}; S_{14}; S_{15}\}$
5	$\{ot_1; ot_2\}$	$\{S_1; S_2; S_5\}$	$\{S_1; S_2; S_5; S_6; S_7; S_8; S_9; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
6	$\{ot_1; ot_3\}$	$\{S_1; S_3; S_6\}$	$\{S_1; S_3; S_5; S_6; S_7; S_8; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
7	$\{ot_1; ot_4\}$	$\{S_1; S_4; S_7\}$	$\{S_1; S_4; S_5; S_6; S_7; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
8	$\{ot_2; ot_3\}$	$\{S_2; S_3; S_8\}$	$\{S_2; S_3; S_5; S_6; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
9	$\{ot_2; ot_4\}$	$\{S_2; S_4; S_9\}$	$\{S_2; S_4; S_5; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
10	$\{ot_3; ot_4\}$	$\{S_3; S_4; S_{10}\}$	$\{S_3; S_4; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
11	$\{ot_1; ot_2; ot_3\}$	$\{S_1; S_2; S_3; S_5; S_6; S_8; S_{11}\}$	$\{S_1; S_2; S_3; S_5; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
12	$\{ot_1; ot_2; ot_4\}$	$\{S_1; S_2; S_4; S_5; S_7; S_9; S_{12}\}$	$\{S_1; S_2; S_4; S_5; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
13	$\{ot_2; ot_3; ot_4\}$	$\{S_2; S_3; S_4; S_8; S_9; S_{10}; S_{13}\}$	$\{S_2; S_3; S_4; S_5; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
14	$\{ot_1; ot_3; ot_4\}$	$\{S_1; S_3; S_4; S_6; S_7; S_{10}; S_{14}\}$	$\{S_1; S_3; S_4; S_5; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$
15	$\{ot_1; ot_2; ot_3; ot_4\}$	$\{S_1; S_2; S_3; S_4; S_5; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$	$\{S_1; S_2; S_3; S_4; S_5; S_6; S_7; S_8; S_9; S_{10}; S_{11}; S_{12}; S_{13}; S_{14}; S_{15}\}$

The graphical display is used to determine whether or not the manufacturing system is in technological balance, i.e., to check whether there is any excess capacity or lack of capacity from certain operation types. Figure 3.4 indicates underload of operation type sets S_1, S_5, S_{11}, and S_{15}. For example, the underload at S_5 indicates that idle capacity may exist on M1 and M2.

Figure 3.4 also shows overload of operation type sets S_4, S_{10}, and S_{13}. For example, the overload of the operations of S_{10} indicates that the capacity requirements of ot_3 and ot_4 together are higher than the available capacities on machines M3 and M4.

In summary, there could be idle capacity on some machines, while there might not be enough capacity on other machines to fulfill production. Figure 3.4 displays the increased available capacity ranges when 25 percent capacity over- and under-utilization are acceptable for management ($\alpha=\beta=0.25$). The thin vertical lines represent the increased upper bounds and lower bounds. The capacity requirements are within this extended range for all operation type sets.

Another capacity analysis could be done by changing the part mix, i.e., increasing and/or decreasing the production requirements of the part types (see Chapter 3.7). Requirements could be changed until there is no capacity overload or underload.

The example of Figure 3.4 shows a potential difficulty when performing capacity analyses based on operation types. There could be a high number of operation type sets. The total number of operation type sets is $K=2^H-1$, where H is the total number of operation types. In our example, $K=15$. In general, however, the value of K is determined by the number of machines and by the number of alternative manufacturing possibilities. In the worst case, where for M machines all possible operation types exist (H=2^M–1), the total number of operation type sets is

$$K = 2^{\left(2^M - 1\right)} - 1 \tag{3.7}$$

In practice, the number of machines is small. Usually a group of 3-5 machines is dedicated to manufacturing a set of part types. If the number of machines is large, then methods exist to decompose the system into small independent or partly independent subsystems (Bertran and Wortmann 1981; Juhasz and Koltai 2003). Even if the number of machines is high and the system cannot be decomposed into subsystems, the value of K can be small; because it is unlikely that for all manufacturing possibilities at least one operation exists. It is more common that some group of operations can alternatively be performed on more than one machine. Therefore, although theoretically the value of H can be large, practically it is within a tractable range (5-20). In this range, computational time is acceptable for practical applications as is seen later in Table 3.5 of Chapter 3.7.

3.7 Model formulation

In this chapter, the route-independent mathematical programming formulation of the FMS capacity constraints is presented, using the notation of Table 3.3.

If the capacity requirements of all operation type sets are lower than the capacity upper bounds, then there is sufficient capacity to manufacture the required quantity. Constraint (3.8) imposes that the capacity requirements for all operations of operation type set k should be less than the upper capacity bound of operation type set k, that is,

$$(1+\beta) \cdot u_k \geq \sum_{i=1}^{I} x_i ps_{ki} \quad k = 1, \ldots, K \tag{3.8}$$

Since the upper capacity bound is the maximum amount of available capacity for operation type set k, constraint (3.8) ensures that there is enough capacity to perform all operations of operation type set k.

The parameter β expresses the acceptable percentage of capacity over-utilization. For example, in Figure 3.4 the capacity requirement for all operations of S_2 is lower than the upper capacity bound of S_2. In this case constraint (3.8) is feasible for k=2. For S_4, the capacity requirement is higher than the upper bound and therefore constraint (3.8) for k=4 is infeasible with β=0, and without a certain amount of overtime, the required production quantities cannot be completed. If, for example, β=0.25, then 2.083 (7.5*0.25/0.9) hours of overtime is allowed by management on those machines (M4) which perform the operations of ot_4.

If the capacity requirements of all operation type sets are higher than the capacity lower bounds, then all machines are fully utilized. Constraint (3.9) imposes that the capacity requirements of all operations of operation type set k should be higher than the lower capacity bound of operation type set k, that is,

$$(1-\alpha) \cdot l_k \leq \sum_{i=1}^{I} x_i ps_{ki} \quad k = 1, \ldots, K \tag{3.9}$$

Since the lower capacity bound of operation type set k is the sum of the capacities of those machines that are capable of performing only the operations of operation type set k, constraint (3.9) ensures that there is no idle capacity on those machines.

Of course, one doesn't want to plan for a fully utilized system. Some amount of idle time is necessary. The parameter α provides the acceptable percentage of capacity under-utilization of the machines. For example, in Figure 3.4, the capacity requirements for all operations of S_1 is lower than the lower capacity bound of S_1, which makes constraint (3.9) infeasible for α=0.1, but feasible for α=0.25. For S_2, the capacity requirement is above the lower bound, so constraint (3.9) is satisfied.

Equation (3.9) helps avoid capacity under-utilization. If machines are idle because of lack of capacity requirements, then there is no feasible solution for equation (3.9). Then other operation management measures can be taken, such as decreasing working hours, increasing order numbers or order sizes, or ultimately accepting underutilized capacity.

Finally, objective function (3.10) allows a variety of management objectives to be expressed when the production quantities of part types are to be determined.

$$Max \sum_{i=1}^{I} w_i x_i \tag{3.10}$$

If, for example, management would like to maximize the production quantity, then objective function (3.10) with w_i=1, i=1,…,I maximizes the sum of the quantities produced of each part type. If management wants to determine an economic part mix, then w_i can be, for example, the contribution margin of a part of type i.

Since the quantity of parts is measured in pieces, the variables x_i i=1,…,I are integers. Equations (3.8) – (3.10) provide an integer programming model, which can be solved by any available mathematical programming solver.

Other constraints may exist in practice. Generally, if other resources limit production (e.g., the material handling system) or waiting time or scheduling problems cause delay, then these issues can be considered in two possible ways by the presented approach.

a) Parameter α in equation (3.9) can be used to provide enough capacity reserve to address the above mentioned problems.

b) If further issues are also considered important by the decision maker (for example, scheduling or material handling details and/or delays), then additional constraints can be formulated.

To illustrate the use of the model, let us determine, for example, the maximum quantity that can be produced of the two products presented in the example in a working day. 15 extended lower bounds and 15 extended upper bounds can be formulated using equations (3.8) and (3.9).

Note that sometimes the constraints for certain operation type sets are redundant. It is easy to see that the lower and upper bound constraints for operation type sets S_6, S_7, and S_{14} can be ignored. To see this, consider operation type set S_6. It contains two operation types, ot_1 and ot_3. ot_1 is assigned to M1 and M2, and ot_3 is assigned to M3. These two operation types do not use the same machines. Therefore, if there is enough capacity separately for ot_1 and ot_3, then there is enough capacity for these two operation types together. Similar considerations explain the redundancy of constraints for operation type sets S_7 and S_{14}. An algorithm that finds redundant operation type set constraints can be found in Juhász and Koltai (2003).

To develop the model, we need the machine capacities; the operation type set assignment parameters, and the processing time information. All machines have the same working hours. Therefore c_m=1 CU, m=1,…,4. Based on Table 3.1 and Figure 3.3, $z_{1,1}$=$z_{3,3}$=$z_{5,2}$=$z_{13,4}$=1, and all other values of z_{km} are equal to zero. The value of ps_{ki} (k=1,…,15, and i=1,2) can be calculated with the data of Table 3.1 and using equations (3.3) and (3.4). Finally, for the production quantity to be maximized, w_1=w_2=1.

The resulting integer programming model consists of two integer variables (x_1 and x_2) and 24 constraints. The lower and upper bounds of the three redundant operation type sets are not needed. This small problem can easily be solved, for example, by Excel Solver. The optimal maximum quantities are x_1=194 and x_2=0 parts/day. If the weight in the objective function forces production of P2, then the total production quantity decreases. If w_1=1 and w_2=5, then the optimum solution is x_1=101 and x_2=23 parts/day.

The optimum solution consists of two values, the manufacturing quantities of the two part types. These quantities are independent of the several available manufacturing routes. We do not know yet how the parts will be routed, but the necessary condition of capacity availability is met with these quantities. This demonstrates that the *first question* of Chapter 3.2 can be answered without considering the routing of parts.

The example is simple for illustration. In real cases, other constraints, more machines, operation types, and products, minimum and maximum production requirements, and more complex objective functions can be formulated.

Table 3.5 provides test results from different size problems solved with the Lingo 6.0 mathematical programming software on a Pentium IV processor. It can be seen that, if only the number of machines increases, then the number of constraints does not change, and the change of CPU time is insignificant. Generally, however, an increase in the number of machines implies more alternative routes. Therefore, the number of operation types may increase. For example, for 8 operation types, 510 lower and upper bound constraints are necessary, but the required CPU time to solve the problem is still very small, 8 or 9 seconds.

Our general experience with solving test problems is that as a consequence of increasing the number of operation types, the increase in computation time is not significant. However, an increase in the number of operation type sets and the memory requirements to generate and store the S'_k and S''_k sets for large problems require careful data management. Even for large problems, however, the capacity constraints are easily formulated with equations (3.8) and (3.9).

Table 3.5 Test results of different size problems
(Lingo 6.0 solver, Pentium IV processor)

Test problems	Number of					CPU time (mm:ss)
	Operation types (H)	Machines (M)	Operation type sets (K)	Constraints	Iterations	
1	4	4	15	30	25	0:01
2	4	5	15	30	15	0:01
3	6	4	63	126	33	0:01
4	6	5	63	126	37	0:01
5	8	4	255	510	90	0:08
6	8	5	255	510	167	0:09
7	8	6	255	510	115	0:09

Finally, note that although the results are route independent, the model is part mix dependent. If new part types are introduced or existing part types are completed, the operation types and type sets may change. However, part mix change, if it exists, can be easily handled by the presented approach. Two possible decision-making scenarios can be envisioned.

a) *Part mix is stable*. Process plans are known, operation types and operation type sets are formed, and management has to answer product mix, maximum production quantity, and capacity availability related questions. In this scenario, there is no part mix change, and the application is straightforward. If the decision-making period is short enough, this scenario is valid.

b) *Frequent part mix changes*. As the mix changes, new operation types have to be determined and new operation type sets have to be formed. Since operation types are easily determined (see the definition in Chapter 3.5) and operation type set formation is straightforward, the extra computational work can be easily performed. Even if part mix frequently changes, our previous remarks concerning the number of operation type sets are valid. Therefore, frequent part mix change does not increase the size of the problem. It only increases the frequency of model update.

This problem, however, exists in the case of machine capacity (route dependent) type models as well; if part mix changes, a new model has to be built. The route independent modeling, however, results in simpler models as compared to route dependent models. To illustrate this, we use the example of Table 3.1 in Chapter 3.2. The number of variables and constraints are summarized in Table 3.6 for both route dependent and route independent approaches. It can be seen that in the route independent model, 2 integer variables and 24 constraints are needed. If a machine-based approach is used, then only 4 capacity lower bounds and 4 capacity upper bounds (8 constraints) are necessary. But for all of the possible manufacturing routes of the two products, an integer variable is needed. Then 10 integer variables are required to determine something (routing), which is not of interest to management at an early stage of the decision-making process.

Table 3.6 Comparison of problem sizes

	Route independent modeling	Route dependent modeling
Number of variables	2	10
Number of constraints	24	8

If more products can be manufactured on several routes, then the number of integer variables increases faster in the machine-based approach than in the operation type-based approach. The price of simplicity is the loss of information about the quantities on the different routes. But if a manager does not want a detailed plan, just information about the feasibility of producing some new orders, then the operation type-based approach is faster and simpler.

3.8 Sensitivity analyses of the operation type set constraints

Sensitivity analyses of the parameters of the model presented in Chapter 3.7 can help to analyze how certain changes affect the capacity over- and under-utilization of the manufacturing system. The capacity requirement (rt_h) of an operation type and the machine capacity (c_m) are the most important parameters altered by unexpected changes. For these parameters, sensitivity analysis can determine the *validity range* of a chosen parameter within which the capacity requirement remains feasible, that is, it remains within the extended available capacity range. This sensitivity range can be determined by the *feasible increase* and by the *feasible decrease* of a given parameter. Like most software packages in linear programming, we provide sensitivity analysis for only one parameter at a time.

3.8.1 Sensitivity of operation type set constraints to capacity requirements

Changes concerning orders may result in changes of their capacity requirements. A customer may request an increase of an order. Another customer may cancel an entire order. A customer may require a small modification of a part, which may result in, for example, more (or less) drilling time. In case of a rush order, the capacity requirements of many operation types may change. All of these changes affect the capacity requirements of certain operations, which affects the capacity requirements of the associated operation type. The sensitivity analysis of the capacity requirements of an operation type can help analyze the consequences of these situations.

The sensitivity range of a particular capacity requirement, rt_h, can be determined by calculating the possible change of rt_h for all operation type sets that contain operation type h. That is, a range is computed for all S_k, when $ot_h \in S_k$. The feasible decrease of the capacity requirements of operation type h, Δrt_h^-, is determined by the minimum of the algebraic differences between the capacity requirements and the capacity lower bounds for all operation type sets that contain operation type h, that is,

$$\Delta rt_h^- = \underset{(k|ot_h \in S_k)}{Min} \left[rs_k - l_k (1-\alpha) \right] \qquad h = 1, \ldots, H; \quad k = 1, \ldots, K \tag{3.11}$$

The feasible increase of the capacity requirement of operation type h, Δrt_h^+, is determined by the minimum of the algebraic differences between the capacity upper bounds and the capacity requirements for all operation type sets that contain operation type h, that is,

$$\Delta rt_h^+ = \underset{(k|ot_h \in S_k)}{Min} \left[u_k (1+\beta) - rs_k \right] \qquad h = 1, \ldots, H; \quad k = 1, \ldots, K \tag{3.12}$$

The results obtained from equations (3.11) and (3.12), when applied to the optimum solution of the sample problem (x_1=101, x_2=23), are given in Table 3.7.

Table 3.7 Sensitivity of operation type set constraints to capacity requirements of operation types (α=β=0.25)

Operation type	rt_h	Δrt_h^-	Δrt_h^+
ot_1	0.87	0.066	0.935
ot_2	0.70	0.066	0.552
ot_3	1.25	0.497	0.004
ot_4	1.25	1.062	0.001

The sensitivity ranges are valid for 25% acceptable capacity over- and under-utilization (α=β=0.25). Table 3.7 shows that the capacity requirements of ot_1 can be decreased by 0.066 CUs (26.73 minutes) without violating the lower capacity bound (Δrt_1^-=0.066). This value is found at S_5, when equation (3.11) is applied. On the other hand, the feasible increase of this operation type set is much higher. The minimum of equation (3.12) is found at S_{12} (Δrt_1^+= 0.935 CUs=378.675 minutes).

The same value is obtained for the feasible decrease of ot_2 (Δrt_2^-=0.066 CUs=26.73 minutes) indicating that a small decrease in the capacity requirements of these operation types will not cause capacity under-utilization. The feasible increase of ot_2 is 0.552 (223.56 minutes), which is much less than the feasible increase of ot_1.

The opposite is true for ot_3 and ot_4. For these capacity requirements, there is a large possibility for decrease (Δrt_3^-=0.497 CUs=201.285 minutes, Δrt_4^-=1.062 CUs=430.11 minutes). But changes in orders that result in an increase in capacity requirements are unacceptable because of lack of capacity (Δrt_3^+=0.004 CUs=1.62 minutes, Δrt_4^+=0.001 CUs=0.405 minutes).

The results in Table 3.7 are independent validity ranges, that is, a feasible decrease or feasible increase is valid only if the capacity requirements of a single operation type change. If the capacity requirements of more than one operation type change, a joint range for all of the simultaneously changing parameters has to be determined. The sensitivity space of several simultaneously changing capacity requirements can be determined by calculating the possible change of all operation type sets that contain these operation types. The result is a multi-dimensional space described by the resulting inequalities.

Finally, note that the values of a feasible increase or a feasible decrease of the capacity requirements of an operation type can be negative as well. This indicates to management the infeasibility of a production plan.

Using the results of Table 3.7, a *route-independent* answer can be given to the *second question* of Chapter 3.2. If a customer requires a modification of a product, and this modification changes the operation times of a specific operation type, then the feasibility of this modification can be checked with the help of the sensitivity range of that operation type.

3.8.2 Sensitivity of the operation type set constraints to machine capacity

Machine capacity may decrease because of machine breakdowns, scheduled maintenance, unexpected production stops, or waiting for operators, repairpersons, tools, or materials. A capacity increase can occur from scheduled overtime or extra shifts. Sensitivity analysis of machine capacity can help analyze both benefits and consequences of these situations.

The sensitivity range of the available capacity of a particular machine can be determined by calculating the feasible change of the upper and lower capacity bounds of all of those operation type sets that are affected by the changes in the operation type sets assigned to that machine. For example, if a machine is tooled just for drilling, then the feasible changes of the upper and lower capacity bounds of all of the operation type sets which contain drilling have to be examined.

The capacity decrease of a machine diminishes both lower and upper capacity bounds. For our purposes, the decrease of an upper bound is relevant, because it may result in an infeasible capacity over-utilization. When the capacity of a particular machine changes, then all of the capacity upper bounds of those operation type sets, which contain any and all of the operation types assigned to this machine, are affected. The feasible decrease of capacity of machine m, Δc_m^-, is determined by the minimum of the algebraic differences between the capacity upper bound and the capacity requirements for all of the affected operation type sets, that is,

$$\Delta c_m^- = \underset{\left(k \middle| z_{k''m}=1 \cap S_{k''} \in S''_k\right)}{Min} \left[u_k(1+\beta) - rs_k\right] \qquad m = 1,\dots,M; \quad k = 1,\dots,K; \quad k'' = 1,\dots K'' \tag{3.13}$$

The capacity increase of a machine augments both the lower and upper capacity bounds. For our purposes, the increase of the lower bound may be relevant, because it may result in infeasible capacity under-utilization. When the capacity of a machine changes, all of those capacity lower bounds of operation type sets, for which the operation type set assigned to the machine is a subset, are affected. The feasible increase of the capacity of machine m, Δc_m^+, is determined by the minimum of the algebraic differences between the capacity requirements and the capacity lower bound for all of the affected operation type sets, that is,

$$\Delta c_m^+ = \underset{\left(k \middle| z_{k'm}=1 \cap S_{k'} \in S'_k\right)}{Min} \left[rs_k - l_k(1-\alpha)\right] \qquad m = 1,\dots,M; \quad k = 1,\dots,K; \quad k' = 1,\dots K' \tag{3.14}$$

The results computed for the optimum solution of the sample problem (x_1=101, x_2=23) using equations (3.13) and (3.14) are given in Table 3.8.

The sensitivity ranges are valid for 25% acceptable capacity over- and under-utilization (α=β=0.25). Table 3.8 shows that the capacity of M1 can be decreased without violating the upper capacity bounds (Δc_1^-=0.935 CUs=378.675 minutes). Only ot_1 is assigned to M1. Therefore, every operation type set that contains ot_1 must be selected, that is, S_1, S_5, S_6, S_7, S_{11}, S_{12}, S_{14}, and S_{15}. The difference between the 25% increase of the upper bounds and the capacity requirements of these operation type sets must be checked. The minimum of these differences is found at S_{12} when equation (3.13) is applied.

Table 3.8 Sensitivity of the operation type set constraints to machine capacity (α=β=0.25)

Machines	c_m	Δc_m^-	Δc_m^+
M1	1	0.935	0.066
M2	1	0.552	0.066
M3	1	0.004	0.497
M4	1	0.001	1.062

The feasible increase is much smaller, it is equal to 0.066 CUs (Δc_1^+=0.066 CUs=26.73 minutes), indicating that only a small possibility of increasing capacity would be feasible. Only ot_1 is assigned to M1. In this simple case, again every operation type set that contains ot_1 must be selected, that is, S_1, S_5, S_6, S_7, S_{11}, S_{12}, S_{14}, and S_{15}. The maximum value is found at S_5, when equation (3.14) is applied.

Table 3.8 shows that the capacity of M2 can also be decreased without violating the upper capacity bounds (Δc_1^-=0.552 CUs=223.56 minutes). Both ot_1 and ot_2 are assigned to M2. Therefore, every operation type set that contains ot_1 or ot_2 must be selected. The difference between the 25% increased value of the upper bounds and the capacity requirements of these operation type sets must be checked. The minimum of these differences is found at S_9 when equation (3.13) is applied.

The feasible increase is much smaller, it is equal to 0.066 CUs (Δc_1^+=0.066 CUs=26.73 minutes), indicating that only a small possibility of increasing capacity would be feasible. Both ot_1 and ot_2 are assigned to M2. Therefore, every operation type set that contains ot_1 and ot_2 must be selected, that is, S_5, S_{11}, S_{12}, and S_{15}. The 0.066 value is found at S_5 when equation (3.14) is applied.

The capacity of machines M3 and M4 cannot be decreased (Δc_3^-=0.004 CUs=1.62 minutes, Δc_4^-=0.001 CUs=0.405 minutes). On the other hand, their capacity can be increased considerably (Δc_3^+=0.497 CUs= 201.285 minutes, Δc_4^+=1.062 CUs= 430.11 CUs).

Note that in practice, the values of a feasible increase or a feasible decrease of machine capacity can be negative. This can indicate to managers a lack of or excess capacity, for a given production plan.

The results in Table 3.8 are independent validity ranges. That is, a feasible decrease or a feasible increase is valid only if the capacity of a single machine changes. If the capacity of more than one machine changes, a joint range for all of the simultaneously changing parameters has to be determined. The sensitivity range of the available capacity of several machines can be determined by calculating the feasible change of the upper and lower capacity bounds of all of those operation type sets that are affected by the change of the operation type sets assigned to the machines in question. The result is again a multi-dimensional space described by the resulting inequalities.

Using the results of Table 3.8, a *route-independent* answer can be given to the *third question* of Chapter 3.2. If, for example, scheduled maintenance decreases the capacity of M2 by less than 50%, then it would not affect the feasibility of the optimum production plan. Maintenance of M3 or M4, however, cannot be scheduled in this specific period.

The answer to the *fourth question* of Chapter 3.2 requires information from both Tables 3.7 and 3.8. If sensitivity ranges in Table 3.7 indicate capacity shortages for some operation type sets, overtime might be needed. The overtime can be applied at those machines whose sensitivity ranges show a lack of capacity in Table 3.8.

3.9 Conclusions of Chapter 3

In this chapter, a new method for the formulation of capacity constraint in FMSs is presented. This new formulation is based on the concept of operation types, and expresses the capacity of operation type sets, instead of the capacity of machines. The proposed method allows the route-independent evaluation of some capacity-related questions in FMSs.

There are two major application areas for the results provided by the presented approach. First, the product mix and sensitivity information may provide guidelines for on-line control. That is, disaggregating operation types into operations can be done by a real time dispatching and scheduling system. Details about how to do this are a subject for future research. Second, an aggregated plan can be disaggregated using a detailed routing and scheduling model (e.g., a disaggregation mathematical programming model). In both cases, however, the suggested quantities are analyzed and major part mix decisions are made with the presented approach, using it as a *rough cut* planning tool.

There are three main reasons to use the proposed methods. First, in an automated flexible manufacturing environment, routing often can be decided in real time. It is not necessary to determine the entire production routing far in advance of production.

Second, the approach and formulations presented in this chapter have major advantages, when a quick, route-independent estimation of available capacity is desired. When a decision maker would like to estimate whether the available capacity of a flexible system is enough to manufacture a set of orders, then it is not necessary (or maybe not even possible) to determine the detailed production plan containing route and machine assignments.

Third, the operation type-based approach can be complemented with a sensitivity analysis of the major parameters of a production system. How a change in machine capacity or a change in the capacity requirements of an operation type changes the feasibility of a production plan can easily be analyzed. In the traditional, machine-based approach, this sensitivity analysis can only be performed by the repeated solution of a mathematical programming model.

The route-independent formulation of capacity constraints in this chapter is for FMS production planning. However, this approach may have other application areas, when the simplification of capacity constraints provides benefits for operations managers, while the missing information about operation (routing) details is acceptable. For example, Farkas, Koltai and Stecke (1999) used the operation type concept for balancing workload of machines in several consecutive production periods in case of given orders. In Koltai, Farkas and Stecke (1998), Koltai, Farkas and Stecke (2001) and Koltai, Stecke and Juhasz, (2004), tooling of machines for a given production requirement is determined using operation type set capacity constraints.

As a summary, based on Chapter 3, the following scientific results can be formulated:

Result 2/1

I have defined the set of those operations, which can be performed on any machine in a particular group of machines, as *operation type*. A specific combination of different operation types is called an *operation type set*. The upper capacity bound of operation type set k (u_k) can be calculated with formula (3.1). The lower capacity bound of operation type set k (l_k) can be calculated with formula (3.2). I showed that there is enough capacity to manufacture a given quantity of parts independently of the specific routing of parts without unnecessary idle time of machines if conditions (3.8) and (3.9) are satisfied.

Result 2/2

The feasibility of a production plan with respect to the change of operation type requirements

(rt_h) can be determined by sensitivity analysis. If the change of an operation type requirement is within the feasible increase (Δrt_h^+) and the feasible decrease (Δrt_h^-), then there is enough capacity to produce the planned quantity without unnecessary idle time of machines. I have determined formula (3.11) for the calculation of the feasible *decrease* and formula (3.12) for the calculation of the feasible *increase* of operation type requirement *h*.

Result 2/3
The available capacity of operation types is determined by the capacity of the machines (c_m), and machine capacity may change during operation. The feasibility of a production plan with respect to machine capacity is determined by sensitivity analysis. If the change of machine capacity is within the feasible increase (Δc_m^+) and the feasible decrease (Δc_m^-), then there is enough capacity to produce the planned quantity without unnecessary idle time of machines. I have determined formula (3.13) for the calculation of the feasible decrease and formula (3.14) for the calculation of the feasible increase of machine capacity of machine *m*.

The importance of routing in FMSs and the effect of routing on capacity analysis is discussed in Guerrero et al. (1999), and Koltai et al. (2000). The introduction of the concept of operation type aggregation and the formulation of operation type set capacity constraints are presented in Koltai and Stecke (2008), and Koltai, Juhász and Stecke (2004). The application possibilities of operation type aggregation in different areas of operation analysis are explored in Koltai, Farkas and Stecke (1998, 2001), Farkas, Koltai and Stecke (1999), Koltai et al. (2004), and Koltai, Stecke and Juhász (2004).

