Studies in Computational Intelligence 320

Hidefumi Sawai
Editor

Biological Functions for Information and Communication Technologies

Theory and Inspiration

Springer

Studies in Computational Intelligence

For further volumes:
http://www.springer.com/series/7092

Hidefumi Sawai
Editor

Biological Functions for Information and Communication Technologies

Theory and Inspiration

Springer

Editor
Hidefumi Sawai
Kobe Research Laboratories
National Institute of Information and Communications Technology
588-2, Iwaoka, Nishi-ku
Kobe, 651-2492
Japan
e-mail: sawai@nict.go.jp

ISSN 1860-949X e-ISSN 1860-9503

ISBN 978-3-642-15101-9 e-ISBN 978-3-642-15102-6

DOI 10.1007/978-3-642-15102-6

Springer Heidelberg Dordrecht London New York

Translation from the original Japanese language edition: Seimei to Jouhou Tsuushin, edited and written by Hidefumi Sawai, © 2009 by Hidefumi Sawai, Published by Ohmsha Ltd., 3-1 Kanda Nishikicho, Chiyodaku, Tokyo, Japan. All rights reserved.

Cover design: Scientific Publishing Services Pvt Ltd

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.Springer.com)

Prologue

In a sense, designing and constructing information communication systems inspired by life will establish a new paradigm in the history of science and technology. Given the 4 billion-years required to evolve intelligent life with the superb structures and functions of human and animal brains, it is clear we have much to gain by studying, with an open mind, the deep, sophisticated structures and functions of life forms, as well as the mechanisms of the evolution of life.

This book introduces trends in research on information and communications technologies inspired by life, current trends in research on molecular communication technologies, and future prospects for research in these areas. Chapter 1: “Reconsidering information and communications technologies from life”, Chap. 2 : “Molecular communication as a Biological ICT”, Chap. 3: “Artificial chemistry and molecular networks”, Chap. 4: “Signal transduction in biological systems and its possible uses in computation and communication systems”, and Chap. 5: “For deeper understanding”. I will also present relevant research currently underway.

Chapter 1 aiming at the connections between life and ICT, focuses on information processing research modeled on the brain, on the various theories of evolution, on the early evolution of cells, or inspired by the evolution of life itself. Chapter 2 introduces molecular communication technologies as bio-ICTs and describes biosystem molecular communication, molecular communication architecture, and molecular communication design, which represent molecular communication technologies within the scope of biological information and communications technologies. Chapter 3 focuses on research on a network-type computing machine based on intermolecular interactions, describing control flow clusters as active machines, network connection switching rules based on energy minimization, self-organizing network configurations based on programming of active nodes, molecular agents, and program-flow computing, with the goal of creating new algorithms based on molecular theory. Chapter 4 discusses biosystem signal transmission and potential applications to computation and communication systems, drawing on cellular signal transmission networks and formal models, dynamic analysis of signal transmission networks, and error correction codes for cellular signaling pathways. Chapter 5, for deeper understanding, introduces briefly the original thoughts aiming at and studying on life, and explains the solutions for real-world problems by complex networks which are currently attracting attention.

These topics will help present the argument that designing information processing models inspired by the superb design and function of life forms will lead to the design of various information processing systems and information communication systems capable of solving real-world problems. They also introduce the topic of using the functions of life forms at the molecular level to construct information communication systems based on molecules. Some of these technologies have already entered practical use, while others are expected to find practical applications in the near future.

Hidefumi Sawai

Acknowledgements

I wish to express my deep gratitude to all those who have provided support and cooperation at various stages of this research and publication, especially to Dr. Hideo Miyahara, President of NICT, for his continuous support, Dr. Claus E. Ascheron, Executive Editor of Springer Verlag, Germany, Mr. Osami Takeo, President of Ohmusha Co. Ltd. Japan, Ms. Motomi Sato, Mr. Masayuki Ishida, and Ms. Mizuki Ishikawa of Ohmsha Co. Ltd. Japan, for their support to smoothly proceed this publication. Also, I wish to express my deep gratitude to all authors who have written each chapter, the reference books and papers to which I refer when I edit and write this book, because without these great achievements, this book would not be formed.

Contents

Chapter 1
Reconsidering Information and Communications Technology from Life

Hidefumi Sawai

Abstract When we consider the advanced Information and Communication Technology (ICT), it has appeared to be useful to know deeply about life by recent studies.

In this chapter (and the latter half part of Chap. 5), several cases for ICT researchers and practitioners to develop the better ICT will be explained. Here, these cases include:

- *Brain structure and functions* as one of the by-products of biological evolution, and various *information processing models* with the time and space structure derived from brain.
- Genetic algorithm as a model of biological evolution itself, and *evolutionary computation algorithm* as an extended form of it.
- Algorithm as a model of cell metabolism in the early stage of biological evolution.
- *Algorithm* based on a model of *sexual selection*.
- A useful guideline for constructing a future ICT society which can be obtained by the survey results of trend in recent *complex network science* (described in the latter half part of Chap. 5).

Keywords Life and ICT · Brain function · Evolutionary mechanism

H. Sawai (✉)
Kobe Research Laboratories, National Institute of Information and Communications Technology, 588-2, Iwaoka, Nishi-ku, Kobe, 651-2492, Japan
e-mail: sawai@nict.go.jp

H. Sawai (ed.), *Biological Functions for Information and Communication Technologies,* Studies in Computational Intelligence, 320,
DOI: 10.1007/978-3-642-15102-6_1, © Springer-Verlag Berlin Heidelberg 2011

> *Where the world ceases to be the scene of our personal hopes and wishes, where we face it as free beings admiring, asking and observing, there we enter the realm of Art and Science.*
>
> —Albert Einstein

1.1 Connection Between Life and ICT

1.1.1 Proximate Factor and Ultimate Factor

The reasons for drawing inspiration from life are self-evident. Take, for example, the brain, the seat of intelligence and awareness. The product of four billion years of sustained evolution, this unparalleled object is most definitely not an overnight creation.

To be "*inspired by life*" and to build intelligent information and communication systems, we must not only investigate the functions of the organisms currently found on Earth, but penetrate deep to discover how the diverse and ingenious functions observed today were acquired over the long course of evolution. In other words, we need to look back at the developmental stages in the evolution of living organisms—an *ultimate factor*—in addition to *proximate factors*, proximate factors being the adaptation of various organisms to their immediate environments. The relationship between proximate factors and ultimate factors is briefly discussed below. No understanding of the historical background and significance of modern life science or cognitive science is complete without an understanding of this relationship.

1.1.2 Nature's Hierarchy

> *The most beautiful thing we can experience is the mysterious. It is the source of all true art and science. He to whom this emotion is a stranger, who can no longer pause to wonder and stand rapt in awe, is as good as dead: his eyes are closed.*
>
> —Albert Einstein

It is believed that the universe was created about 15 billion years ago and the Solar System and the Earth approximately 4.6 billion years ago. Primitive life forms are believed to have appeared around 3.8–4 billion years ago. Chemical evolution during the first 600 million years after the formation of Earth eventually led to the origin of life.

Table 1.1 presents the *hierarchy of nature* and the disciplines and research topics associated with each level of this hierarchy. At the smallest scales are *elementary particles, atoms, and molecules*. The topics discussed in this book that

Table 1.1 Nature's hierarchy

Hierarchical levels	Fields and subjects of research
1. Elementary particles, atoms, and molecules	Nano-bio science, artificial chemistry, molecular communication
2. Genes	Molecular biology, neutral theory of evolution, virus evolution
3. Amino acids	Chemical genetic algorithm/programming (CGA/CGP)
4. Proteins	Protein engineering, molecular communication (motor proteins)
5. Cells	Neurons, molecular communication (Ca ion diffusion)
6. Tissues and organs	Tissue engineering, brain and mind, consciousness
7. Organism	Tissue engineering, brain and mind, consciousness
8. Population	Immune system, ESS*, evolution of altruistic behavior, multi-agent system
9. Species	Darwinian theory of evolution, Neo-Darwinism, co-evolution
10. Ecosystem	Theory of habitat segregation, mimicry, migration strategy of population
11. Earth	The Gaia hypothesis, environmental problems
12. The solar system, the galaxy, and the universe	Origin of life

* *ESS* Evolutionary Stable Strategy

correspond to this level of the hierarchy are *artificial chemistry* in Chap. 3 and *molecular communications* in Chap. 2. The next level of the hierarchy is *genetic*. Section 1.3 introduces genetic algorithms and algorithms based on *genetic duplication*. Beyond this is the *amino-acid level* of this hierarchy; topics associated with this level include *chemical genetic algorithms* (CGAs) and *chemical genetic programming* (CGP), presented in Sect. 1.4. Chapter 2 discusses molecular communication and motor proteins as topics at the *protein level* of the hierarchy.

As issues at the *cellular level*, Sect. 1.2 discusses *neural network modeling* based on the model of neurons and synapses. Note that Ca ion diffusion, a topic discussed in Chap. 2, is also associated with this hierarchical level. At the level of *tissues and organs*, researchers are currently pursuing studies of brain machine interfaces (BMIs) to extract and apply neuronal activity outside the brain as part of efforts intended to investigate *consciousness*, the state emerging from the brain functions. *The Origin of Species* by Darwin is associated with the organism level of the hierarchy. Section 1.3.4 discusses algorithms inspired by the theory of *sexual selection*. At the level of populations and species, perhaps the most characteristic topic is *co-evolution* involving multiple agents; the acquisition of behavioral strategy by multiple agents in CGP discussed in Sect. 1.4.2 is indeed based on the mechanism of co-evolution. At the *ecosystem level*, the topic presented is parallel distributed processing for parameter-free genetic algorithms (presented in Sect. 1.3.2), a model inspired by the migrational strategies of various populations. At the *level of the Earth*, environmental problems such as the mitigation of global warming constitute the most pressing topics. At the level of the *universe*, when we consider that the origin of life is a lifeless molecule, we

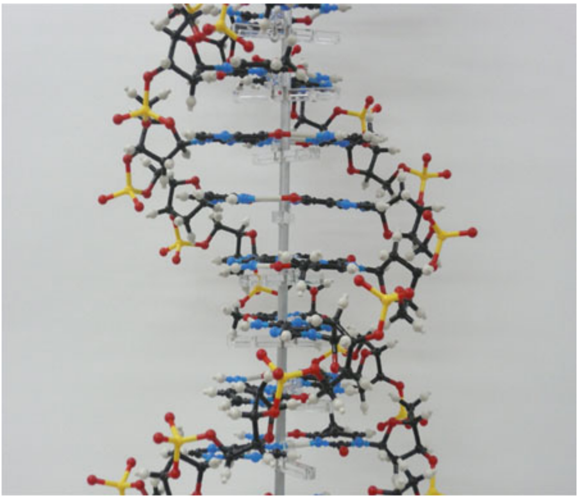


Fig. 1.1 Double helix structure of DNA

understand that the vast scale of the universe is closely linked to small-scale structures at the level of elementary particles, atoms, and molecules. We also consider how what we observe of the universe (*nature's hierarchy*) reflects a mode of observation unique to humans.

A satisfactory *Theory of Everything*, or TOE, that provides a unified explanation for all levels of the hierarchy presented in Table 1.1 has yet to be established. *Micro-scale* theories such as elementary particle theory and quantum mechanics and *macro-scale* theories such as electrodynamics, Newtonian dynamics, and Einstein's general theory of relativity have resolved many of nature's mysteries. But the handling of *mesoscale* problems remain a challenge, and the respective theories remain works in progress, despite the emergence of studies on complexity (complex science) through chaos theory. As is well known, the discovery of DNA's double helix structure (shown in Fig. 1.1) by Watson and Crick in 1953 triggered rapid progress in *molecular biology* and has pushed the discipline to its current prominence.

In the *information sciences*, research has declined in the area of artificial intelligence, applications of which include expert systems that seek to embody human expertise. However, studies continue on artificial neural networks, artificial life, and artificial chemistry, which have come to be regarded as fundamental areas of study. Current chapter and Chap. 3 of this book introduce recent research achievements in this area. While Descartes's formulation of the *mind–body duality* has sequestered physical and spiritual or mental issues as problems in totally distinct dimensions, recent progress in cognitive science and the development of technologies for non-invasive measurement of brain functions such as f-MRI (functional Magnetic Resonance Imaging), MEG (Magneto Encephalography), NIRS (Near-Infrared Spectroscopy), and EEG (Electro Encephalography) have transformed the study of human *consciousness* and subjective perception into subjects of natural science (cognitive science). This is a potentially epochal event in the history of scientific and technological development.

1.2 Hints from Brain Function

1.2.1 Brain Structures and Their Functions

This section presents a summary of the structure and functions of the brain. As shown in Fig. 1.2, the brain is divided into left and right hemispheres. The brain can also be divided into the following units: the cerebrum, consisting of frontal, parietal, temporal, and occipital lobes; the cerebellum; the brain stem; and the spinal cord. Each part has a modular structure consistent with its function. The brain constitutes a system of immense complexity, composed of some 100 billion *neurons* (counting both the cerebrum and cerebellum), with each neuron connected to other neurons by several thousand to tens of thousands of synapses. This complex organ is the product of evolution. Figures 1.3 and 1.4 present the structures of a neuron and a neural network, respectively. As shown in Fig. 1.5, a synapse in essence is the

Fig. 1.2 Brain structure

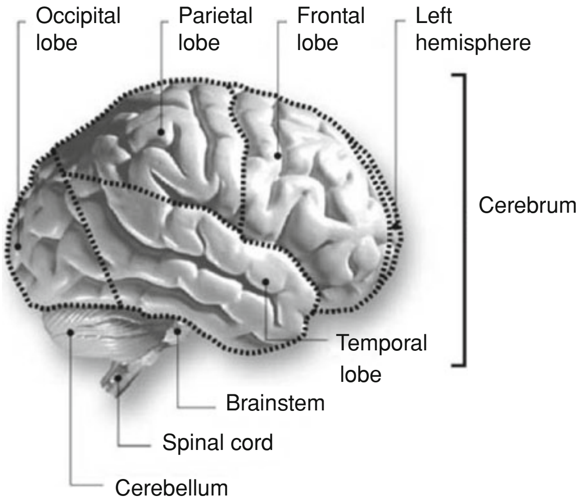


Fig. 1.3 Neuron (nerve cell)

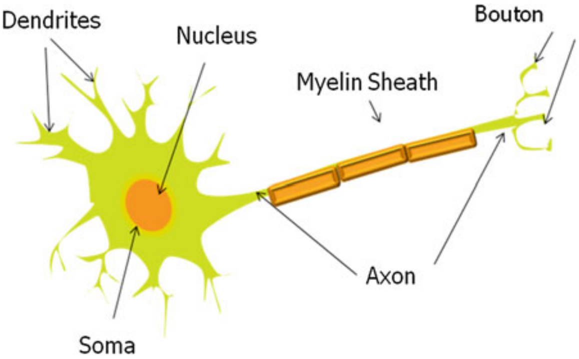
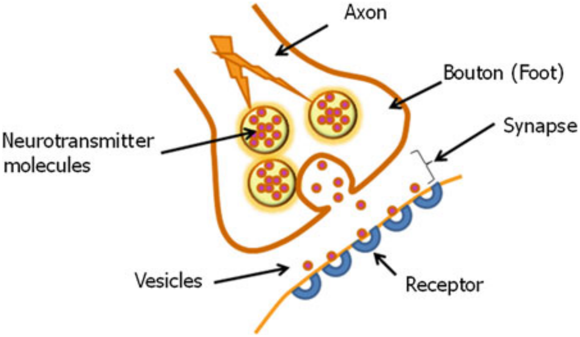


Fig. 1.4 Neural network

Fig. 1.5 Synapse



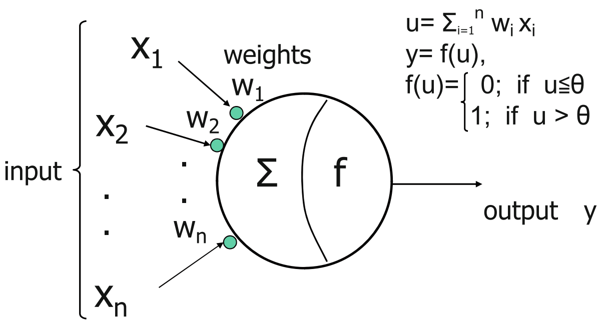
space between two neurons across which electric pulses traveling along an *axon* are relayed to an adjacent *dendrite* via chemical transmitters.

A neural network is an example of an information processing model based on brain function. Presented below are modeling and design methods for *neural network* architectures suitable for speech or image pattern recognition.

Special focus will be given to the feature extraction technique tailored to the temporal structure of speech and the spatial structure (information structure) of images.

1.2.2 Neural Networks Modeling Brain Function

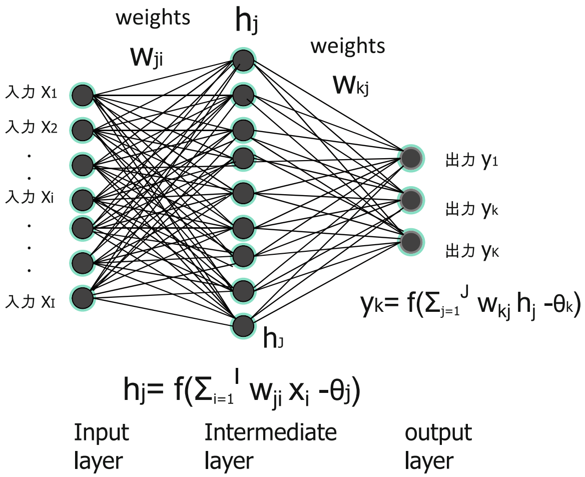
Figure 1.6 is a model of the neuron proposed by McCulloch & Pitts in 1943. The input signal x_i $(i = 1, 2, \ldots, n)$ is input to the neuron, with the links representing synapses weighted by w_i. The resulting internal potential u is the sum of these products, or $\Sigma w_i x_i$. The sum is input to the threshold function $y = f(u)$, which is $y = 0$ when the value of u is below a certain threshold value of θ and $y = 1$ when


input
x_1
x_2
x_n
weights
w_1
w_2
w_n
Σ
f
output y
$u = \Sigma_{i=1}^{n} w_i x_i$
$y = f(u)$,
$f(u) = \begin{cases} 0; & \text{if } u \leqq \theta \\ 1; & \text{if } u > \theta \end{cases}$

Input signals x_i (i=1, 2, …, n) are fed into the neuron with the synapse's weights w_i. As a result, the inner potential u becomes the value ∑wi xi. Then, u is input to the threshold function y=f (u) , and the output y becomes 1 with firing when u is greater than a threshold θ, and 0 when u is less then the threshold.

Fig. 1.6 Formal neuron in McCulloch and Pitts [25]

Fig. 1.7 Artificial neural network (ANN) [26]



u is greater than the threshold. Thus, this neuron model adopts a weighted majority logic based on threshold value. The threshold function $y = f(u)$ is generalized into a differentiable sigmoid function $y = 1/\{1 + \exp(-x + \theta)\}$ and applied to the learning rule by *error back-propagation* for the artificial neural network (ANN) shown in Fig. 1.7.

Column 1: Artificial Neural Networks and Machine Learning

Artificial neural networks (ANNs) are information processing machines that model the structure and functions of a brain. However, the structure of an actual brain consists of vast numbers of neurons (a human brain is said

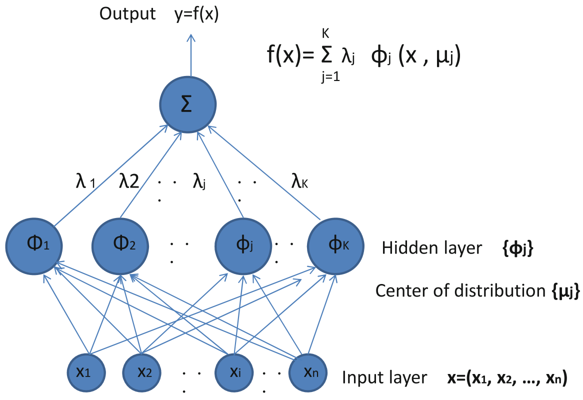

Output y=f(x)
f(x)= Σ_{j=1}^{K} λj ϕj (x , μj)
Σ
λ1 λ2 λj λK
Φ1 Φ2 ϕj ϕK
Hidden layer {ϕj}
Center of distribution {μj}
x1 x2 xi xn
Input layer x=(x1, x2, ..., xn)

Fig. 1.8 Radial basis function (RBF) networks (reproduced from Ref. [1])

to contain more than 100 billion neurons) interlinked to form a vast neural network structure. A brain is an extraordinarily complex system—a super-complex system. An ANN seeks to abstract the brain's structure and functions to extract the essence and to generate the ability to learn.

ANNs can be classified into two categories, depending on learning method: supervised learning models and unsupervised learning models [1, 2]. Typical examples of supervised learning models include the following:

Rosenblatt's perceptron
Multi-layer perceptron (MLP)

These examples separate and recognize patterns in a data space using hyperplanes. In contrast, the two examples below clusterize patterns by combining basic functions (such as Gaussian functions) called kernel functions.

Radial-basis function (RBF) networks (Fig. 1.8 in this column)
Support vector machines (SVM)

Figure 1.9 shows the differences in pattern classification between MLP and RBF.

Other examples include statistical and probabilistic computational models. Two such models are listed below.

Boltzmann machine
Bayesian network

On the other hand, typical examples of the "unsupervised learning model" include the below.

Kohonen's self-organizing feature map (SOM). Although not an ANN, reinforcement learning models [3], inspired by learned behavior in animals, are machine learning models intermediate between supervised and unsupervised learning models. Two examples are given below.

Time difference (TD) learning
Q-learning

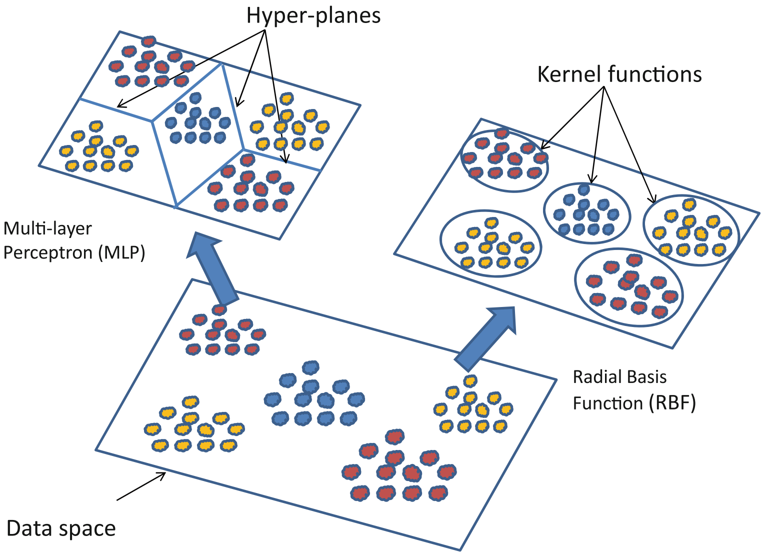


Fig. 1.9 Differences in pattern classification between MLP and RBF (Reproduced from Ref. [2])

1.2.3 Time-Delay Neural Networks Suitable for Processing Sequential Information and Their Expansion [4–8]

Figure 1.10 shows a unit used in a *time-delay neural network* (*TDNN*) [4]. The architecture shown here was proposed to process time series data—for example, speech. To the left is the input unit, which is linked to the host output unit via synapse W, which can be a non-time-delay concatenation or a time-delay concatenation with a delay of $D_1, D_2, \ldots, D_n$. This architecture is suitable for processing speech signal patterns that have temporal structures. Weighted summation ($\Sigma w_{ij}\ x_i$) is performed on the input signals, and the resulting value is sent to the sigmoid function $f(x) = 1/\{1 + \exp(-x)\}$ and the result output.

Figure 1.11 shows the architecture of a TDNN designed to distinguish the voiced plosives /b, d, g/ in the Japanese language [4]. From bottom to top are the input layer, hidden layer 1, hidden layer 2, and output layer. The input layer consists of a total of 240 units, or 15 and 16 units in each of the x- and y-axis directions.

The x- and y-axis correspond to the time- and frequency-axis, respectively. The time–frequency spectrum (sound spectrum) produced by frequency analysis performed every 10 ms is input to the input layer. This time–frequency spectrum is shifted by a time window of 30 ms (3 units) and after being multiplied by the weighting factors of synapses having time-delay, it is linked to the host unit in hidden layer 1. As in the input layer, the total of 40 units in hidden layer 1, five units in the x-axis (time-axis) direction, and the eight units in the y-axis (frequency-axis) direction are linked to hidden layer 2 with the time-delay. In hidden

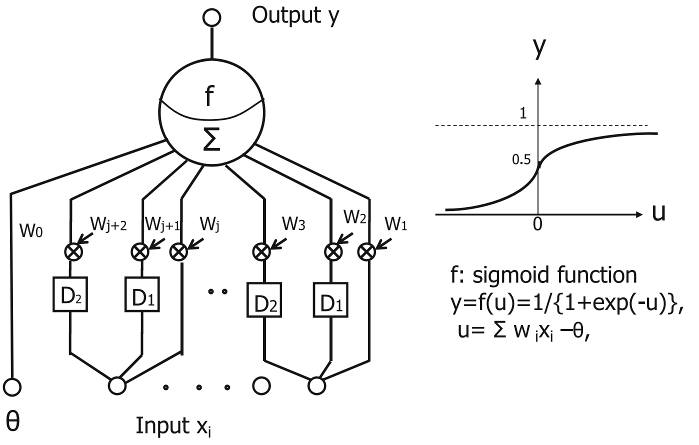
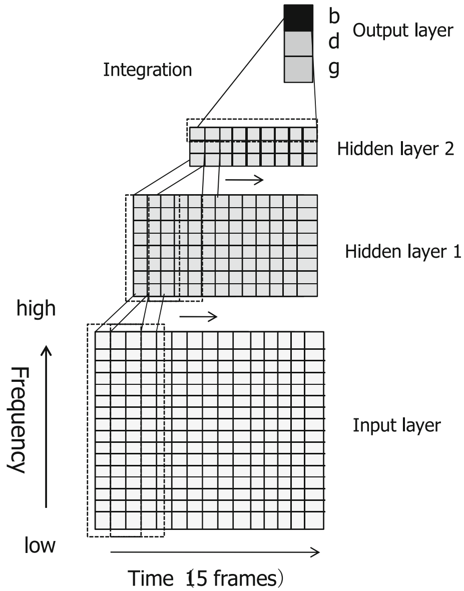


Fig. 1.10 Unit of time-delay neural network (TDNN)

Fig. 1.11 Time-delay neural network (TDNN) architecture



layer 2, each of the nine units in the *x*-axis direction is assigned to one of three categories /b, d, g/, and concatenated to the output unit with the time-delay. This TDNN is trained by error-propagation. The results of discrimination testing for input phonemes not used for training indicate that the TDNN achieves a speaker-dependent recognition rate of approximately 98–99%, representing a reduction in misrecognition rate to approximately 1/4 that of the conventional HMM (Hidden Markov Model) widely used for speech recognition applications. (HMM is associated with a recognition rate of 91–97%.) By expanding the phonemic category in

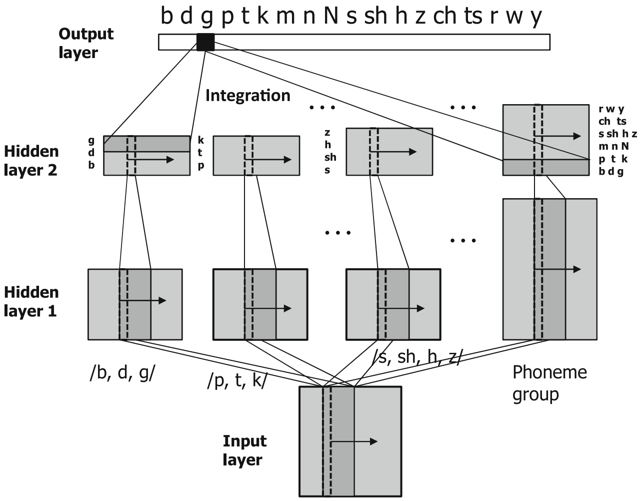


Fig. 1.12 Modular architecture of all consonant network

a similar manner, we should be able to build a TDNN for all Japanese phoneme groups.

Figure 1.12 shows the modular architecture of a TDNN capable of discriminating the 18 consonants of the Japanese language [5]. As the figure shows, the 18 consonants are divided into six groups—the voiced plosives /b, d, g/, unvoiced plosives /p, t, k/, nasals /m, n, N/, fricatives /s, sh, h, z/, affricates /ch, ts/, and liquids and semi-vowels /r, w, y/. A TDNN capable of distinguishing between the six phoneme groups is designed so that the results of discrimination processing within each phoneme group and those from group discrimination processing can be linked in the output layer.

Figure 1.13 shows a TDNN system with a TDNN capable of distinguishing between the Japanese language vowels /a, i, u, e, o/ and a TDNN capable of distinguishing between the six consonant groups and the vowel group added to the TDNN in Fig. 1.12, in addition to a speaker-dependent recognition TDNN expanded to operate as a speaker-independent recognition network [6]. As the figure shows, this TDNN is a large-scale network with a 3D structure. Its scale enables speech recognition of all Japanese consonants and vowels in speaker-independent mode. Adding a discrimination unit (Q) for the presence/lack of voice (voiced/unvoiced) makes it possible to automatically recognize Japanese phonemes (called phoneme spotting) simply by scanning vocalized speech along the temporal direction. Readers are referred to Ref. [6] for more information on the recognition performance of these networks.

Figure 1.14 outlines a different approach that also results in a speaker-adaptive neural network [7]. The three lowermost layers are neural networks that perform speech spectra mapping to adapt the speech of an unknown speaker to that of a standard speaker used in training. This approach—providing the mapping to the

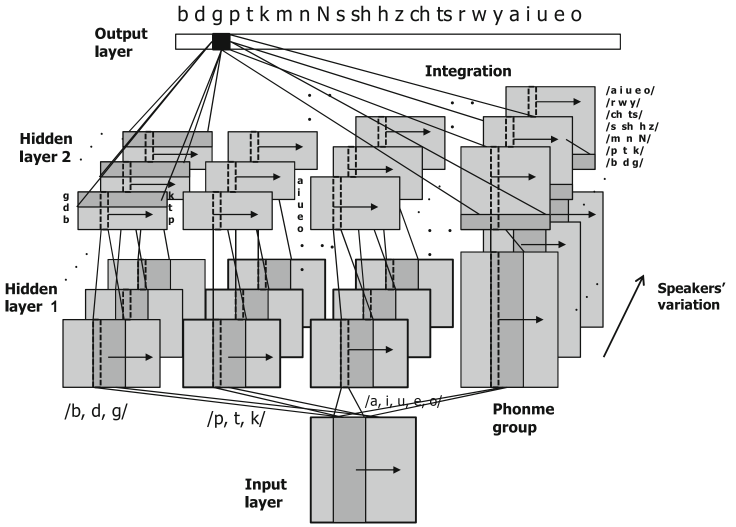


Fig. 1.13 Large-scale TDNN architecture for speaker-independent recognition

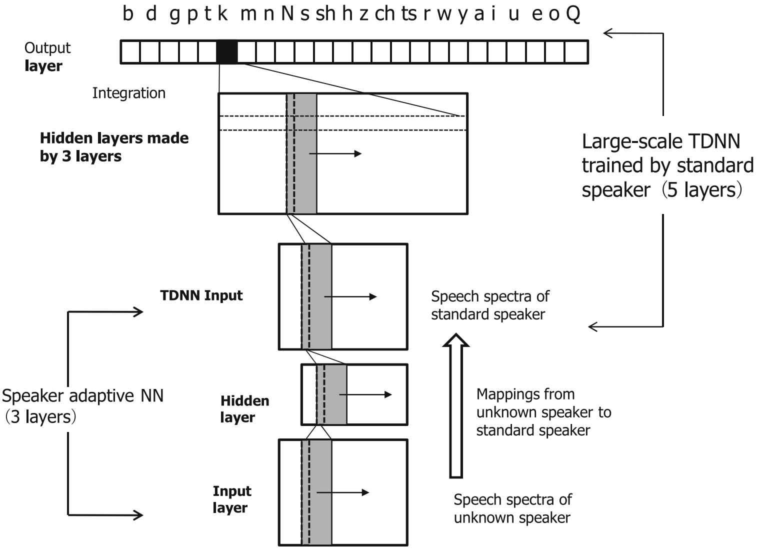

b d g p t k m n N s sh h z ch ts r w y a i u e o Q
Output layer
Integration
Hidden layers made by 3 layers
Large-scale TDNN trained by standard speaker（5 layers）
TDNN Input
Speech spectra of standard speaker
Speaker adaptive NN（3 layers）
Hidden layer
Mappings from unknown speaker to standard speaker
Input layer
Speech spectra of unknown speaker

Fig. 1.14 Large-scale TDNN architecture with speaker-adaptive neural network

TDNN in advance—may represent a highly effective strategy for applying a TDNN trained to a standard speaker to recognize the speech of unknown speakers.

Figure 1.15 shows an expanded architecture model of a new TDNN designed to absorb temporal and frequency fluctuations in speech. Time and frequency windows are installed in the input layer, and extractions of feature quantities (the feature quantity associated with temporal fluctuations and those associated with

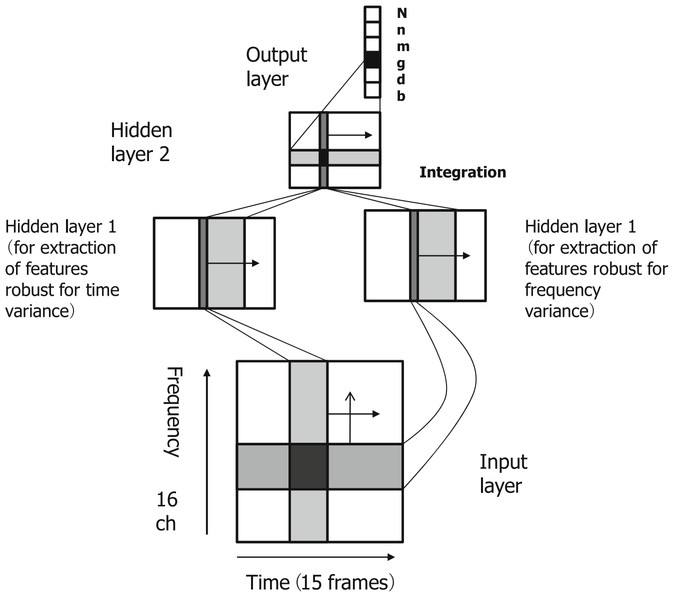

N
n
m
g
d
b
Output layer
Hidden layer 2
Integration
Hidden layer 1 (for extraction of features robust for time variance)
Hidden layer 1 (for extraction of features robust for frequency variance)
Frequency
16 ch
Input layer
Time (15 frames)

Fig. 1.15 Frequency-time-shift-invariant TDNN architecture

frequency fluctuations) are integrated in hidden layer 1. The signal is ready for output by the output layer after integration in hidden layer 2. The figure is an expanded TDNN designed to make precise distinctions within a category consisting of six phonemes associated with high misrecognition rates: the voiced plosives /b, d, g/ and nasals /m, n, N/.

Figure 1.16 is a neural network with block windows inspired by the "neocognitron", extensively investigated for applications in handwritten letter recognition. As with handwritten letter recognition, it is necessary to absorb fluctuations in the absorption along the time (*x*-axis) and frequency (*y*-axis) directions in speech pattern recognition. Thus, we can build an architecture that promotes such absorption by installing block-shaped windows in the lower layers so that the feature quantities can be integrated in succession as connections are made upward. The recognition performance of these neural networks is discussed in Ref. [8].

1.2.4 Expansion of Time-Delay Neural Networks to Rotation-Invariant Pattern Recognition [9]

The architecture in Fig. 1.17 is an expanded NN having *axial symmetry* created by expanding the translation invariance of TDNN to rotational invariance. The synapse weighting factors having parallel assignments from the bottom to upper layers retain the original values acquired through training. This architecture means that if

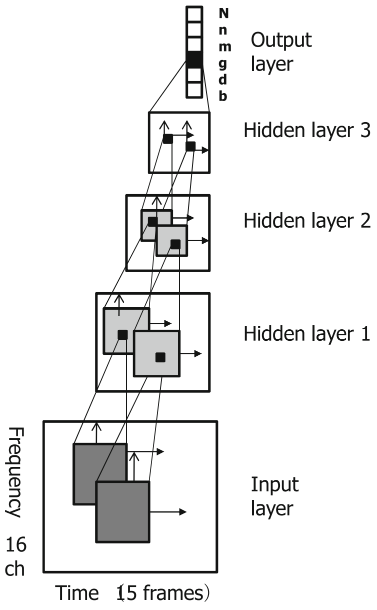


Fig. 1.16 Block-windowed neural network (BWNN) architecture

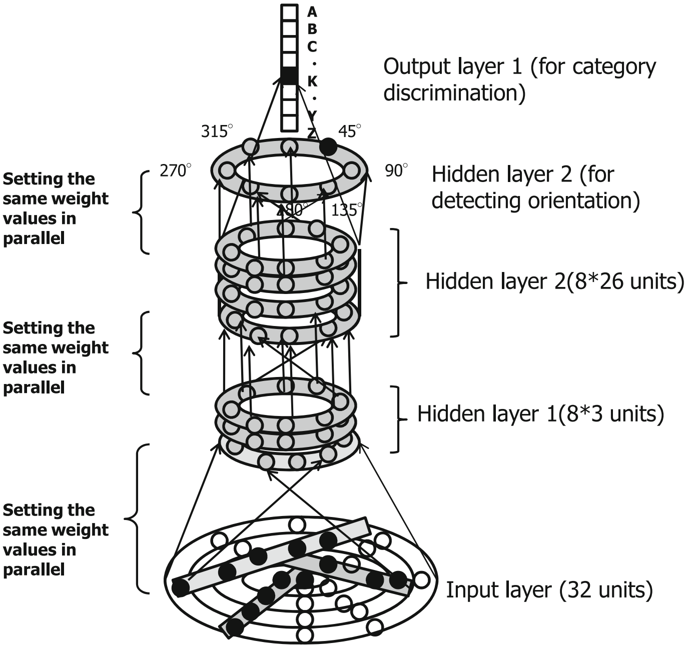

A
B
C
·
K
·
Y
Z
Output layer 1 (for category discrimination)
315°
45°
270°
90°
180°
135°
Hidden layer 2 (for detecting orientation)
Setting the same weight values in parallel
Hidden layer 2(8*26 units)
Setting the same weight values in parallel
Hidden layer 1(8*3 units)
Setting the same weight values in parallel
Input layer (32 units)

Fig. 1.17 Axially symmetric neural network architecture

the synapse weighting factors can be learned using error back-propagation for every pattern of the letter pattern class categories (A–Z) input at a given rotational position (0 deg., for example), the neural network can correctly recognize the class category and rotation angle when an arbitrary class category is input at any angle. The figure shows a neural network architecture capable of recognizing the 26 letters of the alphabet at rotation angle intervals of 45°. From bottom to top are the input layer, hidden layer 1, hidden layer 2, output layer 2 (for rotation angle recognition), and output layer 1 (for class category recognition). As this neural network has an axially symmetric structure, proper execution of the recognition function requires careful alignment of the center of the pattern of the image fed to the input layer.

1.3 Theory of Evolution and Information Processing Model

An information processing model inspired by biological evolution is a computational algorithm based on genetic algorithms and evolutionary computation. Table 1.2 is a compilation of the evolutionary computation algorithms introduced in this and subsequent sections.

In 1859, Charles Darwin released his book, *The Origin of Species* [10], which dealt with the genetics and evolution of organisms. Prompted by the book, John Holland advanced the idea of genetic algorithms (GA) in the 1960s [11]. Genetic algorithms have been applied to problems in numerous fields, including functional optimization, combinatorial optimization, and parameter optimization in machine design. In recent years, genetic algorithms have been integrated with other techniques such as Evolutionary Strategies (ES) by Rechenberg and Evolutionary Programming (EP) by L. Fogel to form a research discipline known as Evolutionary Computation (EC).

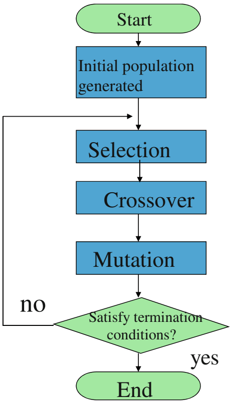
Column 2: Genetic Algorithms [11, 28, 29]

Genetic algorithms offer certain advantages over the algorithms used in other information processing systems: they do not need to assume differentiability

Table 1.2 Some examples of evolutionary computation algorithms

Life phenomenon, theory of evolution (from micro to macro)	Evolutionary computation algorithm
Disparity theory of evolution (Furusawa) [13]	Parameter-free GA [12]
Theory of gene-duplication (Ohno) [16]	Gene duplicating GA [15]
Codon to amino acid translation	Chemical GA [22], Chemical GP [23]
Sexual selection (Darwin) [10]	Evolutionary computation based on sexual selection [19]
Ecosystem	Hierarchical parallel distributed GA [14]

Fig. 1.18 Flow chart of GA (SGA)



of evaluation functions (functions can be indifferentiable); genetic manipulations such as crossovers and mutations readily circumvent the tendency to become mired in local solutions; and they offer a superior capacity for global and local search. Here, we describe the most basic of all such algorithms: the Simple Genetic Algorithm (SGA). Figure 1.18 shows a flowchart for SGA.

We begin by preparing an initial population consisting of *n* individuals. Based o n roulette-wheel selection, we select *n* individuals at a probability proportional to the fitness *fi* of the individual *i*. Two individuals randomly selected from the *n* individuals undergo "crossover" at an arbitrary crossove r-point on a "chromosome." This is called a one-point crossover. The ratio of individuals experiencing crossover to the overall population is known as the crossover rate. In subsequent "mutations," the genes of randomly selected individuals are modified at a specified mutation rate. In a binary coded genotype, the bit-flip from 1 to 0 or from 0 to 1 is performed at the probability corresponding to this mutation rate. The operations up to "selection," "crossover," and "mutation" correspond to a single GA generation. If the fitness of the population is sufficient for environmental evaluations (that is, if the quality of a given solution for a given problem is adequate), the process ends. If not, the process returns to "selection," repeats the series of genetic manipulations described above for several generations, in which process the population evolves, possibly leading to a satisfactory solution population.

In general, SGA is used as an evolutionary method to acquire solutions for relatively easy problems. As the problems become more difficult, SGA has increasing difficulty in efficiently acquiring optimal solutions. For this reason, various other evolutionary computational methods have been proposed. Examples include the parameter-free genetic algorithm (PfGA) [30] and the chemical genetic algorithm (CGA) [22] described in this document. Figures 1.19 and 1.20 in this column show, respectively, the multi-point crossover and block-wise mutation used in PfGA.

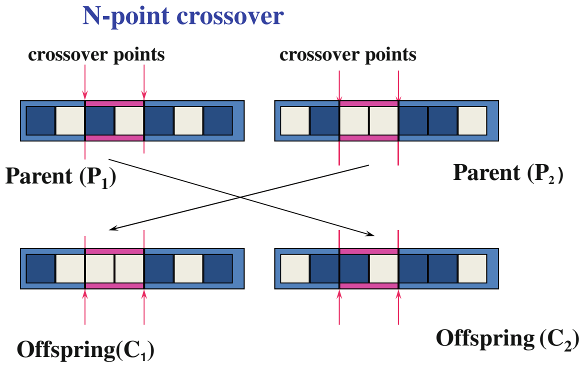


Fig. 1.19 Multi-point crossover (n-point crossover) in the parameter-free genetic algorithm (PfGA)

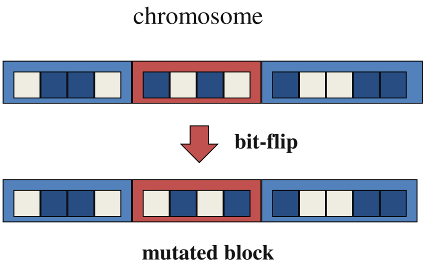


Fig. 1.20 Block-wise mutation in the parameter-free genetic algorithm (PfGA)

The migration of individuals from one local population to another makes it possible to obtain an optimum solution fairly rapidly by parallel distributed processing while expanding population diversity. This document describes several methods of parallel distributed processing based on the PfGA.

1.3.1 Parameter-Free Genetic Algorithms [12] Based on Disparity Theory of Evolution [13]

This section discusses a new algorithm, the parameter-free genetic algorithm (PfGA), which requires no initial setting of genetic parameters such as initial population size, crossover rate, or mutation rate. The PfGA was inspired by and builds on the *disparity theory of evolution* proposed by Furusawa et al. [13], which itself is based on mutations in the double strands of DNA (Figs. 1.21, 1.22). According to disparity theory, when the double strands of DNA unwind and a copy of each is created, a difference emerges in the rate of mutation between leading and lagging strands. This is because the direction of replication in the former is the same as the direction of unwinding, while the direction of replication in the latter is in the opposite direction. While mutations are generally rare in the leading

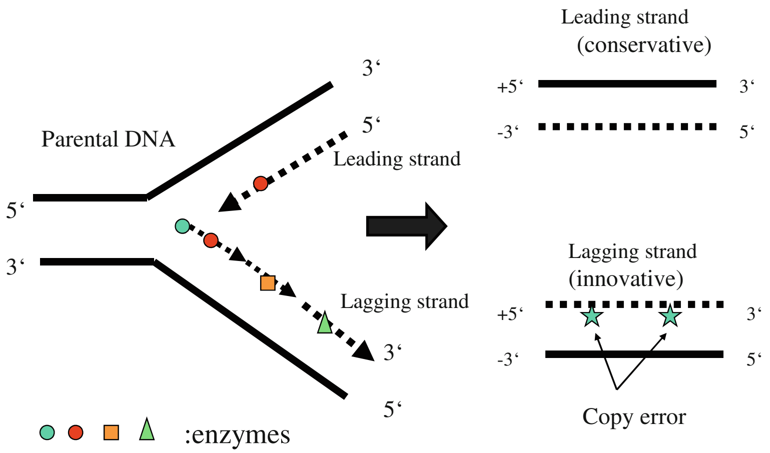


Fig. 1.21 Hypothesis based on the disparity theory of evolution (Reprinted from K. Wada, "Evolution of Digital Life", Fig. 7., page 73, Iwanami Science Library 11, Iwanami Shoten, 1994.)

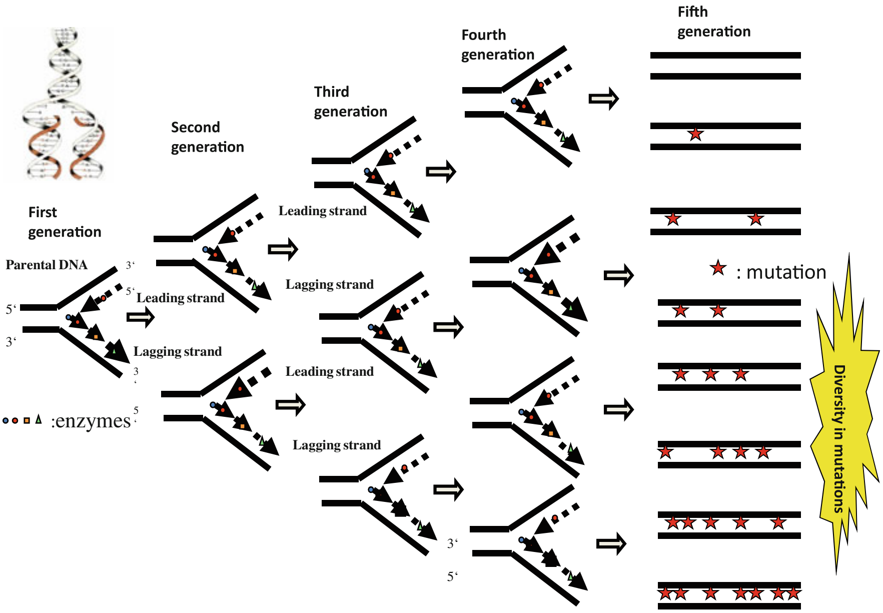


Fig. 1.22 Emergence of diversity according to the disparity theory of evolution (recreated by the author based on Fig. 7 from Ref. [27])

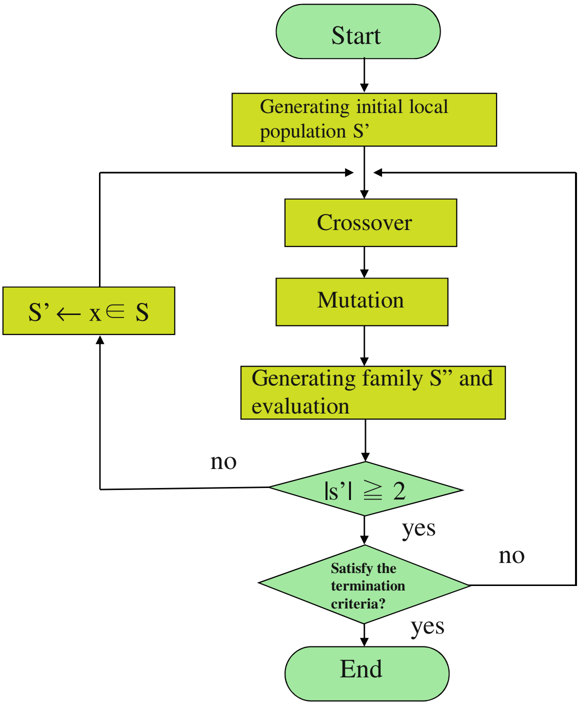
strand (conservative), the lagging strand displays comparatively high mutation rates (innovative). Disparities in replication errors accumulate through crossovers and mutations over generations, creating DNA diversity within a single population via DNA strands experiencing little or no mutation and strands with accumulated mutations. The former type promotes the stability of the population, while the latter promotes flexibility. In the case of PfGA, the former corresponds to the optimized individual at a specific point in time, while the latter corresponds to the offspring produced by crossover and mutation. Thus, under the disparity theory of evolution, the mechanism by which diversity is retained while maintaining a balance between exploitation (localized search) and exploration (global search) can be understood as a balance between genetic *stability* and *flexibility*.

In PfGA, the population is regarded as a set of all possible solutions in the whole search space. In this whole search space S, a local sub-population S' is set. Two individuals are selected from this sub-population S' as parents. The parents are subjected to crossover and mutation to generate a family (S'') of four, with two offspring. The fitness of the four individuals within this family is then evaluated to select or eliminate individuals to evolve local population S' and to execute a search for the solution.

Below are the steps in the basic algorithm for the PfGA (See Fig. 1.23).

1. An individual is randomly extracted from S and is regarded to be initial local population S'.
2. An individual is randomly extracted from S and added to the local population S'.
3. Two individuals are randomly extracted from local population S' for use as parent 1 (P_1) and parent 2 (P_2) in the multi-point crossover.

