Large-scale complex systems (LSS) have traditionally been characterized by large numbers of variables, structure of interconnected subsystems, and other features that complicate the control models such as nonlinearities, time delays, and uncertainties. The decomposition of LSS into smaller, more manageable subsystems allowed for implementing effective decentralization and coordination mechanisms. The last decade revealed new characteristic features of LSS such as the networked structure, enhanced geographical distribution and increased cooperation of subsystems, evolutionary development, and higher risk sensitivity. This chapter aims to present a balanced review of several traditional well-established methods and new approaches together with typical applications. First the hierarchical systems approach is described and the transition from coordinated control to collaborative schemes is highlighted. Three subclasses of methods that are widely utilized in LSS – decentralized control, simulation-based, and artificial-intelligence-based schemes – are then reviewed. Several basic aspects of decision support systems (DSS) that are meant to enable effective cooperation between man and machine and among the humans in charge with LSS management and control are briefly exposed. The chapter concludes by presenting several technology trends in LSS.

There is not yet a universally accepted definition of the large-scale complex systems (LSS) though the LSS movement started more than 40 years ago. However, by convention, one may say that a particular system is a large and complex one if it possesses one or several characteristic features. For example, according to Tomovic [36.1], the set of LSS characteristics includes the structure of interconnected subsystems and the presence of multiple objectives, which, sometimes, are vague and even conflicting. A similar viewpoint is proposed by Mahmoud, who describes a LSS as [36.2]:

A system which is composed of a number of smaller constituents, which serve particular functions, share common resources, are governed by interrelated goals and constraints and, consequently, require more than one controllers.

Šiljak [36.3] states that a LSS is characterized by its high dimensions (large number of variables), constraints in the information infrastructure, and the presence of uncertainties. At present there are software products on the market which can be utilized...
to solve optimization problems with thousands of variables. A good example is Solver.com [36.4]. Complications may still be caused by system non-linearities, time delays, and different time constants, and, especially over recent years, risk sensitivity aspects.

36.1 Background and Scope

In real life one can encounter lots of natural, man-made, and social entities that can be viewed as LSS. From the early years of the LSS movement, the LSS class has included several particular subclasses such as: steelworks, petrochemical plants, power systems, transportation networks, water systems, and societal organizations [36.5–7]. Interest in designing effective control schemes for such systems was primarily motivated by the fact that even small improvements in the LSS operations could lead to large savings and important economic effects.

The structure of interconnected subsystems has apparently been the characteristic feature of LSS to be found in the vast majority of definitions. Several subclasses of interconnections can be noticed (Fig. 36.1).

First there are the resource sharing interconnections described by Findeisen [36.8], which can be identified at the system level as remarked by Takatsu [36.9]. Also, at the system level, subsystems may be interconnected through their common objectives [36.8]. Subsystems may also be interconnected through buffer units (tanks), which are meant to attenuate the effects of possible differences in the operation regimes of plants which feed or drain the stock in the buffer. This type of flexible interconnection can frequently be met in large industrial and related systems such as refineries, steelworks, and water systems [36.10]. The dynamics of the stock value $s$ in the buffer unit can be modeled by a differential equation. In some cases buffering units are not allowed because of technological reasons; for example, electric power cannot be stocked at all and reheated ingots in steelworks must go immediately to rolling mills to be processed. When there are no buffer units, the subsystems are coupled through direct interconnections, at the process level [36.9].

In the 1990s, integration of systems continued and new paradigms such as the extended/networked/virtual enterprise were articulated to reflect real-life developments. In this context, Mårtenson [36.11] remarked that complex systems became even more complex. She provided several arguments to support her remark: first, the ever larger number of interacting subsystems that perform various functions and utilize technologies belonging to different domains such as mechanics, electronics, and information and communication technologies (ICT); second, that experts from different domains can encounter hard-to-solve communication problems; and also, that people in charge of control and maintenance tasks, who have to treat both routine and emergence situations, possess uneven levels of skills and training and might even belong to different cultures.

Nowadays, Nof et al. show that [36.12]:

There is the need to create the next generation manufacturing systems with higher levels of flexibility, allowing these systems to respond as a component of enterprise networks in a timely manner to highly dynamic supply-and-demand networked markets.
They also emphasize that e-Manufacturing is highly dependent on the efficiency of collaborative human–human and human–machine e-Work. See Chap. 88 on Collaborative e-Work, e-Business, and e-Service. In general, there is a growing trend to understand the design, management, and control aspects of complex supersystems or systems of systems (SoS). Systems of systems can be met in space exploration, military and civil applications such as computer networks, integrated education systems, and air transportation systems. There are several definitions of SoS, most of them being articulated in the context of particular applications; for example, Sage and Cuppan [36.13] state that a SoS is not a monolithic entity and possesses the majority of the following characteristics: geographic distribution, operational and management independence of its subsystems, emergent behavior, and evolutionary development. All these developments obviously imply ever more complex control and decision problems. A particular case which has received a lot of attention over recent years is large-scale critical infrastructures (communication networks, the Internet, highways, water systems, and power systems) that serve not only the business sector but society in general [36.14, 15]. All of these recent developments are likely to provide fresh strong stimuli for new research in the LSS domain.

### 36.1.1 Approaches

The progresses made in information and communication technologies have enabled the designer to overcome several difficulties he might have encountered when approaching a LSS, in particular those caused by a large number of variables and the low performance (with respect to throughput and reliability) of communication links. However, as Cassandras points out [36.58]:

> The complexity of systems designed nowadays is mainly defined by the fact that computational power alone does not suffice to overcome all difficulties encountered in analyzing, planning and decision-making in the presence of uncertainties.

A plethora of methods have been proposed over the last four decades for managing and controlling large-scale complex systems such as: decomposition, hierarchical control and optimization, decentralized control, model reduction, robust control, perturbation-
Hierarchical Systems Approach

The central idea of the hierarchical multilevel systems (HMS) approach to LSS consists of replacing the original system (and the associated control problem) with a multilevel structure of smaller subsystems (and associated less complicated problems). The subproblems at the bottom of the hierarchy are defined by the solutions of the lower-level subproblems.

Table 36.1 presents a summary of the main methods to be described in this chapter.