4 FORMULATION OF WORKFORCE SKILL CONSTRAINTS IN ASSEMBLY LINE BALANCING MODELS

Assembly lines are generally dedicated to the production of one or a few similar products in large quantities. The production capacity of an assembly line is strongly influenced by the allocation of tasks to workstations. The tasks assignment to workstations influences the output rate, and consequently the cycle time as well. One important element of production planning of assembly lines is, therefore, the optimal assignment of tasks to workstations. To solve this problem, assembly line balancing (ALB) models are used. Traditional assembly line balancing is generally described as a 0-1 mathematical programming problem. In this chapter a general framework is provided to complete ALB models with workforce skill constraints. The example of a bicycle assembly process shows, how the consideration of workforce skill conditions influences task assignment. The sensitivity of the optimal assignment with respect to the change of production quantity is also presented. The results of this chapter are based on the papers of Koltai and Tatay (2008), Koltai and Tatay (2013) and Koltai, Tatay and Kalló (2013).

4.1 Introduction

Assembly line balancing (ALB) problems occur when several indivisible work elements (tasks) are to be grouped into (work)stations along a continuous production line. Workers may work at each station, and in case of efficient allocation of tasks to workstations, the number of workers and consequently the cost of operation can be decreased. Application of assembly line balancing can be found frequently, for example, in the automobile, electronic, and clothing industry (see Chan et al., 1998; Sawik, 2002; Lapierre and Ruiz, 2004 and Cortes, Onieva and Guadix, 2010). The operation of some service systems, however, is also very similar to assembly line operations (Scholl and Becker, 2006; Boysen, Fliedner and Scholl, 2008).

Tasks cannot be allocated to the stations arbitrarily. Capacity constraints, precedence relations – generally visualized by a precedence graph –, zoning conditions, technological and logical requirements may influence the optimal assignment. Even considering these restrictions many feasible solutions may exist for the allocation of tasks to workstations and optimization models can be used to find the best task assignment.

A simple ALB problem is illustrated in Figure 4.1. This problem is published in an early paper of Bowman (1960), and with some changes, it will be used to illustrate the proposed method in this chapter as well.

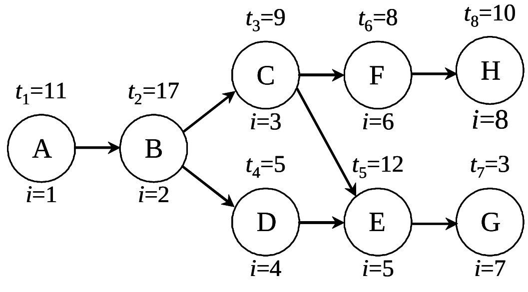

Figure 4.1 Precedence diagram of the sample problem

The time of each operation (t_i) is considered deterministic. The indicated 8 tasks in Figure 4.1 must be assigned to workstations. At each workstation one worker performs all the tasks assigned to the station. Precedence relations, indicated by the arrows in the figure, must be considered at task assignment. The time required to perform all the tasks assigned to a station is the station time (s_j). The workstation with the highest station time is the bottleneck of the system. The station time of the bottleneck station is called *cycle time* (T_c) which determines the production capacity of the assembly line. Several objectives and additional constraints can be considered when the tasks are assigned to workstations.

Early research in this area focused on the simple assembly line balancing problem (SALBP) with its restrictive characteristics such as deterministic task times, no assignment restrictions other than the precedence constraints, serial line layout, etc (Becker and Scholl, 2006; Scholl and Becker, 2006). Extended forms of the SALBP consider for example the possibility of U-shaped lines, parallel stations, and stochastic task times. These models are referred in the literature as general assembly line balancing problems (GALBP). GALBPs may be closer to practical problems, and their solution procedures, in most cases, are based on SALBP algorithms (Scholl and Becker, 2006). Depending on the management objective of assembly line balancing, the two most frequently used versions of SALBPs are the following,

– When management objective is related to operating cost reduction the ALB model minimizes the number of workstations (workers) for a given cycle time. The related problems are referred in the literature as SALBP-1.

– When management objective is related to production quantity the ALB model minimizes the cycle time for a given number of workstation. The related problems are referred in the literature as SALBP-2.

SALBPs can be formulated as mathematical programming models. The first analytical formulation of ALB was given by Bryton (1954) and the first linear programming problem that might have infeasible solutions because of split tasks was given by Salveson (1955). Bowman was the first to suggest integer programming (IP) models to solve the classical ALB problem (Bowman, 1960). Whiten (1961) modified Bowman's IP model and defined 0-1 decision variables for the problem. Since ALB models are NP hard the research in the past focused on reducing the number of variables and constraints in order to reduce the complexity of the models (see for example Thangavelu and Shetty, 1971; Patterson and Albracht, 1975; Baybars, 1986 and Scholl and Becker, 2006).

Today mathematical programming models of practical size ALB problems can be solved by optimization software very efficiently. Therefore, the focus of research should be shifted to practice driven model formulation and to the investigation of new areas of application (Boysen, Fliedner and Scholl, 2008). One of the possibilities of increasing the relevance of ALB models is the consideration of worker skill conditions. There are not too many papers which are dedicated to the consideration of skill constraints.

Johnson (1983) applies some very simple skill constraints in a paper dedicated mostly to some mathematical questions of the optimization process.

Wong, Mok and Leung (2006) used the concept of skill inventory in an apparel assembly process to organize the proper assignment of tasks to workers and to workstations. This concept, however, was used in an on-line control mechanism, and not in an assembly line balancing optimization model.

Miralles et al. (2007) used skill constraint in a production environment for disabled workers. Different task times for the same tasks expressed different skill levels and workstations with similar skill levels were formed. Later this model was extended with the possibility of job rotation as well (Cosat and Miralles, 2009).

Corominas, Pastor and Plans (2008) considered temporary and permanent workers in a motor-cycle assembly process, and these two worker groups are able to perform different set of tasks.

Moon, Logendran, and Lee (2009) considered an assembly line in which multi-functional workers are applied with different salaries, and one of their objectives was to minimize the total annual workstation cost.

Cortes, Onieva and Guadia (2010) prepared the assembly line balancing model of a motorcycle assembly process with homogeneous workers groups. The complexity of the model, however, required the application of sophisticated heuristics to get a feasible solution.

There are models, which consider the change of skill level during the assembly process. The decrease of task time can be attributed to the learning effect (Cohen, Vitner and Sarin, 2006), and the increase of task time can be the consequence of technological and physiological reasons (see for example Toksari et al., 2010 and Emrani et al., 2011). In these cases, however, skill constraints were not added to the ALB models, the change of skill level is embedded in task time functions.

This chapter is structured as follows. First, formulation of the basic ALB models used in this chapter is provided. Next, skill constraints are generalized and the mathematical description of the different skill conditions is given. The results of the suggested models are illustrated with the help of the production process of a bicycle manufacturer. The sensitivity of the optimal assignment with respect to the change of production quantity is analyzed with the production quantity/efficiency chart. All notations used in this chapter are summarized in Table 4.1.

4.2 Formulation of the basic simple ALB models

Tasks are numbered in increasing order. The number i assigned to a task is called the task index. We refer to a task either by its name or by its task index. Those tasks which are not succeeded by any other task are called *last* tasks. The index set of last tasks is denoted by L.

Workstations are also numbered in increasing order. The first workstation is numbered 1 and the last workstation is numbered N. The number j assigned to a workstation is called the workstation index. Workstations are referred in the following by the workstation index. An assumption must be made about the possible number of stations prior to task assignment. The number of stations used in the model is J. That is, J is the number of stations used in the mathematical model, and N is the number of stations used in the actual line.

The assignment of tasks to workstations is expressed with binary decision variable x_{ij}. If task i is assigned to workstation j, then x_{ij}=1, otherwise x_{ij}=0.

In this chapter the following integer linear programming formulation of SALBP-1 is used,

$$\mathrm{Min}(N) \tag{4.1}$$

$$\sum_{i=1}^{I} t_i x_{ij} \leq T_\mathrm{c} \qquad j=1,\ldots,J \tag{4.2}$$

$$\sum_{j=1}^{J} x_{ij} = 1 \qquad i=1,\ldots,I \tag{4.3}$$

$$\sum_{j=1}^{J} j \cdot \left(x_{qj} - x_{pj}\right) \geq 0, \qquad (p,q) \in R \tag{4.4}$$

$$N \geq \sum_{j=1}^{J} \left(j \cdot x_{ij}\right) \qquad i \in L \tag{4.5}$$

$$x_{ij} = 0 \quad j < LT_i \text{ and } j > UT_i \quad i=1,\ldots,I \tag{4.6}$$

The objective of the model is to minimize the number of stations used in the actual system; that is, to minimize the largest index belonging to a station with task assignment.

Table 4.1 Summary of notation of Chapter 4

Subscripts:		
i	–	index of tasks (i=1,…,I),
p	–	index of a subset of tasks,
q	–	index of a subset of tasks,
v	–	index of a subset of tasks,
j	–	index of workstations (j=1,…,J),
k	–	index of skill level (j=1,…,K).
Parameters:		
I	–	number of tasks,
J	–	number of workstations in the mathematical model,
N	–	actual number of workstations applied,
R	–	set of pair of indices which belong to tasks with precedence relations, that is, $(p;q)\in R$, if task p immediate precedes task q,
K	–	number of skill levels,
t_i	–	time necessary to perform task i (task time),
s_j	–	time necessary to perform all tasks at station j (station time),
$s_j(Q)$	–	station time of station j as a function of production quantity,
T_c	–	cycle time of the assembly line,
T	–	total available time for production,
LT_i	–	the earliest workstation which can be used as a consequence of preceding tasks of task i,
UT_i	–	the latest workstations which can be used by task i as a consequence of succeeding tasks of task i,
LS_k	–	the earliest workstation which can be used by tasks belonging to skill level k as a consequence of preceding tasks,
US_k	–	the latest workstations which can be used by tasks belonging to skill level k as a consequence of preceding tasks,
Q	–	production quantity,
$Q(j–d, j)$	–	production quantity at which station $j–d$ enters, and station j leaves the bottleneck,
c_j	–	capacity utilization of station j,
W_k	–	limit on special workers with skill level k,
z	–	sufficiently high number (higher than I),
$E(Q,N)$	–	efficiency of an assembly line with N workstation at Q production quantity,
$Q_{\mathrm{Max}}(N)$	–	maximal production quantity of a line configuration with N stations,
$Q_{\mathrm{Max}}^{\mathrm{OPT}}(N)$	–	maximal production quantity of the optimal line configuration with N stations,
b_j	–	power of the learning curve function at station j,
d	–	distance of the station index of two stations.
Sets:		
L	–	set of final tasks, that is, $i\in L$ if task i does not precede any other tasks,
R	–	set of the index pairs of immediately preceding tasks,
P_i	–	index set of those tasks which must be finished before task i is started,
F_i	–	index set of those tasks which cannot be started before task i is finished,
S_k	–	index set of tasks belonging to skill level/type k.
Decision variables:		
N	–	number of workstations applied,
x_{ij}	–	0-1 variable; if x_{ij}=1, then task i is assigned to workstation j, otherwise x_{ij}=0,
l_{jk}	–	0-1 decision variable in case of low-skill constraints; if l_{jk}=1, worker with skill level k is applied at workstation j, otherwise l_{jk}=0,
h_{jk}	–	0-1 decision variable in case of high-skill constraints; if h_{jk}=1, then worker with skill level k is applied at workstation j, otherwise h_{jk}=0,
e_{jk}	–	0-1 decision variable in case of exclusive-skill constraints; if e_{jk}=1, then worker belonging to skill type k is applied at workstation j, otherwise e_{jk}=0.

The right-hand side of constraint (4.5) determines the index of those workstations which perform last tasks. The highest such index must be minimized. If each of these indices is smaller than or equal to *N*, and *N* is minimized, then the index of the final workstation, and consequently the number of workstations, is minimized.

Cycle time constraints are expressed by constraints (4.2). For each workstation the sum of task times of the assigned tasks is not allowed to exceed the cycle time. As a consequence of constraints (4.3) each task is assigned to one of the workstations.

Constraints (4.4) express the precedence constraints. If task *p* must be performed before task *q*, the difference in the bracket is equal to -1, 0 or 1 for each workstation. Since task *p* must be assigned to an earlier or to the same workstation as task *q*; the weighted sum of these differences is always greater than or equal to 0, if the weights are the indices of the corresponding workstations.

Finally, the number of variables is reduced by constraints (4.6). Some tasks cannot be assigned to very *early* workstations because of preceding tasks. For example, if in the problem indicated by Figure 4.1, the required cycle time is 25 minutes then the earliest station for task B is the second station. On the first station, the sum of task times of tasks A and B (11+17= 28 minutes) would violate the cycle time constraint. The earliest workstations which can be used by task *i* is determined by LT_i. LT_i is a lower limit of the feasible station indices of task *i*, and its value is calculated as follows,

$$LT_i = \left\lceil \frac{t_i + \sum_{k \in P_i} t_k}{T_c} \right\rceil \tag{4.7}$$

where $\lceil x \rceil$ is the smallest integer value not smaller than *x*.

Some tasks cannot be assigned to very *late* workstations because of succeeding tasks. For example, if in the problem indicated by Figure 4.1, the required cycle time is 35 minutes then the latest station for task C is the last but one station. On the last station, the sum of task times of task C and the succeeding tasks (F, E, H, G) would violate the cycle time constraint. The latest workstation which can be used by task *i* is determined by UT_i. UT_i is an upper limit of the feasible station indices of task *i*, and its value is calculated as follows,

$$UT_i = J + 1 - \left\lceil \frac{t_i + \sum_{k \in S_i} t_k}{T_c} \right\rceil \tag{4.8}$$

(4.1)-(4.8) is a linear programming model with several 0-1 variables. The required number of binary variables can be determined using the LT_i and UT_i values with the following formula,

$$\sum_{j=1}^{J} \left(UT_i + 1 - LT_i \right) \tag{4.9}$$

We note that model (4.1)-(4.8) is slightly different from the models used in the literature. Most models formulate the problem for a single *last* task, that is, only one index belongs to *L* (see for example White, 1961). If several final tasks exist (see the sample problem in Figure 4.1) then a dummy task is used which directly succeeds the real final tasks. This dummy task increases the number of 0-1 variables; because in that case *I*+1 task must be assigned to *J* workstations. In formulation (4.1) to (4.8), however, instead of the dummy task, the index of the final workstation is used. This way only one new variable (*N*) is required. The value of *N* must be integer, but as a consequence of the integer lower bound and of the minimization objective, *N* can be considered as continuous variable.

SALBP-2 minimizes the cycle time for a given number of workstations (N), that is, the objective function is as follows,

$$\text{Min } T_c \tag{4.10}$$

Cycle time constraint (4.2), constraints for the performance of each operation (4.3) and precedence constraints (4.4) are the same as in SALBP-1. That is, SALBP-2 is determined by objective function (4.10) and constraints (4.2)-(4.4). The limit on the number of variables in this case can be determined by using an estimate of the upper bound of the cycle time ($UB(T_c)$). A trivial upper bound of T_c is the sum of task times, however, generally more efficient approximations can be found. For example, the cycle time of the optimal solution of a corresponding SALBP-1 can be used to determine an upper bound for T_c.

The earliest workstations which can be used by task i is now the following,

$$LT_i = \left\lceil \frac{t_i + \sum_{k \in P_i} t_k}{UB(T_c)} \right\rceil \tag{4.11}$$

The latest workstation which can be used by task i is now calculated as follows,

$$UT_i = J + 1 - \left\lceil \frac{t_i + \sum_{k \in S_i} t_k}{UB(T_c)} \right\rceil \tag{4.12}$$

Pastor and Ferrer (2009) published an improved method for the calculation of the feasible lower and upper workstation indices. Their method increases computational efficiency in case of large problems. In the problems presented in the following chapters, however, the estimate of the feasible workstation indices with formula (4.7), (4.8), (4.11) and (4.12) is sufficient, because computation time is insignificant.

Consequently SALBM-1 is defined by constraints (4.1)-(4.8) and SALBM-2 is defined by constraints (4.2)-(4.4), (4.6) and (4.10)-(4.12). These models are summarized in the first row of Table 4.2.

In the following chapter the basic SALBP-1 and SALBP-2 models will be completed with constraints expressing work force skill requirements.

4.3 Formulation of workforce skill constraints

Frequently, a set of tasks performed at an assembly line requires special skills of workers, and a set of workers working at an assembly line may have special or limited skills. This must be considered when tasks are assigned to workstations.

It is assumed that each worker is assigned to a skill level k, k=1,…,K. For each task the minimum skill level necessary to perform the task is determined. The index set of those tasks which require skill level k is denoted by S_k. Three different types of skill constraints can be distinguished (Koltai and Terlaky, 2011, 2013).

– *A limited number of workers belonging to skill level k must be applied* at the assembly line. In this case, there are workers who are not able to perform each task. Workers with the *lowest* skill level (k=1) perform only the simplest tasks. Workers with the highest skill level (k=K) can perform the most complicated tasks as well. A worker with skill level k can only perform tasks requiring skill level smaller than or equal to k, and are not able to perform tasks which require skill level higher than k. Consequently, a worker with skill level k can only work at stations which have tasks with skill level equal to or lower than k, and the number of such stations is constrained from *below*. We call the constraint describing this situation low-skill constraint (LSC).

– *Only a limited number of workers are able to perform the most complicated tasks.* In this case there are tasks which require qualified workers. There are only a limited number of workers available to perform such tasks. Workers with the *highest* skill level (k=1) perform the most complicated tasks. Workers with the lowest skill level (k=K) can perform only the simplest tasks. A worker with skill level k can perform tasks requiring skill level higher than or equal to k, and are not able to perform tasks which require skill level smaller than k. Consequently, a worker with skill level k can only work at stations which have tasks with skill level equal to or lower than k, and the number of such stations is constrained from *below*. We call the constraint describing this situation high-skill constraint (HSC).

– *Some tasks can be performed only by special workers*. In this case workers have different skills/specializations, and a worker specialized in one type of skill, is not able to perform tasks requiring other type of skills. Tasks are grouped according to skill requirements, and at a workstation only tasks belonging to a given skill type can be performed. Since a worker working at a station can perform exclusively those tasks which correspond to his/her qualification, we call the constraint describing this situation exclusive-skill constraint (ESC).

4.3.1 Formulation of low-skill constraints (LSC)

In this case each task is assigned to the lowest skill level necessary to perform the task. Index set S_k contains the index of those tasks which require workers with skill level k. The lowest skill level belongs to k=1. The binary skill variable l_{jk} is used to indicate worker assignment. If l_{jk}=1, then worker with skill level k is assigned to workstation j, otherwise l_{jk}=0. In case of LSC any worker with skill level k is capable to perform those tasks, which require skill level smaller than or equal to k, that is the following constraints must be satisfied,

$$\sum_{i \in S_k} x_{ij} \le z \sum_{v=k}^{K} l_{jv} \qquad j = 1,..., J; \quad k = 1,..., K \tag{4.13}$$

If tasks belonging to skill level k are assigned to workstation j, then the left-hand side of constraint (4.13) is higher than zero, and consequently the right-hand side must be higher than zero as well. If z is a sufficiently high number, then a skill variable belonging to skill level k or higher must be equal to 1 in the right-hand side of equation (4.13).

According to (4.13) more than one skill variable belonging to workstation j may have non-zero value. Since only one worker must be assigned to each workstation, the following constraints must be added,

$$\sum_{k=1}^{K} l_{jk} \le 1 \qquad j = 1, \ldots, J \tag{4.14}$$

According to (4.14), the sum of the skill variables belonging to workstation j is either equal to zero, or equal to 1, that is, the maximum number of workers assigned to workstation j is equal to 1.

In some cases tasks are not assigned to a workstation at all. In SALBP-1, for example, at the beginning of the calculation an upper bound (J) is used for the total number of workstations, and finally the optimal number of workstations is equal to N. Consequently, no tasks are assigned to J–N workstations. The skill variable at these workstations must be equal to zero, that is,

$$\sum_{i=1}^{I} x_{ij} \ge l_{jk} \qquad j = 1,..., J; \quad k = 1,..., K \tag{4.15}$$

According to (4.15), if tasks are not assigned to workstation j, then the left-hand side is equal to 0, and consequently the skill variables on the right-hand side are also equal to 0.

Finally, a given number of workers with skill level k (W_k) must be applied, that is,

$$\sum_{j=1}^{J} l_{jk} \geq W_k \qquad k=1,\ldots,K \tag{4.16}$$

According to (4.16) the sum of workstations with nonzero k level skill variables must be higher than or equal to the available number of workers with skill level k. In this case the focus is on the application of low-skilled workers. For example, we may have just two skill levels (K=2), that is, k=1 for unskilled workers, and k=2 for skilled workers. If W_1>0 and W_2=0, then W_1 number of workstations with tasks for unskilled workers will be definitely applied, and skilled workers work at the rest of the workstations.

4.3.2 Formulation of high-skill constraints (HSC)

Each task is assigned to the lowest skill level necessary to perform the task again. Index set S_k contains the index of those tasks which require workers with skill level k. Now, the *highest* skill level belongs to k=1. The binary skill variable h_{jk} is used to indicate worker assignment. If h_{jk}=1, then worker with skill level k is assigned to workstation j, otherwise h_{jk}=0. In case of HSC any worker with skill level k is capable to perform those tasks, which require skill level lower than or equal to k, that is, the following constraints must be satisfied,

$$\sum_{i \in S_k} x_{ij} \leq z \sum_{v=1}^{k} h_{jv} \qquad j=1,\ldots,J; \quad k=1,\ldots,K \tag{4.17}$$

If tasks belonging to skill level k are assigned to workstation j, then the left-hand side of constraint (4.17) is higher than zero, and consequently the right-hand side must be higher than zero as well. If z is a sufficiently high number, then a skill variable belonging to skill level k or lower must be equal to 1 in the right-hand side of equation (4.17).

According to (4.17) more than one skill variable may have non-zero value. Since only one worker must be assigned to each workstation, the following constraints must be added,

$$\sum_{k=1}^{K} h_{jk} \leq 1 \qquad j=1,\ldots,J \tag{4.18}$$

According to (4.18), the sum of the skill variables belonging to workstation j is either equal to zero, or equal to 1, that is, the maximum number of workers assigned to workstation j is equal to 1.

In some cases tasks are not assigned to a workstation at all. In SALBP-1, for example, at the beginning of the calculation an upper bound (J) is used for the total number of workstations, and finally the optimal number of workstations is equal to N. Consequently, no tasks are assigned to J–N workstations in the calculation. The skill variable at these workstations must be equal to zero, that is,

$$\sum_{i=1}^{I} x_{ij} \geq h_{jk} \qquad j=1,\ldots,J; \quad k=1,\ldots,K \tag{4.19}$$

According to (4.19), if tasks are not assigned to workstation j, then the left-hand side is equal to 0, and consequently the skill variables on the right-hand side are also equal to 0.

Finally, no more than the available number of workers with skill level k can be applied, that is,

$$\sum_{j=1}^{J} h_{jk} \leq W_k \qquad k=1,\ldots,K \tag{4.20}$$

According to (4.20) the sum of workstations with nonzero k level skill variables must be lower than or equal to the available number of workers with skill level k. In this case the focus is on the application of high-skilled workers. For example, we may have just two skill levels

(K=2), that is, k=1 for skilled workers, and k=2 for unskilled workers. If W_1>0 and $W_2=\infty$, then no more than W_1 workstations with tasks for skilled workers can be organized, and unskilled workers work at the rest of the workstations.

4.3.3 Formulation of exclusive-skill constraints (ESC)

This case is found in practice when there are special tasks, which require special qualification of workers. The workers with the required qualifications can only perform these special tasks. Tasks requiring the same skill are assigned to skill type k (or keeping the previously used terminology, to skill level k). The index set of the tasks belonging to skill type k is S_k. The binary skill variable e_{jk} is used to indicate worker assignment. If e_{jk}=1, then worker with skill type k is assigned to workstation j, otherwise e_{jk}=0.

Tasks belonging to different skill type cannot be mixed on a workstation. To satisfy this condition two group of constraints must be satisfied.

1. If tasks *belonging* to skill type k are assigned to workstation j, then skill variable e_{jk} must be equal to 1, that is,

$$\sum_{i\in S_k} x_{ij} \le z e_{jk} \qquad j=1,...,J;\quad k=1,...,K \tag{4.21}$$

If tasks belonging to skill type k are assigned to workstation j then the left-hand side of (4.21) is higher than 0, and the right-hand side must be higher than 0 as well. If z is a sufficiently high number, then the right-hand side of (4.21) is higher than zero only if e_{jk} is equal to 1. If tasks belonging to skill type k are not assigned to workstation j, then the left-hand side of (4.21) is equal to 0 and the skill variable e_{jk} on the right-hand side can be either 0 or 1.

2. If tasks *not belonging* to skill type k are assigned to workstation j, then skill variable e_{jk} must be equal to 0, that is,

$$\sum_{i\notin S_k} x_{ij} \le z\left(1-e_{jk}\right) \qquad j=1,...,J;\quad k=1,...,K \tag{4.22}$$

If tasks not belonging to skill type k are assigned to workstation j then the left-hand side of (4.22) is higher than 0, and the right-hand side must be higher than 0 as well. If z is a sufficiently high number, then the right-hand side of (4.22) is higher than zero only if e_{jk} is equal to 0. If tasks not belonging to skill type k are not assigned to workstation j, then the left-hand side of (4.22) is equal to 0 and the skill variable e_{jk} on the right-hand side can be either 0 or 1.

If (4.21) and (4.22) are simultaneously satisfied, then the different groups of tasks are separated on the workstations, and the proper worker skill is applied at each station.

4.3.4 Summary of the suggested worker skill models

Table 4.2 summarizes the simple assembly line balancing models and the corresponding worker skill constraints. The basic models are presented in the first row of the table. SALBP-1 is an integer linear programming model and it is given in the first column. SALBP-2 is a 0-1 linear programming model and it is given in the second column. Note, that if there is only a single final task in SALBP-1, then the right-hand side of (4.5) can be directly minimized, and consequently there is no need for variable N. The workstation index limits (LT_i and UT_i) are calculated with expressions (4.7) and (4.8) or (4.11) and (4.12) respectively.

Table 4.2 Summary of ALB models and skill constraints

	Minimization of workstations (SALBP-1)	Minimization of cycle time (SALBP-2)
Basic models	$\text{Min}(N)$ $\sum_{i=1}^{I} t_i x_{ij} \le T_c \quad j=1,\dots,J$ $\sum_{j=1}^{J} x_{ij} = 1 \quad i=1,\dots,I$ $\sum_{j=1}^{J} j \cdot (x_{qj} - x_{pj}) \ge 0 \quad (p,q)\in R$ $N \ge \sum_{j=1}^{J} (j \cdot x_{ij}) \quad i \in L$ $x_{ij} = 0 \quad j < LT_i,\ j > UT_i,\ i=1,\dots,I$	$\text{Min}(T_c)$ $\sum_{i=1}^{I} t_i x_{ij} \le T_c \quad j=1,\dots,J$ $\sum_{j=1}^{J} x_{ij} = 1 \quad i=1,\dots,I$ $\sum_{j=1}^{J} j \cdot (x_{qj} - x_{pj}) \ge 0 \quad (p,q)\in R$ $x_{ij} = 0 \quad j < LT_i,\ j > UT_i,\ i=1,\dots,I$
Low-skill constraints (LSC)	$\sum_{i\in S_k} x_{ij} \le z \sum_{v=k}^{K} l_{jv} \quad j=1,\dots,J; \quad k=1,\dots,K$ $\sum_{k=1}^{K} l_{jk} \le 1 \quad j=1,\dots,J$ $\sum_{i=1}^{I} x_{ij} \ge l_{jk} \quad j=1,\dots,J; \quad k=1,\dots,K$ $\sum_{j=1}^{J} l_{jk} \ge W_k \quad j=1,\dots,J; \quad k=1,\dots,K$ $l_{jk} = 0 \quad j < LS_k$ and $j > US_k, \quad k=1,\dots,K$	
High-skill constraints (HSC)	$\sum_{i\in S_k} x_{ij} \le z \sum_{v=1}^{k} h_{jv} \quad j=1,\dots,J; \quad k=1,\dots,K$ $\sum_{k=1}^{K} h_{jk} \le 1 \quad j=1,\dots,J$ $\sum_{i=1}^{I} x_{ij} \ge h_{jk} \quad j=1,\dots,J; \quad k=1,\dots,K$ $\sum_{j=1}^{J} h_{jk} \le W_k \quad j=1,\dots,J; \quad k=1,\dots,K$ $h_{jk} = 0 \quad j < LS_k$ and $j > US_k, \quad k=1,\dots,K$	
Exclusive-skill constraints (ESC)	$\sum_{i\in S_k} x_{ij} \le z e_{jk} \quad j=1,\dots,J; \quad k=1,\dots,K$ $\sum_{i\notin S_k} x_{ij} \le z(1-e_{jk}) \quad j=1,\dots,J; \quad k=1,\dots,K$ $e_{jk} = 0 \quad j < LS_k$ and $j > US_k \quad k=1,\dots,K$	

For SALBP-1 and SALBP-2 the corresponding worker skill constraints are given in the LSC, HSC and ESC rows. Note, that LSC and HSC constraints can easily be transformed into each other, because they express similar requirements, just focus on two different management problems: a given number of low-skilled workers must be applied, or the available number of high-skilled workers is limited. If both LSC and HSC constraints exist in a problem, then different W_k and S_k must be determined for the LSC and for the HSC formulations. Applying the different W_k values and S_k sets, the indicated constraints can be simultaneously used.

