14

Cross-Layer Adaptation and Optimization for Cognitive Radio

Hüseyin Arslan[1] and Serhan Yarkan[2]

[1] University of South Florida, Tampa, FL, USA, arslan@eng.usf.edu
[2] University of South Florida, Tampa, FL, USA, syarkan@eng.usf.edu

14.1 Introduction

When we look at the evolution of wireless communication systems, as in most systems, two major developments are most prominent: 1) addition of new features and 2) improvement of already existing capabilities. The first development arises from the fact that communication through a wireless medium makes life easier. Every development brings several new features and these new features contribute positively to this fact. These developments can be seen clearly in the evolution of cell phones. In the past, cell phones were used only for transmitting voice and short text-based messages. Currently, there are cell phones in which an operating system runs and several multimedia applications are available.

The second type of development solely originates from the fact that every physical concept is finite. Therefore, according to the principal of parsimony,[1] the newer systems restrain themselves from wasting resources. Using resources adequately under dynamically changing conditions introduces the notion of adaptation and optimization.

Although systems evolve in terms of 1) and 2), particularly in communication systems, it can be observed that the fundamental design architecture, which is known as "layered architecture," still remains the same. Despite the inefficiency of contemporary communication systems, they still accomplish their tasks. Nevertheless, it can be seen that the aforementioned evolutionary developments are approaching a saturation point for contemporary communication systems. The fundamental design architecture inherently hinders

[1] *Entia non sunt multiplicanda, præter necessitatem* [1]. The famous statement, which is also known as "Occam's Razor" and believed that phrased by William of Ockham, which means "Entities should not be multiplied unnecessarily" [2]. A direct consequence of this statement is that for two systems which accomplish the same task, the one that accomplishes the objective in lesser amount of effort, element, unit, etc., is preferable to the other.

H. Arslan (Ed.), Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems, 421–452.

applicability of some of the new developments in the world of communications. This fact was seen by researchers, but they have only patched the flaws rather than applying radical changes. However, with the emergence of cognitive radio, which was coined by Joseph Mitola III [3], the perception of adaptation and optimization of wireless communication systems gained new dimensions and perspectives. The emergence of cognitive radio (and cognitive engine) is a promising solution for the barrier which arises from the flaws of the fundamental design architecture.

In this chapter, we search for an answer to the question how can a global (or multilayer) adaptation and optimization be established for cognitive radio? In order to be able to provide a concrete answer, first we will briefly review the fundamental design architecture while providing the reasons and relevant efforts of migration from the traditional architecture to cross-layer design. Next, we will outline cross-layer architecture and the place of adaptation along with optimization for the past and contemporary wireless communication systems. Subsequently, we will provide the essentials for an overall adaptation and optimization process in terms of cross-layer architecture. Later, we will introduce three cross-layer application examples depending on the relationships between layers as adjacent layer interaction, nonadjacent layer interaction, and composite interaction. Then, we will investigate the optimization problems from a formal perspective to be able to gain some insight into cross-layer optimization for cognitive radio. We will also address multi-objective optimization problems (MOPs) and relevant solutions which are going to run on cognitive engine. In addition, we will summarize the challenges related to cross-layer adaptation and optimization for cognitive radio. As a final remark, we will extend individual cross-layer adaptation and optimization problems to a network in which there are multiple individuals.

14.2 Why We Need Cross-Layer Design, Adaptation, and Optimization

14.2.1 Traditional Layered Design and Its Evolution

Traditional protocol stack has been designed for dealing with complicated problems by breaking them into smaller parts. It consists of layers whose definitions and tasks are defined explicitly and independently. In other words, each layer is isolated from the others except for providing output to and getting input from adjacent layers [4]. According to the direction of the flow upward/downward, each layer conducts its own task by taking inputs from the layer below/above and conveys the outputs obtained to above/below. This architecture has several advantages. First, defining the tasks explicitly provides modularity, which means simplicity in the design. Second, explicit definitions facilitate the standardization process. Therefore, several vendors can produce various types of products by following the explicit abstractions, and

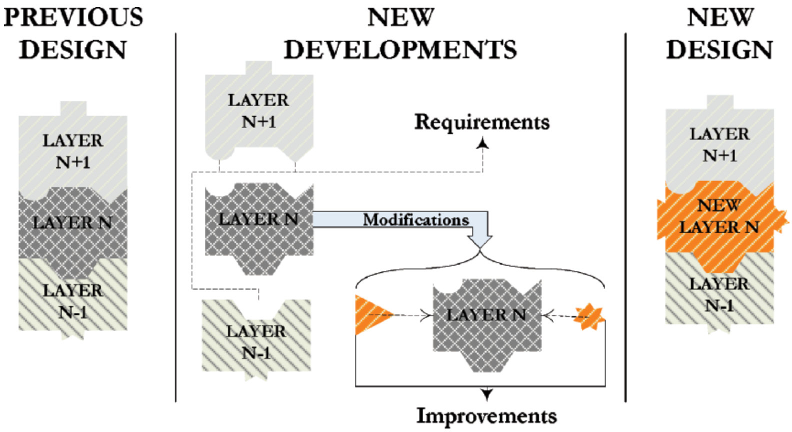


Fig. 14.1. The only consideration in designing a new layer for traditional architecture is that the new layer design should meet the requirements of the adjacent layers. The requirements of layer $N+1$ are denoted as a round and a triangle, whereas that of layer $N-1$ is represented with a half–hexagon.

at the same layer level, any of them can communicate with the others without having any problem. Finally, modularity that comes with independence allows any type of alteration at any layer level as long as the input/output requirements of the adjacent layers are met, which gives rise to the concept of expandability of the layers as illustrated in Figure 14.1.

Even though layered structure overcomes many problems successfully, it has been realized that the stringent architecture creates some problems such as asynchrony and inefficiencies. A brief list of the major advantages and disadvantages of the traditional layered architecture and their effects is given in Table 14.1. At the beginning, the disadvantages have been tried to alleviate by increasing the amount of information flow among the layers [5]. Afterwards, with the emergence of new high-speed communication networks, the major handicap of the traditional design, which is the obligation of an algorithmic ordering, has been emphasized more. As in algorithms, in the traditional architecture, a process which resides in the next adjacent layer cannot be executed unless a process defined in a layer is completed. In early '90s, the obligated ordering concept, which causes long communication delays and low throughput for wired networks, has been tried to overcome by inter-layer optimization [6]. Inter-layer optimization re-organizes the processes for layers in such a way that some of the processes that do not interfere with each other can be integrated and executed in parallel rather than serial.[2] This approach is extended to a new protocol scheme called Horizontally Oriented Protocol

[2] At this point, it must be stated that even though some of the processes are integrated, inter-layer optimization still preserves the strictly isolated structure of layers [6].

Table 14.1. The major advantages and disadvantages of the traditional layered architecture and their effects.

+/−	**Explanation**	**Effect**
Advantages		
Modularity	Each layer can be designed independent of others	Simpler design
Standardization	Design only requires to have the knowledge of explicit definitions and abstractions	Interoperability
Expandability	Layers can be updated, altered, or expanded "independently"	Individual flexibility
Disadvantages		
Ordering	Execution of any process in any layer has to be after the execution of previous processes in former layers	• Inefficiency • Latency
Interaction	Due to strict isolation, information cannot cross other layers	• Unawareness • Redundant processes • Sub-optimal performance
Adaptation	In wireless communications, rapid channel variations cannot be responded immediately	• Decrease in capacity • Sub-optimal performance
Topologies	Some of the network topologies need flexible layer architecture	Inefficiency

Structure (HOPS). In HOPS, the building stone of the architecture is defined as "function," instead of "layer." Thus, the functions that do not need to wait for others' outcome can be binded and executed in parallel, which is the reason why the scheme is called as "horizontally oriented" [7].

Migration from strict layered architecture to a more flexible interactive one has another very strong motivation: *wireless* networks. Because of the different nature of wireless communications, numerous concepts defined in wired networks need careful re-consideration or even modification, so does the protocol stack. Peculiar to wireless communications, due to small-scale fading, wireless channel conditions may change drastically in a very short duration of time [8]. Therefore, in order to take advantage of the durations in which the channel is identified as "good," a flexible design is essential [9]. Large-scale channel variations contribute to the necessity of flexible architecture as well [10]. Interference and time-varying capacity property due to multipath, relative mobility, and shadowing are other very crucial parameters that affect the wireless networks [8, 11]. Apart from those, new transmission schemes for wireless communications such as relay networks [12] may not be established via a strictly isolated layered architecture [9] and might require a different design.

Especially in wireless networks, *cross-layer approaches* emerged by increasing the amount of information provided to the adjacent layers. As an immediate example, packet losses in wireless networks can be considered. In wireless networks, packet losses can occur because of bad wireless channel conditions, congestion, or some other reasons. However, operating in transport layer, the Transmission Control Protocol (TCP) cannot comprehend the reason behind these losses. Therefore, it assumes that packet losses solely depend on congestion [13]. Thus, any loss detected is going to be handled[3] in terms of congestion even though the actual reason might be different. In order to compensate for this flaw, the link reliability is also tried to be improved as much as possible. Hence, in case of a loss, wireless link is eliminated from other possible causes, so TCP can handle the issue by considering other factors [13]. In fact, this approach solely is not enough to optimize the problem because of the drop of throughput. However, if the information about the reason of the loss can be obtained from the layers, the system can only focus on alleviating the actual reason without considering the other possibilities [10], which improves the performance.