36.1.2 History

Though several ideas and methods for controlling LSSs were proposed in the 1960s and even earlier, it is accepted by many authors that the book of Mesarovic et al. published in 1970 [36.7] triggered the LSS movement. The concepts revealed in that book, even though they were strongly criticized in 1972 by Varaiya [36.60] (an authority among the pioneers of the LSS movement), have inspired many academics and practitioners.

A series of books including those of Wismer [36.61], Titi [36.17], Ho and Mitter [36.62], Sage [36.63], Šiljak [36.3, 64], Singh [36.65], Findeisen et al. [36.16], Jamshidi [36.6], Lunze [36.66], and Brdys and Tatjewski [36.18] followed on and contributed to the consolidation of the LSS domain of research and paved the way for practical applications.

In 1976, the first International Federation of Automatic Control (IFAC) conference on Large-Scale Systems: Theory and Applications was held in Udine, Italy. This was followed by a series of symposia which were organized by the specialized Technical Committee of IFAC and took place in various cities in Europe and Asia (Toulouse, Warsaw, Zurich, Berlin, Beijing, London, Patras, Bucharest, Osaka, and Gdansk). The scientific journal Large Scale Systems published by North Holland played an important role in the development of LSS domain, especially in the 1980s.

36.2 Methods and Applications

36.2.1 Hierarchical Systems Approach

The first step in analyzing an LSS and designing the corresponding control scheme consists of model building. As Steward [36.67] points out, practical experience witnessed there is a paradoxical law of systems. If the description of the plant is too complicated, then the designer is tempted to consider only a part of the system or a limited set of aspects which characterize its behavior. In this case it is very likely that the very ignored parts and aspects have a crucial importance. Consequently it emerges that more aspects should be considered, but this may lead to a problem which is too complex to be solved in due time. To solve the conflict between the necessary simplicity (to allow for the usage of existing methods and tools with a reasonable consumption of time and other computer resources) and the acceptable precision (to avoid obtaining wrong or unreliable results), the LSS can be represented by a family of models. These models reflect the behavior of the LSS as viewed from various perspectives, called [36.7] levels of description or strata, or levels of influence [36.63, 68]. The description levels are governed by independent laws and principles and use different sets of descriptive variables. The lower the level is, the more detailed the description of a certain entity is. A unit placed on the n-th level may be viewed as a subsystem at level n − 1. For example, the same manufacturing system can be described from the top stratum in terms of economic and financial models, and, at the same time, by control variables (states, controls, and disturbances) as viewed from the middle stratum, or by physical and chemical variables as viewed from the bottom description level (Fig. 36.2).
Levels of Control

In order to act in due time even in emergency situations, when the available data are uncertain and the decision consequences are not fully explored and evaluated, a hierarchy of specialized control functions can be an effective solution as shown by Eckman and Lefkowitz [36.70]. Several examples of sets of levels of control are:

a) Regulation, optimization, and organization [36.71]
b) Direct control, supervisory control, optimization, and coordination [36.72]
c) Stabilization, dynamic coordination, static optimization, and dynamic optimization [36.8]
d) Measurement and regulation, production planning and scheduling, and business planning [36.73].

The levels of control, also called *layers* by Mesarovic et al. [36.7], can be the result of a time-scale decomposition. They can be defined on the basis of time horizons taken into consideration, or the frequency of disturbances which may show up in process variables, operation conditions, parameters, and structure of the plant as stated by Schoeffler [36.68], as shown in Fig. 36.2.

Levels of Organization

The hierarchies based on the complexity of organization were proposed in mid 1960s by Brosilow et al. [36.74] and Lasdon and Schoeffler [36.75] and were formalized in detail by Mesarovic et al. [36.7]. The hierarchy with several levels of organization, also called *echelons* by Mesarovic et al. [36.7], has been, for many years, a natural solution for management of large-scale military, industrial, and social systems, which are made up of several interconnected subsystems when a centralized scheme cannot be either technically possible or economically acceptable.

The central idea of the multiechelon hierarchy is to place the control/decision units, which might have different objectives and information bases, on several levels of a management and control pyramid. While the multilayer systems implement the vertical division of the control effort, the multiechelon systems include also a horizontal division of work. Thus, on the *n*-th organization level the *i*-th control unit, \( CU^i_n \), has limited autonomy. It sends coordination signals downwards to a well-defined subset of control units which are placed at the level \( n-1 \) and it receives coordination signals from the corresponding unit placed

![Fig. 36.2](https://example.com/fig36.2.png) A hierarchical system approach applied to an industrial plant (after [36.69])
on level \( n + 1 \). The unit on the top of the pyramid is called the supralocal coordinator and the units to be found at the bottom level are called infimal units.

**Manipulation of Complex Mathematical Problems**

To take advantage of possible benefits of hierarchical multilevel systems a systematic decomposition of the original large-scale system and associated control problem is necessary. There are many situations when the control problem may be formulated as (or reduced to) an optimization problem \((P1)\), which is, in general terms as

\[
(P1) : \quad \text{extr} \ J(v) ; \quad v \in V ,
\]

where \( v \) is the decision variable (a scalar, or a vector), \( V \) is the admissible variation domain (which can be defined by differential or difference equations and/or algebraic inequations), and \( J \) is the performance measure (which can be a function or a functional).

The decomposition methods are based on various combinations of several elementary manipulations \([36.76]\). There are two main subsets of elementary manipulations:

1. **Transformations**, which are meant to substitute the original large-scale complex problem by a more manipulable one
2. **Decompositions**, which are meant to replace a large-scale problem by a number of smaller subproblems.

In the sequel several elementary manipulations will be reviewed following the lines exposed by Wilson \([36.76]\).

The **variable transformation** replaces the original problem \((P1)\) by an equivalent one \((P2)\) through the utilization of a new variable \( y = f(x) \) and a new performance measure \( Q(y) \) and admissible domain \( Y \), so that there is the inverse function \( v = f^{-1}(y) \). The new problem is defined as

\[
(P2) : \quad \text{extr} \ Q(y) ; \quad (y \in Y) , \quad (\forall y) = f(v)(Q(y) = J(v)) .
\]

The **Lagrangian transformation** can simplify the admissible domain; for example, let the domain \( V \) be defined by complicated equalities and inequalities

\[
V = \{ v : (v \in V_1) , \quad (g_0(v) , \quad g_-(v) \leq 0) \} ,
\]

where \( V_1 \) is a certain set, \( g_0 \) and \( g_- \) represent equality and inequality constraints, respectively.