The application of skill constraints increases the number of binary variables, which increases computation time. The total number of skill variables in practice, however, is not very high, compared to the total number of variables of the problem. Nevertheless, applying conditions similar to (4.6), the number of skill constraints can be reduced.

The decrease of the number of variables in the basic SALBP-1 and SALBP-2 is based on the calculation of the lower bound and the upper bound of the workstation index of each task. If the lowest feasible workstation index (LT_i) of each task is known, then the lowest feasible workstation index of a skill variable (LS_k) is the minimum of the lowest feasible workstation indexes of those tasks which belong to skill level k, that is,

$$LS_k = \underset{i \in S_k}{Min}\left(LT_i\right) \tag{4.23}$$

Furthermore, if the highest feasible workstation index (UT_i) of each task is known, then the highest feasible workstation index of a skill variable (US_k) is the maximum of the feasible highest workstation indexes of those tasks which belong to skill level k, that is,

$$US_k = \underset{i \in S_k}{Max}\left(UT_i\right) \tag{4.24}$$

Those skill variables, which are definitely equal to 0 in any feasible solution, can be excluded from the calculations with the following constraints,

$$l_{jk} = 0 \qquad j < LS_k \text{ and } j > US_k \quad k = 1,\dots,K \tag{4.25}$$

$$h_{jk} = 0 \qquad j < LS_k \text{ and } j > US_k \quad k = 1,\dots,K \tag{4.26}$$

$$e_{jk} = 0 \qquad j < LS_k \text{ and } j > US_k \quad k = 1,\dots,K \tag{4.27}$$

Finally, in Table 4.2 skill constraints are added to SALBP-1 and to SALBP-2, that is, the number of workstation (line utilization) or the cycle time is minimized. The proposed models, however, can easily incorporate other objective functions which express the different labor cost of differently skilled workers.

The performance of the suggested skill constraints was tested is several examples. An illustration of two level (K=2) skill constraints based on a slightly modified example of Bowman (1960) can be found in Koltai and Tatay (2011) and a multi-level example is given in Koltai and Tatay (2013). The next chapter shows, how skill constraints are applied in case of the assembly process of a bicycle manufacturer.

4.4 Application of simple ALB models with skill constraints at a bicycle manufacturer

Olympia Bicycle Ltd. is a bicycle manufacturer company producing bicycles in small and medium lots. Lot sizes range between 200-1000 units. All parts necessary for assembly are provided by suppliers. The only non-assembly operation is painting. The frames provided by suppliers are painted in the painting shop. Assembly is performed in three stages: there are two preassembly lines and one final assembly line.

The first preassembly line prepares the wheels; the second preassembly line assembles the frame and the handle bar. The preassembly lines are short and simple. The assignment of tasks to workstations does not require any sophisticated quantitative tool. Therefore, this analysis deals only with the final assembly.

Final assembly is made along a U shaped line. U shape, however, only describes the geometrical form of the line. Each workstation is attended by a single worker, and each worker is assigned to a single workstation. Depending on the bicycle model, about 30-80 tasks are performed at 10-15 workstations. The line moves with a steady speed set by the operations manager based on the expected cycle time. Some tasks are simple and can be learned by any workers, while some tasks require more expertise and experience. The tasks of a typical product of the company are given in Table 4.3.

The table shows the list of tasks of final assembly, the immediately preceding tasks, and the task times. Based on the information of Table 4.3, the precedence graph of tasks can be easily depicted (see Figure 4.2).

Demand for this particular product is 200 units and 5 hour is assigned to produce this quantity in a given day. Based on these data, the required cycle time is 90 seconds ($5\cdot 60\cdot 60/200$).

Table 4.4 summarizes the optimal solution of each model presented in this chapter. Boldface numbers in the T_c and N columns indicate the optimal solutions, while regular face numbers are parameters of the corresponding model. The optimal assignment/station time columns show which tasks are assigned to the specific workstations, and how much time is needed to complete these tasks at the given station.

The row of model 1 in Table 4.4 shows the optimal solution of SALBP-1. According to the optimal solution at least 10 workstations are necessary to produce the required number of bicycles. The cycle time belonging to this optimal assignment is 90 seconds. The highest station time is at workstation 1 (90 second) and the line is very unbalanced. The difference between the smallest and the largest station time is 40 second.

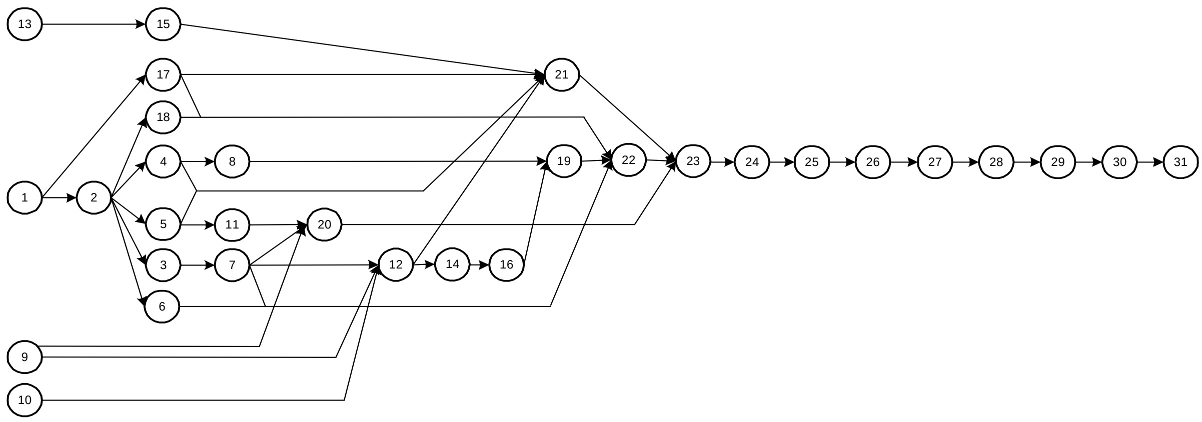

Figure 4.2 Precedence graph of a sample bicycle

The row of model 2 in Table 4.4 shows the optimal solution of SALBP-2. It is assumed that 10 workstations are used. An upper bound on the optimal value of the cycle time is the cycle time of the optimal solution of SALBP-1 (90 sec). It can be seen that the optimal solution is 80 seconds. This line configuration is much more balanced. The difference between the highest and smallest station time is reduced and the smallest station time belongs only to one workstation.

The calculation of the optimal solution of the presented models takes only a few seconds on a common computer using Excel as the input and output interface of Lingo mathematical programming software.

The assignment of tasks to workstations is frequently influenced by workforce skill conditions. In the following, two different workforce skill conditions are illustrated with the help of the bicycle production process.

Table 4.3 Tasks and precedence relations of the sample bicycle model

i	Tasks	Time of task *i* (sec)	Immediate precedent of task *i*	LJ_i	UJ_i
1	Connecting the front break with the Bowden cable housing	21	-	1	25
2	Linking the front part of the rear break with the Bowden cable housing	23	1	1	25
3	Connecting the first part of the front derailleur with the Bowden cable housing	10	2	1	26
4	Linking the rear part of the derailleur with the Bowden cable housing	10	2	1	27
5	Positioning the plastic holder of the Bowden cable housing	10	2	1	27
6	Connecting the rear part of the rear break with the Bowden cable housing	10	2	1	28
7	Linking the first part of the derailleur with the Bowden cable housing	10	3	1	26
8	Linking the middle part of the derailleur with the Bowden cable housing	10	4	1	28
9	Positioning and securing the derailleur	30	-	1	26
10	Supplying the rear derailleur	16	-	1	27
11	Supplying the front derailleur	14	5	1	28
12	Positioning the chain	50	9,10,11	2	27
13	Positioning the front wheel	10	-	1	28
14	Positioning the rear wheel	10	12	2	28
15	Fastening the front wheel	20	13	1	28
16	Fastening the rear wheel	20	14	3	28
17	Installing the front break	24	1	1	27
18	Installing the rear break	24	2	1	28
19	Linking the rear part of the rear derailleur with the Bowden cable housing	10	8,16	3	28
20	Installing the front derailleur	35	7,9,11	2	28
21	Installing the rear derailleur	25	4,5,12,15,17	3	28
22	Cutting the Bowden cables (to right length)	10	6,7,17,18,19	4	28
23	Positioning the ends of the Bowden cable	15	20,21,22	5	28
24	Setting the derailleur	50	23	6	28
25	Setting the breaks	70	24	6	29
26	Positioning the cardboard on the frame	10	25	7	30
27	Positioning quick-release on frame	10	26	7	30
28	Removing the first wheel and secure it to the frame	35	27	7	30
29	Positioning the brakes	15	28	7	30
30	Packing 1	50	29	8	30
31	Packing 2	50	30	8	31

Table 4.4 Optimal solutions of the SALB models

Models	T_c	*N*	*WH/ WS*	*H/S*	Optimal assignment/*station time* (sec)										
					j=1	*j*=2	*j*=3	*j*=4	*j*=5	*j*=6	*j*=7	*j*=8	*j*=9	*j*=10	*j*=11
1. SALBP-1	90	**10**	-	-	1,2,3,4, 5,10	6,7,9, 11,17	13,15,20	12,14,18	8,16, 19,21	22,23, 24	25	26,27, 28	29,30	31	
					90	*88*	*65*	*84*	*65*	*75*	*70*	*55*	*65*	*50*	
2. SALBP-2	**80**	10	-	-	9,10, 13,15	1,2,5, 6,11	3,4,7 12	8,14 18,20	16,17 19,21	22,23, 24	25	26,27 28	29,30	31	
					76	*78*	*80*	*79*	*79*	*75*	*70*	*55*	*65*	*50*	
3. SALBP-1+HSC	90	**Inf.**	1	20,24											
4. SALBP-2+HSC	**100**	10	1	20,24	1,2,4,8, 13,17	3,9,18	5,6,7, 11,15	10,12, 14,16	19,21, 22	**20**,23, **24**	25	26,27	28,29, 30	31	
					98	*64*	*64*	*96*	*45*	*100*	*70*	*20*	*100*	*50*	
5. SALBP-1+HSC	90	**10**	2	20,24	1,2,3,4 7,13	5,9,11, 15	10,12,18	17,**20**,21	6,8,14, 16,19,22	23,**24**	25	26,27 28	29,30	31	
					84	*74*	*90*	*84*	*70*	*65*	*70*	*55*	*65*	*50*	
6. SALBP-2+HSC	**80**	10	2	20,24	1,2,4, 10,13	3,7,8, 17,18	5,9,11, 15	12,14,16	6,19,**20**, 21	22,23, **24**	25	26,27 28	29,30	31	
					80	*78*	*74*	*80*	*80*	*75*	*70*	*55*	*65*	*50*	
7. SALBP-1+LSC	90	**11**	2	3,4,5,6,7, 8,26,27	1,2,**5**, 10,11	9,12, 13	**4**,**6**,**8**	14,15, 16,18,19	**3**,**7**,17, 20,22	21,23, 24	25	**26,27**	28,29	30	31
					84	*90*	*30*	*84*	*89*	*90*	*70*	*20*	*50*	*50*	*50*
8. SALBP-2+LSC	**84**	11	2	3,4,5,6,7, 8,26,27	1,2,9, 13	**4**,**8**	**3**,5,**6**,**7**	10,11,12	14,15,16, 17,19	18,20, 21	22,23, 24	25	**26**,**27**,28, 29	30	31
					84	*20*	*40*	*80*	*84*	*84*	*75*	***70***	*70*	*50*	*50*
9. SALBP-2+LSC	**100**	10	2	3,4,5,6,7, 8,26,27	1,2,9, 13	**3,4,5, 6,7,8**	10,11,15, 17,18	12,14, 16,19	20,21, 22,23	24	25	**26,27**	28,29,30	31	
					84	*60*	*98*	*90*	*85*	*50*	*70*	*20*	*100*	*50*	

4.4.1 Application of high-skill constraints (HSC)

Generally, workers of the bicycle plant are able to perform all the required tasks. The line manager, however, thinks that some complicated tasks must be assigned to the best workers.

In this case, it is implicitly assumed that there are complicated tasks which require special skills and can be performed by special, qualified workers. The tasks which require special skills belong to set S_1. The rest of the tasks do not require special skill and/or special qualification of the workers, consequently two skill levels (K=2) are defined. The regular tasks belong to set $\bar{S}_1$ which is in this case is S_2.

In our sample assembly process, one of the workers at the line is considered the most able and the most complicated tasks are generally assigned to the workstation of this worker. This implicit requirement of the line manager can be formulated explicitly as high-skill constraint. It is assumed that only this high-skilled worker (W=1) is able to perform that subset of the tasks (S_1) which are considered complicated by the line manager. At the product of the sample problem, tasks with indices 20 and 24 are considered complicated (S_1={20, 24}).

Adding constraints (4.17), (4.18), (4.19) and (4.20) to the SALBP-1, the minimum number of workstations considering high-skill constraints can be obtained. The results in the row of model 3 in Table 4.4 indicate that the model has infeasible solution. This can be easily explained by looking at Figure 4.2. Task 20 immediately precedes task 23 and task 23 immediately precedes task 24. Since tasks 20 and 24 have to be assigned to the only high-skilled worker, all these tasks (20, 23, 24) must be performed by the one available high-skilled worker. This would result in a station time equal to 100 seconds (35+15+50), which is infeasible according to the cycle time constraints (100>T_c=90).

Solving the SALBP-2 with 10 workstations (with the optimal solution of the SALBP-1 without HSC) and completed with constraints (4.17), (4.18), (4.19) and (4.20), we obtain 100 seconds for the minimal cycle time (see the row of model 4 in Table 4.4), and the high-skilled worker works at workstation 6. This result also shows that the original cycle time (90 seconds) cannot be met with a single high-skilled worker.

According to the row of model 5 in Table 4.4, the optimal solution of SALBP-1 with 2 high-skilled workers (W=2) is 10 workstations. The two high-skilled workers work at workstations 4 and 6. Worker skill constraints in the analyzed case will not lengthen the assembly line if two high-skilled workers are available; however, the operation costs could be higher because of the application of two high-skilled workers.

Finally, solving the SALBP-2 with 2 high-skilled workers, the minimum of the cycle time is 80 minutes, and high-skilled workers work at workstations 5 and 6 (see the row of model 6 in Table 4.4). Consequently, HSC will not deteriorate cycle time if two high-skilled workers are available.

Based on these results, the management may consider providing special training to some workers, because with only one high-skilled worker the capacity of the line is insufficient.

4.4.2. Application of low-skill constraints (LSC)

Generally workers of our sample bicycle plant are able to perform all the required tasks. Sometimes (e.g. in holiday seasons), however, because of labor shortage, temporary workers are applied. The line manager knows that these workers are not skilled properly and only the simplest tasks can be assigned to them. In this case, it is assumed that only a subset of tasks can be assigned to a limited number of workers.

These tasks are called simple tasks and they belong to set S_1. The rest of the tasks are regular tasks and belong to set $\bar{S}_1$, which is equivalent to S_2. We again face a two-skill level case (K=2). It is assumed that a limited number of low-skilled workers are already employed; therefore, workstations for them must be organized.

Let us assume in the sample assembly process that two temporary workers are applied (W=2) and only eight tasks (listed in Table 4.4 in column S_1 of the last three models) can be assigned to these workers. The solution of model 7 in Table 4.4 shows that the minimum number of workstations necessary in this case is 11. Consequently, the application of low-skilled workers increases the length of the line by one workstation compared to the original case (see the results of model 1).

The minimal cycle time for 11 workstations with low-skill constraints is 84 seconds. This is also higher than the minimal cycle time obtained for the original problem (see the results of model 2). Consequently, the application of temporary workers increases line length and deteriorates cycle time as well.

The deterioration of cycle time is even more apparent if the SALBP-2 is solved for 10 workstations and with low-skill constraints (see the row of model 9 in Table 4.4). In this case, cycle time is 20 percent higher than in the original case (100 seconds).

Based on these results the management may consider, for example, a special training for temporary workers to eliminate the unfavorable effect of low-skilled workers on line length and on cycle time.

4.5 Sensitivity analysis of line efficiency with respect to production quantity

An important problem of assembly line operation is the proper reaction to production requirement changes. If production requirement increases, line capacity generally must be increased. If production requirement decreases utilization of the line decreases as well, and, consequently, the decrease of line capacity is required. These types of problems can be analyzed with the help of the Efficiency-quantity ($E(Q,N)$) chart (Koltai and Tatay, 2010; Koltai, Tatay and Kalló, 2014).

The calculation of line efficiency for a specific line configuration with N workstations is as follows,

$$E(Q,N) = \frac{\sum_{i=1}^{I} t_i}{N \cdot T_c} = \frac{\sum_{i=1}^{I} t_i}{N \cdot \frac{T}{Q}} = Q \cdot \frac{\sum_{i=1}^{I} t_i}{N \cdot T} \tag{4.28}$$

Expression (4.28) shows that line efficiency is determined by the number of workstations (N). At the ideal task assignment, station time of each workstation is equal to the cycle time; that is, s_j=T_c. In this case, the line is perfectly balanced, there is no idle time at the workstations and line efficiency is equal to 1. If the line cannot be perfectly balanced, idle time exists and line efficiency decreases. If, for some reasons, more workstations are applied than necessary, line efficiency also decreases.

Expression (4.28) shows that line efficiency ($E(Q,N)$) is a linear function of production quantity (Q). If production quantity *increases*, then cycle time decreases and, consequently, line efficiency increases. There is, however, a maximum quantity which can be produced during the available total time (T). This quantity is determined by the maximal station time of the line ($\mathrm{Max}\{s_j\}$), that is,

$$Q_{\mathrm{Max}}(N) = \frac{T}{\underset{j=1,\dots,J}{\mathrm{Max}}\{s_j\}} \tag{4.29}$$

If target production quantity is above the maximum production quantity, either the increase of total time required or a new line configuration must be determined. Therefore,

$Q_{\mathrm{Max}}(N)$ is the upper production limit of the current task assignment.

If production quantity *decreases*, line efficiency decreases as well. After a certain amount of production quantity decrease, the line configuration is not optimal anymore; the required quantity can be produced with fewer workstations. The lowest production quantity at which the line with N workstations is still optimal is determined by the maximal production quantity of the optimal line with N–1 workstations. This quantity is denoted by $Q_{\mathrm{Max}}(N-1)$.

If the SALBP-1 is solved, a solution which provides a line configuration with the minimum number of workstations (N) is obtained. As long as production quantity is within the validity range defined by the minimal and maximal production quantity, the line configuration is optimal with respect to the number of workstations and task assignment. If production increases, N workstation is not enough and more workstations are required. If production decreases, less workstation should be applied; therefore, the line with N workstations is not optimal any more. Consequently, the line with N workstation is optimal only if production quantity is within the following range,

$$Q_{\mathrm{Max}}\left(N-1\right) < Q \leq Q_{\mathrm{Max}}\left(N\right) \tag{4.30}$$

Note that expressions (28) and (29) can be calculated for any line configuration with N workstations. Several task assignments may belong to a line with N workstations and different cycle time and maximum production quantity may belong to each assignment. The assignment with the smallest cycle time can be obtained by solving the SALBP-2 for N workstations. In this case, the highest (optimal) maximal production quantity belonging to N can be obtained. The optimal maximum production quantity is denoted by $Q_{\mathrm{Max}}^{\mathrm{OPT}}\left(N\right)$. Using this value, the validity range for the optimal cycle time line configuration, that is, the *maximum validity range* is the following,

$$Q_{\mathrm{Max}}^{\mathrm{OPT}}\left(N-1\right) < Q \leq Q_{\mathrm{Max}}^{\mathrm{OPT}}\left(N\right) \tag{4.31}$$

The validity range of optimality is closely related to line efficiency. If production is within the validity range, line efficiency is the highest possible. If production is outside of this range, a new line must be formed.

Figure 4.3 shows the change of line efficiency with respect to production quantity for each possible line configuration of the bicycle assembly line.

The figure shows the $E(Q,N)$ function for several optimal workstation configurations (N=1,…,14). Each line in the figure is obtained by solving a SALBP-2 with the corresponding N. In the case of N=1, each task is performed at a single workstation. Cycle time is equal to the sum of task times (707 seconds) and the maximum production quantity is equal to 25,5 according to (4.29). At the maximum production quantity line efficiency is 1, because there is no idle time at the workstation. The rest of the functions are obtained by solving the SALBP-2 for N=2,…,14, and the highest value of the independent variable Q is $Q_{\mathrm{Max}}^{\mathrm{OPT}}\left(N\right)$ at each line.

In Chapter 4.4, the SALBP-1 was solved for Q=200 units. According to the results the optimal number of workstations is equal to 10. Cycle time is 90 seconds which is higher than the optimal cycle time obtained by the SALBP-2; therefore, $Q_{\mathrm{Max}}(10)$=200. The optimal cycle time in the case of N=10 is 80 seconds, and $Q_{\mathrm{Max}}^{\mathrm{OPT}}\left(10\right)$=225 units. Solving the SALBP-2 for N=9 we get $Q_{\mathrm{Max}}^{\mathrm{OPT}}\left(9\right)$=189.5. According to (30), the *validity range* at the task assignment obtained by the SALBP-1 in Chapter 4.3 is as follows,

$$189.5 < Q \leq 200 \tag{4.32}$$

According to (31) the *maximum validity range* of the optimal cycle time line configuration with 10 workstations is as follows,

$$189.5 < Q \leq 225 \tag{4.33}$$

Assume now that production decreases to 180 units. Figure 4.3 shows that 180 units can be produced with 9 workstations. The operation manager must decide which line configuration to use. Continue producing with 10 workstations at 0.707 line efficiency or reorganize the line and produce with 9 workstations at 0.78 line efficiency. The answer is partly influenced by expected further changes. If production further decreases, it is probably better to apply 9 workstations. If the production decrease is temporary and production quantity is expected to return to the previous level (200), then it is not worth to reorganize the line.

Assume now that production increases to 210 units. Currently, the solution of the SALBP-1 is used; therefore, 210 is outside of the validity range. This value is, however, within the maximal validity range, therefore, 10 workstations with a task assignment provided by the solution of the SALBP-2 is feasible. Looking at Figure 4.3 managers can see the possibilities of the current line configuration and can make a proper decision when production requirement changes.

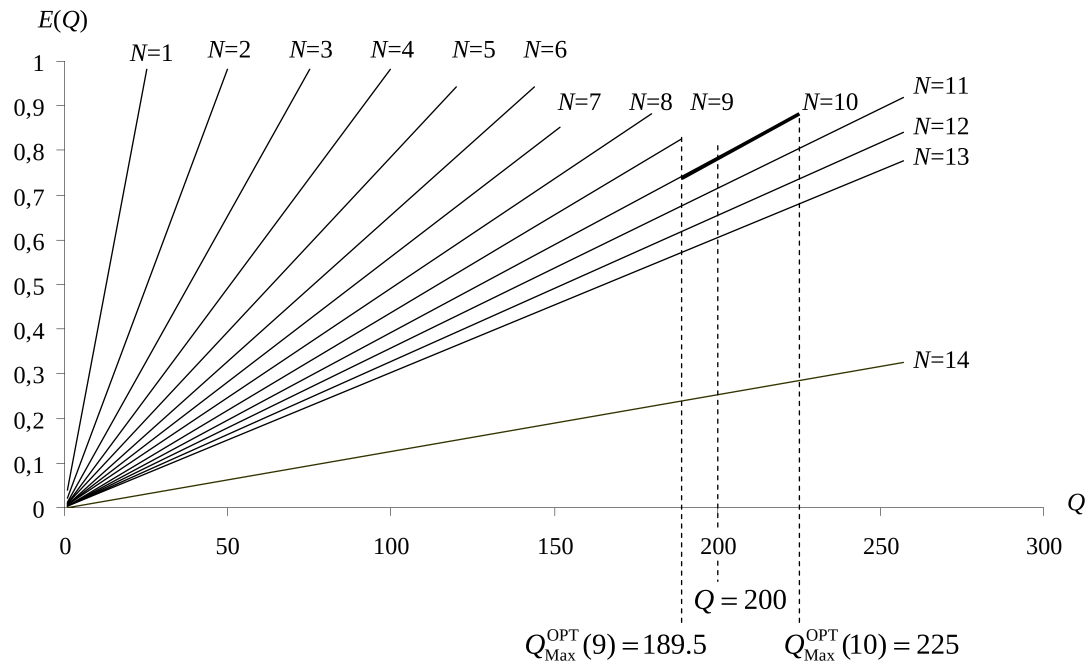

Figure 4.3 Effect of production quantity on line efficiency

4.6 The effect of the change of task times

The task times used in the models presented in this chapter may change for several reasons during production. Two of these reasons are investigated next. First, since tasks are performed by workers, the task times are random variables. Second, as a consequence of the learning process, task times may decrease as production proceeds.

4.6.1 The effect of the variation of task times

The presented simple ALB models cannot take into considerations the stochastic characteristics of task times. Therefore, a simulation model was prepared to analyze the effect of the variability of task times. The objective of simulation is to examine the robustness of the optimal solutions of the mathematical programming models presented in Chapter 4.4 with respect to the stochastic change of task times.

Different distributions (deterministic, normal, and uniform) and different relative standard deviations (5, 10, 15, 20%) were applied to model the variations of task times. The length of the replications was chosen according to the time requirement of the minimal lot size ($200 \cdot 90 = 18000$ sec). Some results based on 500 replications are summarized in Table 4.5.

Table 4.5 shows the statistical characteristics (mean and half width) of production quantities in the given 5 hour production period as a function of the distribution function and relative standard deviation using the optimal solution of the SALBP-1 as an input. The data clearly show that only large standard deviations have considerable effect on the target production quantity (200 units).

We have also examined the effect of high-skilled and temporary workers. We assumed the same mean task times, but different relative standard deviations for the same task if it is performed by differently skilled workers. Lower variance is assumed for high-skilled workers, and higher variance is assumed for low-skilled workers. Simulating the operation of the assembly line using the optimal solutions of each model of Table 4.4 as an input, similar results (with small differences in accordance to the input variations) were obtained to the data shown in Table 4.5.

Table 4.5 Summary of simulation results

Distribution	Relative standard deviation								
	0%	5%		10%		15%		20%	
	Mean	Mean	Half width	Mean	Half width	Mean	Half width	Mean	Half width
Deterministic	200	---	---	---	---	---	---	---	---
Normal	200	199.6	0.04	199.3	0.05	199.0	0.08	198.8	0.09
Uniform	200	199.2	0.01	198.6	0.1	197.9	0.13	196.8	0.16

Based on the simulation results, we can conclude that if efforts are made by the management to reduce the variance of task times by proper organization of the line and by training the workers, the output variance is relatively small. In this case, the effect of variance of task times on the cycle time and, consequently, on the output quantity is relatively small. Therefore, the optimal solutions of SALBM-1 and SALBM-2 can be accepted as valuable information for line configuration decisions.

4.6.1 The influence of the learning effect

In case of the presence of learning effect, it is presumed, that station time decreases as the number of the performance of the tasks at the station increases. In this case s_j is the station time only at the first performance of the operation at station j. An $s_j(Q)$ function describes the station time as a function of the number of processed parts. Applying the classical exponential learning function (Yelle, 1979) the value of $s_j(Q)$ is the following,

$$s_j(Q) = s_j Q^{b_j}, \tag{4.34}$$

where $b_j<0$ determines the decrease of station time at station j in case of learning effect.

In this case, any calculation, which is based on the assumption of a constant cycle time, must be revised, since *cycle time constantly changes* for two main reasons:

1) If station j is the bottleneck of an assembly line, then, as a consequence of the learning effect, cycle time decreases exponentially according to the $s_j(Q)$ function.

2) In case of the presence of learning effect, bottleneck may shift from one station to another at certain point of time.

The cycle time change is continuous (exponentially decreasing) in the case of point 1). In the case of point 2, however, the change of the cycle time is not continuous as it is illustrated in Figure 4.4.