14.3 Cross-Layer Design, Adaptation, and Optimization

Up to this point, we have observed that under some circumstances, strictly layered architecture performs inefficiently. As discussed in Section 14.2.1, in order to overcome this major problem, increasing the amount of information flow between layers and re-organizing the processes according to their dependency on each other have been proposed. All these efforts lead to a novel concept called "cross-layer architecture." Hence, *cross-layer design* can be defined generally as follows: "Any kind of innovation on the traditional structure that blurs, changes, or even removes the boundaries between layers."

In the literature, there are numerous types of cross-layer designs in the frame of the definition stated above. Some of the designs only allow the information to flow upward and/or downward direction [15], whereas some of them are based on coupling of some of the layers [16] or merging some adjacent layers [10]. Even though these innovations are considered as solutions for some

[3] TCP uses slow start, congestion avoidance, fast retransmit, and fast recovery algorithms together to avoid and handle the congestion. The transmission is initiated with "slow start," which is based on a gradual increase of sending rate of the segments. This gradual increase is kept until a congestion is detected. In case of a congestion, TCP employs a special algorithm which slows down the sending rate of segments. When the congestion is cleared, TCP employs the slow start again and tries to attain maximum throughput based on the gradual increase by avoiding congestion [14].

problems, they come at the expense of different problems such as more complicated designs compared to the traditional one. Besides, blurring or removing completely the boundaries between layers in the traditional architecture causes the tasks defined explicitly for each layer to spread into other layers and become others' problems as well. In other words, violating the independence of layers introduces additional dimensions to the tasks of other layers. Consequently, optimization of the tasks is converted from a narrow (single layer) domain to a broader (multi-layer) domain.

Having a cross-layer architecture with optimization is not going to be sufficient for ultimate system design goal. The missing link in the chain is adaptation [8, 17, 18]. In Section 14.2.1, it is outlined that the status information of a wireless communication system needs to travel among the layers because of changing wireless channel conditions, network load, and Quality of Service (QoS). Allowing the status information to travel among the layers is a starting point to complete the chain [17,18]. Therefore, in the subsequent sections, the infrastructure for merging the concepts of cross-layer, adaptation, and cognitive radio will be discussed.

14.3.1 Cognitive Radio, Cross-Layer Design, and Adaptation

Cognitive radio is a radio that can sense, be aware of, learn, and adapt to the surrounding environment according to its inner and outer stimuli. These properties of cognitive radio take their places in the cognition cycle [19]. Overall cognition cycle can be seen as an instance of Artificial Intelligence (AI), since it encompasses observing, learning, reasoning, and adaptation.

Adaptation itself is a complex problem in the cognition cycle, because cognitive radio needs to take into account several input sources at the same time including its own past observations as a result of learning property. For instance, during its adaptation, cognitive radio needs to consider several requirements simultaneously such as user and application preferences, its own capabilities such as battery status, environmental conditions such as the availability of spectrum and propagation characteristics, and so forth. A compromise point, which can be regarded also as optimization, is tried to attain between these requirements. Note that some of the requirements fall into the tasks of specific layers in the traditional design. More explicitly, cognitive radio needs to consider QoS requirements, physical medium options as in traditional architectures beside some additional constraints such as battery consumption and past experiences. Therefore, one can conclude that, cognitive radio needs an overall adaptation that covers multiple layers with the aid of optimization.

Currently, there is no architecture that can meet all the aforementioned requirements of cognitive radio simultaneously. A fundamental reason behind that is the absence of any sort of controller and coordinator governing the overall adaptation process. The obligation of the presence of a controller and coordinator for complete adaptivity can be explained by an analogy. Since cognitive radio has AI capabilities, it is adequate to consider the most intelligent systems on the Earth: humans. Humans have aural, olfactory, tactile, taste, and visual sensors. These sensors help humans to perceive the surrounding environment. Humans are aware of themselves with the aid of inner sensors called as nerves. Humans are also equipped with very complex structures called "organs" to carry out vital operations. Each organ in the human body is physically isolated from the others such as the heart, the kidneys, and the liver. Interestingly, as in the layered architecture, some of the organs operate in algorithmic order such as digestive system. In the human digestive system, the intestines should wait for the stomach to operate. At the end, there is another structure that controls and coordinates every single organ in the human body: the brain. The brain is aware of both inner and outer world of the body via the nervous system and sensors. It gathers all the information from inner and outer world, processes and compares it with its past knowledge, chooses the best (or, in engineering terminology, optimum) decision, acts on, and observes the consequences for future usage. This procedure that the brain follows highly resembles the cognition cycle.

The human body analogy stresses that, a special structure, which has the capability of both controlling and coordinating, is essential in order to obtain a complete adaptive architecture. In cognitive radio domain, the counterpart of this special structure is known as "cognitive engine." Even though currently there is no formal definition of cognitive engine, it is agreed that cognitive engine is responsible for the overall adaptation and optimization process. However, a question arises automatically by introducing cognitive engine to the cross-layer design: What kind of an architecture should be adopted to include both cross-layer design and cognitive radio? Now, we seek for appropriate approaches, if possible, an answer to this question.

14.3.2 Cognitive Engine and Cross-Layer Architecture Design

As stated in Section 14.3.1, when a completely adaptive system is considered, the control and coordination of already defined layers have to be organized. The initial step is to establish the flow of information between each layer regardless of the levels of layers. In the earliest attempts toward cross-layer design, several layers were connected to each other in bi-directional way [17]. With the help of combinatorics, for an n-layered architecture ($n \geq 2$, $n \in \mathbb{Z}$), the number of single-direction flow must be defined is given by $\binom{n}{r}$ and $r = 2$.[4] Considering that the flow has two directions (upward and downward) as in [17] and there are already information paths between adjacent layers, the total number of new information paths to be defined becomes:

$$\mathcal{R} = (n-1)(n-2). \tag{14.1}$$

As can be seen in (14.1), a linear increase in the number of layers causes a quadratic increase in the number of new paths to be defined, which should

[4] This representation is known as binomial coefficient and given by $(n!)/((n-r)!\,r!)$.

be avoided. Besides, solely $\mathcal{R}$ paths are not going to be enough to attain a complete adaptation since pre-defined layers are not capable of converging to a complete adaptation. In addition, cognitive radio not only consists of pre-defined layers, but also includes several other sensors that need to be in connection with the adaptation process. This emphasizes that cognitive engine needs to form a sort of interface between the layers and sensors. In Section 14.3.1, when the cognition cycle was introduced, one of the abilities of the cognitive radio, learning from its past experiences, has been brought forward. Learning from past observations includes memory related processes. Therefore, apart from layers and sensors, cognitive engine needs to interact with different parts of the hardware. Finally, considering the evolution of cognitive radio, cognitive engine can be updated and modified.

By considering the aforementioned aspects, it is reasonable to think cognitive engine as a new layer that has connections with each layer, sensors, and hardware. This assumption, as a side-effect, also removes the quadratic behavior defined in (14.1) and allows one to facilitate the already defined layered architecture to some extent by converting (14.1) into a linear form.

One of the important characteristics of this new architecture is that it should preserve the previous achievements related to cross-layer organization and architecture. More explicitly, cognitive engine must take advantage of the present architecture rather than changing it entirely. The major impact of the cognitive engine on the cross-layer design is to remove the distance between layers on the edges. Consequently, a cognitive engine that is considered as a separate layer can be placed between layered architecture and several other peripherals such as memory and sensors. This is illustrated in Figure 14.2.

In a contextual model as shown in Figure 14.2, the operation of cognitive engine is extremely important. In this architecture, cognitive engine is "attached" to an already existing structure. Thus, the presence of cognitive engine should not create any problem to the layered architecture, since current layered architecture can already handle various issues very efficiently. What cognitive engine introduces is to spread the adaptation among all the layers, which cannot be achieved through traditional layered architecture. On the other hand, cognitive engine forms an interface for the available information coming from peripherals (such as memory and additional sensors) to improve the performance and adaptivity. Hence, it can be said that cognitive engine intervenes only when it is needed.[5] According to the situation, cognitive engine can take over the optimization process because of the necessity

[5] Actually, this behavior of the cognitive engine has also several counterparts in the human body analogy, which is discussed earlier. One of them is known as "involuntary (stereotyped) reflex actions." A person who touches a hot stove immediately pulls his/her hand back without "thinking." This response to the stimuli is handled by a mechanism called "reflex arc." Even though the brain is not involved with the first stage of the action (pulling the hand immediately), later on, the information "pain" is sent to the brain and the brain relates the actions and learns.