A Lagrangian can be defined

\[
L(v, \pi, \gamma) = J(v) + \langle \pi, g_0(v) \rangle - \langle \gamma, g_-(v) \rangle ,
\]

where \( \pi \) are the Lagrange multipliers, \( \gamma \) are the Kuhn–Tucker multipliers, and \( \langle \cdot, \cdot \rangle \) is the scalar product.

If \( L \) possesses a saddle point, the solution of \((P1)\) is also the solution of the transformed problem \((P2)\) defined as

\[
(P2) : \quad \text{max min} \ [L(v, \pi, \gamma)] ; \quad v \in V_1 .
\]

The manipulation called **evolving the problem** is utilized when not all parameters are known or the priorities and the constraints are subject to alteration in time. In such situations, the problem is solved even under uncertainties and then is reformulated to take into account the accumulation of new information. The repetitive control proposed by Findeisen et al. \([36.16]\) is based on such a transformation.

Having transformed the original problem into a convenient form, a subset of smaller subproblems can be obtained through **decomposition** as shown in the sequel. The **partitioning** of the large-scale problem can be applied if several subsets of independent variables can
be identified; for example, let \((P1)\) be defined as
\[
(P1): \quad \text{extr}[J^1(v^1) + J^2(v^2)]; \quad v^1 \in V^1; \quad v^2 \in V^2,
\]
then, two independent subproblems \((P2^1)\) and \((P2^2)\) can be obtained
\[
(P2^1): \quad \text{extr}[J^1(v^1)]; \quad v^1 \in V^1, \tag{36.6}
\]
\[
(P2^2): \quad \text{extr}[J^2(v^2)]; \quad v^2 \in V^2. \tag{36.7}
\]
This decomposition is utilized in assigning separate subproblems to the controllers which are situated at the same level of a hierarchical pyramid or in decentralized control schemes where the controllers act independently.

The parametric decomposition divides the large-scale problem into a pair of subproblems by setting temporary values to a set of coupling parameters. While in one problem the pair of coupling parameters are fixed and all other variables are free, in the second subproblem they are free and the remaining variables are fixed as solutions of the first subproblem. The two subproblems are solved through an iterative scheme which starts with a set of guessed values of the coupling parameters; for example, let the large-scale problem be defined as follows
\[
(P1): \quad \text{extr}[J(v)]; \quad v = (\alpha, \beta); \quad (\alpha \in A), (\beta \in B); \quad \alpha R \beta,
\]
where \(\alpha\) and \(\beta\) are the components of \(v\), \(A\) and \(B\) are two admissible sets, \(\beta\) is the coupling parameter, and \(R\) is a relation between \(\alpha\) and \(\beta\). The problem \((P1)\) can be divided into the pair of subproblems \((P2)\) and \((P3)\)
\[
(P2): \quad \text{extr}_{\alpha} \left[ J \left( \alpha, \beta \right) \right]; \quad (\alpha \in A), (\alpha R \beta), \tag{36.8}
\]
\[
(P3): \quad \text{extr}_{\beta} \left[ J \left( \hat{\alpha} (\beta), \beta \right) \right]; \quad (\beta \in B), \quad \exists \hat{\alpha} \left[ (\hat{\alpha} \in A), (\hat{\alpha} R \beta) \right],
\]
where \(\hat{\alpha} (\beta)\) is the solution of \((P2)\) for the given value \(\beta = \beta^*\) and \(\hat{\beta}\) is the solution of \((P3)\) for the given value \(\alpha = \alpha^*\).

The parametric decomposition is utilized to divide the effort between a coordinating unit and the subset of coordinated units situated at lower organization level (echelon).

The structural decomposition divides the large-scale problem into a pair of subproblems through modifying the performance measure and/or constraints. While one subproblem consists in setting the best/satisfactory formulation of the performance measure and/or admissible domain, the second one is to find the solution of the modified problem. This manipulation is utilized to divide the control effort between two levels of control (layers).

From Coordination to Cooperation

The traditional multilevel systems proposed in the 1970s to be used for the management and control of large-scale systems can be viewed as pure hierarchies [36.77]. They are characterized by the circulation of feedback and intervention signals only along the vertical axis, up and down, respectively, in accordance with traditional concepts of the command and control systems. They constituted a theoretical basis for various industrial distributed control systems which possess at highest level a powerful minicomputer. Also the multilayer and multiechelon hierarchies served in the 1980s as a conceptual reference model for the efforts to design computer-integrated manufacturing (CIM) systems [36.78, 79].

Several new schemes have been proposed over the last 25 years to overcome the drawbacks and limits of the practical management and control systems designed in accordance with the concepts of pure hierarchies such as: inflexibility, difficult maintenance, and limited robustness to major disturbances. The more recent solutions exhibit ever more increased communication and cooperation capabilities of the management and control units. This trend has been supported by the advances in communication technology and artificial intelligence; for example, even in 1977, Binder [36.80] introduced the concept of decentralized coordinated control with cooperation, which allowed limited communication among the control unit placed at the same level. Several years later, Hatvany [36.81] proposed the heterarchical organization, which allows for exchange of information among the units placed at various levels of the hierarchy.

The term holon was first proposed by Koestler in 1967 [36.82] with a view to describing a general organization scheme able to explain the evolution and life of biological and social systems. A holon cooperates with other holons to build up a larger structure (or to solve a complex problem) and, at the same time, it works toward attaining its own objectives and treats the various situations it faces without waiting for any instructions from the entities placed at higher levels. A holarchy is
a hierarchy made up of holons. It is characterized by several features as follows [36.83]:

- It has a tendency to continuously grow up by attracting new holons.
- The structure of the holarchy may permanently change.
- There are various patterns of interactions among holons such as: communication messages, negotiations, and even aggressions.
- A holon may belong to more than one holarchy if it observes their operation rules.
- Some holarchies may work as pure hierarchies and others may behave as heterarchical organizations.