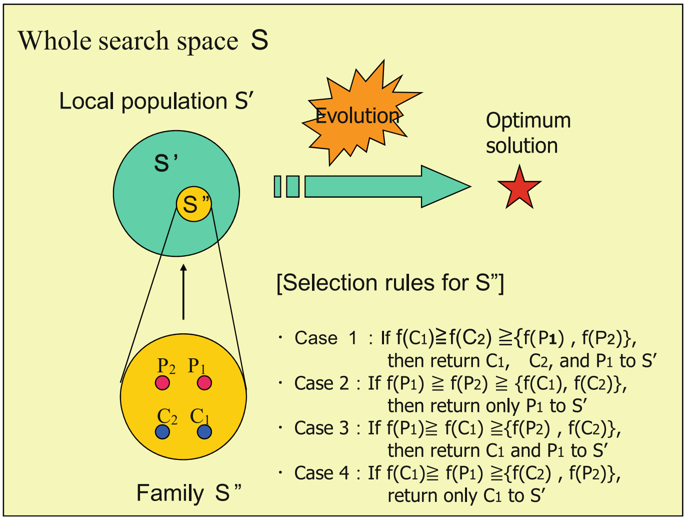
Fig. 1.23 Flowchart of parameter-free GA (PfGA)


Start
Generating initial local population S’
Crossover
Mutation
Generating family S” and evaluation
S’ ← x∈ S
no
|s’| ≧ 2
yes
no
Satisfy the termination criteria?
yes
End

4. Of the two individuals generated by the crossover, one is randomly selected and inverse mutation is applied at a random number of points at random positions.
5. Selection and elimination is performed for a total of four individuals (referred to as a family) consisting of the two generated offspring (C_1 and C_2) and the two parents (P_1 and P_2), by selecting either one or three members of the family to be returned to local population S', based on the calculated fitness.
6. If the local population size $|S'| \geq 2$, then return to step 3; if $|S'| = 1$, then return to step 2 and repeat the cycle.

Multi-point crossover is used as the crossover mode in PfGA. In multi-point crossover, both the number and positions of crossover points are determined randomly, and crossover is executed between chromosomes of two different individuals. Mutations use the erroneous copy of chromosomes generated during the crossover. Thus, of the two offspring produced, one is randomly selected and a partial inversion of the gene sequence carried out to create mutations at a random number of points at random positions. Here, of the two offspring produced by the crossover, one is left untouched by mutation to allow one offspring to retain at least a portion of the parents' traits. In this manner, the PfGA is implemented to execute genetic manipulation based on random numbers and to minimize *ad hoc* choices.

For selection and elimination, to retain the diversity of the local population while maintaining a balance between global search (exploration) and local search (exploitation), the fitness of the four members of the family (S'') are compared and selections made in the manner presented in the following four cases as shown in the left plot of Fig. 1.24. Local population S' is evolved while dynamically maintaining a balance between global and local search and concurrently changing



[Selection rules for S"]

- Case 1 : If $f(C_1) \geqq f(C_2) \geqq \{f(P_1), f(P_2)\}$, then return C_1, C_2, and P_1 to S′
- Case 2 : If $f(P_1) \geqq f(P_2) \geqq \{f(C_1), f(C_2)\}$, then return only P_1 to S′
- Case 3 : If $f(P_1) \geqq f(C_1) \geqq \{f(P_2), f(C_2)\}$, then return C_1 and P_1 to S′
- Case 4 : If $f(C_1) \geqq f(P_1) \geqq \{f(C_2), f(P_2)\}$, return only C_1 to S′

Fig. 1.24 Population and selection rule in PfGA

the size of the local population S' based on an implicit set of rules for switching between cases 1–4, depending on the relative superiority of the fitness of family members. This feature improved the search efficiency of PfGA over other GAs of fixed population size, since it eliminates the need to perform unproductive searches. In addition, the best individual among the four family members is always returned to the local population S'. Thus, the family may be said to be adopting an elite-preserving strategy: The algorithm guarantees the retention of the best individual at a given point in time while simultaneously performing an active search over a very wide (neighborhood) space. If a better individual than the currently best individual is generated, the center of the search transfers to that offspring; if not, the current best individual is retained. This avoids fitness degradations during the course of evolution.

Column 3: The Disparity Theory of Evolution [13, 31–33]

Mutation is the driving force of evolution. DNA mutations are now known to be distributed unevenly—to exhibit disparity. The two strands forming the double helix of the DNA replicate in opposite directions. While one strand (the leading strand) is continuously replicated by an enzyme known as polymerase, the other strand (the lagging strand) is replicated by several enzymes in a complex process of connecting partially replicated fragments (called Okazaki fragments), resulting in a relatively high error (mutation) rate during replication. The error rate for the lagging strand is ten to a hundred times (or even more) than that of the leading strand. In other words, a disparity results wherein the rates of mutations in the replicated double helix strands are unequal. One strand may have many mutations, the other very few.

Based on what is known as the parity model, mutations were previously believed to be distributed evenly in a population. However, an excessive mutation rate will tend to lead to the death of the entire population. The parity model distributes mutations evenly in the DNA of the offspring. In contrast, the disparity model produces both DNA without mutations (or wildtype) and DNA with more mutations than predicted by the parity model. In short, populations with diverse mutations appear, while the wildtype is maintained. Figure 1.22 shows the emergence of diversity according to the disparity theory of evolution. It shows expanding diversity in a population based on the accumulation of mutations generation after generation.

These patterns hold not just for lower organisms such as *Escherichia coli*, but for more complex organisms. Therefore, it is possible to accelerate the rate of evolution by increasing the mutation rate while avoiding the risk of extinction. Dr. Mitsuru Furusawa, who proposed the disparity theory of evolution, calls it the “creation of diversity with guaranteed capital." The disparity theory of evolution features both "conservation," which reduces the risk of extinction by guaranteeing the wildtype after generations, and "innovation," which generates the diversity of individuals needed for evolution. The authors proposed the parameter-free genetic algorithm (PfGA) [30] as an information processing model that accelerates evolution while incorporating a dynamic balance between conservation and innovation.

1.3.2 Expansion of Parameter-Free Genetic Algorithm to Parallel Distributed Processing Techniques [14]

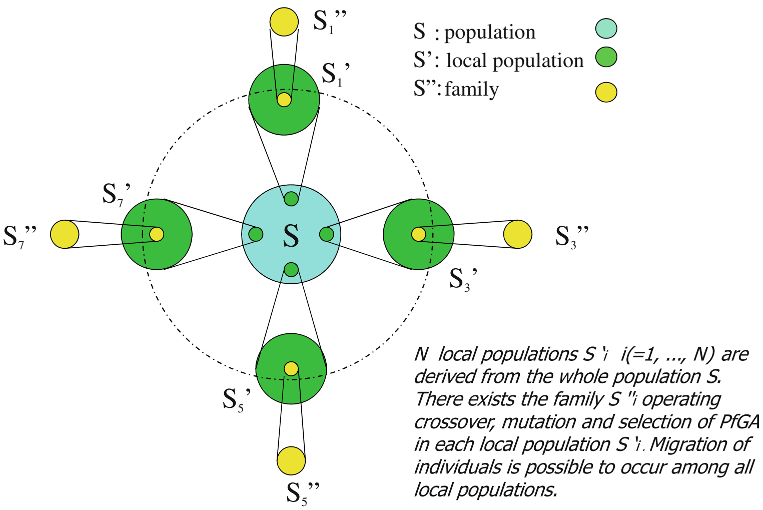
This section describes techniques for *parallel distributed processing* related to the parameter-free genetic algorithm (PfGA) inspired by ecosystems. In general, the main objective of parallel processing in any processing, including GAs, is to increase processing speed. However, we can dramatically enhance the efficiency of search problem processing based on a GA by introducing interactions between individuals by migration, rather than simply dividing up the task. In the case of a coarse-grained parallel GA, the local population is treated as the processing unit, and individuals are migrated between local populations at appropriate frequencies. In a fine-grained GA, the neighborhood of a given individual is treated as the processing unit, and overlaps are set among neighbors. The former is frequently referred to as the island model, wherein a single local population constitutes the deme of a single species. We use this as the model.

1.3.2.1 Two Parallel Processing Architectures

Two types of parallel processing architectures are used: the *uniformly-distributed type* and the *master–slave type*. The uniformly-distributed type corresponds to a situation in which all local populations have the same role and local population monitoring functions are absent.

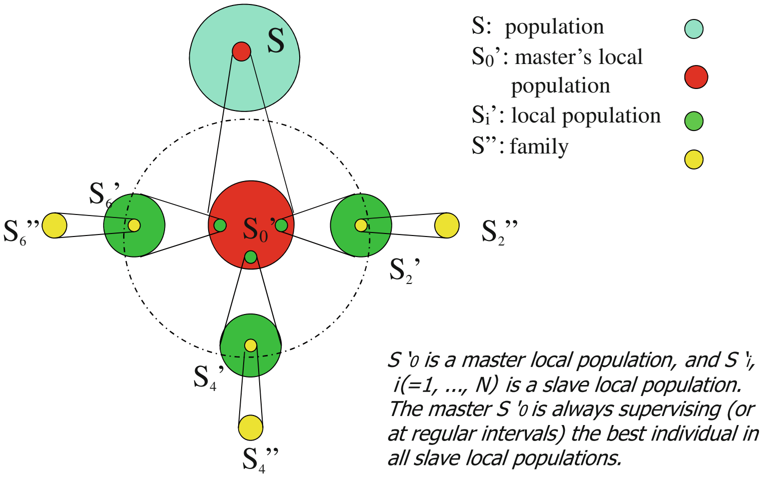
On the other hand, in the master–slave type, a master local population is equipped with a function for monitoring the processing of other local populations, called slaves. Figure 1.25 shows the uniformly-distributed-type PfGA architecture. From the whole search space S, an N number of local populations S_i' $(i = 1, \ldots, N)$ is derived; in each local population S_i', there exists a family (S_i'') that performs PfGA crossovers, mutations, and selection. Migration of individuals may occur between any of the local populations.

Figure 1.26 shows the master–slave architecture. S_0' is the master local population, while S_i' $(i = 1, \ldots, N)$ are the slave local populations. The master S_0' consistently (or at constant intervals) seeks to identify the best individual in all slave populations.



N local populations S'_i i(=1, ..., N) are derived from the whole population S. There exists the family S''_i operating crossover, mutation and selection of PfGA in each local population S'_i. Migration of individuals is possible to occur among all local populations.

Fig. 1.25 Uniformly-distributed (UD) type architecture

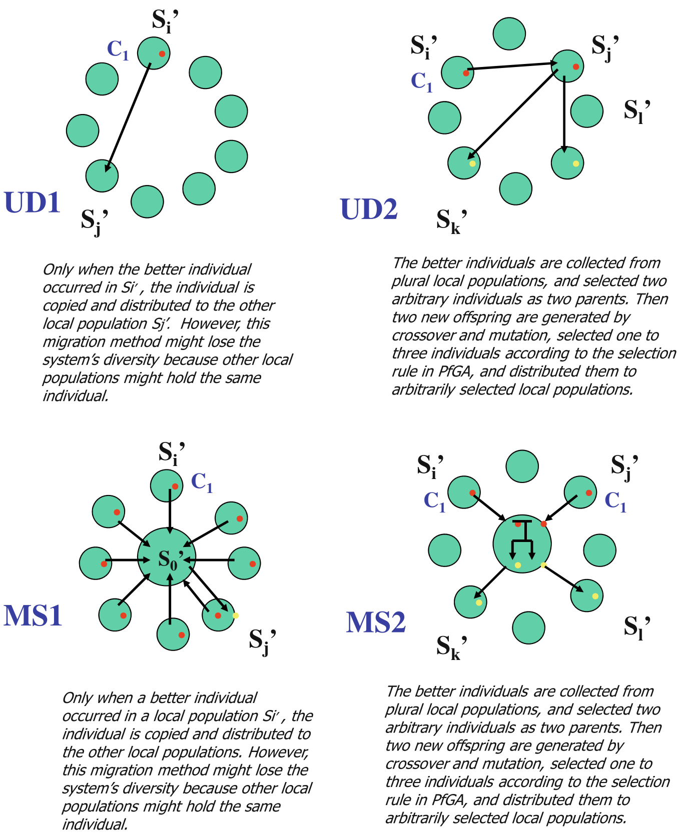


S'_0 is a master local population, and S'_i, i(=1, ..., N) is a slave local population. The master S'_0 is always supervising (or at regular intervals) the best individual in all slave local populations.

Fig. 1.26 Mater–slave (MS) type architecture

1.3.2.2 Two Migration Methods

Several migration strategies may come to mind, but here we adopt the following two. In the first, an individual in a given local population is copied and distributed to other local populations only when a good individual emerges. This is called the direct migration type. The disadvantage of this method is that the same individual is retained by other local populations after migration, threatening system diversity.



Only when the better individual occurred in Si′, the individual is copied and distributed to the other local population Sj′. However, this migration method might lose the system's diversity because other local populations might hold the same individual.

The better individuals are collected from plural local populations, and selected two arbitrary individuals as two parents. Then two new offspring are generated by crossover and mutation, selected one to three individuals according to the selection rule in PfGA, and distributed them to arbitrarily selected local populations.

Only when a better individual occurred in a local population Si′, the individual is copied and distributed to the other local populations. However, this migration method might lose the system's diversity because other local populations might hold the same individual.

The better individuals are collected from plural local populations, and selected two arbitrary individuals as two parents. Then two new offspring are generated by crossover and mutation, selected one to three individuals according to the selection rule in PfGA, and distributed them to arbitrarily selected local populations.

Fig. 1.27 Migration strategy selection method in UD: direct migration type(UD1) and hierarchical migration type (UD2). Migration strategy selection method in MS: Direct migration type (MS1) and hierarchical migration type (MS2)

We adopt the second strategy, in which good individuals are gathered from multiple local populations and two individuals are arbitrarily selected to be the new parents. They bear two offspring (by crossover and mutation), and one to three members of the family are distributed according to the selection rules in PfGA to arbitrarily selected local populations. Since this migration method implements meta-level PfGA operations from the perspective of the local population, it is called the hierarchical migration type (Fig. 1.27).

1.3.2.3 Performance Results

We performed parallel processing for the four different combinations of parallel architecture (uniformly-distributed/master–slave) and migration type (direct/hierarchical) to investigate the effects of migration.

An evaluation of search performance showed that increasing the number of local populations *reduced* the number of evaluations required before success by *a ratio of 1/N* (N: the number of local populations). Among the four types of architecture/migration method examined, search performance, from high to low, fell into the following sequence: UD1, MS1, MS2, UD2. We confirmed that increasing the number of local populations increases the chance of success through the effects of migration relative to serial processing.

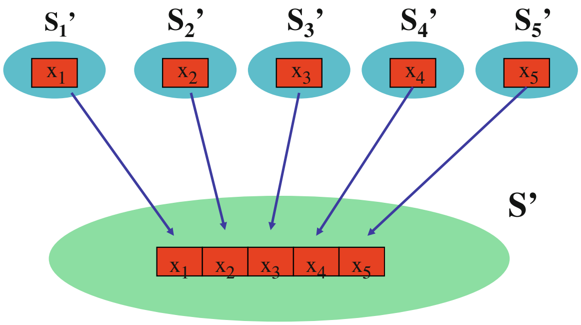
1.3.3 Information Processing Model Based on Gene Duplication [15]

This section discusses the gene-duplicating GA (GDGA) inspired by the theory of *gene-duplication* proposed by Susumu Ohno in the 1970s. The theory of gene-duplication claims that the replication and reuse of gene fragments in the evolution of all organisms, from viruses and plants to animals, fuels a drive toward life-forms of ever-growing sophistication.

Ohno distills this phenomenon into a succinct formula: "*a single creation and one hundred plagiarisms*." [16] Gene duplication is assumed to occur by unequal crossover between chromatids on a single chromosome, unequal crossover between homologous chromosomes during the meiotic process, and partial repetitive duplication of DNA. Inspired by this gene duplication mechanism, we have proposed four gene duplication models: gene concatenating (Fig. 1.28), gene-prolonging (Fig. 1.29), gene coupling (Fig. 1.30), and extended gene coupling (Fig. 1.31).

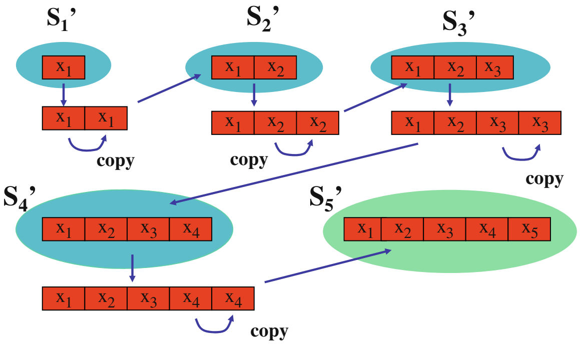
This computational method is based on a *divide-and-conquer* GA in which a given problem is broken down into sub-problems, which are then combined to obtain the solution for the original problem. Each individual concatenates the partial solution that each had accumulated up to that point in time, then the

Fig. 1.28 Gene duplication in a gene-concatenating model



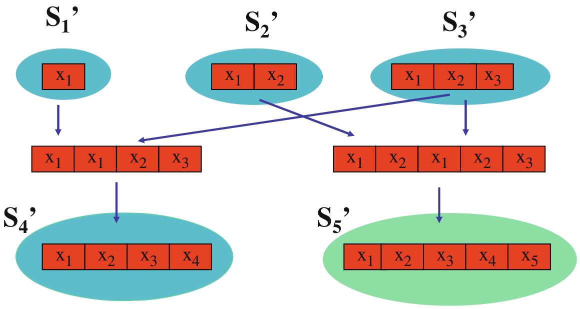
n–dimensional value *(x1, x2, ..., xn) is divided into (x1), (x2), ...(xn) , coded onto a gene, and then, concatenated as a whole.*

Fig. 1.29 Gene duplication in a gene-prolonging model



Low dimensional genes are copied step by step such as (x1), (x1, x2), ...(x1, x2, ..., xn), migrated to other sub-populations, and then prolonged by one dimension by one dimension.

Fig. 1.30 Gene duplication in a gene-coupling model

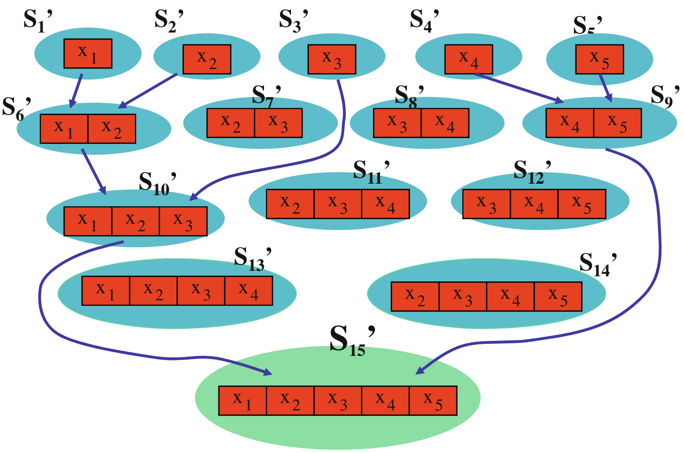


Coupling (x1) with (x1, x2, x3) becomes (x1, x2, x3, x4), and coupling (x1, x2) with (x1, x2, x3) becomes (x1, x2, x3, x4, x5).

individual migrates between the local populations. This strategy makes it possible to obtain the solution more efficiently and quickly.

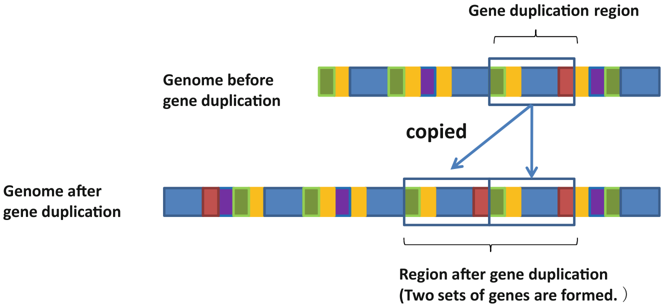
Gene duplication, a powerful tool in solving multi-dimensional functional optimization problems, is a genetic operator applicable to individuals of different gene lengths. It is applied by first coding variables to genes for each sub-dimension, then setting the fitness function for each subspace and running the GA to obtain the (quasi) optimum solution. By concatenating individuals owning the gene corresponding to this (quasi) optimum solution, we can solve an optimization problem in higher dimensions. This algorithm is implemented by having individuals with differing gene lengths migrate between local populations. Overall, the algorithm performs crossover, mutation, and selection within a local population; between separate local populations, it performs duplication and migration, in that sequence.

In a simulation evaluating the search performance of the four types, we used a functional optimization benchmark problem to compare success rates, probabilities, and convergence performance in obtaining the optimum solution. We found that increasing the number of migrating individuals increases population diversity,



By considering the locus of each gene corresponding to each dimension, coupling (x1) with (x2) becomes (x1, x2), and coupling (x1, x2) with (x3) becomes (x1, x2, x3).

Fig. 1.31 Gene duplication in an extended gene-coupling model


Gene duplication region
Genome before gene duplication
copied
Genome after gene duplication
Region after gene duplication
(Two sets of genes are formed.)

The gene duplication region is the region created when living creatures copy genes during the process of evolution. This is believed to be central to the evolution of higher organisms from lower organisms.

Fig. 1.32 Gene duplication

thereby confirming improvements in convergence performance and the effectiveness of this computation method.

Column 4: Theory of Gene Duplication [16 – 18, 34]

Gene duplication refers to the presence of two or more identical genes in the genome or set of genes (Fig. 1.32). Even if one set of duplicate genes undergoes a debilitating mutation that seriously affects its functions, the functions of the living creature proceed unimpaired if the other continues to function normally.

According to "Seimei no Tanjo to Shinka (The Birth and Evolution of Life)" by Susumu Ohno [17], "evolution by gene duplication" [16] can be summarized as follows:

> Since most existing genes have already evolved to a state of near-perfection with regard to function, evolution is impossible if natural selection monitors all genes. True evolution can occur only by acquiring genes with unprecedented new functions. For this to occur, the existing active sites must change. This raises a problem: Natural selection will not permit this. To evade monitoring by natural selection, one must make copies of oneself and make the monitoring focus on one of the copies. This is the only way to acquire such unprecedented new functions.

Gene duplication is one of evolution's mechanisms. The relationship between the original gene and the replicated gene corresponds to the relationship between the leading strand and the lagging strand described in Column 3: The Disparity Theory of Evolution. It also demonstrates the importance of the exquisite balance between the conservation strategy based on guaranteed capital (i.e., the status quo) and an innovation strategy entailing radical risks (since mutations very rarely generate a preferable characteristic) in an overall evolution strategy that avoids extinction. The gene duplicated GA (GDGA) described in the main text, an information processing model modeling the gene replication process of the theory of gene duplication, implements various duplication processes while migrating them between local populations.

1.3.4 Information Processing Model Based on Sexual Selection [19]

The theory of *sexual selection* seeks in part to explain the extensive differences between the phenotype and behavior of the two sexes in certain sexually reproducing organisms. Certain well-known examples include competition between males (generating antlers) and female preferences (generating peacock plumage) [20, 21]. Numerous hypotheses based on female preference seek to explain the evolution of traits that appear disadvantageous from the perspective of natural selection.

Two famous hypotheses are the *runaway hypothesis* and the *excellent gene hypothesis*. The former assumes that female preferences within a population are always biased and that male traits are always variable due to random mutations. In such cases, males with the preferred traits are more likely to father large numbers of offspring, regardless of fitness in terms of natural selection, thereby conferring an indirect advantage to the gene manifesting that trait and resulting in the rise of the male trait preferred by females. The latter hypothesis claims that the male shows off the quality of his genes even if doing so comes at some cost, with the result that the trait and the female preference for that trait grow more common. However, the actual processes at work in sexual selection remain unclear.

The genetic assignment of sex to individuals has led to a state in which an individual of one of the sexes effectively observes and chooses the phenotype of the other, and we will focus on the effect of such asymmetric roles of males and females on the process of evolution.

We will also examine the role of mutation (the simplest transition rule) as a genetic operator that drives organisms toward the direction of evolution based on the fitness landscape. A model is assumed in which a mutation rate is coded in the gene as a parameter for mutation and in which fitness varies.

We will focus in particular on the interaction between sexual selection and mutation.

Working from these perspectives, we introduce an evolutionary computational model in which mutation rates are encoded in the gene and also account for sex and sexual selection. Based on this model, we investigate how mutation rates become self-adaptive within a population and how the direction of evolution is determined in phenotypic space.

Column 5: Sexual Selection [35 – 38]

Male and female individuals of the same species inhabiting a similar natural environment can take significantly different forms. This cannot be explained by natural selection and presented a riddle that puzzled Darwin himself. He focused on the competition between males to win females. In nature, males must compete to gain access to females, a major factor leading to the evolution of characteristics like the splendid horns worn by stags. On the other hand, the beautiful plumage of a peacock (male) is clearly unlikely to have developed to aid in combat between males. This characteristic is believed to help females choose a mate in assessments of courtship displays. Darwin believed females selected males with more beautiful plumage in the male population, resulting in males in the next generation with even more striking plumage. In this manner, in a process called sexual selection, female preference and competition between males would eventually lead to sexual dimorphism, or differences in characteristic between males and females. Although the process whereby sexual selection leads to sexual dimorphism is similar to natural selection, additional energy and resources are needed to create and maintain the horns of the stag or the plumage of the peacock. These characteristics may confer reproductive advantages that offset their disadvantages with regard to natural selection.

1.3.4.1 Constructing a Computational Model

How does an asymmetric relationship between the sexes whereby one sex observes and selects the phenotype of the other affect mutation rates? We propose a model based on sexual reproduction with its own mutation rate encoded into the genes. For the sake of convenience, the observing and observed sex, respectively, are regarded as female and male.

A real-valued genetic algorithm (real-valued GA) is used as the evolutionary computational model. The methods of genetic recombination are chromosomal exchange between individuals and isotropic mutation of each gene.

Sexual selection will focus on relative phenotypic value. (Example: a strong preference for taller individuals or individuals of a certain stronger coloring [e.g., bluer].) In such modeling schemes, the direction of the transition of next-generation males in phenotypic space is determined by the direction of female preferences.

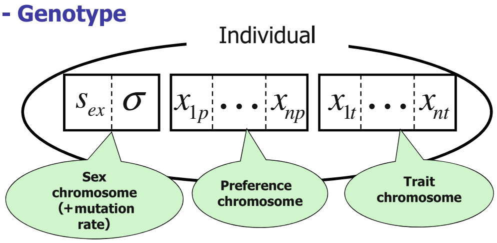
1.3.4.2 Individual Phenotype

Each individual is assigned a sex and with two broad phenotypic categories, *trait* and *preference*. Traits are expressed in both sexes and determine the individual's fitness. In contrast, preferences are expressed only by females and act as a mechanism by which males are assessed and selected. Preferences do not affect natural selection. Traits and preferences are represented in phenotypic space by trait vector $t = \left(x_t^1, x_t^2, \ldots, x_t^n\right)$ and preference vector $p = \left(x_p^1, x_p^2, \ldots, x_p^n\right)$ in n-dimension Euclidean space.

Sexually reproducing organisms are diploid in nature, but here, given the emphasis on interactions due to preference rather than the mode of reproduction itself, the model assumes that the genotype of each individual is haploid for the sake of simplicity, and an individual will have three kinds of chromosomes: sex chromosome, preference chromosome, and a trait chromosome (Fig. 1.33).

The sex gene is coded in a single bit and the other genes are encoded as real values. However, the preference gene in the male is considered not expressed, creating a buffer for male preference and sustaining diversity in preference. Furthermore, we place the gene encoding for the mutation rate (σ) for the trait and

Fig. 1.33 Proposed model of sexual selection



preference genes on the sex chromosome, making the expressed mutation rate dependent on sex.

1.3.4.3 Natural Selection

A known hummingbird species displays sexual dimorphism in beak morphology, with the males and females of the species feeding on flowers of different shapes. This constitutes habitat segregation in the form of resource partitioning. The male and female members of this species can be considered to have followed different paths in natural selection.

Natural selection is posited to operate separately on the sexes. This renders a constant sex ratio while allowing the sexes to generate different traits (sex difference).

1.3.4.4 Sexual Selection

Under sexual selection, a female chooses a male to form a pair, and the offspring produced are one male and one female to maintain the constant sex ratio. All females will always be part of a pair at least once, while males are allowed to pair only when selected by a female as a preferred male. In short, this population is polygamous.

(1) The female i observes M numbers of males at random as potential mates. The average trait vector t_i^0 of the observed male population is calculated, and the relative trait vector $t_{ij} = t_j - t_i^0$ of male j $(j = 1, 2, \ldots M)$ is used as the selection target.
(2) The difference in direction θ_{ij} between t_{ij} and the female preference vector p_i is calculated. The strength of preference is defined as given by $\cos(\theta_{ij})$. The male having a trait vector in the direction closer to the direction of the female preference vector will be strongly favored.
(3) The most preferred male is selected deterministically.

By using the relative trait vector as the selection target, we can search for the direction toward which the male population should shift in phenotypic space—or the direction of evolution—based on the direction of the female preference vector.

In sexual selection, males preferred by more females (i.e., attractive males) gain the advantage, and selection works directly on the male, manifesting as the number of offspring in the next generation. While females in this model are exempt from the direct operation of sexual selection, through genetic exchange with preferred males, the offspring of females having a genetic preference for males with advantages in terms of natural and sexual selection is likely to have the advantage in the next generation. This means sexual selection works indirectly to the advantage of females having genes for such preferences.

1.3.4.5 Genetic Manipulation

Two types of chromosomal exchange and mutation are used as methods of genetic manipulation. With respect to interactions between mutation rate and sexual selection, parent chromosomes are exchanged at a constant probability when parent genes are copied to produce offspring.

Mutations are accomplished by imparting perturbations that follow the normal distribution function $N(0,\ \sigma)$ on the offspring trait and preference genes $x_t^i, x_p^i (\imath = 1, 2, \ldots, n)$, respectively.

Here, the standard deviation σ corresponds to the mutation rate, which is varied adaptively by gene encoding.

$$\sigma = \sigma + \delta \quad \delta \sim \mathrm{N}(0, \sigma_0)$$

Note here that the standard deviation σ_0 of the normal distribution function of the perturbation imparted to σ is constant.

1.3.4.6 Steps in Simulation

(1) Population is initialized at a sex ratio of 1:1.
(2) The following procedure is repeated until the termination condition is satisfied:
   (a) Natural selection
   (b) Sexual selection
   (c) Genetic manipulation

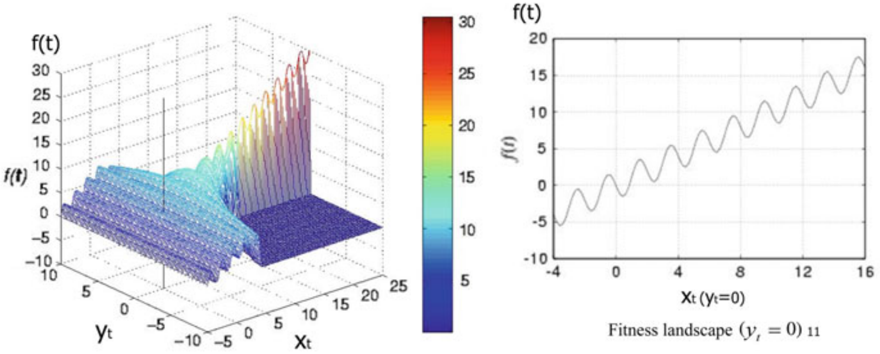
1.3.4.7 Problem

Traits and preferences, respectively, are represented using two-dimensional vectors $t = (x_t,\ y_t)$ and $p = (x_p,\ y_p)$ in examining the following functional maximization problem.

$$\mathrm{Max}\, \mathrm{f}(t) = x_t - 2\sin(\pi x_t) - 0.001 y_t^2 \exp(x_t)$$

The initial population is positioned near the point of origin, while each individual's fitness is determined by trait alone. In the above equation, local optimum solutions may be found on the line $y_t = 0$ with period 2. The problem, then, is to find a way to balance the search for the local optimum solution and the escape from the local optimum solution.

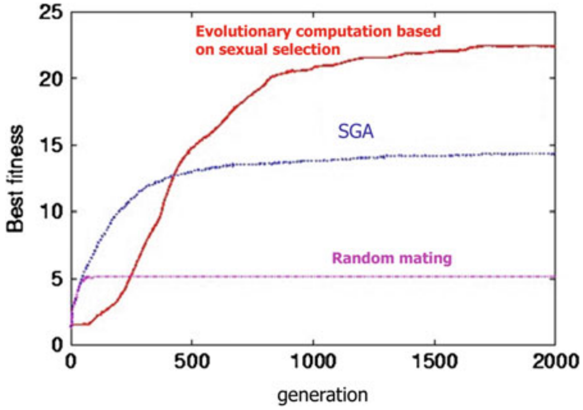
The effects of the third term become large with increasing x_t as the search progresses, and a slight transition of y_t from 0 results in a precipitous decline in fitness, rendering the search more difficult. Although there is no upper limit to the function, GA restricted to isotropic mutation alone places a practical limit on the search. This extremely simple model is an example of how adaptive acquisition of



Fitness landscape (y_t = 0) 11

Fig. 1.34 Validation of the proposed model with an illustrative problem

Fig. 1.35 Experimental results



a direction advantageous to evolution (positive x_t direction) concurrent with a search for the optimum solution makes it possible to search as efficiently as possible. (Fig. 1.34)

Natural selection is executed by performing a roulette-wheel selection based on g, where $g = \exp(\alpha f\,(t))$ (α: scaling rate) when $f(t)$ is fitness. This is selected to make the search progress more smoothly when the exponential term in the above equation starts to have a strong effect on the latter part of the search and when the average fitness of the population has decreased exponentially.

We use preference vector $p = (x_p, y_p)$ as the unit vector to normalize the results after each mutation. This means that the preference selection of a male by a female is based solely on the direction of the trait.

1.3.4.8 Results of Experiment

As we can see from Fig. 1.35, the search in the proposed method progressed the most among the three types of strategies (random mating, SGA, and proposed method), indicating the effects of sexual selection on the search. We see no apparent differences in mutation rates between the sexes for the results of random mating. Under the proposed method, as the search proceeded, the male mutation rate surpassed the mutation rate for females, confirming that female preference triggers unequal mutation rates.

1.3.4.9 Search Process

Under the present method, when the search process becomes trapped in a local optimum solution, a runaway situation between the male trait and female preference results that intermittently drives explosive evolution out of the equilibrium state. As the search progresses, differences appear in average mutation rates between male and female populations and a division of roles arises. The sex exercising choice (female) carries out a conservative search with low mutation rates and the chosen sex (male) carries out an innovative search. This results in only the male population performing the search when escaping from the local optimum solution, while females dedicate themselves to maintaining present conditions and standing by to move on to better solutions through chromosomal exchange only after they have been found by the male population. In many sexually reproducing organisms, the production processes differ for reproductive cells between the sexes, and sperm cells have higher mutation rates than ova. The similarity between the characteristics of organisms and the proposed method is quite interesting. The advantages of balancing an innovative and conservative search during the search process through the acquisition of various mutation rates have been discussed in relation to the Neo-Darwinian algorithm by Wada et al. [13]. Since the error frequency varies between the two strands in DNA, they proposed a model in which the mutation rate varies within a single individual, thereby permitting a wide range of searches. This realizes a broad range of mutation rates within the population, allowing them to perform an extensive search in problems involving a high level of risk while maintaining present conditions.

1.4 Information Processing Based on the Modeling of Cells in Early Stage of Evolution

1.4.1 Chemical Genetic Algorithm (CGA) [22]

The mechanism of cell metabolism emerged over an astoundingly lengthy evolutionary process. Modeling this process should make it possible to search efficiently for an optimum solution by techniques totally different from conventional methods. This section discusses the *chemical genetic algorithm* (*CGA*), used for solving difficult problems by dynamically converting them into simpler

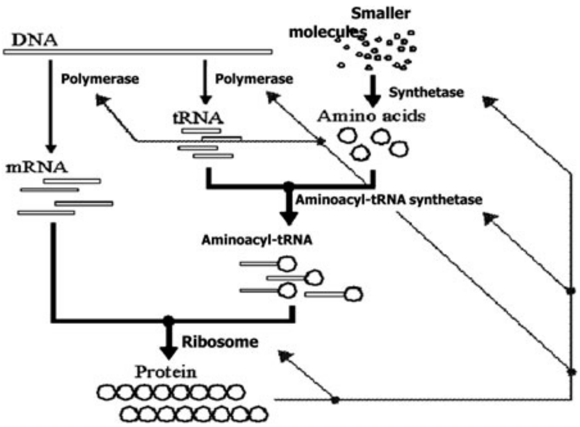


Fig. 1.36 Biochemical reaction for translating genetic information in a cell

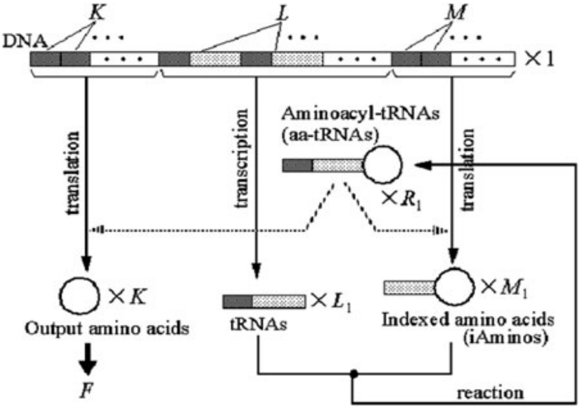


Fig. 1.37 A cell structure used in the CGA

problems—in short, by dynamically changing mapping from the genotype to the phenotype, as inspired by the mechanism of cell metabolism in the early stages of the evolutionary process. The section will also discuss *chemical genetic programming* (*CGP*), in which this solution method is applied to the evolution of programming, and introduce the problem of symbolic regression in artificial intelligence and describe the results of its application to the acquisition of multiagent behavioral strategy.

1.4.1.1 Mechanisms of Cell Metabolism Generated in the Early Stages of the Evolutionary Process

In the early stages of cell evolution, cells are believed to have acquired their present metabolic processes by dynamically changing the mapping behavior from genotype to phenotype mapping (Fig. 1.36). Figure 1.37 presents a model based on this mechanism.

1.4.1.2 CGA Generation Cycle (See Fig. 1.38)

The steps in the CGA generation cycle are given below:

1. Initialization: First, we prepare a number, *N*, of cells having the structure presented in Fig. 1.37. In the initial state, no cell possesses aminoacyl tRNA (aa-tRNA), tRNA, or outputs amino acids. However, they do have random DNA strands and amino values.
2. Chemical reaction: The following 4-step reaction takes place in all cells: transcription, tRNA-amino acid reaction, translation into internal amino acid, and translation into output amino acid. In the several generations of the early stage, this reaction produces new tRNA and aa-tRNA, and their sizes grow. Within the next few generations, we exceed the size of the molecular pool size.
3. Selection: The fitness of the cell is calculated based on the output amino acid, and cells marking high fitness are selected by roulette-wheel selection. The selected cells are regenerated, and the complete internal information (DNA, 3 molecular pools) of each cell is copied to the daughter cell.
4. DNA mutation: As in normal GA, point mutations of genes are performed.
5. DNA crossover and molecular exchange between cells: Gene crossovers occur as in normal GA, and half the molecules are exchanged between two cells.
6. Calculation of fitness of the cell population: If the termination conditions are satisfied, the computation is complete. If not, return to step 2.

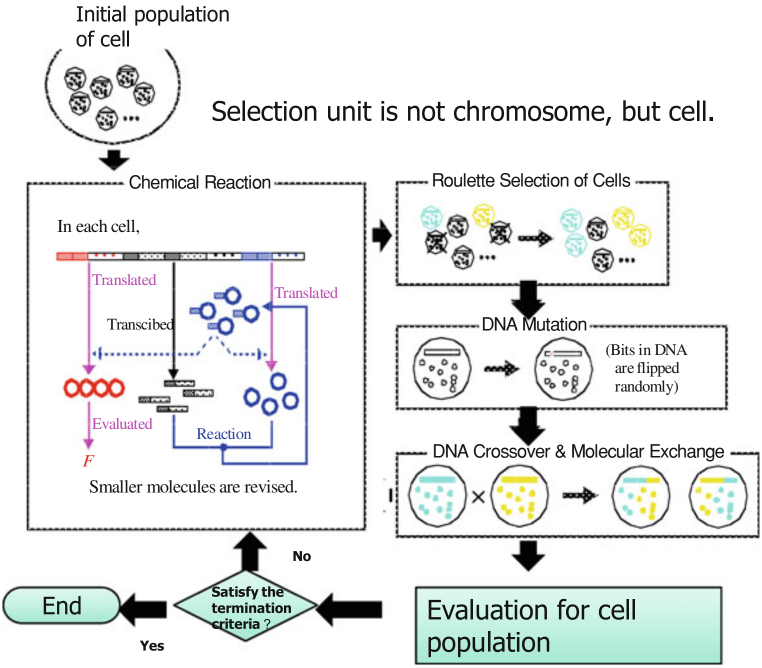

Initial population of cell
Selection unit is not chromosome, but cell.
Chemical Reaction
In each cell,
Translated
Transcibed
Translated
Evaluated
Reaction
F
Smaller molecules are revised.
Roulette Selection of Cells
DNA Mutation
(Bits in DNA are flipped randomly)
DNA Crossover & Molecular Exchange
×
Evaluation for cell population
No
Satisfy the termination criteria ?
Yes
End

Fig. 1.38 Entire algorithm of CGA

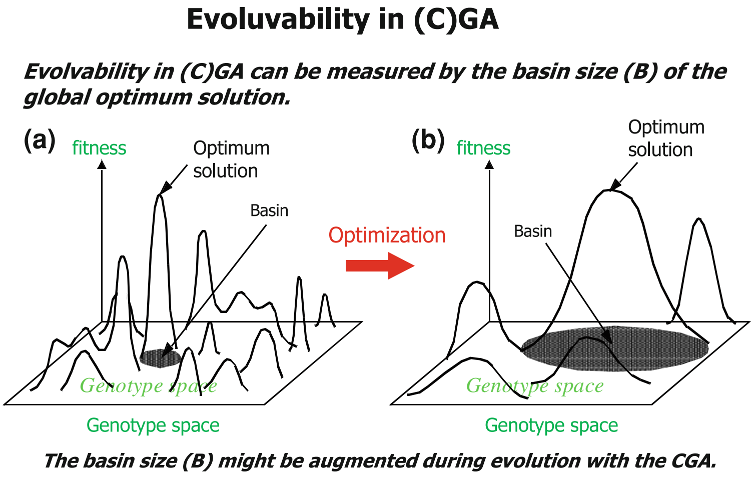
1.4.1.3 GA Evolvability (See Fig. 1.39)

By converting the "ragged fitness landscape" as seen in the genotype space presented on the left side of Fig. 1.39a into a smoothed landscape on the right side (Fig. 1.39b), we can improve *evolvability*.

Three types of deception problems (Types I, II, and III) and 2 benchmark functions were used to validate the search performance of CGA. SGA (simple GA) and PfGA (Parameter-free GA) are used for comparisons. In the simple Type I, $F(x)$ assumes the maximum value (optimum value) of 1 when $x_k = 1$ in all dimensions of $k(k = 1,\ldots,K)$. Type II is an intermediate complex type, and $F(x)$ assumes the maximum value of 1 only when, in each dimension k, $f_k(x)$ randomly takes a maximum at $x_k = 0$ or $x_k = 1$. In Type III, $F(x)$ assumes the maximum value of 1 only when $f_k(x)$ assumes a maximum of 1 at $x_k = \alpha_k$ (α_k is a uniformly random number between 0 and 1) in each dimension of k. As can be seen from Fig. 1.40, the ratio of the probability of $f(x)$ assuming a maximum (optimum value) of 1 in each dimension to the probability of taking the localized optimum value of 0.8 is 1:4. Thus, the probability of $f(x)$ taking optimum values at all dimensions $(k = 1,\ldots,K)$ is $(1/5)^K$. Types I and II constitute special cases of Type III (Table. 1.3).

1.4.1.4 Results of Analysis

Figures 1.41 and 1.42 present the evolution of the function $F(x)$ of CGA and SGA, respectively. The dispersion of the function values is large for CGA. In contrast, the dispersion is small for SGA. The optimum value of CGA (CGA best) is 0.8 or



Evoluvability in (C)GA

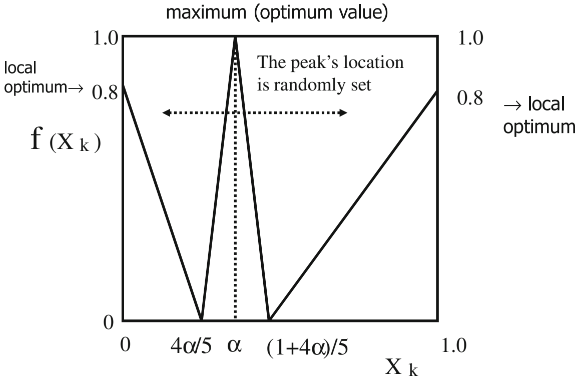
Evolvability in (C)GA can be measured by the basin size (B) of the global optimum solution.

The basin size (B) might be augmented during evolution with the CGA.

Fig. 1.39 Evolvability in C(GA)

Table 1.3 Three deceptive Problems

Type	Name	Features
I	Simple type	$f(x)$ becomes maximum (optimum) when $x_k = 1$ for all dimensions $k(k = 1, 2, \ldots, K)$. This type is a special case of type III.
II	Intermediate type	$f(x)$ becomes maximum (optimum) only when $f_k(x)$ becomes 1 at x_k = 1 or 0 randomly for all dimensions $k(k = 1, 2, \ldots, K)$. This type is a special case of type III.
III	Complex type	$f(x)$ becomes maximum (optimum) only when all $f_k(x)$ become 1 at $x_k = \alpha_k$ (where α_k are different uniform random values between 1 or 0) for all dimensions $k(k = 1, 2, \ldots, K)$. As the ratio that $f_k(x)$ takes the maximum value of 1 and the value of 0.8 for each dimension, is 1–4, the probability that takes the optimum value 1 is $(1/5)^K$. This type includes the type I and II as its special cases.



The "deceptive problem" is easily trapped by a local optimum, and cannot reach the global optimum on purpose using a usual naïve search algorithm. In type III, the peak value 1 at *Xk=αk* ,(*k*=1, ..., *K)* is global optimum, while the value 0.8 at *Xk=0,1,* (*k*=1, ..., *K)* is local optimum *(i.e., it is deceived as the optimum value).* The function F(X) to be optimized is defined as the following equation:

$$F(X)=\{(1/K)\,\Sigma\, f_k\,(x)\}^{\beta} \quad , \text{ where } \quad K=5, \text{ or } 10, \beta=5 .$$

Fig. 1.40 Complex deceptive problem (type III)

higher and attains the status of the optimum solution. On the other hand, the optimum value of SGA fails to surpass 0.8 and is trapped in a local optimum solution.

Figures 1.43 and 1.44 present the time series of amino value histograms for CGA and SGA, respectively. The results are for a 5-dimensional Type III deception problem. We see that for all α_k assuming maximum values at each dimension of k, the amino value for CGA exceeds a certain value. In contrast, for SGA, the amino value exceeds a certain value in certain dimensions but not in others. This is a typical example of a search that has fallen into a local optimum solution.

Fig. 1.41 Evolution of CGA for deceptive problem (type III)

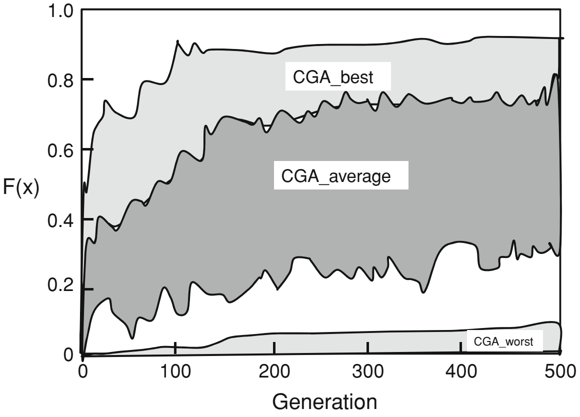
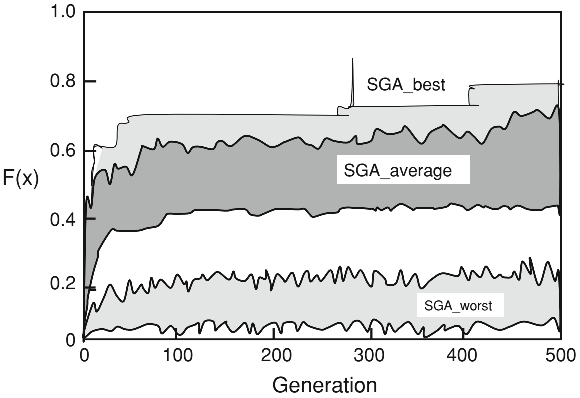


Fig. 1.42 Evolution of SGA for deceptive problem (type III)



1.4.1.5 Performance Comparison Among SGA, CGA and PfGA

Tables 1.4 and 1.5 present comparisons of performance evaluation results for SGA, CGA, and PfGA.

Table 1.4 gives the results for deception problems. A comparison of the success rates of CGA and SGA shows that the performance of CGA far outpaces SGA. Furthermore, PfGA is 100% successful for all types of deception problems. Table 1.5 gives the results for benchmark problems (Shekel's foxhole problem, Langerman function), and we see that CGA gives success rates comparable to PfGA.

Figure 1.45 shows the changes in basin size in the case of CGA. The figure shows a sudden increase in basin size at a certain point in the evolution (100 generations), considered to reflect the achievement of punctuated equilibrium associated with the transition stage in the evolution, mapping from genotype (binary value) to phenotype (function value).