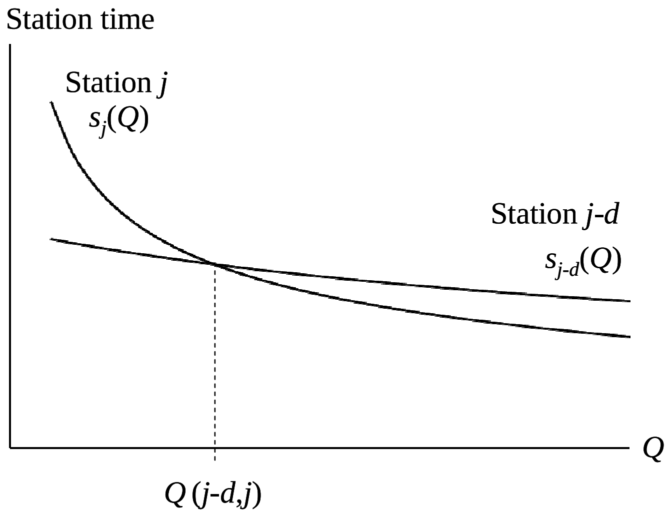

Figure 4.4 Illustration of bottleneck shift

Figure 4.4 shows the station time functions of two different stations (station $j-d$ and j) as a function of production quantity. The difference between the two station indexes is expressed with d. To depict the station time functions of stations $j-d$ and j in the same diagram, a common independent variable must be selected since each station processes a different part at the same time. For practical reasons, we select the production quantity of the latest workstation (station j in this case) as independent variable. Consequently, $s_{j-d}(Q)$ denotes the station time of station $j-d$ as a function of the part manufactured at station j.

According to Figure 4.4, at first, station j is in the bottleneck, since $s_j(Q)>s_{j\text{-}d}(Q)$, and cycle time exponentially decreases. Station time of stations $j-d$ and j are equal at quantity $Q(j\text{-}d,j)$. After this quantity, station $j-d$ is in the bottleneck of the line because $s_j(Q)<s_{j\text{-}d}(Q)$. $Q(j-d,j)$ indicates bottleneck shift. At this quantity, station j leaves the bottleneck and station $j-d$ enters the bottleneck. In case of an assembly line with several workstations, the output rate of the line is determined by the envelopment curve of all the station time functions of the line.

Assembly line balancing in case of learning effect is very complicated, since a constantly changing cycle time cannot be minimized (Cohen, Vitner and, 2006). However, the results of assembly line balancing based on constant cycle time can be acceptable as a good approximation of optimal line configuration under the following conditions:

– Station time doesn’t decreases infinitely in practice. Even in case of learning effect, a constant station time is assumed after an initial production period (warm-up period) of the line. If this constant station time is used, the line configuration is optimal for the operation after the warm-up period.

– Bottleneck change occurs generally at small quantities (small $Q(j\text{-}d,j)$ values), therefore, its effect on production quantity and on throughput time is significant only in case of small production batches.

Consequently, if steady state station times can be estimated, and production batches are relatively large, the results of assembly line balancing models presented in the previous chapters can provide acceptable information for line configuration decisions.

A detailed examination of the effect of learning on the bottleneck shifts and on the throughput time can be found in Koltai and Györkös (2012) and in Koltai, Györkös and Kalló (2014). The change of the efficiency function as a consequence of learning is discussed in Koltai and Györkös (2013).

4.7 Conclusions of Chapter 4

This chapter showed how basic assembly line balancing models can be completed with workforce skill constraints. First, the two basic models, that is, the workstation minimization model and the cycle time minimization model are presented. Next, in order to generalize workforce skill constraints the basic cases are classified into three categories. *Low-skill constraints* are applied when the focus of operations manager is to provide work for low-skilled workers. *High-skill constraints* are applied when the responsibility of operations managers is the performance of complicated tasks with limited number of high-skilled workers. In both cases the consideration of several skill levels makes the constraints generally applicable in practical context. Finally, *exclusive-skill constraints* are applied for tasks requiring specialists. Mathematical formulation of the three skill constraint types is presented and the constraints are integrated into the basic simple assembly line balancing models.

The effect of skill constraints on the optimal solution of ALB models is analyzed in a bicycle assembly process. The optimal solution of the models helps the operations manager to make decisions when frequent reconfiguration of the line is required as a consequence of frequent production quantity changes under changing workforce skill conditions.

Implicit management considerations about workforce skill are translated into explicit mathematical constraints. The solution of the ALB problems completed with HSC and LSC may help to evaluate the effect of the availability of different workers on line length and on line capacity.

Based on the optimal solutions of several SALB-2 problems a graphical tool is developed to analyze the effect of production quantity changes on efficiency.

The presented ALB models are deterministic; therefore, they do not take into consideration the variance of task times. However, simulation analysis of the operation of the assembly line based on the optimal solution of ALB models was performed. The results of simulation revealed that the optimal solution of ALB models provides robust information related to production quantity. Consequently, the solution of the presented simple ALB models can provide relevant information for the production manager of the bicycle assembly plant.

The mathematical programming models in the case of the bicycle manufacturer require less than 1200 binary variables even in the most complicated case. The application of skill constraints with two skill levels requires about 24 new skill variables (2 skill level*12 workstations=24 skill variables). Run time is strongly influenced by the structure of the precedence graph. For any problem at this company, however, the optimal solution can be obtained in less than 1 minute with Lingo software on an average laptop computer. These data show that there are no computational constraints when these models are applied in practice.

As a summary, based on Chapter 4 the following scientific results can be formulated:

Result 3/1

I have defined the following three types of multi-level workforce skill constraints in case of simple assembly line balancing problems:

– *Low-skill-constraint* (LSC): *A given number of workers belonging to skill level k must be applied* at the assembly line. These workers cannot perform tasks belonging to skill levels higher than *k*. They can only work at stations which have tasks with skill level equal to or lower than *k*, and the number of such stations is constrained from *below*.

– *High-skill-constraint* (HSC): *Only a limited number of workers are able to perform the most complicated tasks.* These workers cannot perform tasks belonging to skill levels higher than *k*. They can only work at stations which have tasks with skill level equal to or lower than *k*, and the number of such stations is constrained from *above*.

– *Exclusive-skill-constraint* (ESC): *Some tasks can be performed only by special workers*. In this case workers have different skills/specializations, and a worker specialized in one type of skills is not able to perform tasks requiring other type of skills. Consequently, tasks belonging to different skill types cannot be mixed at a station.

The formulation of the three different constraint types is summarized in Table 4.2.

Result 3/2

The efficiency of an optimal simple assembly line configuration is a linear function of production quantity. For several production quantities, the same task assignment is optimal, and these production quantities determine the sensitivity range of the optimal task assignment. I showed that a task assignment of a line with N workstations is optimal if production quantity is within the range specified by constraint (4.31). Within this range, line efficiency linearly changes with respect to production quantity, but all the resulting efficiencies are optimal.

The definition and mathematical description of the different workforce skill constraints are presented in Koltai and Tatay (2011, 2013), Tatay and Koltai (2011) and Koltai (2013). The practical application of the presented approach, and the interpretation of the results in a practical case are presented in Koltai and Tatay (2010), Tatay and Koltai (2010), Koltai, Tatay and Kalló (2011, 2014), Koltai (2012) and Koltai and Györkös (2012). The effect of learning on the operation of simple assembly lines and the analysis of the efficiency function in case of the presence of learning effect can be found in Koltai and Györkös (2013), Koltai, Györkös and Kalló (2014), and Györkös, Koltai and Kalló (2014).

5 APPLICATION OF PERTURBATION ANALYSIS FOR SENSITIVITY ANALYSIS OF A PRODUCTION SCHEDULE

Production planning generally determines the production quantities of parts/products in a given production period, and the amount of resources assigned to perform production tasks. Next, a detailed schedule of the performance of the production tasks must be determined. Production scheduling is generally presented in form of a Gantt chart for every machine participating in production. The Gantt chart shows the starting and completion time of the operations on the machines. A production schedule, in most cases, is very sensitive to several stochastic events. Operation times may change, machines may break down and operators may not be available. An important question is, how the occurrence of unexpected events influences the production schedule. This chapter shows how the sensitivity of a production schedule can be analyzed with perturbation analysis (PA). The example of a continuous steal casting process shows, how the change of production schedules can be analyzed with PA, and how requirement for the modification of the schedule can be forecasted. The results of this chapter are based on the papers of Koltai (1992), Koltai, Larraneta and Onieva (1993), Koltai, Larraneta and Onieva (1994), Koltai et al., 1994 and Koltai and Lozano (1998).

5.1 Introduction

A challenging possibility for the examination of discrete event dynamic systems (DEDS) is the application of perturbation analysis (PA) which can provide gradient information from a single simulation experiment (see for example Ho, Euler and Chen, 1983 and Ho, 1987). The idea of PA is to perform a simulation experiment, and via an algorithm an estimate can be derived about the gradient of a performance measure of the system with respect to one of its parameters (Ho, 1983). This gradient information can be used for iterative improvement of system performance (Ho et al., 1984; Rubinstein, 1986).

Various intriguing problems have been solved since the first publication of the method. Propagation rules for infinitesimal and finite perturbations (Ho, Cao and Cassandras, 1983), examination of multi-class networks (Cao, 1988), various suggestions for avoiding or at least smoothing the effect of discontinuities are extending the application area of PA (Ho and Li, 1988). Researchers of this field, however, have mostly concentrated on generating and/or propagating perturbations, but have avoided the examination of the validity range within which the gradient information is valid. The infinitesimal approach deals with this problem by simply saying that the size of the perturbation is small enough not to hurt the deterministic similarity. The finite approach calculates accurately the effect of finite changes of a parameter with higher order propagation rules, but it also fails to provide information about the validity range (Ho, Cao and Cassandras, 1983). The effect of a specific perturbation is calculated correctly but if the perturbation changes the calculation has to be performed again.

In the proceeding part of this chapter I shortly explain the concept of PA, and show how the validity range of deterministic similarity can be derived if the event sequence table is generated by simulation. A small example illustrates the implementation of the calculation in a discrete event simulation environment. The application of PA is also presented for the calculation of the gradient of the throughput with respect to a routing parameter in flexible manufacturing systems. Finally, the application of PA and validity range calculation is illustrated with a practical example. In this example validity range calculation is performed in case of an automated steel manufacturing process, where the sensitivity of a production schedule must be examined, but the event sequence table is provided by a scheduler, and not by discrete event simulation.

5.2 Basic concepts of perturbation analysis (PA)

PA was developed for the gradient estimation of performance measures with respect to certain system control variables in DEDS, when the performance measure is obtained by discrete event simulation. The basic idea is that a sample path of the simulation contains information about certain system characteristics. Therefore, it is not necessary to rerun the simulation when the performance measure sensitivity is estimated (Ho and Cassandras, 1983). To facilitate further discussion we introduce here some basic concepts of PA based on the work of Ho and Cao (1991).

The change of a system control variable is called *perturbation*. The original sample path is called *nominal path* and the one belonging to the perturbed control variable is called the *perturbed path*. A sample path represents a specific structure of events that occur at the resource elements of the system. Three of these events can be specified:

a.) *operation* (OP) when a resource performs operation on an entity.
b.) *no-input* (NI) when a resource is idle because there is no entity it could work on.
c.) *full-output* (FO) when a resource is blocked, because the entity it was working on, cannot leave the resource.

The *event sequence table* contains the order of these events at the resources. The event sequence table changes if any of its events disappears or a new one appears as a consequence of any change of the system control variables. *Deterministic similarity* means that the event sequence tables of the nominal and perturbed sample path are equal.

The appearance of perturbation is called *perturbation generation*. When a perturbation appears at a specific entity, it may spread through the system, changing the beginning and ending operation time of other entities. This is called *perturbation propagation*. The change of the finishing time of the last entity at a sample path is the *sample path gradient*. If deterministic similarity is not hurt when the control variable changes, then the performance measure is a linear function of the control variable. The interval of the perturbation, within which this linear function has the same gradient, is the *validity range of the gradient* of the sample path (Koltai 1992). When many sample paths are generated both the gradient and its validity range can be estimated.

My main concern is the calculation of the gradient of the throughput time with respect to the mean of the operation time of a resource, furthermore, the calculation of the range within which changing this mean operation time, the gradient remains the same.

5.3 Formal treatment of perturbation analysis

Let us consider a queuing network consisting of R_j, j=1,...,M single server resources with finite buffers and FIFO queuing disciplines. The capacity of the queue of R_j is c_j, and includes the entity occupying the resource as well. The number of entities in the queue of R_j is q_j. The E_i, i=1,...,N entities are served at the resources. The mean operation time of R_k is perturbed, and the throughput time is measured when E_u terminates the operation at R_v. Relative to E_i at R_j we define the following functions:

$nr(i,j)$ − the next resource entity i visits after finishing the operation at R_j,
$pr(i,j)$ − the previous resource entity i visited before arriving to R_j,
$ne(i,j)$ − the next entity following the operation of E_i on R_j,
$pe(i,j)$ − the entity processed previous to E_i on R_j,
$a(i,j)$ − the entity which is in process at $R_{nr(i,j)}$ the time E_i finishes its operation at R_j,

Notations used in this chapter are summarized in Table 5.1. Note, that in order to avoid the application of two level indexing, in case of some variables running indices are written in parenthesis.

Our main objective is to determine the expected value of the gradient of a performance measure, based on the gradient of the performance measure obtained from a single sample path, that is,

$$\frac{\partial E[L(\theta_k,\xi)]}{\partial \theta_k} = ? = E\left[\frac{\partial L(\theta_k,\xi)}{\partial \theta_k}\right] \tag{5.1}$$

and its validity,

$$E[LL(\theta_k,\xi)] \le \theta_k \le E[UL(\theta_k,\xi)] \tag{5.2}$$

The conditions of interchangeability of differentiation and expectation are briefly discussed in Chapter 5.5.

To calculate (5.1) and (5.2) the sample path gradient and the sample path validity have to be determined. To facilitate the formal description of the calculations, first the occurrence of NI and FO events are analyzed.

5.3.1 Analysis of the NI and FO events

In Chapter 5.2 I have already introduced the concept of full-output (FO) and no-input (NI) events. In this section the duration of these events will be formally described.

a) The occurrence of NI is illustrated in Figure 5.1. Entity E_i, after leaving resource R_j, finds $R_{nr(i,j)}$ idle, therefore its service can be initiated immediately upon arrival. The waiting time of $R_{nr(i,j)}$ for the arrival of E_i after finishing the operation of $E_{pe(i,nr(i,j))}$, is calculated as follows:

$$ni_{ij} = b_{pe(i,j),2nr(i,j)} - b_{i,2j} \tag{5.3}$$

This time interval is calculated when $q_{nr(i,j)}=0$, therefore, $ni_{ij} \le 0$ in case of the occurrence of NI.

b.) The occurrence of FO is represented in Figure 5.2. E_i, after finishing its operation on R_j, finds the queue of $R_{nr(i,j)}$ full, therefore it stays in R_j, keeping it blocked until $R_{nr(i,j)}$ finishes the operation of $E_{a(i,j)}$. The time, while R_j is blocked, can be calculated as follows:

$$fo_{ij} = b_{a(i,j),2nr(i,j)} - b_{i,2j} \tag{5.4}$$

This time interval is calculated when $q_{nr(i,j)} = c_{nr(i,j)}$, therefore, $fo_{ij} \ge 0$ in case of the occurrence of FO.

c.) If R_i finds neither NI nor FO, there is no idle time before E_i enters $R_{nr(i,j)}$ and after E_i leaves R_j. In these cases $q_{nr(i,j)}$ must be evaluated from the point of view of the possibility of NI or FO. If there is just one entity in the queue of $R_{nr(i,j)}$, there is a high possibility of the occurrence of NI. This situation is called *potential no-input* (PNI) (Ho, Cao and Cassandras 1983) and it is illustrated in Figure 5.3. PNI occurs if at the moment E_i finishes its operation at R_j the value of $q_{nr(i,j)}=1$. The duration of the PNI can be calculated as follows:

$$pni_{ij} = b_{pe(i,nr(i,j)),2nr(i,j)} - b_{i,2j} \tag{5.5}$$

This time interval is calculated when $q_{nr(i,j)}=1$, therefore, $pni_{ij} \ge 0$ in case of the occurrence of PNI.

d.) If there is just one free space in the queue of $R_{nr(i,j)}$ when E_i finishes its operation at R_j, then there is a high possibility of the occurrence of FO. This situation is called *potential full-output* (PFO) (Ho, Cao and Cassandras, 1983) and it is illustrated in Figure 5.4. PFO occurs if at the moment E_i finishes its operation at R_j the value of $q_{nr(i,j)}=c_{nr(i,j)}-1$. The duration of the PFO can be calculated as follows:

$$pfo_{ij} = b_{a(i,j),nr(i,j)} - b_{i,2j} \tag{5.6}$$

This time interval is calculated when $q_{nr(i,j)}=c_{nr(i,j)}-1$, therefore, $pfo_{ij} \le 0$ in case of the occurrence of PFO.

Table 5.1 Summary of notation of Chapter 5

Subscripts:

i	–	index of entities (i=,…,I),
j	–	index of resources (j=,…,J),
k	–	index of the resource at which perturbation is generated,
u	–	index of the last entity,
v	–	index of the last resource.

Parameters:

c_j	–	the capacity of the queue of R_j, including the entity in R_j as well,
t_{ij}	–	operation time of E_i at R_j,
$b_{i,2j-1}$	–	beginning time of the operation of E_i at R_j,
$b_{i,2j}$	–	ending time of the operation of E_i at R_j.

Variables:

$nr(i,j)$	–	the next resource entity i visits after finishing the operation at R_j,
$pr(i,j)$	–	the previous resource entity i visited before arriving to R_j,
$ne(i,j)$	–	the next entity following the operation of E_i on R_j,
$pe(i,j)$	–	the entity processed previous to E_i on R_j,
$a(i,j)$	–	the entity which is in process at $R_{nr(i,j)}$ the time E_i finishes its operation on R_j,
θ_k	–	mean operation time of R_k,
ξ	–	random variable, representing a particular realization of all the random variables in the system,
$L(\theta_k,\xi)$	–	performance measure of the system belonging to a sample path,
$q(j)$	–	number of entities occupying the queue of R_j,
ni_{ij}	–	duration of no-input caused by E_i when finishing operation at R_j,
fo_{ij}	–	duration of full-output experienced by E_i when finishing operation at R_j,
pni_{ij}	–	duration of potential no-input caused by E_i when finishing operation at R_j,
pfo_{ij}	–	duration of potential full-output experienced by E_i when finishing operation at R_j,
ot_{ij}	–	duration of overtake belonging to E_i on R_j,
$\hat{t}_{ik}$	–	random variable of the operation time of E_i at R_k,
δ_{ij}	–	perturbation of the finishing time of the operation of E_i at R_j,
$LL(\theta_k,\xi)$	–	lower limit of the change of θ_k at which deterministic similarity still holds,
$UL(\theta_k,\xi)$	–	upper limit of the change of θ_k at which deterministic similarity still holds,
$D_{ij}^{(\mathrm{NI})}$	–	coefficient of accumulated perturbations in case of no-input caused by E_i when finishing operation at R_j,
$D_{ij}^{(\mathrm{FO})}$	–	coefficient of accumulated perturbations in case of full-output experienced by E_i when finishing operation at R_j,
$D_{ij}^{(\mathrm{PNI})}$	–	coefficient of accumulated perturbations in case of potential no-input caused by E_i when finishing operation at R_j,
$D_{ij}^{(\mathrm{PFO})}$	–	coefficient of accumulated perturbations in case of potential full-output experienced by E_i when finishing operation at R_j,
$D_{ij}^{(\mathrm{OT})}$	–	coefficient of accumulated perturbations in case of overtake belonging to E_i on R_j,
$w_{ij}(\xi)$	–	waiting time of E_i before entering R_j in a simulation experiment,
$LL_{ij}^{(k)}$	–	lower limit of the change of the operation time of E_i on R_j in the kth iteration step,
$UL_{ij}^{(k)}$	–	upper limit of the change of the operation time of E_i on R_j in the kth iteration step,
TP_0	–	system throughput in the FMS example,
θ_{ts}	–	ratio of part type t visiting route s in the FMS example.

Set:

$s\{i,j\}$	–	index set of those E_m on R_k whose generated perturbation participate in the perturbation of E_i on R_j.

Other notation:

E_i	–	identification of entity i,
R_j	–	identification of resource j.

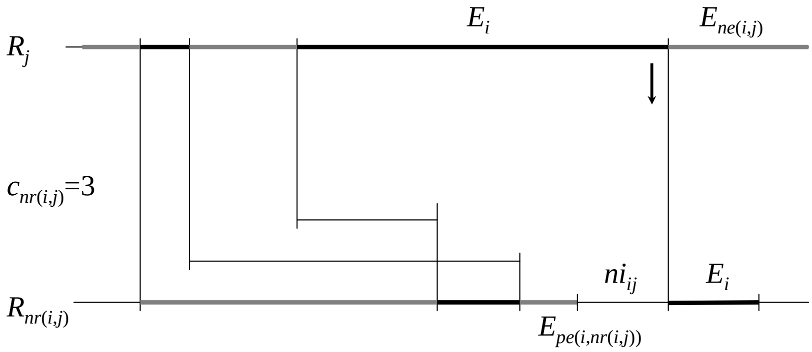

Figure 5.1 Illustration of no-input

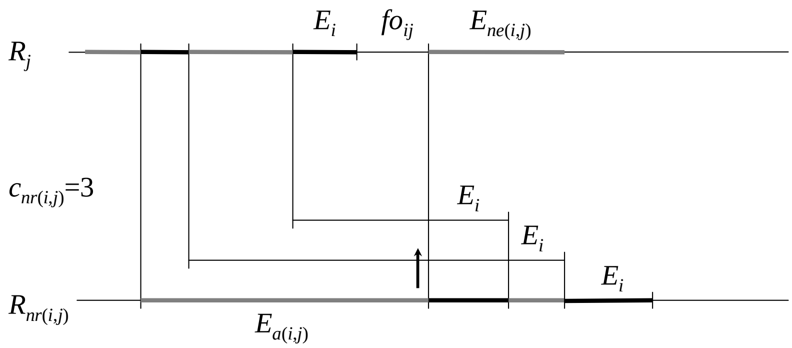

Figure 5.2 Illustration of full-output

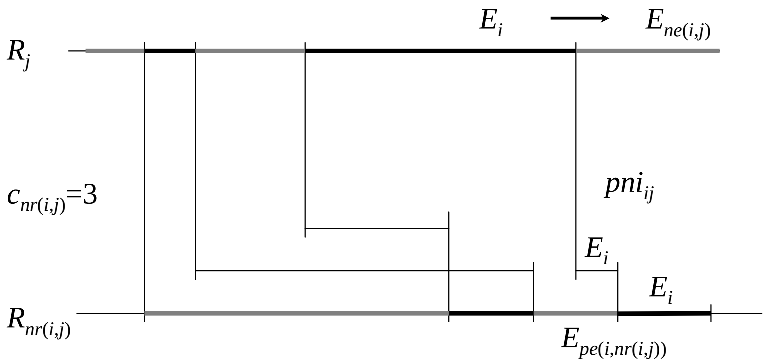

Figure 5.3 Illustration of potential no-input

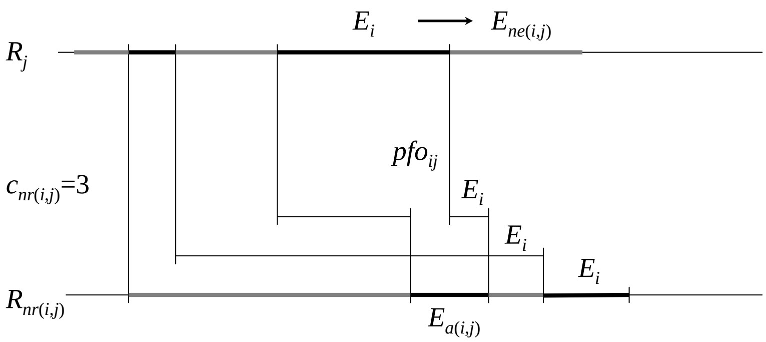

Figure 5.4 Illustration of potential full-output

5.3.2 Generation of perturbations

Our objective is to evaluate the effect of the change of the mean of the operation time of R_k. Therefore we have to provide

$$\hat{t}_{ik} = t_{ik} + \Delta t_{ik}; \quad i = 1, \ldots, N \tag{5.7}$$

so that,

$$E\left[\hat{t}\right] = \theta_k + \Delta\theta_k \tag{5.8}$$

Δt_{ik} can be generated based on the perturbation generation rules introduced by Suri and Zazanis (1988). It can be proved that if θ_k is a scale parameter of a distribution, then Δt_{ik} are calculated as follows:

$$\Delta t_{ik} = t_{ik} \frac{\Delta\theta_k}{\theta_k} \tag{5.9}$$

and if θ_k is a location parameter, then

$$\Delta t_{ik} = \Delta\theta_k \tag{5.10}$$

5.3.3 Perturbation propagation

The spread of the Δt_{ik} values in the event sequence table is based on the structure of the event sequence table and can be formally treated based on the evaluation of the event, preceding the beginning of the operation of entities. When an entity receives a perturbation, it is added to the perturbation generated throughout its operation. Three basic types of propagation operations can be defined for infinitesimal perturbations (Ho and Cao, 1991).

1) If the operation of an entity is preceded by a NI event, then this entity receives perturbation from the downstream resource. Based on Figure 5.1, its perturbation can be calculated as follows:

$$\delta_{i,nr(i,j)} = \delta_{ij} + \Delta t_{i,nr(i,j)} \tag{5.11}$$

2) If the operation of an entity is preceded by a FO event, then this entity receives perturbation from the upstream resource. Based on Figure 5.2, its perturbation can be calculated as follows:

$$\delta_{ne(i,j),j} = \delta_{a(i,j),nr(i,j)} + \Delta t_{ne(i,j),j} \tag{5.12}$$

3) If the operation of an entity is preceded by an OP event, then this entity receives perturbation from the preceding entity on the same resource. Based on Figure 5.3 and 5.4, its perturbation can be calculated as follows:

$$\delta_{ne(i,j),j} = \delta_{ij} + \Delta t_{ne(i,j),j} \tag{5.13}$$

We assume finite perturbations, which are, however, small enough not to hurt deterministic similarity. That is the way infinitesimal perturbation propagation rules (Ho, Cao and Cassandras, 1983) are used, although the perturbations are finite.

Finally if E_u on R_v is the last entity, the gradient of the throughput time of the sample path is calculated as follows:

$$\frac{\partial L(\theta_k, \xi)}{\partial\theta_k} = \frac{\delta_{uv}}{\Delta\theta_k} \tag{5.14}$$

5.3.4 Calculation of the validity range of deterministic similarity

Deterministic similarity holds if the nominal and the perturbed sample path are equal. The calculation of the validity limits requires the calculation of the highest and the lowest value of $\Delta\theta_k$ for which the original and the perturbed event sequence tables are identical. This occurs if as a consequence of the appearance of perturbations:

- existing NI or FO do not disappear,
- new NI or FO do not appear,
- there is no overtake (OT) of entities at queues which are fed by more than one resource.

The conditions on the non-disappearance of NI and FO can easily be checked. The change of the duration of FO and NI, due to the appearance of the perturbations, should be less than the actual duration of NI and FO, that is,

$$\Delta ni_{ij} = \delta_{pe(i,nr(i,j)),nr(i,j)} - \delta_{ij} \le -ni_{ij} \tag{5.15}$$

$$\Delta fo_{ij} = \delta_{a(i,j),nr(i,j)} - \delta_{ij} \ge -fo_{ij} \tag{5.16}$$

The condition on the appearance of new NI and new FO requires some further remarks. A new NI appears before E_i on $R_{nr(i,j)}$ if the waiting time of E_i upon arrival to $R_{nr(i,j)}$ will decrease to 0 (see Figure 5.3). This situation can be approximated by saying that it is enough to check the decrease of the waiting time when E_i finds PNI, that is, when there is just one entity in the queue of $R_{nr(i,j)}$. It is very probable that limits will be imposed by entities waiting for the end of one operation instead of, by entities waiting for the end of two or more operations. This is not necessarily true, because a waiting time caused by one long operation can be longer than a waiting time caused by two or more short operations. Considering that generally small (but finite) perturbations occur and operation times with relatively small variances are applied, this approximation may give good results. Similarly, when the appearance of FO is examined, only those cases will be checked, when upon arrival to a resource, there is only one free space in the queue (see Figure 5.4).