Figure 36.4 shows an object-oriented representation of a holarchy. The rectangles represent various classes of objects such as pure hierarchies, heterarchical systems, channels, and holons. This shows that the class of holarchies may have particular subclases such as pure hierarchies and heterarchical systems. Also a holarchy is composed of several constituents (subclasses) such as: holons (at least one coordinator unit and two infimal/coordinated units in the care of pure hierarchies) and channels for coordination (in the case of pure hierarchies) or channels for cooperation (in the case of pure heterarchies). Coordination channels link the supremal unit to, at least, two infimal units. While there are, at least, two such coordination links in the case of pure hierarchies, a heterarchical system may have no such link. While, at least, one cooperation channel is present in a heterarchical system, no such a link is allowed in a pure hierarchy.

Management and control structures based on holarchy concepts were proposed by Van Brussel et al. [36.84], Valckenauers et al. [36.85] for implementation in complex discrete-part manufacturing systems.

To increase the autonomy of the decision and control units and their cooperation the multiagent technology is recommended by Parunak [36.57] and Hadeli et al. [36.53]. An intelligent software agent encapsulates its code and data, is able to act in a proactive way, and cooperates with other agents to achieve a common goal [36.86]. The control structures which utilize the agent technology have the advantage of simplifying industrial transfer by incorporating existing legacy systems, which can be encapsulated in specific agents. Mařík and Lažanský [36.55] make a survey of industrial applications of agent technologies which also considers pros and cons of agent-based systems. They also present two applications:

- a) A shipboard automation system which provides flexible and distributed control of a ship’s equipment
- b) A production planning and scheduling system which is designed for a factory with the possibility of influencing the developed schedules by customers and suppliers.

### 36.2.2 Other Methods and Applications

**Decentralized Control**

Feedback control of large-scale systems poses the standard control problem: to find a controller for a given system with control input and control output ensuring closed-loop systems stability and reach a suitable input–output behavior. The fundamental difference between small and large systems is usually described by a pragmatic view: a system is large if it is conceptually or computationally attractive to decompose it into interconnected subsystems. Such subsystems are typically of small size and can be solved easier than the original system. The subsystem solutions can be combined in some manner to obtain a satisfactory solution for the overall system [36.87].

Decentralized control has consistently been a control of choice for large-scale systems. The prominent reason for adopting this approach is its capability to solve effectively the particular problems of dimensionality, uncertainty, information structure constraints, and time delays. It also attenuates the problems that communication lines may cause. While in the hierarchical control schemes, as shown above, the control
units are coordinated through intervention signals and may be allowed to exchange cooperation messages, in decentralized control, the units are completely independent or at least almost independent. This means that the information flow network among the control units can be divided into completely independent partitions. The units that belong to different subnetworks are completely separate from each other. Only restricted communication at certain time moments or intervals or limited to small part of information among the units is allowed. Decentralized structures are often used but their performance is worse compared with the centralized case. The basic decentralized control schemes are as follows:

- **Multichannel system.** The global system is considered as one whole. The control inputs and the control outputs operate only locally. This means that each channel has available only local information about the system and influences only a local part of the system.
- **Interconnected systems.** The overall system is decomposed according to a selected criterion. Then local controllers are designed for each subsystem. Finally, the local closed-loop subsystems and interconnections are tested to satisfy the desired overall system requirements.

At present a serious problem is the lack of relevant theoretic and methodological tools to support the scalable solution of new networked complex large-scale problems including asynchronous issues. The recent accomplishments are aimed at broadening the scope of decentralized control design methods using linear matrix inequalities (LMIs) [36.31], dynamic interaction coordinator design to ensure the desired level of interconnections [36.32], advanced decentralized control strategies for complex switching systems [36.26], hybrid large-scale systems [36.27], Petri nets [36.25], large-scale supply chain decentralized coordination [36.28, 29], and distributed control systems with network communication [36.30].

**Simulation-Based Scheduling and Control in LSS**

In continuous, large-scale industrial plants such as in chemical, power, and paper industries and waste-water treatment plants, simulation-based scheduling starts from creating scenarios for production and comparing these scenarios for optimality and availability. Problems can vary from order allocation between multiple production lines to optimal storage usage and detection and compensating for bottlenecks. Heuristic rules are usually connected to simulation, making it possible to adjust the production to varying customer needs, minimize the use of raw materials and energy, decrease the environmental load, stabilize or improve the quality, etc. Early applications in the paper industry are given by Leiviskä et al. [36.38, 39]. The main problem is to balance the production and several intermediate storages in (multiple) production lines, and give room for maintenance shutdowns and coordinate production rate changes. The model is based on the state model with storage capacities as the state variables and production rates as the control variables. Heuristics and bottleneck considerations are connected to these systems. A newer, agent-based solution has also been proposed [36.52]. There are also several classical optimization-based solutions for this problem [36.19, 21–24].

Modern chemical batch processes are large scale, complex, serial/parallel, multipurpose processes. They are especially common in the food and fine chemicals industries. They resemble flexible manufacturing systems common in electronics production. From the scheduling and control point of view complexity brings along also difficult interactions and uncertainty that are difficult to tackle with conventional tools. Simulation-based scheduling can include as much complexity as needed, and it is a largely used tool in the evaluation of the performance of different optimizing systems. Connecting heuristics or rule-based systems to simulation makes it also a flexible tool for batch process scheduling. Modeling approaches differ, e.g., real-time simulation using Petri nets [36.36] and the combination of discrete event simulation with genetic algorithms for the steel annealing shop have been proposed [36.40].

Flexible manufacturing systems, e.g., for components assembly, offer several difficulties for production scheduling and control. Dynamic, random nature is one main concern in operation control. Also quickly changing products and production environments, especially in electronics production, lead to a great variability in requirements for production control. In real cases, it is also typical that several scenarios must be created and evaluated. The handling of uncertain and vague information itself causes also problems in real-world applications. Uncertain data has to be extracted from data sources avoiding noise, or at least avoiding increasing it.

Discrete event simulation models the system as it propagates over time, describing the changes as separate
discrete events. This approach also found a lot of applications in manufacturing industries, queuing systems, and so on. An early application to jobshop scheduling is presented by Filip et al. [36.20], who utilize various combinations of several dispatching rules to create the list of future events. Taylor [36.17] reported on an application of discrete event simulation, combined with heuristics, to the scheduling of the printed circuit board (PCB) assembly line. The situation is complicated by the fact that the production control must operate on three levels: at the system level concerning production mix problems, at the cell level for routing problems, and at the machine level to solve sequencing problems. Discrete event simulation is also the key element in the shop floor scheduling system proposed by Gupta et al. [36.35]. The procedure starts by creating feasible schedules for the telephone terminals plant, helps in taking other requirements into account and in tackling uncertainties, and makes rescheduling possible. A system integrating simulation and neural networks has been used in photolithography toolset scheduling in wafer production [36.33]. The system uses the weighted-score approach, and the role of the neural network is to update the weights set to different selection criteria. Fuzzy logic provides the arsenal of methods for dealing with uncertainties. Several examples for PCB production are given by Leiviskä [36.45].