Figure 1.46 presents the values of the codon-amino acid translation table. Under the initial conditions of evolution, variations in the values in the table

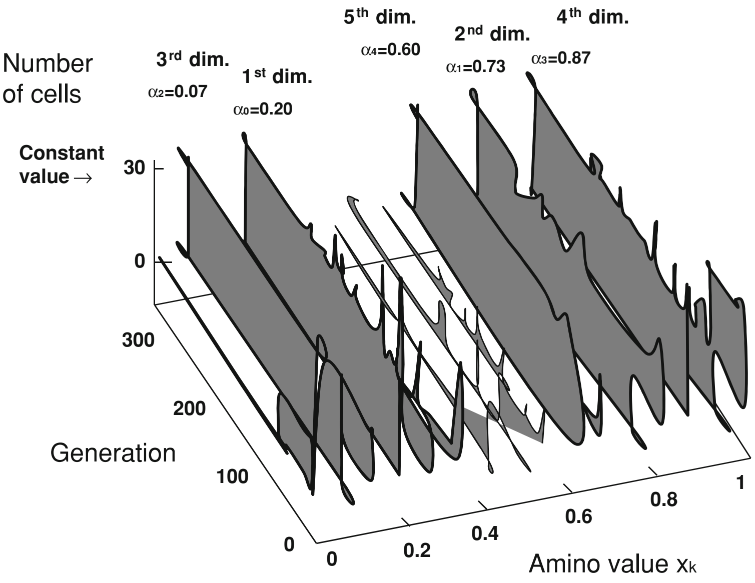


Fig. 1.43 Time series of amino value histogram for CGA

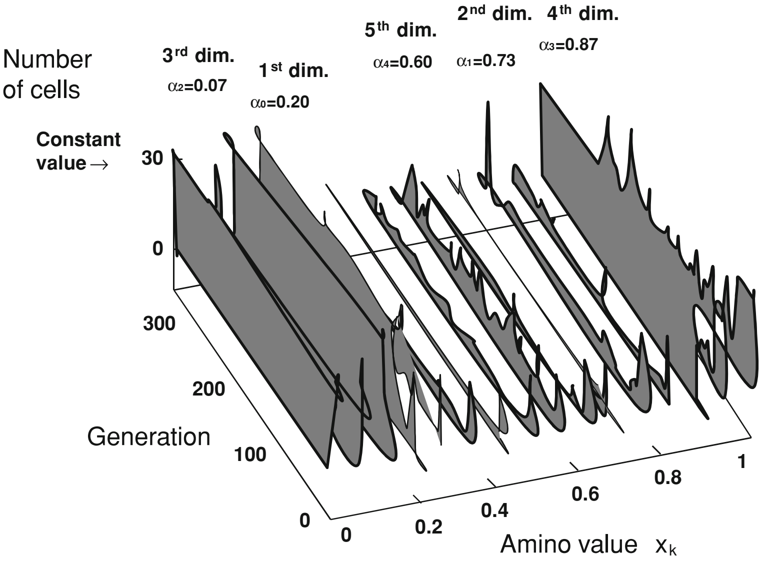


Fig. 1.44 Time series of amino value histogram for SGA

appear to be large changes in amino value per single bit of change in the codon (lower right). But as evolution progresses, the amino value changes only gradually relative to the change in codon value. This corresponds to the evolution (transition)

Table 1.4 Success rate for deceptive problems

GA	SGA	SGA*	CGA	PfGA	SGA	SGA*	CGA	PfGA
Dimension	5	5	5	5	10	10	10	10
Codon length	6	6	6	20	6	6	6	20
Scaling	Linear	Linear	Exponential	None	Linear	Linear	Exponential	None
Type I	2%	9%	100%	100%	0%	3%	100%	100%
Type II	12%	20%	100%	100%	0%	5%	100%	100%
Type III	47%	85%	95%	100%	14%	79%	43%	100%

Table 1.5 Success rate for benchmark problems

GA	SGA	SGA*	CGA (WF)	CGA (NF)	PfGA	SGA	SGA*	CGA (WF)	CGA (NF)	PfGA
Dimension	5	5	5	5	5	10	10	10	10	10
Codon length	6	6	6	6	20	6	6	6	6	20
Scaling	Linear	Linear	Exponential	Exponential	None	Linear	Linear	Exponential	Exponential	None
Shekel	5%	5%	5%	50%	37%	0%	0%	0%	0%	1.3%
Langerman	41%	47%	13%	35%	83%	0%	0%	0%	3%	1.7%

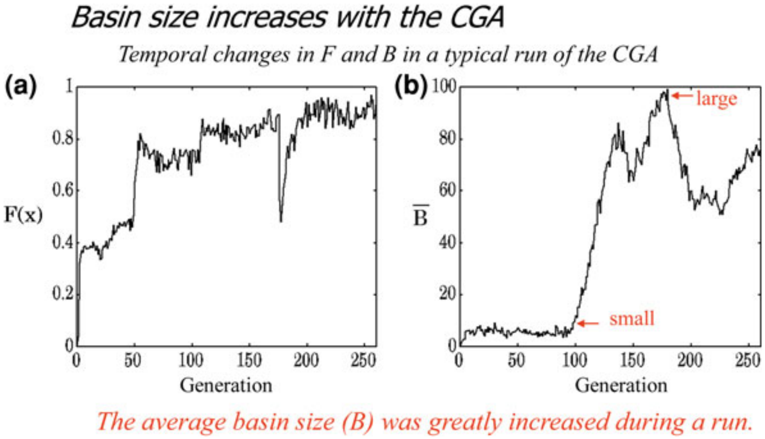
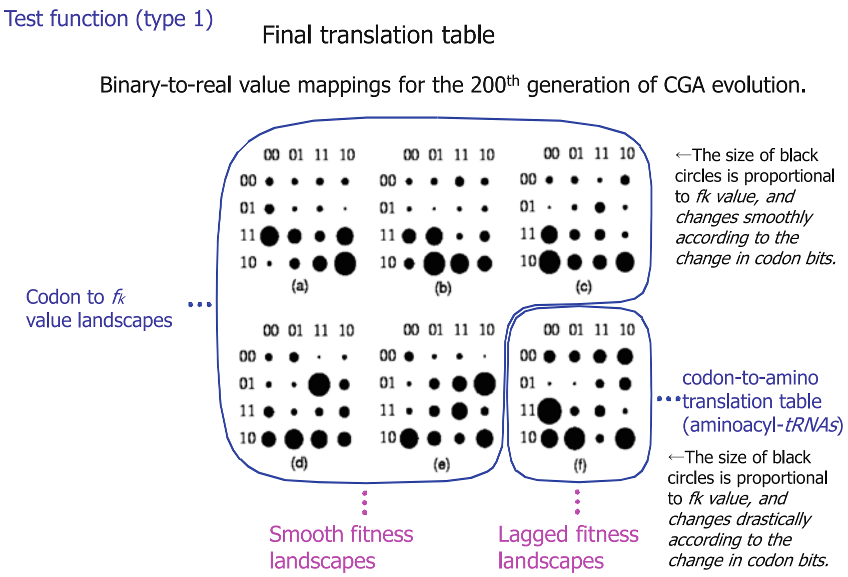


Fig. 1.45 Increase in basin size for CGA

of the landscape represented by the left plot of Fig. 1.39 into the smooth landscape represented by the right plot. The evolvability of CGA has greatly increased, indicating that the algorithm generates a solution method for difficult problems in an evolutionary manner, while automatically (in evolutionary fashion) converting difficult problems into easier problems.

The method for improving mapping techniques from genotype to phenotype through the evolutionary process is a highly generalizable optimization technique. The following section discusses a method for expanding CGA to genetic programming (CGP).


Test function (type 1)

Final translation table

Binary-to-real value mappings for the 200th generation of CGA evolution.

←The size of black circles is proportional to *fk value, and changes smoothly according to the change in codon bits.*

Codon to *fk* value landscapes

codon-to-amino translation table (aminoacyl-*tRNAs*)

←The size of black circles is proportional to *fk value, and changes drastically according to the change in codon bits.*

Smooth fitness landscapes

Lagged fitness landscapes

Fig. 1.46 Final translation table in binary-to-real value mapping for CGA

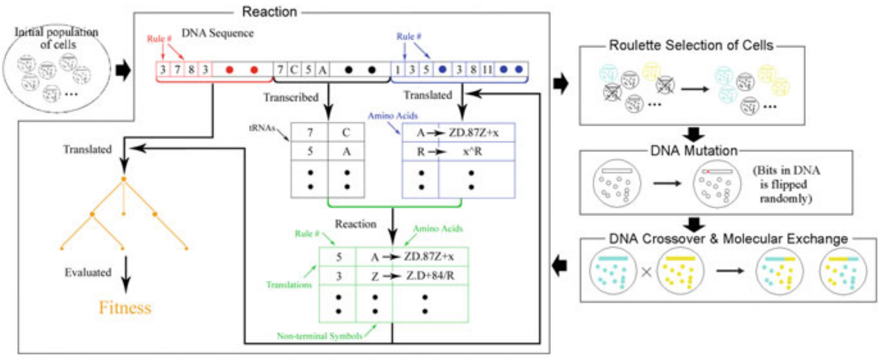


Fig. 1.47 CGP algorithm

1.4.2 Chemical Genetic Programming (CGP) [23]

Figure 1.47 shows how CGA is expanded to CGP. A comparison of Figs. 1.38 and 1.47 shows how the genes (DNA sequence) in CGA are converted into a combination of the rewriting rule numbers and the left sides of the rewriting rules. DNA is translated into protein synthesized by concatenating amino acids, after which fitness is calculated. At this point, the other portions of the DNA have been transcribed into tRNA or translated into amino acids. Aminoacyl tRNA, produced by their reaction, acts as a catalyst in this translation process. Modeling these metabolic processes inside the cell results in the evolutionary generation of the rewriting rule itself, and generates rules completely different from those in the initial state to make it possible to acquire rules with higher fitness scores.

Column 6: Genetic Programming

The procedures for genetic programming (GP) are basically the same as those for genetic algorithms (GA). However, while the genotype of the candidate solution in GA is a bit sequence or a real value sequence, it is a tree structure in GP. The tree structure in GP consists of a terminal set (a set of terminal symbols) containing independent variables and constants and a non-terminal set (also called a function set; a set of non-terminal symbols) containing functions and the four arithmetic operations of addition, subtraction, multiplication, and division. GP randomly generates an initial population, then evaluates the fitness of the population to select individuals. Next, as shown in Fig. 1.48, crossovers occur at probability *Pc* between a pair of individuals arbitrarily selected from the population. In the example shown in this figure, the terminal set is $\{x, y, z, a, b, c\}$, and the function set is {+, −, *, %, sin, cos, exp}. The two parents—Parent 1: $(x - a)/z + \sin(b*y)$ and Parent 2: $\cos(y + c)*(x - \exp(z))$—are each represented by a tree structure, and subtrees $b*y$ and $x - \exp(z)$ are stochastically selected for the crossover. The end result are the two offspring Offspring 1: $(x - a)/z + \sin(x - \exp(z))$ and Offspring 2: $\cos(y + c)*(b*y)$.

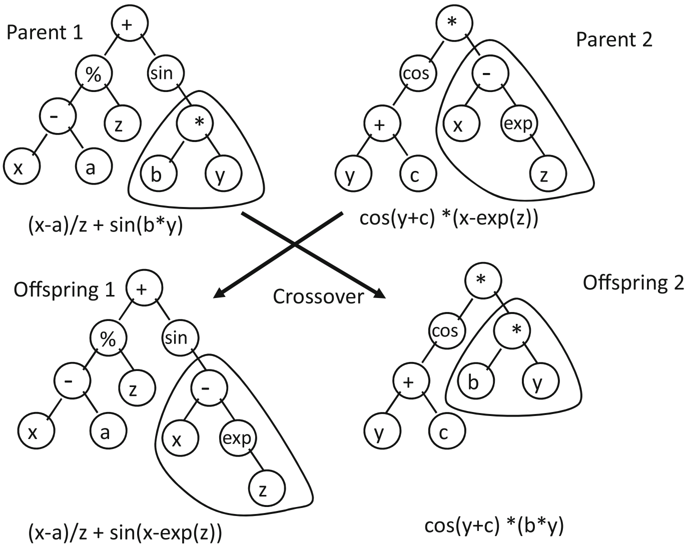


Fig. 1.48 Crossover in genetic programming

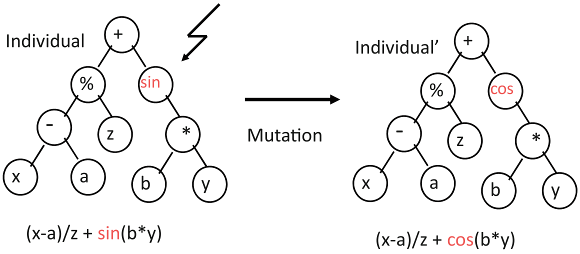


Fig. 1.49 Mutation in genetic programming

Next, as shown in Fig. 1.49 in this column, a mutation occurs with probability Pm in the non-terminal node representing sin. The function in the node changes from sin to cos, and the phenotype of the individual also changes from $(x - a)/z + \sin(b*y)$ to $(x - a)/z + \cos(b*y)$. Later, the population is evaluated. If the solution obtained is satisfactory, the optimum solution is output, and processing ends. If not, GP repeats the series of genetic manipulations above while advancing from generation to generation.

Genetic programming is widely applied as an approach to solving real-world problems in various areas of artificial intelligence, including symbolic regression problems, system identification, optimization control, planning, time series predictions, automatic programming, discovery of game strategies, solutions for inverse problems, rules discovery, pattern classification, evolution of emergent behavior, automatic programming of cellular automata, and evolvable hardware (EHW).

Chemical genetic programming (CGP), described in the main text, has expanded and developed from GP and can adapt to various environments by dynamically changing the mapping from the genotype to the phenotype. In other words, since feedback from the phenotype to the genotype makes it possible to maintain diversity in a population with a small number of individuals, it requires smaller populations than ordinary GP. This reduces the computational loads and memory required and increases the range of evolutionary possibilities.

1.4.2.1 Example of Application 1: Symbolic Regression Problems [23]

Figure 1.50 compares the fitness evolution curve in CGP and conventional GE (grammatical evolution). We see that evolution proceeds faster in CGP than GE, resulting in a good solution after 140 generations. Furthermore, even though the best solution appears faster, the average fitness of the population consistently remains below the best value, indicating that population diversity is sustained.

Figure 1.51 compares the solution generated by CGP and GE. For the target function $2x^6 + 3x^4 + 4x^2 + 100$, CGP gives $2x^6 + 501$, while GE gives $1.9x^6$. Since the normalized fitness values are 0.95 and 0.81 for CGP and GE, respectively, we may conclude that CGP is superior.

- *fitness= exp (a*exp(-d/b)),* *a=8, b=5*10⁷*
- Good solution found after 140 generations.
- Population average remains significantly lower than best individual.
  i.e., population diversity is maintained.

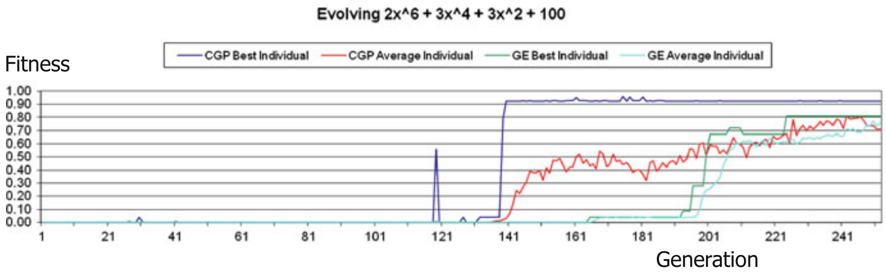


Fig. 1.50 Evolution curve of fitness

Generated formulas

- Target: $2x^6+3x^4+4x^2+100$
- CGP: $2x^{6}+501$ normalized fitness=0.95
- GE: $1.9*x^6$ normalized fitness=0.81

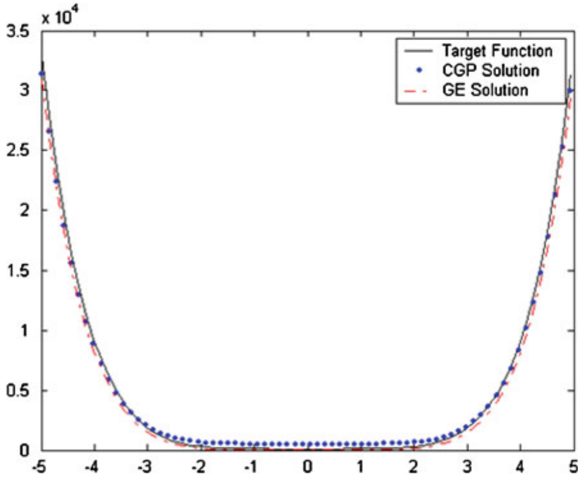


Fig. 1.51 Generated function: CGP vs. GE

1.4.2.2 Example of Application 2: Behavioral Strategy of Agents [24]

Figure 1.52 applies CGP to a multi (2)-agent problem, the game of "tag." Two agents use CGP to generate in *co-evolutionary* fashion a behavioral strategy for catching the other as quickly as possible, or to evade the other for as long as possible.

Table 1.6 is a list of basic functions used in CGP.

Figure 1.53 shows the two agents in motion. S denotes the starting point. The objects numbered 1 and 2 in the center are obstacles. We see how the two agents (pursuer and evader) skillfully avoid the obstacle in the chase. Table 1.7 presents

Fig. 1.52 The game of tag

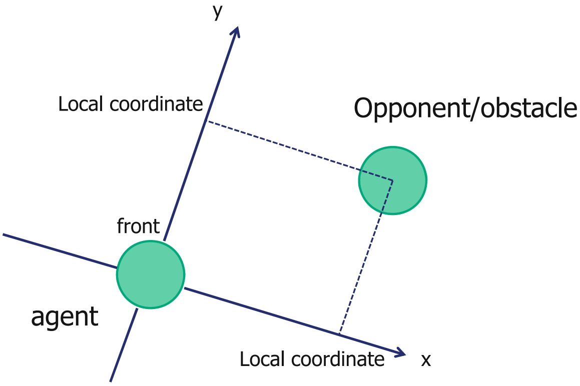
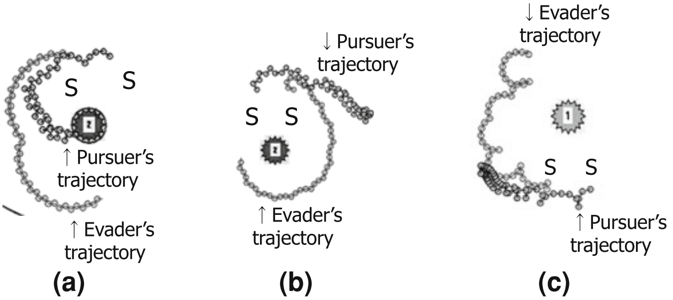


Table 1.6 List of basic functions

Function	Usage	Description
+	(+ *a* *b*)	Addition: $a + b$
−	(− *a* *b*)	Subtraction: $a - b$
*	(* *a* *b*)	Multiplication: $a * b$
%	(% *a* *b*)	If $b = 0$, then 1, else $a \div b$
Min	(min *a* *b*)	If $a < b$, then *a*, else *b*
Max	(max *a* *b*)	If a > *b*, then *a*, else *b*
Abs	(abs *a*)	Absolute value of *a*: $\lvert a \rvert$
Neg	(neg *a*)	Negative value of *a*: $-a$
Iflte	(iflte *a* *b* *c* *d*)	If $a \leqq b$, then *c*, else *d*



Snap-shot at the 200th generation, S: starting points.
(a):The pursuer strikes against the obstacle 2(representing the circle).
(b)and(c):The evader escapes from the pursuer avoiding the obstacles.

Fig. 1.53 Results of agent behavior

the behavioral strategies generated by the agents. Although the details are not given here, we can appreciate how both agents acquire various behavioral strategies through *co-evolutionary* generation.

Table 1.7 Emergent strategies

Emergent strategy	Generation	Player
Tagging	200–450	Pursuer
Weaving	200–450	Pursuer
Safe haven	200–450	Evader
Obstacle hugging	200–450	Evader
Boundary diversion	$\sim$450	Evader
Focused tagging	$\sim$450	Pursuer
Point circling	$\sim$500	Evader
Arcing	$\sim$1000	Evader
Shorting	$\sim$1000	Pursuer

References

1. H. Simon, *Neural Networks and Leaning Machines*, 3rd edn. (Prentice Hall, Upper Saddle River, 2008)
2. M.A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks* (MIT Press, Cambridge, 2006)
3. R.S. Sutton, G.B. Andrew, *Reinforcement Learning—An Introduction (Adaptive Computation and Machine Learning)* (MIT Press, Cambridge, 1998)
4. A. Waibel, in *Phoneme Recognition Using Time-Delay Neural Networks,Report of Speech Committee SP87-100* (December 1987), pp. 19–24
5. A. Waibel, H. Sawai, K. Shikano, Modularity and scaling in large phonemic neural networks. IEEE Trans. ASSP **37**(12), 1888–1898 (1989)
6. S. Nakamura, H. Sawai, Performance comparison of neural network architecture for speaker-independent phoneme recognition. Trans. IEICE D-II **J74-D-II**(10), 1361–1369 (1991)
7. K. Fukuzawa, Y. Komori, H. Sawai, M. Sugiyama, A Segment-Based Speaker Adaptation Neural Network Applied to Continuous Speech Recognition. in *Proceeding of the ICASSP'92*, San Francisco, vol. 1 (March, 1992), pp. 433–436
8. H. Sawai, Frequency-Time-Shift-Invariant Time-Delay Neural Networks for Robust Continuous Speech Recognition. in *Proceedings of the ICASSP'91, Toronto, S2.1*, vol. 1 (May 1991), pp. 45–48
9. H. Sawai, Axially Symmetric Neural Network Architecture for Rotation-Invariant Pattern Recognition. in *IEEE, WCCI (World Congress on Computational Intelligence)*, *Proceedings of the IEEE ICNN'94*, vol. 7 (June–July 1994), pp. 4253–4258
10. C. Darwin, *On the Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life* (John Murray, London, 1859)
11. J.H. Holland, *Adaptation in Natural and Artificial System* (The University of Michigan Press, Ann Arbor, 1975)
12. H. Sawai, S. Kizu, T. Endo, Parameter-free genetic algorithm (PfGA), Tech. Rep. IEICE **J81-D-II**(2), 450–452 (1998)
13. K. Wada, H. Doi, S. Tanaka, Y. Wada, M. Fu-rusawa, A neo-Darwinian algorithm: asymmetrical mutations due to semi-conservative DNA-type replication promote evolution. Proc. Natl. Acad. Sci. USA **90**, 11934–11938 (1993)
14. S. Adachi, H. Sawai, Effects of migration methods in parallel distributed parameter-free genetic algorithm. Trans. IEICE D-I **J83-D-I**(8) 834–843 (2000)
15. S. Adachi, H. Sawai, Evolutionary computation method inspired by gene-duplication: application to functional optimization. J. IPSJ **42**(11), 2663–2671 (2001)
16. S. Ohno, *Evolution by Gene Duplication* (Springer, New York, 1970)
17. S. Ohno, *Seimei no Tanjo to Shinka (The Birth and Evolution of Life)* (University of Tokyo Press, Tokyo, 1988)

18. S. Ohno, *Ooinaru Kasetsu—DNA karano Messeji (The Great Hypothesis: Messages from DNA)* (Yodosha, Tokyo, 1991)
19. K. Omori, Y. Fujiwara, S. Maekawa, H. Sawai, S. Kitamura, Evolutionary computation based on sexual selections driving asymmetrical mutation. J. Inst. Syst. Control Inf. Eng. **15**(8), 422–429 (2002)
20. M. Hasegara, *Why is the Male Peacock Beautiful?* (Kinokuniya Shoten, Tokyo, 1992)
21. Y. Iwasa, *Introduction to Mathematical Biology—Investigating the Dynamics in Biological Society* (Kyoritsu Shuppan, Tokyo, 1998)
22. H. Suzuki, H. Sawai, W. Piaseczny, Chemical genetic algorithms—evolutionary optimization of binary-to-real-value translation in GAs. J. Artif. Life **12**, 1–27, (2006)
23. W. Piaseczny, H. Suzuki, H. Sawai, Chemical genetic programming—evolutionary optimization of the genotype-to-phenotype translation set. J. Artif. Life Robot. **9**, 202–208 (2006)
24. C.S. Chan, J.C. Tay, H. Sawai, Discovering Pursuit and Evasion Strategies for a Bounded Environment with Two Obstacles. in *Proceedings of the Third International Conference for Computational Intelligence*, *Robotics and Autonomous systems* (*CIRAS 2005*) (Singapore, 13–16th December, 2005)
25. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **7**, 115–133 (1943)
26. D. Rumelhart, J. L. McCleland, et al., *Parallel Distributed Processing*, vol. 1 (MIT Press, Cambridge, 1986)
27. K. Wada, Dejitaru Seimei no Shinka (The Evolution of Digital Life). in *Iwanami Kagaku Raiburari 11 (Iwanami Chemistry Library 11)* (Iwanami Shoten, Tokyo, 1994)
28. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley Professional, New York, 1989)
29. D.E. Goldberg, *Genetic Algorithms: The Design of Innovation* (Springer, New York, 2009)
30. H. Sawai, S. Kizu, in *Parameter-free Genetic Algorithm Inspired by Disparity Theory of Evolution*. Lecture Notes in Computer Science, vol. 1498 (Springer, New York, 1998)
31. M. Furusawa, H. Doi, Promotion of evolution: disparity in the frequency of strand-specific misreading between the lagging and leading DNA strands enhances disproportionate accumulation of mutation. J. Theo. Biol. **157**, 127–133 (1992)
32. Chiba Bio Network, http://www.chibabio.net/comp/comp_07.html
33. M. Furusawa, Shinka wa Isshun de Okiru—Tainai ni Yadoru Shinka no Chikara (Evolution Occurs in an Instant—Evolutional Power inside the Body), *Seimeishi* (*Biohistory*), no. 31 (2001). http://www.brh.co.jp/seimeishi/1993-2002/31/ss_5.html
34. Seibutsugaku Jiten (Dai 4 Han) Idenshi Chofuku (*Dictionary of Biology*, 4th edn.), Gene Duplication), (Iwanami Shoten, Tokyo, 2000), p. 77
35. C. Darwin, *The Descent of Man and Selection in Relation to Sex*, Volume 1 of the Japanese translation by Mariko Hasegawa, Ningen no Shinka to Sei Tota I (Bun-ichi Co. Ltd., 1999, Japan)
36. C. Darwin, *The Descent of Man and Selection in Relation to Sex*, Volume 2 of the Japanese translation by Mariko Hasegawa, Ningen no Shinka to Sei Tota II (Bun-ichi Co. Ltd., Japan, 2000)
37. M. Hasegawa, *Osu to Mesu no Kazu wo Meguru Fushigi (Wonders about Numbers between Males and Females*) (NTT Publishing Co. Ltd., Tokyo, 1996)
38. M. Hasegawa, *Osu to Mesu—Sei wa Naze Aruka* (*Males and Females—Why Sex Exists*) (NHK Human University, Japan, 1997)

Chapter 2
Molecular Communication Technology as a Biological ICT

Tadashi Nakano, Michael Moore, Akihiro Enomoto and Tatsuya Suda

Abstract This chapter provides a comprehensive overview of state-of-the art research on molecular communication—a molecule-based communication paradigm for biological machines. Unlike current telecommunications based on electric or optical signals, molecular communication exploits biological molecules as information carriers. In molecular communication, senders of communication encode information onto molecules and transmit to the environment. The information coded molecules then propagate in the environment to reach receivers of communication, which capture and biochemically react to the molecules (i.e., decode the information from the information coded molecules). Since biological molecules are compatible with biological systems, molecular communication is expected to impact medical domains such as human health monitoring where implant biological machines interact with biological cells through molecular communication. This chapter describes key concepts, architecture, potential applications of molecular communication as well as existing research on engineering molecular communication components and systems.

Keywords Molecular communication · Nano-networks · Nanomachines

T. Nakano (✉)
Osaka University, Osaka, Japan
e-mail: tnakano@wakate.frc.eng.osaka-u.ac.jp

M. Moore, A. Enomoto and T. Suda
University of California, Irvine, USA

H. Sawai (ed.), *Biological Functions for Information and Communication Technologies,* Studies in Computational Intelligence, 320,
DOI: 10.1007/978-3-642-15102-6_2, © Springer-Verlag Berlin Heidelberg 2011

2.1 Introduction

Biological nanomachines (or “nanomachines,” for short) are nanoscale to microscale devices that either exist in the biological world or are artificially created from biological materials and that perform simple functions such as sensing, logic, and actuation. Example nanomachines from the biological world include biological cells [4], molecular motors that produce mechanical work using chemical energy [34], and any biochemical molecules, complexes, circuits that act as processing units [10] while examples of synthetic biological nanomachines are artificial molecular machines [33] and genetically engineered biological cells that are programmed to produce intended functions [13, 17, 21, 54].

Molecular communication is a new paradigm for communication between nanomachines over a short (nanoscale or microscale) range [27]. In molecular communication, information is encoded to and decoded from molecules, rather than electrons or electromagnetic waves. Using electrons or electromagnetic waves for communication is particularly limited at the nanoscale and microscale range because of power constraints and physical limitations in the size and nature of communication components (i.e., biological nanomachines). Because nanomachines are too small and simple to communicate using electrons or electromagnetic waves, molecular communication provides a novel mechanism for nanomachines to communicate by propagating molecules that represent information.

A promising area that molecular communication contributes to is medical domains [38]. Imagine that a biological system (e.g., the human body) is composed of a large number of cells (i.e., nanomachines), that each cell performs simple and specific operations such as uptake, processing, and release of molecules, and that cells interact to perform various functions of the body (e.g., distributing molecules for metabolism and replication of cells). Molecular communication provides mechanisms to transport molecules between cells, and thus, it may help perform targeted drug delivery by providing mechanisms to transport drugs (information-encoded molecules) between drug repositories embedded in a human body (sender nanomachines) and specific cells in a human body (receiver nanomachines). Molecular communication also provides mechanisms for nanomachines to communicate, and thus, it may help drug repositories (sender nanomachines) to coordinate and control the amount and timing of drug release. Targeting delivery of drugs to specific cells in a human body and creating molecular communication for such applications involve understanding how molecules are transported within a biological system, how molecules are addressed to specific locations within a biological system (i.e., molecule-addressing mechanisms in a biological system), or even adding receptor molecules to a biological system to produce an addressing mechanism [5, 30, 38, 45].

Research on molecular communication has started when the initial idea of molecular communication was presented in 2005 [27]. Existing research on artificially creating molecular communication is based on understanding the design of molecular communication in biological systems and on modifying the functionality of existing molecular communication in biological systems (e.g., [18, 26, 37, 40]). Research on theoretical foundations of molecular communication is also underway. The importance of information theory for small scale communication systems is addressed and research efforts are currently being made for addressing information theory for various classes of molecular communication (e.g., [2, 15, 16, 31, 32, 36]).

This article is organized in the following manner. Section 2.2 gives an overview of various forms of molecular communication in biological systems. Section 2.3 describes a generic architecture of molecular communication where senders and receivers communicate using molecules through key communication processes. Section 2.4 describes existing research on engineering molecular communication components and systems, and Sect. 2.5 concludes this article.

Column 1: Molecular Communication

A New Information and Communication Paradigm

Molecular communication is a new information and communication paradigm based on mechanisms and materials from biological systems. In molecular communication, information is represented with biochemical molecules, called *information molecules*, such as protein and DNA molecules. Information molecules are transmitted by the sender of communication, propagated in the environment, and received by the recipient of communication, which in turn decodes the information (i.e., reacts to the information molecules). Molecular communication provides a biocompatible solution as it adopts mechanisms and materials from biological systems, and its applications to medical and environmental domains are highly anticipated.

Comparison with Telecommunication

Molecular communication displays distinctive features that are not found in the current method of telecommunication based on electrical or optical signals. In telecommunication, electromagnetic waves propagate in space, or electrons/optical signals propagate via communication cables/links, by which digital information (texts, audio, and movies) is transmitted at a high speed and over a long distance. In molecular communication, on the other hand, information molecules (chemical signals) propagate in an aqueous environment. The speed and range of propagation are extremely limited compared to those of telecommunication, yet molecular communication potentially transmits information that cannot be transmitted using the current telecommunication technology. For instance, a macromolecule such as a protein exhibits an extremely complex structure which represents some biological function (e.g., catalysis of some chemical reactions). Molecular

Table 2.1 Telecommunication and molecular communication

	Telecommunication	Molecular communication
Information carrier	Electromagnetic waves, electrical/optical signals	Chemical signals
Media	Space, cables	Aqueous
Speed	Speed of light (3×10^8 m/s)	Extremely slow (nm ∼ μm/s)
Range	Long distance (∼km)	Short distance (nm ∼ m)
Information	Texts, audio, videos	Chemical reactions, states
Other features	Reliable, high energy consumption	Unreliable, biocompatible energy efficient

communication therefore enables transmission of functional information, allowing the recipient of communication to undergo biochemical reactions.

Also, current telecommunication is designed to transmit information accurately and reliably by consuming electrical power. On the other hand, molecular communication becomes fundamentally stochastic and unreliable due to the noise effects in the molecular environment. However, molecular communication presents unique features such as biocompatibility and energy efficiency that may be explored to enable new Information Communication Technology applications. Table 2.1 summarizes the features of molecular communication and telecommunication.

Column 2: Potentials of Molecular Communication

A Communication Paradigm for Bio-nanomachines

In the current telecommunication paradigm, silicon based devices communicate via electrical or optical signals. The recent engineering technology makes it possible to miniaturize as well as mass produce such communication devices. In the area of sensor networks, tiny sensing devices, called motes, are distributed to form a communication network. In molecular communication, even smaller devices, called bio-nanomachines or simply nanomachines, communicate. Nanomachines are nano-to-micro meter scale devices capable of computing, sensing, actuating, and such nanomachines can be synthesized using naturally existing biological machines and mechanisms by using bio-engineering technology.

Applications of Molecular Communication

Molecular communication is expected to impact a variety of technological domains. First, molecular communication provides a direct method for interfacing with biological systems, and thus applications to health and environmental domains are anticipated. As an example, consider a body network system where implant bionanomachines monitor health conditions. The body sensor networks technology mainly embeds silicon devices in the body, while in the molecular communication paradigm, silicon devices may be replaced with biocompatible nanomachines, allowing them to directly communicate with cells, organs and tissues in the body. Also, continuing research on molecular communication may develop drug delivery systems where the delivery of molecules is controllable, environmentally friendly communication systems that may degrade and are integrated with the environment, and brain machine interfaces through which biochemical molecules directly interact with the brain and brain tissues.

In addition, molecular communication is expected to create a new unconventional computing paradigm. The area of molecular computing has demonstrated that basic computing and memory units can be made from biological materials and mechanisms. Molecular communication may provide a means to interconnect these basic units to form a large-scale massively parallel computing circuit. It may be possible in the future to have a computer that does not require electrical power and that does not produce heat. Further, unique features found in biological systems such as stochastic behavior, self-replication (cell division), self-organization (morphogenesis), and energy efficiency, may be explored to create a radically new computing paradigm.

2.2 Molecular Communication in Biological Systems

Molecule-based communication or molecular communication is a common and ubiquitous method by which biological nanomachines communicate. Various modes and mechanisms of molecular communication are found within and between cells. In the following, modes and mechanisms of molecular communication are categorized based on how signal molecules are propagated, namely, whether signal molecules simply diffuse in the environment or they directionally propagate by consuming chemical energy. The former type of communication is called passive transport-based molecular communication, and the latter type active transport-based molecular communication.

2.2.1 Passive Transport-Based Molecular Communication

Passive transport provides a simple method of propagating signal molecules within a cell and between cells. In passive transport, signal molecules randomly diffuse in all available directions, making it particularly suited to environments that are highly dynamic and unpredictable. Passive transport is also suited to situations in which infrastructure for communications is not feasible. Passive transport, however, requires a large number of signal molecules to reach a distant destination, and owing to the random movement of molecules, the time to reach a destination increases with the square of the distance. Passive transport is also not suitable to propagate large signal molecules in a crowded environment such as the interior of cells.

In the following, we describe three examples of passive transport-based molecular communication from biological systems, including (a) free diffusion-based molecular communication, (b) gap junction mediated diffusion-based molecular communication, and (c) reaction–diffusion-based molecular communication.

(a) *Free diffusion-based molecular communication*. In this mode of molecular communication, cells release signal molecules (e.g., proteins and peptides) into the extracellular environment, and neighboring cells capture the signal molecules with protein receptors, resulting in the activation of a chemical reaction (e.g., increased metabolism or transcription of cellular proteins). An example of free diffusion-based molecular communication is quorum sensing, a communication mechanism for bacterial cells. In quorum sensing, bacterial cells release an autoinducer, acyl homoserine lactone (AHL) into the environment, and detect the concentration of AHL in the environment, to estimate the number of nearby bacteria. When the AHL concentration is sufficiently high in the environment, the bacteria interpret this as a large number of bacteria in the environment, and thus, the bacteria start transcribing DNA to perform group functions (e.g., enough bacteria to generate an infection, form a biofilm, or generate luminescence) [14].

(b) *Gap junction mediated diffusion-based molecular communication*. Diffusion of signal molecules can be guided through cell–cell communication channels called gap junction channels [4]. Gap junction channels are physical channels formed between two adjacent cells, connecting the cytoplasm of the two cells. Gap-junction channels allow only connected cells to communicate, enabling coordinated actions among adjacent cells, such as synchronized heart-beating by cardiomyocytes.

(c) *Reaction-diffusion based molecular communication*. Diffusion of signal molecules can involve biochemical reactions to achieve a different mode of communication that allows propagation of impulses. As a result of the quick increase and decrease of signal molecules in their concentrations, the signal molecules appear as an impulse that propagates in the environment. For instance, some glial cells produce impulses of calcium ions (Ca^{2+}) for intercellular communication. The endoplasmic reticulum (ER) in a cell gathers and stores calcium ions, and when a cell is stimulated (e.g., by a physical stimulus), it releases the stored calcium ions from the ER and the calcium diffuses to adjacent cells through cell–cell junction channels. The diffused calcium in turn stimulates the adjacent cells, causing a chain reaction of calcium stimulation. Shortly after being stimulated and releasing calcium, a cell pumps calcium within the cell back into the ER and suppresses further stimulation, thus creating a short impulse of calcium through the cell. Because the communication propagates in a short impulse of calcium concentration, cells can communicate at a higher frequency. Neurons similarly produce ion impulses (called action potentials) that propagate over the length of the neuron.

2.2.2 Active Transport-Based Molecular Communication

Active transport provides a communication mechanism to directionally transport signal molecules to specific locations. Active transport can propagate signal molecules over longer distances (up to meters in length) as compared with diffusion-based passive transport. Large signal molecules and vesicles diffuse poorly in passive transport because of their size; on the other hand, active transport consumes chemical energy and generates sufficient force to directionally transport large signal molecules. Active transport provides a communication mechanism with a high degree of reliability even when the number of signal molecules to transport is small. Because the transport of signal molecules is directional, the probability of signal molecules reaching the destination is higher than when using passive transport, and thus active transport requires fewer signal molecules to perform communication. However, active transport often requires communication infrastructure to produce and maintain the transport, guide, and interface molecules (e.g., molecular motors, microtubules filaments, and vesicles). Active transport of molecules also requires a regular supply of energy to overcome the chemical interactions between signal molecules and molecules in the environment.

In the following, we describe two examples of active transport-based molecular communication from biological systems, including (a) molecular motor-based molecular communication and (b) bacterial motor-based molecular communication.

(a) *Molecular motor-based molecular communication.* This type of molecular communication is found within a cell where molecular motors are used to transport signal molecules. A molecular motor is a protein or a protein complex that converts chemical energy (e.g., ATP hydrolysis) into mechanical work at the molecular scale. Inside a cell, molecular motors transport signal molecules or large vesicles (e.g., liposomes, cell organelles) that contain signal molecules [44, 47, 51]. Molecular motors consume chemical energy (e.g., ATP) to transport signal molecules or the vesicles along the preestablished guide molecules (e.g., along a star-shaped topology that the guide molecules form inside a cell).

(b) *Bacterial motor-based molecular communication.* Bacteria move directionally based on chemical concentrations in the environment. Bacteria also exchange DNA through the process of conjugation. In this process, two types of bacteria, a sender bacterium with an F-plasmid (i.e., a genetic sequence that enhances the transfer of genetic information) and a receiver bacterium without F-plasmid, transfer a DNA chromosome through a pilus (i.e., a projection from the sender bacterium to the receiver bacterium forming a bridge for transmitting DNA). Transfer of DNA between bacteria may allow the receiver bacterium to acquire DNA that produces some useful cellular functionality (e.g., protein production, antibiotic resistance). Therefore, bacteria essentially transport DNA to other bacteria in the environment [9, 50]. The receiver bacterium may also release pheromones that form a chemical gradient that guides a sender

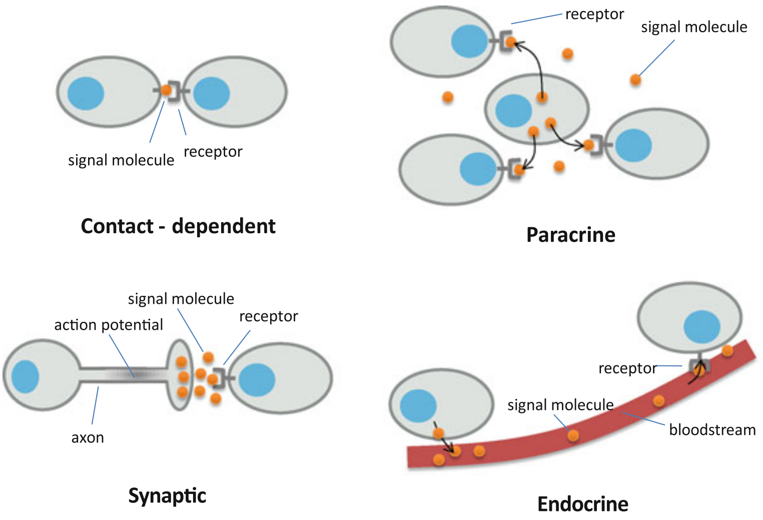


Fig. 2.1 Molecular communication in biological systems

bacterium toward a receiver bacterium (i.e., the closer to the receiver, the higher the chemical concentration).

Column 3: Communication in Our Body

Cell Communication

In molecular communication, communicating entities communicate information by sending, propagating and receiving information molecules. This is a common and ubiquitous method by which biological cells in our body communicate. Molecular communication provides a method for our cells to coordinate and maintain bodily functions and conditions necessary to sustain our life. In addition to those cells constituting a multi-cellular organism, unicellular organisms such as bacterial cells also communicate with each other by exchanging chemical signals in order to coordinate activities among a group of cells.

Examples of Cell Communication

In cell biology, cell communication mechanisms are often divided into four major types as shown in Fig. 2.1 in this column, out of which the first two types are for nearby cells to communicate (i.e., short distance communication) and the last two are for distant cells to communicate (i.e., longer distance communication.)

The first type is called contact-dependent by which signal molecules propagate from one cell to the other when the two cells physically contact. In this case, signal molecules expressed on the surface of one cell binds to the cell surface receptors on the other cell. The second type of cell communication is called paracrine, in which sender cells secrete signal molecules such as protein and peptide molecules to the extracellular space, which bind to the receptors of nearby receiver cells. The third type is synaptic, and this is how neurons propagate signals. In this type of communication, one end of a neuron, upon stimulation, generates an electrical gradient called the action potential. The action potential propagates over the axon toward the other end of the neuron, where this action potential produce chemical signals (neurotransmitters) that diffuse in the synapse to stimulate the next neuron (post-synaptic neuron). Synaptic communication provides a fast (e.g., 100 m/s) and long distance communication (a few meters). There exists another long distance communication mechanism, called endocrine—the fourth type. In this type, hormonal signals are transported or broadcast via a bloodstream to distant target cells or organs in our body.

Column 4: Intracellular Material Transport

Passive and Active Modes of Cell Communication

Modes (or mechanisms) of cell communication can be classified into passive or active modes based on how signal molecules propagate. The passive mode of cell communication relies on the random walk or free diffusion of signal molecules. Thus, it is difficult to propagate signal molecules over a long distance as well as to move large signal molecules in a high viscosity environment such as the cellular cytosol. The active mode provides a solution by consuming chemical energy, and one instance that enables the active mode of cell communication is a family of molecular motors, such as myosin and dynine (Fig. 2.2).

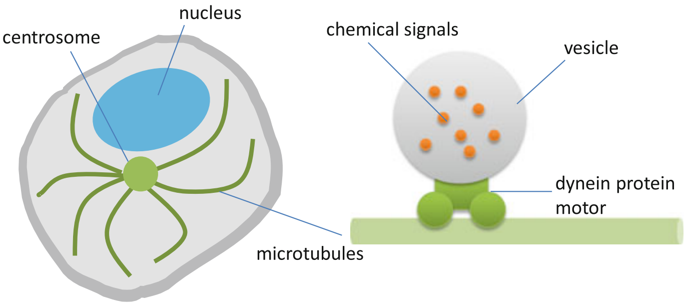


Fig. 2.2 Active mode of communication by molecular motors

Molecular Motors Transport Materials inside the Biological Cell

Molecular motors directionally walk along the protein rails by consuming chemical energy obtained from ATP (Adenosine Triphosphate) hydrolysis. Molecular motors are attached to a vesicle or cargo containing a set of signal molecules and actively move toward the destination. Molecular motors are in this sense vehicles running on the highway of protein rails within the cell. The active mode of cell communication propagates signal molecules directionally to a target location, providing a more reliable communication method than the passive mode. With the current bionanotechnology, motor proteins can be purified from the cell and the active transport mechanism can be reconstructed in vitro. Recent research successfully demonstrated the application of molecular motors to an in vitro molecule delivery system where molecules can be loaded to molecular motors, transported, and unloaded at the destination [1], which is directly applicable to the development and advancement of Lab-on-a-chip or μ-TAS (Micro-Total Analysis Systems).

1. S. Hiyama, R. Gojo, T. Shima, S. Takeuchi, K. Sutoh, “Biomolecular-Motor-Based Nano- or Microscale Particle Translocations on DNA Microarrays,” Nano Lett, **9**(6):2407–2413 (2009)

2.3 Molecular Communication Architecture

As described in Sect. 2.2, there exist varieties of modes and mechanisms of molecular communication in biological systems. One may ask whether it is possible to generalize necessary components and processes for molecular communication. Establishing a generalized architecture for molecular communication may help understand design principles of biological systems as well as help

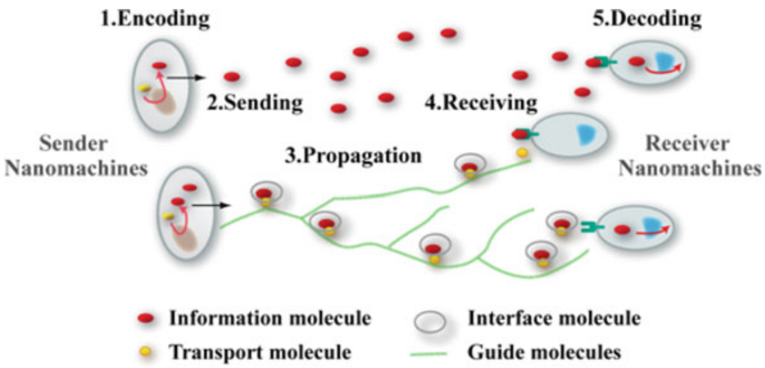


Fig. 2.3 Generic representation of molecular communication

engineer artificial biological systems. In this section, we attempt to describe an abstract architecture for molecular communication.

2.3.1 Generic Representation of Molecular Communication

Figure 2.3 depicts a generic molecular communication architecture. It consists of components functioning as information molecules (i.e., signal molecules such as proteins or ions) that represent the information to be transmitted [48], sender nanomachines that release the information molecules (e.g., cells, cellular organelles, protein machinery), receiver nanomachines that detect information molecules, and the environment in which the information molecules propagate from the sender nanomachine to the receiver nanomachine [49]. It may also include transport molecules (e.g., molecular motors) that transport information molecules, guide molecules (e.g., microtubule or actin filaments) that direct the movement of transport molecules, and interface molecules (e.g., vesicles) that allow any type of information molecule to be transported by a transport molecule [37].

Components of molecular communication are made up of molecules, and must be designed with the thought in mind that molecular communication occurs in an aqueous or air environment, which generates a significant amount of noise. The noise in the environment comes from a number of factors. For instance, the environment contains energy and forces that constantly interact with the various molecular communication components. Such energy and forces include thermal energy, electrical fields, magnetic fields, and electromagnetic waves (e.g., light energy that is absorbed by molecules). The environment also contains molecules and nanomachines that do not participate in molecular communication, and these generate noise. Such molecules and nanomachines include solvent molecules (e.g., water molecules), solute molecules (e.g., energy molecules necessary to support the nanomachines), and other nanomachines (e.g., nanomachines to maintain chemical concentrations in the environment or act as structural components). The noise in the environment determines the degree of randomness in molecular communication (e.g., randomness in the movement of information molecules and randomness in the timing of, rate of, and energy for chemical reactions).

2.3.2 Molecular Communication Processes

The components in molecular communication perform the following general phases of communication (Fig. 2.1): encoding of information into an information molecule by the sender nanomachine, sending of the information molecule into the environment, propagation of the information molecule through the environment, receiving of the information molecule by the receiver nanomachine, and decoding of the information molecule into a chemical reaction at the receiver nanomachine.

2.3.2.1 Encoding

Encoding is the phase during which a sender nanomachine translates information into information molecules that the receiver nanomachine can detect. Information may be encoded onto the type of information molecules used. Information may also be encoded in various characteristics of information molecules such as the three-dimensional structure of the information molecule, in the specific molecules that compose the information molecules, or in the concentration of information molecules (the number of information molecules per unit volume of solvent molecules).

The amount of information that a sender encodes onto a single information molecule is limited by the molecular structure of a receiver nanomachine. A receiver nanomachine is capable of only a limited number of configurations; thus, when a receiver nanomachine receives an information molecule, the receiver nanomachine receives only the amount of information corresponding to the number of configurations possible in the receiver nanomachine [46]. If a receiver nanomachine achieves only two configurations, the receiver nanomachine only understands one bit of information at a time.

A sender nanomachine may transmit information about which a receiver nanomachine does not have information. For instance, in biological systems, certain types of cells communicate through exchanging DNA molecules. An arbitrary DNA sequence is decoded at a receiving cell into proteins that introduce new functionality into a receiver cell. If a sender nanomachine in molecular communication can transmit arbitrary information molecules, it is possible to send new functionality to the receiver nanomachine.

2.3.2.2 Sending

Sending is the phase during which a sender nanomachine releases information molecules into the environment. A sender nanomachine may release information molecules by either unbinding the information molecules from the sender nanomachine (e.g., by budding vesicles from a biological cell if a sender nanomachine is a biological cell with endoplasmic reticulum), or by opening a gate that allows the information molecule to diffuse away (e.g., by opening a gap junction channel in the cell membrane of a sender nanomachine). A sender nanomachine may also catalyze a chemical reaction that produces transport molecules elsewhere.

Because of its size, a sender nanomachine contains a limited amount of energy and information molecules. This results in a limited communication capability that the single sender nanomachine alone can generate. Thus, the sender nanomachine of molecular communication generally relies on the environment for supplying (chemical) energy and information molecules. In addition, multiple sender nanomachines may release the same information molecules, resulting in a stronger signal in the environment.

2.3.2.3 Propagation

The sender and receiver nanomachines are at different locations in the environment, and propagation is the phase during which the information molecule moves from the sender nanomachine through the environment to the receiver nanomachine. The information molecule may diffuse passively through the environment without using chemical energy or may bind to a transport molecule (e.g., a molecular motor that generates motion [7]) and actively propagate through the environment using energy.

In passive transport, information molecules randomly move according to forces in the environment. Large information molecules or high-viscosity environments result in slower diffusion through the environment. In active transport, a transport molecule carries information molecules through the environment, and it consumes chemical energy to reduce the randomness in its movement in the environment. By providing more control over the movement of information molecules, active transport may decrease the time necessary for information molecules to reach the receiver nanomachine and also increase the probability of information molecules successfully reaching the receiver nanomachine.

During propagation, an interface molecule may also be necessary to protect information molecules from noise in the environment. For instance, information molecules may be contained in a vesicle-based interface molecule and propagate through the environment [37]. The vesicle prevents the information molecules from chemically reacting with other molecules outside the vesicle.

2.3.2.4 Receiving/Decoding

Receiving is the phase during which the receiver nanomachine captures information molecules being propagated in the environment. Decoding is the phase during which the receiver nanomachine, upon capturing information molecules, decodes the received molecules into a chemical reaction. One option for capturing information molecules is to use receptors that are capable of binding to a specific type of information molecule. Another option for capturing them is to use channels (e.g., gap-junction channels) that allow them to flow into a receiver nanomachine without using receptors. Chemical reactions for decoding at the receiver nanomachine may include producing new molecules, performing a simple task, or producing another signal (e.g., sending other information molecules).

2.3.3 Characteristics of Molecular Communication

A variety of communication characteristics result from using biological nanomachines as senders and receivers, using molecules as an information carrier, and from the propagation of information molecules through an aqueous environment.

First, the environment noise causes the propagation of information molecules to be fundamentally stochastic and to have a relatively large communication delay. Second, information is encoded onto molecules and the amount of information that a receiver nanomachine can receive is limited by the number of possible configurations in the receiver nanomachine. Third, the biological molecules and materials used to create components for molecular communication cause communication to exhibit characteristics of biocompatibility and energy efficiency. Each of the unique characteristics of molecular communication is described below.