Ho and Cao (1991) recommended the application of the first order propagation rules to approximate the propagation of finite perturbations, on the same basis. The errors committed by this approximation are analyzed in various examples and are found acceptable for practical purposes (Ho, Cao and Cassandras, 1983). An exact treatment is given by Koltai (1992) by introducing the two-level FO and NI matrices. The approximation by PNI and PFO, however, may considerably facilitate data management at the calculations. Applying the first order approximation of validity limits, the conditions on the non-appearance of NI and FO are the following;

$$\Delta pni_{ij} = \delta_{pe(i,nr(i,j)),nr(i,j)} - \delta_{ij} \ge -pni_{ij} \tag{5.17}$$

$$\Delta pfo_{ij} = \delta_{pe(a(i,j),nr(i,j)),nr(i,j)} - \delta_{ij} \le -pfo(i,j) \tag{5.18}$$

There will be no overtake at a queue which is fed by more than one resource if, as a consequence of perturbation, the order of arrival of entities will not change. Based on Figure 5.5 this condition is as follows:

$$ot_{ij} = b_{pe(i,j),2pr(pe(i,j),j)} - b_{i,2pr(i,j)} \tag{5.19}$$

and

$$\Delta ot_{ij} = \delta_{pe(i,j),2pr(pe(i,j),j)} - \delta_{i,2pr(i,j)} \le -ot_{ij} \tag{5.20}$$

Due to the form of perturbation propagation rules, and to the simple form of the perturbation generation rules for location and scale parameters, the validity limits of $\Delta\theta_k$ can be easily calculated.

Let s$\{i,j\}$ be the index set of those E_m on R_k whose generated perturbation participate in the perturbation of E_i at R_j. In case of scale parameters equation (5.15) has the following form,

$$\frac{\Delta\theta_k}{\theta_k}\left[\sum_{s\{pe(i,nr(i,j)),nr(i,j)\}} t_{ik} - \sum_{s\{i,j\}} t_{ik}\right] \le -ni_{ij} \quad if \quad q_{nr(i,j)} = 0 \tag{5.21}$$

Let us introduce the following coefficient for the expression of the accumulated perturbations,

$$D_{ij}^{(\mathrm{NI})} = \sum_{s\{pe(i,nr(i,j),nr(i,j)\}} t_{ik} - \sum_{s\{i,j\}} t_{ik}; \quad if \quad q_{nr(i,j)} = 0 \tag{5.22}$$

Using the notation introduced with the help of (5.22), equation (5.21) can be written as,

$$\frac{\Delta\theta_k}{\theta_k} \le -\frac{ni_{ij}}{D_{ij}^{(\mathrm{NI})}} \tag{5.23}$$

After similar transformation of (5.16), (5.17), (5.18), and (5.20), furthermore introducing, $D_{ij}^{(\mathrm{FO})}$, $D_{ij}^{(\mathrm{PNI})}$, $D_{ij}^{(\mathrm{PFO})}$ and $D_{ij}^{(\mathrm{OT})}$ we get the following limits for the validity of deterministic similarity of the sample path,

$$UL(\theta_k,\xi) = \theta_k \cdot MIN\left[-\frac{ni_{ij}}{D_{ij}^{(\mathrm{NI})}}; -\frac{fo_{ij}}{D_{ij}^{(\mathrm{FO})}}; -\frac{pni_{ij}}{D_{ij}^{(\mathrm{PNI})}}; -\frac{pfo_{ij}}{D_{ij}^{(\mathrm{PFO})}}; -\frac{ot_{ij}}{D_{ij}^{(\mathrm{OT})}}\right] \tag{5.24}$$

$$D_{ij}^{(\mathrm{NI})} \ge 0;\ D_{ij}^{(\mathrm{FO})} \le 0;\ D_{ij}^{(\mathrm{PNI})} \le 0;\ D_{ij}^{(\mathrm{PFO})} \ge 0;\ D_{ij}^{(\mathrm{OT})} \ge 0$$

and

$$LL(\theta_k,\xi) = \theta_k \cdot MIN\left[-\frac{ni_{ij}}{D_{ij}^{(\mathrm{NI})}}; -\frac{fo_{ij}}{D_{ij}^{(\mathrm{FO})}}; -\frac{pni_{ij}}{D_{ij}^{(\mathrm{PNI})}}; -\frac{pfo_{ij}}{D_{ij}^{(\mathrm{PFO})}}; -\frac{ot_{ij}}{D_{ij}^{(\mathrm{OT})}}\right] \tag{5.25}$$

$$D_{ij}^{(\mathrm{NI})} \ge 0;\ D_{ij}^{(\mathrm{FO})} \le 0;\ D_{ij}^{(\mathrm{PNI})} \ge 0;\ D_{ij}^{(\mathrm{PFO})} \le 0;\ D_{ij}^{(\mathrm{OT})} \ge 0$$

In case of location parameters equation (5.15) is as follows,

$$\theta_k\left[\sum_{s\{pe(i,nr(i,j),nr(i,j)\}} 1 - \sum_{s\{i,j\}} 1\right] \le -ni_{ij} \tag{5.26}$$

Introducing

$$D_{ij}^{(\mathrm{NI})} = \sum_{s\{pe(i,nr(i,j),nr(i,j)\}} 1 - \sum_{s\{i,j\}} 1; \quad if \quad q(nr(i,j) = 0 \tag{5.27}$$

and applying the previous procedure again for FO, PNI, PFO and OT, (5.24) and (5.25) also hold.

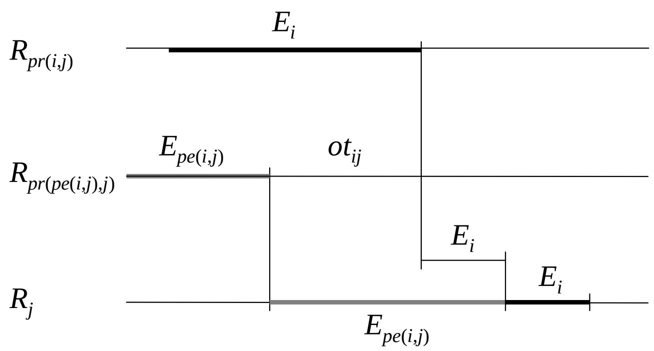

Figure 5.5 Illustration of the overtake possibility

5.4 Gradient of the mean waiting time in the queue

The presented analysis can easily be extended for the gradient and validity range calculation of the waiting time in queue.

Based on Figure 5.3 and 5.4 the waiting time of E_i in the queue of $R_{n(j)}$ is the following:

$$W_{ij}(\theta_k,\xi) = b_{pe(i,nr(i,j)),2nr(i,j)} - b_{i,2j} \tag{5.28}$$

and the gradient is

$$\frac{\partial W_{ij}(\theta_k,\xi)}{\partial \theta_k} = \frac{\partial b_{pe(i,nr(i,j)),2nr(i,j)}}{\partial \theta_k} - \frac{\partial b_{i,2j}}{\partial \theta_k} \tag{5.29}$$

Expression (5.29) shows that the gradient of the waiting time in queue of an entity is the difference of the gradient of the finishing times of two different entities. It looks as if it were the difference of the gradient of the throughput time of two sub processes of the whole process. These gradients can easily be calculated according to the presented method, applying the appropriate indices in (5.14).

As the gradients of the sub-processes are valid within the validity range of the throughput time gradient of the whole process, the same validity will be true for the gradient of the waiting time in the queue of R_j as well.

5.5 Convergence properties of the gradient estimates

So far, we have presented the calculation of the gradients and validity limits of one sample path. The estimate of these characteristics can be carried out correctly only if (5.1) holds. The conditions of interchangeability of differentiation and expectation are a constant research topic of PA (see for example Heidelberg et al., 1988; Glasserman, 1991).

If deterministic similarity cannot be assumed than, the extended perturbation analysis (EPA) can be applied (Ho and Li, 1988). It is based on the stochastic similarity concept and consists of a cut-and-paste method which constructs a new nominal path. This nominal path is created by removing certain parts of the original nominal path. These removed parts should start and end with the same system state and the resulting reconstructed nominal path should be deterministically similar to the perturbed one. A dual version of this approach (paste and insert), and additionally a state and event matching algorithm exists to facilitate the computationally efficient implementation (Ho and Cao, 1991).

With EPA, all previously mentioned gradients can be calculated, and their validity limits describe the range of θ_k, within which the constructed nominal path and the perturbed path are deterministically similar.

5.6 Computational results

In this chapter the implementation of sensitivity analysis with PA will be presented with the help of three examples. The first to examples show how PA can be performed in a discrete event simulation environment. The third example shows how PA can be applied if the event sequence table is given in the form of a Gantt chart (Wilson, 2003) of a production schedule.

5.6.1 Implementation of gradient calculation with PA in discrete event simulation

With the description of the gradient and validity range estimation, my objective was to give a formal treatment, which facilitates the implementation of perturbation analysis in a discrete simulation environment. Entities are passing through the system and at the occurrence of appropriate events, the beginning and ending operation times of entities preceding or following them on the current or next resource are evaluated. The queue content of the destination resources is also observed. The logic of the calculation can be seen in Figure 5.6.

E_i, after *leaving the queue* of R_j, takes perturbation from the upstream or downstream resources or from the entity preceding it on the same resource (P1). The origin of the received perturbation depends on the state of the queue of R_j at the termination time of the operation of E_i at the previous resource (I2). This is also the time epoch of evaluating the change of validity limits since all the information for calculating (5.21) is available due to the observation made at the preceding resource (I2, I3). The waiting time perturbation of this entity can also be calculated.

At the *end of the operation*, perturbation is propagated upstream if NI is observed at the destination resource. If this is the resource where the mean operation time is perturbed, then the operation time change is also generated (P2).

If the queue of the destination resource is blocked, then at the *end of blocking*, perturbation is propagated from the blocking resource.

If there is a possibility of overtake of entities, its effect on the validity has to be evaluated *before entering* the queue (C4).

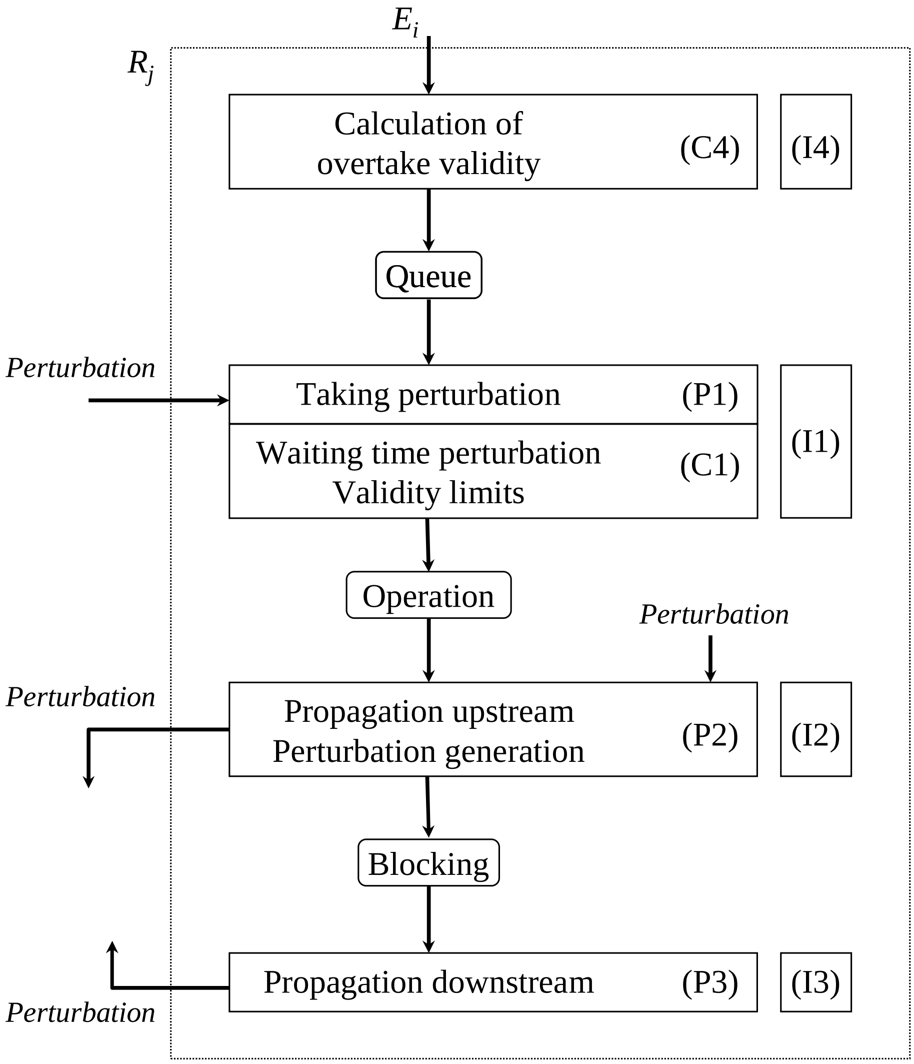

Figure 5.6 Implementation of perturbation analysis in discrete event simulation

The implementation of the presented method was carried out in the SIMAN IV simulation language (Pedgen, Shannon and Sadowski, 1991). The steps of the algorithms were performed by 4 event blocks, denoted by I1-I4 in Figure 5.6.

A simple problem is illustrated in Figure 5.7 and the simulation results of this example are summarized in Tables 5.2 and 5.3.

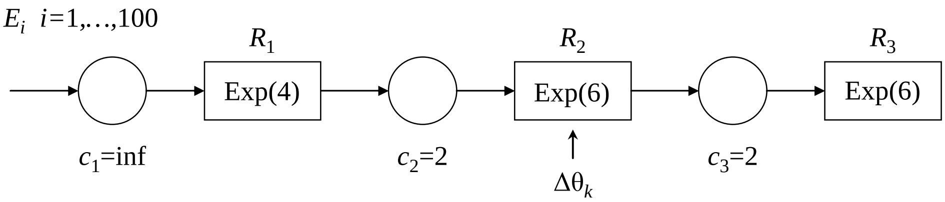

Figure 5.7 Transfer line example with three resources

The main characteristics of the problem are the following:

- three resources are used (R_1, R_2 and R_3),
- 100 entities passing through the system,
- queue capacities are the followings, c_1=4, c_2=2, c_3=2,
- FIFO queuing disciplines are applied at each queue,
- expected values of the exponential operation times (in minutes) are the followings, θ_1=4, θ_1=6, θ_1=6,
- all the 100 entities are available at the beginning of the process, in the first queue.

The estimates of the traditional outputs and the *gradients* with respect to θ_2, based on 100 simulation runs can be seen in Table 5.2. The rigorous examination of the distribution of the gradients is not discussed here.

Data concerning the calculation of the difference based on repetition of the experiment with $\theta_k+\Delta\theta_k$ is given in Table 5.3. Note, that in the table difference calculation is based on the relative change of operation time ($\Delta\theta_k/\ \theta_k$).

The results show the most attractive features of PA. Based on one experiment similar sensitivity information can be gained, than by brute force repetition of the simulation.

Table 5.2 Results of a single simulation run

IDENTIFIER	AVERAGE	STANDARD DEVIATION	.950 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
THROUGHPUT	423.00	42.90	8.51	310.00	519.00	100
TIME IN Q2	6.39	1.05	0.209	3.93	8.53	100
TIME IN Q3	3.58	0.866	0.172	1.40	6.27	100

IDENTIFIER	AVERAGE	STANDARD DEVIATION	.950 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
GR THROUGHPUT	196.00	54.90	10.90	80.20	358.00	100
GR TIME IN Q2	4.66	1.25	.248	2.27	8.32	100
GR TIME IN Q3	-1.66	0.713	.141	-4.29	-0.551	100
LOWER LIMIT	-2.014E-02	2.092E-02	4.150E-03	-9.375E-02	-3.477E-05	100
UPPER LIMIT	2.165E-02	2.141E-02	4.248E-03	3.604E-04	.122	100

The next example shows how the gradient of the throughput with respect to the routing parameter can be determined with PA in a flexible manufacturing system. I proved in Koltai and Lozano (1998) that the gradient of the throughput with respect to a routing parameter is a function of the gradient of the throughput with respect to the operation time of workstations and this gradient can be determined with PA. The following simple example illustrates the gradient calculation results.

Table 5.3 Comparison of difference calculation and gradient estimation

Identifier	θ_k=6	θ_k=6.6	Difference calculation	Gradient estimation	Difference (%)
THROUGHPUT	423	443	201	196	2.5
TIME IN Q2	6.39	6.86	4.73	4.66	1.5
TIME IN Q3	3.58	3.42	-1.66	-1.66	0.0

Suppose we have an FMS consisting of four single server workstations (R_j, j=1,...4) (see Figure 5.8). Two part types are produced (PR1 and PR2) and two types of pallets are used. Three pallets are available in each pallet class. Each part type uses a different pallet type (two pallet class). Each part type may follow two different routes. The sequence of operations with their respective mean operation times are given in Table 5.4. The last column of the table gives the routing mix applied in the experiments. The operation times are exponentially distributed. In total, 10 000 parts are produced. In this example, the workstations are the resources and the parts are the entities.

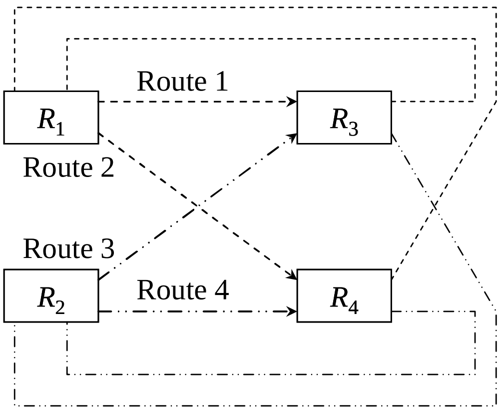

Figure 5.8 FMS sample problem

Table 5.4 Basic data of the FMS routing sample problem

Part type (t)	Route number (s)	Operation sequence (station no./mean operation time)	θ_{ts}
1	1	(1/0.5) → (3/0.5)	0.25
	2	(1/0.5) → (4/1.0)	0.75
2	1	(2/1.0) → (3/0.5)	1.00
	2	(2/1.0) → (4/1.0)	0.00

Figure 5.8 shows that there is not a unique load/unload station, however, R_1 is visited by every PR1 once and not visited at all by PR2. At the same time R_2 is visited by every PR2 once and not visited at all by PR1. This can be interpreted as if each class had a different load/unload station. In this case the relative visiting ratios of each class at its own load/unload station should be set to one. Alternatively, one could consider a virtual unique load/unload station (R_0) – with zero operation time – which is visited by all part types prior to starting their true sequence of operations.

The throughput as a function of θ_{11} and θ_{21} is shown in Figure 5.9. The values of this function were calculated using the MVA algorithm by Kobayashi and Gerla (1983), and two local maxima of this function were determined. One local maximum was found at $\theta_{1,1}$=0.544, $\theta_{2,1}$=1 which can very well be seen in the $\theta_{2,1}$=1 cut of Figure 5.9. The results of the gradient calculation at one of the points in the $\theta_{2,1}$=1 cut ($\theta_{1,1}$=0.25, $\theta_{2,1}$=1) are summarized in Table 5.5. A total of 30 simulation runs were carried out to calculate the mean and 95% confidence interval of the throughput gradients with respect to the routing mix ($\partial TP_0/\partial\theta_{ts}$). Change of the routing of any part type means that at least two routing ratios change with opposite signs, since $\theta_{1,1}+\theta_{1,2}=1$ and $\theta_{2,1}+\theta_{2,2}=1$. The data of Table 5.5 show that $\partial TP_0/\partial\theta_{1,2}<\partial TP_0/\partial\theta_{1,1}<0$ therefore directing a small fraction of PR1 from route 2 to route 1 ($\theta_{1,1}$ increases and $\theta_{1,2}$ decreases) would result in the increase of the throughput. This coincides with the results of Kobayashi and Gerla (1983) because they found a local maximum at $\theta_{1,1}$=0.544 which is higher than the $\theta_{1,1}$=0.25 value applied in the experiment. Table 5.5 also shows that $\partial TP_0/\partial\theta_{2,2}<\partial TP_0/\partial\theta_{2,1}<0$ therefore directing a small part of PR2 from route 2 to route 1 ($\theta_{2,1}$ increases and $\theta_{2,2}$ decreases) would result in the increase of the throughput. This is not possible because $\theta_{2,1}$ cannot be higher than 1 but the gradient correctly indicates that the throughput function in the $\theta_{1,1}$ cut reaches a feasible maximum at $\theta_{2,1}$=1.

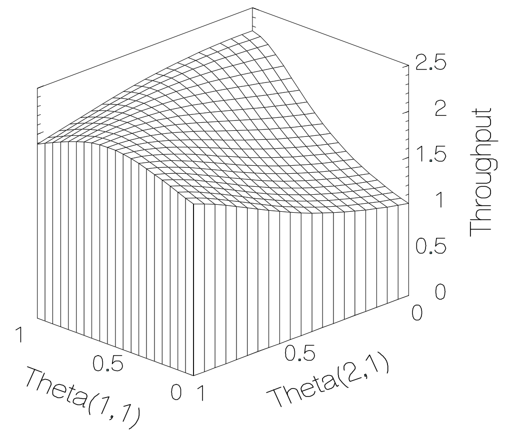

Figure 5.9 Throughput as a function of routing parameters

Table 5.5 Gradient information at $\theta_{1,1}$=0.25, $\theta_{2,1}$=1

$\partial TP_0/\partial\theta_{ts}$	Route (*s*)	
	s=1	*s*=2
t=1	-0.3496 0.0100	-0.8011 ±0.0423
t=2	-1.1020 ±0.0435	-1.5549 ±0.2148

Figures 5.10 and 5.11 show the throughput as a function of the routing mix. In Figure 5.10, $\theta_{2,1}$ is held constant and equal to the value of the local maximum (1.00). The system throughput and the corresponding directional derivative along $\theta_{1,1}$ are shown in the same chart. The change of routing mix means that $\theta_{1,1}$ increases from 0.00 to 1.00, while at the same time $\theta_{1,2}$ decreases from 1.00 to 0.00. The directional derivative ($\partial TP_0/\partial\theta_{1,1}-\partial TP_0/\partial\theta_{1,2}$) contains the effect of both changes.

This derivative indicates a maximum between 0.45 and 0.55 which agrees with the results of Kobayashi and Gerla (1983). Figure 5.11 shows the same function but with the other directional derivatives ($\partial TP_0/\partial\theta_{2,1}-\partial TP_0/\partial\theta_{2,2}$). This derivative indicates that when $\theta_{1,1}$<0.45 then the throughput function in the different $\theta_{1,1}$ cuts reaches a feasible maximum while when $\theta_{1,1}$>0.55 then the throughput function in the different $\theta_{1,1}$ cuts reaches a feasible minimum. It can also be seen that in the neighborhood of the optimum the directional derivatives do not give statistically significant results about the sign of the gradient. This is due to the fact that in this example the throughput function is very flat around the local maximum.

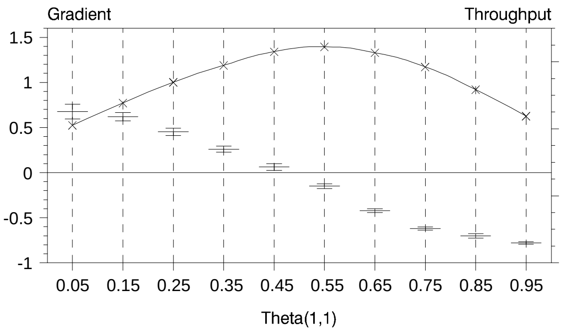

Figure 5.10 The throughput and the $\partial TP_0/\partial\theta_{1,1}-\partial TP_0/\partial\theta_{1,2}$ values in the $\theta_{2,1}$=1 cuts

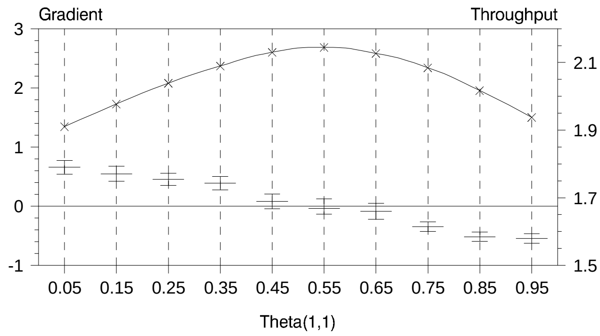

Figure 5.11 The throughput and the $\partial TP_0/\partial\theta_{2,1}-\partial TP_0/\partial\theta_{2,2}$ values in the $\theta_{2,1}$=1 cuts

5.6.2 Implementation of gradient calculation when a production schedule is given

In this chapter I illustrate the application of PA when the event sequence table is determined by a short term production schedule of a production system. The production schedule is described with a Gantt chart. The starting and completion time of each operation on each workstation is given, that is, the $b_{i,j}$ and $b_{i,2j}$ values are known. In this case the $nr(i,j)$, $pr(i,j)$, $ne(i,j)$, $pe(i,j)$ $a(i,j)$ values, furthermore, the $ni(i,j)$, $fo(i,j)$, $pni(i,j)$, $pfo(i,j)$ and $ot(i,j)$ values must be determined from the data of the Gantt chart. The details of the generation of all these data, based on the concept of *virtual queue*, can be found in Koltai (1992) and in Koltai, Larraneta and Onieva (1994). No matter, however, how the schedule is determined (with discrete event simulation or with any deterministic scheduling technique) if the $b_{i,j}$ and $b_{i,2j}$ values are known, the gradient of the throughput time or the gradient of the waiting time and the corresponding validity ranges can be determined with PA.

I have implemented PA for the examination of some critical waiting times in an automated continuous steel casting process. The queuing network representation of the process is illustrated in Figure 5.12.

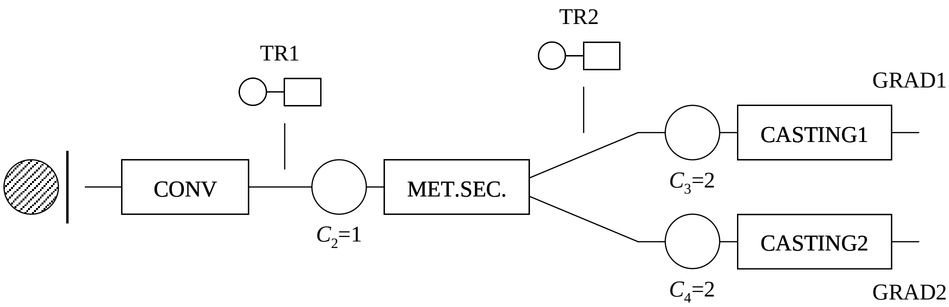

Figure 5.12 Queuing network representation of the steel casting process

The system consists of four workstations, and transforms pig iron into steel slabs (Díaz et al., 1991). The production process manufactures about 40 steel slabs a day. The daily production schedule is generated by a heuristic. Entering the system the first work station is a converter where high pressure oxygen is injected into a furnace at high temperature to reduce the carbon content of the iron. From here a transporter (TR1) carries the workload to the second workstation. The queue in front of the second workstation has a capacity of one work load. If this machine is occupied the converter is blocked. At the second workstation chemical treatment of the steel is carried out (secondary metallurgical process). From here a second transporter (TR2) carries the workload to one of the two feeders of the continuous casting machines according to the production program. In front of the feeders there are queues with capacity of two workloads. These queues are called "WAIT POSITION", and those are critical points of the process. If the melted steel has to wait more than 25 minutes then the steal freezes and has to be re-melted again. If there is no workload available then the continuous casting process breaks, and the size and quality of the steel slabs will not meet the technological requirements. For security reasons the minimum waiting times in the 3rd and 4th queue are 5 minutes. The heuristic provides a feasible production plan, but perturbations may occur especially at the first workstation, where an on-line quality check automatically modifies the operation time of the converter.

The presented queuing network is a special mixture of a transfer line and a general type queuing network. There are no assembly type nodes but routing information is necessary due to branching after the second workstation. The number of resources can be kept at 4 by ignoring the transporters under certain conditions. The planned workload never exceeds 100. At these parameters the CPU time for calculating the gradients and the validity limits is negligible.

I applied perturbation analysis for the examination of the following two problems:

a) I determined that range of the operation time of a workload at the first workstation, within which, break of sequence or excess waiting time do not occur.

b) I analyzed the effect of a given finite perturbation on the "WAIT POSITION" and on the throughput time. This waiting time is a linear function of the operation time with constant slope within the calculated validity range.

Each problem is solved by PA. The logic of the calculation is illustrated in Figure 5.13.