Two-stage approaches have been used in bottleneck-based approaches [36.34]. The first-pass simulation recognizes the bottlenecks, and their operation are optimized during the second-pass simulation. Better control of work in bottlenecks improves the performance of the whole system. The main dispatching rule is to group together the lots that need the same setups. The system also reveals the non-bottleneck machines and makes it possible to apply different dispatching rules according to the process state. The example is from semiconductor production.

In practice, scheduling is a part of the decision hierarchy starting from the enterprise-level strategic decisions and going down to machine-level order or tools scheduling. Simulation is used at different levels of this hierarchy to provide interactive means for guaranteeing the overall optimality or at least the feasibility of the decisions made at different levels. Such integrated and interactive approaches exist also in supply-chain management systems. In large-scale manufacturing systems, supply-chain control must take four interacting factors into account: suppliers, manufacturing, distribution, network, and customers. To control all these interactions successfully, various operating factors and constraints – processing times, production capacities, availability of raw materials, inventory levels, and transportation times – must be considered.

Discrete event simulation is also one possibility to create an object-oriented, scalable, simulation-based control architecture for supply-chain control [36.41]. Requirements for modularity and maintainability also lead to distributed simulation models, especially when a simulation-based control architecture is controlling supply chain interactions. This means a modeling technique including a federation of simulation models that are solved in a coordinated manner. The system architecture is presented in [36.42]. Each supply-chain entity has two simulation models associated with it – one running in real time and the other as a lookahead simulation. The lookahead model is capable of predicting the impact of a disturbance observed by the real-time model. A federation object coordinator (FOC) coordinates the real-time simulation models. In this case, a master event calendar allocates interprocess events to all simulation models and resynchronizes all simulations at the end of every activity [36.37].

In simulation-based control the controller makes decisions based both on the current state of the system and future scenarios, usually produced by simulation. Here, the techniques for calculation of these scenarios play the main role. Ramakrishnan and Thakur [36.42] proposed the extension sequential dynamic systems (SDS) that they call input–output SDS to model and analyze distributed control systems and to compensate for the weaknesses of automata-based models. They use the discrete-part production plant as an example.

**Artificial Intelligence-Based Control in LSS**

*Artificial intelligence (AI)-based control in large-scale systems uses, in practice, all the usual methods of intelligent control: fuzzy logic, neural networks, and genetic algorithms together with different kinds of hybrid solutions [36.88]. The complex nature of applications makes the use of intelligent systems advantageous. Dealing with this complexity is also the biggest challenge for the methodological development: the large-scale process structures, complicated interconnections, nonlinearity, and multiple time scales make the systems difficult to model and control. Fuzzy logic control (FLC) has found most of its applications in cases which are difficult to model, suffer from uncertainty or imprecision, and where a skilful operator is superior to conventional automation systems. Artificial neural networks (ANN) contribute to modeling and forecasting tasks and combined with fuzzy logic in neuro-fuzzy systems.
combine the benefits of both approaches. Genetic algorithms (GA), which are basically optimization systems, are used in tuning models and controllers. See Chap. 14 on Artificial Intelligence and Automation for additional content.

As shown above, the control of large-scale industrial plants have usually been based on distributed hardware and hierarchical design of control functions [36.89, 90]. The supervisory and local control levels lay under enterprise and mill-wide control levels. Supervisory control provides the local controls with the set points that fulfill the quality and schedule requirements coming from the mill-wide level and help in optimizing the operation of the whole plant. This optimization leaves room for versatile application of intelligent methods. Local units on the other hand, control the actual process variables according to the set points given by the supervisory control level. Even though the proportional–integral–differential (PID) controller is by far the most important tool, intelligent control plays an increasing role also at the local control level. Intelligent methods have been useful in tuning local PID controllers. In practice, fuzzy controllers must have adaptive capabilities. Gain scheduling is a typical approach for large-scale systems, but applications of model reference adaptive control and self-tuning adaptive control exist. Self-tuning has been used in controlling a pilot-scale rotary drum where the disturbances are due to long and varying time delays and changes in the raw materials [36.46].

Model-based control techniques, e.g., model predictive control (MPC), have been applied for the control of processes with a long delay or dead time. In MPC, the controller based on a plant model determines a manipulated variable profile that optimizes some performance objectives over the time in question. ANN are used in replacing the mathematical models in optimization as shown in a survey made by Hussain [36.48]. Also Takagi–Sugeno fuzzy models are used in connection with model-based predictive control [36.44]. Hybrid systems include both continuous- and/or discrete-time dynamics together with discrete events. So their state consists of real-valued, discrete-valued, and/or logical variables. Support vector machines have been used as a part of MPC strategy for hybrid systems [36.91].

Power systems have been an important application field for intelligent control since 1990s [36.92]. Design of centralized controllers is difficult for many obvious reasons: power systems are large scale and decentralized by nature. They are also nonlinear and have multiple dynamics and considerable time delays. Decentralized local control can apply linear models and purely local measurements. Available transfer capability (ATC) is a real-time index used in monitoring and controlling the power transactions and avoiding overloading of the transmission lines [36.47]. There are difficulties in calculating it accurately online for large-scale systems. Decreasing the number of input variables to only three and using fuzzy modeling helps in this. Simulations show that neural-networks-based local excitation controls can take care of interactions between generators and dampen oscillations effectively. Neural networks are used in approximating unknown dynamics and interconnections [36.40]. The designing of the controller for two-area hydrothermal power systems based on genetic algorithm improves the rise time and settling time, and simulations show that the proposed technique is superior to the traditional methods [36.51]. A local Kalman filter and genetic algorithms estimate all local states and interactions between subsystems in a large-scale power system. The controller uses these estimates, optimizes a given performance index, and then regulates the system states [36.50].