2.3.3.1 Stochastic Communication

The stochastic behavior of molecular communication arises from environmental factors such as unpredictable movement of molecules in the environment, as well as from communication component factors such as nanomachines stochastically reacting to information molecules and nanomachines and information molecules degrading over time. Environmental factors and communication component factors inherently affect the design of molecular communication. For instance, in order to be robust to the environmental noise, the sender nanomachines increase the signal-to-noise ratio by releasing a large number of information molecules, and receiver nanomachines chemically react to information molecules only when a relatively large number of information molecules are present in the environment. Sending a large number of information molecules is highly redundant, so that degradation of a few information molecules during propagation does not impact communication. In addition, non–information molecules (noise-causing molecules) in the environment may not trigger chemical reaction at a receiver nanomachine as long as the noise from those molecules is significantly less than the signal from a large number of information molecules.

2.3.3.2 Large Communication Delay

Molecules in the environment that do not participate in molecular communication cause information molecules to propagate slowly through the environment. In both passive and active transport, the speed of propagation is bounded at a relatively lower speed (i.e., micrometers per second in an aqueous environment) because of the large amount of interference with the molecules in the environment. In addition, because of the possibly large delay in the environment, information molecules may remain in the environment for a period of time and arrive at a receiver nanomachine after a widely varying amount of time (e.g., hours after release at the sender nanomachine).

The delay in an information molecule propagating from a sender to a receiver nanomachine determines various communication characteristics of molecular communication. If the information molecules stay in the environment for an extended length of time (e.g., DNA molecules can be chemically stable for months), a sender nanomachine must delay a new communication until the information molecules used in an old communication degrade in the environment to prevent the old communication from interfering with the new communication. Thus, the frequency at which a sender nanomachine sends information molecules is determined by the length of time information molecules stay in the environment before degrading.

2.3.3.3 Molecule Based Coding

In molecular communication, information is encoded in various characteristics of information molecules such as the type of information molecules used, three-dimensional structure, chemical structure (e.g., protein), sequence information (e.g., DNA), or concentration (e.g., calcium concentration) of information molecules [48]. A physical substance (i.e., the information molecule) propagates from a sender nanomachine to a receiver nanomachine, and the receiver nanomachine chemically reacts to incoming information molecules. Because information is encoded onto molecules in molecular communication, the amount of information that a sender encodes onto a single information molecule (or the amount of information that a receiver nanomachine receives) is limited by the number of possible configurations in the receiver nanomachine.

2.3.3.4 Biocompatibility

Sender and receiver nanomachines in molecular communication communicate through sending, receiving, and decoding (chemically reacting to) information molecules. Since molecular communication uses the same communication mechanisms as biological systems, nanomachines in molecular communication can directly communicate with various natural components in a biological system using encoding and decoding methods available to the biological system. Biocompatibility of molecular communication may enable applications such as inserting nanomachines into a biological system for medical applications that require biologically friendly nanomachines.

2.3.3.5 Energy Efficiency

Through the fine-tuning of natural evolution, molecular communication in biological systems has achieved a high degree of energy efficiency. For example, under some conditions, myosin (a type of a molecular motor) converts chemical energy (i.e., ATP) to mechanical work with 90% energy efficiency [28, 55]. Chemical energy may be possibly supplied by the environment in which nanomachines operate. For example, molecular communication systems deployed in a human body may harvest energy (e.g., glucose) from the human body, requiring no external energy sources.

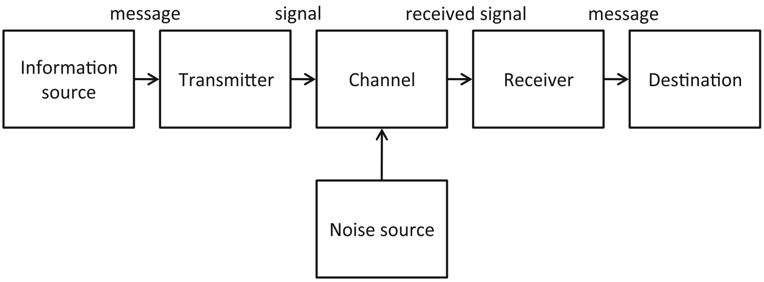


Fig. 2.4 Shannon's communication model

Column 5: Communication Model

Shannon's Communication Model

Claude Shannon, the founder of information theory, published "A Mathematical Theory of Communication" in 1948. As shown in Fig. 2.4 in this column, the model consists of information source, transmitter, channel containing the noise source, receiver and destination. This model assumes that the information source produces a message which is converted by the transmitter to the signal in the form that can propagate over the channel. During propagation, the signal may be altered or lost being affected by the noise. The signal is received by the receiver and a message is reconstructed from the received signal at the destination.

Molecular Communication Model

Molecular communication may be understood based on the Shannon's communication model described above where the goal is to deliver a message produced by the information source to the destination. In the first phase of communication, a message (or information) needs to be coded using signal molecules. One approach for coding is to represent information based on the number of molecules. For example, 10 molecules may be used to represent 10. Another approach is to represent information based on types of molecules. In this case, different molecules are used to convey different information, assuming that the receiver has receptors to receive the molecules. Some other approaches include modifying a molecular structure to represent information, or modulating a concentration change pattern similar to FM (Frequency Moderation) in radio communication.

In the second phase of communication, information carrying molecules propagate in the molecular communication environment, i.e., a communication channel. An approach to propagate molecules is passive propagation. In this case, signal molecules randomly walk in the environment to reach the receiver. Another approach is active propagation. In this case, chemical energy is consumed to propagate signal molecules directionally or against the concentration gradient. The noise in the molecular communication environment is thermal noise or other molecules that chemically react with the information molecules.

The last phase of communication is to reconstruct (or decode) information from received signals at the receiver. Decoding in molecular communication indicates that the receiver reacts to received molecules. Decoding may for instance lead to chemical reactions, conformational changes, and energy conversion at the destination. At the cellular scale, decoding may lead to growth, differentiation, migration or death of biological cells.

Column 6: How Many Bits of Information can be Transmitted via Molecular Communication?

Information Carrying Molecules

Molecular communication is very slow and only applicable to a small range of the environment (e.g., μm range), although molecular communication may be able to deliver a large amount of information. Consider using a protein molecule to encode information. Assume that the protein molecule can take two structurally and functionally different states: phosphorylated and unphosphorylated states, then this protein can store one bit of information. If the protein molecule can take multiple stable states, say N states, then the amount of information that can be stored is $\log_2 N$. Let's scale up to the size of a cell and consider the amount of information that can be stored therein. It is known that an *E. Coli* bacterium has chromosome containing a 4.6 million basepair in a 2 μm^2 cross-sectional area [1]. Since a DNA base is either one of the four: cytosine (C), guanine (G), adenine (A), thymine (T), the number of possible states N is $4^{4,600,000}$, and the information amount ($\log_2 N$) is 9.2 M bits. As a comparison, a memory density achieved with the silicon technology is expected to be 24.5 Gbits/cm^2 in 2014. This means that only 490 bits are stored in a 2 μm^2 area. This indicates that molecular communication may provide a method for transferring a large amount of information.

Functional Molecule

A unique feature of molecular communication is that information carries (i.e., information molecules) have a physical property (e.g., a chemical state or structure) that represents some function. An enzyme, for instance, is a protein family that acts as a catalyst of a specific chemical reaction. A set of the DNA molecules, genetic information of a bacterial cell from the previous example, carries functions that allow the cell to perform massively parallel information processing by detecting, processing and synthesizing a number of molecules in the environment. Coming back to the advancement of silicon technology, an expected transistor density around 2014 reaches 664 M/cm^2. Assume that a simple logic requires four transistors, then only three logic units can be realized in a 2 μm^2 area, whereas the bacterial cell performs complex information processing. Molecular communication transmits functional information carries—there is a potential as a breakthrough information technology.

1. M.L. Simpson, G.S. Saylerb, J.T. Flemingb, B. Applegate, "Wholecell biocomputing," Trends in Biotechnology, vol 19, issue 8, pp. 317–323 (2001)

2.4 Engineered Molecular Communication

Advanced bio-nanotechnology and improved understanding in cell and molecular biology have made it possible to design and engineer molecular communication systems. A common approach for designing and engineering molecular communication systems is to extend or modify existing biological systems. Accordingly, many molecular communication systems are engineered by exploiting biological systems. This section overviews the state-of-the-art research in engineering of molecular communication components and systems. A collection of molecular communication research including that in related areas is categorized below into (1) engineering of sender and receiver nanomachines, (2) engineering of transport mechanisms for propagating information molecules, and (3) engineering of communication mechanisms that are necessary to build larger-scale integrated molecular communication systems.

2.4.1 Engineering of Sender and Receiver Nanomachines

Sender and receiver nanomachines need communication functionality such as that to synthesize, store and release information molecules for sender nanomachines, and to capture and react to specific information molecules for receiver nanomachines. There are two basic classes of approaches for developing nanomachines: they are modifying existing biological cells, or producing simplified cell-like structures using biological materials that achieve communication functionality.

2.4.1.1 Synthetic Biological Cells

The area of synthetic biology has demonstrated that new communication capabilities can be added to biological cells through genetic engineering [6, 12, 53, 54]. In [6], sender nanomachines are engineered to synthesize and release information molecules (AHL molecules) using specific metabolic pathways. Receiver nanomachines are also engineered to respond to the information molecules by synthesizing specific reporter proteins in a concentration dependent manner (if the concentration of the information molecule falls in a certain concentration range.) The information molecules that sender nanomachines release are membrane permeable and freely diffuse from senders to receivers; therefore, they communicate through free-diffusion-based molecular communication.

Synthetic biology has also demonstrated that many other capabilities can be introduced into biological cells through genetic engineering. Demonstrated functionality in the area includes the following:

- *Logic functions*: In [54], computational building blocks were engineered based on DNA transcription and translation processes. An example of building blocks is a biochemical inverter, where the input mRNA generates a repressor protein that prevents DNA transcription processes, producing no output mRNA. In the absence of the input mRNA, DNA transcription processes proceed to generate an output mRNA. Other building blocks were designed. Similarly, in [13], signal transduction pathways in eukaryotic cells were modified by synthetic signaling proteins to demonstrate gating behaviors such as AND or OR gates.
- *Toggle switches*: In [21], a one bit memory was implemented with two genes inserted into a bacterium. Its bistability is achieved by the two genes under the control of promoters that are mutually repressed by the product of the other gene. Two inducers are also introduced to suppress the product of one of the two genes, which are used to switch from one state to the other.
- *Oscillators*: In [17], a bacterium has inserted DNA sequences that cause the bacterium to oscillate the concentration of three protein products (TetR, cI, LacI) over time. The DNA sequences to be inserted into the bacterium were selected, so that TetR blocked the promoter sequence of cI, cI blocked the promoter sequence of LacI, and LacI blocked the promoter sequence of TetR. Thus, at time t, TetR is active; cI is blocked; and LacI starts being produced. At time t_1, LacI becomes active; TetR is blocked; and cI starts being produced.

These functions can be used to increase the complexity of senders and receivers of molecular communication. For example, logic functions can be used at receiver nanomachines to produce programmed responses based on received information molecules; Toggle switches (i.e., 1 bit memories) can be used to retain a communication-related memory inside cells (e.g.,. a communication status of whether sending or waiting); oscillators (i.e., clocks) can be used at sender nanomachines to control the timing of release. However, it is noted that introducing multiple functions faces a technical difficulty of a possible interference with existing functions (e.g., at the protein level), and it is not technically easy.

2.4.1.2 Artificial Cells

The other approach to engineering sender and receiver nanomachines is to create simplified cell-like structures using biological materials (e.g., through embedding of proteins in a vesicle). Simplification of biological systems (e.g., cells) may lead to components that are easier to engineer.

One example of the approach of simplification is to start with a lipid bilayer and then add functionality as necessary [19, 29]. The lipid bilayer is similar to the membrane that encloses a cell. The lipid bilayer forms into a vesicle (a spherical lipid bilayer), and functional proteins (e.g., a receptor) are either embedded into the vesicle or are captured inside the vesicle. In [19], chemical components for a minimal cell that is capable of replication are proposed. In [29], an active receptor and enzyme are successfully embedded into a lipid bilayer, producing sender and receiver functionalities. By restricting the number of functionalities that are embedded in the vesicle, it is not necessary to consider the various complex processes (e.g., other metabolic pathways in the cell and processes to maintain the structure of the cell) occurring in a natural biological nanomachine. For instance, it is possible to select chemicals to cause the vesicles to form tubular projections to connect the vesicles without concern for the various other chemical processes [3, 43].

Column 7: Engineering Nanomachines I

Synthetic Biology: Programming Biological Cells

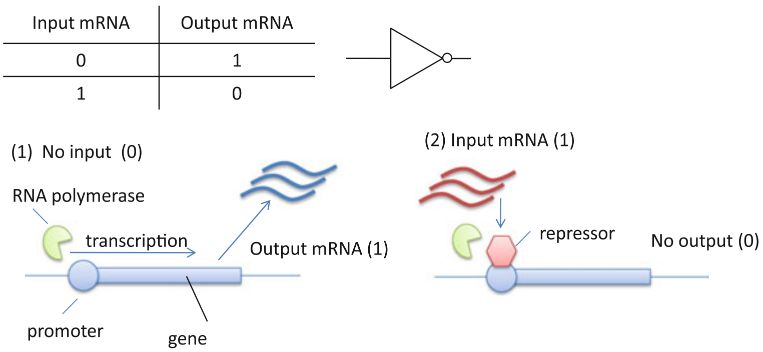
Biological nanomachines (nanomachines in short) are small scale molecular machines with computing, sensing, and actuating functions. An approach to synthesizing nanomachines is to program a naturally existing cell via genetic engineering to perform an intended function. In the area of Synthetic Biology, a biological cell was genetically modified to induce a programmed behavior that is not found in the native version of the cell.

Genetically Engineered Biological Cells

Representative work in Synthetic Biology includes engineering of an inverter, bi-stable switch (called the toggle switch) and oscillator, each implemented in a bacteria cell.

In the development of the inverter [1], a DNA sequence is inserted into a bacterium to output an mRNA sequence only in the absence of an input mRNA (i.e., a NOT gate) (Fig. 2.5). In the absence of an input mRNA, an RNA polymerase recognizes the promoter region of the DNA sequence to perform transcription, producing an output mRNA. When an input mRNA coding a repressor is present as an input, the expressed repressor binds to the promoter region, blocking the binding of the RNA polymerase to the promoter region, thereby no output mRNA is produced. In addition to the inverter, other basic units (NOR and NAND) as well as more complex circuits have been demonstrated.

In the development of the toggle switch, a DNA sequence inserted in a bacterium contains two genes A and B that repress each other's expression



Input mRNA	Output mRNA
0	1
1	0

Fig. 2.5 Inverter design

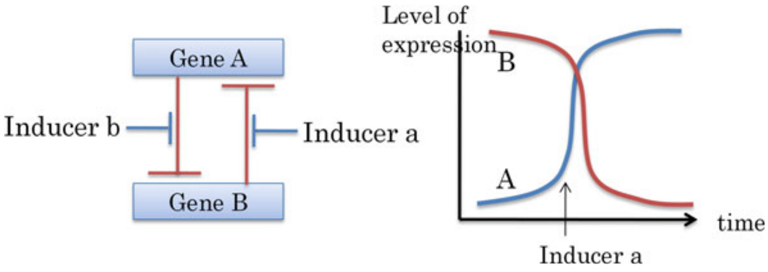


Fig. 2.6 Toggle switch design

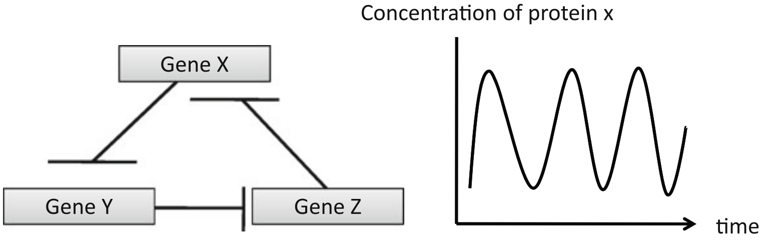


Fig. 2.7 Oscillator design

(Fig. 2.6, left). Since the expression of one gene represses that of the other, this system can take one of the two stable states (i.e., either of the protein product reaches a high concentration), representing one bit memory. The bit-state in this system can be externally altered by adding an inducer that forces the expression of gene A or B (Fig. 2.6, right).

In the development of the oscillator, a DNA sequence inserted in a bacterium causes three protein products to oscillate the concentrations over time. The DNA sequence here contains three genes X, Y, Z respectively coding proteins x, y, z that repress each other as shown in Fig. 2.7, left.

When X is expressed, protein product x represses the expression of Y, which then promotes expression of Z. As a result of Z expressed, X is then repressed. In this manner, the protein product x (and also other two) oscillates in the concentration as shown in Fig. 2.7, right.

The functional modules being developed in synthetic biology will be useful for developing a molecular communication system. The MIT (Massachusetts Institute of Technology)'s BioBricks project [2] aims at establishing a standard library for engineering and programming biological cells. It is expected in Synthetic Biology that one can program a biological cell in a manner we currently develop a software program using an API (application Programming Interface).

1. R. Weiss, S. Basu, S. Hooshangi, A. Kalmbach, D. Karig, R. Mehreja, I. Netravali, "Genetic circuit building blocks for cellular computation, communications, and signal processing," Nat. Comput. vol 2, pp. 47–84 (2003). The BioBricks Foundation, http://bbf.openwetware.org/

Column 8: Engineering Nanomachines II

Engineering a Cell from Scratch

Another approach to engineering nanomachines is to chemically synthesize a cell from scratch using biological materials. Briefly, a cell-like container called liposome can be made from a lipid bi-layer that assembles into a spherical form, and chemical compounds necessary for genetic transcription and translation (e.g., genetic materials and regulatory proteins) can be embedded. The cell-like container provides a local environment where specific chemical reactions can occur. The cell-like container separates the inside from the outer environment, preventing the inside compounds from reacting with other molecules in the environment. Although the lipid bi-layer is highly compatible with our body and its use for medical applications is highly anticipated, the structure is not so durable in our body. Polymersomes, emerging synthetic materials that exhibit increased stability, are thus considered an alternative to encapsulate chemical compounds to create an artificial cell [1].

Molecular Communication Using Artificial Cells

It is possible to implement a communication interface by adding a transport protein on the surface of an artificial cell (Fig. 2.8). A transport protein is a transmembrane protein that allows specific molecules to cross the membrane. When embedded on the surface of an artificial cell, it allows the artificial cell to uptake molecules in the environment and to release inside molecules to the outside. This is indeed nanomachine capable of receiving and sending molecules. Surprisingly, in the area of

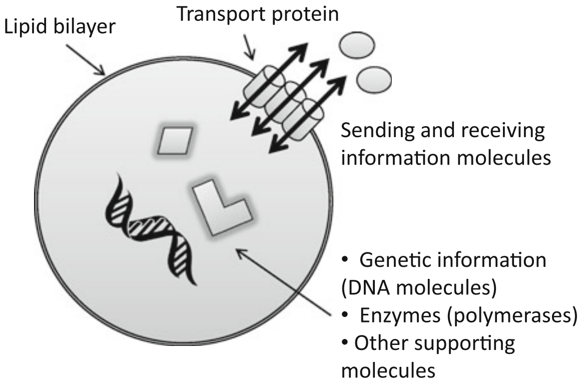

Lipid bilayer
Transport protein
Sending and receiving information molecules
• Genetic information (DNA molecules)
• Enzymes (polymerases)
• Other supporting molecules

Fig. 2.8 Design of an artificial cell

artificial cells, a self-replicating artificial cell is under consideration and development [2].

1. M. Doktycz, M. Simpson, Nano-enabled synthetic biology, Mol. Syst. Biol. 3:125 (2007)

2. A.C. Forster, G.M. Church, "Towards synthesis of a minimal cell," Mol. Syst. Biol. 2:45 (2006)

2.5 Engineering of Transport Mechanisms

Research in engineering molecular communication includes developing transport mechanisms for propagating information molecules. This section describes engineering of passive transport mechanisms and active transport mechanisms. As described in Sect. 2.2, passive transport propagates information molecules freely in the environment while active transport propagates information molecules directionally using motors.

2.5.1 Engineered Passive Transport Mechanisms

2.5.1.1 Free-diffusion Based Molecular Communication

One type of engineered passive transport is to use free diffusion based molecular communication, where communication is inherently a broadcast or random walk of information molecules. In free diffusion-based passive transport, a receiver may have receptors that are specific and that respond only when a certain type of information molecule is present in a high enough concentration [22, 53]. Thus, the sender may select the receiver nanomachine by sending the information molecules that cause a response at the receptors of the desired receiver nanomachine.

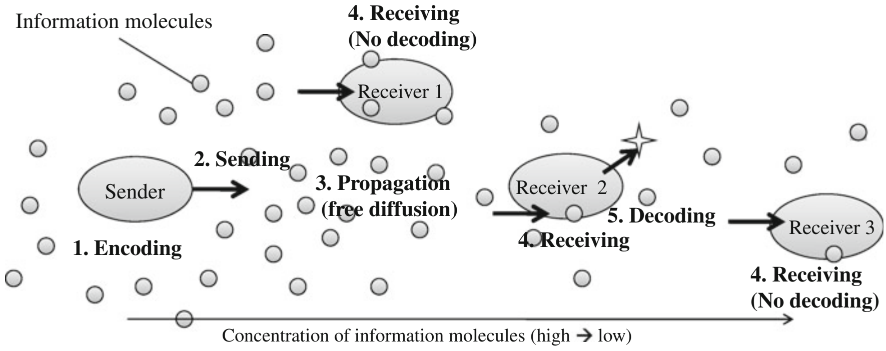


Fig. 2.9 An example communication sequence based on free-diffusion based molecular communication

The sender may also set the concentration of information molecules (by adjusting the number of information molecules to transmit) so that only nearby receivers with the specific receptor are capable of responding, and distant receivers do not respond. For instance, an engineered bacterium produces VAI (Vibrio fischeri autoinducer) information molecules, and the VAI information molecules diffuse through the environment and bind to VAI-specific receptors (i.e., a specific DNA sequence that binds VAI) in a receiver bacterium that produces a response (e.g., GFP expression).

An example communication sequence based on this type of molecular communication is illustrated in Fig. 2.9, where the sender of communication specifies receivers of communication in a concentration-dependent manner. The sender synthesizes and transmits information molecules in the environment (encoding and sending). The information molecules transmitted into the environment create a concentration gradient with its highest concentration value around the sender as shown in Fig. 2.9. The receivers implement a band detector [6]; i.e., they produce a reaction if its concentration falls in a specific concentration range. Therefore, those receivers that are within the concentration range (receiver 2) respond (receiving and decoding), but those that are outside the concentration range (receivers 1 and 3) do not produce any reactions (receiving only and no decoding).

2.5.1.2 Gap Junction Mediated Reaction–Diffusion Based Molecular Communication

Another type of engineered passive transport is to use reaction–diffusion based molecular communication, where information molecules react in the environment to increase and decrease their concentrations while diffusing. This form of transport can cause a wave-like impulse of information molecules that propagate through the environment.

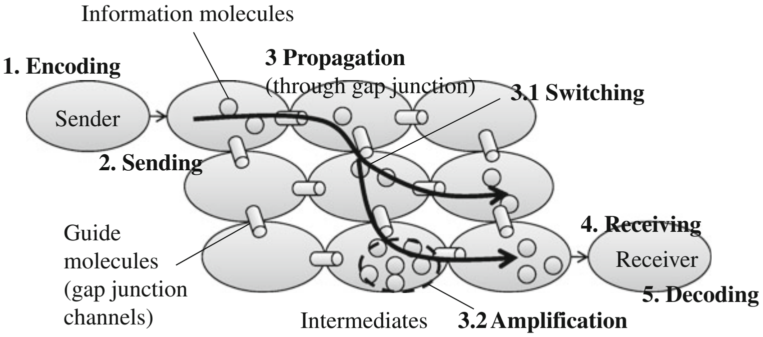


Fig. 2.10 An example communication sequence based on gap junction mediated reaction–diffusion based molecular communication

In [42], cell wires have been demonstrated to propagate calcium waves over a line of patterned cells. Diffusion of calcium waves are mediated by diffusion of calcium signals through gap junction channels, and therefore, this is gap junction mediated reaction–diffusion based molecular communication. Inside a cell, a calcium store (e.g., ER) has calcium release channels. The calcium release channels release calcium signals from the calcium store when calcium signals activate the calcium channels. Released calcium signals diffuse inside a cell and activate nearby calcium channels, which, in turn, release calcium signals from their calcium stores. The concentration of calcium inside a cell increases through this positive feedback. When the concentration of calcium becomes high, the cell uses various calcium pumps to remove calcium signals from the cell. This acts as a negative feedback. The positive and negative feedback produce a propagating impulse wave of calcium signals inside a cell. In addition, if multiple cells are coupled through cell–cell channels called gap-junction channels, Ca^{2+} increase of one cell can propagate to the other cell, allowing a longer distance molecular communication between nanomachines. Since gap junction channels can have different selectivity and permeability, this property can be used to implement filters and switching mechanisms to control the direction and range of propagation [41].

An example communication sequence based on this type of molecular communication is illustrated in Fig. 2.10. The sender synthesizes some stimulating molecules that can initiate cell–cell signaling at intermediate cells (encoding). The sender then emits the stimulating molecules (sending), which trigger cell–cell signaling at intermediate cells. Upon receiving the stimulating molecules, intermediate cells generate diffusive molecules and propagate them through gap junction channels intercellularly (propagation). During propagation, diffusive molecules may be directed toward target receivers (switching) or amplified for longer distance propagation (amplification). The receiver reacts to nearby intermediate cells that receive the diffusive molecules and initiates biochemical processes (receiving and decoding).

2.5.2 Engineered Active Transport Mechanisms

2.5.2.1 Molecular Motor-based Molecular Communication

One type of engineered active transport is to use molecular motors to transport information molecules or interface molecules containing information molecules. Biological systems include a variety of molecular motors (e.g., kinesin, myosin, cilia, flagella, etc.). Existing research on engineering active transport, however, focuses on only a few types of molecular motors (i.e., kinesin, dynein). Two examples of active transport using molecular motors introduced here are molecular motors walking on filaments and filaments propagating on a patterned surface coated with molecular motors.

In the first example, engineered active transport uses molecular motors, guide molecules (e.g., microtubule filaments) that self-organize into a network, and molecular motors (e.g., dynein, kinesin) that actively transport information molecules along the guide molecules [35, 49]. This first example uses components and processes similar to the active transport mechanisms in biological systems. In biological systems, a network of guide molecules (e.g., actin or microtubule filaments) is created within a cell in a self-organizing manner through dynamic instability of guide molecules, and molecular motors (such as dynein and kinesin) transport information molecules to specific locations within the cell by walking along a network of guide molecules. Engineering of the self-organization of the filaments can produce simple patterns of filaments (e.g., a star-shaped pattern or a random mesh pattern) [11, 18]. Through designing self-organization processes of creating filament patterns and through selectively transporting on the designed filament pattern, molecular motors may be guided to desired locations (e.g., desired receiver nanomachine or a desired set of receiver nanomachines).

In the second example of engineered active transport using molecular motors, the arrangement of microtubules and motors is inverted. A surface of a glass is coated with the molecular motor (e.g., kinesin), and the motors push the filaments along the surface. In this arrangement, transport molecules (i.e., the filament) load information molecules at a sender nanomachine and unload the information molecules at a receiver [24, 26]. The direction of filament movement is guided by adding walls (e.g., deposited proteins) onto the glass surface through lithographic techniques. Various patterns may be generated that gather filaments toward a receiver nanomachine. For instance, an arrow-shaped wall pattern acts a directional rectifier to ensure that filaments propagate in a single direction (e.g., only clockwise) by rectifying filaments that are propagating opposite the arrowhead direction [25].

An example communication sequence based on the first example is illustrated in Fig. 2.11. The sender encodes information using information molecules, and injects into interface molecules (e.g., vesicles). The sender then emits the interface molecules to molecular motors (e.g., through a budding mechanism). The interface molecules are then attached to and loaded on molecular motors. The interface molecules are propagated by molecular motors that move along

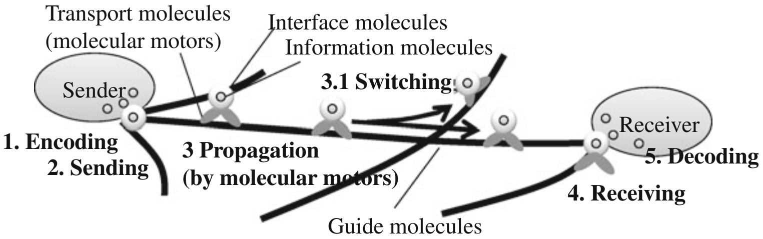


Fig. 2.11 An example communication sequence based on molecular motor-based molecular communication

guide molecules (e.g., rail molecules). During propagation, molecular motors may switch guide molecules to reach destination receivers. When molecular motors approach to the receiver, interface molecules containing information molecules are fused into the receiver. This allows receivers to receive information molecules and invoke reactions in response to information molecules.

2.5.2.2 Bacterial Motor-based Molecular Communication

In another type of engineered active transport, a transport molecule itself is a nanomachine using molecular motors to move itself toward a receiver nanomachine along chemical gradients in the environment. This example uses components and processes similar to self-propulsion of single-cell organisms (e.g., a bacterium) through an aqueous solution. In biological systems, a bacterium (i.e., a transport nanomachine) uses cilia or flagella (molecular motors that generate propeller-like forces) to move itself along chemical gradients toward favorable conditions (e.g., food source, light) and away from unfavorable conditions (e.g., harmful chemicals, light). With the engineered transport nanomachine with molecular motors, the receiver may generate guide molecules (similar to a gradient of a food source or pheromone in the environment in the bacterium example) to indicate where the receiver is and to influence the direction in which the transport nanomachine moves through the environment. The transport nanomachine does not require preestablished filament networks, since guide molecules emitted by a receiver diffuse to form the chemical gradient around the receiver. For instance, the pseudopodia of a cell (projections from a cell) can form complex shapes such as a structure that links the entrance and exit of a maze [39]. The pseudopodia form the structure by growing toward a chemical gradient of food originating from both the entrance and the exit of the maze. Active transport using pseudopodia is robust, since the pseudopodia adapt match to the shape of the maze. Another example of this type of molecular communication is found in bacterial cells, *E. faecalis* that perform conjugative plasmid transfer [52]. Female *E. faecalis* secretes pheromones to the environment, and male *E. faecalis* respond to the pheromones in the environment. Once male *E. faecalis* receive the pheromones, they activate gene

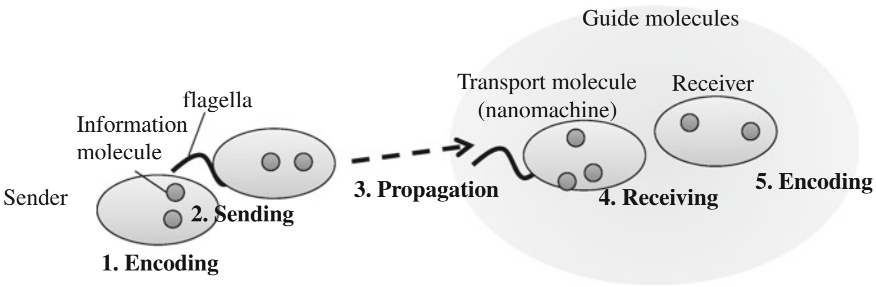


Fig. 2.12 An example communication sequence based on bacterial motor-based molecular communication

transcriptions to express cell surface adhesion proteins, allowing male and female *E. faecalis* to aggregate, so that the plasmid is transferred from male to female *E. faecalis*.

An example communication sequence based on this type of molecular communication is illustrated in Fig. 2.12. In this type of molecular communication, the sender and receivers contain information molecules therein (e.g., DNA or RNA molecules within cells), and the transport molecule (or the transport nanomachine) is used to transport information molecules between them. First, the sender transports information molecules to the transport nanomachine via bacterial conjugation. The receiver of communication emits guide molecules (e.g. pheromones) in the environment to attract the transport molecule (or the transport nanomachine). The transport molecule propels closer to the receiver, indicating the propagation process. Once the transport molecule and receivers are physically touched (aggregated), the information molecule is transported to the receiver which shows some response (receiving and decoding).

2.6 Engineering of Communication Mechanisms

Existing research in molecular communication focuses on building components for molecular communication as overviewed in this section. Such components, however, must work together to produce useful molecular communication. Existing research in creating integrated molecular communication systems includes developing a generic interface for interacting with a biological system and creating an addressing mechanism to enhance communications, each of which is briefly reviewed in this subsection.

Communication Interfaces

A communication interface provides a generic abstraction for communication that is reused by a variety of communication mechanisms. In molecular communication, a generic and abstract communication interface allows sender and receiver nanomachines to transport a variety of molecules using the same communication mechanism. Creating a generic and abstract communication interface is important in a molecular communication network, since, with such a communication interface, a communication component is designed without requiring details of other communication components [37].

One example of a generic and abstract communication interface is found in drug delivery [30]. In drug delivery, varieties of drugs are encapsulated in nanoscale capsules and target some location in the human body. Targeting molecules embedded in the nanoscale capsule bind to a specific receptor embedded in the target location inside the human body. The location of the receptor molecule in the human body thus determines the eventual location of the nanoscale capsules. With this interface, drug targeting mechanisms do not rely on characteristics of the drug (i.e., the information molecule) being transported. Nanoscale capsules act as a generic and abstract communication interface. The interface molecules allow any type of information molecule to concentrate at the correct location, which reduces the amount of drug necessary for results, and thus reduces side effects from drugs reaching undesired locations. The nanoscale capsule also protects information molecules from environmental noise and prevents the information molecules from chemically reacting with molecules outside the interface during the propagation.

Addressing Mechanisms

One important feature in communication is the ability to send information to a specific receiver. In molecular communication, an addressing mechanism allows a sender nanomachine to specify the receiver nanomachine. With a generic addressing mechanism, the sender nanomachine would have a generic abstraction for specifying receiver nanomachines. In diffusion-based communication in biological systems, a receiver nanomachine is addressed according to which information molecule is being used. On the other hand, a molecular communication system using a generic addressing mechanism is more flexible and imposes fewer constraints on which information molecules are used in molecular communication.

One approach for a generic addressing mechanism is using DNA sequences [26]. In the addressing mechanism investigated, the information molecule includes a single-stranded DNA with a sequence that specifies the address of a receiver nanomachine. The receiver nanomachine has an address DNA that is complementary to and thus binds to the single-stranded DNA on the information molecule. Using DNA sequences in addressing provides a method for generating a large number of addresses, since DNA sequences may be arbitrarily produced (with existing technology, DNA up to 10,000 base pairs in length) and binding of DNA sequence is understood (as evidenced by construction of various shapes from designed DNA sequences). Adding an addressing mechanism to molecular communication would allow the creation of more complex molecular communication networks.

Column 9: Pattern Formation

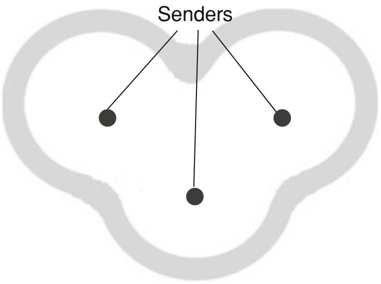
Morphogenesis and Molecular Communication

Molecular communication plays a key role in morphogenesis—the developmental processes of a biological organism. During the early stage of embryo development, a group of cells communicate through chemical signals referred to as morphogens. Morphogens are secreted by a cell to form a morphogen gradient in the environment. The concentration of the morphogen is the highest around the secreting cell and it becomes lower as the distance from the cell becomes longer. The cells in the environment respond to the morphogen concentration; they grow and differentiate to form a complex multi-cellular structure such as tissues and organs.

Artificial Morphogenesis

It is possible to control the biological structure to be formed by using information molecules that code morphological information. Such a study may be called artificial morphogenesis. As an example, consider a case where the three sender cells that secrete some signal molecules are placed as in Fig. 2.13. Assume that the receiver cells called band-detectors that detect and react to a specific concentration range of the signal molecules, are uniformly distributed in the environment. When the three sender cells synthesize and secrete about the same amount of signal molecules, the receiver cells on the edge of the heart-like shape (Fig. 2.13) will respond. If these receiver cells differentiate into different cell types, the heart-like structure emerges. Artificial morphogenesis is one of the promising application areas of molecular communication. It has a potential to advance and contribute to the development of artificial organs and regenerative medicine.

Fig. 2.13 An example of pattern formation



Column 10: A Molecular Communication System

Cellular Internet

We are currently developing a molecular communication system based on biological cells and cell–cell communication channels called gap junction channels—a micro-scale internet made of biological cells. We have applied bionanotechnology to create a cell wire, a line of connected cells that propagates chemical signals [1]. We have also successfully identified conditions under which chemical signals are amplified to propagate over a long distance [2]. Our ongoing efforts are toward developing a programmable molecular communication system that can control cellular behavior such as the growth, death, and differentiation of biological cells.

Dynamic Switching With Gap Junction Channels

Gap junction channels can have a different selectivity and permeability in response to various factors. Such dynamic properties can be exploited to implement networking functions such as switching. Figure 2.14a illustrates a possible design of switching cells. It is assumed in the figure that:

- Cell A expresses Cx_1 and Cx_2; Cell X expresses Cx_1; Cell Y expresses Cx_2, where Cx_n denotes a specific protein that constitutes gap junction channels.
- The same types of Cx can form a channel, but different types of Cx cannot.
- Signal molecules are more permeable to gap junction channels with Cx_1 than those with Cx_2.

If cells are structured as shown in Fig. 2.14a, gap junction channels are formed between Cell A and Cell X with Cx_1 and between Cell A and Cell Y with Cx_2;and signal molecules incoming to Cell A propagate preferentially to Cell X. The same cellular structure can then perform dynamic switching as shown in Fig. 2.14b. In this figure, an external signal is applied to decrease the permeability of gap junction channels made of Cx_1, allowing more signals to propagate toward gap junction channels made of Cx_2 (i.e.,

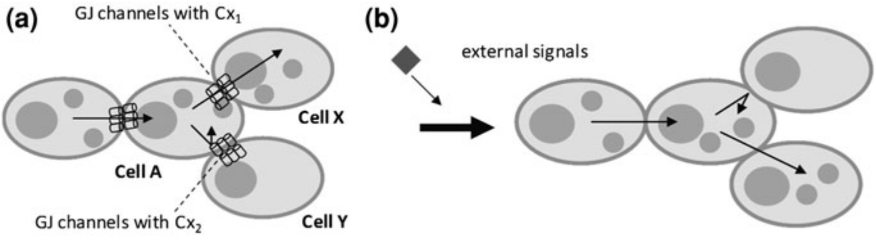


Fig. 2.14 Dynamic switching

toward Cell Y). Such switching may be possible with an external signal that can activate a specific kinase inside cells, which in turn phosphorylate a specific type of a gap junction constituting protein (i.e., Cx_1 in this case) that leads to the decreased permeability of the channels.

1. T. Nakano, T. Koujin, T. Suda, Y. Hiraoka, T. Haraguchi, “A Locally Induced Increase in Intracellular Ca^{2+} Propagates Cell-to-cell in the Presence of Plasma Membrane ATPase Inhibitors in Non-excitable Cells,” FEBS Lett. vol 583, issue 22, pp. 3593-3599 (2009)
2. T. Nakano, Y.H. Hsu, W.C. Tang, T. Suda, D. Lin, T. Koujin, T. Haraguchi, Y. Hiraoka, “Microplatform for Intercellular Communication,” in Proc. Third Annual IEEE International Conference on Nano/Micro Engineered and Molecular Systems, pp. 476–479 (2008)

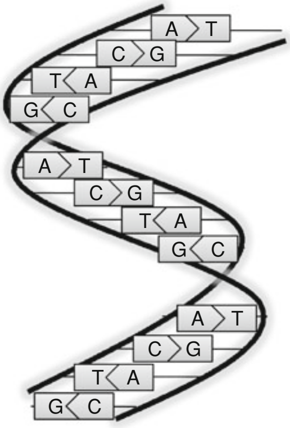
Column 11: DNA Molecules Walk

DNA (Deoxyribonucleic acid)

It is very well known that DNA carries and stores genetic information of biological systems. The double helix structure of DNA discovered by James D. Watson is also well known. Chemically, basic units of a DNA strand, namely DNA basis, are adenine (abbreviated A), guanine (G), cytosine (C), and thymine (T), where A and T, as well as G and C, are structurally complementary and can base pair via a hydrogen bond (Fig. 2.15).

DNA molecules have been considered new materials in the area of engineering. For example, in DNA computing and DNA nanotechnology areas, DNA molecules are used to solve computational problems or to create a nanoscale structure based on the self-assembling properties. In addition, recent research and development in the areas demonstrated that

Fig. 2.15 DNA double helix structure



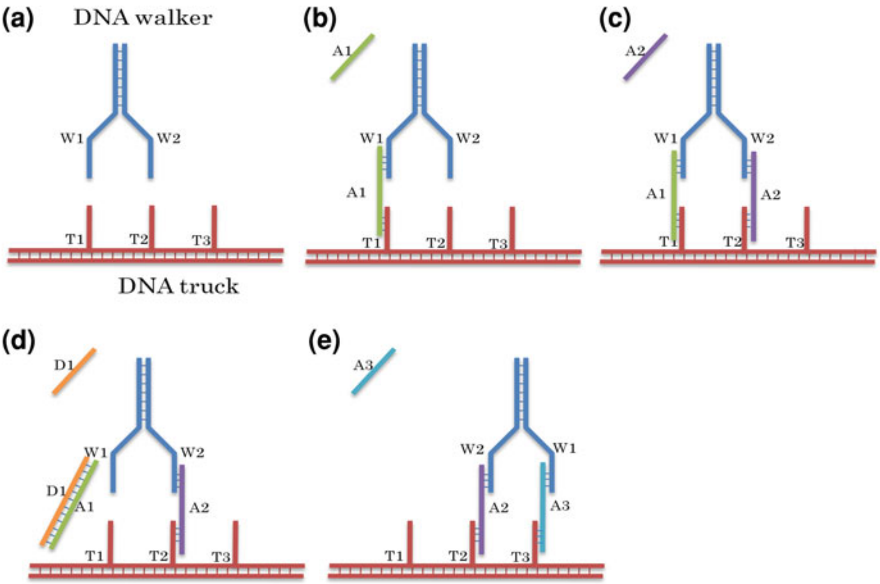


Fig. 2.16 DNA walker

specific structures of DNA molecules called DNA walkers or DNA spiders can "walk" in the nanoscale environment. Further, in this chapter, we discussed that an addressing mechanism can be implemented using DNA molecules.

DNA Walker

Figure 2.16 illustrates that a DNA walker progresses along the DNA track from the left to right. As shown in (a), the DNA walker has two legs, W1 and W2, and the DNA track has three branches, T1, T2, and T3. Each of the two legs and three branches is a single-stranded DNA sequence containing specific DNA basis.

First, a single strand DNA sequence A1, partially complementary to both W1 and T1, is introduced in the environment, then A1 forms a bridge to bind W1 to T1 (b). Second, a single strand DNA sequence A2, partially complementary to both W2 and T2, is introduced, then A2 forms a bridge to bind W2 to T2 (c). Third, a single strand DNA sequence D1, perfectly complementary to A1 is introduced, then D1 detaches W1 from T1, freeing leg W1 from the DNA track (d). Last, a single strand DNA sequence A3, partially complementary to W1 and T3 is introduced, then A3 forms a bridge to bind W1 to T3 (e). The DNA walker in this manner can walk in the nanoscale environment.

1. P. Yin, H. Yan, X.G. Daniell, A.J. Turberfield, J.H. Reif, A unidirectional DNA walker that moves autonomously along a track. Communications of Angewandte Chemie International Edition 43:4903–11 (2004)

Column 12: Recent Activities and Future Challenges

Recent Activities

The idea of using biological molecules in communication technology, namely molecular communication, has first appeared in 2005 and a line of research since then has started to address various issues on molecular communication. Molecular communication represents an interdisciplinary research area that spans over information technology, biology and nanotechnology, and as such information scientists, biologists and nanotechnologies commonly form a collaborative research group. Currently, research efforts are being made at National Institute of Information and Communications Technology (Japan), NTT DoCoMo (Japan), Tokyo University (Japan), Tokyo Medical and Dental University (Japan), Nara Institute of Science and Technology (Japan), University of California, Irvine (USA), Georgia Institute of Technology (USA), York University (Canada), Waterford Institute of Technology (Ireland), Middle East Institute of Technology (Turkey), and many others.

The first international symposium on molecular communication technology was held in Tokyo 2006 supported by National Institute of Information and Communications Technology, Japan. A workshop on molecular communication was held in Washington, DC 2008 financially supported by NSF (National Science Foundation, USA). In addition, technical sessions and panel discussions were held at various international conferences including IEEE INFOCOM 2005 (International Conference on Computer Communications), BIONETICS 2006 (Bio Inspired mOdels of NEtwork, Information and Computing Systems), and IEEE CCW 2007 (Computer Communications Workshop). Currently, BIONETICS [1] and Nano-Net [2] are the two major conferences that gather molecular communication researchers.

Future Challenges

Experimental efforts are so far focused on addressing typical physical layer issues of computer communications in addition to addressing issues related to designing and engineering system components. Theoretical efforts are also underway to investigate information representation using molecules and analyze the information transfer capacity in nano-to-micro scale environments. To further advance the area, the research on molecular communication will face the following challenges.

The first challenge is to identify, design, and engineer (or fabricate) building blocks necessary to provide a full molecular communication functionality. It is necessary to identify applicable biological materials and mechanisms, design system components using biological materials and mechanisms, apply bio-nanotechnology or biologically inspired methods (e.g., self-organization) to fabricate system components, and quantify the capacity and characteristics (e.g., durability in a biological environment).

As the second challenge, independent components engineered need to be integrated into a working communication system that robustly and stably functions. General research challenges include investigating a robust design architecture that allows communication components to self-organize and function under the influence of noise, and developing protocols, mechanisms and architectures that allow interfacing and networking among components.

The last challenge is to develop innovative applications of molecular communication systems. As mentioned, molecular communication systems are highly anticipated in medical domains. For instance, nanomachines capable of molecular communication may be implanted in a human body to interact with cells, tissues and organs for the purpose of human health monitoring. There are many other domains that molecular communication or networking of nanomachines may impact. To facilitate the design and development of applications, it is necessary to develop standard libraries, tools and frameworks.

1. BIONETICS (International Conference on Bio-Inspired Models of Network, Information, and Computing Systems) http://www.bionetics.org/

2. Nano-net (International ICST Conference on Nano-Networks) http://www.nanonets.org/

2.7 Summary

Molecular communication integrates techniques from biology to interact with biological systems, from nanotechnology to enable nanoscale and microscale interactions, and from computer science to integrate into larger-scale information and communication processing systems. Although research in molecular communication is in its infancy and has only reproduced functionality already available in biological systems, continuing research in molecular communication will lead to integrated molecular communication systems in which various components of molecular communication work together to provide full communication functionality. Molecular communication has significant potential, since it interacts directly with biological systems at nanoscales and microscales, and it may potentially impacts various technological domains including health (e.g., nanomedicine [20] and tissue engineering [8, 23]), environment (e.g., environmental monitoring), IT (e.g., unconventional computing [1] and body sensor networks [55]), and military (e.g., biochemical sensors).