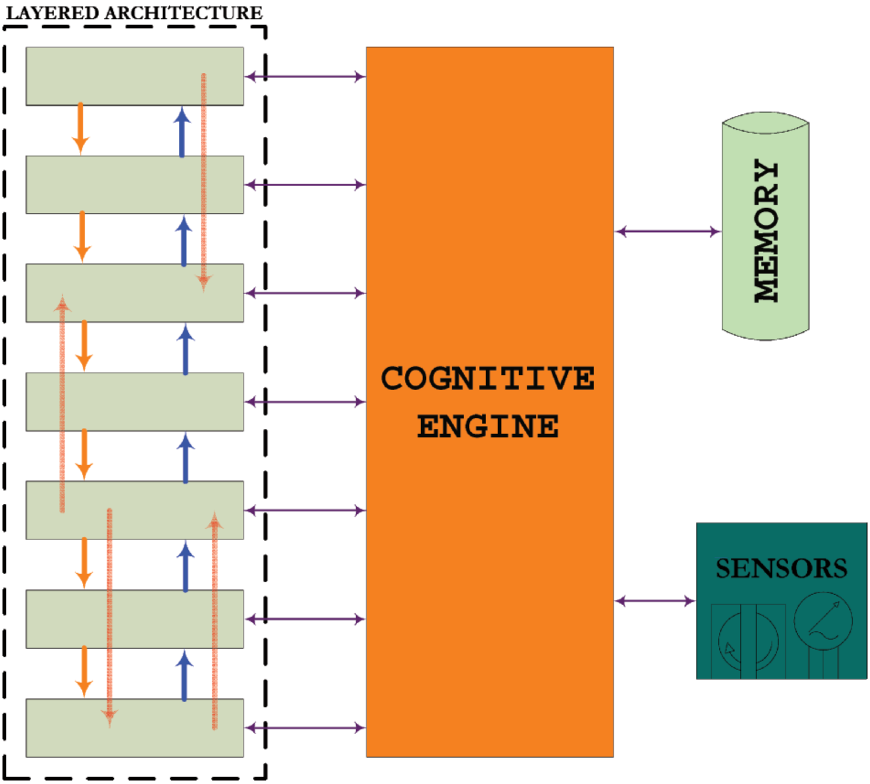


Fig. 14.2. A contextual model for cognitive radio. Cross-layer architecture and peripherals such as memory and sensors are all inter-connected through cognitive engine.

of the contribution of other layers. Similarly, cognitive engine takes action to by-pass some of the layers for the sake of optimization and/or speed or battery requirements.

Some illustrative examples – which do not contain any structure such as cognitive engine – for cross-layer adaptation and optimization can be presented for the sake of further comprehension before going deep into cognitive engine.

Illustrative Examples of Cross-Layer Adaptation and Optimization

Wireless channel possesses various characteristics. Multipath phenomenon introduces spreading in time, whereas due to the Doppler effect, the signal spreads in frequency domain. In addition to multipath propagation and Doppler spread, the transmission bandwidth is of great importance in understanding the characteristics of the wireless channel [8].[6] These concepts are the most prominent factors affecting the small-scale fading in wireless channels, which can cause erroneous data reception.

In order to be able to achieve a reliable communication over fading channels, channel coding is used to detect and correct possible errors. The essence of channel coding is to introduce redundancy into the data to be sent in order to restore it at the receiver side. There are several channel coding schemes such as block codes, convolutional codes, and turbo codes. One of the most important parameters in the channel coding is the coding rate. Coding rate is defined as $\mathbf{R}_c = k/n$, where k denotes the number of bits before channel coding and n represents the number of bits after encoding operation. Thus, $n-k$ is the total redundancy which is a measure of spectral inefficiency due to coding process [8]. Consequently, $\mathbf{R}_c = 1$ means no redundancy at all, in other words 100% efficiency. It is important to remember that even though $\mathbf{R}_c = 1$ promises 100% spectral efficiency, in case of an error, the data may not be recovered appropriately, which requires re-transmission meaning low data rate. We can exemplify channel coding with contemporary communication technologies such as Global Service for Mobile (GSM) and Global Packet Radio Service (GPRS). GSM uses 0.5 code rate for speech data [20], whereas GPRS uses four different code rates between 0.5–1 [21], according to the pre-defined channel quality schemes. Measuring the link quality consecutively allows GPRS to switch between different code rates to establish high throughput [21].

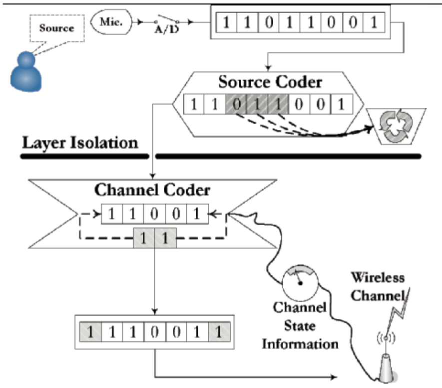
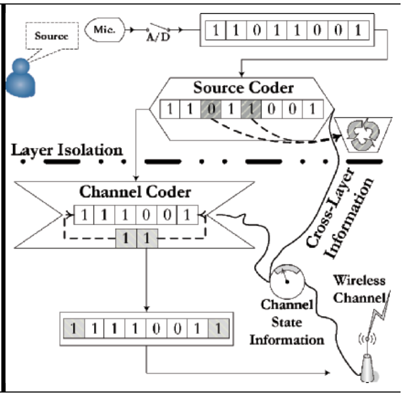
It is also possible to get higher data rates by preserving the spectral efficiency. This is achieved by switching to higher order modulations in case of having a good link [8]. In Enhanced GPRS (EGPRS), two different modulations are available with different coding rates [8,21]. After link measurements, the communication system adjusts itself with the aid of Adaptive Modulation and Coding (AMC). If the link quality is good, then EGPRS can switch from lower order modulation (Gaussian Minimum Shift Keying [GMSK]) to a higher order modulation (8–Phase Shift Keying [PSK]) with reducing the channel coding power $((\mathbf{R}_c \to 1) \equiv (n \to k))$ [22].

EGPRS also provides a different cross-layer application which includes again the collaboration of data link and physical layer. Instead of AMC, this time, a technique, which is known as "incremental redundancy" (a method in Type–II Hybrid Automatic Repeat Request (ARQ)[7]), is used [21]. In incremental redundancy, first, the data is sent through the channel with a weak coding power ($\mathbf{R}_c$). If no error occurs, a high bit rate is achieved. Unless there is an erroneous reception, which is going to be notified by an ARQ scheme

[6] Note that if the transmission bandwidth is less than the coherence bandwidth of the channel, the effects of signal spreading can be neglected.

[7] It is named as Type-II, because it stores the erroneous packet whereas Type-I discards it. The other method of Type-II Hybrid ARQ is known as "chase combining."

Table 14.2. Separation theorem and JSCC.

<table>
<tr><th>Traditional architecture</th><th>Cross-layer architecture</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>Separation theorem [34]</td><td>JSCC</td></tr>
</table>

to the transmitter side, the coding power is never increased. In case of an erroneous packet reception, the coding power is increased step by step until the reception becomes error-free [23].[8]

Up to this point, some applications that facilitate the collaboration of the layers close to each other have been outlined. There are also other applications that can use cross-layer approach for the layers residing at the edge of the traditional architecture such as the collaboration of application and physical layer. A very well-known example of these sorts of applications is known as Joint Source–Channel Coding (JSCC).

In multimedia applications the notion of perceptual quality is of vital importance. Therefore, the main purpose of the multimedia transmission can be defined as to obtain the best perceptual quality. However, when the transmission is carried out over wireless channels, numerous constraints (such as fading, shadowing, interference, and so on) that affect the perceptual quality must be considered as well (see Table 14.2). Especially video transport through wireless channels is one of the prominent applications for JSCC. A JSCC that is aware of the wireless channel[9] improves the performance significantly [25, and references therein]. The main purpose of any type of communications is to make sure that the information that is intended to be sent arrives at the receiver side. Otherwise, by definition, there is no communications, which means that there is no need to think further. Therefore, a link must exist between transmitter and receiver. In terms of JSCC for video transportation, we cannot talk anything about the source coder unless a link exists. If there is a link, then

[8] The working principle of incremental redundancy method can be considered as the reciprocal of slow start method for TCP, which was introduced before.

[9] Note that some JSCC approaches do not assume the presence of any sort of channel knowledge [24].

both source and channel coder get into the picture. In order to protect the video information from link degradations, channel coding is applied. However, this comes at the expense of reducing the efficiency of the bandwidth. If the video information is not protected, it may not arrive or may arrive at the receiver side but with unrecoverable errors, which causes the video to lose its intelligibility. Source encoder removes redundancies from the video and prepares the encoded scheme in an error-resilient way [26]. Before the delivery, the information is packed as video frames. At this point, bit rate of each frame is controlled by the rate controller according to the Channel State Information (CSI), which is provided by lower layers such as physical layer. At the final stage, channel coder adds redundancies depending on the link qualifications. Hence, in JSCC, the problem reduces to the optimization or allocation of the total bit rate between source and channel coding operation [27].[10]

Cross-layer design (and therefore, cross-layer optimization) manifests itself in a more complicated way for wireless ad hoc networks compared to the previous examples. Lack of communication infrastructure brings several other issues into the picture. Unlike the systems that have infrastructure, in ad hoc networks, each node should consider very challenging tasks in several layers such as routing in network layer due to the dynamic topology; scheduling for wireless channel access in Medium Access Control (MAC) layer; and power control in physical layer [29]. Security can also be added on top of these considerations [30]. Optimizing each layer one by one (as in traditional architecture approach) may end up with high network throughput, but this gives rise to several other considerations such as unfair transmission rates for some of the nodes in the network [31]. Therefore, the overall optimization must include throughput and resource utilization, congestion control [32, 33], scheduling [33, and references therein], and efficient routing [31], which are established in different layers in traditional architecture.