Agent-based technologies have been used in complex, distributed systems. Good examples come from intelligent control of highly distributed systems in the chemical industry and in the area of utility distribution (power, gas, and waste-water treatment). As shown above, holonic agents take care of machine or cell-level (local) controls, sometimes even integrated with machines. Intelligent agents can be associated with each manufacturing unit and they communicate, coordinate their activities, and cooperate with each other.

Fault detection and diagnosis (FDD) may be tackled by decomposing the large-scale problem into smaller subtasks and performing control and FDD locally [36.93]. Large-scale complex power systems need systematic tools for protection and control. The supervisory control technique and a design procedure of a supervisor that coordinates the behavior of relay agents to isolate fault areas are presented in [36.56]. Multiagent systems have also been used in identification and control of a 600 MW boiler-turbine-generator unit [36.54]. In this case, online identifiers are used for control and offline identifiers for fault diagnosis. Event-based approaches are used for building large-scale distributed systems and applications, especially in a networked environment. A hybrid approach of event-based communications for real-time manufacturing supervisory control is applied for large-scale warehouse management [36.94]. See Chap. 30 on Automating Error and Conflict Prognostics and Prevention for additional content.
Computer-Supported Decision Making in Large-Scale Complex Systems

As shown above, a possible solution to many LSS control problems is the use of artificial-intelligence methods. However, in the field, due to strange combinations of external influences and circumstances, rare or new situations may show up that were not taken into consideration at design time. Already in 1990, Martin et al. remarked that [36.95]:

> although AI and expert systems were successful in solving problems that resisted to classical numerical methods, their role remains confined to support functions, whereas the belief that evaluation by man of the computerized solutions may become superfluous is a very dangerous error.

Based on this observation, Martin et al. [36.95] recommended appropriate automation, which integrates technical, human, organizational, economical, and cultural factors.

The decision support system concept (DSS) appeared in the early 1970s. As with any new term, the significance of DSS was in the beginning rather vague and controversial. While some people viewed it as a new redundant term used to describe a subset of management information systems (MIS), some other argued it was a new label abusively used by some vendors to take advantage of a new fashion. Since then many research and development activities and applications have witnessed that the DSS concept definitely meets a real need and there is a market for it even in the context of real-time applications in the industrial milieu [36.96,97].

The Nobel Prize winner H. Simon [36.98] identified three steps of the decision-making (DM) process, namely:

a) Intelligence, consisting of activities such as data collection and analysis in order to recognize a decision problem
b) Design, including activities such as model statement and identification/production and evaluation of various potential solutions to the problem
c) Choice, or selection of a feasible alternative for implementation.

Later, he added a fourth step – implementation and result evaluation – which may correspond to supervisory control in industrial milieu. If a decision problem cannot be entirely clarified and all possible decision alternatives cannot be fully explored and evaluated before a choice is made, then the problem is said to be unstructured or semistructured. If the problem were completely structured, an automatic device could have solved the problem without any human intervention. On the other hand, if the problem has no structure at all, nothing but hazard can help. If the problem is semistructured a computer-aided decision can be envisaged.

Most of the developments in the DSS domain have initially addressed business applications not involving any real-time control. However, even in the early 1980s DSS were reported to be used in manufacturing control [36.20, 99]. In 1987, Bosman [36.100] stated that control problems could be looked upon as a natural extension and as a distinct element of planning decision-making processes (DMP). Almost 20 years later, Nof et al. state [36.12]:

> … the development and application of intelligent decision support systems can help enterprises cope with problems of uncertainty and complexity, to increase efficiency, join competitively in production networks, and improve the scope and quality of their customer relations management (CRM).

Real-time decision-making processes (RT DMPs) for control applications are characterized by several particular aspects such as:

a) They involve continuous monitoring of a dynamic environment.
b) They are short time horizon oriented and are carried out on a repetitive basis.
c) They normally occur under time pressure.
d) Long-term effects are difficult to predict [36.101].

It is quite unlikely that an econological (economically logic) approach, involving optimization, be technically possible for genuine RT DMPs. Satisficing approaches, which reduce the search space at the expense of the decision quality, or fully automated DM systems, if taken separately, cannot be accepted either, but for some exceptions. At the same time, one can notice that genuine RT DMP can show up in crisis situations only; for example, if a process unit must be shut down due to an unexpected event, the production schedule of the entire plant might become obsolete. The right decision will be to take the most appropriate compensation measures to manage the crisis over the time period needed to recomputed a new schedule or update the current one. In this case, a satisficing decision may be appropriate. If the crisis situation has been met previously and successfully surpassed, an almost automated solution based on past decisions stored in the
Table 36.2 Possible task assignment in DSS

<table>
<thead>
<tr>
<th>Decision steps and activities</th>
<th>EU</th>
<th>NU</th>
<th>NM</th>
<th>ES</th>
<th>ANN</th>
<th>CBR</th>
<th>IA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intellgence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Setting objectives</td>
<td>I</td>
<td>M</td>
<td>I/M</td>
<td>P</td>
<td>M/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Perception of DM situation</td>
<td>I</td>
<td>M</td>
<td>P</td>
<td></td>
<td>M/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Problem recognition</td>
<td>I</td>
<td>M</td>
<td>P</td>
<td></td>
<td>M/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Model selection</td>
<td>E</td>
<td>M</td>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Model building</td>
<td>M</td>
<td>P</td>
<td>I/M</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Model validation</td>
<td>I</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Setting alternatives</td>
<td>P</td>
<td>M</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Model experimenting</td>
<td></td>
<td></td>
<td>M/I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Model solving</td>
<td>E</td>
<td>E</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Result interpreting</td>
<td>I</td>
<td>P</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>– Parameter changing</td>
<td>E</td>
<td>M/I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Solution adopting</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Sensitivity analysis</td>
<td>M</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release for implementation</td>
<td>E</td>
<td>E</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Information system can be accepted and validated by the human operator. On the other hand, the minimization of the probability of occurrences of crisis situations should be considered as one of the inputs (expressed as a set of constraints or/and objectives) in the scheduling problem [36.96, 102].

In many problems, decisions are made by a group of persons instead of an individual. Because the group decision is either a combination of individual decisions or a result of the selection of one individual decision, this may not be rational in Simon’s acceptance. The group decision is not necessarily the best choice or a combination of individual decisions, even though those might be optimal, because various individuals might have various perspectives, goals, information bases, and criteria of choice. Therefore, group decisions show a high social nature, including possible conflicts of interest, different visions, influences, and relations [36.103]. Consequently, a group (or multiparticpant) DSS needs an important communication facility.