References

1. A. Adamatzky, B.D.L. Costello, T. Asai, *Reaction-diffusion computers*, Elsevier (2005)
2. B. Atakan, O.B. Akan, “An information theoretical approach for molecular communication,” in *Proceedings of 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, Dec. 2007
3. K. Akiyoshi, A. Itaya, S.M. Nomura, N. Ono, K. Yoshikawa (2003) Induction of neuron-like tubes and liposome networks by cooperative effect of gangliosides and phospholipids. Fed. Eur. Biochem. Soc. Lett. **534**(1–3), 33–38
4. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular biology of the cell.* (Garland Science, New York, 2002)
5. T.M. Allen, P.R. Cullis, Drug delivery systems: entering the mainstream. Science **303**, 1818–1822 (2004)
6. S. Basu, Y. Gerchman, C. Collins, F. Arnold, R. Weiss, A synthetic multicellular system for programmed pattern formation, Nature. April 21 2005, vol 434, pp. 1130–1134
7. J.T. Barron, L. Gu, J.E. Parrillo, Malate-aspartate shuttle, cytoplasmic NADH redox potential, and energetics in vascular smooth muscle. J. Mol. Cell Cardiol. **30**, 1571–1579 (1998)
8. K.J.L Burg, T. Boland, Minimally invasive tissue engineering composites and cell printing. IEEE Eng. Med. Biol. Mag. (2003)
9. J.M. Berg, J.L. Tymoczko, L. Stryer, Biochemisty. 5th edn. (Freeman, New York, 2002)
10. D. Bray, Protein molecules as computational elements in living cells. Nature vol. 376, 27 July 1995
11. A. Chakravarty, L. Howard, D.A. Compton, A mechanistic model for the organization of microtubule asters by motor and non-motor proteins in a mammalian mitotic extract. Mol. Biol. Cell **15**, 2116–2132 (2004)
12. T. Dennis, J. Lee, T. Ozdere, T.J. Lee, L. You, Engineering gene circuits: foundations and applications, in *Nanotechnology in Biology and Medicine Methods, Devices and Applications*, Chapter 21, ed. by T. Vo-Dinh (CRC Press, USA, 2007)
13. J.E. Dueber, E.A. Mirsky, W.A. Lim, Engineering synthetic signaling proteins with ultrasensitive input/output control, Nat. Biotechnol. **25**, 660–662 (2007).
14. T.R. de Kievit, B.H. Iglewski, Bacterial quorum sensing in pathogenic relationships. Infect. Immun. **68**(9), 4839–4849 (2000)
15. A. Eckford, Nanoscale communication with Brownian motion, in *Proceedings of 41st Annual Conference on Information Sciences and Systems* (2007)
16. A. Eckford, Achievable information rates for molecular communication with distinct molecules, in *Proceedings of Workshop on Computing and Communications from Biological Systems: Theory and Applications* (2007)
17. M.B. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators. Nature Jan 20, **403**(6767), 335–338 (2000)
18. A. Enomoto, M. Moore, T. Nakano, R. Egashira, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara, K. Oiwa, A molecular communication system using a network of cytoskeletal filaments. in *Technical Proceedings of the 2006 NSTI Nanotechnology Conference and Trade Show***1**, 725–728 (2006)
19. A.C. Forster, G.M. Church, Towards synthesis of a minimal cell. Mol. Syst. Biol. (2006)
20. R.A. Freitas Jr., Nanomedicine, vol. I. *Basic Capabilities* (Landes Bioscience, USA, 1999)
21. T.S. Gardner, C.R. Cantor, J.J. Collins, Construction of a genetic toggle switch in *Escherichia coli*. Nature Jan 20, **403**(6767), 339–342 (2000)
22. Y. Gerchman, R. Weiss, Teaching bacteria a new language.in *Proceedings of the National Academy of Sciences***101**(8), 2221–2222 (2004)
23. L.G. Griffith, G. Naughton, Tissue engineering—current challenges and expanding opportunities. Science. **295**, 1009–1014 (2002)

24. H. Hess, C.M. Matzke, R.K. Doot, J. Clemmens, G.D. Bachand, B.C. Bunker, V. Vogel, Molecular shuttles operating undercover: a new photolithographic approach for the fabrication of structured surfaces supporting directed motility. Nano Lett **3**(12), 1651–1655 (2003)
25. Y. Hiratsuka, T. Tada, K. Oiwa, T. Kanayama, T.Q.P. Uyeda, Controlling the direction of kinesin-driven microtubule movements along microlithographic tracks. Biophys. J. **81**, 1555–1561 (2001)
26. S. Hiyama, Y. Isogawa, T. Suda, Y. Moritani, K. Suto, A design of an autonomous molecule loading/transporting/unloading system using DNA hybridization and biomolecular linear motors in molecular communication (European Nano Systems, Grenoble, France, 2005)
27. S. Hiyama, Y. Moritani, T. Suda, R. Egashira, A. Enomoto, M. Moore, T. Nakano, Molecular Communication, in *Proceedings of the 2005 NSTI Nanotechnology Conference, poster presentation*, USA, May 2005
28. J. Howard, *Mechanics of motor proteins and the cytoskeleton* (Sinauer, Sunderland, 2001)
29. J. Kikuchi, A. Ikeda, M. Hashizume, *Biomimetic Materials: Encyclopedia of Biomaterials and Biomedical Engineering* (Marcel Dekker, New York, 2004)
30. R. Langer, Perspectives: drug delivery—drugs on target. Science **293**, 58–59 (2001)
31. J.Q. Liu, H. Sawai, A new channel coding algorithm based on photo-proteins and GTPases, in *1st International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, Dec. 2006
32. J.Q. Liu, T. Nakano, An information theoretic model of molecular communication based on cellular signalng, in *Proceedings of Workshop on Computing and Communications from Biological Systems: Theory and Applications* (2007)
33. C. Mavroidis, A. Dubey, M.L. Yarmush, Molecular machines. Annu. Rev. Biomed. Eng. **6**, 363–395 (2004)
34. C.D. Montemagno, Nanomachines: a roadmap for realizing the vision. Biomed J Nanopart Res **3**(1), 1–3 (2001)
35. M. Moore, A. Enomoto, T. Nakano, R. Egashira, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara, K. Oiwa, A design of a molecular communication system for nanomachines using molecular motors. in *Proceedings of the Fourth Annual IEEE Conference on Pervasive Computing and Communications Workshops.* (IEEE Computer Society, Washington, DC, 2006)
36. M. Moore, A. Enomoto, T. Nakano, A. Kayasuga, H. Kojima, H. Sakakibara, K. Oiwa, T. Suda, Molecular-motor based communication on a microtubule topology, *2nd International Workshop on Natural Computing* (2007)
37. Y. Moritani, S. Hiyama, T. Suda, Molecular communication among nanomachines using vesicles. in *NSTI Nanotechnology Conference and Trade Show* (NSTI, Cambridge, 2006)
38. Y. Moritani, S. Hiyama, T. Suda, Molecular communication for health care applications. in *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops* (IEEE Computer Society, Washington, DC, 2006)
39. T. Nakagaki, H. Yamada, Á. Tóth, Maze-solving by an amoeboid organism. Nature **407**, 470 (2000)
40. T. Nakano, T. Suda, M. Moore, R. Egashira, A. Enomoto, K. Arima, Molecular communication for nanomachines using intercellular calcium signaling, in *Proceedings of the 5th IEEE Conference on Nanotechnology*, Nagoya, Japan, July 11–15 (2005)
41. T. Nakano, T. Suda, T. Koujin, T. Haraguchi, Y. Hiraoka, Molecular communication through gap junction channels: system design, experiments and modeling, in *Proceedings of the 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems* (*BIONETICS 2007*), Dec. 2007
42. T. Nakano, Y.H. Hsu, W.C. Tang, T. Suda, D. Lin, T. Koujin, T. Haraguchi, Y. Hiraoka, Microplatform for intercellular communication, in *Proceedings of the Third Annual IEEE International Conference on Nano/Micro Engineered and Molecular Systems* (2008)

43. S. Nomura, Y. Mizutani, K. Kurita, A. Watanabe, K. Akiyoshi, Changes in the morphology of cell-size liposomes in the presence of cholesterol: formation of neuron-like tubes and liposome networks. Biochim. Biophys. Acta **1669**(2), 164–169 (2005)
44. K. Oiwa, H. Sakakibara, Recent progress in dynein structure and mechanism. Curr. Opin. Cell Biol. **17**, 98–103 (2005)
45. N.A. Peppas, Y. Huang, Nanoscale technology of mucoadhesive interactions. Adv. Drug Deliv. Rev. **56**, 1675–1687 (2004)
46. T.D. Schneider, Theory of molecular machines I. Channel capacity of molecular machines. J. Theor. Biol. **148**, 83–123 (1991)
47. T. Shima, T. Kon, K. Imamula, R. Ohkura, K. Sutoh, Two modes of microtubule sliding driven by cytoplasmic dynein. Proc. Nat. Acad. Sci. **103**(47), 17736–17740 (2006)
48. J.M. Smith, The concept of information in biology. Philos. Sci. **67**(2), 177–194 (2000)
49. T. Suda, M. Moore, T. Nakano, R. Egashira, A. Enomoto, Exploratory research on molecular communication between nanomachines. in *2005 Genetic and Evolutionary Computation Conference, Late-breaking Papers* (ACM press, New York, 2005)
50. R.H. Tamarin, *Principles of genetics* (WCB/McGraw-Hill, New York, 1999)
51. S. Toba, K. Oiwa, Swing or embrace? New aspects of motility inspired by dynein structure in situ. Bioforum Eur. **10**, 14–16 (2006)
52. K. Wakabayashi, M. Yamamura, A realization of information gate by using enterococcus faecalis pheromone system, DNA7, LNCS 2340, pp. 269–278 (2002)
53. R. Weiss, T.F. Knight, Engineered communications for microbial robotics. DNA computing. in *6th International Meeting on DNA Based Computers, DNA,2000* (Springer Lecture Notes in Computer Science, 2054, New York, 2000)
54. R. Weiss, S. Basu, S. Hooshangi, A. Kalmbach, D. Karig, R. Mehreja, and I. Netravali, Genetic circuit building blocks for cellular computation, communications, and signal processing, Nat. Comput. vol. 2, pp. 47–84 (2003)
55. G.-Z. Yang, (ed.), *Body sensor networks*, Springer (2006)

Chapter 3
Artificial Chemistry and Molecular Network

Hideaki Suzuki

Abstract This chapter focuses on artificial chemistry, a research approach for constructing life-like systems in artificial environments, and presents its fundamental concepts and system design requirements. Based on this discussion, we move on to evaluate typical artificial chemistry systems: We propose 13 conditions necessary for emergent evolution and three topological conditions that must be met by the rules for transport of symbols. We also introduce the concept of a molecular network, which emulates the spatial relationship of the molecules. With hard-sphere random-walk simulations, we show seven topological conditions that the molecular network must satisfy. In the latter half of this chapter, we feature the example of network artificial chemistry (NAC), in which a molecular network is used for spatial representation, and present some of the research results derived since this was first proposed. We begin by introducing a model wherein a cluster formed by folding a node chain functions as an active machine. We then overview studies on rewiring rules for weak edges, which form the basis for network dynamics. We discuss the merits and demerits of the method expressing spatial constraints derived from the network energy, then introduce models devised to circumvent the difficulties: a model that implements active functions (programs) in the nodes and an improved model that implements the programs in agents to allow them to move within the network. The improved model, called "program-flow computing," is expected to be refined into a new computing model.

Keywords Artificial chemistry · Chemical reaction · Molecular interaction · Self-organization

H. Suzuki (✉)
National Institute of Information and Communications Technology, Kobe, Japan
e-mail: hsuzuki@nict.go.jp

H. Sawai (ed.), *Biological Functions for Information and Communication Technologies,* Studies in Computational Intelligence, 320,
DOI: 10.1007/978-3-642-15102-6_3, © Springer-Verlag Berlin Heidelberg 2011

3.1 Introduction

Each of the cells that make up our body is a gigantic machine in which several hundreds of millions of molecules (excluding water) interact in exquisite fashion in diverse biological processes, including genetic and metabolic activities. These machines are highly autonomous, adaptable, robust, reliable, and self-repairing, traits artificial machines have yet to approach. In recent years, researchers across a wide range of fields have pursued studies combining biology, chemistry, physics, and informatics to explore the design principles and mechanisms of the cell. Ambitious attempts in this area based on informatics include systems biology, which constructs detailed models of an entire living cell and performs simulations consuming vast computational resources [6, 27, 28, 36]. Nevertheless, even if we could acquire infinite computational resources and copy all the events and processes that occur in a living cell to computer code, then run simulations, would we really understand the "design principle" of a cell? Would we understand the diverse useful properties of the cells given above?

At the Bio-algorithms Project (http://www-karc.nict.go.jp/BA/index_J.html) of the Kobe Advanced ICT Research Center, the National Institute of Information and Communications Technology (NICT), our approach to research emphasizes a "bottom-up approach" (a constructive approach) (Fig. 3.1). This approach diverges significantly from the techniques used to create the detailed model described above. The goal is to build a computational and communication device with the properties of a living creature. We investigate the details of cell activity at the molecular level and build models of information processing and learning. The

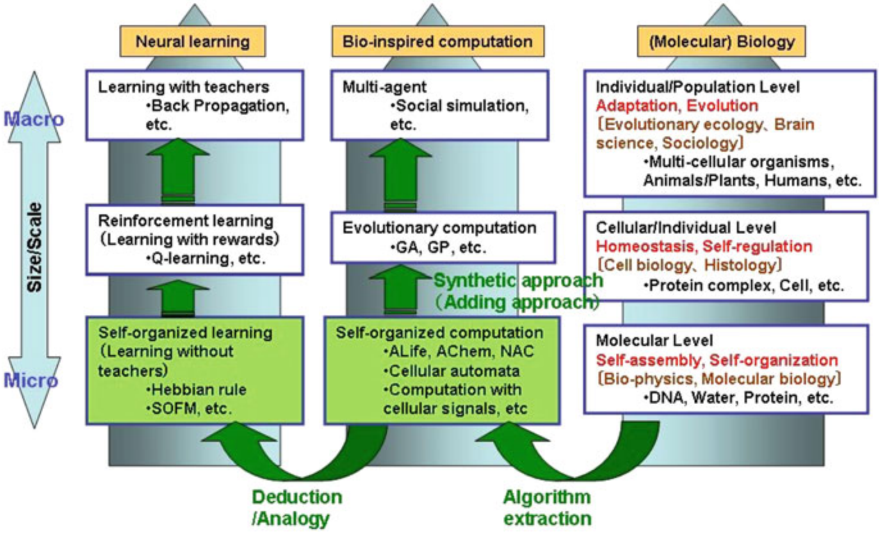


Fig. 3.1 Conceptual diagram of the bottom-up approach

resulting model may be less fine-grained than a comprehensive model that describes every physical and chemical property of the molecules in question (at exorbitant computational cost), and yet is required to be detailed enough to reflect molecular logic and to exhibit desirable properties of bio-molecules such as self-assembly and self-organization. By constructing such models and assessing their suitability, we seek to better understand the essence of biological properties at the molecular level and perhaps to propose new (non-von Neumann) information processing and communication models.

In order to present one example of this constructive approach, this chapter focuses on a research field known as artificial chemistry and provides a brief overview. After describing the basic concepts and requirements that must be met by artificial chemistry designs, we focus on NAC, which describes intermolecular interactions with mathematical graphs, and introduce the history of research in this area. In Sect. 3.7, we discuss future prospects for this area of study.

3.2 Artificial Chemistry

3.2.1 Basic Elements of Design in Artificial Chemistry

Artificial life is among several historical efforts to capture the essence of life via a constructive approach [3]. In brief, artificial life seeks to introduce the principles of biological evolution into computing or other artificial systems and to perform simulations that shed light on the nature of life using a constructive approach, observing life-like behaviors that emerge in the simulations and applying them to design engineering systems. Artificial chemistry has recently emerged as a subfield whose goal is to model the conditions under which bio-chemical reactions and chemical evolution occurred on the primeval Earth, eventually leading to emergent phenomena such as cell evolution and genetic encoding [9, 11, 14, 74]. In the design of self-organizing computational systems, the ideal model would be a chemical reaction system, confirmed to undergo self-organization or self-assembly and to produce complex structures and functions, composed of elemental processes whose mechanisms can be identified down to the level of quantum mechanics. Such a system could be used to design or evaluate models more accurate and reliable than the simplified systems (i.e., conventional artificial life) available now, which rely so heavily on human intuition.

In 2003, Suzuki et al. considered the environment required for such a system of artificial chemistry, distilling the essential characteristics to the five factors listed in Fig. 3.2 [61].

Here, of course, the elementary symbols are compared to bio-molecules in the narrow sense. In more generic artificial chemistry systems, the symbols might be compared to particles in the broad sense, including bases comprising DNA and RNA and other atomic clusters. The rules for reaction and transport can also be broadly

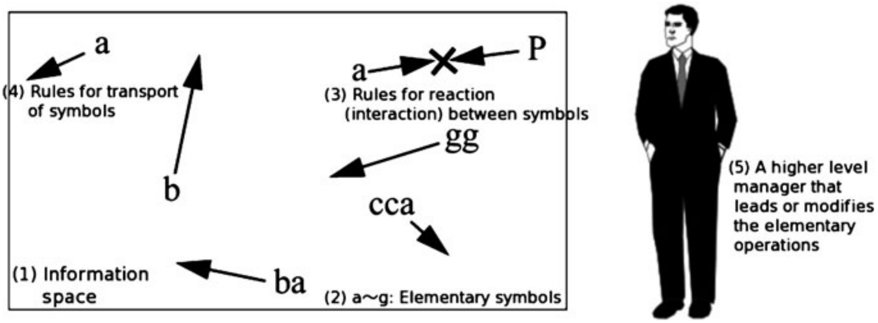

a
(4) Rules for transport of symbols
b
a
P
(3) Rules for reaction (interaction) between symbols
gg
cca
(1) Information space
ba
(2) a~g: Elementary symbols
(5) A higher level manager that leads or modifies the elementary operations

Fig. 3.2 Elements comprising the artificial chemistry framework. Elementary symbols such as *a* and *b* move and react with each other in a "rectangular" information space according to the rules

defined. For example, cellular automata [29, 30, 52, 53], a representative model for artificial life, may also be regarded as an artificial chemistry system in a broader sense. In this case, elementary symbols are the internal states of the automata, the information space is the cell space, and the rules governing transport and reactions are defined by the transition rules for the automata. In another example, Tierra [50, 51], a well-known core memory-based artificial life system, elementary symbols are machine instructions, the information space is the core memory space, rules governing transport and reactions are the computational operations defined for the machine instructions, and the higher-level manager is represented by external programs that cause mutations and kill individual programs. We also note the need to minimize or eliminate any interventions of a higher-level manager in artificial life/artificial chemistry models in which we take a bottom-up approach, allowing elements to react with each other as freely as possible.

3.2.2 Requirements for Artificial Chemistry System's Design: From the Perspective of Evolution and Emergence

Artificial chemistry is a research approach that emulates chemical evolution on the primeval Earth and generates emergent phenomena. The scenario for chemical evolution [33] currently posited in the life sciences states that primitive life emerged through several major events that happened in the ancient sea: the creation of polymers such as amino acids and proteins, the establishment of metabolic reactions, membrane formation, the creation of genetic materials such as RNA or DNA, and the establishment of a scheme for the translation of genetic information. It exceeds the author's capabilities and the scope of this chapter to reproduce or verify this scenario. Nevertheless, to design a better artificial chemistry system, we must set the five above-mentioned elements appropriately while considering chemical evolution.

In order to meet these requirements, in this section, we enumerate the qualitative conditions needed to generate the major events in the chemical evolution from the informatics perspective. Since the actual scenario of chemical evolution on Earth has yet to be revealed, the conditions specified in this section cannot be sufficient conditions (sufficient in the sense that emergent evolution will occur each time the conditions are met). Still, they likely constitute, at minimum, necessary conditions, or conditions without which emergent evolution cannot occur. We use the conditions obtained here to evaluate later various artificial chemistry system designs (Sect. 3.4).

Of the 13 conditions for emergence described below, Conditions (1)–(4) involve symbols and their reaction rules. Conditions (5) and (6) involve spatial structures. Conditions (7)–(11) involve rules of transport for symbols and compartments. Of these, Conditions (7)–(9) involve passive transport and conversion, which increase entropy, while Conditions (10) and (11) involve active transport, which decreases entropy. Finally, Conditions (12) and (13) are related to the gigantic symbol complex.

Emergence Condition (1): The total amount of symbols must be conserved as a resource.

Resources for the evolution of life fall into three general categories: energy, space, and matter. Of these, energy is a consumable resource, while space and matter are recyclable. A near-inexhaustible stream of energy is provided by the sun from outside the ecosystem; the ecosystem is a mere consumer of this resource. On the other hand, the total amount of space and matter are limited, and living creatures recycle both constantly. In particular, the total amount of matter (atoms) is all but perfectly conserved in chemical reactions, the elemental processes of biological activity, which makes it possible to recycle matter. What would happen if matter was not conserved and could be created or annihilated in the elemental processes of chemical reactions?

In order to answer this question, here we consider Tierra, the system of artificial life described earlier. In Tierra, machine instructions constitute the matter (symbol) resource. Tierra can allocate instructions freely to other locations in memory (space) by copying or by other operations. As a result, when Tierra operates, the memory quickly fills with instructions (matter), and the system is no longer able to function. In order to prevent this, Tierra applies a rule that results in the death of individuals—in other words, regular memory deallocation—through higher-level routines. Ideally, as this example indicates, the total amount of resources should be conserved for the evolution system to continue operating.

Emergence Condition (2): The system must be able to store and replicate an unlimited amount of genetic information.

In 1997, Szathmáry and Maynard-Smith [80] classified the information appearing, transmitted, and replicated in the initial stages of chemical evolution into three categories: Fixed Information, Compositional Information, and Programed Information. Fixed Information refers to information like molecular orientation, stored and replicated in processes like crystal growth. Compositional

Information refers to compositional distributions (which define cell types) of molecules in cells replicated in primitive cell division. Programed Information refers to the digital information encoded in the base sequences of the so-called informational macromolecules, including RNA and DNA.

Of course, the process of evolution generates a complex of individuals and diverse populations from mutations generated as characteristics are passed from parents to offspring. Evolution cannot occur if the information inherited from the parent to the offspring in this genetic mechanism cannot change or is significantly constrained. A system of artificial evolution also requires the presence and replication of unlimited information in the same manner as Programed Information in biological evolution.

Emergence Condition (3): An activator symbol (group) must exist that allows modification of symbols with specificity.

In addition to conditions governing genetic information, a cell, as a phenotype, requires another important function: a catalytic function with specificity. In a living cell, tens of thousands of different types of chemical reactions progress simultaneously in a single solution, and these reactions must occur in a time-specific manner at the required locations or the cell cannot function as an overall entity. Living creatures achieve this through enzymes (proteins), which are catalysts with specificity. We can also say that the set of these enzymes specifies the cell phenotype. An artificial evolution system must also provide such functions through its symbols or symbol complexes.

Emergence Condition (4): A translator from the genotype to the phenotype must exist, and the translation relationship must be expressed as a phenotype.

When we consider the above two conditions—the unlimited amount of genetic information and the activator symbol—from the evolutionary perspective, it becomes clear that we need some mechanism that relates the information to the activator. Living creatures cannot evolve without this mechanism, which maps mutational changes in genetic information to modifications in biological functions.

This relationship (normally called the translation relationship) plays an important role not just in living creatures, but in self-reproducing automata [84], explored by John von Neumann in his later years. Column 1 below describes von Neumann's self-reproducing automata in detail. One conclusion derived from this self-reproducing automata is this: If the translation relationship is not a fixed relationship prepared externally but established by the automata themselves as phenotypes, the translation relationship itself changes and is optimized during the process of evolution, resulting in an evolutionary system capable of even greater adaptability. In living creatures, this translation relationship is achieved by a set of molecules called transfer ribonucleic acid (aminoacyl tRNA) and by ribosomes. This protein family is yet another phenotype created from DNA and optimized through the process of evolution [32]. As in this biological translation system, an artificial system must have a mechanism for translation from the genotype to the phenotype, the mechanism itself being implemented as a phenotype.

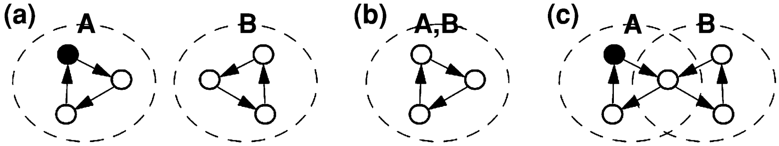


Fig. 3.3 Relationship between replicator species groups. **a** Groups A and B share no molecular species. **b** Groups A and B consist of identical molecular species. **c** Groups A and B share some, but not all molecular species. The *open* and *filled circles* are molecular species. The *arrows* indicate relationships whereby the molecular species at the tail of the arrow promotes the replication of the molecular species at the head of the arrow. The *filled circles* in **a** and **c** indicate molecular species with slightly increased rates of replication

Emergence Condition (5): Walls must be able to form and maintain compartment structures.

Up to the present, we have used the term *cell* without a clear definition. Here, we clarify the meaning of a cell or compartment. Compartmentalization and its grouping of the symbols have roughly two meanings.

First, consider a group of replicators floating in the sea—a solvent. Generally speaking, molecular replication is achieved through the collaborative action of multiple molecular species, not just one. Figure 3.3 shows the relationships between two molecular species groups, A and B. Assume here that Group A is slightly more efficient than Group B under a compartmentalized or non-compartmentalized environment. If Groups A and B have no relationship to each other or are identical, as in Fig. 3.3a, b, the presence or the absence of compartments has no effect on the subsequent relationship between Groups A and B. In Case (a), Group A will increase reproduction efficiency independent of Group B, with or without compartments. In Case (b), Groups A and B generate no differences. The problem arises in Case (c), when Groups A and B share some but not all molecular species. In this case, if the molecular species indicated by the filled circles improves the efficiency of Group A slightly, the molecular species common to Groups A and B are replicated more efficiently. Consequently, the replication efficiency of Group B increases slightly, whereas the increase in the replication efficiency of Group A declines slightly. On the other hand, if Groups A and B are enclosed in separate compartments, Group A can increase its efficiency without interference from Group B; as a result the efficiency in the compartment containing Group A increases dramatically compared to Case (a). As this example shows, compartmentalization can lead to large differences between molecular species groups and promote selection among them, which is a significant advantage in the unfolding of chemical evolution.

Another advantage of compartmentalization is observed in the situation in which the proliferation of replicators or cells has progressed to such an extent that the space is occupied by replicators or cells and resource molecules have begun to run short. In order for chemical evolution to continue from this point, the groups of molecular species must acquire the ability to catabolize (decompose) other molecular species into smaller molecules. Under these conditions, the presence of compartments affects the outcome of the evolution of molecular species; if the system lacks compartments and all molecular species groups coexist in the same solution field, a particular replicator species group decomposes the molecules produced by its own proliferation (its children, so to speak). Thus, the resulting total proliferation rate does not readily increase. If the replicator species groups are enclosed in compartments, on the other hand, decomposition (predation) may begin after the molecules formed by their proliferation are completely separated by compartment division. This separation of the target of proliferation and decomposition by compartmentalization can increase proliferation rates and confer another advantage in chemical evolution. Emergence Condition (7), discussed later, guarantees to a certain extent that the compartment targeted for predation will differ from the compartment divided.

Emergence Condition (6): It must be possible for symbol operations—or at least the replication of the compartment—to change the effective size of a compartment.

This condition is based on the observation that the effective size of a complex cell (compartment)—defined as the number of molecules (symbols) contained—is larger than a simple cell. Since the process of evolutionary emergence involves the creation of larger, more complex cells through several major events, the size of a compartment in the information space prepared in a system of artificial chemistry must also be structurally modifiable. This susceptibility to modification should be achieved at either of the two levels described below.

The first short-term susceptibility to modification is enabled by the condition that the number of symbols in a compartment can change without generational change. Various artificial chemistry spaces previously proposed can be classified as rigid (solid) or elastic (liquid) spaces. Another symbol can be freely inserted into two arbitrary symbols in an elastic space, but not in a rigid space. An example of an elastic space is a lattice space [58, 59], in which, the lattice points are not fixed and another lattice point can be inserted between two lattice points. Another is a lattice space in which the lattice points are fixed, but the number of symbols belonging to each lattice point can be changed freely [38–40]. For either approach, an elastic space can achieve short-term susceptibility to modification.

The second longer-term susceptibility to modification is enabled by the possibility of change in the effective size of a compartment when the compartment is replicated from parent to offspring. For example, in the case of Tierra, although the core memory space is rigid, the size of a compartment (creature) can change due to transcription errors during replication from the parent to the children, making the compartment size susceptible to modification.

Emergence Condition (7): The compartments must gradually mingle with each other over time due to the effects of randomization.

As is well-known, we eukaryotes have cells (compartment structures) with highly structured internal forms comprising organelles such as the nucleus and mitochondria and numerous liposomes (bags made of lipid bilayer membranes) involved in endocytosis (absorption), exocytosis (egestion), and the transport of molecules floating in the cell. Cases requiring more speed rely on transport along microtubules (rails) by motor proteins, but most cases depend on the slower mode of diffusion. In primitive cells lacking active transport systems, such passive transport is responsible for transporting all substances. For an artificial chemistry system too, therefore, a similar environment allowing the passive transport of compartments must be present.

The mingling (passive transport) of the compartments is also crucial for the emergence of patterns of interaction among individuals, such as predation, parasitism, and symbiosis, which are basic ecosystem activities. When cells or individuals occupy the space after repeated division and proliferation, offspring tend to occupy the space near the parents. When we observe spatial distributions after proliferation, individuals closer in pedigree are positioned nearby, while those distant in pedigree are positioned at a distance. If this tendency continues for an extended period, predation (to take one example) becomes highly disadvantageous to gene survival, targeting as it does the individuals (close relatives) located nearby. In addition, parasitism and symbiosis would not readily occur, since these relationships involve different species. In reality, in marine- or land-based environments, diffusion and randomization caused by wave or wind action would result in a mingled distribution of individuals.

Emergence Condition (8): Signal transmission by symbol transport must be possible.

Another outcome of passive transport (mingling) is signal transmission via the passive transport of symbols. We know that the various schemes of intercellular and intracellular signal transmission in living creatures are based on molecular diffusion, including for example, the well-known gene regulatory network formed by molecular interactions between DNA and switching proteins such as enhancers and repressors. Despite the difficulties associated with the straightforward implementation of symbol diffusion, particularly in spaces such as a core, an artificial chemistry system must provide some means of passive transport of symbols to allow emergent interactions between symbols. Chapter 2 of this book considers this condition in greater detail from a slightly different perspective, from which signal transmission by molecular transport is called molecular communication.

Emergence Condition (9): Some randomizing operation must be able to change the symbols or allow new combinations of symbols to form.

The last condition required as an outcome of passive transport is the creation of new functions by randomization. Consider the metabolic reactions established in a living cell. The chemical reaction velocity theory (see Column 2) always specifies the rate of molecular participation in a reaction at any given point in time. This means that the remaining molecules remain as a reaction reservoir from which molecules can readily be recruited if another reaction becomes possible. Thus, in a bio-molecular system, randomization and super redundancy serve as the driving forces for generating new reactions and functions.

Condition (9) requires equivalent properties of an artificial chemistry system. In most cases, a function in artificial chemistry is expressed as a symbol or a combination (arrangement) of symbols. If the arrangement (order) of symbols is not randomized by diffusion or mutation and is fixed permanently, new functions cannot be tested in the system. An artificial chemistry system must therefore have a mechanism that modifies the symbols and combinations of symbols, through randomization.

Emergence Condition (10): It must be possible to transport a symbol (or a symbol set) selectively according to the rules of transport or through the actions of activator symbol(s).

A biological system is nothing more or less than a physical and chemical system, the dynamics of which are specified by the direction in which the free energy, $F = E - TS$, decreases (where E is energy, T is absolute temperature, and S is entropy). According to this equation, if T is sufficiently high, the system shifts only in directions that increase S. However, if T is low, the system may reduce S to reduce E to achieve a total reduction in F. In this manner, at a certain temperature, the system can form orders (decrease S) according to the free energy minimization criteria. This is the driving force of creation of orders in living creatures. From cell formation and division to cell digestion and egestion, most primitive functions and orders can be emulated by assuming appropriate energy models for the molecules and applying the minimization principle. Examples include the chemotons proposed by Gánti [16] and simulations of the formation and division of artificial compartments by Ono et al. [38, 40] The transport of molecules based on energy minimization as such can be regarded as a means of active transport that makes indirect use of energy.

On the other hand, the cells of most present-day higher organisms have evolved an active transport system that uses energy more directly through the molecules known as ATP. Recently, Hirokawa et al. [20] have shown that a living cell has a network of microtubules, a system of rails, along which motor proteins carry loads (molecules in vesicles, in most cases) to destinations at various speeds. This transport system allows the cell to transport specific molecules to specific destinations in a timely manner. Lacking this mechanism, cells within higher organisms could not maintain or divide themselves.

Based on the above facts, an artificial chemistry system must feature a direct or indirect active transport method. Here, active transport refers to the operation of moving a symbol (or a symbol set) from one point in space to another. Out of the active transport, indirect transport refers to transport to the predefined destination (based on rules for transport of the symbols prepared in advance), whereas direct transport refers to transport to a specific destination (performed by a specific activator symbol). In general, in direct active transport the distance between the symbol (or a symbol set) transported and the destination is observed to decrease after transport.

Emergence Condition (11): Selective transport across walls of symbols must be possible.

As discussed earlier, membranes partition cell interiors into layered compartments to form organelles (including the nucleus, the Golgi apparatus, and mitochondria) and the liposomes used to transport molecules. Unlike the walls that (once formed) separate the interior from the exterior, the walls of these compartments are not completely sealed. Rather, these walls feature a mechanism involving numerous membrane proteins on the membrane surface that sense molecules inside or outside the compartments and selectively move molecules across the membrane in one direction or the other. A compartment lacking such a mechanism is essentially an inanimate object and cannot contribute to the functions of a living cell. Similarly, an artificial chemistry system must provide a mechanism allowing the selective transport of symbols across walls to allow the exchange of information in and out of compartments.

Emergence Condition (12): Multiple activator symbols must be able to operate simultaneously on a single operand symbol.

The genetic information programed without limits as required in Condition (2) is generally encoded as a one-dimensional symbol sequence on a long, thin symbol complex. With an increase in the amount of encoded information, this sequence grows to extraordinary length, which makes it takes time for an activator molecule to traces out the long sequence.

These operations, of course, are implemented in biological systems by informational macromolecules such as DNA and RNA or proteins such as polymerase. In living creatures, we need to pay attention to the fact that more than one protein acts on a single DNA molecule at any given time, which prevents degradation of replication or transfer rates as sequences grow longer. The same mechanism also enables mutual switching of genetic transcription (for example, a repressor protein suppresses the transcription of genes by attaching to a DNA molecule and preventing the polymerase from passing).

Such cooperative or competitive interactions between activator molecules are essential in generating intricate functions based on a single piece of genetic information. Likewise, an artificial chemistry system must allow multiple activator symbols to act simultaneously on a given operand.

Emergence Condition (13): Symbol complexes must be able to move across space, combine, and form new symbol complexes.

An active function in a biological system is not always achieved by a single activator molecule (protein); a multimeric complex is a molecule containing two or more polypeptides (chains of amino acids constituting proteins) in a single unit. In an extreme case, complexes like ribosomes (molecules involved in protein synthesis) contain several dozen amino acid chains to generate complex functions.

These molecular complexes form through a process called self-assembly. The constituent molecules move through the solvent, encounter each other, and combine (coalescing rather than reacting) with matching physical and chemical properties. Large protein complexes—ribosomes, for example—form from self-assembly occurring at multiple, hierarchical stages.

What properties are required for a solution molecular system to generate such hierarchical self-assembly? Coalescence, the elementary process of self-assembly, can be described quantitatively in the same way as a chemical reaction. We can apply chemical reaction velocity theory (refer to Column 2, discussed further below) to estimate the rate of coalescence. Theory states that rates of chemical reaction in a solution remain constant regardless of molecule size. Even as molecules grow in size, they take part in reactions with no loss of activity, a property that ensures self-assembly will occur at hierarchical stages.

The discussion so far points to the condition that must be met when we design an artificial environment to allow functions to emerge hierarchically through self-assembly. In general, in an artificial environment, the designed symbol complex becomes heavier as it grows, and concomitantly, tends to lose mobility and efficiency—something we need to prevent. In other words, large symbol complexes must be able to move around and combine with each other with no loss of mobility or efficiency.

Column 1: Von Neumann's Self-reproducing Automata

John von Neumann's self-reproducing automata consist of three parts: Tape T (*) (the parentheses enclose the content of the tape) for storing genetic information, a Machine C for making copies of the tape, and a Machine S for building machines based on the information on the tape. Machine C is generally called the copier and Machine S is generally called the constructor. Machine S is also called the Universal Constructor, meaning it can construct any machine if the appropriate tape information is supplied. The functions of these parts are expressed by the equations below.

$$\forall X: \quad C + T(X) \xrightarrow{C} C + T(X) + T(X)$$

$$\forall X: \quad S + T(X) \xrightarrow{S} S + T(X) + X$$

Here, X is the information for encoding Machine X.

The self-replication cycle of von Neumann's self-reproducing automaton, "C + S + T(C + S)," is expressed by the equation below.

$$\begin{aligned} C + S + T(C + S) &\xrightarrow{C} C + S + T(C + S) + T(C + S) \\ &\xrightarrow{S} C + S + T(C + S) + C + S + T(C + S) \end{aligned}$$

Figure 3.4 in this column is a schematic diagram illustrating this mechanism. (a) Copier C makes a copy of Tape T and (b) constructor S constructs C and S, based on the tape information.

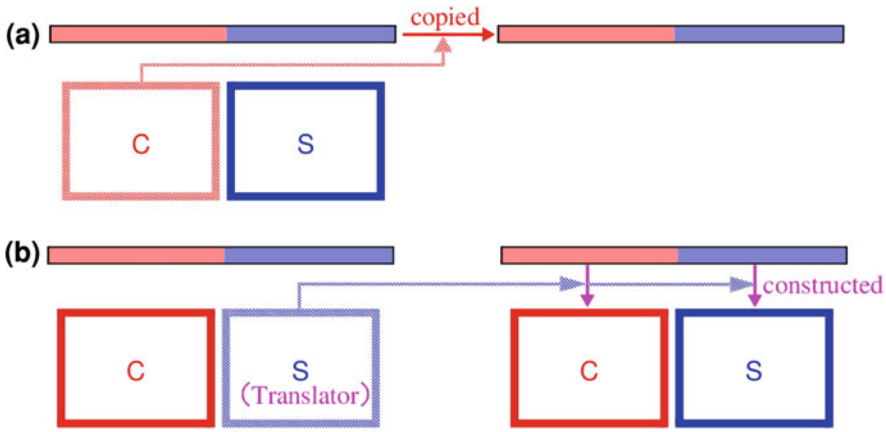


Fig. 3.4 Self-replication cycle of von Neumann automaton

When a mutation occurring during the course of evolution changes the content of the tape, the self-replication cycle is expressed by the equation below.

$$\begin{aligned} \mathrm{C}+\mathrm{S}+\mathrm{T}(\mathrm{C}+\mathrm{S}) &\overset{\text{mutation}}{\longrightarrow} \mathrm{C}+\mathrm{S}+\mathrm{T}(\mathrm{C}'+\mathrm{S}') \\ &\xrightarrow{\mathrm{C}} \mathrm{C}+\mathrm{S}+\mathrm{T}(\mathrm{C}'+\mathrm{S}')+\mathrm{T}(\mathrm{C}'+\mathrm{S}') \\ &\xrightarrow{\mathrm{S}} \mathrm{C}+\mathrm{S}+\mathrm{T}(\mathrm{C}'+\mathrm{S}')\mathrm{C}'+\mathrm{S}'+\mathrm{T}(\mathrm{C}'+\mathrm{S}') \end{aligned}$$

As the tape information changes to C′ and S′, the machines are also modified to C′ and S′. As indicated by this example, if mutations change the tape information in any manner during the self-replication of the automata, Machine S itself, which translates the genetic information (tape information) to the phenotype (machine), also changes. In von Neumann's self-reproducing automata, mutations can modify the translation relationship itself.

3.3 Topological Properties of Intermolecular Interactions

In the preceding section, we considered the properties of the solution as a biochemical reaction field from an evolutionary perspective and from the perspective of emergent functionality. In this section, we will discuss the properties of the solution reaction field from a slightly different perspective—that of its physical and chemical properties. In particular, we provide an overview of the topological conditions for molecular collisions—a prerequisite for reactions—and summarize their topological properties by introducing molecular networks that express the spatial relationships between molecules.

3.3.1 Intermolecular Forces and Chemical Reaction Velocity Theory

Generally speaking, the bonds between the atoms and molecules constituting living creatures fall into four categories, based on bonding energy. In order of increasing energy (each step being roughly a tenfold increase), we have van der Waals bonds, hydrogen bonds, ionic bonds, and covalent bonds (Fig. 3.5).

Table 3.1 shows the origins of the intermolecular forces—forces other than the covalent bond—in greater detail. Molecules can be ions (monopoles having static

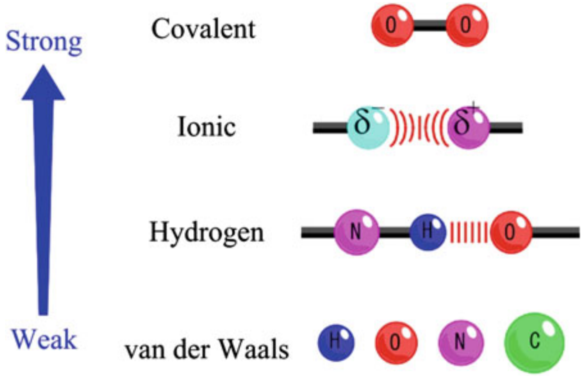
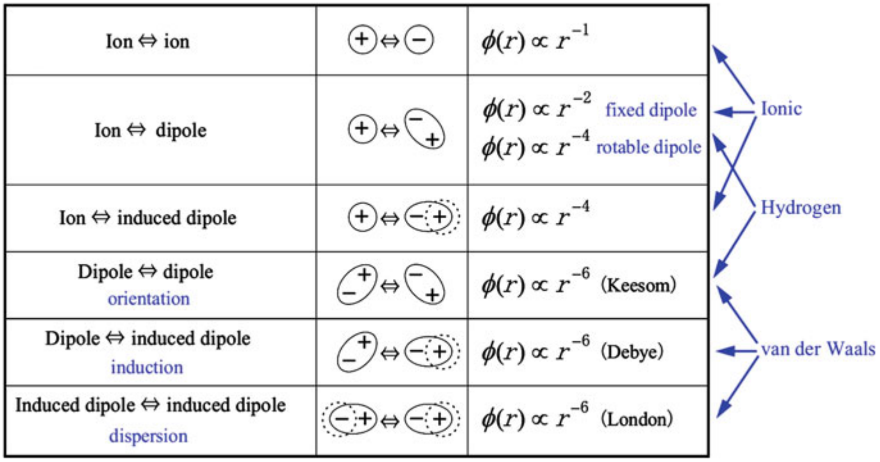


Fig. 3.5 Forces between the atoms and molecules that make up living creatures

Table 3.1 Classification of intermolecular forces of attraction


Ionic
Hydrogen
van der Waals

Ion ⇔ ion	⊕⇔⊖	$\phi(r) \propto r^{-1}$
Ion ⇔ dipole	⊕⇔(− +)	$\phi(r) \propto r^{-2}$ fixed dipole $\phi(r) \propto r^{-4}$ rotable dipole
Ion ⇔ induced dipole	⊕⇔(−:+)	$\phi(r) \propto r^{-4}$
Dipole ⇔ dipole orientation	(+ −)⇔(− +)	$\phi(r) \propto r^{-6}$ (Keesom)
Dipole ⇔ induced dipole induction	(+ −)⇔(−:+)	$\phi(r) \propto r^{-6}$ (Debye)
Induced dipole ⇔ induced dipole dispersion	(−:+)⇔(−:+)	$\phi(r) \propto r^{-6}$ (London)

Type of intermolecular attractive force, schematic diagram, potential energy φ on intermolecular distance r

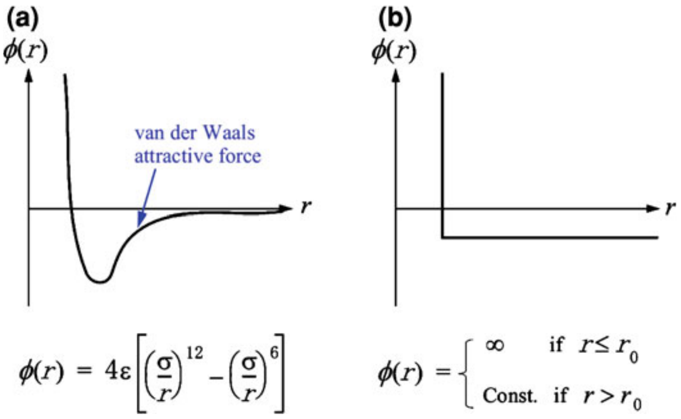


Fig. 3.6 Lennard-Jones potential and hard-sphere potential

charges); dipoles (which have uneven charge distributions in the static state); or induced dipoles (which lack electric charge in the static state but become dipoles when electric charge is induced by a nearby charge). The potential energies of the attractive forces are affected by distances in different ways, depending on the molecule type. In general, attractive forces between molecules without distinct electrostatic charges (for example, between induced dipoles) tend to fall off rapidly with increasing intermolecular distance. (The indices of r dependence of φ in Table 3.1 take large negative values.) Keesom, Debye, and London in this table refer to the individuals who formulated the inter-dipole interactions named orientation, induction, and dispersion, respectively.

Owing to overlapping electron clouds, molecules and atoms experience significant repulsive forces when they come into very close proximity, something not shown in this table. For this reason—for example, in the case of molecules without charge—they experience attractive forces (dispersion forces) when distant and repulsive forces when close together. We can express this as a Lennard-Jones potential (Fig. 3.6a). The Lennard-Jones potential is often used in molecular dynamics to help model interatomic forces. This section simplifies this potential into the hard-sphere potential (Fig. 3.6b) for use in subsequent discussions.

Chemical reaction, namely, the recombination of molecular bonds, starts with the molecular collision. In order to estimate the rate of chemical reactions, we have to calculate the frequency of molecular collisions incorporating the properties of environments (including the gas and liquid phases). This has long been one of the most important topics in chemical reaction velocity theory. The theory succeeded in formulating rate constant k as a function of various physical parameters (including molecule size and temperature) by assuming the molecules to be hard spheres and molecular distributions to be homogeneous. Column 2 describes this in greater detail; interested readers are referred there.

Column 2: Chemical Reaction Velocity Theory

The rate constant k [m^3/(mol s)] (the brackets enclose the unit), which indicates the rate of a chemical reaction, has a different parameter dependence in the gas phase and in the liquid phase. First, consider a basic chemical reaction in the gas phase, as shown in Fig. 3.7 of this column.

Here, I indicates the ingredient molecule. P indicates the product molecule. In general, the time dependence of the concentration $[P]$ [mol/m^3] of P: is expressed as

$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = k[I]^2 \tag{3.1}$$

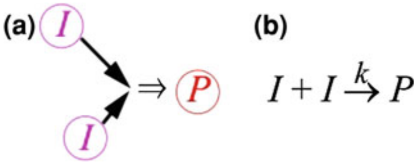
We introduce the following notations:

N:	Number of ingredient molecules [1]
V:	Volume of the gas phase considered [m^3]
m:	Mass of a single ingredient molecule [kg]
d:	Diameter of the ingredient molecule when regarded as a hard sphere [m]
$\langle v \rangle$:	Average speed of ingredient molecules [m/s]
$\langle v_\mathrm{r} \rangle = \sqrt{2}\ \langle v \rangle$:	Average relative speed between ingredient molecules [m/s]
z:	Average collision frequency per ingredient molecule [1/s]
$Z = (1/2)(N/V)z$:	Number of collisions per second per unit volume [1/(m^{3}s)]
ε_0:	Activation energy of the reaction [J = m^{2}kg/s^2]
N_A:	Avogadro number [1/mol]
k_B:	Boltzmann constant [m^{2}kg/(s^{2}K)]
T:	Absolute temperature [K]
$\beta = 1/(k_\mathrm{B}T)$:	Temperature factor [1/J = s^2/(m^{2}kg)]

Here, $\langle v_\mathrm{r} \rangle$ is square root of two times $\langle v \rangle$; the ingredient molecules move near-randomly in the gas phase, and we can assume that the hopping vectors of the ingredient molecules form an average angle of 90°. The factor 1/2 in the expression for Z corrects for the double-counting of collisions (two ingredient molecules take part in a single collision). Based on this notation, we express the concentration of the ingredient molecules as:

$$[I] = \frac{N}{N_\mathrm{A} V} \tag{3.2}$$

Fig. 3.7 Chemical reaction in gas phase



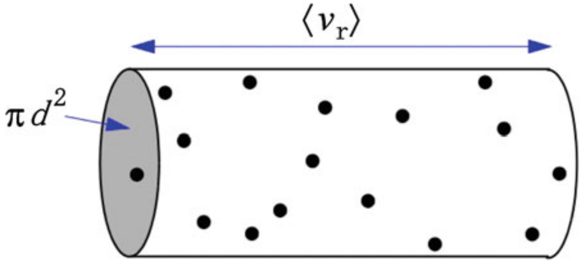


Fig. 3.8 Region through which a molecule in the gas phase moves each second

With the Maxwell–Boltzmann distribution derived from the kinetic theory of gases, we can derive

$$\langle v\rangle = \left(\frac{8k_\mathrm{B}T}{\pi m}\right)^{1/2} \tag{3.3}$$

Assume that each molecule is a hard sphere of diameter d. Then, each second, a molecule engages in z number of collisions with other ingredient molecules, with z being equal to the number of particles—the points representing the central position of other ingredient molecules—in the cylindrical region in Fig. 3.8 of this column. Accordingly,

$$z = \frac{N}{V}\cdot \pi d^2 \cdot \sqrt{2}\langle v\rangle$$

$$\therefore Z = \frac{1}{\sqrt{2}}\cdot\frac{N^2}{V^2}\cdot \pi d^2\langle v\rangle \tag{3.4}$$

The final number of reactions occurring each second—d[P]/dt—is the number of collisions obtained multiplied by the Boltzmann factor which includes the activation energy as:

$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = k[I]^2 = \frac{Z}{N_\mathrm{A}}\cdot\exp(-\beta\varepsilon_0) \tag{3.5}$$

Substituting Eqs. 3.2–3.4 for this equation, we obtain

$$\therefore k = \frac{N_\mathrm{A}}{\sqrt{2}}\cdot \pi d^2\left(\frac{8k_\mathrm{B}T}{\pi m}\right)^{1/2}\cdot\exp\left(-\frac{\varepsilon_0}{k_\mathrm{B}T}\right)$$

Here, $\exp(-\varepsilon_0/(k_\mathrm{B}T))$ is called the Arrhenius factor; the other part is called the pre-exponential factor. Since m is proportional to d^3, the dependence of rate constant k on molecule size d is estimated to be $k^{\propto}d^{1/2}$, indicating a gradual increase with molecular size.

Next, consider the reaction in the liquid phase illustrated in Fig. 3.9 in this column. The liquid phase differs from the gas phase in that the ingredient molecules I and the product molecules P are surrounded by solvent molecules S. An ingredient molecule hops by the collision with the solvent

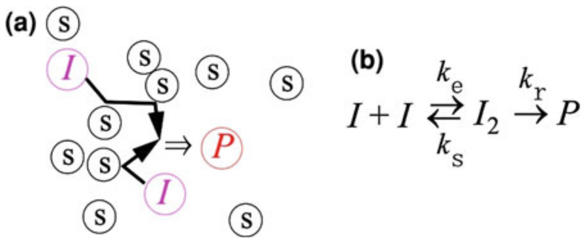

(a)
S
I
P
I
⇒
(b)
$I + I \underset{k_s}{\overset{k_e}{\rightleftarrows}} I_2 \xrightarrow{k_r} P$

Fig. 3.9 Chemical reaction in liquid phase

molecules (Brownian motion), and as a result, encounters and collides with another ingredient molecule. In the liquid phase, these encounter and collision are defined as different events. Even if the two ingredient molecules encounter, if they cannot overcome the activation energy and do not generate P immediately, they collide with each other several times on the spot, and finally react or separate. This means the solute molecules are surrounded by the solvent in the liquid phase in a manner similar to conditions in a cage. Once the molecules belong to the same cage (i.e., once they encounter), they are not readily separated and tend to collide repeatedly. This is called the cage effect.

We can formulate such reactions in the liquid phase by assuming, as in Fig. 3.9b, that two ingredient molecules first form an encounter complex (activated complex) I_2 at the rate constant k_e, and I_2 reacts at rate k_r or separates at rate k_s. We consider the stationary state—the state in which intermediate product I_2 neither increases nor decreases—and derive

$$\frac{\mathrm{d}[I_2]}{\mathrm{d}t} = k_e[I]^2 - (k_s + k_r)[I_2] = 0$$

$$\therefore\ [I_2] = \frac{k_e}{k_s + k_r}[I]^2$$

This equation can be rewritten as:

$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = k_r[I_2] = \frac{k_e k_r}{k_s + k_r}[I]^2 = k[I]^2$$

Here, k is the rate constant of the entire reaction defined as

$$k \equiv \frac{k_e k_r}{k_s + k_r} = k_e - k_s\frac{k_e}{k_s + k_r} = k_e - k_s\frac{[I_2]}{[I]^2} \tag{3.6}$$

With this equation, we can formulate the reaction velocity as:

$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = k[I]^2 = k_e[I]^2 - k_s[I_2] = \frac{Z_e}{N_A} - \frac{Z_s}{N_A} \tag{3.7}$$

Here, Z_e [1/(m^{3}s)] and Z_s [1/(m^{3}s)] are the numbers of encounters and separations per second per unit volume, respectively; $k_e[I]^2 = Z_e/N_A$ and

$k_\mathrm{s}[I_2] = Z_\mathrm{s}/N_\mathrm{A}$ are the numbers of encounters and separations per second per mole, respectively. In the stationary state, the number of product molecules, P, is equal to the number of encounters minus the number of separations.

Now, assume that an ingredient molecule moves with the Brownian motion and approach another (target) ingredient molecule. The relevant parameters are:

r: Distance to the target molecule [m]
$[I]_r$: Density of the ingredient molecules at distance r [mol/m^3]
D: Diffusion coefficient of the ingredient molecule in the solution [m^2/s]
η: Viscosity of the solution [kg/(ms)]
z_e: Average frequency of encounters of an ingredient molecule [1/s]
z_s: Average frequency of separations of an ingredient molecule [1/s]

Here, we can relate z_e and z_s to the numbers of encounters and separations, respectively, by

$$Z_\mathrm{e} = (1/2)(N/V)z_\mathrm{e}$$

and by

$$Z_\mathrm{s} = (1/2)(N/V)z_\mathrm{s}.$$

We can calculate D from η and other parameters using the Einstein–Stokes's relation

$$D = \frac{k_\mathrm{B}T}{6\pi\eta(d/2)} \tag{3.8}$$

Though in a solution all ingredient molecules diffuse with D, here we assume that the target ingredient molecule is fixed in space and other molecules diffuse with magnitude $2D$, which gives an equivalent analysis. Around the target molecule, other ingredient molecules collide the target, disappear, and generate product P; hence in this area (Fig. 3.10a), we can assume that density $[I]_r$ of the ingredient molecules takes a minimum value at $r = d$ and increases as r increases until reaching bulk value $[I]$ at infinity ($r = \infty$) (shown by the upper curve in Fig. 3.10b of this column).