After solidifying the cross-layer concept with several examples including interaction of closer layers, distant layers, and multiple layers, it is appropriate to introduce some of the adaptation parameters that are being used in contemporary communication systems.

14.3.3 Some of the Adaptation Parameters That Are Popularly Used in Contemporary Communication Systems

Although the emergence of cognitive radio emphasizes the term "adaptivity" stronger than the previous communication technologies, we must note that the evolution of wireless communications has already been going toward adaptivity. It is not hard to see this reality when the progression of wireless communications in time is reviewed. When we look at the whole wireless

[10] As stated in [27], this sort of approach of JSCC is limited only to the source and channel encoders. The actual optimization includes more detailed investigation such as power considerations [28].

communication history, we can see that the early technologies or standards use fixed schemes in the system design such as fixed resource allocation, fixed frequency assignment, or fixed average signal quality for the receiver designs, and so forth [8]. Fixed scheme provides a simple design architecture, but it also comes at the expense of sub-optimum performance because of similar reasons discussed about traditional and cross-layer design architecture, in Section 14.2.1. This trade-off has been realized and tried to overcome by flexing the design, which was led by adaptation.

As stated in Section 14.3.1, there are numerous examples of adaptive wireless systems that have already been used. Especially recent standards such as WiMAX include many adaptation capabilities. However, these recent efforts as well as the previous adaptation methodologies focus on individual layers and look at the problem from a narrower perspective compared to cognitive radio. This is not surprising, because global adaptation requires perfect knowledge about all the parameters in every level including the relationships between them. But, cognitive radio, by its very definition, aims global adaptation. In global adaptation, some of the parameters conflict with each other for specific optimization criteria. Therefore, cognitive radio must be aware of what to change for adaptation and how those changes affect the system.

There are adaptation forms that serve to attain the same goal in wireless communication systems. One of the very well-known examples of these sorts of adaptation methods is to maintain Bit-Error-Rate (BER) at a certain level (constant BER). In many adaptive wireless communication systems, maintaining the desired BER level is established by increasing the power level or applying Forward-Error-Correction (FEC) techniques [8]. Increasing the power level has several side-effects such as faster battery consumption, increase in interference, and so forth. Similarly, applying FEC techniques reduces the efficiency of the use of the bandwidth. In this case, exploiting the options of attaining the same goal requires the evaluation of the side-effects of each path.

Typically, when upper layer requirements are also taken into account as in multimedia transmission, the global adaptation encompasses several constraints farther such as delay, perceptual quality, and so on. At this point, cognitive radio chooses one available option that takes it to the global optimum, if possible. However, introducing more constraints into the optimization process increases the probability of conflict between constraints. When an application that requires both high data rate and a constant BER is considered, applying adaptive modulation will cause the two goals to conflict, because maintaining BER under a desired level is possible with reducing the order of the modulation. This automatically reduces the data rate, under the assumption that the other limitations are constant. Conversely, under the same conditions, a high-data rate communication requires higher order modulation, which increases BER.

As we see, introducing even one constraint complicates the problem. Thus, cognitive radio needs to consider the trade-offs mentioned above comprehensively, since there are many parameters to be adjusted. Table 14.3 provides

Table 14.3. Some of the writable parameters for adaptive wireless communication systems.

Layer	Parameters
RF	Antenna powers Dynamic range Pre-distortion parameter Pre-equalization parameter
Physical layer	Transmit power Digital modulation order Carrier frequency Operation bandwidth Processing gain Duty cycle Waveform Pulse shaping filter type FFT size (for OFDM) Cyclic prefix size (for OFDM)
Data link layer	Channel coding rate Channel coding type Packet size Packet type Data rate Interleaving depth Channel/Slot allocation Carrier allocation (in multi-carrier systems) MAC scheduling algorithm Handover (Handoff) Number of slots
Network	Routing algorithm/metric Clustering parameters Network scheduling algorithm
Transport	Congestion control parameters Rate control parameters
Upper	Communication modes (simplex, duplex, etc.) Source coding Encryption Service personalization

some of the currently used popular adaptation parameters with respect to the layers. In the frame of global optimization, cognitive radio needs to consider these and many others which will appear jointly in the future.

14.4 Cross-Layer Optimization

Although cognitive engine conceptually looks like the missing part of overall cross-layer adaptation and optimization, in the implementation stage, the real challenge is to construct the formal methods that are going to run on cognitive engine. Currently, there is no unified mathematical model that can handle each and every one of the capabilities mentioned above. Consequently, cognitive engine needs to have –at least until a unified model appears –several mathematical models to cope with different aspects of the cognition cycle.

Fortunately, there are very successful individual formal models that can operate in particular domains of the cognition cycle such as learning, reasoning, multiobjective optimization, and so on. Specifically, in this section, we are going to investigate how a MOP can be handled via available formal models. Before getting into the details of multiobjective optimization concept, it is appropriate to introduce the optimization problem in a general context.

14.4.1 Optimization Problems

No matter how complex the optimization problems are, the main goal of all of them is the same: "to find the best solution among available set of solutions under limited resources." In order to be able to visualize this definition, we can consider a very well-known example called as "0–1 knapsack problem." According to the story, a hiker wants to put several items (such as cans of food, bed roll, and so on) into his bag, but he cannot carry more than 70 lb [35].[11] He wants to find "the best" combination of the items which weight as close as possible to 70 lb according to the relative value of each item determined by himself. For instance, he may think of cans of food as more valuable than bed roll, because food is essential for survival even though its price is less than that of the bed roll. More explicitly, in 0–1 knapsack problem, the limited source is the bag (or the hiker) that cannot carry more than 70 lb, whereas the best solution corresponds to the combination of the hiker's favorite items that weight as close as possible to 70 lb without exceeding it.

Establishing a 'reasonable' solution requires the optimization problems to have a formal model. In a general optimization problem, the formal model relies on defining the following three items:

- **Variables:** They comprise the essence of the problem via the mathematical relations between each other.
- **Objective Function:** It represents the concept that is going to be optimized. It can be univariate or multivariate depending on the structure of the problem. The purpose of the problem corresponds to obtaining either maximum or minimum value of this function.

[11] The problem is called as 0–1 knapsack problem, because the hiker either chooses an item (which is represented by "1") or leaves it (which is represented by "0").

- **Constraints:**[12] As the name implies, these are the limitations by which the objective function is going to be optimized. Along with the domain of the objective function, it defines the feasibility region, which means that any probable solution outside this region is going to be ignored.

Having the items listed above on our hands, the statement of the optimization problem can be written as follows:

$$\begin{aligned} &\text{Find } \mathbf{x}^* \text{ which} \\ &\text{minimizes } f(\mathbf{x}) \\ &\text{subject to } c_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \ldots, r \\ &\text{with } m_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, h, \end{aligned} \tag{14.2}$$

where $\mathbf{x}^* = [x_1, x_2, \ldots, x_n]^T$, $(\cdot)^T$ denotes the transpose operation, f represents the objective function, c_i and m_j denotes the constraints.[13]

If one wanted to apply (14.2) to 0–1 knapsack problem, the formal model would be as follows:

$$\begin{aligned} &\text{Maximize } f = \sum_{i=1}^{n} b_i x_i \\ &\text{subject to } \sum_{i=1}^{n} a_i x_i \leq 70 \\ &\text{with } x_i \in \{0, 1\}, \end{aligned} \tag{14.3}$$

where b_i denotes the relative value of the i-th item according to the hiker, a_i is the weight of the i-th item.

Even though (14.2) gives the formal statement of a general optimization problem, there are numerous types of different optimization problems. Since there is no unified and comprehensive method available yet as stated at the beginning of Section 14.4, classification of the problem is extremely important, because the approach (or the strategy) for obtaining the solution depends on the particular class (or category). Therefore, we can briefly glance at classifications of optimization problems.

14.4.2 Classifications of Optimization Problems

Generally, the classification of optimization problems is divided into three coarse categories as follows:

1. existence of constraints,
2. structure of the variables,
3. equation types of objective function and/or constraints.

Let us now briefly introduce each classification item.

[12] Note that, some types of optimization problems may not require a set of constraints [36].

[13] In the literature, sometimes m_j is omitted.

Existence of Constraints

This sort of classification of optimization problems has already been introduced previously as constrained and unconstrained optimization problems. Even though there are numerous types of unconstrained optimization problems (and a significant amount techniques devoted), in cognitive radio domain, most of the adaptations require at least one or two constraints such as battery level, channel state, and so on. Therefore, most of the time the optimization problems related to cognitive radio fall into the category of constrained.