The generic framework of a DSS, proposed by Bonczek et al. in 1980 [36.104] and refined later by Holsapple and Whinston [36.105] is quite general and can accommodate the most recent technologies and architectural solutions. It is based on three essential components. The first one is the language (and communications) subsystem (LS). This is used for:

- Directing data retrieval, allowing the user to invoke one out of a number of report generators
- Directing numerical or symbolic computation, enabling the user either to invoke the models by names or construct model and perform some computation at his/her free will
- Maintaining knowledge and information in the system
- Allowing communication among people in case of a group DM
- Personalizing the user interface.

The knowledge subsystem (KS) normally contains:

- Empirical knowledge about the state of the application environment in which the DSS operates
b) **Modeling knowledge**, including basic modeling blocks and computerized simulation and optimization algorithms to use for deriving new knowledge from the existing knowledge

c) **Derived knowledge** containing the constructed models and the results of various computations

d) **Meta-knowledge** (knowledge about knowledge) supporting model building and experimentation and result evaluation

e) **Linguistic knowledge** allowing the adaptation of system vocabulary to a specific application

f) **Presentation knowledge** to allow for the most appropriate information presentation to the user.

The third essential component of a DSS is the problem processing subsystem (PPS), which enables combinations of abilities and functions such as information acquisition, model formulation, analysis, evaluation, etc.

It has been noticed that some DSS are oriented towards the left hemisphere of the human brain and some others are oriented towards the right hemisphere. While in the first case quantitative and computational aspects are important, in the second pattern recognition and reasoning based on analogy prevail. In this context, there is a significant trend towards combining numerical models and models that emulate the human reasoning to build advanced DSS [36.106]. A great number of optimization algorithms have been developed and carefully tested so far. However, their effectiveness in decision making has been limited. Over the last three decades traditional numerical methods have, along with databases, been essential ingredients of DSS. From an information technology perspective, their main advantages [36.107] are: compactness, computational efficiency (if the model is correctly formulated), and the market availability of software products. On the other hand, they present several disadvantages. Because they are the result of intellectual processes of abstraction and idealization, they can be applied to problems which possess a certain structure, which is hardly the case in many real-life problems. In addition, the use of numerical models requires that the user possesses certain skills to formulate and experiment the model. As was shown in the previous section, AI-based methods supporting decision making are already promising alternatives and possible complements to numerical models. New terms such as tandem systems, or expert DSS (XDSS) have been proposed for systems that combine numerical models with AI-based techniques. An ideal task assignment is given in Table 36.2 [36.97].

### 36.3 Case Studies

The following case studies illustrate how combinations of methods may be utilized to solve large-scale complex problems.

#### 36.3.1 Case Study 1: Pulp Mill Production Scheduling

Figure 36.5 shows the pulp mill modeled as a common state-space system. The state of the system $s(t)$ is described by the amount of material in each storage tank. The production rates of the processes are chosen as control variables forming the control vector $m(t)$. The required pulp production is usually taken as a deterministic known disturbance vector $w(t)$.

The operation of the plant presented in Fig. 36.5 is described by the vector–matrix differential equation

$$\frac{ds(t)}{dt} = Bm(t) + Cw(t),$$

where $B$ and $C$ are coefficient matrices describing the relationships between the model flows (transfer ratios).
Since the most storage tanks have only one input flow and one output flow, most elements in $B$ and $C$ matrices equal zero.

If the steam balance (dashed line in Fig. 36.5) is included in scheduling, an additional variable describing the steam development in the auxiliary boiler is required. It is a scalar variable denoted by $S$. Accordingly, the steam balance is

$$S(t) = Dm(t) + Ew(t).$$

Note that the right-hand side of the balance includes both consumption and generation terms. The variables in the model are constrained by the capacity limits of tanks and processes in the following way:

$$s_{\text{min}} \leq s(t) \leq s_{\text{max}},$$
$$m_{\text{min}} \leq m(t) \leq m_{\text{max}},$$
$$S_{\text{min}} \leq S(t) \leq S_{\text{max}}.$$

Due to the fact that scheduling is concerned with relatively long time intervals, no complete and complicated process models are necessary. If all the small storage tanks are included in the model, the system dimensions increase and it becomes difficult to deal with. These tanks also have no meaning from the control point of view. Simpler model follows by combining small storage tanks.

There are several ways to solve the scheduling problem as shown before. Optimization can benefit from decomposition and solving of smaller problems as described in Sect. 36.2.1. A review of methods is presented in [36.39]. It seems, however, that no approach alone can deal with this problem successfully. Hybrid systems, consisting of algorithmic, rule-based, and intelligent parts integrated with each other, and also agent-based systems, could be the best possible answer [36.97, 110].

### 36.3.2 Case Study 2: Decision Support in Complex Disassembly Lines

In [36.108], the control of a complex industrial disassembly process of out-of-use manufactured products is studied. The disassembly processes are subject to uncertainties. The most difficult problem in such systems is that a disassembly operation can fail at any moment because of the product or component degradation. In this case one has to choose between applying an alternative disassembly destructive operation (dismantling), and aborting the disassembly procedure. This decision must be taken in real time because in a used product the components states are not known from the beginning of the process. The solution is to integrate a decision support system (DSS) in the architecture of a multilayer system. As shown in Fig. 36.6, the control and decision tasks are distributed among three levels: planning, decision support, and direct control. The disassembly planner gives the sequence of the components that must be separated to achieve the target component. The planner fuses the information from the artificial vision system with that contained in the database for each component or subassembly. A model of the product is generated. The DSS integrates the model and performs the simulation to rec-

---

**Fig. 36.6** DSS integration in the multilayer control system (after [36.108])

**Fig. 36.7** The results of time delay estimation for one group (after [36.109])
ommend a good disassembly sequence with respect to the economical criteria.

**36.3.3 Case Study 3: Time Delay Estimation in Large-Scale Complex Systems**

In data-based modeling of large-scale complex systems, the exact determination of time delays is extremely difficult. The methods for delay estimation are widely studied in control engineering, but these studies are mainly limited to the two-variable cases, i.e., estimating the delay between the manipulated and the controlled variable in the feedback control loop. The situation is totally different when dealing with a large number of variables grouped in several groups for modeling or monitoring purposes.