Here, using Fick's Law, the number j_r (1/s) of ingredient molecules that cross outwardly the surface of a sphere of radius r centered at the target molecule per unit time can be expressed as

$$j_r = -4\pi r^2 \cdot 2D \cdot N_\mathrm{A} \cdot \frac{\mathrm{d}[I]_r}{\mathrm{d}r} \tag{3.9}$$

Since in the stationary state, j_r must have constant value j that is independent of distance r (the law of conservation of mass), if we integrate both

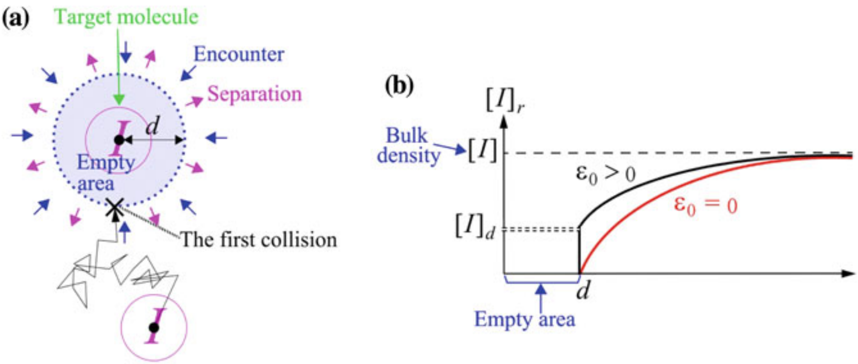


Fig. 3.10 Conditions around the target molecule in the liquid phase and distance dependence of molecular density

sides of the equation above with respect to r, j goes out of the integral and we obtain

$$\int_d^\infty \frac{j}{4\pi r^2}\mathrm{d}r = -\int_d^\infty 2D \cdot N_{\mathrm{A}} \cdot \frac{\mathrm{d}[I]_r}{\mathrm{d}r}\mathrm{d}r$$

$$\therefore j = -4\pi d \cdot 2D \cdot N_{\mathrm{A}} \cdot ([I] - [I]_d)$$

On the other hand, as shown in Fig. 3.10a, j is the frequency of separations minus the frequency of encounters for a target molecule: $j = z_{\mathrm{s}} - z_{\mathrm{e}}$. Thus, $Z_{\mathrm{e}} - Z_{\mathrm{s}}$ can be formulated as:

$$\begin{aligned} Z_{\mathrm{e}} - Z_{\mathrm{s}} &= \frac{1}{2}\frac{N}{V}(z_{\mathrm{e}} - z_{\mathrm{s}}) = \frac{1}{2}\frac{N}{V}(-j) \\ &= 4\pi d \cdot D \cdot N_{\mathrm{A}}^2[I]([I] - [I]_d) \\ &= 4\pi d \cdot D \cdot N_{\mathrm{A}}^2[I]^2 \exp(-\beta\varepsilon_0) \end{aligned} \tag{3.10}$$

The last equation is derived from the argument that positive activation energy ($\varepsilon_0 > 0$) induces separation. If we assume the reaction lacks activation energy ($\varepsilon_0 = 0$), all collisions instantly generate product P. (This state of affairs is said to be *diffusion-controlled*, since the rate of reaction is determined solely by diffusion.) When this happens, the ingredient molecules are no longer present on the surface of the sphere of radius d, which leads to $[I]_d = 0$; in other words, $[I] - [I]_d = [I]$ (the lower curve in Fig. 3.10b). If we assume activation energy is extremely large ($\varepsilon_0 = \infty$), on the other hand, collisions between the ingredient molecules never causes the generation of P. In this case, the number of separations equals the number of encounters ($Z_{\mathrm{e}} = Z_{\mathrm{s}}$), which gives $[I] - [I]_d = 0$. In intermediate cases ($0 < \varepsilon_0 < \infty$), we can assume that $[I] - [I]_d$ has the ε_0 dependence described by the Boltzmann factor.

From Eqs. 3.7, 3.8, and 3.10, we can formulate k as

$$\begin{aligned} k[I]^2 &= \frac{Z_\mathrm{e}}{N_\mathrm{A}} - \frac{Z_\mathrm{s}}{N_\mathrm{A}} = 4\pi d \cdot D \cdot N_\mathrm{A}[I]^2 \exp(-\beta\varepsilon_0) \\ \therefore k &= \frac{4k_\mathrm{B}T \cdot N_\mathrm{A}}{3\eta_T} \cdot \exp\left(-\frac{\varepsilon_0}{k_\mathrm{B}T}\right) \end{aligned} \tag{3.11}$$

Here, viscosity η is temperature-dependent and expressed as η_T. As with Eq. 3.5 for the gas phase, $\exp(-\varepsilon_0/(k_\mathrm{B}T))$ on the right hand side of Eq. 3.11 is called the Arrhenius factor. The remaining part is called the pre-exponential factor. Equation 3.11 differs from Eq. 3.5 in that it does not depend on size d or mass m of the molecule. This is attributable, more or less, to the mutually canceling effects of the slower motion and larger cross section of the collision of larger molecules in the liquid phase.

The viscosity η_T is often formulated with Andrade's law as

$$\eta_T = A \exp\left(\frac{\varepsilon_0}{k_\mathrm{B}T}\right) \tag{3.12}$$

This law applies to many liquids, but not, unfortunately, to water, the biological solvent.

Author's Refs. [87–89] for this column. Readers wishing to learn more about chemical reaction velocity theory are referred to these works.

3.3.2 Topological Conditions on Molecular Movement

The chemical reaction velocity theory is key to estimating rates of chemical reactions. However, the theory alone is inadequate for building virtual models to track each symbol (molecule) moving in information space within the artificial chemistry framework. Since chemical reaction velocity theory posits the diffusion of gases and liquids in three-dimensional Euclidean space and assumes homogeneous molecular distributions, it says nothing about what molecule, one particular molecule will most likely meet after colliding with another molecule.

A solute molecule that diffuses with Brownian motion is located in a three-dimensional Euclidean space. This means that a solute molecule is more likely to encounter a spatially close molecule and that molecules located far from each other are unlikely to meet soon. In preparing an artificial space and designing rules for the transport and collision of symbols, we need to consider such topological properties of molecular positions, which chemical reaction velocity theory does not specify explicitly.

In order to represent such constraint, here we set the three conditions as prerequisites for the information space and for the rules for transport of symbols.

These conditions set the ground rules for transport and compartmentalization of symbols in the artificial system, and should be satisfied for the system to comply with Emergence Conditions (5)–(11).

Topological Condition (1): The concept of distance applies between the symbols.

Topological Condition (2): The distance between the symbols can change over time.

Topological Condition (3): Collisions and reactions between symbols are more likely to occur if the distances separating them are smaller.

3.3.3 Intermolecular Distance and the Molecular Network

This section discusses the topological relationship between the molecules described in the Sect. 3.3.2 from a more quantitative perspective. First, we consider the vicinity matrix $V = (v_{ij})$ below. Element v_{ij} of the matrix **V** is a variable that can be expressed as $v_{ij} = 1/r_{ij}$, with distance r_{ij}, between the ith and jth molecules. Matrix **V** expresses the proximity relation (topological distance) between all molecules in the target region, which can be expressed, for example, as:

$$\mathbf{V} = \begin{bmatrix} \infty & 0.21 & 0.79 & \cdots & \\ 0.21 & \infty & 0.02 & \cdots & \vdots \\ 0.79 & 0.02 & \infty & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ & \cdots & \cdots & \cdots & \infty \end{bmatrix}$$

V is an **N** × **N** symmetric square matrix incorporating all topological information between the molecules, except information on angles (**N** is the total number of molecules). Since analyzing the matrix in this form is difficult, we simplify it based on our knowledge of intermolecular interactions. As discussed in Sect. 3.3.1, intermolecular forces, particularly those operating on molecules without electrostatic charge, rapidly fall to zero as the distance between the molecules increases. Thus, intermolecular interactions are negligible at distances above a certain value. As discussed with chemical reaction velocity theory in the liquid phase (refer to Column 2, discussed earlier), reactions in the liquid phase can be considered in two stages: encounters and reactions. In order to react, two molecules must approach each other and belong to the same cage (encounter). On that basis, we binarize the elements v_{ij} of the matrix **V** by substituting with a 0 when a value corresponding to an element is below a threshold value (i.e., when the intermolecular interaction is negligible) and by substituting with a 1 otherwise (i.e., when the molecules participate in an encounter). This generates encounter matrix $\mathbf{A} = (\delta_{ij})$, a new binary matrix. If the threshold value is 0.5, for example, the matrix **A** obtained from matrix **V** is expressed as:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & \cdots & \\ 0 & 1 & 0 & \cdots & \vdots \\ 1 & 0 & 1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ & \cdots & \cdots & \cdots & 1 \end{bmatrix}$$

The graph (network) having this matrix as the adjacency matrix expresses the spatial relationship of the molecules as a graph topology, although only approximately. We call this graph an encounter network or molecular network.

3.3.4 Topological Properties of the Molecular Network

A molecular network is a graph, and we can apply graph theory to examine its characteristics. As the derivation in Sect. 3.3.3 clearly shows, a randomly generated binary matrix cannot be the adjacency matrix of a molecular network. A molecular network provides a simplified expression of the positional distribution of molecules in Euclidean space—the topology of which has special properties, reflecting the original spatial relationship of the molecules.

This section focuses on the topological properties of the molecular network and examines the static and dynamic properties of edge rewiring, which reflects the Brownian motion of the molecules. We simulate the movement of the molecules in the solution as a random walk of hard spheres in a two- or three-dimensional Euclidean space. We also calculate various characteristic quantities and edge creation or deletion probabilities in the encounter network of the hard spheres produced in the simulation [69].

The results obtained can be summarized as the topological properties of the molecular network:

Topological Property (1): Cluster coefficient C is approximately 10–20 times that for the random graph.

Topological Property (2): Average path length L is approximately 1.6–2 times that for the random graph.

Topological Property (3): Diameter D is small.

Topological Property (4): Tangledness T is low.

Topological Property (5): Edge joining probability P_{join} decreases linearly as the degree k of the end point increases.

Topological Property (6): Edge joining probability P_{join} decreases exponentially as shortest path length l between the end points increases.

Topological Property (7): Edge cutting probability P_{cut} is large when degree k of the end point or second shortest path length l_2 between the end points is large.

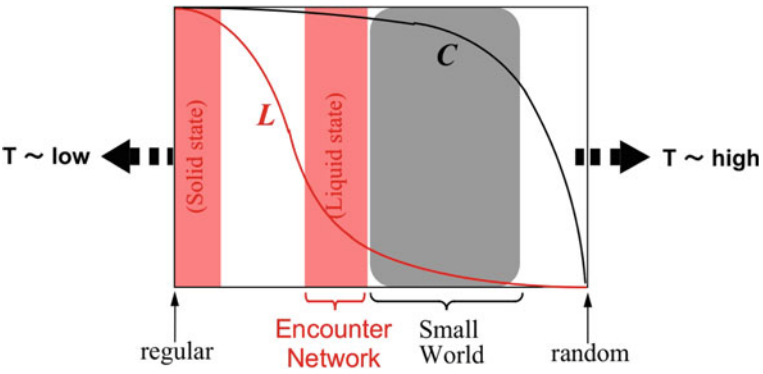


Fig. 3.11 Positioning of encounter network shown schematically in the C and L's plot with the Watts and Strogatz's scenario

Cluster coefficient C and average path length L are parameters that characterize small-world networks (see Column 3), investigated intensively in recent studies of complex networks. According to the C and L's values specified by Properties (1) and (2), we see that a molecular network is positioned somewhere between a regular network (corresponding to the solid phase) and a random network (corresponding to the gas phase). Figure 3.11 illustrates this state with plots for L and C values. As discussed in Column 3 further below, a small-world network is also a graph intermediate between the two network extremes. However, as the L value suggests, the encounter network is positioned somewhat closer to a regular network than a small-world network (Fig. 3.11).

Here, diameter D in Topological Property (3) is the maximum value of the shortest path length between all node pairs in the graph, which is large for a misshapen or distributed graph. Tangledness T in Topological Property (4) is measured by the number of edge crossings when the graph is projected onto a plane; it assumes a large value when the degree of tangling of the graph is large. Since a molecular network is a network defined by the positions of molecules in Euclidean space, it takes small values for D and T.

Column 3: Small-World Network

As described in a paper by Watts and Strogatz [1], a seminal publication in complex network studies, a *small-world network* refers to a group of graphs with special properties. In order to characterize a small-world graph, we need to define two parameters that express the graph's characteristics: cluster coefficient C and average path length L.

Cluster coefficient C is defined as follows:

$$C = \frac{1}{N}\sum_{i=1}^{N} C_i, \quad C_i = \frac{2E_i}{k_i(k_i - 1)}$$

Here, N is the number of nodes, k_i the degree of the ith node (the number of edges connected to the node), and E_i the number of edges connecting the node pairs adjacent to the ith node. C_i corresponds to the ratio of the edge pairs constituting triangles (clusters) among the $k_i(k_i - 1)$ edge pairs selected from the k_i edges.

Average path length L is defined as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i, \quad L_i = \frac{1}{N-1}\sum_{j \neq i} d_{ij}$$

Here, d_{ij}, the shortest path length connecting the ith and jth nodes, is assumed to have a finite value; in other words, the graph is connected.

A small-world network is defined, using these parameters, as a graph with large C and small L. A regular graph with highly regular topology has large C and large L; a random graph has small C and small L. A small-world network is positioned between these extremes. Alluding to a colloquial truism ("It's a small world!"), *small world* reflects the short paths connecting many node pairs; that is, small L.

In [90], Watts and Strogatz demonstrated the ubiquity of networks with small-world characteristics throughout the human sphere, including the Internet and social networks, and proposed a scenario for building such networks artificially (see Fig. 3.12 in this column). The simple scenario starts with a regular network with a one-dimensional structure. Then edges are randomly chosen and rewired, with which the graph gradually approaches a random network. This rewiring initially rapidly diminishes L, while C declines at a slower rate. A graph with small-world characteristics emerges as an intermediate state on the path to becoming a random graph.

Though a regular graph, the initial condition assumed in the Watts and Strogatz's scenario, is rarely observed in nature or in society, Davidsen et al. [91] subsequently proposed a scenario starting from a random graph and claimed that human social networks has small-world characteristics on account of the scenario. Similarly, the energy-based NAC rewiring rule

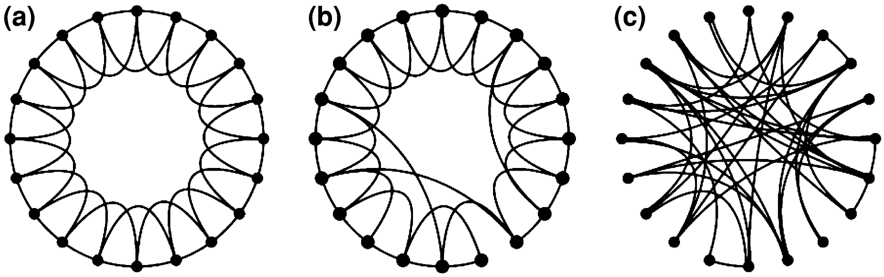


Fig. 3.12 Scenario by Watts and Strogatz (reproduced from [1]). **a** Starting from a one-dimensional regular network, the edges are randomly rewired. **b** A small-world network emerges. **c** A random network appears

described in Sect. 3.5 generates a molecular network (a kind of small-world network) starting from a random graph. These scenarios not starting from the initial regular graph create a small-world network, so to speak, in more realistic ways than by the Watts and Strogatz's scenario.

Chapter 5 of this book provides an overview of the application of small-world and other complex networks to the real world. An extensive body of literature exists on the topic, some in Japanese. We refer interested readers to these works [92–95].

Random-Walk Simulation of Hard Spheres

In order to model molecular diffusion in a solution (Brownian motion) by a random walk in Euclidean space, we consider a D-dimensional continuous space that satisfies the periodic boundary conditions as in Fig. 3.13. In this space, we introduce N D-dimensional hard spheres of radius R_1, and for each iteration, each hard sphere moves its center coordinates (hops) by a randomly chosen hopping vector. If the sphere hopping reduces the distance between the center points of the two hard spheres to below $2R_1$, the hopping is canceled. In order to simplify the problem and reduce computational costs, R_1 and hopping vector length $\varDelta$ are assumed to be constant for all spheres and all hops. In addition, the direction of the hopping vector is not arbitrary, but randomly selected from $2D$ discrete values (a pair of positive and negative values for each dimension).

Collisions and contacts involving hard spheres are defined using contact radius R_2. When Euclidean distance d between the center points of a pair of hard spheres satisfies the condition $2R_1 \le d \le 2R_2$, the pair is regarded to be in contact (encounter). R_2 is taken to equal $2R_1$, reflecting the condition that the two spheres are regarded to be in contact if another hard sphere cannot enter between the two hard spheres ([8]; p. 890 of Japanese translation). While the contact graph (encounter network) thus created changes over time according to the random-walk model, we measure several graph characteristics.

First, we measure cluster coefficient C and average path length L of the encounter network, based on the usual definitions of these two variables. We also calculate edge joining and cutting probabilities P_{join} and P_{cut}, respectively, as follows.

Assume that Node A moves at a specific point in time and forms a new edge with Node C. Here, we measure directly l, the shortest path length between Nodes A and C, and degree k of Node C immediately before the edge forms, then add 1 to the frequency matrix $N_{\mathrm{join}}(l,k)$. We also measure l and k for all nodes in the graph, including Node C, and add 1 to $N(l,k)$. At the end of the simulation, we calculate $P_{\mathrm{join}}(l,k) = N_{\mathrm{join}}(l,k)/N(l,k)$ to obtain the expectation value of P_{join}, the edge joining probability. Likewise, if Node A moves at a specific point in time, eliminating Edge AB, we measure directly l_2, the second shortest path length between Nodes A and B, and degree k of Node B, immediately before the edge disappears, then add 1 to frequency matrix $N_{\mathrm{cut}}(l_2,k)$. We also check l_2 and k for all

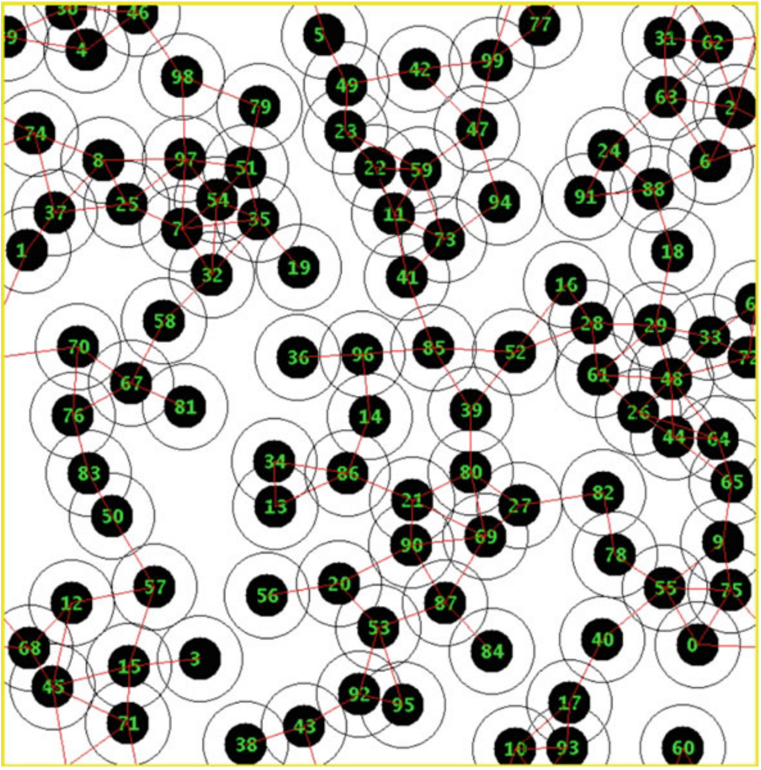


Fig. 3.13 Snapshot of hard-sphere random-walk simulation in two-dimensional space

nodes adjacent to Node A and add 1 to $N_2(l_2,k)$. At the end of the simulation, we calculate P_{cut} by using $P_{\text{cut}}(l_2,k) = N_{\text{cut}}(l_2,k)/N_2(l_2,k)$.

The parameter settings for this simulation are too detailed for the present discussion and are summarized in Column 4. We refer interested readers there.

Column 4: Parameter Setting for Hard Sphere Random-Walk Simulations

We calculate the typical values for the ratio R_1/Δ of the radius of the hard spheres and hopping vector length, based on the assumption that the hydrophilic head of the lipid molecule (solute) is surrounded by water (solvent) and is characterized by Brownian motion. We denote the number of collisions between the solute molecule and the solvent molecule per second as z, the radius of the solute molecule as r_1, and the average relative velocity of the solute molecule with respect to the solvent molecule as $\mathbf{V}_r$. Here, z equals the average number of solvent molecules in a cylinder with radius $R_1 + r_1$ and height $|\mathbf{V}_r|$ (see Column 2, previously discussed) and can be expressed as shown in Fig. 3.14a. Here, ρ is the number of solvent molecules per unit volume. $\mathbf{V}_r$ is the composite vector of the average velocity vector $\mathbf{V}$ of the solute molecules and the average velocity vector $\mathbf{v}$ of the solvent molecules. In this case, we can assume that the directions of the solvent molecules are completely random. Thus, $\mathbf{v}$ and $\mathbf{V}$ can be assumed to be orthogonal to each other. The resulting expression is as given in Fig. 3.14b. Here, we use the principle of equipartition of energy, $(1/2)M|\mathbf{V}|^2 = (1/2)m|\mathbf{v}|^2$, to estimate the velocity ratio as shown in Fig. 3.14c. $M = 255$ and $m = 18$ are the molecular weights of the lipid head ($C_8O_6PNH_{18}$) and water (H_2O), respectively. The equations shown in Fig. 3.14a, b give the $R_1/\varDelta$ ratio shown in Fig. 3.14d. Based on this reference value, values for radius, contact radius, and hopping vector length of $R_1 = 4.0$, $R_2 = 8.0$, and $\varDelta = 0.1$, respectively, are selected for this experiment.

We can evaluate average degree $\bar{k}$ of the encounter network if the hard spheres are randomly distributed in space as follows. Denote the volume of the D-dimensional sphere with radius r as $V_D\,(r) = \pi^{D/2} r^D/(D/2)!$. In view of

(a)
$$z = \pi(R_1 + r_1)^2 \cdot |\boldsymbol{V}_{\mathrm{r}}| \cdot \rho$$

(b)
$$|\boldsymbol{V}_{\mathrm{r}}| = \sqrt{|\boldsymbol{V}|^2 + |\boldsymbol{v}|^2} = \sqrt{1 + \frac{255}{18}} \cdot |\boldsymbol{V}| = 3.89 \cdot |\boldsymbol{V}|$$

(c)
$$\frac{|\boldsymbol{v}|}{|\boldsymbol{V}|} = \sqrt{\frac{M}{m}} = \sqrt{\frac{255}{18}}$$

(d)
$$\begin{aligned}
\frac{R_1}{\Delta} &= \frac{R_1}{|\boldsymbol{V}|/z} \\
&= \frac{R_1}{|\boldsymbol{V}|} \cdot \pi(R_1 + r_1)^2 \cdot |\boldsymbol{V}_{\mathrm{r}}| \cdot \rho \\
&= 3.89 \cdot R_1 \cdot \pi(R_1 + r_1)^2 \cdot \rho \\
&= 3.89 \times (3.63 \times 10^{-10}[m]) \times 3.14 \\
&\qquad \times (3.63 \times 10^{-10}[m] + 1.5 \times 10^{-10}[m])^2 \times 3.34 \times 10^{28}[1/m^3] \\
&\simeq 39.0
\end{aligned}$$

(e)
$$\frac{N}{X^D - N \cdot V_D(R_1)}$$

(f)
$$\bar{k} = \frac{N}{X^D - N \cdot V_D(R_1)} \times (V_D(2R_2) - V_D(2R_1))$$

Fig. 3.14 Equations for parameter settings

the number density, the state in which N hard spheres with volume V_D (R_1) are distributed in a cube with sides of length X can be approximated by the state in which N points with zero volume are in the volume of $X^{\mathrm{D}} - N{\cdot}V_D$ (R_1). Here, the number density is expressed as shown in Fig. 3.14e. $\bar{k}$ is this density multiplied by the volume of the D-dimensional spherical shell of inner radius greater than or equal to $2R_1$ and outer radius less than or equal to $2R_2$, which is obtained as shown in Fig. 3.14f. From this equation, for example, under the condition $D = 3$, $X = 93.0$, $N = 200$, $R_1 = 4.0$, and $R_2 = 8.0$, the average degree is estimated to be $\bar{k} = 4.0$.

Simulation Results

The experiment is performed with both $D = 2$ and $D = 3$. In both cases, the C and L values are confirmed to have the following characteristics. The C value increased to 10–20 times that for the random graph. The L value maintained a value approximately 1.6–2 times that for the random graph.

Figure 3.15 shows results for edge joining and cutting probabilities for the experiment with $D = 3$. (We confirmed that the results for $D = 2$ are similar.) Based on the results, where l is small, P_{join} (l, k) decreases exponentially as l increases, and decreases linearly as k increases. This result is intuitive based on what we know of molecular collisions. The collision probability is regarded to decrease as intermolecular distance increases due to the increase in l and with declining effective cross sections of collisions due to contact resulting from increased k. Here, according to Fig. 3.15a, in the region of $l = 2$ and $k \sim$ large, P_{join} increases slightly. This result is only observed under these particular conditions and is regarded to be attributable to noise. In the region with large l, P_{join} increases again, due to confined space. If the size of the space and the number of nodes in the graph are infinite, $N\,(l, k)$ in the denominator increases infinitely as l increases. However, in a finite graph, $N\,(l, k)$ decreases in a region with $l \sim$ large, and P_{join} is thus regarded to increase based on the actions of a small number of exceptional nodes.

Figure 3.15b shows that P_{cut} (l_2, k) is kept small in the region with $l_2 \sim$ small but rapidly increases when l_2 is above a certain value. The threshold value of l_2 decreases as k increases. For example, when $k = 8$, P_{cut} (l_2, k) is extremely large, even when $l_2 = 3$. This result can be understood by the "unnaturalness" of the graph topology with spatial restrictions considered. As the C and L values discussed above indicate, the encounter network has a type of regularity attributable to spatial restrictions. One conspicuous characteristic of a D-dimensional regular graph is that the nodes are distributed almost evenly in D-dimensional space and therefore the graph contains many paths of similar path length as the shortest path between a node pair. (In other words, the frequency is quite large near the shortest path in the histogram of the path length between node pairs.) An edge with a large second shortest path length ($l_2 \sim$ large) surrounded by many nodes ($k \sim$ large) contradicts this characteristic. The encounter network cuts such an edge with high probability, resulting in approaching a regular graph.

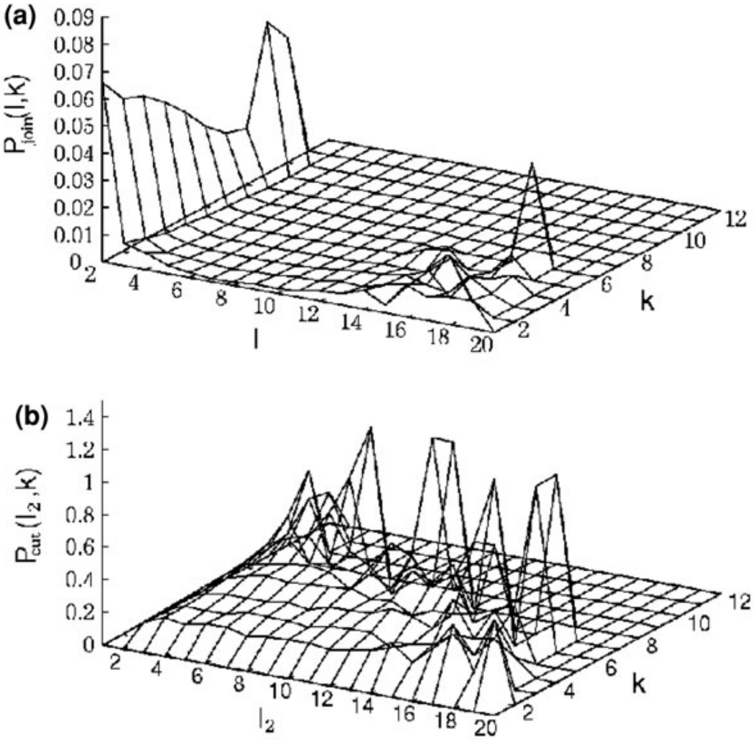


Fig. 3.15 Results of hard-sphere random-walk simulation. **a** Edge joining probability P_{join} between node pairs. (l is the shortest path length between pairs; k is the degree of one of the end points.) **b** Edge cutting probability P_{cut}. (l_2 is the second shortest path length between the end points of an edge; k is the degree of one of the end points.) $N = 200$ hard spheres are placed in Euclidean space with dimension $D = 3$ and length of sides $X = 93.0$. The radius of the hard spheres is $R_1 = 4.0$, the contact radius $R_2 = 8.0$, and the hopping vector length $\Delta = 0.1$. Under these conditions, the volumetric density of the hard spheres is 0.067, and the measured degree of the contact graph is $\bar{k} = 3.8$. These values are average values over 100 thousand iterations in a simulation that required about some 4 days to run on a desktop PC

3.4 Evaluating Artificial Chemistry Systems

Section 3.2 summarized the conditions necessary for the basic design of an artificial chemistry, based on the first stages in the evolution of life and functional emergence. Section 3.3 summarizes the topological conditions and topological properties of molecular networks, focusing on the positions and movement of molecules. Referring to these conditions, this section will verify and assess the basic designs of the artificial life and artificial chemistry systems proposed so far.

We select 26 representative systems to evaluate from those previously proposed and studied. Table 3.2 lists up the five elements of artificial chemistry systems comprising these systems. The systems are categorized by the following features:

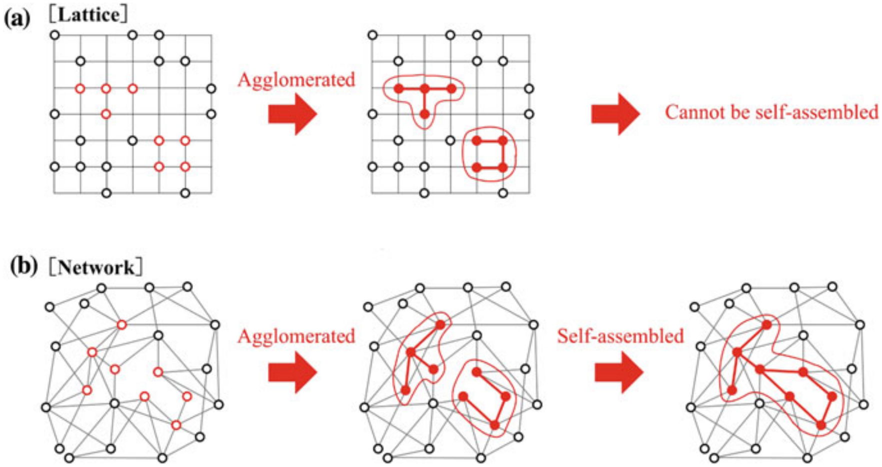


Fig. 3.16 Mobility of symbol complex

whether symbol types are fixed (determinative) or can be increased by combination (combinatorial); whether the space is rigid or elastic; and the dimensions of the space. (See Emergence Condition (6) in Sect. 3.2.2 for a detailed discussion of the rigidness of space.)

Table 3.3 gives the results of the evaluation. Reference [61] gives a detailed justification of the evaluation on which the results are based, although for part of the systems and for Conditions (1)–(10) only.

As Table 3.3 clearly shows, not one system among the 26 systems satisfies all 16 conditions. Emergence Conditions (12) and (13) have especially low pass rates. Condition (13) is met by few lattice spaces of discretized space. Figure 3.16 shows why.

In a lattice space like the one shown in Fig. 3.16a, when symbols assemble and form complexes occupying a large region in the lattice space, it becomes increasingly difficult to move the complexes within the space. Merely shifting a complex to the next lattice point entails checking the vacancy of many adjacent lattice points and associated computational costs. Rotational movement is out of the question. In contrast, these issues are less apparent in networks like the one shown in Fig. 3.16b. Among the systems evaluated, the planar graph of Speroni et al. and NAC fall into this category. Avida and Ameba are exceptions among lattice spaces. These systems always map a symbol complex to a single lattice point as a single program, so that the measure of mobility remains constant, regardless of the size of the complex.

The space and the transport rules for the symbols in the space form the most basic part of the framework of an artificial environment. The design of this framework significantly affects the quality of the model. All of the models examined here feature advantages and disadvantages, but none can generate self-organizing walls and flexibly express the transport/self-assembly of symbols.

Table 3.2 Five elements of artificial life and artificial chemistry systems

Example	(i) Information space	(ii) Elementary symbol	(iii) Reaction rule	(iv) Transport rule	(v) Higher-level manager
Tierra	1D core memory rigid	Machine instructions	Computational functions of machine instructions	Replication by copy command	Mutation, reaper (creature killer), time slicer, etc.
Modified Tierra	(Same as above)	(Same as above)	(Same as above)	(Same as above)	Same as above plus bonus conferrer (when the problem is solved)
SeMar	1D core memory elastic	Machine instructions and operands (classified as DNA/protein/membrane/nutrient)		Word transport rule by protein command	Mutation, reaper (creature killer), etc.
ID Proto-ell model	1D discrete elastic	Particles corresponding to solute and solvent molecules (classified as water/membrane/enzyme/resource and so forth)	Fixed metabolic reaction rules for synthesizing membrane particles from water and resource particles	Random walk of particles between lattice points, incorporating attractive and repulsive forces between nearby particles	
von Neumann's CA	2D discrete rigid (square lattice)	States of automata attributed to lattice points	Transition rules for automata	Transition rules for automata causing state propagation	
CA	2D/3D discrete rigid (square/cubic lattice)	(Same as above)	(Same as above)	(Same as above)	
Swarm chemistry	2D continuous rigid	Molecular agents with position, velocity, and characteristics vectors		Fixed rules for changing directions of agents	Human alchemist performing interactive selection and mixing of molecular species

(continued)

Table 3.2 (continued)

Example	(i) Information space	(ii) Elementary symbol	(iii) Reaction rule	(iv) Transport rule	(v) Higher-level manager
SCL-model	2D discrete rigid	Particles representing resources (substrate), catalysts, and membranes (link)	Inter-particle reaction rules for synthesizing/decomposing/joining membrane particles	Randomized transport rules conserving bonds between membrane particles	
Squirm3	2D continuous/discrete rigid	Particles ("atoms" or "enzymes") of fixed types and variable states	Externally prepared fixed state transition rules or state transition rules generated by interpreting enzyme states	Newtonian motion equation, incorporating attractive and repulsive forces between nearby particles	
Jonny Von	2D continuous rigid	T-shaped or cross-shaped particles (codons)	Transition rules for charges, states, and bonds of particles induced by collisions	Motion equation incorporating attractive and repulsive forces between nearby particles, friction (viscosity), and Brownian motion (noise)	
Dissipative particle dynamics (DPD)	2D/3D continuous rigid	Particles corresponding to solute and solvent molecules (particles having mass)		MD-like motion equation incorporating attractive and repulsive forces between nearby particles, friction (dissipative force), and random forces (noise)	
Langevin equation model	2D/3D continuous rigid	Particles corresponding to solute molecules (particles having mass, hydrophilic or hydrophobic characteristics, etc.)	Fixed metabolic reactions to synthesize membrane particles from resource particles	Langevin equation incorporating attractive and repulsive forces between nearby particles, friction (viscosity), and Brownian motion (noise)	
Cellular Potts model (CPM)	2D discrete rigid (square/hexagonal lattice)	Spin value of 0 or 1	Spin transition rules incorporating adhesion and spatial restrictions (energy minimization by the Monte Carlo method)		

(continued)

Table 3.2 (continued)

Example	(i) Information space	(ii) Elementary symbol	(iii) Reaction rule	(iv) Transport rule	(v) Higher-level manager
Avida	2D discrete elastic	Machine instructions	Computational functions of machine instructions	Replication by copy command, diffusion to adjacent lattice points associated with division of the creature, data transfers between adjacent lattice points	(Same as modified Tierra)
Ameba	(Same as above)	Patterns (codons) and machine instructions	(Same as above)	(Same as above)	(Same as Tierra)
SIVA	2D discrete elastic	Alphabets corresponding to small molecules and monomers	Functions encoded in virtual protein (short alphabetical strings)	Diffusion to adjacent lattice points associated with division of the creature	Modification of alphabets in the division of a creature (e.g., mutation and crossover operations)
2/3D Proto-cell model	2D/3D discrete elastic	(Same as 1D Proto-cell model, except that membrane particles have specific orientations)	(Same as 1D Proto-cell model)	(Same as 1D Proto-cell model)	
ARMS/ACS	Nested multisets	Characters (classified as passive or active)	Rewriting rules for passive characters defined in active characters	Transfer rules of characters between multisets	Division and removal rules for nesting, and mutation in character rewriting rules
AChemy	Tank	λ-expression	Rules for the λ calculus		Reaper for conserving the number of λ-expressions
Assembler automaton	Tank	Fixed-length binary numbers	Computational functions of machine instructions interpreting binary numbers		Reaper for conserving the number of λ-expressions

(continued)

Table 3.2 (continued)

Example	(i) Information space	(ii) Elementary symbol	(iii) Reaction rule	(iv) Transport rule	(v) Higher-level manager
Combinator model	Tank	Characters (basic combinators) or strings structured by "()" (combinators)	String rewriting rules defined in each basic combinator		Operation for conserving the number of basic combinators
SAC	Tank	Characters (classified as passive or active) or strings	Externally prepared fixed interpretation rules from active strings to rewriting rules		Operations for decomposing strings and supplying/disposing of characters
Tominaga et al.	Tank	Characters (elements) or strings (objects)	Externally prepared fixed recombination rules for strings		
Tile automaton	2D continuous rigid	Tile clusters of various shapes	Rules for the creation and deletion of tile clusters caused by collisions of tiles	Newtonian motion equation and collision constraints	
Speroni et al.	2D triangular graph	(Same as combinator model)	(Same as combinator model)	Rules for the creation and deletion of contact edges resulting from reactions	
NAC	Graph assuming spatial restrictions	Characters (or strings) attributed to nodes (with hydro-properties, operation codes, or programs)	Interpretation and execution rules for operation codes/programs for modifying strings and rewiring strong edges	Passive rewiring rules for weak edges (such as network energy minimization by Monte Carlo methods)	Operations for folding node chains, token generation, etc.

The references for the systems used as examples are given in the order of their appearance: Tierra [50, 33], modified Tierra [2], SeMar [58, 59], 1D Proto-cell model [39], von Neumann's CA [84], CA [26, 29, 30, 52, 53], Swarm chemistry [54], SCL-model [34, 35, 43, 75], Squirm3 [22–25], JohnnyVon [13, 55], DPD [12, 19, 21, 85], Langevin equation model [7, 37], CPM [17, 18, 46], Avida [1, 3, 4], Ameba [47–49], SIVA [42–45], 2D/3D Proto-cell models [31, 38, 40], ARMS/ACS [76–79], AlChemy [14, 15], Assembler automaton [10], Combinator model [56], SAC [41, 60], Tominaga et al. [81–83], Tile Automaton [86], Speroni et al. [57], NAC [62–70]

Table 3.3 Evaluation results for artificial life and artificial chemistry systems with respect to 16 conditions

			Emergence condition													Topological condition		
Symbol	Space	Example	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(1)	(2)	(3)
Determinative	1D rigid	Tierra	△	○	○	△	○	△	○	×	○	○	○	×	×	○	○	○
		Modified Tierra	△	○	○	△	○	△	○	○	○	○	○	×	×	○	○	○
	1D elastic	SeMar	△	○	○	△	○	○	×	○	○	○	×	○	×	○	○	○
		ID Proto-cell model	○	×	○	×	○	○	×	×	×	△	×	×	×	○	○	○
	2D/3D rigid	von Neumann's CA	×	○	○	○	×	△	○	○	○	○	×	×	×	○	○	○
		CA	×	×	○	△	○	△	○	○	○	○	○	×	×	○	○	○
		Swarm chemistry	○	×	×	×	×	×	×	○	×	○	×	×	×	○	○	○
		SCL-model	○	×	○	×	○	○	○	×	×	×	○	×	×	○	○	○
		Squirm3	○	○	○	△	○	○	○	○	○	×	○	○	×	○	○	○
		Jonny Von	○	○	×	×	×	×	×	○	○	×	×	×	○	○	○	○
		Dissipative particle dynamics	○	×	×	×	○	○	○	×	×	×	×	×	×	○	○	○
		Langevin equation model	○	×	○	×	○	○	○	×	×	×	×	×	×	○	○	○
		Cellular Potts model	×	×	×	×	○	○	○	×	×	×	×	×	×	○	○	○
	2D/3D elastic	Avida	△	○	○	△	○	○	○	○	○	○	○	×	○	○	○	○
		Ameba	△	○	○	△	○	○	○	×	○	○	○	×	○	○	○	○
		SIVA	○	○	○	△	×	×	×	○	○	×	×	×	×	○	○	○
		2/3D Proto-cell model	○	×	○	×	○	○	○	×	×	△	×	×	×	○	○	○
	Nested multisets	ARMS/ACS	×	○	○	×	○	○	○	○	○	△	△	×	×	×	×	×

(continued)

Table 3.3 (continued)

			Emergence condition													Topological condition		
Symbol	Space	Example	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(1)	(2)	(3)
Combinatorial	Tank	AlChemy	×	△	○	△	×	×	○	×	○	×	×	×	×	×	×	×
		Assembler automaton	△	×	○	△	×	×	○	×	○	×	×	×	×	×	×	×
		Combinator model	○	△	○	△	×	×	○	×	○	×	×	×	×	×	×	×
		SAC	○	○	○	○	×	×	○	×	○	×	×	×	×	×	×	×
		Tominaga et al.	○	○	○	△	×	×	×	×	×	×	×	×	×	×	×	×
	2D rigid	The automaton	×	×	×	×	○	×	×	○	○	×	×	○	○	○	○	○
	Planar graphs elastic	Speroni et al.	○	△	○	△	○	○	×	×	○	×	×	×	○	○	○	○
	Graphs elastic	NAC	○	○	○	○	○	○	△	△	○	○	△	○	○	○	○	○

In the table, *circles* indicate that conditions are met; *crosses* indicate conditions are not met. *Triangles* correspond to evaluation results as follows: satisfied with aid from higher-level manager (Condition (1)); possible, in principle, but not observed to date (Conditions (2), (4), (7), (8), (10), (11)); translation schemes exist, but are fixed (Condition (4); variable in the long term, but not in the short term (Condition (6)); only active transport using energy indirectly is possible (Condition (10))

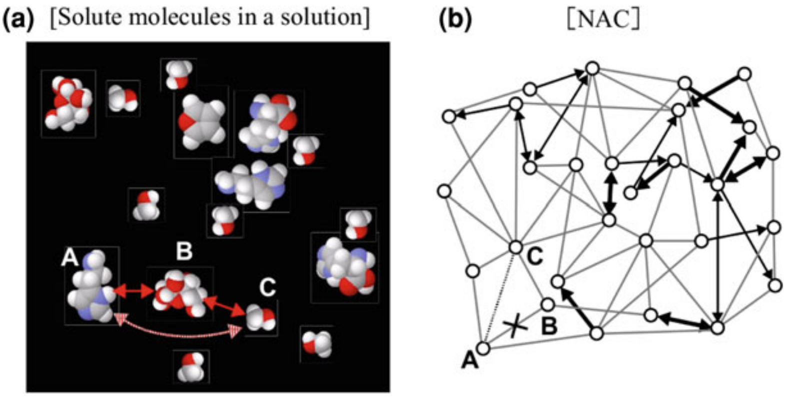


Fig. 3.17 Relationship diagram of solute molecules in solution and network artificial chemistry. **a** Solute molecules in a solution. The characteristics of the three-dimensional solution space restrict molecular motion. **b** Corresponding network artificial chemistry (NAC). The rewiring rules restrict node transport. After collisions between molecules A and B and between molecules B and C, the probability of a collision between molecules A and C is quite high

This poses a future challenge: to design systems meeting all of these conditions and featuring high emergent performance.

3.5 Network Artificial Chemistry

3.5.1 Basic Concept

Network artificial chemistry is the generic term for the artificial chemistry systems proposed and explored by Suzuki et al. [62–73] NAC systems express spatial relationships between symbols with purely topological graphs, making spatial descriptions more flexible. This section starts by describing some basic concepts of NAC.

The solute molecules, which play the central role in bio-chemical reactions, undergo repeated collisions when moving through the solvent (water), and their movements are restricted by the physical properties of the three-dimensional solution space and the size and shape of each molecule. NAC replaces these constraints with rewiring rules for the edges of a network. Figure 3.17 compares and draws parallels between this solution system and NAC.

The networks used in NAC are graphs consisting of nodes and edges. The nodes represent molecules or atomic clusters; the edges represent contacts or bonds. Since biological molecular interactions generally fall into one of four bonds—van der Waals, hydrogen, ionic, or covalent bonds—NAC has four edge types: wa, hy, io, and cv. The four edges are defined by different bond strengths. Weak edges are rewired one after another by a passive rule prepared in advance. Strong edges are formed by the active functions of the nodes and clusters.

Section 3.5.2 introduces typical examples of early NAC studies, while Sect. 3.5.3 introduces research on passive edge rewiring rules, which establish the basis for NAC dynamics. Section 3.5.4 describes the implementation of active programs into nodes to enhance the self-organization capabilities of the network structure.

3.5.2 Control Flow Cluster as Active Machine

Here, we introduce a model of the NAC in which one-dimensional node chains are converted to form clusters by "folding," functioning as active machines in a manner similar to proteins [68]. This mathematical folding and the folding of bio-protein molecules have the following common features.

Completeness of Expression

Both biological and mathematical foldings can generate clusters of various shapes based on differences in their one-dimensional sequences. The diverse range of protein functions and the diversity in the life forms indirectly demonstrate sufficient variations in the shapes of bio-proteins. When clusters are constructed according to the algorithm presented in this section, a node cluster (partial network) of any topology may also form, in principle, given the appropriate sequences.

Complementary Matching

Both are based on complementary matching of shapes or characters. Protein folding depends on the shapes, sizes, and physicochemical properties of the amino-acid residues making up the protein. (See Column 5 further below.) If a protein is folded with better matching between the residue sequences, the resulting protein is energetically stable. The folding of the node chain from Fig. 3.22b–c illustrated in this section can be regarded as a simplified mathematical counterpart of this process.

Column 5: Specificity and Complementarity of Chemical Reactions

In almost all of the reactions in molecular biology, from reactions of enzymes to protein self-assembly reactions, molecules specify their partners by complementary shape-matching (Fig. 3.18 in this column). Based on molecular shape, which vary as widely as human faces, molecules "choose" their partners in the reactions. This characteristic (called "specificity")

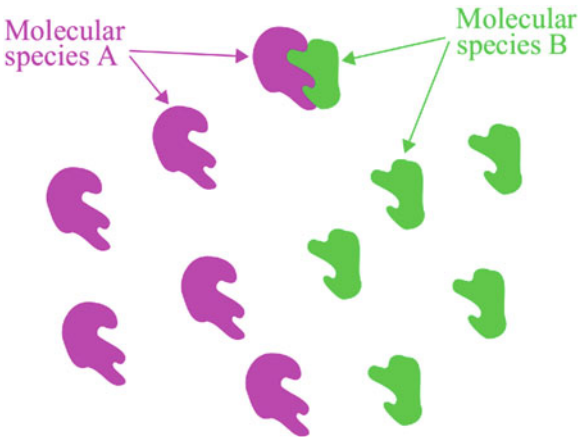


Fig. 3.18 Specificity based on complementary matching of shapes. The illustration schematically shows the shapes of molecular species A and B in a two-dimensional plane. In reactions between molecular species A and B, molecules of species A and molecules of species B have parts whose surface shapes complement each other. In this case, a molecule of species A will match with (react with) molecules of species B, but never with molecules of species A, its own species

propels one specific reaction in a timely manner in the presence of many other molecules within a cell.

From the viewpoint of intermolecular relation, complementary matching and equality matching (i.e., matching of like to like) have totally different outcome. If we observe only the matching relationship in specific areas on the surface of the molecules, interaction relationships characterized by only equality matching are transitive (i.e., if a ⇔ b and b ⇔ c, then a ⇔ c). As a result, the interacting molecules form clusters that tend to separate from the remaining molecules. With complementary matching, on the other hand, reactions do not occur among molecules of the same species; all reaction interactions involve different species. This creates the potential for more complex topologies in the interaction network.

Figure 3.19 in this column illustrates this situation with the most basic example of interactions among three molecules. In the case of equality matching (Fig. 3.19a), the molecules separate into two clusters. In the case of complementary matching (Fig. 3.19b), the graph is more complex—it is connected by links but not completely bonded.

In fact, the complementarity of the intermolecular interaction implicitly generates complementary characteristics at various levels of interactions in the biological sphere, including interactions between hosts and parasites, fertilization interactions within the same species, and interactions between different species, such as predation and symbiosis. To model such complementarity, some artificial life systems (e.g., [96]) implement complementary

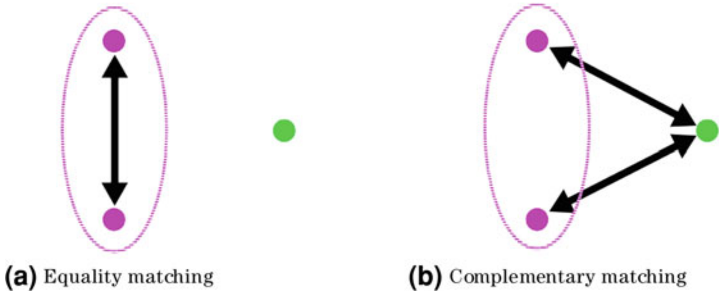


Fig. 3.19 Graphs of interactions among three molecules based on matching. **a** Equality matching; **b** complementary matching. The two nodes enclosed in the ellipse represent molecules of species A. The other node represents a molecule of species B. Two molecules connected with a *black two-headed arrow* interact (react) with each other; two molecules not connected with an *arrow* do not interact

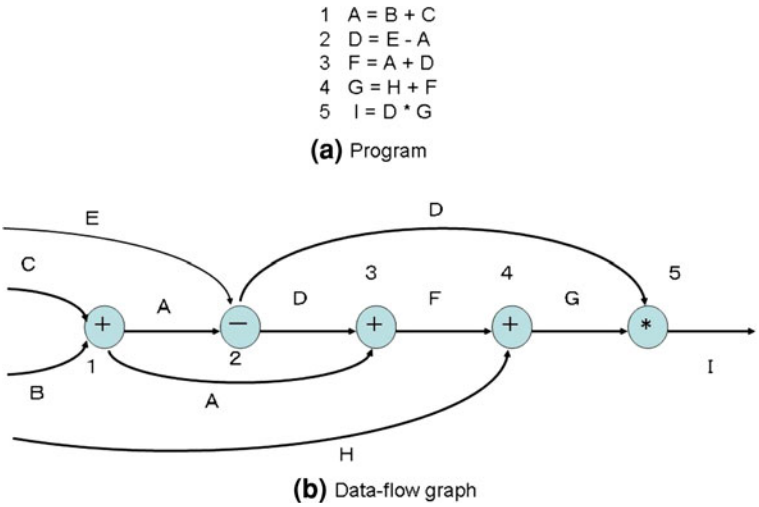
matching between character strings or bit sequences to select the target of an operation, thereby generating diverse interactions between artificial creatures.

The clusters created by this mathematical folding ultimately are bound with strong edges, and function as data-flow machines (DFMs) by operating in parallel while exchanging tokens along the edges. (See Column 6 further below.) In the following experiment, one-dimensional node chains undergo mathematical folding to form two types of clusters: splitase and replicase. We present how active processing by these machines, such as modification of nodes and rewiring of strong edges, separates hydrophilic and hydrophobic regions in the network and replicates a molecular chain. The folding proposed here, a type of conversion from a genotype (node chain) to a phenotype (DFM), presents a method for introducing a large functional machine into NAC.