Structure of the Variables

Structure of the variables determines the domain of the problem in which it is going to be investigated. There are several structure categories such as continuous–discrete, deterministic–stochastic, and so on. In cognitive radio domain, for instance, in the optimization problem which includes rate controller as mentioned in Section 14.3.2, the number of bits in a frame is of discrete type. Conversely, channel state information based on Received Signal Strength Indicator (RSSI) is of both continuous and stochastic type.

Apart from the categorization above, a different categorization for this topic is also possible. A problem can be considered as combinatorial or variational depending on the cardinality of the set of variables. In combinatorial category [37], the solution set is finite.[14] In variational category, basically, the solution set is infinite [38]. Especially combinatorial optimization problems are very important for cross-layer adaptation, as mentioned in Section 14.3.2 while discussing AMC.

Equation Types of Objective Function and/or Constraints

Formal aspects of the objective function and/or constraints are extremely important to treat an optimization problem. This type of classification is slightly different from the others, since particular categories define particular mathematical tools. The following categories are generally referred in the literature according to the types of the equations within the problem: linear, quadratic, polynomial, non-linear, and sparse.

The classification of optimization problems introduced above provides just an idea about the concept. Apart from the classification above, there can be defined several other categories such as “number of variables in the objective function.” The formal and detailed classification of the optimization problems and relevant approaches are out of the scope of this chapter. Interested readers, who want to gain more information about the classification, may refer to [39–41].

[14] Knapsack problem is a combinatorial optimization problem, since the hiker can establish his favorite combination of items over a finite set.

Cognitive Radio and Optimization Problems

Two fundamental concerns behind the conceptual model of cognitive radio are (i) to provide backward compatibility and at the same time (ii) to improve an already working architecture. Since the layered architecture is going to be kept to some extent and this approach is going to be combined with overall adaptation, we automatically face to a different type of major classification in optimization problems, which is known as MOPs. Before investigating MOPs, it is worth mentioning another approach in the literature, for cross-layer architecture, which comprises a mid-step between single objective function and multiobjective function.

This approach considers the use of the interaction of multiple layers to optimize only one of the objectives. The main idea behind this approach is to take advantage of the multi-modal structure of the current communication standards with the aid of combinatorial optimization. Recall that in EGPRS, the number of possible modulation schemes is limited to two, as stated in Section 14.3.2. This means that the contribution of the physical layer to the problem in terms of modulation is a set which has only two elements. Similarly, in a IEEE 802.11a system, the modulation set has four different modulation options, which are Binary PSK (BPSK), Quadrature PSK (QPSK), 16-Quadrature Amplitude Modulation (QAM), and 64-QAM. As in modulation, channel coding can be treated in the same way. If this approach is followed for every possible layer, at the end, a comprehensive set which is composed of elements formed by the Cartesian product of every possible parameter set across the layers is obtained. In other words, the Cartesian product of every possible parameter set forms the solution set in which "the best" is going to be sought for, as illustrated in Figure 14.3. Although the Cartesian product can form a set that has a large cardinality, it is still finite [42]. Of course, this reasoning gives rise to an obvious question: Who is responsible for the optimization? This question can be answered in two ways: (I) without cognitive engine and (II) with cognitive engine. For (I), there are several approaches [42]:

- *Bottom-up approach:* The lower layers try to save the upper layer from the losses. This approach cannot provide an overall optimization, since it is going to fail in multimedia applications.
- *Application-centric approach:* In contrast to bottom-up approach, this approach gives the priority to the application layer to have the control on the optimization process. This approach cannot support the overall optimization either, because the response time of the application layer to sudden changes in the lower layers (especially in the channel) is not sufficient.
- *MAC-centric approach:* After seeing that pushing the responsibility towards edges comes at the expense of sub-optimality, this approach tries to keep the control of the optimization process around the center of

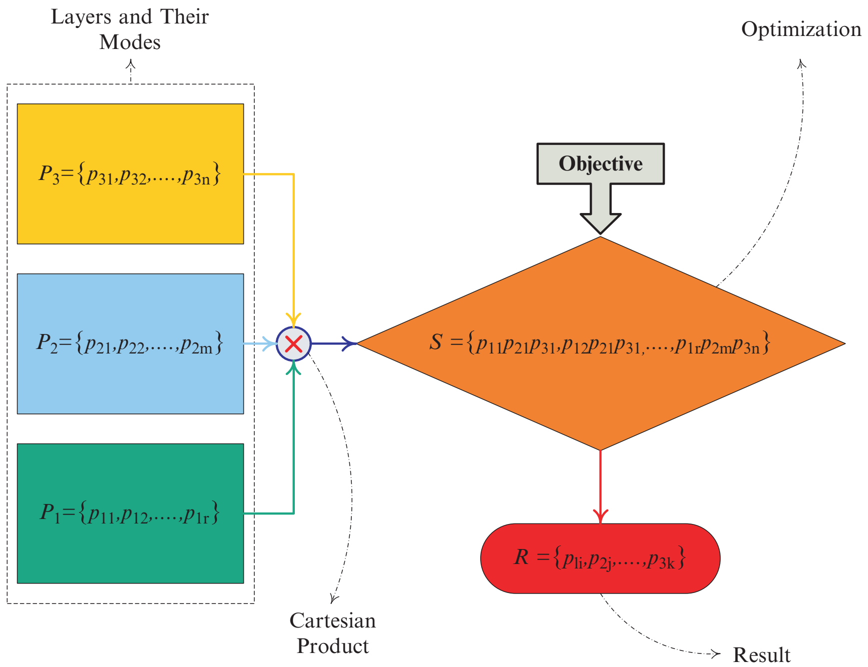


Fig. 14.3. The mid-step between single and multiobjective optimization. The design still has one objective, but, the variables are coming from different layers, which means that the type of the variables are not unique, as introduced in classification of optimization problems.

the stack. It gathers the information from the upper and lower layers and decides upon its own criteria. The major drawback occurs in JSCC.[15]

For (II), the answer is easy: "cognitive engine." Since we have not provided a substantial answer to the optimization in terms of cognitive engine, one may ask how does cognitive radio know which combination in the solution set provides the optimum solution? As being aware of its inner and outer world, cognitive engine defines the constraints according to its inner and outer environment. In one scenario, cognitive engine may realize that it is running out of battery. This automatically affects the selection of the best combination and forces cognitive engine to find a power efficient one. Conversely, as soon as the device understands that it is plugged into the electric outlet, cognitive engine immediately drops the constraint of power efficiency and defines another constraint accordingly.

[15] In [42], there is another approach proposed for (I), which is called "integrated approach." Interestingly, the key point of this approach is stated as learning and classification techniques [43].

Combinatorial approach that takes advantage of multi-modal standards draws us near our ultimate propose, which is to get solutions for MOP. However, we have to be aware of that MOPs are of different formalization from that in (14.2), since the number of objective functions is more than one. Nevertheless, (14.2) can be modified to reflect the multiobjective structure while maintaining the general frame of the optimization concept as follows:[16]

$$\begin{aligned} \text{Minimize } \mathbf{F}(\mathbf{x}) &= [\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}), \ldots, \mathbf{F}_n(\mathbf{x})]^T, \\ \text{subject to } c_i(\mathbf{x}) &\leq 0, \\ g_j(\mathbf{x}) &= 0. \end{aligned} \tag{14.4}$$

In multi-dimensional cases, the solution for optimization problem becomes harder.[17] Considering the ultimate boundary, which is time, the challenge becomes clearer. Recalling from the previous sections, in cognitive radio, it is desired to have a complete adaptivity across all the layers in a short period of time since the wireless channel is highly dynamic. Apart from that, formalizing the requirements of each layer depending on stochastic events (such as channel conditions) is extremely difficult. We will investigate the challenges of cross-layer optimization in detail later. First, we outline the statement of the problem formally.

Multiobjective Optimization Problems and Related Approaches

As their name implies, MOPs occur in a single design having multiple objectives which usually contend against each other. If MOPs are also considered in terms of the definition given at the beginning of Section 14.4.1, it is seen that the challenging part is the allocation of resources between contenders.

We have seen some examples explicitly referring to MOPs such as the relation between channel coding power and spectral efficiency. Recall that, it is impossible to have 100% spectral efficiency and maximum data protection simultaneously. When MOPs are considered, aiming maximum (or minimum) generally loses its meaning. We say "generally," because, in typical MOPs, it is extremely difficult to find a solution that can maximize (or minimize) each individual objectives simultaneously. Instead, the term solution corresponds to a set which is composed of some alternatives representing the trade-offs between objective functions. However, there may be some extreme cases that

[16] Note that, mathematically, maximization of any function f is equivalent to minimizing $-f$. Therefore, for the sake of brevity, every optimization problem can be defined only through minimization and vice versa.

[17] Again, the concept "harder" can also be defined formally. As a special form of the problem of sum-of-subsets, knapsack problem is classified as Non-deterministic Polynomial-time Complete (**NP**-Complete) in connection with the decision problem (*Entscheidungsproblem*) [44] which is believed that coined by David Hilbert. For the relations between optimization problem and computational complexity, [45] can be referred. For proofs and further discussion please see [46, 47].

lead to a solution rather than a set of alternatives [48]. These extreme cases are known as utopia [49]. Then, that solution which satisfies the maximization (or minimization) of each individual objective simultaneously is called optimal.