Mäyrä et al. [36.109] discuss a delay estimation scheme combining genetic algorithms and principal component analysis (PCA). Delays are optimized with genetic algorithms with objective functions based on PCA. Typically, a genetic algorithm maximizes the variance explained by the first or two first principal components. The paper gives an example using simulation data of the paper machine, which includes over 50 variables. The variables were first grouped based on the cross-correlation and graphical analysis into five groups, and delays were estimated both for the variables inside the groups and between the groups. The results for one group of 15 variables are given in Fig. 36.7. The estimation was repeated 60 times and the figure shows the median and standard variance of these simulations.

**36.4 Emerging Trends**

Large-scale complex systems have become a research and development domain of automation with a series of rather established method and technologies and industrial application. Table 36.3 contains a summary of references to basic concepts.

**Table 36.3 Key to references on basic concepts**

<table>
<thead>
<tr>
<th>Type</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic books</td>
<td>Mesarovic, Macko, Takahara [36.7]; Wismer [36.61]; Titli [36.17];</td>
</tr>
<tr>
<td></td>
<td>Ho and Mitter [36.62]; Sage [36.63]; Šiljak [36.3, 64]; Singh [36.65];</td>
</tr>
<tr>
<td></td>
<td>Findelis et al. [36.16]; Jamshidi [36.6]; Lunze [36.66];</td>
</tr>
<tr>
<td></td>
<td>Brdys and Tatjewski [36.18]</td>
</tr>
<tr>
<td>Hierarchies</td>
<td>Mesarovic, Macko, Takahara [36.7]</td>
</tr>
<tr>
<td>Strata</td>
<td>Sage [36.63]; Schoeffler [36.68]</td>
</tr>
<tr>
<td>Layers</td>
<td>Findelis [36.8]; Havlena and Lu [36.73]; Isermann [36.72];</td>
</tr>
<tr>
<td></td>
<td>Lefkowitz [36.111]; Schoeffler [36.68]; Brdys and Ulanicki [36.112]</td>
</tr>
<tr>
<td>Echelons</td>
<td>Brosilow, Lasdon and Pearson [36.74]; Lasdon and Schoeffler [36.75]</td>
</tr>
<tr>
<td>Heterarchy</td>
<td>Hatvany [36.81]</td>
</tr>
<tr>
<td>Holarchy</td>
<td>Hop and Schaeffler [36.83]; Koestler [36.82]; Van Brussel et. al. [36.84];</td>
</tr>
<tr>
<td></td>
<td>Valckenaers et al. [36.85]</td>
</tr>
<tr>
<td>Decision support systems</td>
<td>Bonczek, Holsapple and Whinston [36.104]; Bosman [36.100];</td>
</tr>
<tr>
<td></td>
<td>Chaturverdi et al. [36.101]; De Michelis [36.103]; Dutta [36.107];</td>
</tr>
<tr>
<td></td>
<td>Filip [36.96]; Filip et. al. [36.21]; Filip, Donciulescu and Filip [36.97];</td>
</tr>
<tr>
<td></td>
<td>Holsapple and Whinston [36.105]; Kusiak [36.106]; Martin et. al. [36.95];</td>
</tr>
<tr>
<td></td>
<td>Nof [36.99]; Nof et al. [36.12]; Simon [36.98]</td>
</tr>
</tbody>
</table>
A promising modern form to coordinate the actions of the intelligent agents is *stigmergy*. This is inspired by the behavior of social insects which use a form of indirect communication mediated by an active environment to coordinate their actions [36.53].

Advanced decentralized control strategies for large-scale complex systems have recently been extended into new applied areas, such as flexible structures [36.113, 114], Internet congestion control [36.115], aerial vehicles [36.116], or traffic control [36.117], to mention a few of them.

Recent theoretic achievements in decentralized control can be progressively extended into the areas of integrated/embedded control, distributed control (over communication networks), hybrid/discrete-event systems and networks, and autonomous systems to serve as a very efficient tool to solve various large-scale control problems.

Incorporation and combination of newly developed numeric optimization and simulation models and symbolic/and connectionist or agent-based will continue in an effort to reach the *unification of humans*, numerical models, and AI-based tools.

Mobile communications and web technology will be ever more considered in LSS management and control applications. In multiparticipant DSS, people will make co-decisions in *virtual teams*, no matter where they are temporarily located.

**References**


36.3   D.D. Siljak: *Large Scale Dynamical Systems: Stability and Structure* (North Holland, Amsterdam 1978)


36.6   W. Findeisen: Decentralized and hierarchical control of large-scale systems, Plenary paper, Proc. 7th IFAC Congr. (Pergamon, Oxford 1978), 871–2382


36.8   W. Findeisen: Decentralized and hierarchical control under consistence or disagreements of interests, Automatica 18(6), 647–664 (1982)


36.17  A. Titli: *Commande hierarchisée des systemes complexes* (Dunod Automatique, Paris 1975)


M. Dehghani, A. Afshar, S.K. Nikravesh: Decentralized stochastic control of power systems using genetic algorithms for interaction estimation, Proc. 16th IFAC World Congr. (Prague 2005), on CD ROM


36.57 H.V.D. Parunak: Practical and Industrial Applications of Agent-Based Systems (Industrial Technology Institute, Ann Arbor 1998)


36.65 M.G. Singh: Dynamic Hierarchical Control (North Holland, Amsterdam 1978)


36.69 J. Minsker, S. Piggot, G. Freidson: Hierarchical automation control systems for large-scale systems and applications, Proc. 5th IFAC World Congr. (Paris 1972)


36.78 T.J. Williams: Analysis and Design of Hierarchical Control Systems with Special Reference to Steel Plant Operations (Elsevier, Amsterdam 1985)


36.80 Z. Binder: Sur l’organisation et la conduite des systemes complexes, These de Docteur (IAG, Grenoble 1977) in French


36.86 G. Tecuci: Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodol-
638 Part D Automation Design: Theory and Methods for Integration


36.108 L. Dutta, J.M. Henrioud, F.G. Filip: Applying equal piles approach to disaster line balancing problem, Proc. 16th IFAC World Congr. (Session Industrial Assembly and Disassembly) (Prague 2005), on CD-ROM