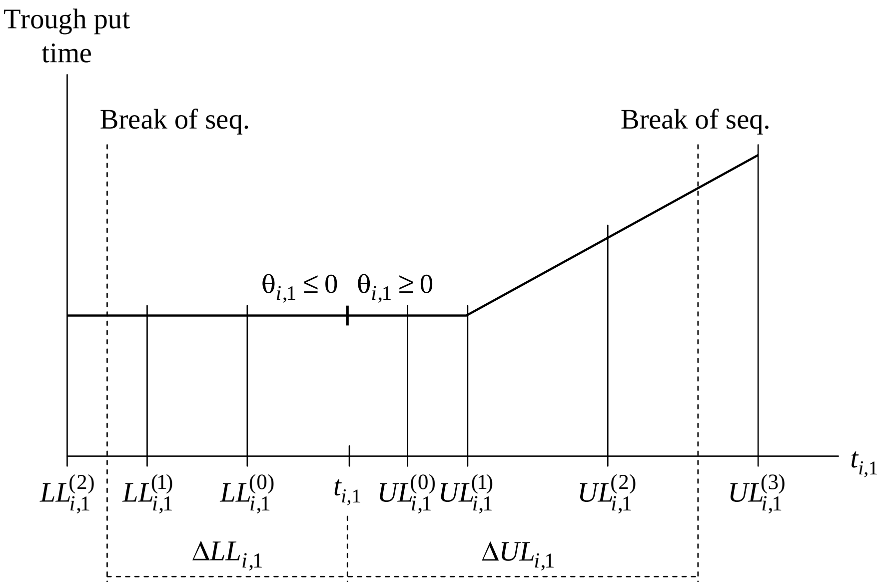

Figure 5.13 Sensitivity analysis of the break of sequence

An infinitesimal perturbation is introduced at workload i on machine j=1. The validity range is calculated ($LL^{(0)}_{i,1}$, $UL^{(0)}_{i,1}$) and the waiting time is checked in the "WAIT POSITION". If there is no problem then a perturbation equals to the upper limit of deterministic similarity is introduced, the new Gantt chart is generated with the help of the perturbation propagation rules and the calculation is started again. The whole process goes on until the break of sequence limits ($\Delta LL_{i,1}$, $\Delta UL_{i,1}$) are found.

Table 5.6 shows the output list of the analysis. In the table, the result of each iteration step is presented. At the end a comment on the feasibility of the introduced perturbation is made. If there is no infeasibility then the new Gant chart is generated, and the program is ready to receive the next perturbation. If the perturbation is infeasible then indications are given, on how to modify the operation schedule to get feasible schedule again.

A table about the upper and lower limits of the feasibility of all the operation times at the various workstations can be generated for an overall preliminary sensitivity evaluation of the schedule. A part of the break of sequence sensitivity list is presented in Table 5.7.

Table 5.6 Calculation steps of sensitivity analysis

RESULTS OF SENSITIVITY CALCULATION

THE PERTURBATION IS INTRODUCED AT MACHINE: CONV
SERIAL: 5
WORKLOAD: 6
SIZE: 18

STEP	TIME	ΔLL	ΔUL	GRAD1	GRAD2
1	30	0	8	0	0
2	38	-8	6	0	0
3	44	0	7	0	1

*** BREAK AT: 45 min

*** SERIAL: 5 WORKLOAD: 6 WAIT POSITION: 4 min

*** SERIAL: 5 WORKLOAD: 8 WAIT POSITION: 26 min

Table 5.7 Break of sequence sensitivity table

BREAK OF SEQUENCE SENSITIVITY

MACHINE	SERIAL	WORKLOAD	TIME	$\Delta LL_{i,1}$	$\Delta UL_{i,1}$
CONV	1	1	30	16	30
CONV	1	2	30	0	36
CONV	1	3	30	30	30
CONV	1	4	30	0	35
CONV	1	5	30	30	50
CONV	1	6	30	30	50
CONV	1	7	30	30	30
CONV	2	1	30	0	30
CONV	2	2	30	30	30
CONV	2	3	30	30	30
.	.	.	.	.	.
.	.	.	.	.	.

5.7 Conclusions of Chapter 5

In this chapter an algorithm is presented to calculate the validity range of deterministic similarity of a sample path of a discrete event dynamic system when a single perturbation is introduced at the operation times. Based on the calculated range, sensitivity analysis concerning both the gradient of the throughput and some special technological feasibility of an operation schedule are analyzed. The presented method completes scheduling models which fail to give sensitivity information.

Due to the big amount of input data the application is recommended in systems where the number of entities is relatively small. This is the case in many types of manufacturing systems when small scale, technology intensive production is performed. The efficiency of the algorithm can be increased, either by taking advantage of the information incorporated into the model used for the generation of the operation schedule, or by exploiting some special dual characteristics existing among no-input and full-output activities. The suggested calculations are illustrated with two sample problems. Furthermore, the successful implementation of the method at a real continuous steal manufacturing process has also provided to show the application possibilities of validity range calculation with PA in practice.

We note that the proposed method provides gradient information of the performance measure related to a production schedule but fails to provide information about the methods of rescheduling. The gradient information indicates the requirement for rescheduling, which afterwards must be carried out with any methods available in the literature (see for example Pfeiffer et al., 2008).

As a summary, based on Chapter 5, the following scientific result can be formulated:

Result 4

If a production schedule is generated by the single simulation run of a discrete event simulation model then the gradient of the throughput time $L(\theta,\xi)$ with respect to the operation time θ_k is valid if the change of θ_k is within the feasible range. I have derived formula (5.24) for the calculation of the feasible upper bound and formula (5.25) for the calculation of the feasible lower bound of θ_k.

The calculations of the validity range are based on the definition of the no-input, full-output, potential no-input, potential full-output and overtake matrixes. The basic data for the calculation are generated by discrete event simulation. The data for the defined matrixes, however, can also be obtained from any production schedule if the schedule is given in the form of a Gantt chart. This way the proposed validity range calculation can be used for the examination of the robustness of any production schedule.

The definition of the sensitivity range of the gradient of the throughput time and the algorithm for calculating the gradient and range is published in Koltai (1992) and Koltai, Larraneta and Onieva (1993, 1994). The generalization of sensitivity range calculation for any schedule which is defined by a Gantt chart is discussed in Koltai (1992) and Koltai et al. (1994). The practical application of the gradient calculation and the extension of the results to other performance measures are presented in Koltai, Larraneta and Onieva (1993), Koltai and Lozano (1996, 1998), and Koltai (1994).

6 SENSITIVITY OF A PRODUCTION SEQUENCE TO INVENTORY COST CALCULATION METHODS IN CASE OF A SINGE RESOURCE, DETERMINISTIC SCHEDULING PROBLEM

Scheduling rules are frequently used either to determine the optimal production sequence or as heuristics to get acceptable solutions in complex sequencing situations. Single resource scheduling is a simple special case of practical scheduling situations. Frequently, however, complex systems can be approximated as single resource scheduling problems. Many times the objective of scheduling is the minimization of inventory holding cost. There are several ways to calculate or approximate the value of inventory holding cost. This chapter shows that scheduling decisions can be very insensitive to the method of inventory holding cost calculation. Financial conditions strongly influence the financial result of the company but not necessarily relevant at scheduling decisions. The case of a calendar manufacturer illustrates this statement, and helps to derive several new scheduling rules. The results of this chapter are based on the papers of Koltai (2006) and Koltai (2009).

6.1 Introduction

In practice, operations management objectives frequently contradict financial objectives. For example, operations management might be interested in high inventory level to satisfy fluctuating demand while financial management might be interested in low inventory level to reduce inventory holding cost. At times, operations management is interested in low capacity utilization of service facilities to reduce waiting time of customers while financial management is interested in high machine utilization to show high return on investment of expensive resources. There are cases, however, when the contradiction between operational and financial objectives is only apparent. This chapter presents a production scheduling situation in which scheduling decision is relatively insensitive to certain financial considerations.

The research presented in this chapter was motivated by the production scheduling problem of a small calendar manufacturer. Raw materials for calendars arrive to the production process at the required time, and their cost has to be paid to the supplier upon arrival. Income, however, is received only at the delivery time of finished products. All calendars are prepared for a fixed common due date around the last quarter of the year. Delay is not allowed because calendars are perishable items, generally can only be sold around the beginning of the New Year. Based on the analysis of the production process the cutting machine was identified as the bottleneck of the system. Since the company manufactures without any income in the first three quarters of the year, minimization of inventory holding cost is a major objective for production scheduling.

The objective of this chapter is to provide production schedules, which minimize inventory holding cost of the calendar manufacturer and to analyze how the optimal schedule is influenced by the method of inventory holding cost calculation.

This problem outlined above is a single machine scheduling problem with fixed common due date and sequence independent setup times. The scheduling criterion is to minimize a function of total lateness. However, since all calendars are shipped on time, an earliness related cost function must be minimized (Baker and Scudder, 1990). Depending on the calculation of the cost of financing raw materials a linear or a non-linear objective function is appropriate.

Sequencing jobs on a single machine is a well-known and thoroughly studied problem in the literature. Since the appearance of the classical sequencing rules of Smith (1956) several other special cases for optimizing flow time and tardiness related objective functions have been solved (e.g. Baker, 1974; Convey, Maxwell and Miller, 1976). However, as a consequence of the combinatorial nature of sequencing problems most of the practically relevant situations can only be handled by heuristics.

Sequencing with a common due date for all jobs is an important set of sequencing problems (Bector, Gupta and Gupta, 1991). If the common due date is fixed in advance, then the problem is more tractable but still most of the problems are NP hard. When the due date is fixed in advance and it is higher than the completion time of each job, the problem is reduced to an earliness related single machine sequencing problem.

In most cases, the objective of scheduling is to improve some cost related performance measures. If inventory holding cost is minimized, the cost of capital is an important element of the calculation. Inventory holding cost is generally calculated with the help of inventory holding rate. (see, for example Anderson, 1994; Waters, 1996; Wollmann, Berry and Whybarck, 1997) This rate expresses the percentage of the cost of materials which should be considered as holding cost.

The application of inventory holding rate is a pragmatic approach. Generally, there are several causes of the change of inventory holding cost with respect to the change of inventory level. Instead of identifying all these causes and determining the effect of each cause, an aggregate measure, the inventory holding rate is applied. Sometimes, the cost of capital tied up by inventory can be simply calculated, especially if inventory is financed from credit. In this case, a more accurate inventory holding cost calculation can be given by calculating the exact value of the interest. There are several ways of determining this interest. All these methods can be approximated by two extreme situations: interest is not compounded, and interest is continuously compounded. In the first case the objective function is a linear function of flow time while in the second case the objective function is non-linear (exponential). Inventory holding cost approximated by these two situations provides a lower and an upper approximation of the exact value of interest for all practically relevant situations.

If the inventory cost is financed directly and completely from credit, and credit conditions are known, then the appropriate cost should be calculated using the actual credit payment. If, however, conditions are not known at the time of the inventory holding cost estimation, or it is not decided yet, how inventory should be financed, then the lower and upper approximation of inventory holding cost is equivalent to the estimation of opportunity cost.

Scheduling based on a non-linear objective function is widely discussed in the literature (e.g. Rinnoy Kan, Lageweg, Lenstra, 1975; Sung and Joo, 1992; Alidaee, 1993). Since most of these problems are also NP hard, generally branch and bound based heuristics are suggested for the solutions. In some special cases (like the one presented in this chapter), efficient algorithms using the Adjacent Pair Interchange (API) principle can be applied (Andreson, 1994).

Successful applications of classical scheduling theory results are constrained, on the one hand, by several restricting conditions and, on the other hand, by the complex and dynamic nature of reality (McKay, Safayeni and Buzacott, 1988). However, in some simple situations, the application of scheduling rules may lead to better results than random or habit-driven sequencing of jobs. When the situation is complex, sensitivity analysis can help to outline the validity of a simple approach by filtering out the non-relevant complicating factors (constraints, parameter). For this reason, sensitivity analysis is used frequently in various areas of management when the complexity of a problem must be reduced (see for example Borgonovo and Peccati, 2004; Borgonovo and Peccati, 2006; Koltai and Terlaky, 2006).

In the following, firs, two scheduling rules for minimizing inventory holding cost are derived. First, the interest on the tied up capital by inventory is not compounded, next the interest is continuously compounded. Next, sensitivity of the schedule to the method of interest calculation is analytically examined, and the original problem is extended for different due dates. Finally, the application of the suggested rules in the case of the calendar manufacturer is presented, and some general conclusions are provided. Notations used in this chapter are summarized in Table 6.1.

Table 6.1 Summary of notation of Chapter 6

Subscript:		
i	−	index of jobs (i=1,…,N).
Parameters:		
N	−	number of jobs,
t_i	−	operation time of job i,
$f(t_i)$	–	transformed exponential operation time of job i,
c_i	−	raw material cost of job i,
d_i	–	delivery date of job i,
D	−	common delivery date of all jobs,
r	−	periodic yearly interest rate,
q	−	continuous yearly interest rate.
Variables:		
T_i	−	flow time of job i,
T_0	−	flow time of the last job directly preceding jobs i and j,
R_i	–	residence time of job i, i=1,…,N,
I_i	–	inventory holding cost of job i, i=1,…,N,
W	−	objective function value for all jobs except jobs i and j.

6.2 Minimization of inventory holding cost with common due dates

The major element of inventory holding cost, in the case of the calendar manufacturer, is the cost of capital tied up by raw materials. Raw materials are financed from credit; therefore the cost of capital can be approximated by the interest on the cost of raw materials. For the sake of simplicity, inventory holding cost will be approximated by the interest accumulated on the cost of raw materials until the arrival of income for finished products in the rest of this chapter.

In the case of the calendar manufacturer it is assumed that raw material for job i arrives only when its cutting operation is started. The cost of raw materials is paid upon arrival, and all jobs can be finished for the delivery date. According to this assumption, raw material delivery is organized as a just-in-time system. This delivery process implies that no holding cost is incurred for the raw materials before the start of the manufacturing operation. Let delivery due date (D) be equal to the sum of the operation times of all jobs, that is,

$$D = \sum_{k=1}^{N} t_k \tag{6.1}$$

If all jobs finished earlier than the delivery due date, then for all jobs a fixed inventory holding cost incurs. This cost is not influenced by scheduling therefore the above simplification is acceptable.

Inventory holding cost of a job is calculated based on the interest incurred on the cost of its raw material during the period between the starting time of cutting operation and the delivery time of the finished products. Let us call this period the *residence time* (R_i) of job i. If T_i is the flow time of job i and t_i is the operation time of job i, the starting time of the cutting operation of job i is equal to T_i–t_i. The residence time of the raw material of job i is calculated as follows,

$$R_i = \sum_{k=1}^{N} t_k - (T_i - t_i) \tag{6.2}$$

The residence time is illustrated in Figure 6.1.

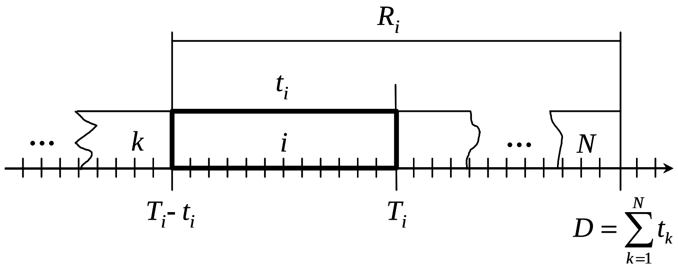

Figure 6.1 Illustration of the residence time of job i

For a residence time related objective function an adjacent pair interchange (API) algorithm can provide optimal solution in finite calculation steps. Figure 6.2 shows the principle of API algorithms.

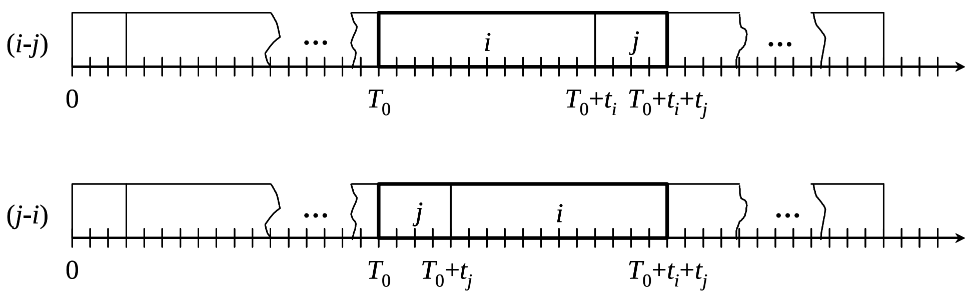

Figure 6.2 Interchange of job i and job j

If jobs i and j are adjacent jobs, interchanging these jobs will not influence the residence times of the other jobs, that is, the inventory holding cost of the other jobs will not change either. The interchange of adjacent jobs can be continued as long as objective function improves. If inventory holding cost of a job increases with the increase of residence time, then there is a finite possibility of interchanging adjacent jobs, and consequently the API algorithm converges to the optimum.

In the following, inventory holding cost is calculated in two different ways. First, the cost of capital is calculated without compounding interest. Next, the cost of capital is determined by continuously compounded interest calculation.

6.2.1 Optimal schedule when interest is not compounded

If r is the yearly periodic interest rate and time is measured in days, inventory holding cost of a production sequence is the following,

$$\sum_{i=1}^{N} I_i = \sum_{i=1}^{N} c_i \frac{r}{365} R_i \tag{6.3}$$

The objective is to find a production sequence of N jobs which minimizes (6.3). If setup time is sequence independent and included in t_i, an API algorithm can be constructed to find the minimum for objective function (6.3).

Let us write objective function (6.3) for both cases of Figure 6.2; first for an i–j sequence and next for a j–i sequence of the indicated adjacent jobs. Let T_0 denote the time when the i and j job pair can be started. Furthermore, let W denote the total inventory holding cost of all jobs except i and j. To compare the inventory holding costs of the two sequences of Figure 6.2 only the inventory holding cost of jobs i and j must be detailed.

In the case of the i–j sequence of the jobs, job i is started at T_0, job j is started at T_0+t_i, and the objective function is the following,

$$OF^{(i-j)} = W + c_i \frac{r}{365} \cdot (D - T_0) + c_j \frac{r}{365} \cdot (D - T_0 - t_i) \tag{6.4}$$

In the case of the j–i sequence of jobs, job j is started at T_0, job i is started at T_0+t_j, and the objective function is the following,

$$OF^{(j-i)} = W + c_j \frac{r}{365} \cdot (D - T_0) + c_i \frac{r}{365} \cdot (D - T_0 - t_j) \tag{6.5}$$

The objective function improves by the change of the i–j sequence of jobs if inventory holding cost decreases, that is, if

$$OF^{(i-j)} - OF^{(j-i)} > 0 \tag{6.6}$$

Otherwise, the i–j sequence is optimal. Subtracting (6.5) from (6.4) we get the following simple condition for the optimality of the i–j sequence,

$$-c_j \frac{r}{365} t_i + c_i \frac{r}{365} t_j \le 0 \quad \rightarrow \quad \frac{t_j}{c_j} - \frac{t_i}{c_i} \le 0 \quad \rightarrow \quad \frac{t_i}{c_i} \ge \frac{t_j}{c_j} \tag{6.7}$$

Based on condition (6.7), it can be concluded that a schedule of jobs is optimal only if the jobs are sequenced according to a non-increasing order of the t_i/c_i values.

This result is the opposite of the classical weighted shortest processing time (WSPT) rule for minimizing weighted total flow time (Smith, 1956). The sequencing rule based on (6.7) can be called Weighted Longest Processing Time (WLPT) rule. Finally, it can be seen in condition (6.7) that the optimal sequence of jobs is independent of the interest rate r.

6.2.2 Optimal schedule when interest is continuously compounded

If interest is continuously compounded, the inventory holding cost of a job is calculated using the continuously compounded interest formula. In this case the inventory holding cost of job i is the following,

$$I_i = c_i (e^{R_i \frac{q}{365}} - 1) = c_i e^{R_i \frac{q}{365}} - c_i \tag{6.8}$$

where q is the yearly continuous interest rate and R_i is expressed in days. The objective is to find the job sequence which minimizes total inventory holding cost, that is, to find the minimum of the following function,

$$\sum_{i=1}^{N} I_i = \sum_{i=1}^{N} c_i e^{R_i \frac{q}{365}} - \sum_{i=1}^{N} c_i \tag{6.9}$$

The second term of (6.9) is independent of the production sequence, therefore, it is sufficient to find the minimum of the first summation, that is,

$$\mathrm{Min} \sum_{i=1}^{N} c_i e^{R_i \frac{q}{365}} \tag{6.10}$$

For objective function (6.10) the API principle again leads to the minimal inventory holding cost job sequence.

Based on Figure 6.2, the objective function for the i–j sequence of jobs is the following,

$$OF^{(i-j)} = W + c_i e^{\frac{q}{365}(D-T_0)} + c_j e^{\frac{q}{365}(D-T_0-t_i)} \tag{6.11}$$

In the case of the *j–i* sequence of jobs, the objective function is the following,

$$OF^{(j-i)} = W + c_j e^{\frac{q}{365}(D-T_0)} + c_i e^{\frac{q}{365}(D-T_0-t_j)} \tag{6.12}$$

The objective function improves by the change of the *i–j* sequence of jobs if inventory holding cost decreases, that is, if

$$OF^{(i-j)} - OF^{(j-i)} > 0 \tag{6.13}$$

Otherwise, the *i–j* sequence is optimal. Subtracting (6.12) from (6.11) the following condition is obtained for the optimality of the *i–j* sequence,

$$c_i e^{\frac{q}{365}(D-T_0)} + c_j e^{\frac{q}{365}(D-T_0-t_i)} \le c_j e^{\frac{q}{365}(D-T_0)} + c_i e^{\frac{q}{365}(D-T_0-t_j)} \tag{6.14}$$

Simplifying and rearranging condition (6.14) we get the following simple condition,

$$\frac{1-e^{-\frac{q}{365}t_i}}{c_i} \ge \frac{1-e^{-\frac{q}{365}t_j}}{c_j} \tag{6.15}$$

Condition (6.15) is similar to condition (6.7) except that operation time is transformed with the help of an exponential function.

Let us call the numerators in (6.15) *transformed exponential operation time* and consequently, the scheduling rule can be called Weighted Longest Transformed Exponential Processing Time (WLTEPT) rule. According to condition (6.15) the value of objective function (6.10) is minimal if jobs are sequenced according to a non-increasing order of the weighted transformed exponential operation times. The optimal sequence of jobs *now* depends on the value of the interest rate q.

6.2.3 Comparison of the optimal sequences

The two different interest calculations provide a lower and an upper bound for the possible value of inventory holding cost. If the interest is not compounded, then the smallest possible value of inventory holding cost is approximated. If the interest is calculated continuously during the residence time, then the highest possible value of inventory holding cost is approximated. In reality, inventory holding cost is between these two approximations.

The optimal production sequence of jobs is, however, not necessarily different for the different calculation methods. Conditions (6.7) and (6.15) provide optimality conditions for the sequence of two adjacent jobs. According to condition (6.7) the *i–j* sequence is optimal if the c_i/c_j ratio is smaller than the ratio of the corresponding *operation times*. According to condition (6.15) the *i–j* sequence is optimal if the c_i/c_j ratio is smaller than the ratio of the corresponding *transformed exponential operation times*. Consequently, the *i–j* sequence is optimal for both methods of interest calculations if the c_i/c_j ratio is smaller than both the ratio of corresponding operation times and the ratio of the corresponding transformed exponential operation times, that is,

$$\frac{c_i}{c_j} \le \frac{t_i}{t_j} \quad \text{and} \quad \frac{c_i}{c_j} \le \frac{1-e^{-\frac{q}{365}t_i}}{1-e^{-\frac{q}{365}t_j}} \tag{6.16}$$

To draw general conclusions about the robustness of the optimal sequence to the method of interest calculations the relationship of the right-hand sides of conditions (6.16) must be analyzed. This relationship can be made transparent by analyzing the transformed exponential operation time function

$$f(t) = 1 - e^{-\frac{q}{365}t} \tag{6.17}$$

Function (6.17) is depicted in Figure 6.3. It is straightforward to show by differentiation that the slope of the *f*(*t*) function is monotonically decreasing, that is,

$$\frac{f(t_i)}{t_i} \geq \frac{f(t_j)}{t_j} \quad \text{if} \quad t_i < t_j \tag{6.18}$$

Applying conditions (6.18) for the adjacent *i* and *j* jobs, it can be concluded that the *i*–*j* sequence is optimal for both type of interest calculations if either of the two following conditions holds,

$$1) \text{ If } \ t_i \leq t_j, \ \text{ then } \ \frac{c_i}{c_j} \leq \frac{t_i}{t_j} \leq \frac{f(t_i)}{f(t_j)} \tag{6.19}$$

$$2) \text{ If } \ t_i > t_j, \ \text{ then } \ \frac{c_i}{c_j} \leq \frac{f(t_i)}{f(t_j)} \leq \frac{t_i}{t_j} \tag{6.20}$$

Based on conditions (6.19) and (6.20), it can be seen that for the *i*–*j* adjacent pair of jobs

a) if $t_i \leq t_j$, then the *i*–*j* sequence is optimal for both methods of interest calculations only if the *i*–*j* sequence is optimal for the *first* type of calculation (interest is not compounded),

b) if $t_i > t_j$, then the *i*–*j* sequence is optimal for both methods of interest calculations only if the *i*–*j* sequence is optimal for the *second* type of calculation (interest is continuously compounded).

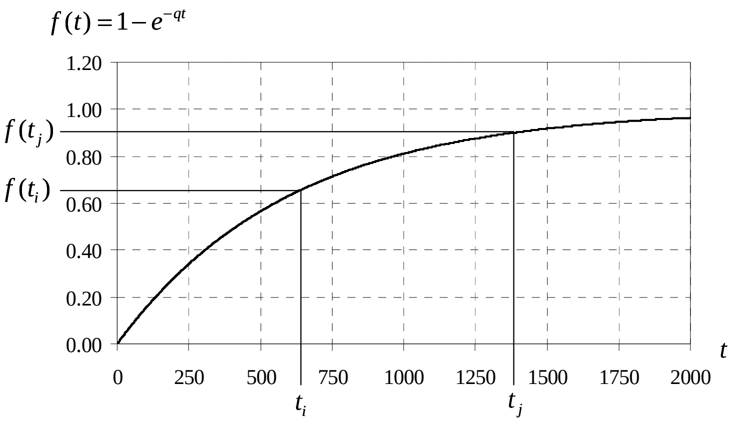

Figure 6.3 Transformed exponential operation time function (q=0.6)

In both cases the two optimal sequences are different if the ratio of the corresponding costs falls between the ratio of operation times and the ratio of transformed exponential operation times. In other words, the ratio of corresponding costs falls between the ratio of two *independent* values and the ratio of the corresponding two *dependent* values of the *f*(*t*) function.

Studying the *f*(*t*) function (see Figure 6.3) it can be concluded that this may occur very rarely in practice. There are *two* typical situations when the sequence of two adjacent jobs can be different in the optimal solutions of the two different interest calculation methods:

a) *Large difference between the operation times of two adjacent jobs*. As a consequence of the concave nature of the *f*(*t*) function if differences are large between the operation times of adjacent jobs then the differences between the ratio of the dependent values and the ratio of the independent values can be also large. The larger the difference between these two ratios is, the higher the possibility for the cost ratios to fall between the ratio of the dependent and the ratio of the independent values.

b) *Large continuous interest rate* (q). The larger the value of q, the larger the decrease of the gradient of the $f(t)$ function is. For relatively small values of q the $f(t)$ function is almost linear in the practically relevant range of t. Therefore, the difference between the ratio of two dependent values and the ratio of the corresponding two independent values is very small.

Figure 6.3 shows the $f(t)$ function for q=0.6. This high interest rate is used only to emphasize the shape of the function. Continuous interest rates in practice are much smaller. Furthermore operation times generally fall into the nearly linear section of the $f(t)$ function. Therefore, there is just a very small chance of having different optimal sequence of jobs for the different interest calculation methods. As a result, from a practical point of view, sequencing decisions are very insensitive to the method of interest calculation. This conclusion does not mean that the inventory holding costs for different methods of interest calculations are not different. But the best way of operating the system is generally not influenced by the corresponding inventory holding cost calculation method.

6.3 Extension of the calculation for different due dates of jobs

In Chapters 6.2.1 and 6.2.2 it was assumed that each job is delivered to the customer at a given common delivery date D. Even in case of the calendar manufacturer it is possible to give different due dates for each job or for some groups of jobs within a small range of the common delivery date.

In the following, it will be shown that although the difference of delivery dates slightly modifies the calculation of the optimal schedule the general conclusions of Chapter 6.2 are still valid. In the following it is assumed that each job has a different delivery date equal to d_i. Furthermore, it is also assumed that all jobs are delivered in time, that is, delay and the corresponding penalty cost are not considered.