Ignoring extreme cases for MOPs and focusing on typical ones, we want to consider what to do when the decision time comes. At that time, an element has to be drawn from the set of alternatives for a decision to act on. At this point, a new notion called *preference* gets into the picture. It is important to note that the notion of preference distinguishes MOPs from Global Optimization Problems (GOPs). GOPs search for a single *solution*, whereas MOPs are based on getting the best *compromise* between multiple objectives in a set of alternatives. Hence, for MOPs, trade-off approach is adopted rather than a search.[18] According to the preference of the decision-maker, the notion of optimality turns into another concept called *efficiency*. Efficiency is the assessment of an element in the set of alternatives, which is chosen by decision-maker according to his preference. There are several definitions of assessment of the efficiency of an element in the set of alternatives. Before explaining the relationship between efficiency and its assessment, we need to introduce the notion of *dominance*. In MOPs, dominance expresses the preference level of the elements to each other. If one of the elements in the set, say x_2, is less preferred to another element, say x_1 (because x_1 provides better values for each individual objective function simultaneously), then, it is said that x_1 dominates x_2. Now, putting all the things together, we can assess the efficiency of the choice of the decision-maker. Edgeworth–Pareto optimal (or Pareto optimal [50], efficient solution, nondominated solution, a noninferior, or functional efficient solution [51]) is one of the most important assessment definitions of the efficiency of the element of interest. Informally, Edgeworth–Pareto[19] optimality can be defined as follows:

Definition 1 (Edgeworth–Pareto optimality) *In the set of alternatives (trade-offs or nondominated set), if there is no other element that can dominate the element chosen from the set, the element chosen from the set is called Edgeworth–Pareto optimal.*

In order to be able to solidify the concepts mentioned up until now, it is appropriate to examine Figure 14.4. Figure 14.4 illustrates a simple, two-objective function optimization problem which has a convex solution set or feasibility region. The horizontal and vertical axes denote the objective function 1 and 2, respectively. The solution set is represented with diagonally

[18] In the literature, "preferences" belong to decision-maker. Decision-maker is the one who is responsible for the final decision [48]. Note that, even in mid-step example introduced in Section 14.4.2 (see also Figure 14.3), the *responsibility* must be taken by someone.

[19] In the literature, Pareto optimal has a vaster usage than Edgeworth–Pareto optimal. Edgeworth is the one who proposed the correspondent term of optimum for multiobjective optimization problem, whereas Pareto is the one who generalized it. For further historical discussions, please see [52, 53].

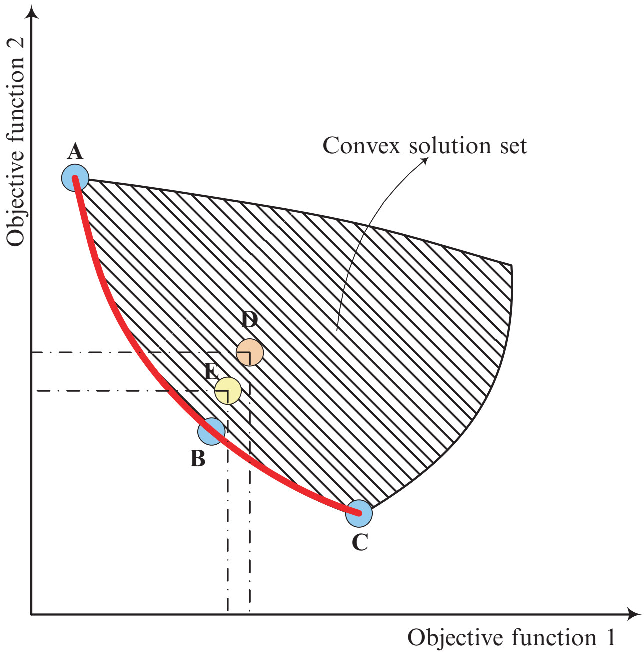


Fig. 14.4. Convex solution set, efficiency, nondominated set, and Edgeworth–Pareto optimality are represented. The circles labeled as "**A**, **B**, **C**, **D**, and **E**" denote points inside the set.

shaded, convex shape. Recall that this region is determined by the constraints peculiar to the problem. The decision-maker is allowed to choose any point within the convex solution set including its border. As stated above, the assessment is going to be established by the notion of dominance. In Figure 14.4, point **D** is being dominated by point **E**, because point **E** provides lower values for both objective functions 1 and 2, simultaneously [46, 48]. This fact can be seen through the area bounded by two dashed lines originating from point **D** to the axes. Similarly, the relationship of the dominance between point **E** and point **B**; due to the property of transitivity, the relationship between point **D** and point **B** can be seen as well. Conversely, the curve $\widehat{\mathbf{ABC}}$ denotes the nondominated set, since no element can be found in the convex solution set which can provide lower values for both objective functions 1 and 2, simultaneously. Thus, depending on the preference of the decision-maker, any of the elements laying on $\widehat{\mathbf{ABC}}$ is considered as Edgeworth–Pareto optimal or efficient.

As a final remark, it must be stated that the mathematical definitions of efficiency, inefficiency, Pareto optimality, and dominance have slight differences between each other. Since these are out of the scope of this section, we will not discuss them here. However, the readers who are interested in may refer to [46, 48–51, and references therein].

Related Approaches

As mentioned previously, MOPs are complicated structures by their nature. Number of objective functions and vast variety of the constraints are two prominent factors that increase their complexities. Besides, there are some newer concepts that are not in single objective optimization problems such as decision-maker and preferences. These complexity factors and newer concepts together lead to have a different classification from the one introduced in Section 14.4.2.

MOPs are categorized as follows depending on when the preferences of the decision-maker involve with the algorithm [49, 51]:

1. ***No articulation of preferences:*** In this sub-category, the algorithm totally ignores the decision-maker before and during its run. However, after the algorithm ends, it strives to provide the whole feasible set to the decision-maker.
2. ***A priori:*** This approach assumes that the algorithm to be used has the knowledge of the preference of the decision-maker before it runs.
3. ***Progressive (interactive):*** In this one, unlike 2, there is no chronological order between the preference of the decision-maker and the initiation of the relevant algorithm. The decision-maker can provide its feedback during the operation of the algorithm. The algorithm provides a candidate solution to the decision-maker and waits for the response of the decision-maker. If the candidate solution is not accepted by the decision-maker, then algorithm strives to find a better candidate until it is accepted.[20] A further sub-classification of this category is also possible in terms of how the preferences expressed during the procedure as follows [55]:
   - target values,
   - ranking of alternatives or objectives,
   - other than above.
4. ***A posteriori:*** Unlike 1, these sorts of algorithms have the capability of narrowing down the solution space to Pareto set. The essence of these sorts of algorithms can be summarized as "generate-first-choose-later."

As discussed before, due to lack of a unified method, we need to categorize the problems so that we can apply specific tools to specific classes. Table 14.4 tabulates some of the methods available.

If the list of taxonomy above is examined in detail, it can be observed that the involvement of the decision-maker with the process needs formalization

[20] An application of such an interactive optimization procedure and the relevant flow chart is presented in [54].

Table 14.4. Multiobjective optimization problems, their classifications, and relevant solution methods.

Sub-classes	Solution methods
No preferences	Global criterion Achievment function Compromise function Objective sum Minimax method Nash arbitration Objective product Ideal distance minimization Maximal effectiveness principle
A priori preferences	Global criterion method Weighted sum Lexicographic method Weighted Tchebycheff Exponential weight criterion Weighted product method Goal programming Bounded objective function method Physical programming Multiobjective decomposition
Interactive preferences	Hierarchical decomposition method STEM method Multiobjective graph theory Method of constraints Parameter space investigation method Random search method Vector-relaxation method Interactive ϵ-grid method Method of local improvements Pareto boundary maps method
A posteriori preferences	Physical programming Normal boundary intersection method Normal constraint method Dynamic multiobjective programming Reachable set method Piecewise linear approximation method Genetic algorithms

as well. Then, how can we put the preferences of the decision-maker into the mathematical model we developed? The answer to this question, again, comes from the roots of the theory, which is economics. The function which represents the preferences of the decision-maker is called preference function or utility function.[21] With the aid of utility function, the formal model can be applied for the models which require decision-maker's feedback.