Column 6: Data-Flow Machine

A DFM or a data-flow computer (DFC) [97–99] is a computational architecture intensively studied by researchers around the world from the 1970s to 1980s as a parallel processing computer system potentially capable of circumventing the limitations of von Neumann computers. A DFC expresses a program as a graph consisting of operation nodes and data edges linking the nodes (Fig. 3.20b in this column). Data are carried by tokens along the edges; each node fires on receiving a token and transmits the results of an operation as another token to the output edges. A DFC always performs the operation (node firing) when the data (token) arrives, creating a highly parallel and distributed architecture.

Data-flow computer architecture and software were first explored at universities and research institutions including MIT (USA), the University



```
1  A = B + C
2  D = E - A
3  F = A + D
4  G = H + F
5   I = D * G
```

(a) Program

(b) Data-flow graph

Fig. 3.20 Examples of **a** program and **b** data-flow graph used in data-flow computing (reproduced from [97])

of Utah (USA), the University of Manchester (UK), Toulouse University (France), NTT (Japan), and the Electrotechnical Laboratory (Japan). While several prototypes were built, the DFC architecture imposed unexpectedly severe constraints on software and prevented the sharing of existing software assets. Rapid advances in hardware technologies for von Neumann computers eclipsed the potential advantages of dedicated parallel machines, and DFC-based computers never reached the stage of commercial production. Interest in developing such machines dwindled in the 1990s.

Although DFCs failed to supplant von Neumann computers commercially, current superscalar processors inherit technologies from these research efforts, including out-of-order execution and register renaming [100].

Basic Model

The NAC graph used in this section consists of two types of nodes and three types of edges. Using an analogy drawn from living creatures, the nodes are classified as hydrophilic or hydrophobic nodes and the edges as Covalent (cv, directional, covalent bond), Hydrogen (hy, directional, hydrogen bond), or van der Waals (wa, non-directional, van der Waals bond) edges, in order of decreasing strength.

The wa edges of NAC are locally rewired by repeating the process sequences given in Steps (1)–(4) below.

(1) Reference Node A is randomly selected from the network.
(2) Node B with maximum degree is selected from adjacent nodes of A, connected by wa edges.
(3) Node C with minimum degree is selected from nodes connected by two or three wa edges from A.
(4) The hydrophilic and hydrophobic properties of Nodes A and C are investigated. If they satisfy specific conditions for bonding, Edge A–B is cut and Edge A–C created.

Here, the bonding conditions for Step (4) are set so that bonding is formed preferentially in the following order: hydrophilic–hydrophilic; hydrophobic–hydrophobic; and hydrophilic–hydrophobic [40]. Under his rewiring rule the graph tends to evolve toward having stronger small-world properties (see Column 3 discussed earlier) and more uniform degree distribution.

Mathematical Folding of Node Chains

The present NAC model expresses genetic information in terms of one-dimensional node chains connected by cv edges (Fig. 3.21). Each node has a character or character string composed of functional characters (a, b, …) or template characters (0, 2, …). The genetic information in the figure is the string obtained by joining the characters in sequence (e.g., a1122e0…). Each of the functional characters has a predefined function, but they do not operate in the form of the node chain.

Such node chains are folded after processing as shown in Fig. 3.22. First, the "agglomeration" process divides the adjacent node sequences directly before a functional character and converts each of the divided node chain sections into a

Fig. 3.21 Example of NAC node chain

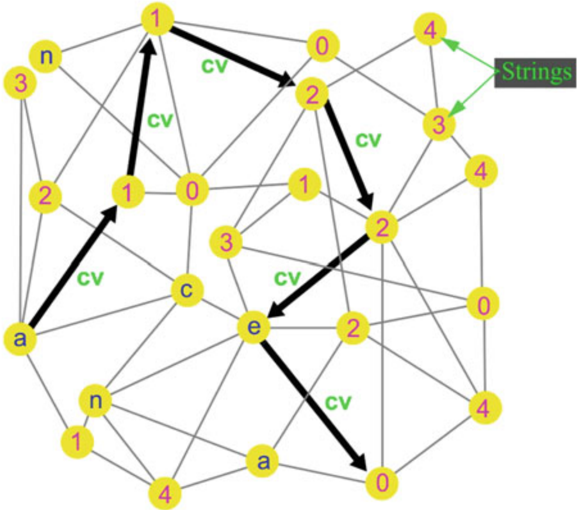
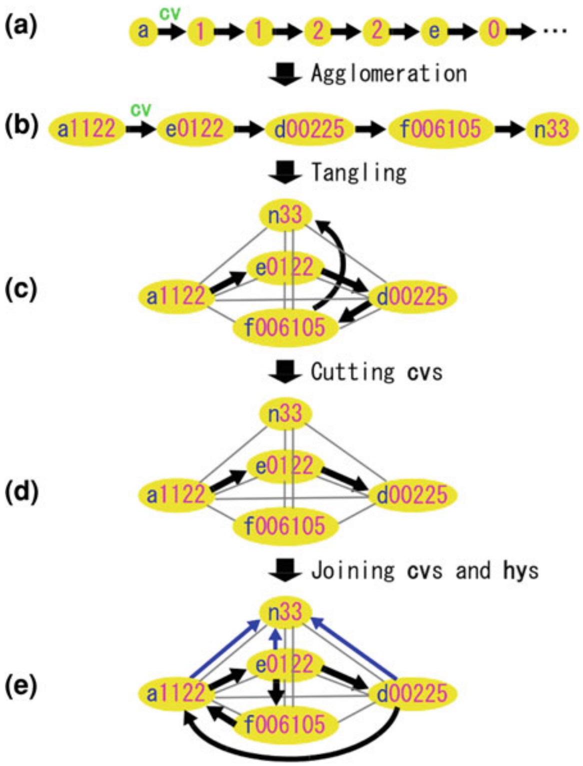


Fig. 3.22 Node chain folding process. **a** Initial node chain; **b** node chain after agglomeration; **c** cluster after tangling; **d** cluster after cutting cvs; **e** cluster after folding



single node with character strings such as a1122 or e0122 (Fig. 3.22b). Then, the wa edges link each of the node pairs in the node chain in (b), tangling the node chain (“tangling”; Fig. 3.22c). The template characters in each node are checked, and if “5” is included, the cv edge directly after is cut (cutting cvs; Fig. 3.22d).

Once the node chain is prepared for folding in this manner, the “folding” process (“folding,” Fig. 3.22e) creates the cv and hy edges, based on complementary matching between the template character strings. Here, the template characters, 0–3, are matched complementarily based on the relationships 0 ⇔ 1 and 2 ⇔ 3. Nodes with matched templates are newly linked to cv or hy edges. The directions of the edges go from a template starting with 0 to the template starting with 1 and from the template starting with 2 to the template starting with 3. In the example of Fig. 3.22, new cv edges are formed from 00 to 11 and 01 to 10, and new hy edges are created from 22 to 33.

Operation of Data-Flow Clusters

When the node chain is folded into a node cluster, it operates as a DFM functioning in parallel while transmitting tokens along the cv edges. In general, when a

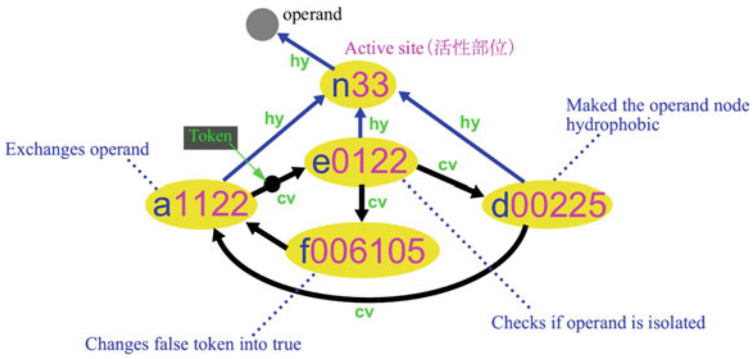


Fig. 3.23 Example of data-flow machine, "splitase." The cluster is made from the folding of a node chain consisting of 26 characters: "a1122e0122d00225f006105n33"

node receives a token, it fires if the value is "true," executing a routine assigned to the functional character (a, d, …). The value of the token is set according to the result and passed on to the next node.

Figure 3.23 shows an example of a DFM, called "*splitase*," generated by folding in the previous section. This machine consists of an active site (n33) and three operation nodes (a1122, e0122, d00225, and f006105). The operation nodes share a single active site via the hy edges and execute various processes to external nodes through this site. The processing starts with Node a1122 firing and the active site rewiring the hy edge to another operand node. Then, node e0122 fires. A check is performed to determine whether the new operand node is isolated with respect to cv. If so, the "true" token is transmitted from Node e0122 along the cv edges. If not, a "false" token is transmitted. When Node d00225 receives the true token, it operates on the operand node, changes its property to hydrophobic, and then returns the token to Node a1122. The function of Node f006105 is to fire when it receives a false token. If this node fires, the token value is converted to "true" and transmitted to Node a1122. This data-flow cluster has the capability of changing the operand nodes adjacent to Active site n33 to hydrophobic, one after another. This process rapidly produces one oily molecule after another.

Figure 3.24 shows a more complex example of designing the "*replicase*" data-flow cluster. The cluster consists of three active site nodes and 16 operation nodes. The basic replication processing of the node chain is based on processing proposed by Hutton [22]. The cv loop composed of Nodes 11–17 on the right half of this machine plays the main role. With the nodes in this part operating one after another by transmitting tokens, the active site randomly creates hy edges to nearby nodes, compares the character strings with the chain node, creates cv edges if they are the same, and moves one step along the chain. When the replication of the node chain is completed, the active site moves away from the node chain (cuts the hy edge) and searches for another node chain.

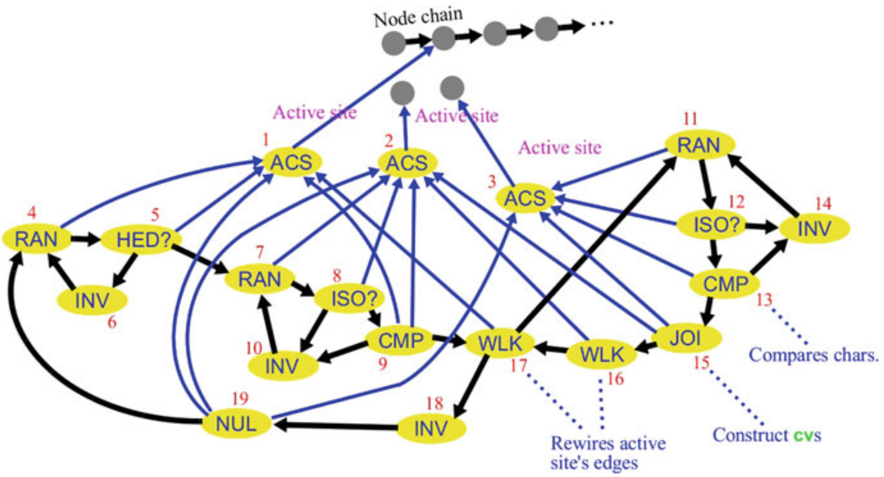


Fig. 3.24 Design example of data-flow machine, "replicase." The cluster is made from the folding of a node chain consisting of 191 characters: "a1101122311000k00011223f001005a11100611001232e01000232g22300101232f001106101115a11110222e00000222g22201001223f11116000015h101102326222i232i1101022300001f00010j22262326223001115n3325n3235n333" Mnemonics consisting of three to four characters in the node help identify the functions of the functional characters

Experimental Results

Network separation. We perform a network separation experiment using the *splitase* shown in Fig. 3.23. A node chain consisting of 26 nodes is introduced into an initial random network. After agglomeration, tangling, and cv cutting, the following three processes are performed: (1) rewiring of passive wa edges; (2) folding by wa edge connection; and (3) active processing by token transmission. Repeating these three processes changes the network topology and causes its structure to self-organize. Figure 3.25 shows a typical result of this experiment. As the hydrophobic nodes are generated by the *splitase*, the graph is separated into hydrophilic and hydrophobic regions and generates a structure with the hydrophilic cluster enclosed in the hydrophobic network.

Replication of Node Chain

A *replicase* node chain consisting of 191 nodes and an unfolded operand node chain consisting of ten nodes are introduced into an initial random network and processed in a manner similar to the above. Figure 3.26 shows a typical result. In this example, the *replicase* cluster creates replicated node chains with the same character strings as the operand node chain in the network.

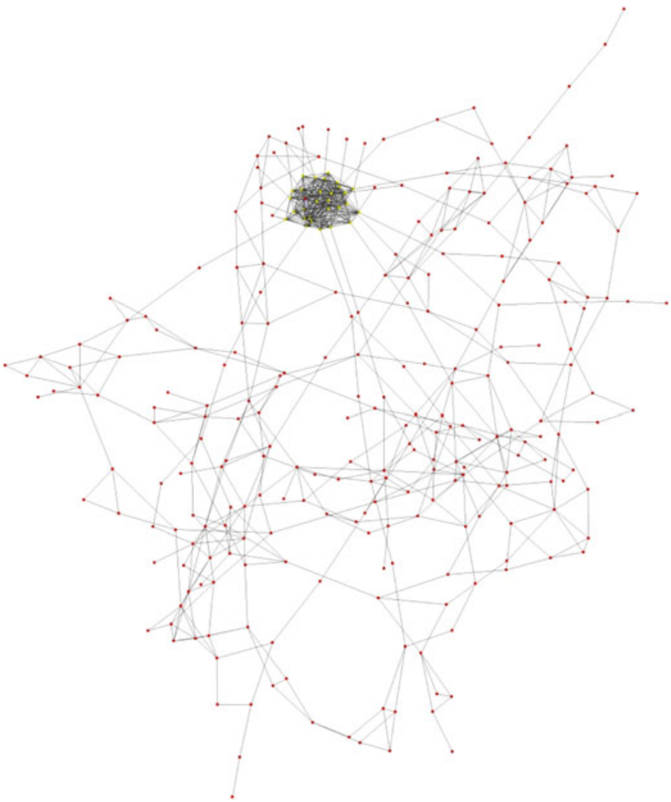


Fig. 3.25 Example of separation of network into hydrophilic and hydrophobic regions. The small cluster in the upper section is the hydrophilic cluster, and the rest is hydrophobic. A splitase chain is placed in a network with number of nodes $N = 300$ and average degree $W_p = 5$. After initial processing, the process is repeated $t = 260$ times. The wa edge joining probability is set as follows: $P_{ww} = 1.0$ between the hydrophilic–hydrophilic pair, $P_{ww} = 0.3$ between the hydrophobic–hydrophobic pair, and $P_{ww} = 0.0$ between the hydrophilic–hydrophobic pair. Since the present model has no mechanism to halt the splitase, if we continue the repetitive processing, the hydrophilic region other than the cluster itself disappears after 40 iterations. For all the figures shown hereafter in this chapter commercially available software aiSee [5] is used for the visualization (planar drawing) of networks

3.5.3 Passive Rewiring Rule Based on Energy Minimization

Network artificial chemistry generally provides four edge types: wa (van der Waals), hy (hydrogen), io (ionic), and cv (covalent). Of these, the passive rewiring rule of the weakest wa edges forms the basis of the dynamics. As shown in Fig. 3.6a, the intermolecular distance at which van der Waals attraction exerts its influence approximately equals the distance between molecules in the encounter state. (This distance is defined in this chapter as no greater than twice the hard-sphere diameter.) This means the passive rewiring rule for wa edges in NAC must be designed so that

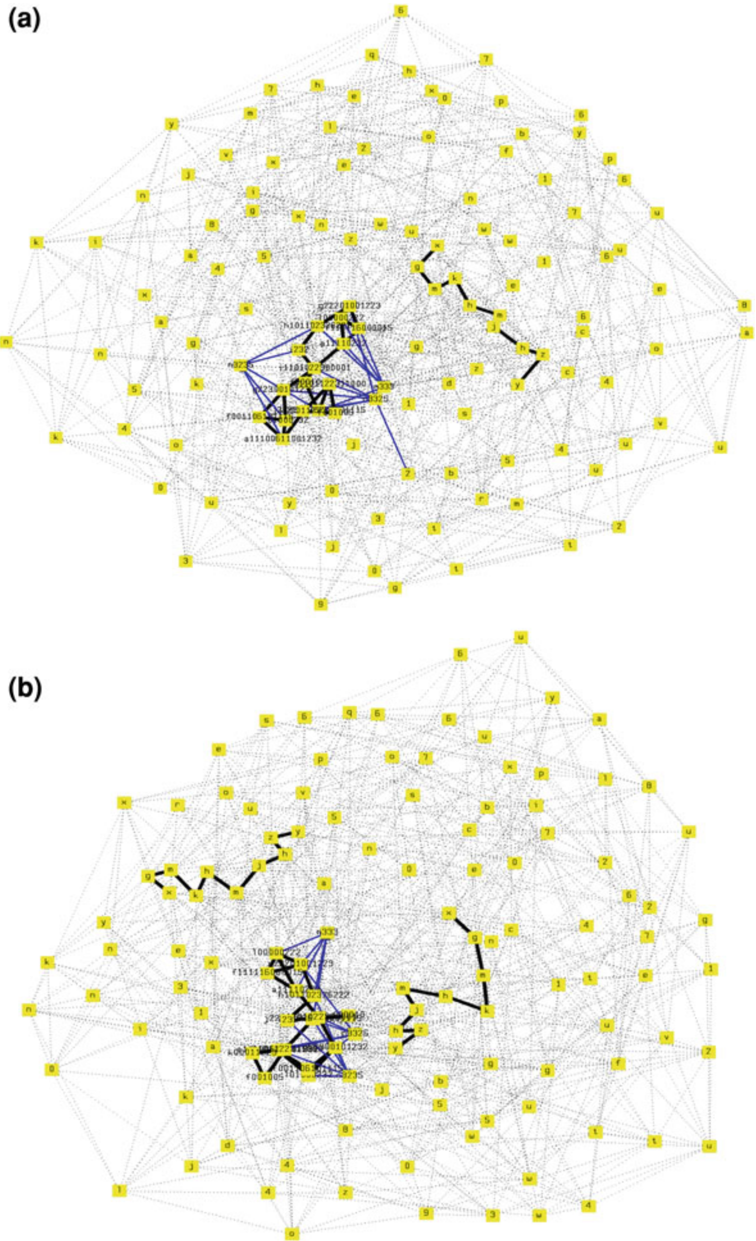


Fig. 3.26 Example of node chain replication by replicase. **a** Snapshot after 50 iterations; **b** snapshot after 710 iterations. The number of nodes is $N = 300$, the average degree is $W_p = 5$, and the wa edge joining probability is $P_{ww} = 1.0$

the topological properties of the resulting network are at least approximately equal to that of the encounter network of molecules discussed in Sect. 3.3.4.

Here, we describe an attempt to design the rewiring rule for weak edges [69]. The goal is to implement the feature of the design conditions previously mentioned. In order to formulate the rewiring rule, we first introduce energy defined for the entire graph. We then formulate an algorithm to minimize the energy stochastically by the Metropolis method and allow edges to be created or deleted as determined by the algorithm. Under this rule, we perform a NAC rewiring simulation, measure the topological properties using the graph obtained as described in Sect. 3.3.4, and try to determine whether they agree with the properties of the molecular network. The results are summarized below.

Topological Properties (1) and (2): The C and L values are intermediate between those for a regular graph and a small-world graph.

Topological Property (3): The resulting graph projected onto a plane forms a unity without distortions.

Topological Property (4): The resulting graph projected onto a plane has few edge crossings and small tangledness.

Topological Properties (5)–(7): The parameter dependence is qualitatively consistent with that of the hard-sphere simulation.

NAC Rewiring Simulation

Since the specific formulation of the rewiring rule used in this experiment is highly mathematical, it is summarized later in Column 7 and not discussed in depth in the main text. Based on preliminary experiments, we set the parameters defined in the formulation as: mean free path: $\lambda = 2$; upper limit for the shortest path length: $L_{\mathrm{max}} = 15$; indices in the equation for spatial restriction energy: $\mu = 0.01$, $\sigma = 4$, $v = 0.02$, $\gamma = 0.1$, and $\alpha = 2$; average degree: $\bar{k} = 4$; the threshold for energy change in the modified Metropolis method: $dE_{\mathrm{th}} = 0.0015$; the acceptance probability for $-dE_{\mathrm{th}} \leq \Delta E < 0$: $\kappa = 0.2$; and the temperature factor $\beta = 1{,}500$. Since this section focuses on spatial restriction energy, we use only wa edges and set the bonding energy at $\varepsilon^{(\mathrm{wa})} = 0$.

Experimental Results

Starting from an initial random graph with number of nodes $N = 200$ and average degree $\bar{k} = 4$, we rewire the NAC graph. Figure 3.27 plots the cluster coefficient (C) and the average path length (L) as functions of time. As the figure shows, after approximately 100 thousand iterations, the C value of the NAC graph is roughly 19.4 times the initial value ($C \approx 0.018$) for the random graph. The L value is roughly 1.5 times the initial value ($L \approx 3.95$) for the random graph. The values indicate the resulting graph has properties approaching those of the encounter network of hard spheres.

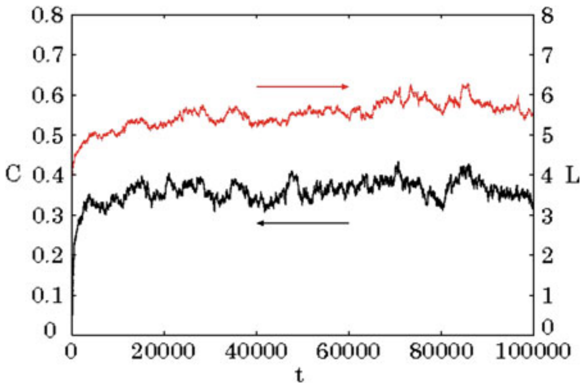


Fig. 3.27 Cluster coefficient (C) and average path length (L) in NAC rewiring simulation. The time propagation is measured for 100 thousand iterations of graph rewiring

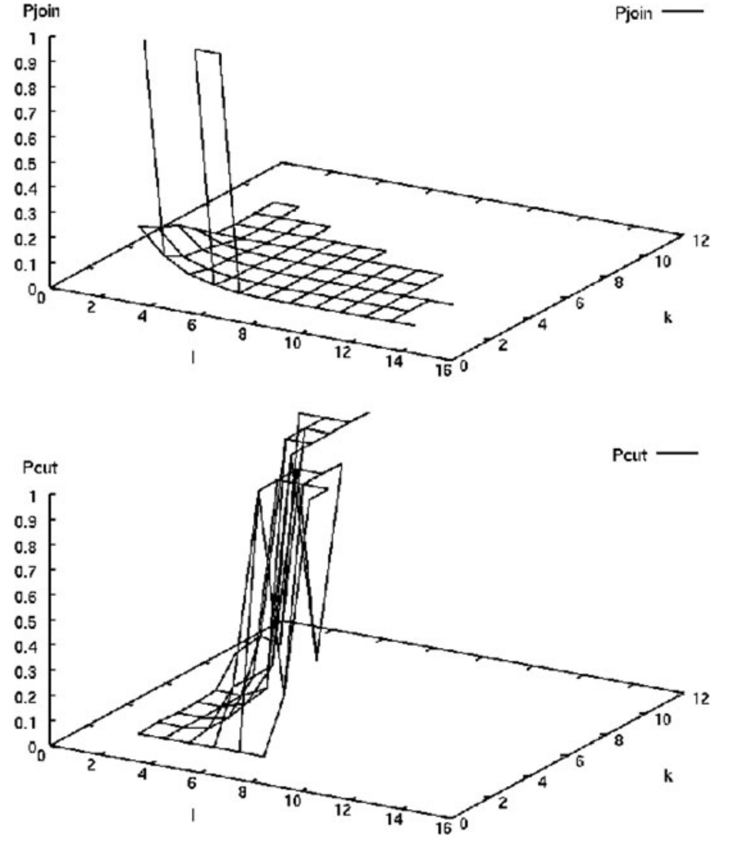


Fig. 3.28 Edge joining and cutting probabilities calculated based on NAC rewiring simulation result. We apply Eq. 3.19 from Column 7 (discussed earlier) to convert the average value for ΔE calculated for 200 thousand iterations of rewiring to P. As in Fig. 3.15, P_{join} is plotted as a function of the shortest path length (l) between pairs and the degree (k) of one of the end points; P_{cut} is plotted as a function of second shortest path length (l_2) between the end points of an edge and degree (k) of one of the end points

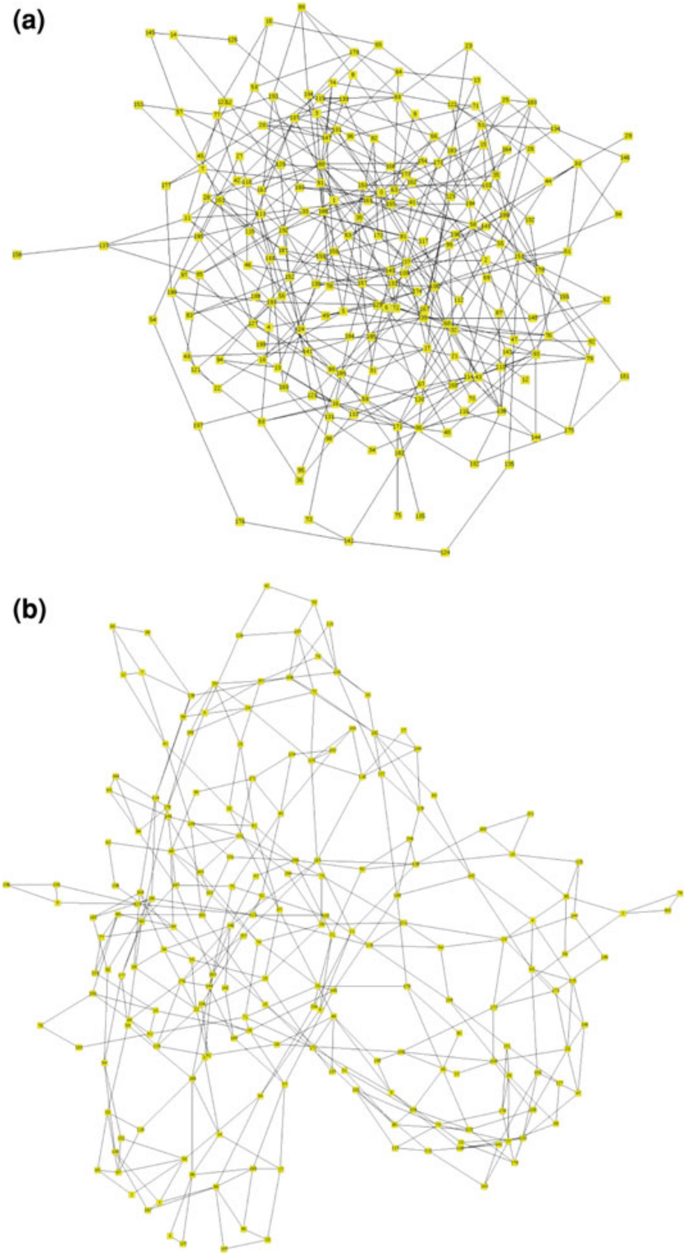


Fig. 3.29 Two-dimensional rendering of graphs created in NAC rewiring simulation. **a** Initial random graph; **b** NAC graph after 80 thousand iterations of rewiring visualized with aiSee [5]

Figure 3.28 shows the expectation values of the probabilities P_{join} and P_{cut} measured in the simulation. They agree qualitatively with the results for the random-walk simulation (Fig. 3.15).

Figure 3.29 shows the initial NAC graph and the NAC graph after 80 thousand iterations projected onto a plane. The graph after rewiring (Fig. 3.29b) is bunched and contains fewer edge crossings, generating a topology easier to project onto a plane.

Column 7: Formulation of Energy-Based NAC Edge Rewiring

Network artificial chemistry expresses molecular collisions as the generation of edges. Determining what molecule a given molecule will collide with next is equivalent to the problem of determining between what node pairs an edge should be generated. Here, we assume that the intermolecular distance in a NAC is measured by the shortest path length between the node pair and that the node pair for which the new edge generation is attempted is selected in proportion to the free path function below.

$$P_{\mathrm{fp}}(l) = \exp(-l/\lambda) \tag{3.13}$$

This equation is based on Topological Property (6) of the molecular network in Sect. 3.3.4. Here, the parameter λ is called the NAC mean free path and corresponds to the mean free path of the molecules in the statistical physics (average distance traveled by a molecule without colliding with another molecule). As λ in the physics is independent of temperature (namely, the speed of the molecules) and depends only on the size and density of the molecules, so λ in a NAC takes a constant value independent of temperature. Here we also set the restriction $l \leq L_{\mathrm{max}}$, to avoid occasional situations where the system searches the graph for far-distant node pairs, resulting in the driving up of the computational cost.

We define the energy of the NAC graph as the sum of the topology-related spatial restriction energy E_{s} and edge bonding energy E_{b}, as:

$$E = E_{\mathrm{s}} + E_{\mathrm{b}} \tag{3.14}$$

E_{s}, the sum of the term (μ-term) for suppressing fluctuations in the degree of each node and the term (ν-term) for providing regularity, is expressed as:

$$E_{\mathrm{s}} = \frac{\mu}{N}\sum_{n}\left|k_n - \bar{k}_n\right|^{\sigma} + \frac{\nu}{N}\sum_{\langle mn\rangle}\left\{\left(\frac{k_m k_n}{\bar{k}^2}\right)^{\gamma}(l_2)_{mn}\right\}^{\alpha} \tag{3.15}$$

In the expression, Σ_n is the summation over all nodes, each denoted by n, while $\Sigma_{\langle mn\rangle}$ expresses the summation over all adjacent node pairs, each denoted by mn. Here, k_n is the degree of Node n, $\bar{k}_n$ the expectation value of the degree of Node n (which differs for each node), $\bar{k}$the average degree for all nodes, and $(l_2)_{mn}$ the second shortest path length for Node Pair mn. When the constant σ is positive, the μ-term has the function of making the node degree closer to the target value, $\bar{k}_n$, (in the case of hard spheres of uniform size distributed in space, a single hard sphere cannot randomly come into contact with an unlimited number of hard spheres), and the ν-term has the function of cutting the edges between nodes with large degrees on both ends and a large second shortest path length with high probability.

E_b, the edge bonding energy, is expressed as:

$$E_\mathrm{b} = -\frac{1}{N}\sum_{\langle mn\rangle} e_{mn} \tag{3.16}$$

Here, e_{mn}, the bonding energy for Edge mn (the energy required to cut the edge), is defined by the following, according to the edge type: $(0 \le \varepsilon^{(\mathrm{wa})} \le \varepsilon^{(\mathrm{io})} \le \varepsilon^{(\mathrm{cv})})$.

$$e_{mn} = \begin{cases} \varepsilon^{(\mathrm{cv})} & \text{when Edge } mn \text{ is a cv edge} \\ \varepsilon^{(\mathrm{io})} & \text{when Edge } mn \text{ is an io edge} \\ \varepsilon^{(\mathrm{wa})} & \text{when Edge } mn \text{ is a wa edge} \end{cases} \tag{3.17}$$

The rule for joining or cutting the edges using the formulated E is determined by the modified Metropolis method to stochastically minimize E. More strictly speaking, edge rewiring is done by executing the two following steps at each iteration.

Step 1: Edge joining

Select a random node in the graph. Select randomly another node with a probability proportional to $P_\mathrm{fp}(l)$ where l is the path length between the two nodes. Based on network energy change ΔE before and after the new edge is generated between the two nodes, calculate acceptance probability $P(\Delta E)$, and then join the edge at this probability.

Step 2: Edge cutting

Select a random edge in the graph. Calculate network energy change ΔE before and after the edge is cut. Based on the energy change, calculate acceptance probability $P(\Delta E)$, and cut the edges at this probability.

The energy change $\Delta E = \Delta E_\mathrm{s} + \Delta E_\mathrm{b}$ associated with edge joining or cutting is obtained from Eqs. 3.15–3.17 and Eq. 3.18 below. (Here, $\varepsilon^{(\text{cv-act})}$ is the activation energy of a cv edge.)

$$\Delta E_\mathrm{b} = \begin{cases} \frac{1}{N}\varepsilon^{(\mathrm{cv-act})} & \text{when joining a cv edge} \\ \frac{1}{N}\left(\varepsilon^{(\mathrm{cv-act})} + e_{mn}\right) & \text{when cutting a cv edge} \\ -\frac{1}{N}e_{mn} & \text{when joining an edge other than a cv edge} \\ \frac{1}{N}e_{mn} & \text{when cutting an edge other than a cv edge} \end{cases} \tag{3.18}$$

Accurately evaluating the change in the ν-term in Eq. 3.15 requires calculations over all edges in the graph. Here, we consider only the partial

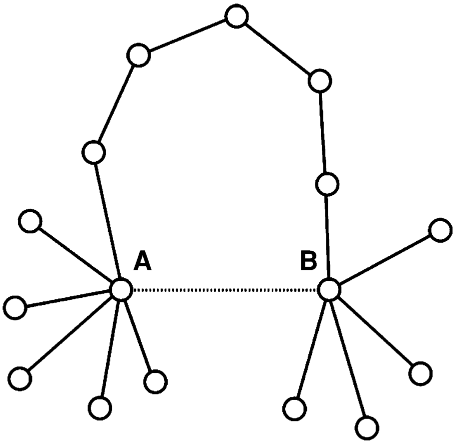


Fig. 3.30 Nearby section graph for approximate calculation of change in the spatial restriction energy term. The graph consists of edges directly connected to Nodes A and B, the end points of the edge about to be joined or cut, and edges constituting the shortest path (when joining) or the second shortest path (when cutting) between Nodes A and B

graphs shown in Fig. 3.30 in this column to evaluate the summation $\Sigma_{\langle mn \rangle}$. The results of a preliminary NAC rewiring experiment confirm that the energy difference ΔE, calculated as such, provides a satisfactory approximation of the true value. To reduce the computational cost, we also restrict the value of l_2 to be $l_2 \leq L_{\max}$.

Based on the above formulation, we calculate the final $P(\Delta E)$ from the following equation, which slightly modifies the acceptance probability of the Metropolis method.

$$P(\Delta E) = \begin{cases} \kappa \exp(-\beta \Delta E) & \text{when } \Delta E \geq 0 \\ \kappa & \text{when } -dE_{th} \leq \Delta E < 0 \\ 1 & \text{when } \Delta E < -dE_{th} \end{cases} \tag{3.19}$$

We would expect that setting probability $\kappa < 1$ for $\Delta E \geq -dE_{\text{th}}$ at a value different from the probability for $\Delta E < -dE_{\text{th}}$ would approximately reproduce the rapid rise in probability, as observed in Fig. 3.15b of the main text.

3.5.4 Organization of Network Structure by Active Node Program

The rewiring rule based on the network energy discussed in Sect. 3.5.3 produces qualitatively the same properties in the NAC graph as the contact graph of hard spheres. Nevertheless, when we view the network generated in this way as a virtual space replacing the three-dimensional solvent space, the results remain unsatisfactory. One reason for this is that this network fails to model liquid crystallization.

For example, water as a material at room temperature forms a network in which water molecules are loosely bonded by hydrogen bonds. At temperatures below 0°C, this network becomes regular, forming a crystal structure (ice). In an extreme sense, this phenomenon is generated by the physical properties of the water molecules and three-dimensional spatial restrictions. However, it is difficult to create such a regular structure in a graph without spatial information such as angles and positions, based only on the energy minimization principle. The cluster coefficient and average path length of the graph obtained in the numerical simulation of Sect. 3.3 are also limited by the values of the small-world network (see Column 3, discussed earlier) or slightly closer to those of a regular network. Simulations with various energy functions that use a lower temperature value (higher value of the temperature factor β of the Metropolis method) cannot prompt the graph to converge into one with a large average path length (L)—in short, one with high regularity—while maintaining the connectivity of the overall graph [66, 69].

One of the reasons for this difficulty may be the specialty of the water molecules, ignored in the past NAC solvent node. Unlike organic solvents such as benzene, water is an abnormal liquid that establishes a pseudo-regular lattice over the space that it occupies based on hydrogen bonds. This nature is also the main cause of other behavior, including hydrophobic interactions and lipid bilayer formation. (Hydrophobic interactions are generally described as a phenomenon in which non-polar molecules cannot join the regular network of the water molecules, but are instead repelled and thereby segregated.) Intermolecular interactions based on hydrophilic and hydrophobic properties constitute one of the basic dynamics of bio-molecules. Thus, ultimately, water is capable of nurturing life because water molecules are not inert, but active molecules with unique functions. Up to this point—as in Sect. 3.5.2, for example—we have assumed that the active functions in a NAC result from the assembly of many nodes—such as a data-flow cluster (corresponding to a protein)—and that the solvent nodes (corresponding to water) are static and lack specific functions. However, the preceding discussion indicates that a graph, a pure mathematical model, has certain limitations in producing the regularity (crystal structures) created by spatial restrictions. In order to circumvent these difficulties, we need to implement more active functions in the nodes and allow them to interact.

Based on this concept, here we describe a new simulation model of the NAC which attributes programs (active functions) to NAC nodes and executes them in parallel [70]. In the solvent nodes, assumed to be inert in the NAC studies up to this point, we implement a program designed to model the complex functions of water. The program rewires the surrounding edges based only on local information, generating pseudo-regular structures in the network such as sheets and strings.

This example allows a network, a purely mathematical model, to self-organize an emergent structure similar to that from physical restrictions. The result suggests that the simplest and most efficient way to induce self-organizing structures in a graph is to attribute functions of sufficient complexity to the nodes and allow them to interact vigorously. Out of many possible ways to describe such functions, this simulation takes a method to implement functions as programs, allowing an artificial local program to generate global structures in a network, in a manner similar to amorphous computing (see Column 8).

Column 8: Amorphous Computing

The word *amorphous* means something lacking a defined shape or integrity. As used here, the term *amorphous* has a slightly different meaning: We can better understand the term *amorphous computing* as the formation of patterns from a bottom-up approach. This technique has been studied since the late 1990s, mainly by researchers at MIT, including Knight and Weiss [101].

These researchers pursued the following line of thought. When microfabrication technologies reach the point at which line widths in the integrated circuits formed on a silicon wafer reach the nanoscale order, design needs may outstrip current technologies, which assume that humans design patterns and impose them, top-down, on the wafer. At this point, perhaps the solution would be bottom-up patterning. Perhaps patterns might form from functional particles scattered on the silicon wafer—a wondrous result, if true.

Although the concept of such particles was merely dreamed up by computer scientists, they proceeded by positing that the problem, the fabrication of functional particles, had been solved. Based on this assumption, they proceeded to address an interesting informatics problem: What particle program should they prepare to form the patterns desired?

Here, let us consider a well-known example: PL-Gakuen's human letter displays at the summer Koshien baseball games. Each member of the cheerleading battalion holds several color cards, all numbered. When the captain of the cheerleading team calls out a number, each team member holds up the card corresponding to that number. This is an example of a top-down process; the resulting patterns are based on instructions issued by the captain. Let us take a slightly different approach and form the patterns with a bottom-up approach. Now, each member holds a program that specifies when and how the cards should be held up, and the team captain issues no instructions. Each member must determine the card to be held up next based only on local information—for example, the cards that the member or member's neighbors are currently holding up. What program do we need to prepare for each member to create the desired patterns? This, in fact, is a profound problem in informatics which information scientists today still have no generic prescription to solve with.

The informatics problem addressed in amorphous computing by Knight et al. entails the design of a program under such conditions. Afterward, the researchers involved in this study shifted the direction of research to an approach that applied engineering methods to real living cells (such as yeast) to equip cells with specific functions [102]. Living creatures are the only known systems that take an amorphous approach—behaving according to genomes, not instructors—and function in an orderly fashion as a whole. Perhaps life can inspire a successful approach to amorphous programing. This is one of the hottest research areas in the bioengineering. The fabrication of functional particles is also currently a topic of intense interest in nanotechnology [103, 104].

Model

Here, we consider the NAC solvent nodes as active nodes and design the node program. The design is based on the complex functions of water molecules. We run the program to investigate how structures form in the network. As a target structure, we consider the two-dimensional square lattice.

We implement the experimental program in Java, where each node or edge is expressed as an instance of the node class or edge class in Java. These instances are linked to each other with instance variables. The node program is implemented as a method of the node class. The node instance has a pointer array hy [ ] of size 4 and can indicate up to four hy edges connected to other nodes.

Figure 3.31 shows the algorithm of the node program used to implement rewiring. The method ad_rand() selects an adjacent node via an hy edge; the

Fig. 3.31 The node program algorithm

```
public void conduct_nd_prog()
{
  // choosing pointer's displacement
  int dp = [either -1 or 1];

  // choosing this node's initial
  //   adjacent node randomly
  Node nda = ad_rand();

  // choosing the next node
  Node ndb = nda.ad_next(this,dp);
  if(ndb==null) return;

  // choosing the second next node
  Node ndc = ndb.ad(nda,dp);

  // choosing this node's another
  //   adjacent node
  Node nde = ad_next(nda,-dp);
  if(ndc==null OR ndc==this){
    if(nde==null) return;
    if(nde==ndb) return;

    remake_edge(ndb,nde);
    return;
  }

  // choosing the third next node
  Node ndd = ndc.ad_next(ndb,dp);
  if(ndd==this) return;

  remake_edge(this,ndc);
}
```

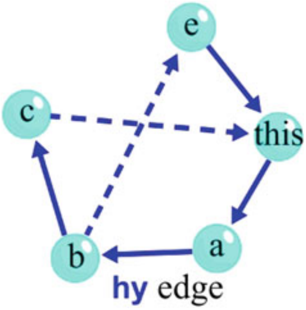


Fig. 3.32 Schematic diagram of the algorithm from Fig. 3.31. By creating the edge indicated by the *broken line*, the algorithm can create a tetragonal loop

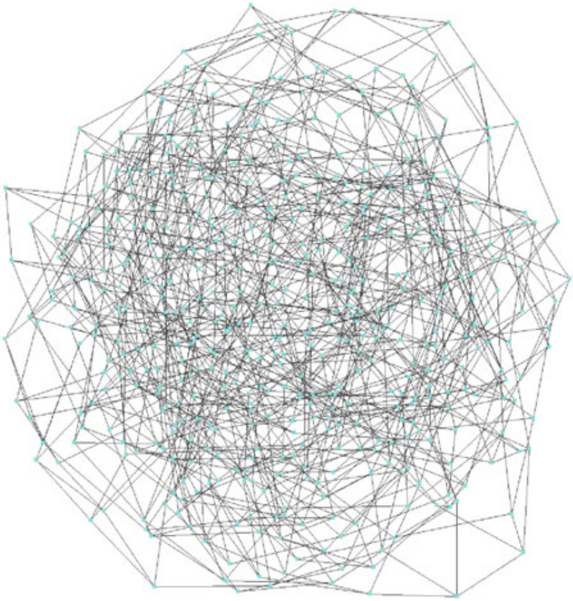


Fig. 3.33 Two-dimensional rendering of initial random graph based on the number of nodes $N = 512$ and uniform degree $K = 4$ (the cluster coefficient is $C \approx 0.0020$; the average path length is $L \approx 5.0$)

method ad_next(nd1, dp) returns an adjacent node via hy[p + dp] if nd1 is adjacent via hy[p]; the method remake_edge(nd0, nd1) generates a new hy edge between the two nodes nd0 and nd1 with a pointer; the variable dp = ±1 specifies the direction of the path search. Two adjacent hy edges in a loop are registered to locations that differ by dp with hy[ ] of the node between them. The Java operator "." represents a class member—for example, nda.xx specifies member xx belonging to nda.

Figure 3.32 is a schematic diagram illustrating the functions of the algorithm. This algorithm first extracts the path, nde–this–nda–ndb–ndc–ndd, with reference to the present node (this); it then creates the edge, ndb–nde or ndc–this, forming a loop of length 4.*Experimental results*. We start from a random graph with number of nodes $N = 512$ and uniform degree $K = 4$ and run conduct_nd_prog() K times in each iteration in each node. Figures 3.33, 3.34, and 3.35 show typical results.

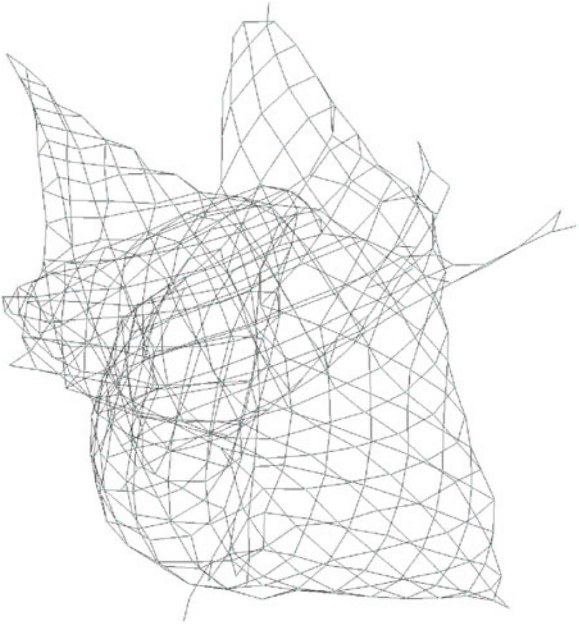


Fig. 3.34 NAC graph after 100 iterations (the average degree is $K \approx 3.81$, the cluster coefficient is $C \approx 0.0$, and the average path length is $L \approx 13.5$. Approximately 87% of the nodes have degree $K = 4$)

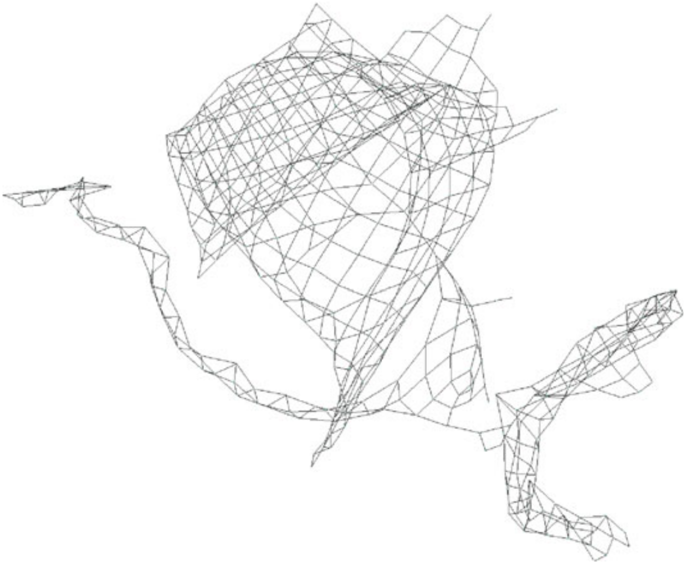


Fig. 3.35 NAC graph after 1,000 iterations (The average degree is $K \approx 3.84$, the cluster coefficient is $C \approx 0.0$, and the average path length is $L \approx 18.0$. Approximately 90% of the nodes have degree $K = 4$)

As shown in these figures, the rewiring by conduct_nd_prog() gradually creates regularity within the graph. After approximately 100 iterations, the graph assumes the structure of a folded two-dimensional square lattice sheet (Fig. 3.34). Beyond this stage, the sheet is gradually unfolded while maintaining regularity. After approximately 1,000 iterations, the graph assumes a structure with sheets and strings combined (Fig. 3.35). The algorithm conduct_nd_prog() specifies only the local links of the graph, not the overall structure. It does not distinguish between two-dimensional sheets and one-dimensional strings as the overall structure. (Several experimental results suggest that the current algorithm tends to converge the graph to a string rather than a sheet.)

3.6 Modified Network Artificial Chemistry

3.6.1 Concept

In the last part of the NAC studies described in the previous section, we implement a functional program at each node and demonstrate that the graph structure can be organized emulating the physical world. The program (Figures 3.31 and 3.32) used is implemented at each node instance as a class method. Interactions between nodes are implemented with reference to the nearby nodes ndc and ndd at a distance of 3–4. However, this program is not perfect with respect to the localization of processing. As described in Sects. 3.3 and 3.5.3, the intermolecular interactions in a solution work between encountering molecules—in other words, between a pair of nodes connected by an edge in NAC. A program that remotely operates the third- or fourth-neighbor nodes would violate this restriction.

In this section, we describe a study starting from a token [72] introduced into the network to resolve this problem. The term *token* refers to an information unit that transfers data and starts node operations in a DFC (see Column 6 discussed earlier). In this section, we add other functions to this token—which we will refer to as an *agent* hereafter—using it as an information carrier that transfers not just a "value," but a "program." As in the previous section, molecular functions are expressed in programs. However, these programs are carried by agents; they are not resident in each node. A program executes (firing or reaction) when it reaches a node, causing the rewiring of edges or the creation or deletion of agents.

If we modify the NAC in this way, how should we assign the correspondence between the nodes and agents and elements of the biological system? Through Sect. 3.5, we have assumed that a NAC node represents a molecule or group of atoms and that edges express the bonding or contact between the molecules. If the main body of the operation is not the nodes but the agents, if the nodes provide only a place for the agents to execute programs, perhaps agents—not nodes—would better represent molecules. Perhaps nodes are better interpreted as sites

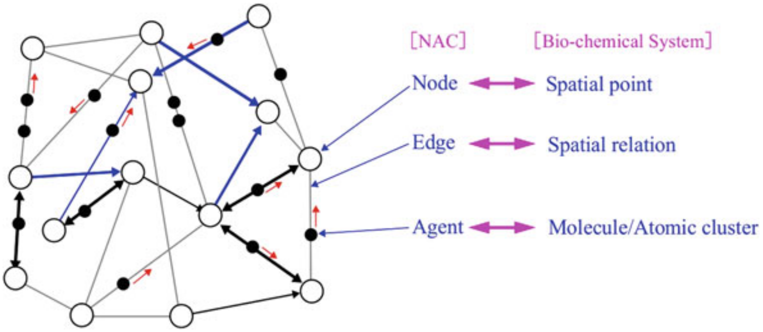


Fig. 3.36 Relationship diagram of modified network artificial chemistry (MNAC) and bio-chemical reaction system. (Contact and bonding relationship between molecular groups belonging to a spatial point.) A molecule or an atomic cluster is expressed as an agent moving in the network. A node represents a minute spatial region within which the molecular agent is located at a given point in time, and the edges collectively represent the contact or bonding relationship between the molecules or atomic clusters belonging to the spatial region

where molecular agents act. We will call an artificial chemistry model based on such a correspondence a Modified NAC (MNAC) to distinguish it from a conventional NAC. Figure 3.36 summarizes the correspondence between a graph in MNAC and a bio-chemical reaction system.

Moreover, if we regard the nodes as CPUs, edges as communication lines, and agents as programs, MNAC becomes a computational model that could be called "program-flow computing," in which many CPUs connected by communication channels execute numerous diverse programs in parallel. We discuss the significance and future research agenda of this model in Sect. 3.7.

In Sect. 3.6.2, as a typical example of MNAC studies, we describe the simulation of formation and splitting of a hydrophilic cluster using molecular agents.

3.6.2 Formation and Splitting of Hydrophilic Cluster by Molecular Agents

Here, we describe an example experiment in which the network structure is organized within the MNAC framework by the functions of the molecular agents [71, 73]. The experiment uses three types of designed agent programs (one each for rewiring the wa edges, rewiring the hy edges, and splitting the cluster). The programs are written as a sequence consisting of 45 machine instructions defined in advance. In the experiment, the programs fire and execute when the agents with the programs arrive at the nodes, resulting in the emergence of a hydrophilic cluster structure containing a pseudo-lattice structure in the graph, with a "centrosome" finally splitting the structure.