6.3.1 Optimal schedule when delivery dates are different and interest is not compounded

Applying an API algorithm for the different delivery date situation the objective function must be written for an i–j sequence and next for a j–i sequence of the indicated adjacent jobs.

The objective function for the i–j sequence of the jobs is obtained if D is substituted by d_i in case of job i and by d_j in case of job j in equation (6.4), that is,

$$OF^{(i-j)} = W + c_i \frac{r}{365} \cdot (d_i - T_0) + c_j \frac{r}{365} \cdot (d_j - T_0 - t_i) \qquad (6.21)$$

The objective function for the j–i sequence of the jobs is obtained if D is substituted by d_i in case of job i and by d_j in case of job j in equation (6.5), that is,

$$OF^{(j-i)} = W + c_j \frac{r}{365} \cdot (d_j - T_0) + c_i \frac{r}{365} \cdot (d_i - T_0 - t_j) \qquad (6.22)$$

The change of the objective function as a consequence of the interchange of jobs i and j can be analyzed again by taking the difference between (6.21) and (6.22). It can easily be seen that due dates d_i and d_j drop out of the calculation and the optimal sequence can again be given by the WLPT sequence. That is, an i–j sequence is optimal if

$$-c_j \frac{r}{365} t_i + c_i \frac{r}{365} t_j \le 0 \quad \rightarrow \quad \frac{t_j}{c_j} - \frac{t_i}{c_i} \le 0 \quad \rightarrow \quad \frac{t_i}{c_i} \ge \frac{t_j}{c_j} \qquad (6.23)$$

as before in Chapter 6.2.1.

6.3.2 Optimal schedule when delivery dates are different and interest is continuously compounded

Applying an API algorithm for the different delivery date situation the objective function must be written again for an i–j sequence and next for a j–i sequence of the indicated adjacent jobs.

The objective function for the i–j sequence of jobs can be obtained if d_i and d_j substitute D in equation (6.11), that is,

$$OF^{(i-j)} = W + c_i e^{\frac{q}{365}(d_i - T_0)} + c_j e^{\frac{q}{365}(d_j - T_0 - t_i)} \tag{6.24}$$

The objective function for the j–i sequence of jobs can be obtained if d_i and d_j substitute D in equation (6.12), that is,

$$OF^{(j-i)} = W + c_j e^{\frac{q}{365}(d_j - T_0)} + c_i e^{\frac{q}{365}(d_i - T_0 - t_j)} \tag{6.25}$$

The objective function is improved by the change of the i–j sequence of jobs if inventory holding cost decreases. Subtracting (6.25) from (6.24) the following condition is obtained for the optimality of the i–j sequence of jobs,

$$c_i e^{\frac{q}{365}(d_i - T_0)} + c_j e^{\frac{q}{365}(d_j - T_0 - t_i)} \le c_j e^{\frac{q}{365}(d_j - T_0)} + c_i e^{\frac{q}{365}(d_i - T_0 - t_j)} \tag{6.26}$$

Simplifying and rearranging condition (6.25) we get the following condition,

$$\frac{1 - e^{-\frac{q}{365}t_i}}{c_i e^{\frac{q}{365}d_i}} \ge \frac{1 - e^{-\frac{q}{365}t_j}}{c_j e^{\frac{q}{365}d_j}} \tag{6.27}$$

Applying again notation $f(t)$ for the transformed exponential operation time we get the following,

$$\frac{f(t_i)}{c_i e^{\frac{q}{365}d_i}} \ge \frac{f(t_j)}{c_j e^{\frac{q}{365}d_j}} \tag{6.28}$$

Condition (6.28) shows that if interest is continuously compounded, due dates can influence the optimal order.

In the following, let us call the transformation of the due date with the help of the exponential function *exponential due date*. Condition (6.28) shows that in the case of different due dates the WLTEPT rule must be modified by including the corresponding exponential due date values in the weights.

6.3.3 Comparison of the optimal sequences

The analysis of the robustness of a schedule to the interest calculation method is not so straightforward now as it was in the case of common due dates. The condition for the optimal sequence of an i–j order in the case of continuously compounded interest calculation is determined by both the ratio of the *cost* of jobs and by the ratio of the *exponential due dates*.

Applying the previously demonstrated condition (6.18) and knowing that the exponential function is monotonically decreasing, four different cases can be distinguished. The possible cases are summarized in Table 6.2.

In *Case 1*, if the i–j sequence is optimal for the *first* type of interest calculation (interest is not compounded), then it is also optimal for the second type of interest calculation (interest is continuously compounded), and the optimal sequence is *not influenced* by the due dates.

In *Case 4*, if the i–j sequence is optimal for the *second* type of interest calculation (interest is continuously compounded), then it is also optimal for the first type of interest calculation (interest is not compounded), and the optimal sequence *is influenced* by the due dates.

In *Cases 2* and *3*, the optimality of a sequence for one type of interest calculation is not a sufficient condition of optimality for the other type of interest calculation. Furthermore, when the interest is continuously compounded the optimal sequence *is influenced* by the due dates.

In the next chapter it will be demonstrated, however that if due dates are not very different, then the ratio of exponential due dates hardly modifies the ratio of costs, consequently their effect can be ignored.

Table 6.2 Optimality conditions of the i–j sequence of jobs

	$d_i \le d_j$	$d_i > d_j$
$t_i \le t_j$	**Case 1** $\frac{e^{\frac{q}{365}d_i}}{e^{\frac{q}{365}d_j}} \cdot \frac{c_i}{c_j} \le \frac{c_i}{c_j} \le \frac{t_i}{t_j} \le \frac{f(t_i)}{f(t_j)}$	**Case 2** $\frac{c_i}{c_j} \le \frac{t_i}{t_j}$ and $\frac{e^{\frac{q}{365}d_i}}{e^{\frac{q}{365}d_j}} \cdot \frac{c_i}{c_j} \le \frac{f(t_i)}{f(t_j)}$
$t_i > t_j$	**Case 3** $\frac{c_i}{c_j} \le \frac{t_i}{t_j}$ and $\frac{e^{\frac{q}{365}d_i}}{e^{\frac{q}{365}d_j}} \cdot \frac{c_i}{c_j} \le \frac{f(t_i)}{f(t_j)}$	**Case 4** $\frac{c_i}{c_j} \le \frac{e^{\frac{q}{365}d_i}}{e^{\frac{q}{365}d_j}} \cdot \frac{c_i}{c_j} \le \frac{f(t_i)}{f(t_j)} \le \frac{t_i}{t_j}$

6.4 Illustration of the results with the help of a calendar manufacturing process

In the following, the effect of the derived sequencing rules is demonstrated with the help of the situation of the calendar manufacturer. The company buys sheets of printed calendar pages for several types of calendars from printing companies. The sheets are cut into pages, bound with the corresponding technology, packaged, and stored until the delivery date. The company prepares approximately 200 different calendars yearly, in lot sizes ranging between 50 and 15 000 pieces.

The bottleneck of the production process is the cutting machine. There are very sophisticated cutting machines on the market with short setup time and with high cutting accuracy. However, these types of machines are financially not feasible for the company because of the relatively small volume and range of production. A manual cutting machine is used with long (sequence independent) setup times and operated by a skilled worker. Identical calendars are cut in one batch to reduce setup time. The cutting machine has enough capacity to cut the yearly production requirement if the machine operates continuously. Therefore, one of the major *operational constraints* of scheduling is to provide work for the cutting machine continuously.

Calendars are perishable items which can only be sold around the end of the year. Those calendars which are not sold in this period can be considered waste. The majority of calendars are produced for orders. Orders are known in advance, and the ordered calendars are delivered in September to the customers. In this chapter I deal with production scheduling of those orders which are known in advance and all have to be shipped at a given date in September. The scheduling of random orders occurring after September is not topic of this research.

The objective of scheduling is to minimize inventory holding cost. Calendars are produced during 9 months but delivery and income occur only in September. The printed sheets are delivered for the required date and paid upon arrival. Therefore, the raw materials of calendars have to be financed from the beginning of production until the date of delivery in September. Raw materials (and production) are financed from credits.

The company currently schedules production randomly. Some raw material supply priorities and common sense considerations are applied but scheduling theory results are not used. The schedule of the production of 78 orders of calendars for an 89 day long production period from last year with all the relevant data (cutting operation times, raw material costs) were provided by the company to demonstrate inventory cost saving when the derived scheduling rules are applied.

Table 6.3 contains the inventory holding cost for the actual production and also for the schedules generated by the WLPT and WLTEPT rules in the case of both type of interest calculation. Optimal values in the table are indicated by bold face numbers.

The first column shows the result when interest is not compounded and the second column shows the result when interest is continuously compounded. These two columns provide a lower and an upper estimate for the real inventory holding cost. Since the difference is very small, the method of interest calculation does not seem to be relevant. This is true even if the interests for the different type of interest calculation methods differ slightly.

Applying the WLPT rule derived in Chapter 6.2.1 a considerable saving can be seen in Table 6.3. Cost saving is 1218.9 Euros (3864.3–2645.4) if production is performed in the sequence provided by the WLPT rule. This is 31.54% cost decrease which can be realized simply by changing the production sequence of jobs.

Table 6.3 also shows that the same inventory holding cost is obtained for the WLPT and for the WLTEPT rules. The detailed analysis of the sequences shows that the identical inventory holding costs belong to identical production sequences. In the case of the WLTEPT rule the optimal value of the inventory holding cost is 2666.2 Euros. The slight increase of cost compared to the WLPT optimum (2645.4 Euros) is the result of the different interest calculation methods and not the consequence of the change of production sequence.

Table 6.3 Inventory holding cost of different production schedules (Euros)

	Interest is not compounded (r=0.1092)	Interest is continuously compounded (q=0.1092)
Actual	3864.3	3900.6
WLPT	**2645.4**	2666.2
WLTEPT	2645.4	**2666.2**

The similar results for the two different scheduling rules are not surprising based on the conclusions of Chapter 6.2. The $f(t)$ function is almost linear at 10.92% interest rate in the range of operation times (1-15 days) relevant at the company (see Figure 6.4).

The range between the ratio of operation times and the ratio of transformed exponential operation times is very small (around 0.0002) in the relevant section of the $f(t)$ function. Therefore, there is only a very small possibility that the two optimal sequences will differ. Even if the sequences were different, applying one of the two optimums for both interest calculations would have no significant cost consequences.

If due dates of orders are different, then, according to Table 6.2, the ratio of exponential due dates has to be considered. Due dates, however, cannot be very different in practice. On the one hand customers do not need next year calendars very early; on the other hand customers cannot get calendars very late because the product cannot be sold long after the beginning of New Year. If the company applies different due dates, those due dates are certainly within a narrow range of the existing common due date. Applying the 10.92% interest rate used at the company and a 25% possible increase and decrease of the existing due date for some calendars, the smallest and highest possible values of the ratio of the exponential due dates are as follows,

$$\frac{e^{\frac{q}{365}d_i}}{e^{\frac{q}{365}d_j}} = \frac{e^{89\frac{0.1092}{365}}}{e^{89(1+0,25)\frac{0.1092}{365}}} = 0.993; \qquad \frac{e^{\frac{q}{365}d_i}}{e^{\frac{q}{365}d_j}} = \frac{e^{89\frac{0.1092}{365}}}{e^{89(1-0,25)\frac{0.1092}{365}}} = 1.007 \tag{6.29}$$

These results show that despite of the difference of the delivery dates optimal sequence of jobs is practically not influenced by the interest calculation method.

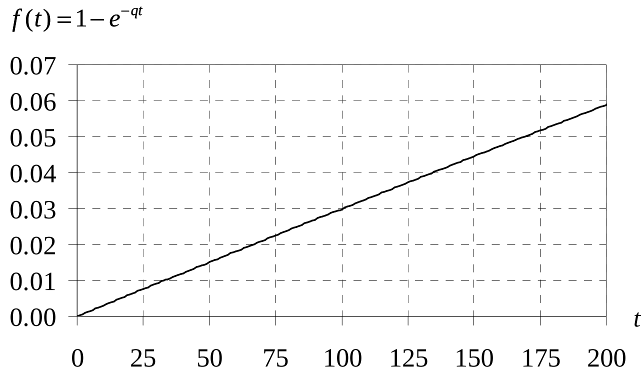

Figure 6.4 The f(t) function for the parameters of the calendar manufacturer (q=0.1092)

6.5 Conclusions of Chapter 6

In this chapter the effect of interest calculation methods on inventory holding cost and consequently on production scheduling decisions is analyzed.

First, two scheduling rules are derived; one for a linear and one for a non-linear earliness related objective function, when all due dates are identical. The WLPT rule – for the case when interest is not compounded – is independent of the interest rate. The WLTEPT rule – in case of continuous interest calculation – depends on the interest rate. However, it is proved in Chapter 6.2 and it is illustrated in Chapter 6.4 that in practice the optimal sequences provided by the WLPT and by the WLTEPT rules are very frequently identical. Even if the optimal sequences are different, applying any of the optimal sequences is financially acceptable.

Second, the problem is extended to the consideration of different due dates. It is demonstrated that in Case 1 (see Table 6.3) the conclusions made for common due dates are valid for different due dates as well. In all other cases (Cases 2, 3 and 4 in Table 6.3) the difference of due dates theoretically influences the optimal sequence but in practice the effect of this difference is insignificant.

Based on the insensitivity of the optimal sequences to the method of interest calculation, it can be concluded that in most cases the optimal sequence can be determined by using only the raw material costs (c_i) and the processing times (t_i). The method of interest calculation and the value of interest rate do not affect the optimal sequence. This conclusion does not mean that the *value* of inventory holding cost is not influenced by financial conditions but the optimal operation of the system is independent of these factors.

The results presented in this chapter assume that all orders are delivered for the required due dates. If due dates are not met and penalty cost for delay must be considered, the suggested API algorithm cannot be applied. In this case mathematical programming models and the cumbersome application of branch and bound algorithms may be appropriate.

As a summary, based on Chapter 6, the following scientific result can be formulated:

Result 5/1
I proved that in the case of a single resource scheduling problem

− if task times (t_i) are deterministic,
− the sequence independent setup time is part of the task time (t_i),
− there are no precedence constraints of tasks,
− each task must be performed for the *same due date,*
− inventory holding cost is calculated with *periodic interest calculation*,

then inventory holding cost is minimized if condition (6.7) is met by any two adjacent *i* and *j* jobs if $i<j$. This condition can be called weighted longest processing time (WLPT) rule.

If inventory holding cost is calculated with *continuous interest calculation* (compounded interest), then condition (6.15) must be met by any two adjacent *i* and *j* jobs if $i<j$. This condition can be called weighted longest transformed exponential processing time (WLTEPT) rule.

Result 5/2

I proved that in the case of a single resource scheduling problem

− if task times (t_i) are deterministic,
− the sequence independent setup time is part of the task time (t_i),
− there are no precedence constraints of tasks,
− each task has *different due date* (d_i),
− inventory holding cost is calculated with *periodic interest calculation*,

then inventory holding cost is minimized if condition (6.23) is met by any two adjacent *i* and *j* jobs if $i<j$. This condition is equivalent to the longest processing time (WLPT) rule. Based on formula (6.23) it can be concluded, that the due dates (d_i) *do not* influence the optimal schedule.

If inventory holding cost is calculated with *continuous interest calculation* (compounded interest), then condition (6.28) must be met by any two adjacent *i* and *j* jobs if $i<j$. This condition is equivalent to the weighted longest transformed exponential processing time (WLTEPT) rule, but in this case the due dates (d_i) *influence* the optimal schedule and are incorporated in the weights.

Result 5/3

I proved that in the case of a single resource scheduling problem

− if task times (t_i) are deterministic,
− the sequence independent setup time is part of the task time (t_i),
− there are no precedence constraints of tasks,

then the optimal schedule based on the lower estimate of inventory holding cost (calculated by periodic interest calculation), and the optimal schedule based on the upper estimate of inventory holding cost (calculated by continuous interest calculation) differs only in extreme situations in practice. That is, scheduling results of the presented problems are very insensitive to the inventory cost calculation methods.

The derivation of the scheduling rules which minimizes inventory holding cost in case of common due dates are published in Koltai (2006a, 2006b). The generalization of the result for different due dates are presented in Koltai (2009).

7 SUMMARY OF THE DISSERTATION

This dissertation summarizes my main research results related to sensitivity analysis in the area of production planning and scheduling in the past 20 years. Production planning and scheduling is a very diversified area. Depending on the type of production processes (project based, small batch production, mass production), depending on the demand characteristics (hectic, stable) and depending on the implemented production planning and control systems, there are several ways of planning and scheduling production. Frequently, in practice, only a simple spreadsheet model is used to get a feasible plan. Sometimes, complex simulation models are built to analyze operation possibilities. Occasionally, large mathematical programming models are solved to get optimal solution. No matter, however, how a production plan and production schedule is generated, the change of some planning parameters or planning conditions is expected. If some planning parameters change, the analysis of the effect of these changes on the production plan or on the production schedule must be analyzed and evaluated. Consequently, sensitivity analysis provides important information for production related decision-making.

Since the formulations and solution techniques of production planning and scheduling problems are very diverse, it is not possible to develop a general theory of sensitivity analysis. In case of each problem presented in this work, special methods must have been developed, based on the characteristics of the model applied for the generation of the plan.

The dissertation contains five different methods for sensitivity analysis:

− In Chapter 2 (Results 1/1 and 1/2), sensitivity analysis of a linear production planning model is used to illustrate some problems related to degenerate optimal solutions.

− In Chapter 3 (Results 2/1, 2/2 and 2/3), a new aggregation approach is developed for capacity analysis of FMSs, and a special sensitivity analysis based on the characteristics of this new approach is developed.

− In Chapter 4 (Results 3/1 and 3/2), integer mathematical programming models are applied, and the characteristics of a pricewise linear efficiency function is analyzed.

− In Chapter 5 (Result 4), a scheduling problem is modeled as discrete a event dynamic system and perturbation analysis is implemented to get sensitivity results.

− Finally, in Chapter 6 (Results 5/1, 5/2 and 5/3), a scheduling problem is approached as a combinatorial problem, and the special characteristics of the investigated case lead to sensitivity conclusions.

Each presented problem is different, consequently, the applied techniques for sensitivity analysis are different as well, but the objective is the same in all cases: *the minimal change of a planning parameter which requires the modification of the plan has to be determined*.

The presented scientific results have different scientific and practical relevance. The results of Chapters 2 and 3 have mostly theoretical significance.

− The analysis of citations shows that, the proposed new classification of sensitivity information (Chapter 2) motivated several researchers to calculate these sensitivities in different special LP models (see for example Lin and Wen, 2003; Kavitha and Pandian, 2012 or Ma, Lin and Wen, 2013). Furthermore, the better understanding of the traps of misleading information, and the proposed tool for getting the appropriate sensitivity values, may improve the decision-making process in the related areas (Arsham, 2012).

− The proposed aggregation method based on the concept of operation types (Chapter 3) may contribute to a the better understanding of the available capacity of flexible manufacturing systems (see for example Matta, Tomasella and Valente, 2007 or Zaeh and Mueller, 2007).

Besides the theoretical significance, the results of Chapters 4, 5 and 6 have direct practical impacts as well.

− The extension of ALB models with workforce skill constraints (Chapter 4) provides information to line configuration decisions in case of a bicycle assembly process (Koltai, Tatay and Kalló, 2013).

− The application of perturbation analysis for the examination of the sensitivity of production schedules (Chapter 5) facilitates the control of waiting time in a technologically critical point of a continuous steel casting process (Koltai, Larraneta and Onieva, 1993).

− The proposed new scheduling rules (Chapter 6) can help to improve the financial performance of a simple calendar manufacturing process (Koltai, 2006).

Finally, it must be noted that, although, the objective of this dissertation is sensitivity analysis related to *production systems*, sensitivity information are very important in *service systems* as well. For example, Koltai, Kalló and Lakatos (2009) examined, how the arrival characteristics of customers and the main parameters of the purchasing process influence waiting time in front of the check-out counters in a supermarket. In general, it can be concluded that what is true for production systems, it is also true for service systems: *Some major planning parameters may change, and the effect of these changes on some performance measures must be analyzed and evaluated with special methods, developed for the problem in question.*

As a summary, it can be concluded that in production systems, some major parameters used for planning may change for several reasons. In these cases, information about the effect of the change on some performance measures is important information for the decision maker. Today, company excellence depends more and more on the efficient collection and processing of a large amount of data related to the production process and to the production environment, in order to support decision-making. In this environment, sensitivity analysis related research is highly relevant and its frontiers are constantly extending. The presented results are going to provide some modest contribution to this area.

8 LIST OF PUBLICATIONS RELATED TO SCIENTIFIC RESULTS

8.1 List of pear reviewed journal papers related to scientific results

Guerrero, F., Lozano, S., Koltai, T. and Larraneta, J., 1999. Machine loading and part type selection in flexible manufacturing systems. *International Journal of Production Research*, 37(6), pp.1303-1317. (IF=0,512)

Koltai, T., 1995. Fixed cost oriented bottleneck analysis with linear programming. *Omega: International Journal of Management Science*, 23(1), pp.89-95. (IF: 0,286)

Koltai, T., 2009. Robustness of a production schedule to inventory cost calculations. *International Journal of Production Economics*, 121(2), pp.494-504. (IF=2,07)

Koltai. T., Kalló, N. and Lakatos, L., 2009. Optimization of express line performance: Numerical examination and management considerations. *Optimization and Engineering*, 10(3), pp.377-396. (IF=1,00)

Koltai, T., Larraneta, J. and Onieva, L., 1993. Examination of the sensitivity of an operation schedule with perturbation analysis. *International Journal of Production Research*, 31(12), pp.2777-2787. (IF=0,314)

Koltai, T., Larraneta, J. and Onieva, L., 1994. An operations management approach to perturbation analysis. *JORBEL, Belgian Journal of Operations Research Statistics and Computer Science*, 33(4), pp.17-41.

Koltai, T., Larraneta, H., Onieva, L. and Lozano, S., 1994. Sensitivity examination of the simulation result of discrete event dynamic systems with perturbation analysis. *Questió: Quaderns D Estadística Sistemes Informática Investigació Operativa*,18(2), pp.209-228.

Koltai, T. and Lozano, S., 1996. The illustration of the routing sensitivity calculation of flexible manufacturing systems with perturbation analysis. *Periodica Polytechnica-Social and Management Sciences*, 4(1), pp.5-28.

Koltai, T. and Lozano, S., 1998. Sensitivity calculation of the throughput of an FMS with respect to the routing mix using perturbation analysis. *European Journal of Operational Research*, 105, pp.483-493. (IF=0,413)

Koltai, T., Lozano, S., Guerrero, F. and Onieva, L., 2000. A flexible costing system for flexible manufacturing systems using activity based costing. *International Journal of Production Research*, 38(7) pp.1615-1630. (IF=0,504)

Koltai T., Romhányi G. és Tatay V., 2009. Optimalizálás bizonytalan paraméterekkel a termelés- és szolgáltatásmenedzsmentben (Optimization with uncertain parameters in production and operations management). *Vezetéstudomány*, 40 (különszám), pp.68-73.

Koltai, T. and Stecke, K.E., 2008. Route-independent analysis of available capacity in flexible manufacturing systems. *Production and Operations Management*, 17(2), pp.211-223. (IF=1,933)

Koltai, T. and Tatay, V., 2011. A practical approach to sensitivity analysis in linear programming under degeneracy for management decision making. *International Journal of Production Economics*, 131(1), pp.392-398. (IF=1,76)

Koltai, T. and Tatay, V., 2011. Formulation of simple workforce skill constraints in assembly line balancing models. *Periodica Polytechnica-Social and Management Sciences*, 19(1), pp.43-50.

Koltai, T. and Tatay. V., 2013. Formulation of workforce skill constraints in assembly line balancing models. *Optimization and Engineering*, 14, pp.529-545. (IF=0,955*)

Koltai, T., Tatay, V. and Kalló, N., 2014. Application of the results of simple assembly line balancing models in practice: The case of a bicycle manufacturer. *International Journal of Computer Integrated Manufacturing*, 27(9), pp.887-898. (IF=1,019*)

Koltai, T. and Terlaky, T., 2000. The difference between the managerial and mathematical interpretation of sensitivity results in linear programming. *International Journal of Production Economics*, 65(3), pp.257-274. (IF=0,258)

8.2 List of published conference papers related to the scientific results

Györkös, R., Koltai, T. and Kalló, N., 2014. Empirical analysis of the significance of learning effect and task assignment on assembly line performance. In: *microCAD 2014: XXVIII. microCAD International Scientific Conference: Economic Challenges in the 21st Century*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.1-6. (CD-ROM)

Juhász, V. and Koltai, T., 2003. Some practical issues of the capacity analysis of FMS based on the concept of operation types. In: L. Lehoczky and L. Kalmár, eds. *microCAD 2003, International Computer Science Conference: Production engineering, manufacturing systems*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.89-94.

Koltai, T., 1992. Sensitivity analysis of discrete event dynamic systems. In: R. Trappel, ed. *Cybernetics and systems research'92 Vol. 1*. Singapore: World Scientific. pp.145-151.

Koltai, T., 2006a. Robustness of a production schedule to the method of cost of capital calculation. In: R.W. Grubbström and H.H. Hinterhuber, eds. *14th International Working Seminar on Production Economics: Pre-Prints Volume 1*. Innsbruck, Ausztria, pp.207-216.

Koltai, T., 2006b. Economic analysis of production scheduling of a calendar manufacturing process. In: L. Huw, B. Gaughran, S. Burke, W.G. Sullivan and A. Munir, eds. *Proceedings of the 16th International Conference on Flexible Automation and Intelligent Manufacturing*. Limerick, Ireland: University of Limerick, pp.617-624.

Koltai, T., 2012. Supporting line configuration decisions with assembly line balancing models: A practical case. In: E. Ilie-Zudor, Zs. Kemény and L. Monostori, eds. *Proceedings of the 14th International Conference on Modern Information Technology in the Innovation Processes of the Industrial Enterprises*. Budapest, Hungary: HAS Computer and Automation Research Institute, pp.68-79.

Koltai, T., 2013. Formulation of multi-level workforce skill constraints in assembly line balancing models. In: *Preprints of the IFAC Conference on Manufacturing Modelling, Management, and Control*. Saint-Petersburg, Russia: IFAC by Pergamon Press, pp.802-807.

Koltai, T., Farkas, A. and Stecke, K.E., 2001. An aggregate capacity analysis model for a flexible manufacturing environment. In: *Proceedings of the 2000 Japan-USA Flexible Automation Conference*. Ann Arbor, USA: ASME, pp.1381-1388.

Koltai, T. and Györkös, R., 2012. Comparison of the optimal performance of assembly line configurations with simple assembly line balancing models. In: P. Bikfalvi, ed. *XXVI. microCAD International Scientific Conference: Economic Challenges in the 21st Century*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.1-6. (CD-ROM)

Koltai, T. and Györkös, R., 2013. Analysis of the efficiency of task assignment in the presence of learning effect. In: *microCAD 2013: XXVII. International Scientific Conference: Economic Challenges in the 21st Century*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.1-6. (CD-ROM)

Koltai, T., Györkös, R. and Kalló, N., 2014. Analysis of the bottleneck of simple assembly lines with learning effect. In: R.W. Grubbström and H.H. Hinterhuber, eds. *18th International Working Seminar on Production Economics: Pre-prints Volume 3.* Innsbruck, Ausztria, pp.289-300.

Koltai, T., Juhász, V. and Stecke, K.E., 2004. A new formulation of capacity constraints in the production planning of flexible manufacturing systems. In: L. Wang, J. Xi, W.G. Sullivan, A. Munir, eds. *Proceedings of the 14th International Conference of Flexible Automation and Intelligent Manufacturing*. Toronto, Kanada, pp.775-782.

Koltai, T., Juhász, V., Stecke, K.E. and Varlaki, P., 2004. A new approach for the production planning of flexible manufacturing systems based on the concept of operation types. In: S. Gupta, ed. *Proceedings of the 2nd World Congress on Production and Operations Management*. Cancun, Mexiko, pp.1-25. (CD-ROM)

Koltai, T., Stecke, K. and Juhász, V., 2004. Planning of flexibility of flexible manufacturing systems. In: *Proceedings of the 2004 Japan-USA Symposium on Flexible Automation*. New York, USA: ASME, pp.1-8. (CD-ROM)

Koltai, T. and Tatay, V., 2008a. Support of production management decisions by sensitivity analysis of linear production planning models. In: T. van der Vaart, D.P. van Donk, W. van Wezel, G. Welker, J. Wijngaard, eds. 2008. *15th International Annual EurOMA Conference.* Groningen, The Netherlands, pp.1-10. (CD-ROM)

Koltai, T. and Tatay, V., 2008b. The effect of degenerate LP sensitivity analysis results on management decision making. In: L. Lehoczky, ed. *microCAD 2008 International Scientific Conference: Company competitiveness in the XXI century*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre*, pp.27-32.*

Koltai, T. and Tatay, V., 2010. Application of simple assembly line balancing models to support production quantity related decisions. In: R.W. Grubbström and H.H. Hinterhuber, eds. *16th International Working Seminar on Production Economics: Preprints Volume 1*. Innsbruck, Ausztria, pp.285-296.