According to the taxonomy of MOPs and Table 14.4, we see that for "no articulation of preferences," the decision-maker is being dictated by the algorithm, which means that decision-maker has no control on the process [49, 51, 55]. When we consider cognitive engine, it looks a bit controversial, because whatever the solution method is chosen, it is going to run on cognitive engine. This problem arises from the fact that proposed solution methods are implemented on a different platform such as computer. Conversely, for cognitive radio, the problem, the solution, and even the decision-maker are all in the same platform. This implication proves that cognitive engine must have different units which govern the separated processes such as running the algorithm, evaluating the preferences, and so on.[22]

14.4.3 Challenges for Cross-Layer Optimization

Before integrating cognitive engine into the traditional architecture, we have to pay more attention to several aspects of cross-layer design and optimization. First and foremost, MOPs are naturally challenging due to the number of objectives involved with the problem. It is clear that the less the number of objectives, the less complex the problem. In addition, since we want to maintain the layered architecture to some extent, we must be aware of that each layer has its own design criteria. Besides, we seek for a formal model that encompasses all the layers, variables, constraints, and even objectives which are defined on different domains [18]. For instance, minimum bit-error-rate and no congestion are two separate objectives for physical and network layer, respectively. Thus, formalizing these different objectives together and putting them onto a common mathematical platform are very challenging problems. Furthermore, in spite of the fact that some solutions for MOPs have really fast convergence rates, for wireless devices that operates in very dynamic environments, the delay requirements are very tight. Another dimension of delay bottleneck challenges us in sharing the information through layers [9,18], sensors, and other peripherals of cognitive radio. In order for the solution to work appropriately, all the information should be available to cognitive engine before the optimization process starts. As a direct consequence of number of

[21] In economics, the concept of "utility" (or satisfaction) comes from the basic consumer–entrepreneur relationship. In a system that includes both consumer and entrepreneur, the purpose is to attain maximum utility (or satisfaction) for the consumer and maximum profit for the entrepreneur [56]. Morover, including/removing uncertainty (or risk) into/from the system causes this function to be named utility function/value function, respectively [48]. Nevertheless, the term preference function can be used for both risk and risk-free systems.

[22] Interestingly, current computers are designed based on von Neumann architecture, which has separate control, calculation, memory, input and output units [57]. Furthermore, this design operates as Universal Turing Machine.

objectives and delay requirements, solution for MOPs themselves may need an optimization since it consumes resources of the device. In other words, calculating the optimum allocation of a resource consumes the other system resources as well. Finally, it is known that cross-layer design can cause loops due to some interactions of layers bringing forward another challenging notion to be considered, which is known as stability [58].

At this point, we need to address another concern about cross-layer design and optimization. Before the concept of cognitive radio and therefore cognitive engine were introduced, one of the biggest challenges for cross-layer architecture had been that: "who has control? [18]"[23] With the emergence of cognitive radio, we can say that this question becomes obsolete.

Throughout Section 14.4.2, we expressed that the concept of decision-maker lies on the heart of MOPs along with preferences. We can say the same thing for cognitive engine in cognitive radio architecture. However, we also state that some of the problems have already been solved by the traditional architecture very efficiently. Therefore, in cognitive radio architecture, it is very reasonable to avoid MOPs unless they are needed. In other words, unless the intervention of cognitive radio is inevitable, cognitive engine does not need to get involved with the optimization process.

14.5 Further Notes

Up to this section, we outlined the methods to solve optimization problems for cognitive engine. However, it is worth mentioning that there are two very important accessories, which assist cognitive engine in solving the optimization problems: game theory and neural networks.

Our main focus was the optimization of the cross-layer architecture for a single device up to this point. One can suspect that optimizing one device in a network may not end up with a fully optimized network. Actually, this can easily be seen by using the same notions of multiobjective optimization discussed at the beginning of Section 14.4.2. In a network in which each node is trying to optimize its own objectives, a conflict between each node for several resources is highly probable. This phenomenon manifests itself in ad hoc networks very clearly [32]. Therefore, individual optimizations may need to be established for the sake of optimization of the network. Of course, these sorts of optimizations are of different type compared to individual optimizations, since there are two nested systems to be optimized: individual optimization and network optimization. Meanwhile, as in MOPs, we need to refrain ourselves from using the term "optimization" under those circumstances. Instead, the term "common satisfaction" may be much more appropriate. Then, the crucial question becomes: how do we attain a common satisfaction? In order to answer this question, we are going to review game theory briefly.

[23] Recall that, this question is investigated in Section 14.4.2 in a different form.

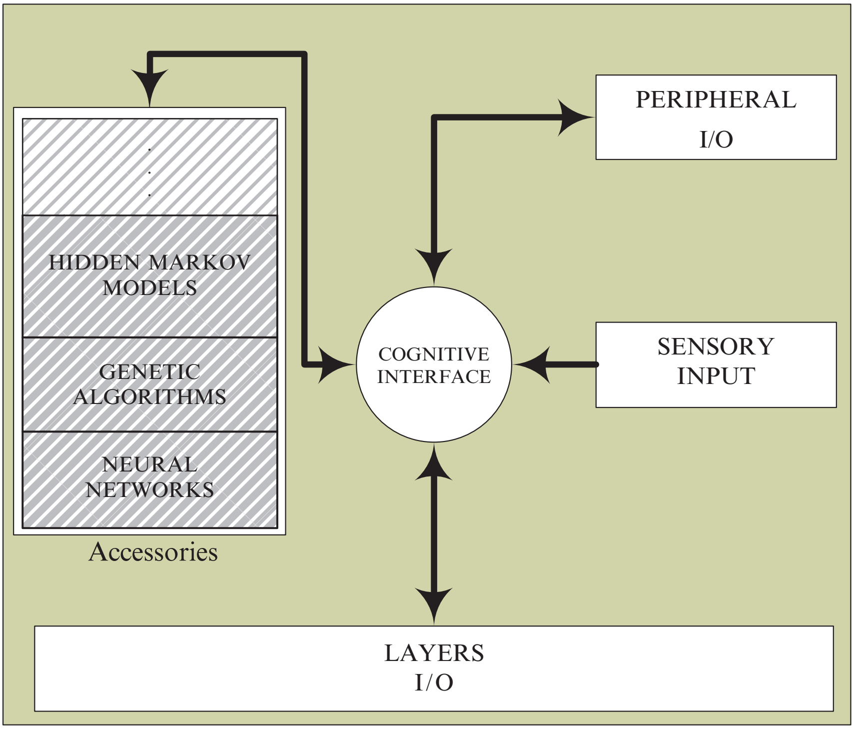


Fig. 14.5. A basic cognitive engine structure. Note that the construction highly resembles von Neumann architecture [57] because of the simplicity in the design simplicity.

Foundation of game theory is based on economics. The main idea behind the game theory is that, as defined in [56], the quantification of the problem of the rational behavior. The theory is constructed first for an isolated economical system, which is inspired from Robinson Crusoe.[24] Then, by introducing several individuals more, a social economy concept is introduced. Each individual acts rationally to attain its own satisfaction. Each individual develops a strategy depending on the system. Then, as in MOPs, each strategy is defined by a utility function which is tried to be reached to a *static equilibrium* rather than an "optimum" [56].

With this new perspective of rational behavior, many engineering problems in which there are several participants can be formalized and treated. In the

[24] As a historical note, it must be stated that the idea of rational behavior of an isolated human first appears with Abu Bakr Ibn Tufail (known as Abubacer in Latin) who was the author of "Hay bin Yakzan" (Hayy Ibn Yaqzan), the predecessor of Robinson Crusoe. In fact, the name of the hero of the story comes from another story with the same name written previously by a Turkish philosopher İbni Sina [59], who is known as Avicenna in Latin.

literature, related to wireless communications, there are also several studies that apply game theory to solve complex problems such as random access and power control [60].

Similar to game theory, neural networks must be addressed as another assisting tool for cross-layer adaptation and optimization. Stemming from their parallel processing power, neural networks are one of the essential tools for cognitive engine to cope with several problems. Inspired by the operation of nervous systems of biological organisms, neural networks have the capability of generalization, familiarity recognition, categorization, error-correction, and time sequence retention [61]. As can be seen, such properties can assist cognitive engine in many different areas including cross-layer adaptation and optimization. There are several applications related to amalgamation of neural networks and cross-layer design [62,63]. Furthermore, we begin to see that the intelligence governing and/or assisting the protocol process is being embedded into cross-layer structure through specialized layers called "cognitive layer," as well [64].

In light of this discussion, a basic cognitive engine structure can be as in Figure 14.5.

14.6 Conclusion

In this chapter, we tried to elucidate the entangled relationship between cognitive radio and cross-layer architecture. The rise of cognitive radio implies an inherent cross-layer design by its very definition including self- and environmental-awareness and adaptation. In parallel, the efforts to relax the traditional architecture point out a new intelligent structure that can provide a complete adaptation. The missing link in the chain of amalgamation of cognitive radio and cross-layer design is cognitive engine. Various work has been carried out to establish this unification. It is not hard to see that there are very challenging issues including both formal and design aspects of the merge. However, no matter how hard the problems are, the innovations that come together with cognitive radio tempt the research community to make the dream come true.

References

1. W. M. Thorburn, "The Myth of Occam's Razor," *Mind*, vol. 27, no. 107, pp. 345–353, July 1918.
2. R. K. Standish, "Why Occam's Razor," *Foundations of Physics Letters*, vol. 17, no. 3, pp. 255–266, June 2004.
3. J. Mitola and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
4. J. D. Day and H. Zimmermann, "The OSI reference model," in *Proceedings of IEEE*, vol. 71, no. 12, Dec. 1983, pp. 1334–1340.

5. G. H. Cooper, “An argument for soft layering of protocols,” Massachusetts Institute of Technology, Cambridge, MA, U.S.A, Tech. Rep., May 1983.
6. D. D. Clark and D. L. Tennenhouse, “Architectural considerations for a new generation of protocols,” *SIGCOMM Computer Communication Review*, vol. 20, no. 4, pp. 200–208, Sept. 1990.
7. Z. J. Haas, “A protocol structure for high-speed communication over broadband ISDN,” *IEEE Network*, vol. 5, no. 1, pp. 64–70, Jan. 1991.
8. H. Arslan, “Adaptation techniques and the enabling parameter estimation algorithms for wireless communication systems,” *Signal Processing for Mobile Communications Handbook, M. Ibnkahla, Ed. Boca Raton, FL: CRC.* CRC Press, 2004, pp. 28-1 – 28-26.
9. V. Srivastava and M. Motani, “Cross-layer design: a survey and the road ahead,” *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112–119, Dec. 2005.
10. S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer design for wireless networks,” *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, Oct. 2003.
11. X. Lin, N. B. Shroff, and R. Srikant, “A tutorial on cross-layer optimization in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.
12. B. Zhao and M. C. Valenti, “Practical relay networks: A generalization of hybrid–ARQ,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, Jan. 2005.
13. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Transactions Network*, vol. 5, no. 6, pp. 756–769, Dec. 1997.
14. W. Stevens, “TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms,” *Request for Command – 2001, RFC2001*, Jan. 1997.
15. V. T. Raisinghani and S. Iyer, “Cross-layer design optimizations in wireless protocol stacks,” *Computer Communications*, vol. 27, no. 8, pp. 720–724, May 2004.
16. G. Xylomenos and G. C. Polyzos, “Internet protocol performance over networks with wireless links,” *IEEE Network*, vol. 13, no. 4, pp. 55–63, July/Aug. 1999.
17. Z. J. Haas, “Design methodologies for adaptive and multimedia networks,” *IEEE Communications Magazine*, vol. 39, no. 11, pp. 106–107, Nov. 2001.
18. T. S. Rappaport, A. Annamalai, R. M. Buehrera, and W. H. Tranter, “Wireless communications: past events and a future perspective,” *IEEE Communications Magazine*, vol. 40, no. 5, pp. 148–161, May 2002.
19. J. Mitola III, “Cognitive radio an integrated agent architecture for software defined radio,” Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, Sweden, May 8, 2000.
20. K. Järvinen, J. Vainio, P. Kapanen, T. Honkanen, P. Haavisto, R. Salami, C. Laflamme, and J.-P. Adoul, “GSM enhanced full rate speech codec,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-97*, vol. 2, Munich, Germany, Apr. 21–24, 1997, pp. 771–774.
21. A. Furuskär, S. Mazur, F. Müller, and H. Olofsson, “EDGE: enhanced data rates for GSM and TDMA/136 evolution,” *IEEE Personal Communications [see also IEEE Wireless Communications]*, vol. 6, no. 3, pp. 56–66, June 1999.
22. S. Nanda, K. Balachandran, and S. Kumar, “Adaptation techniques in wireless packet data services,” *IEEE Communications Magazine*, vol. 38, no. 1, pp. 54–64, Jan. 2000.

23. S. Kallel, "Analysis of a Type II hybrid ARQ scheme with code combining," *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1133–1137, Aug. 1990.
24. T. R. Fischer and M. W. Marcellin, "Joint trellis coded quantization/ modulation," *IEEE Transactions on Communications*, vol. 39, no. 2, pp. 172–176, Feb. 1991.
25. R. van Dyck and D. J. Miller, "Transport of wireless video using separate, concatenated, and joint source-channel coding," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1734–1750, Oct. 1999.
26. Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, July 2000.
27. A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, Nov. 1998.
28. Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and transmission power management for energy efficient wireless video communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 411–424, June 2002.
29. P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
30. C. K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 138–147, June 2001.
31. M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Transactions on Wireless Communications*, vol. 5, no. 2, pp. 435–445, Feb. 2006.
32. M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, Jan. 2005.
33. A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, vol. 3, Miami, Florida, U.S.A., Mar. 13–17, 2005, pp. 1794–1803.
34. C. E. Shannon, "The mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423; 623–656, July, October, 1948.
35. G. B. Dantzig, "Discrete-variable extremum problems," *Operations Research*, vol. 5, no. 2, pp. 266–277, Apr. 1957.
36. J. Nocedal, "Theory of algorithms for unconstrained optimization," *Acta Numerica*, vol. 1, pp. 199–242, 1992.
37. M. Bruglieri, F. Maffioli, and M. Ehrgott, "Cardinality constrained minimum cut problems: complexity and algorithms," *Discrete Applied Mathematics*, vol. 137, no. 3, pp. 311–341, Mar. 2004.
38. A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855–867, Sept. 1990.
39. R. Horst and P. M. Pardalos, *Handbook of Global Optimization.* Boston, U.S.A.: Kluwer Academic Publishers, 1995.
40. Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies.* Berlin, Germany: Springer, 2003.

41. G. E. Evans, “An overview of techniques for solving multiobjective mathematical programs,” *Management Science*, vol. 30, no. 11, pp. 1268–1282, Nov. 1984.
42. M. van Der Schaar and S. Shankar, “Cross-layer wireless multimedia transmission: challenges, principles and new paradigms,” *IEEE Wireless Communications [see also IEEE Personal Communications]*, vol. 12, no. 4, pp. 50–58, Aug. 2005.
43. M. van Der Schaar and M. Tekalp, “Network and content-adaptive cross-layer optimization for wireless multimedia communication by learning,” *IEEE International Symposium on Circuits and Systems*, 2005.
44. A. M. Turing, “On computable numbers, with an application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, vol. 42, no. 2, pp. 230–265, 1936.
45. C. Papadimitriou and M. Yannakakis, “Optimization, approximation, and complexity classes,” in *STOC ’88: Proceedings of the twentieth annual ACM symposium on Theory of computing.* New York, NY, U.S.A: ACM Press, 1988, pp. 229–234.
46. M. Ehrgott, *Multicriteria Optimization*, 2nd ed. Berlin, Germany: Springer, 2005.
47. P. Serafini, “Some considerations about computational complexity for multi objective combinatorial problems,” in *International Conference on Vector Optimization – Lecture Notes in Economics and Mathematical Systems [294].* New York, NY, U.S.A: Springer, Aug. 4–7, 1986, pp. 222–232.
48. Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*, ser. Monographs and Textbooks [Volume 176]. Academic Press, Inc., Orlando, FL, 1985.
49. R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, Apr. 2004.
50. V. Pareto, *Manual of Political Economy [translated by A. Schwier].* Edited by A. Schwier and A. Page, reprinted by Augustus M. Kelley: New York, NY, 1971.
51. U. M. Diwekar, *Introduction to Applied Optimization*, ser. Applied Optimization [Volume 80]. Norwell, Massachusetts, U.S.A.: Kluwer Academic Publishers, 2003.
52. W. Stadler, “A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960,” *Journal of Optimization Theory and Applications*, vol. 29, no. 1, pp. 1–52, Sept. 1979.
53. W. Stadler, “Inititators of multicriteria optimization,” in *International Conference on Vector Optimization – Lecture Notes in Economics and Mathematical Systems [294].* New York, NY, U.S.A: Springer, Aug. 4–7, 1986, pp. 229–234.
54. J. P. Hämäläinen, K. Miettinen, P. Tarvainen, and J. Toivanen, “Interactive solution approach to a multiobjective optimization problem in a paper machine headbox design,” *Journal of Optimization Theory and Applications*, vol. 116, no. 2, pp. 265–281, Feb. 2003.
55. E. R. Lieberman, “Soviet multi–objective mathematical programming methods: an overview,” *Management Science*, vol. 37, no. 9, pp. 1147–1165, Sept. 1991.
56. J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, 60th-Anniversary ed. Princeton, New Jersey, U.S.A.: Princeton University Press, 2004.

57. J. von Neumann, “First draft of a report on the EDVAC,” Moore School of Electrical Enginnering, University of Pennsylvania, Tech. Rep. W-670ORD-4926, June 30, 1945.
58. V. Kawadia and P. R. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Communications [see also IEEE Personal Communications]*, vol. 12, no. 1, pp. 3–11, Feb. 2005.
59. A. M. Sayili, “Was Ibni Sina an Iranian or a Turk?” *Isis*, vol. 31, no. 1, pp. 8–24, Nov. 1939.
60. A. B. MacKenzie and S. B. Wicker, “Game theory and the design of self-configuring, adaptive wireless networks,” *IEEE Communications Magazine*, vol. 39, pp. 126–131, Nov. 2001.
61. J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 15, 1982.
62. C. Wang, P. C. Lin, and T. Lin, “A cross-layer adaptation scheme for improving IEEE 802.11e QoS by Learning,” IEEE Transactions *on Neural Networks*, vol. 17, no. 6, pp. 1661–1665, Nov. 2006.
63. P. C. Lin, C. Wang, and T. N. Lin, “A context–aware approach for multimedia performance optimization using neural networks in wireless LAN environments,” in *Proc. IEEE International Conference on Multimedia & Expo (ICME ’06)*, Toronto, Ontario, Canada, July 9–12, 2006, pp. 1177–1180.
64. T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi, “Adaptive packet video streaming over IP networks: a cross-layer approach,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 385–401, Feb. 2005.