Koltai, T., Tatay, V. and Kalló, N., 2011. Application of simple assembly line balancing models to support quick response operation in a bicycle production process. In: *Proceedings of the 3rd Rapid Modelling Conference: Rapid Modelling for Sustainability*. Leuven, Belgium: Katholieke Universiteit Leuven, pp.1-10. (CD-ROM)

Koltai, T. and Terlaky, T., 1999. Sensitivity analysis problems of linear programming in management decision making. In: L. Lehoczky and L. Kalmár, eds. *microCAD '99 International Computer Science Conference*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.83-88.

Tatay, V. and Koltai, T., 2010. Solving assembly line balancing models in Excel environment to support production management decisions. In: P. Bikfalvy Péter, ed. *XXIV. microCAD International Scientific Conference: Economic Challenges in the 21st Century*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.165-170.

Tatay, V. and Koltai, T., 2011. Supporting production management decisions with assembly line balancing models in the presence of skilled and unskilled workers. In: P. Bikfalvi, ed. *XXV. microCAD International Scientific Conference: Economic Challenges in the 21st Century*. Miskolc, Hungary: University of Miskolc Innovation and Technology Transfer Centre, pp.125-130.

8.3 List of other publications related to the scientific results

Farkas, A., Koltai, T. and Stecke, K.E., 1999. *Workload balancing using the concept of* *operation types*. Working Paper. Michigan, USA: University of Michigan Business School, 31p.

Koltai, T., 1994. Performance evaluation of production and service systems: Basic concepts, applications and future trends. In: S. Péter and A. Farkas, eds. *Business Research and Management Challenges. Papers and Cases from Central/Eastern Europe: IMC's Fifth Anniversary Yearbook*. Budapest, Hungary: International Management Center, pp. 95-121.

Koltai T., 2006. *Termelésmenedzsment (Production Management)*. Budapest: Typotex Kiadó, 279p.

Koltai, T., Farkas, A. and Stecke, K.E., 1998. *Aggregate production planning of flexible manufacturing systems using the concept of operation types*. Working Paper, 98/007. Michigan, USA: University of Michigan Business School, 29p.

9 LITERATURE

Akgül, M., 1984. A note on shadow prices in linear programming. *Journal of the Operational Research Society*, 35, pp.425-431.

Alidaee B., 1993. Numerical method for single machine scheduling with non-linear cost functions to minimize total cost. *Journal of the Operations Research Society*, 44(2), pp.125-132.

Anderson, E.J., 1994. *The management of manufacturing*. *Models and analysis*. Addison-Wesley.

Anderson, D.R., Sweeney, D.J. and Williams, T.A., 1994. *An introduction to management science*. Minneapolis:West Publishing Company.

Arbib, C., Lucertini, M. and Nicolo, F., 1990. Workload balance and part-transfer minimization in flexible manufacturing systems. *The International Journal of Flexible Manufacturing Systems*, 3, pp.5-25.

Arıkan, M. and Erol, S., 2012. A hybrid simulated annealing-tabu search algorithm for the part selection and machine loading problems in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 59(5-8), pp.669-679.

Arsham, H., 2012. Foundation of linear programming: A managerial perspective from solving system of inequalities to software implementation. *International Journal of Strategic Decision Science*, 3(3), pp.40-60.

Atan, T.S. and Pandit, R., 1996. Auxiliary tool allocation in flexible manufacturing systems. *European Journal of Operational Research*, 89, pp.642-659.

Aucamp, D.C. and Steinberg, D.I., 1982. The computation of shadow prices in linear programming. *Journal of the Operational Research Society*. 33, pp.557-565.

Baker, R.K., 1974. *Introduction to sequencing and scheduling*. John Wiley and Sons.

Baker, R.K. and Scudder, G.D., 1990. Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38(1), pp.22-36.

Basnet, C., 1996. Technical note: Tabu search heuristic for a loading problem in flexible manufacturing systems. *International Journal of Production Research*, 34(4), pp.1171-1174.

Bastos, J.M., 1988. Batching and routing: Two functions in the operational planning of flexible manufacturing systems. *European Journal of Operational Research*, 33, pp.230-244.

Battaïa, O. and Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, 142(2), pp.259–277.

Baybars, Í., 1986. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32, pp.909-932.

Becker, C. and Scholl A., 2006. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168, pp.694-715.

Bector, C.R., Gupta, Y.P. and Gupta, M.C., 1991. Optimal scheduling of jobs about a common due date on a single machine. *International Journal of Systems Science*, 22(12), pp.2541-2552.

Berkelaar, A.B., Roos, C. and Terlaky, T., 1997. The optimal set and optimal partition approach. In: H.J. Greenberg and T. Gal, eds. 1997. *Advances in sensitivity analysis and parametric programming*. Dorderecht: Kluwer Academic Publishers.

Bernardo, J.J. and Mohamed, Z., 1992. The measurement and use of operational flexibility in the loading of Flexible Manufacturing Systems. *European Journal of Operational Research*, 60, pp.144-155.

Berrada, M. and Stecke, K.E., 1986. A branch and bound approach for machine load balancing in flexible manufacturing systems. *Management Science*, 32(10), pp.1316-1335.

Bertrand, J.W.M. and Wortmann, J.C., 1981. *Production control and information systems for component – manufacturing shops*. Amsterdam: Elsevier Science Publishers B.V.

Borgonovo E. and Peccati L., 2004. Sensitivity analysis in investment project evaluation. *International Journal of Production Economics*. 90(1), pp.17-25.

Borgonovo E. and Peccati L., 2006. Uncertainty and global sensitivity analysis in the evaluation of investment projects. *International Journal of Production Economics*. 104(1), pp.62-73.

Boysen N., Fliedner M. and Scholl A., 2008. Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111, pp.509-528.

Bowman E.H., 1960. Assembly line balancing by linear programming. *Operations Research,* 8, pp.385-389.

Bryton, B., 1954. Balancing of a continuous production line. *Unpublished M.S. Thesis,* Evanston, ILL: Northwestern University.

Caine, D.J. and Parker, B.J., 1996. Linear programming comes of age: A decision support tool for every manager. *Management Decision*, 34(4), pp.46-53.

Cao, X.R., 1988. Realization probability in multi-class closed queuing networks. *European Journal of Operations Research*, 36, pp.393-401.

Chan, K.C.C., Hui, P.C.L., Yeung, K.W.Ng. and Frency, S.F., 1998. Handling the assembly line balancing problem in the clothing industry using a genetic algorithm. *International Journal of Clothing Science and Technology,* 10(1), pp.21-37.

Chakravarty, A.K. and Shtub, A., 1984. Selecting parts and loading flexible manufacturing systems. In: *Proceedings of the first ORSA/TIMS special conference on flexible manufacturing systems*, Ann Arbor, Michigan, pp.284-289.

Chen, I.J. and Chung, C-H., 1991. Effects of loading and routing decisions on performance of flexible manufacturing systems. *International Journal of Production Research*, 29(11), pp.2209-2225.

Chen, I.J. and Chung, C-S., 1996. Sequential modeling of the planning and scheduling problems of flexible manufacturing systems. *Journal of the Operational Research Society*, 47, 1216-1227.

Cohen, Y. and Dar-El, E., 1998. Optimizing the number of stations in assembly lines under learning for limited production. *Production Planning and Control*, 9(3), pp.230-240.

Cohen, Y., Vitner, G. and Sarin S.C., 2006. Optimal allocation of work in assembly lines for lots with homogenous learning. *European Journal of Operational Research*, 168(3), pp.922–931.

Conway, R.W., Maxwell, W.L. and Miller, W.L., 1976. *Theory of scheduling*. John Wiley and Sons.

Corominas A., Pastor F. and Plans, J., 2008. Balancing assembly line with skilled and unskilled workers. *Omega,* 36, pp.1126-1132.

Cortes, P., Onieva, L. and Guadix, J., 2010. Optimizing and simulating the assembly line balancing problem in a motorcycle manufacturing company: A case study. *International Journal of Production Research*, 48(12), pp.3637-3656.

Costa, A.M. and Miralles, C., 2009. Job rotation in assembly lines employing disabled workers. *International Journal of Production Economics*, 120(2), pp.625-632.

D′Alfonso, T.H. and Ventura, J.A., 1995. Assignment of tools to machines in a flexible manufacturing system. *European Journal of Operational Research*, 81, pp.115-133.

Davenport, T.H., 2006. Competing on analytics. *Harvard Business Review*, January, pp.99-107.

Davenport, T.H., 2013. Analytics 3.0. *Harvard Business Review*, December, pp.64-72.

Davenport, T.H. and Harris, J.G., 2007. *Competing on analytics. The new science of winning*. Boston:Harvard Business School Press.

De Werra, D. and Widmer, M., 1990. Loading problems with tool management in flexible manufacturing systems: A few integer programming models. *The International Journal of Flexible Manufacturing Systems*, 3, pp.71-82.

Díaz, A., Sancho, L., García R. and Larrańeta, J., 1991. A dynamic scheduling and control system in an ENSIDESA steel plant. *Interfaces*, 21, pp.53-62.

Emmons, H., Flowers, A.D., Khot, C.M. and Mathur K., 2001. *STORM 4.0. Quantitative Modeling for Decision Support*. Euclid, OH: Lakeshore Communications.

Emrani N.M., Bahalke, U., Dolatkhahi, K., Dolatkhahi, S. and Makui, A., 2011. Simple assembly line balancing problem under task deterioration. *International Journal of Industrial Engineering Computations*, 2(3), pp.583-592.

Eschenbach, T.G., 1992. Spiderplots versus tornado diagrams for sensitivity analysis. *Interfaces*, 22(6), pp.40-46.

Evans, J.R. and Baker, N.R., 1982. Degeneracy and the (mis)interpretation of sensitivity analysis in linear programming. *Decision Science*, 13, pp.348-354.

Gal, T., 1979. *Postoptimal analysis, parametric programming and related topics.* New York: McGraw-Hill.

Gal, T., 1986. Shadow prices and sensitivity analysis in linear programming under degeneracy. *OR Spektrum*, 8, pp.59-71.

Glasserman, P., 1991. *Gradient estimation via perturbation analysis*. Boston: Kluwer Academic Publisher.

Graves, S.C., Rinnoy Kan, A.H.G. and Zipkin, P.H., 1993. *Logistics of production and inventory.* Amsterdam: North-Holland.

Greenberg, H.J., 1986. An analysis of degeneracy. *Naval Research Logistics Quarterly*, 33, pp.635-655.

Greene, T.J. and Sadowski, R.P., 1986. A mixed integer program for loading and scheduling multiple flexible manufacturing cells. *European Journal of Operational Research*, 24, pp.379-386.

Guerrero, F., Lozano, S., Koltai, T. and Larraneta, J., 1999. Machine loading and part type selection in flexible manufacturing systems. *International Journal of Production Research*, 37(6), pp.1303-1317.

Hall, N.G. and Posner, M.E., 2004. Sensitivity analysis for scheduling problems. *Journal of Scheduling*, 7(1), pp.49-83.

Harris, F.W., 1913. How many parts to make at once? *Factory, The Magazine of Management*, 10(2), pp.135-136, p.152 (Reprinted in 1990 in *Operations Research*, 38(6), pp.947-950).

Hax, A.C. and Candea, C., 1984. *Production and inventory management*. Englewood Cliffs: Prentice-Hall.

Heidelberg, P., Cao, X.R., Zazanis, M.A. and Suri, R., 1988. Convergence properties of infinitesimal perturbation analysis estimates. *Management Science*, 34, pp.1281-1302.

Hicks, P.E., 1977. *Introduction to industrial engineering and management science*. McGrew-Hill Company.

Higle, J.L. and Wallace, S.W., 2003. Sensitivity analysis and uncertainty in linear programming. *Interfaces*, 33(4), pp.53-60.

Hillier, S.F. and Lieberman, G.J., 1995. *Introduction to operations research*. McGrew-Hill International Editions.

Ho, Y.C. and Cassandras, C., 1983. A new approach to the analysis of discrete event dynamic systems. *Automatica*, 19, pp.149-167.

Ho, Y.C., 1987. Performance evaluation and perturbation analysis of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, AC-32, pp.563-572.

Ho, Y. C., Cao, X.R. and Cassandras, C., 1983. Infinitesimal and finite perturbation analysis for queuing networks. *Automatica*, 19, 439-445.

Ho, Y.C. and Cao, X.R., 1991. *Perturbation analysis of discrete event dynamic systems*. Kluwer Academic Publisher.

Ho, Y.C., Eyler, M.A. and Chien, T.T., 1983. A new approach to determine parameter sensitivity of transfer lines. *Management Science*, 29, pp.700-714.

Ho, Y.C. and Li, S., 1988. Extension of infinitesimal perturbation analysis. *IEEE Transaction on Automatic Control*, AC-33, pp.427-438.

Ho, Y.C., Suri, R., Cao, X.R., Diehl, G.W., Dille, L.W. and Zazanis, M., 1984. Optimization of large multi-class (non-product-form) queuing networks using perturbation analysis. *Large Scale Systems*, 7, pp.165-180.

Holt, C.C., Modigliani, F., Muth, J.F. and Simon, H.A., 1960. *Planning production, inventories, and work force*. Englewood Cliffs: Prentice-Hall.

Hwang, S.S. and Shogan, A.W., 1989. Modeling and solving an FMS part selection problem. *International Journal of Production Research*, 27(8), pp.1349-1366.

Jackson, J., 2002. Data mining: A conceptual overview. *Communications of the Association for Information Systems*, 8, pp.267-296.

Jansen, B., de Jong, J.J., Roos, C. and Terlaky, T., 1997. Sensitivity analysis in linear programming: Just be careful! *European Journal of Operational Research*, 101, pp.15-28.

Johnson, R.V., 1983. A branch and bound algorithm for assembly line balancing problems with formulation irregularities. *Management Science*, 29, pp.1309-1324.

Johnson, T.H., 1991. Activity-based management: Past, present, and future. *The Engineering Economist*, 36, pp.219-238.

Johnson, L.A. and Montgomery, D.C., 1974. *Operations research in production planning, scheduling and inventory control*. New York: John Wiley and Sons.

Kavitha, K. and Pandian, P., 2012. Type II sensitivity analysis in solid assignment problems. *Modern Applied Science*, 6(12), pp.22-26.

Kim, Y-D. and Yano, C.A., 1993. Heuristic approaches for loading problems in flexible manufacturing systems. *IIE Transactions*, 25(1), pp.26-39.

Kim, Y-D. and Yano, C.A., 1994. A new branch and bound algorithm for loading problems in flexible manufacturing systems. *The International Journal of Flexible Manufacturing Systems*, 6, pp.361-382.

Kirkavak, N. and Dinçer, C., 1993. Analytical loading models in flexible manufacturing systems. *European Journal of Operational Research*, 71, pp.17-31.

Kobayashi, H. and Gerla, M., 1983. Optimal routing in closed queuing networks. *ACM Transactions on Computer Systems*, 4(4), pp.294-310.

Koltai T., 2007. A kvantitatív eszközökkel támogatott versenyzés lehetőségei és korlátai. *CEO Magazin*, 8(4), pp.1-8.

Kövesi, J. ed., 2011. *Minőség és megbízhatóság a menedzsmentben (Quality and reliability in management)*. Budapest: Typotex Kiadó.

Kouvelis, P. and Lee, C., 1991. Block angular structures and the loading problem in flexible manufacturing systems. *Operations Research*, 39(4), pp.666-676.

Lapierre, S.D. and Ruiz, A.B., 2004. Balancing assembly lines: An industrial case study. *Journal of the Operational Research Society*, 55(6), pp.589-597.

Lashkari, R.S., Boparai, R. and Paulo, J., 2004. Towards an integrated model of operation allocation and material handling selection in cellular manufacturing systems. *International Journal of Production Economics*, 87(2), pp.115-139.

Lashkari, R.S., Dutta, S.P. and Padhye, A.M., 1987. A new formulation of operation allocation problem in flexible manufacturing systems: mathematical modeling and computational experience. *International Journal of Production Research*, 25(9), pp.1267-1283.

Liang, M. and Dutta, S.P., 1990. A mixed-integer programming approach to the machine loading and process planning problem in a process layout environment. *International Journal of Production Research*, 28, pp.1471-1484

Liang, M. and Dutta, S.P., 1993. An integrated approach to the part selection and machine loading problem in a class of flexible manufacturing systems. *European Journal of Operational Research*, 67, pp.387-404.

Lin, C.J. and Wen, U.P., 2003. Sensitivity analysis of the optimal assignment. *European Journal of Operational Research*, 149(1), pp.35-46.

Little, J.D.C., 1970. Models and managers: The concept of a decision calculus. *Management Sciences*, 16(8), pp.B466-B485.

Ma, K.T., Lin, C.J. and Wen, U.P., 2013. Type II sensitivity analysis of cost coefficients in the degenerate transportation problem. *European Journal of Operational Research*, 227(2). pp.293-300.

Matta, A., Tomasella, M. and Valente, A., 2007. Impact of ramp-up on the optimal capacity-related reconfiguration policy. *International Journal of Flexible Manufacturing Systems*, 19(3), pp.173-194.

McKay, K.N., Safayeni, F.R. and Buzacott, J.A., 1988. Job-shop scheduling theory: What is relevant? *Interfaces*, 18(4), pp.84-90.

Miralles, C., Garcia-Sabater, J.P., Andres, C. and Carlos M., 2007. Advantages of assembly lines in sheltered work centres for disabled. A case study. *International Journal of Production Economics*, 110(1-2), pp.187-197.

Mohamed, Z.M., 1996. A flexible approach to (re)configure flexible manufacturing cells. *European Journal of Operational Research*, 95, pp.566-576.

Monostori, L., Erdős, G., Kádár, B., Kis, T., Kovács, A., Pfeiffer, A. and Váncza, J., 2010.

Digital enterprise solution for integrated production planning and control. *Computer sin Industry*, 61, pp.112-126.

Moon, I., Logendran, R. and Lee, J., 2009. Integrated assembly line balancing with resource restrictions. *International Journal of Production Research*, 47(19), pp.5525-5541.

Moreno, A.A. and Ding, F., 1993. Heuristics for the FMS-loading and part-type-selection problems. *The International Journal of Flexible Manufacturing Systems*, 5, pp.287-300.

Nagarjuna, N., Mahes, O. and Rajagopal, K., 2006. A heuristic based on multi-stage programming approach for machine-loading problem in a flexible manufacturing system. *Robotics and Computer Integrated Manufacturing*, 22(4), pp.342-352.

Nahmias, S., 1993. *Production and operations analysis*. Boston: Richard D. Irwin, Inc.

Nam, S. and Logendran, R., 1992. Aggregate production planning – A survey of models and methodologies. *European Journal of Operational Research*, 61, pp.255-272.

Nanvala, H. and Awari, G., K., 2011. Approaches for solving machine loading problem in FMS: A review. *International Journal of Engineering and Technology*, 3(2), pp.64-73.

Niess, P.S., 1980. Kapacität Abgleich bei Flexiblen Fertigungssytemen. *IPA Forschung und Praxis*. Berlin: Springer Verlag.

Pastor, R. and Ferrer, L., 2009. An improved mathematical program to solve the simple assembly line balancing problem. *International Journal of Production Research*, 47(11), pp.2943-2959.

Patterson, J.H. and Albracht J.J., 1975. Assembly-line balancing: Zero-one programming with Fibonacci search. *Operations Research*, 23, pp.166-174.

Pedgen, D.C., Shannon, R.E. and Sadowski, R.P., 1990. *Introduction to simulation using SIMAN*. New York: McGraw-Hill.

Pfeiffer A., Kádár, B., Monostori, L. and Karnok, D. 2008. Simulation as one of the core technologies for digital enterprise: assessment of hybrid rescheduling methods. *International Journal of Computer Integrated Manufacturing*, 21(2), pp.206-214.

Prékopa A., 1968. *Az Operációkutatás Matematikai Módzserei*. Budapest: Bólyai János Matematikai Társulat Kiadványa

Ragsdale, C.T., 2007. *Managerial Decision Modeling*. Thomson South-Western.

Ram, B., Sarin, S. and Chen, C.S., 1990. A model and a solution approach for the machine loading and tool allocation problem in a flexible manufacturing system. *International Journal of Production Research*, 28(4), pp.637-645.

Rinnooy Kan, A.H.G., Lageweg, B.J. and Lenstra, K.L., 1975. Minimizing total costs in one-machine scheduling. *Operations Research*, 23(5), pp.908-927.

Roos, C., Terlaky, T. and Vial, J.-Ph., 1997. *Theory and algorithms for linear optimization - an interior point approach*. Chichester: John Wiley and Sons.

Rubin, D.S. and Wagner, H.M., 1990. Shadow prices: Tips and traps for managers and instructors. *Interfaces*, 20(4), pp.150-157.

Rubinstein, R.Y., 1986. *Monte Carlo optimization, simulation and sensitivity of queuing networks*. New York: John Wiley & Sons.

Saltelli., A, Tarantola, S. and Campolongo, F., 2000. Sensitivity analysis as an ingredient of modeling. *Statistical Science*, 15(4), pp.377-395.

Salvendy, G. ed., 1992. *Handbook of industrial engineering*. Chichester:John Wiley and Sons Inc.

Salveson, M.E., 1955. The assembly line balancing problem. *Journal of Industrial Engineering*, 6, pp.18-25.

Sarin, S.C. and Chen, C.S., 1987. The machine loading and tool allocation problem in a flexible manufacturing system. *International Journal of Production Research*, 25(7), pp.1081-1094.

Sawik, T., 1990. Modeling and scheduling of a flexible manufacturing system. *European Journal of Operational Research*, 45, pp.177-190.

Sawik, T., 2002. Balancing and scheduling of surface mount technology lines. *International Journal of Production Research*, 40(9), pp.1973-1991.

Scholl, A. and Becker, C., 2006. State-of-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168, pp.666-693.

Schrage, L., 2006. *Optimization modeling with Lingo*. Chicago: Lindo Systems, Inc.

Shanker, K. and Rajamarthandan, S., 1989. Loading problem in FMS: Part movement minimization, In: K.E. Stecke and R. Suri, eds. 1989. *Proceedings of the third ORSA/TIMS conference on FMS: Operations research models and applications,* Elsevier, pp.99-104.

Shanker, K. and Srinivasulu, A., 1989. Some solution methodologies for loading problems in a flexible manufacturing system. *International Journal of Production Research*, 27(6), pp.1019-1034.

Sherali, H.D., Sarin, S.C. and Desai, R., 1990. Models and algorithms for job selection, routing, and scheduling in a flexible manufacturing system. *Annals of Operations Research*, 26, pp.433-453.

Shanker, K. and Tzen, Y.J., 1985. A loading and dispatching problem in a random flexible manufacturing system. *International Journal of Production Research*, 23(3), pp.579-595.

Singhal, J. and Singhal, K., 2007. Holt, Modigliani, Muth, and Simon’s work and its role in renaissance and evolution of operations kanagement. *Journal of Operations Management*, 25(2), pp.300-309.

Smith, W.E., 1956. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3, pp.59-66.

Sodhi, M.S., Agnetis, A. and Askin, R.G., 1994. Tool addition strategies for flexible manufacturing systems. *The International Journal of Flexible Manufacturing Systems*, 6, pp.287-310.

Sodhi, M.S., Askin, R.G. and Sen, S., 1994. A hierarchical model for control of flexible manufacturing systems. *Journal of the Operational Research Society*, 45(10), pp.1185-1196.

Solomon, M.M., Millen, R.A. and Afentakis, P., 1995. The formation of subfamilies for machine loading of flow-line cells. *International Journal of Production Research,* 33(9), pp.2357-2374.

Somló, J. and Kodeekha, E., 2007. Improving FMS scheduling by lot streaming. *Periodoca Polytechnica. Mechanical Engineering,* 51(1), pp.3-14.

Srivastava, B. and Chen, W., 1993. Part type selection problem in flexible manufacturing systems: Tabu search algorithms. *Annals of Operations Research*, 41, pp.279-297.

Stecke, K.E., 1983. Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Management Science*, 29(3), pp.273-288.

Stecke, K.E., 1986. A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems. *European Journal of Operational Research*, 24, pp.369-378.

Stecke, K.E. and Kim, I., 1991. A flexible approach to part type selection in flexible flow systems using part mix ratios. *International Journal of Production Research*, 29(1), pp.53-75.

Stecke, K.E. and Raman, N., 1994. Production planning decisions in flexible manufacturing systems with random materials flow. *IIE Transactions*, 26(5), pp.2-17.

Sujono, S. and Lashkari, R.S., 2007. A multi-objective model of operation allocation and material handling system selection in FMS design. *International Journal of Production Economics*, 105(1), pp.116-133.

Sung, C.S. and Joo, U.G., 1992. Single machine scheduling problem with earliness/tardiness and starting time penalties under a common due date. *Computers and Operations Research*, 19(8), pp.757-766.

Suri, R. and Zazanis, M.A., 1988. Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue. *Management Science*, 34(1), pp.39-64.

Taal, M. and Wortmann, J.C., 1997. Integrating MRP and finite capacity planning. *Production Planning & Control*, 8(3), pp.245-254.

Thangavelu, S.R. and Shetty. C.M., 1971. Assembly line balancing by zero-one integer programming. *AIIE Transactions*, 3(1), pp.61-68.

Thomas, L.J. and McClain, J.O., 1993. An overview of production planning. In: S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, eds. 1993. *Logistics of Production and Inventory*. Amsterdam: North-Holland, pp.333-370.

Toksari, M.D., Isleyen, S.K., Guner, E. and Baykoc, O.F., 2010. Simple assembly line balancing problem under the combinations of the effects of learning and deterioration. *Assembly Automation*, 30(3), pp.268-275.

Tolio, T., Urgo, M. and Váncza, J., 2011, Robust production control against propagation of disruption. *CIRP Annals – Manufacturing Technology,* 60, pp.489-492.

Towne, H., 1886., The engineer as an economist. *Transaction of the American Society of Mechanical Engineers*, 7, pp.428-432.

Váncza, J. and Kovács, A., 2004. Aggregation –The key to integrating production planning and scheduling. *CIRP Annals – Manufacturing Technology*, 53(1), pp.377-380.

Vollmann, T.E., Berry, W.L. and Whybark, D.C., 1997. *Manufacturing planning and control systems*. Irwin/McGraw-Hill.

Wagner, H.M., 1995. Global sensitivity analysis. *Operations Research*, 43(6), pp.948-969.

Waters, D., 1996. *Operations management. Producing goods & services*. Addison-Wesley Publishing Company.

Wendell, R.E., 1985. The tolerance approach to sensitivity analysis in linear programming. *Management Science,* 31(5), pp.564-578.

Wendell, R.E., 1992. Sensitivity analysis revisited and extended. *Decision Sciences*, 23. pp.1127-1142.

White, W.W., 1961. Comments on a paper by Bowman. *Operations Research*, 9, pp.274-276.

Wilson, J.M., 1989. An alternative formulation of the operation-allocation problem in flexible manufacturing systems. *International Journal of Production Research*, 27(8), pp.1405-1412.

Wilson, J.M., 1992. Approaches to machine load balancing in flexible manufacturing systems. *Journal of the Operational Research Society*, 43(5), pp.415-423.

Wilson, J.M., 2003. Gantt charts: A centenary appreciation. *European Journal of Operational Research*, 149, pp.430-437.

Wong, W.K., Mok, P.Y. and Leung, S.Y.S., 2006. Developing a genetic optimization approach to balance an apparel assembly line. *International Journal of Advanced Manufacturing Technology*, 28(3-4), pp.387-394.

Wortman, J.C., Euwe, J.M., Taal, M. and Wiers, S.C.V., 1996. A review of capacity planning techniques within standard software packages. *Production Planning & Control*, 7(2), pp.117-128.

Yelle, L.E., 1979. The learning curve: Historical review and comprehensive survey. *Decision Sciences*, 10, 302-328.

Zaeh, M.F. and Mueller, N., 2007. A modelling approach for evaluating capacity flexibilities in uncertain markets. *International Journal of Flexible Manufacturing Systems*, 19(3), pp.151-172.