If we apply the scheme of artificial chemistry, the five elements stated in Fig. 3.2, to MNAC, the space is expressed by the network, the symbol is expressed by the agent with a program, the rules for reaction of symbols are expressed by the computational functions of the program, and the rules for transport of symbols are expressed by the rule for restricting the movement of the agents. While no higher-level manager is in the picture at this point, we are likely to need one if we seek to develop MNAC into an evolving system. Based on this correspondence, MNAC is evaluated with reference to the emergence conditions (Sect. 3.2.2), topological conditions (Sect. 3.3.2), and topological properties (Sect. 3.3.4) discussed earlier. The results are summarized below. Here, we omit Topological Properties (5)–(7), since the edges in this MNAC do not represent the encounter and bonding relationships between molecules. Rather, they collectively represent the encounter and bonding relationships between agent groups belonging to spatial points and cannot be compared directly to the encounter network.

Emergence Condition (1): (Not met) A symbol can be independently created and removed by an agent.

Emergence Condition (2): (Met) There are no restrictions on the amount of genetic information held by the agent program.

Emergence Condition (3): (Met) Machine instructions that modify other agent programs can be prepared.

Emergence Condition (4): (Not met) Both genotypes and phenotypes are written as agent programs and do not require translation.

Emergence Condition (5): (Met) Splitting of clusters means walls.

Emergence Condition (6): (Met) The cluster size is variable.

Emergence Condition (7): (Partially met) The change in the order of clusters (mingling) does not occur naturally, but may be prompted by external disturbances.

Emergence Condition (8): (Met) The appropriate design of transport control for agents produces signal agents.

Emergence Condition (9): (Met) Randomness in agent movement may randomize the system.

Emergence Condition (10): (Met) The agent movement is transport-controlled by agent type and by edge type and direction.

Emergence Condition (11): (Partially met) Appropriate design of the transport control enables movement across walls.

Emergence Condition (12): (Not met) The symbol sequence is implemented as a program and cannot describe many-to-one interactions.

Emergence Condition (13): (Not met) Agents cannot combine with each other.

Topological Condition (1): (Met) A topological distance is defined in the graph between agents containing symbols.

Topological Condition (2): (Met) The topological distance changes with agent transport.

Topological Condition (3): (Met) A reaction is possible only between agents belonging to the same node.

Topological Properties (1)–(4): (Met) The hy network in the hydrophilic cluster produces a structure close to a pseudo-two-dimensional square lattice.

As shown here, the MNAC expressing molecules as agents rather than nodes gives molecules (complexes) significant mobility, whereas it fails to satisfy Emergence Conditions (12) and (13). Improving the MNAC to satisfy these conditions remains a problem for the future.

Graph Elements

The experimental program is coded in Java with nodes, edges, and agents implemented as classes. The nodes are classified into hydrophilic and hydrophobic types. The edges are classified into covalent (cv, directional), hydrogen (hy, directional), and van der Waals (wa, non-directional) types. The agents are classified into Type 0 (for splitting clusters, centrosomes), Type 1 (for rewiring the hy edges), and Type 2 (for rewiring the wa edges).

Figure 3.37 schematically illustrates the functions of these class instances with major variables and methods. Each instance has basic variables such as label and type and additional variables for the link. Changing these variables by executing the methods changes the link relationship in the graph.

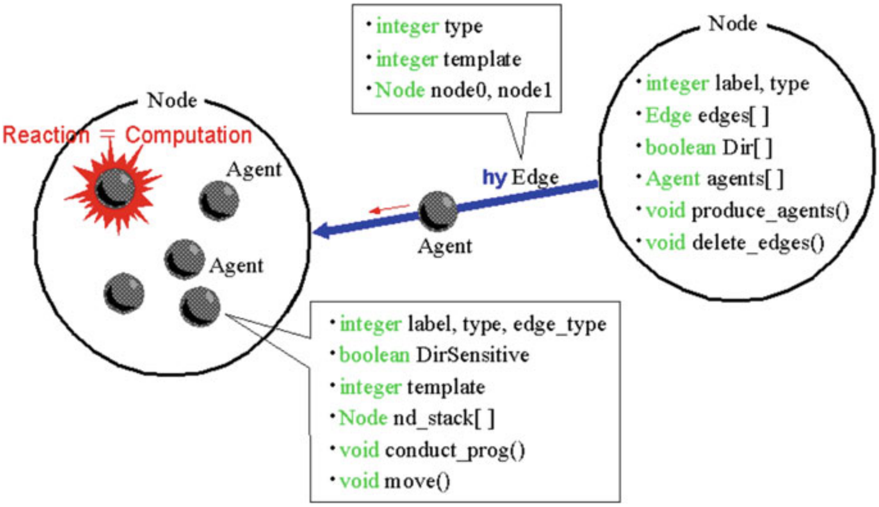


Fig. 3.37 Schematic diagram of nodes, edges, and agents in MNAC and their major instance variables and methods. The node variable, edges[ ], is a pointer to the edges connected to the node. The node variable, agents[ ], is a pointer to the agents belonging to the node. The edge variables, node0 and node1, are pointers to nodes on the end points of the edge. The agent stack, nd_stck[ ], is a pointer to nodes previously visited by the agent. The node methods, produce_ agents() and delete_edges(), and the agent methods, conduct_prog() and move(), are methods of execution (see the main text)

Execution Operations

The simulation progresses by repeating the following four step operations:

Step 1: Node.produce_agents ()

(Agent creation by node):

This operation creates or removes the agents at each node to reach the target number specified in advance. The target number is separately determined according to node and agent types.

Step 2: Agent.conduct_prog ()

(Program execution by agent):

This operation executes the agent program, rewiring the edges.

Step 3: Node.delete_edges ()

(Edge cutting by node):

This operation cuts the edges so that the edge degree in each node does not exceed the upper limit. The upper limit of the degree is determined by node and edge types.

Step 4: Agent.move ()

(Agent movement):

This operation moves all agents from the current to the next node. The edge through which an agent passes is determined by agent variables: edge_type, DirSensitive, and template.

Among the four operations above, the three other than Agent.conduct_prog () are common to all nodes and agents. However, the algorithm executed by Agent.conduct_prog () depends on the agent type. Roughly speaking, Type 1 or 2 agents function locally to form hy-squares and wa-triangles, while Type 0 agents recognize differences between the labels of the other Type 0 agents in the same node, removing agents with different labels and cutting the edges through which they pass. For further discussion of this algorithm, see [71, 73].

The target number and upper limit degree used in Steps 1 and 3 specify the node type (spatial point). For example, the degree of the hy edge and the number of Type 1 agents for a hydrophilic node are 4, independent of the direction or template, and set to zero for a hydrophobic node. This specifies the hydrophobic nature of the node (the absence of hy edges). Table 3.4 summarizes the criteria used when the agent selects the edges in Step 4.

Table 3.4 Edge transport control of agents in Agent.move()

Edge type	Agent type		
	cv	hy	wa
Type 0	×	○	×
Type 1	×	○[a]	×
Type 2	○	○	○

The *circles* indicate that the agent can move through an edge of the type; the *crosses* indicate that the agent cannot move through an edge of the type

[a] Edge selection considering both direction and template

Experimental Results

Basic Setting

The experiment starts from an initial regular random graph (random graph with uniform degree) containing $N = 2{,}000$ nodes (1,400 nodes are hydrophilic, 600 nodes are hydrophobic). The degrees of the hy and wa edges are set to $K = 0$ and $K = 4$, respectively, in all nodes. Each of the 600 hydrophobic nodes is linked to a hydrophilic node with a cv edge to form 600 amphipathic dipoles in a manner analogous to biological lipid molecules. Since this experiment does not prepare operations for cutting and creating cv edges, these dipoles are completely maintained throughout the experiment.

After starting the execution operation, we prepare two Type 0 agents with different labels at the point of 100 iterations and substitute them into two randomly selected hydrophilic nodes. We observe the change in the graph topology as the operation executes.

Formation and Splitting of Hydrophilic Clusters with Pseudo-regular Structures

Figure 3.38 shows the typical results obtained. The Types 1 and 2 agents created and their creation and rewiring of hy and wa edges change the topology of the initial random graph shown in Fig. 3.38a. After 100 iterations, the graph assumes a shell structure in which hydrophilic nodes are densely connected by the hy network and accumulated at the center and hydrophobic nodes are pushed out to peripheral areas (Fig. 3.38b). Owing to the functions of the Type 1 agent, the hy network here has a structure close to a pseudo-two-dimensional square lattice with large average path length (L), forming the basis for the split in the next stage.

Although not shown here, each graph contains (a) 0, (b) 7,867, (c) 22,222, or (d) 28,449 molecular agents.

Figure 3.38b and the subsequent figures show the growth and percolation of the injected Type 2 agents. This agent behaves much like a centrosome in a living cell and clusters (splits) the hydrophilic nodes by their label values. This split occurs between iterations 150 and 250. Figure 3.38c shows the intermediate graph. The state with split clusters continues through to the final iteration, 500, until the simulation is stopped (Fig. 3.38d).

This experiment is performed ten times with different random number seeds. Figure 3.39 compares the graph for each case after 300 iterations projected onto a plane. As the figures show, the sizes of the clusters formed differ in each case, but all of the experiments produce the same qualitative results.

Although we do not show the resulting graph here, according to a separate supplementary experiment performed to investigate the function of the Type 1 agent, if only Type 1 agents operate in a graph consisting of $N = 512$ hydrophilic nodes, the average path length of the hy network formed can be as large as

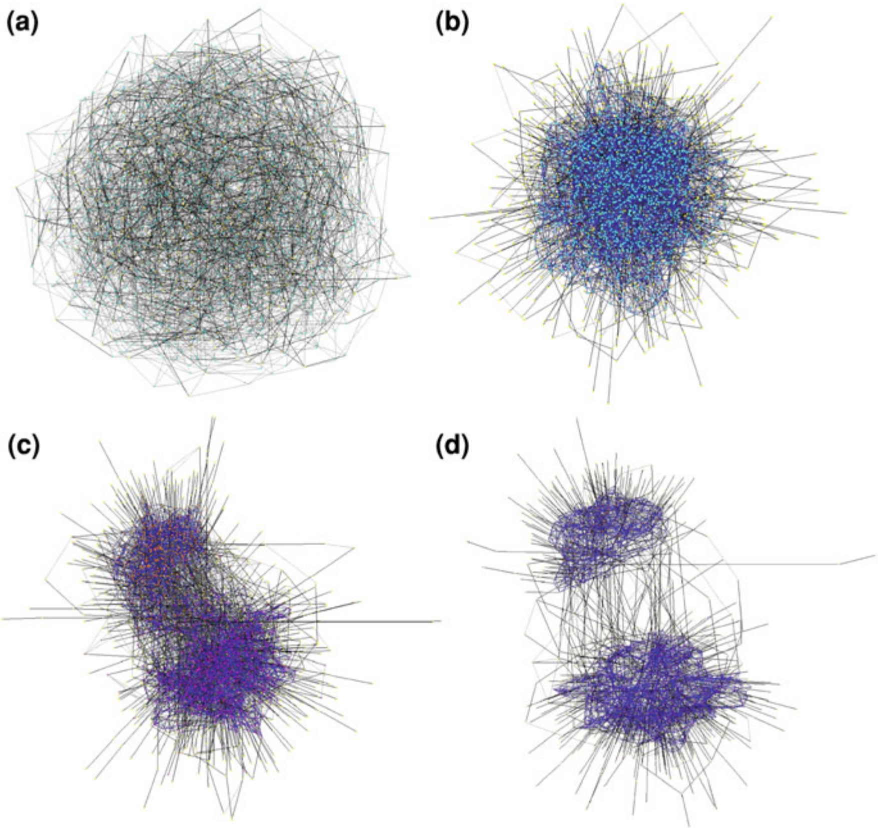


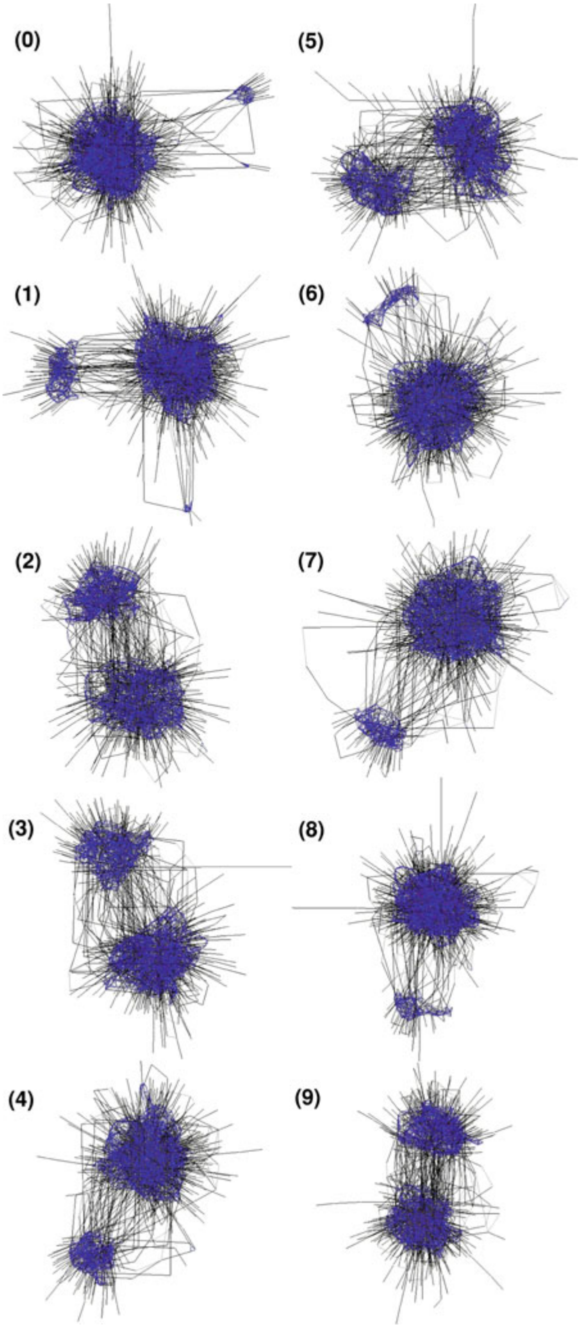
Fig. 3.38 Typical results from the experiment. **a** Two-dimensional rendering of initial regular random MNAC graph; **b** two-dimensional rendering of MNAC graph after 100 iterations; **c** two-dimensional rendering of MNAC graph after 180 iterations; **d** two-dimensional rendering of MNAC graph after 500 iterations

$L \approx 6.7$. This value exceeds $L = 6.01$, the value for a three-dimensional cubic lattice with the same number of nodes.

3.7 Future Prospects

This chapter discussed studies of artificial chemistry performed over the past 17 years, focusing mainly on two aspects: investigations and evaluations of design methodologies and progress in research on NAC. As described in Sect. 3.6.2, the network artificial chemistry (NAC and MNAC) systems obtained via this research remain incomplete—many problems remain to be solved before we can use this NAC to construct an artificial evolutionary system.

Fig. 3.39 Two-dimensional rendering of MNAC graphs after 300 iterations for each of ten experiments conducted with different random number sequences



In lieu of a conclusion, the last section of this chapter focuses on the new program-flow computing computational model, which has emerged as a byproduct of MNAC research, and describes future research problems and potential engineering applications.

In short, program-flow computing is a network computational model in which CPUs (nodes) connected by communication lines (edges) execute in parallel many programs that pass through them. This model differs from an ordinary parallel programing model in that the programs themselves are carried by the agents. This means the functions of the nodes or the operations performed in the nodes can change dynamically during execution. In this manner, for example, we can prepare various different types of agent programs, transport them in the network, and execute them on each node asynchronously, imparting complex functions to the entire network. We can design new engineering models for several systems based on program-flow computing. Some examples are given below.

3.7.1 Application to the Graph Coloring Problem

If the functions of the agents are selected appropriately, program-flow computing can be used to solve an optimization problem embedded in a network in which the agents operate. Figure 3.40 shows one example. Here, to solve the node coloring problem of a graph, a simple rule is prepared as the agent program: “If the color of the node to which the agent currently belongs is the same as the color of the node to which the agent belonged one step previously, the agent changes the color of the present node to another color.” The figure shows how the movement and parallel execution of the agents change the node colors of the graph over time. This preliminary experiment shows that the initial randomly colored graph (Fig. 3.40a) gradually eliminates competition between the nodes through the execution of the agent program (Fig. 3.40b). In the end, the graph has different colors assigned to the ends of all edges (Fig. 3.40c). This example results in an optimal global solution for “graph coloring” based on the local selection by a single type of simple program. We believe we can apply a similar method to combinatorial optimization problems related to many other graphs.

3.7.2 Application to Neural Network Modeling

Another application and development of program-flow computing involves applications to neural network information processing. Neural networks, which began as mathematical models, are information processing models based on the signal processing that takes place in animal brains. Information processing within and between living neurons in the brain is basically performed by “molecules.” Likewise, implementing molecular agents that perform processing while moving

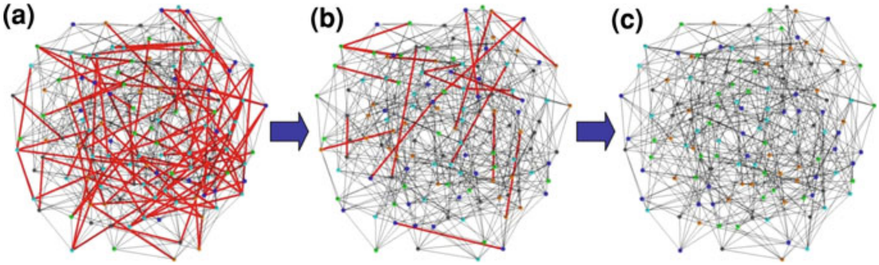


Fig. 3.40 Example of applying program-flow computing to node coloring problem of regular random graph. The problem is coloring the nodes of a regular random graph with the node number $N = 128$ and the uniform degree $k = 8$ using five predefined colors. **a** Is the initial state, **b** is the intermediate state, and **c** is the final state. In the figure, the *edges* indicated by heavy lines have nodes of the same color at both end points. The graph in the final state contains 1,024 molecular agents

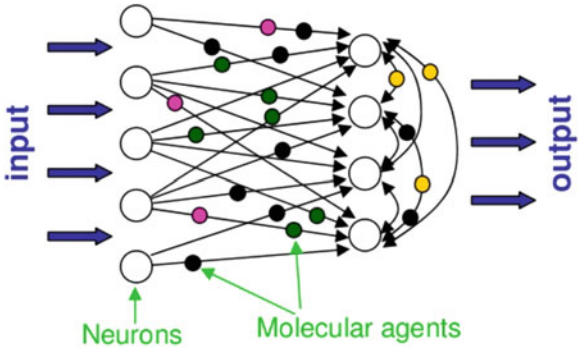


Fig. 3.41 Applying program-flow computing to neural network information processing. Various molecular agents are prepared, including those for the Hebbian rule, lateral inhibition, axon generation, and cell differentiation. The firing (reaction) and movement of these agents emulate signal propagation and learning in the neural network

between the nodes (neurons) via the edges (axons) in an artificial neural network may help generate an integrated account of processing related to signal propagation and learning (Fig. 3.41).

Our ongoing research efforts include building these models and evaluating their validity. One future goal is to propose new models for non-von Neumann information processing and information communication.

The Academic Frontier, Intelligent Information Science Project

The research described in this chapter was supported in part through the Academic Frontier, Intelligent Information Science (AFIIS) project entitled “The Clarification of Intelligence in Humans and Living Systems and its Applications”

(Representative: Mitsunori Miki, Professor at Doshisha University) at Doshisha University when the author was affiliated with both the ATR Network Informatics Laboratories (former Director, Dr. Katsunori Shimohara, now Professor at Doshisha University) and the NICT.

References

1. C. Adami, C.T. Brown, Evolutionary learning in the 2D artificial life system "Avida", in *Artificial Life IV: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systemsed*, ed. by R. Brooks, P. Maes (MIT Press, Cambridge, 1994), pp. 377–381
2. C. Adami, Learning and complexity in genetic auto-adaptive systems. Physica D **80**, 154–170 (1995)
3. C. Adami, *Introduction to Artificial Life* (Springer-Verlag, Santa Clara, CA, 1998)
4. C. Adami, C. Ofria, T.C. Collier, Evolution of biological complexity. Proc. Natl. Acad. Sci. U S A **97**, 4463–4468 (2000)
5. aiSee: Commercial software for visualizing graphs with various algorithms such as rubberband. http://www.aisee.com/
6. U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall (Crc Mathematical and Computational Biology Series, 2006)
7. Y. Asada, R. Suzuki, T. Arita, A simple artificial chemistry model based on the relationships between particles. In: Proceedings of the 36th SICE Symposium on Intelligent Systems, The Society of Instrument and Control Engineers (2009), pp. 227–232
8. G.M. Barrow, *Physical Chemistry* (McGraw-Hill Education, New York, 1988), Chapters 15–17. Japanese translation: Barrow, translated into Japanese by H. Daimon, K. Domen, Butsuri Kagaku Dai 6 Han (Ge) (Physical Chemistry, 6th edn.). (Tokyo Kagaku Dojin, Tokyo, 1999)
9. M. Bedau, P. Husbands, T. Hutton, S. Kumar, H. Suzuki (eds.) Workshop and Tutorial Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (Alife IX). (2004)
10. P. Dittrich, W. Banzhaf, Self-evolution in a constructive binary string system. Artificial Life **4**, 203–220 (1998)
11. P. Dittrich, J. Ziegler, W. Banzhaf, Artificial chemistries-a review. Artificial Life **7**, 225–275 (2001)
12. P. Espanol, P.B. Warren, Statistical-mechanics of dissipative particle dynamics. Europhysics Letters **30**(4), 191–196 (1995)
13. R. Ewaschuk, P.D. Turney, Self-replication and self-assembly for manufacturing. Artificial Life **12**(3), 411–433 (2006)
14. W. Fontana, Algorithmic chemistry, in *Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, ed. by C.G. Langton et al. Santa Fe Institute Studies in the Sciences of Complexity, Vol. 10 (Addison-Wesley, 1992), pp. 159–209
15. W. Fontana, L.W. Buss, 'The arrival of the fittest': toward a theory of biological organization. Bull. Math. Biol. **56**, 1–64 (1994)
16. T. Gánti, Organisation of chemical reactions into dividing and metabolizing units: the chemotons. BioSystems **7**, 189–195 (1975)
17. J.A. Glazier, F. Graner, Simulation of the differential adhesion driven rearrangement of biological cells. Phys. Rev. E **47**, 2128–2154 (1993)
18. F. Graner, J.A. Glazier, Simulation of biological cell sorting using a two-dimensional extended Potts model. Phys. Rev. Lett. **69**, 2013–2016 (1992)

19. R.D. Groot, P.B. Warren, Dissipative particle dynamics: bridging the gap between atomistic and mesoscopic simulation. Journal of Chemical Physics **107**, 4423–4435 (1997)
20. N. Hirokawa, Kinesin and Dynein Superfamily Proteins and the Mechanism of Organelle Transport. Science **279**, 519–526 (1998)
21. P.J. Hoogerbrugge, J.M.V.A. Koelman, Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. Europhysics Letters **19**(3), 155–160 (1992)
22. T.J. Hutton, Evolvable self-replicating molecules in an artificial chemistry. Artificial Life **8**, 341–356 (2002)
23. T.J. Hutton, Information-replicating molecules with programmable enzymes, in *Proceedings of the Sixth International Conference on Humans and Computers (HC-2003)*, 2003, pp. 170–175
24. T.J. Hutton, Evolvable self-reproducing cells in a two-dimensional artificial chemistry. Artificial Life **13**(1), 11–30 (2007)
25. T.J. Hutton, The organic builder: a public experiment in artificial chemistries and self-replication. Artificial Life **15**(1), 21–28 (2009)
26. K. Imai, T. Hori, K. Morita, Self-reproduction in three-dimensional reversible cellular space. Artificial Life **8**(2), 155–174 (2002)
27. N. Ishii, K. Nakahigashi, T. Baba, M. Robert, T. Soga, A. Kanai, T. Hirasawa, M. Naba, K. Hirai, A. Hoque, P.Y. Ho, Y. Kakazu, K. Sugawara, S. Igarashi, S. Harada, T. Masuda, N. Sugiyama, T. Togashi, M. Hasegawa, Y. Takai, K. Yugi, K. Arakawa, N. Iwata, Y. Toya, Y. Nakayama, T. Nishioka, K. Shimizu, H. Mori, M. Tomita, Multiple high-throughput analyses monitor the response of E. coli to perturbations. Science **316**(5824), 593–597 (2007)
28. H. Kitano, Systems biology: a brief overview. Science **295**(5560), 1662–1664 (2002)
29. C.G. Langton, Self-reproduction in cellular automata. Physica D **10**, 135–144 (1984)
30. J.D. Lohn, J.A. Reggia, Automatic discovery of self-replicating structures in cellular automata. IEEE Transactions on Evolutionary Computation **1**, 165–178 (1997)
31. D. Madina, N. Ono, T. Ikegami, Cellular evolution in a 3D lattice artificial chemistry, in *Advances in Artificial Life (7th European Conference on Artificial Life Proceedings)*, ed. by W. Banzhaf, T. Christaller, P. Dittrich, J.T. Kim, J. Ziegler (Springer-Verlag, Berlin, 2003), pp. 59–68
32. T. Maeshiro, M. Kimura, The role of robustness and changeability on the origin and evolution of genetic codes. Proc. Nat. Acad. Sci. USA **95**, 5088–5093 (1998)
33. J. Maynard-Smith, E. Szathmáry, *The Major Transitions in Evolution*. Springer-Verlag, Berlin (1995). Japanese translation: J. Maynard-Smith, E. Szathmáry, translated into Japanese by K. Nagano, Shinka Suru Kaisou (Springer-Verlag Tokyo, 1997)
34. B. McMullin, F.R. Varela, Rediscovering computational autopoieses, in *Proceedings of the 4th European Conference on Artificial Life*, ed. by P. Husband, I. Harvey (MIT Press, Cambridge, MA, 1997), pp. 38–47
35. B. McMullin, D. Groß, D, Towards the Implementation of evolving autopoietic artificial agents, in *Advances in Artificial Life (6th European Conference on Artificial Life Proceedings)*, ed. by J. Kelemen, P. Sosik (Springer-Verlag, Berlin, 2001), pp. 440–443
36. D. Noble, *The Music of Life: Biology Beyond Genes* (Oxford University Press, 2008). Japanese translation: D. Noble, translated into Japanese by Y. Kurachi, *Seimei-no Ongaku – Genomu wo Koete System Biology Heno Shoutai* (Shin-you sha, 2009)
37. H. Noguchi, M. Takasu, Self-assembly of amphiphiles into vesicles: a Brownian dynamics simulation. Phys. Rev. E **64** (2001) 041913
38. N. Ono, T. Ikegami, Model of self-replicating cell capable of self-maintenance, in *Advances in Artificial Life (5th European Conference on Artificial Life Proceedings)*, ed. by D. Floreano (Springer-Verlag, Berlin, 1999), pp. 399–406
39. N. Ono, T. Ikegami, Self-maintenance and self-reproduction in an abstract cell model. J. theor. Biol. **206**, 243–253 (2000)
40. N. Ono, T. Ikegami, Artificial chemistry: computational studies on the emergence of self-reproducing units, in *Advances in Artificial Life (6th European Conference on Artificial Life Proceedings)*, ed. by J. Kelemen, P. Sosik (Springer-Verlag, Berlin, 2001), pp. 186–195

41. N. Ono, H. Suzuki, String-based artificial chemistry that allows maintenance of different types of self-replicators. The Journal of Three Dimensional Images **16**(4), 148–153 (2002)
42. T. Oohashi, H. Sayama, O. Ueno, T. Maekawa, Programmed self-decomposition model and artificial life. *Prodeedings of the 1995 International Workshop on Biologically Inspired Evolutionary Systems*. Sony CSL, Tokyo (1995), pp. 85–92
43. T. Oohashi, T. Maekawa, O. Ueno, E. Nishina, N. Kawai, Requirements for immortal ALife to exterminate mortal ALife in one finite, heterogeneous ecosystem, in *Advances in Artificial Life (5th European Conference on Artificial Life Proceedings)*, ed. by D. Floreano et al. (Springer-Verlag, Berlin, 1999), pp. 49–53
44. T. Oohashi, T. Maekawa, O. Ueno, N. Kawai, E. Nishina, K. Shimohara, Artificial life based on the programmed self-decomposition model: SIVA. Journal of Artificial Life and Robotics **5**, 77–87 (2001)
45. T. Oohashi, O. Ueno, T. Maekawa, N. Kawai, E. Nishina, M. Honda, An effective hierarchical model for the biomolecular covalent bond: an approach integrating artificial chemistry and an actual terrestrial life system. Artificial Life **15**(1), 29–58 (2009)
46. N.B. Ouchi, J.A. Glazier, J.P. Rieu, A. Upadhyaya, Y. Sawada, Improving the realism of the cellular Potts model in simulations of biological cells. Physica A **329**(3–4), 451–458 (2003)
47. A.N. Pargellis, The spontaneous generation of digital ''Life''. Physica D **91**, 86–96 (1996a)
48. A.N. Pargellis, The evolution of self-replicating computer organisms. Physica D **98**, 111–127 (1996b)
49. A.N. Pargellis, Digital life behavior in the Amoeba world. Artificial Life **7**, 63–75 (2001)
50. T.S. Ray, An approach to the synthesis of life, in *Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, ed. by C.G. Langton et al. Santa Fe Institute Studies in the Sciences of Complexity, Vol. 10 (Addison-Wesley, 1992), pp. 371–408
51. T.S. Ray, J. Hart, Evolution of differentiated multi-threaded digital organisms, in *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, ed. by C. Adami, R.K. Belew, H. Kitano, C.E. Taylor (MIT Press, Cambridge, MA, 1998), pp. 295–304
52. H. Sayama, A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. Artificial Life **5**(4), 343–365 (2000)
53. H. Sayama, Self-replicating worms that increase structural complexity through gene transmission, in *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, ed. by M.A. Bedau et al. (MIT Press, Cambridge, 2000), pp. 21–30
54. H. Sayama, Swarm chemistry. Artificial Life **15**(1), 105–114 (2009)
55. A. Smith, P. Turney, R. Ewaschuk, Self-replicating machines in continuous space with virtual physics. Artificial Life **9**(1), 21–40 (2003)
56. P. Speroni di Fenizio, W. Banzhaf, Stability of metabolic and balanced organisations, in *Advances in Artificial Life (6th European Conference on Artificial Life Proceedings)*, ed. by J. Kelemen, P. Sosik (Springer-Verlag, Berlin, 2001), pp. 196–205
57. P. Speroni di Fenizio, P. Dittrich, W. Banzhaf, Spontaneous formation of proto-cells in an universal artificial chemistry on a planar graph, in *Advances in Artificial Life (6th European Conference on Artificial Life Proceedings)*, ed. by J. Kelemen, P. Sosik (Springer-Verlag, Berlin, 2001), pp. 206–215
58. H. Suzuki, An approach to biological computation: unicellular core-memory creatures evolved using genetic algorithms. Artificial Life **5**(4), 367–386 (1999)
59. H. Suzuki, Evolution of self-reproducing programs in a core propelled by parallel protein execution. Artificial Life **6**(2), 103–108 (2000a)
60. H. Suzuki, N. Ono, Universal replication in a string-based artificial chemistry system. The Journal of Three Dimensional Images **16**(4), 154–159 (2002)
61. H. Suzuki, N. Ono, K. Yuta, Several necessary conditions for the evolution of complex forms of life in an artificial environment. Artificial Life **9**(2), 537–558 (2003)
62. H. Suzuki, Artificial chemistry on small-world networks, in *Proceedings of the 18th Annual Conference of JSAI*, 2H4-03 (2004)

63. H. Suzuki, Spacial representation for artificial chemistry based on small-world networks, in *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (Artificial Life IX)* (2004), ed. by J. Pollack, M. Bedau, P. Husbands, T. Ikegami, R.A. Watson, pp. 507–513
64. H. Suzuki, Network artificial chemistry—molecular interaction represented by a graph, in *Workshop and Tutorial Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (Alife IX)* (2004). ed. by M. Bedau, P. Husbands, T. Hutton, S. Kumar, H. Suzuki), pp. 63–70
65. H. Suzuki, Computational folding of molecular chains in network artificial chemistry, in *Proceedings of the 32th SICE Symposium on Intelligent Systems*. The Society of Instrument and Control Engineers (2005), pp. 383–386
66. H. Suzuki, N. Ono, *Statistical mechanical rewiring in network artificial chemistry. In: The 8th European Conference on Artificial Life (ECAL) Workshop Proceedings CD-ROM* (Canterbury, UK, 2005)
67. H. Suzuki, N. Ono, Network rewiring rules representing molecular diffusion, in *Proceedings of the 11th Emergent System Symposium (ESS) "Emergence Summer School 2005"*, Toyama, Japan. The Society of Instrument and Control Engineers (2005), pp. 127–130
68. H. Suzuki, Mathematical folding of node chains in a molecular network. BioSystems **87**, 125–135 (2007)
69. H. Suzuki, An approach toward emulating molecular interaction with a graph. Australian Journal of Chemistry **59**, 869–873 (2006)
70. H. Suzuki, A node program that creates regular structure in a graph, in *International Conference on Morphological Computation, Conference Proceedings*, March 26–28, 2007, ECLT, Venice Italy
71. H. Suzuki, A network cell with molecular tokens that divides from centrosome signals, in *Proceedings of the Seventh International Workshop on Information Processing in Cells and Tissues (IPCAT)*, ed. by N. Crook, T. Scheper (Oxford Brookes University, 2007), pp. 293–304
72. H. Suzuki, Structural organization in network artificial chemistry by node programs and token flow, in *SICE Annual Conference 2007 Proceedings*. The Society of Instrument and Control Engineers (SICE), Japan (2007) 1C10, pp. 884–889
73. H. Suzuki, A network cell with molecular agents that divides from centrosome signals. BioSystems **94**, 118–125 (2008)
74. H. Suzuki, P. Dittrich (eds.), Special Issue on Artificial Chemistry. Artif. Life **15**(1) (2009)
75. K. Suzuki, T. Ikegami, Shapes and self-movement in protocell systems. Artificial Life **15**(1), 59–70 (2009)
76. Y. Suzuki, S. Tsumoto, H. Tanaka, H, Analysis of cycles in symbolic chemical system based on abstract rewriting system on multisets, in *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, ed. by C. Langton, K. Shimohara (MIT Press, Cambridge, MA, 1997), pp. 521–528
77. Y. Suzuki, H. Tanaka, Order parameter for a symbolic chemical system, in *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, ed. by C. Adami, R.K. Belew, H. Kitano, C.E. Taylor (MIT Press, Cambridge, MA, 1998), pp. 130–139
78. Y. Suzuki, H. Tanaka, Chemical evolution among artificial proto-cells, in *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, ed. by M.A. Bedau et al. (MIT Press, Cambridge, 2000), pp. 54–63
79. Y. Suzuki, Y. Fujiwara, Y., J. Takabayashi, H. Tanaka, Artificial life applications of a class of P systems: abstract rewriting systems on multisets. *Lecture Notes in Computer Science*, Vol. 2235 (Multiset Processing) Springer, Berlin/Heidelberg (2001), pp. 299–346
80. E. Szathmáry, J. Maynard-Smith, From replicators to reproducers: the first major transitions leading to life. J. theor. Biol. **187**, 555–571 (1997)
81. K. Tominaga, Modelling DNA computation by an artificial chemistry based on pattern matching and recombination, in *Workshop and Tutorial Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (Alife IX)*, ed. by M. Bedau, P. Husbands, T. Hutton, S. Kumar, H. Suzuki (2004), pp. 56–62

82. K. Tominaga, T. Watanabe, K. Kobayashi, M. Nakamura, K. Kishi, M. Kazuno, Modeling molecular computing systems by an artificial chemistry - its expressive power and application. Artificial Life **13**(3), 223–247 (2007)
83. K. Tominaga, Y. Suzuki, K. Kobayashi, T. Watanabe, K. Koizumi, K. Kishi, Modeling biochemical pathways using an artificial chemistry. Artificial Life **15**(1), 115–129 (2009)
84. J. von Neumann, *Theory of self-reproducing automata* (University of Illinois Press, Urbana. Edited and completed by A.W. Burks, 1966)
85. S. Yamamoto, Y. Maruyama, S. Hyodo, The dissipative particle dynamics study of spontaneous vesicle formation of amphiphilic molecules. Journal of Chemical Physics **116**(13), 5842–5849 (2002)
86. T. Yamamoto, K. Kaneko, Tile automation: a model for an architecture of a living system. Artificial Life **5**, 37–76 (1999)
87. Barrow G. M. Physical chemistry, McGraw-Hill Education, Chapters 15–17, 1988. Japanese translation: Barrow, translated into Japanese by Hiroshi Daimon and Kazunari Domen, Butsuri Kagaku Dai 6 Han (Ge) (Physical Chemistry 6th Edition), Tokyo Kagaku Dojin, 1999
88. Masahiro Kotani, Kiyohiko Someda, and Seiichiro Kouda, edited by Tamotsu Kondo, Daigakuin Kogi Butsuri Kagaku (Graduate School Course Physical Chemistry), Tokyo Kagaku Dojin, 1997
89. Vemulapalli G.K. Physical chemistry, Prentice-Hall Inc., Chapters 23–33, 1993. Japanese translation: Vemulapalli, translated into Japanese and supervised by Ueno et al., Butsuri Kagaku III Kagaku Hanno Sokudoron to Tokei Netsurikigaku (Physical Chemistry III Chemical Reaction Velocity Theory and Statistical Thermodynamics), Maruzen Co. Ltd., 2000
90. R. Albert, A.L. Bárabasi, Statistical mechanics of complex networks. Reviews of Modern Physics **74**(1), 47–98 (2002)
91. J. Davidsen, H. Ebel, S. Bornholdt, Emergence of a small world from local interactions - Modeling acquaintance networks. Physical Review Letters **88**(12), 128701 (2002)
92. Naoki Masuda, Norio Konno, Fukuzatsu Nettowaku no Kagaku (Science of Complex Networks), Sangyo Tosho, 2005
93. M.E.J Newman, The structure and function of complex networks. SIAM Review **45**, 167–256 (2003)
94. Wataru Soma, Katsunori Shimohara, Sumoru Warudo Nettowaku no Yakuwari (Role of Small-world Networks), Documents for Category II Meeting of Institute of Electronics, Information and Communication Engineers, NGN2001-12, 13–20, 2001
95. D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
96. T.S. Ray, An approach to the synthesis of life, C.G. Langton , et al. (eds.). Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Santa Fe Institute Studies in the Sciences of Complexity, **10**), 371-408, Addison-Wesley, 1992
97. Yoichi Muraoka, Heiretsu Shori (Sofutowea Koza) (Parallel Processing (Software Course)), Shokodo Co., Ltd., Section 5.7, 1986
98. Sharp J.A. Data flow computing (Ellis Horwood Series in Computers and Their Applications), Ellis Horwood Ltd., 1985
99. Masahiro Sowa, Deta-furo Mashin to Gengo (Sofutowea Koza) (Data flow Machine and Languages (Software Course)), Shokodo Co., Ltd., 1986
100. Ryota Shioya, Hidetsugu Irie, Masahiro Goshima, Shuichi Sakai, Evaluation of Area-Oriented Register Cache, Proceedings of Information Processing Society of Japan, 2008-ARC-178, pp. 13–18, 2008
101. Abelson H., Allen D., Coore D., Hanson C., Homsy G., Knight T.F., Nagpal R., Rauch E., Sussman G.J., Weiss R. Amorphous computing, Communications of the ACM, vol. 43(5), pp. 74–82, 2000

102. The Bio FAB Group, Baker D., Church G., Collins J., Endy D., Jacobson J., Keasling J., Modrich P., Smolke C., Weiss R. Engineering life: building a fab for biology, Scientific American, vol. 294(6), pp. 44–51, 2006. Japanese translation: Bio FAB Group, Baker D., Church G., Collins J., Endy D., Jacobson J., Keasling J., Modrich P., Smolke C., Weiss R., Gosei Seibutsugaku wo Kasokusuru Baio Fabu, Nikkei Science, vol. 36 (9), pp. 32–41, 2006

103. Katsuhiko Ariga, Toyoki Kunitake, Iwanami Koza Gendai Kagaku eno Nyumon <16>Chobunsikagaku eno Tenkai (Iwanami Lecture Series, Introduction to Modern Chemistry<16> Development of Supermolecular Chemistry), Iwanami Shoten, 2000

104. Seiji Shinkai, Kazuki Sada, Masayuki Takeuchi, Norifumi Fujita, Bunshi Kikai: Seitai wo Tegakari to shite (Molecular Machine: Using Living Bodies as Clues), Edited by Naoki Sugimoto, Kagaku Furontia 13 Nano Baio Enjiniaringu - Seimei to Busshitsu no Yugo wo Mezasite (Chemistry Frontier 13 Nanobioengineering—Toward a Fusion of Life and Materials), Kagaku-Dojin Publishing Company, Inc., 50–60, 2004

Chapter 4
Signal Transduction in Biological Systems and its Possible Uses in Computation and Communication Systems

Jian-Qin Liu and Ferdinand Peper

Abstract Cellular signal transduction is crucial for cell communications and is described by signaling pathway networks that sustain the biological functions of living cells. The robustness of the molecular mechanism of cellular signal transduction forms an important inspiration in the design of future communication networks based on the information processing mechanisms of cellular signal transduction. This chapter discusses some important aspects of the computational issues on cellular signal transduction: (1) How to formally represent kernel information of cellular signal transduction; (2) How to get a fixed point from a pathway network with feedbacks; and (3) How to encode information in signal transduction pathways by error-correcting codes, such as to increase the fault tolerance of the system, while at the same time conform to the unstructured nature of such pathways. The results obtained provide a basis for innovative future communications networks, with biological signaling pathway networks acting as references for systems with improved performance in factors like robustness.

Keywords Cell communication · Error-correcting codes · Signaling pathway networks · Signal transduction

4.1 Introduction

Signal transduction plays a crucial role in the complex dynamics of living cells to the extent that it is considered a fundamental information processing mechanism in living systems. The recent availability of data on signal transduction has the

J.-Q. Liu (✉) and F. Peper
National Institute of Information and Communications Technology, Kobe, Japan
e-mail: liu@nict.go.jp

H. Sawai (ed.), *Biological Functions for Information and Communication Technologies,* Studies in Computational Intelligence, 320,
DOI: 10.1007/978-3-642-15102-6_4, © Springer-Verlag Berlin Heidelberg 2011

potential for the creation of artificial systems conducting computation and communication using its inherent mechanisms. It also promises to give inspiration on building computation and communication systems in our world, which are based on novel principles only used in biological organisms till date.

It is yet unknown what advantages can be gained from using biologically inspired mechanisms in the application to information processing systems, but given the high efficiency by which biological organisms function, it makes sense to study them, especially in the framework of an information processing paradigm. The boundary conditions of the processes in biological systems tend to be quite different from what we are normally used to in our daily lives. Noise, for example, is large compared to signal levels. Mechanisms to cope with it in traditional systems include error-correcting codes, and it is an interesting issue to investigate whether and to what extent such techniques are applicable in biological systems. To investigate issues like these, it is important to have a formal model describing biological signal transduction. The most commonly used model is the network, which has topological features such as hubs in a scale-free network [1]. This suggests the exploration of efficient ways to systematically understand the robustness of networks in terms of graphs, where the building block of the signal transduction networks that are treated as complex systems called *network motif* in [2] are defined. Based on the technology of networks, we can model the dynamics of signal transduction networks and find a quantitative description of its signaling mechanism that sustains the robustness of the corresponding cellular signaling processes, which have been widely reported in the signal transduction networks for chemotaxis, heat shock response, ultrasensitivity, and cell cycle control [3].

In this chapter, we formulate a model of signal transduction in terms of graph theory to increase our understanding of its information processing role in biological systems. A graph is a set of nodes named as *vertexes* with relations between them called *edges*. In the framework of a communication network, for example, the edges represent the communication channels that exist and through which communication takes place between the nodes. Applied to biological systems, we obtain a formulation of the structure of signaling pathways of signal transduction. We illustrate our concepts through a particular protein called MAPK (Mitogen-Activating Protein Kinase), which plays an important role in intracellular communications processes. We then add concepts related to error-correcting codes to study how robustness of the above processes can be improved, as well as a new concept of *fixed points* in biological systems, which serve to restore signals through nonlinear dynamics with feedback.

This chapter is focused on computational aspects of signal transduction in cells. In Sect. 4.2, based on the biochemistry features of signal transduction processes, the data structure of graph is presented to formulate the signaling pathway of signal transduction, in which MAPK is discussed as an instance of pathways for describing the dynamic processes of signal transduction networks. In Sect. 4.3, a fixed point phenomenon is studied as well as the robustness factor for developing molecular communication systems. In Sect. 4.4, the molecular codes for error correction are designed from the network level of signal transduction.

Column 1: Network Motifs

Establishing a model is a key to describe biological signaling processes. One of the most widely used models is a scale-free network with topological features such as hubs. This model suggests an efficient method for systematically understanding network robustness in graphic form. It also specifies the smallest unit of network construction when viewing the signaling network as a complex system.

One reference [2] proposes the concept of *network motif*. With networking technologies, we can model the dynamics of a signaling network and quantitatively describe the signaling mechanisms supporting the robustness of the corresponding cellular signaling processes.

An example of the motif is the genetic sequence of telomeres obtained from fission yeast [4].

G(1–9)TTAC(0–1)A(0–1)C(0–1).

The sequence consists of duplicate genes. Here, G(1–9) represents the sequences shown below.

G, GG, GGG, GGGG, GGGGG, GGGGGG, GGGGGGG, GGGGGGGG, GGGGGGGGG

C(0–1) represents a blank or C. Similarly, A(0–1) represents a blank or A. The motif in this example is G(1–9)TTA. TTA appears as a fixed character string, while G(1–9) is variable. A network motif is an element of network construction, which appears frequently in the network structure as a characteristic template. It is the basis for understanding the configuration of the entire network from the viewpoint of systems science, with respect to network reconstruction.

Besides, as network motifs are studied in gene regulatory networks, they are also considered as a generic concept. However, experimental proof is still needed to confirm if they are applicable to all biomolecular networks.

Although the entire network is formulated as a graph, it is not so easy to handle it as a graph generated by a uniform distribution. Ideally, to analyze an actual network, we would analyze it qualitatively and quantitatively (comprehensively) using subgraphs (the subgraphs being network motifs) that replicate at high frequency. In a signaling pathway network, the above representation is used to describe each individual pathway. (See Fig. 4.2 of the main text.)

4.2 Cellular Signal Transduction Networks and their Formal Model

Signal transduction in cells is a biochemical process that is of fundamental importance for their functioning. In living cells, signal transduction is carried out by series of biochemical reactions that are regulated by genetic factors. The signals used in signal transduction are usually quantified by concentrations of the corresponding chemicals.

4.2.1 Some Preliminaries of the Biochemistry of Signal Transduction

Cellular signal transduction is defined as a phenomenon, process, or mechanism that realizes a series of biochemical reactions in cells in response to stimuli of chemical signals outside the cells; this function includes the so-called cell communication.

Cell communication is the term used for the communication processes in cells that take the form of chemical signals and that is realized by the biochemical reactions in cells through cellular signal transduction. Cell communication can be distinguished into intercellular and intracellular communications.

Inter-cell communication describes how cells interact with each other. An important mechanism in inter-cell communication is formed by signaling molecules, which are also known as *first messengers*. Intercell communication has four types [5,12]: (1) contact-dependent signaling, in which cells have direct membrane-to-membrane contact to exchange signals, (2) paracrine signaling, in which cells release signals into the extracellular space to act locally on neighboring cells, (3) synaptic signaling, in which neuronal cells transmit signals electrically along their axons and release neurotransmitters at synapses, and (4) endocrine signaling, in which hormonal signals are secreted into the blood stream to be distributed on a wide scale throughout an organism's body.

Intra-cell communication concerns the communication within cells. When an incoming first messenger molecule reaches a cell, it cannot directly pass the cell membrane, but is bound by specific receptors that effectuate the activation of certain signaling molecule proteins within the cell. Referred to as *secondary messengers*, these signals are relayed by a chemical reaction process through a signaling cascade, which relays the signals to the nucleus of the cell. In this chapter, we will mainly discuss about intra-cell communication. Important signaling molecules in cells are proteins. We will be especially interested in proteins that can bind to a phosphate molecule. Such proteins are called *phosphoproteins*. When a phosphate is attached to a protein, it becomes *phosphorylized* through a *phosphorylation* process; when it becomes detached, it becomes *dephosphorylized* through a *dephosphorylation* process. To switch between the two states, special enzymes are required. The enzyme that realizes phosphorylation is called *kinase*, the enzyme that realizes dephosphorylation is called *phosphatase*.

The phosphorylation/dephosphorylation state of a protein will be used in the following to encode the binary state of a variable. This state can be detected by *immunofluorescence analysis*, which provides us with a possible tool to read out such a variable.

The signaling pathway in cells is a series of biochemical reactions, which have specific biological functions.

4.2.2 Graphic Representation for Signal Transduction

The reactions in a signaling pathway will be described by a directed graph with input and output. By the graphic representation, we can get the information form of the signal transduction network, from which we can investigate the structure, encoding, and networks of signal transduction in cells.

To study the structural relations concerning the transduction of signals, we define transduction in a spatial form as a graph. Let a molecule be represented by a vertex (node) in a graph and let a biochemical reaction be represented by an edge (link), then we obtain a graph.

$$G = \langle V, E \rangle,$$

where the vertex set is defined as the set $V = \{V_1, V_2, \ldots, V_n\}$, and the edge set is defined as $E = E(V_i, V_j)$ $(V_i, V_j \in V)$. Any parameters of a biochemical reaction represented by an edge are depicted as labels to that edge.

The direction of a biochemical reaction is represented in this formalism by a directed edge in the graph, which is graphically depicted as an arrow from one vertex to another vertex. In case a biochemical reaction is bidirectional, the corresponding edge is also bidirectional, and it will be depicted as merely a line between its vertices. Figure 4.1 shows a graph representing a pathway from a substrate vertex to a product vertex.

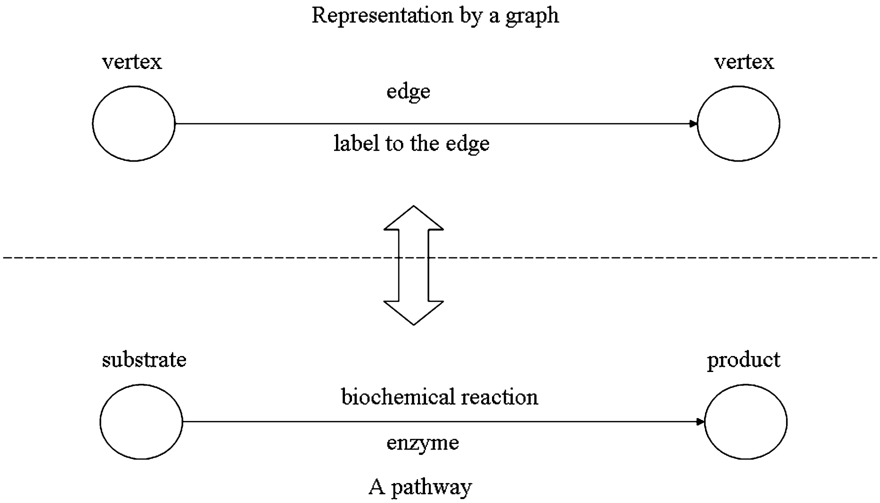


Fig. 4.1 A pathway: a graph versus signals

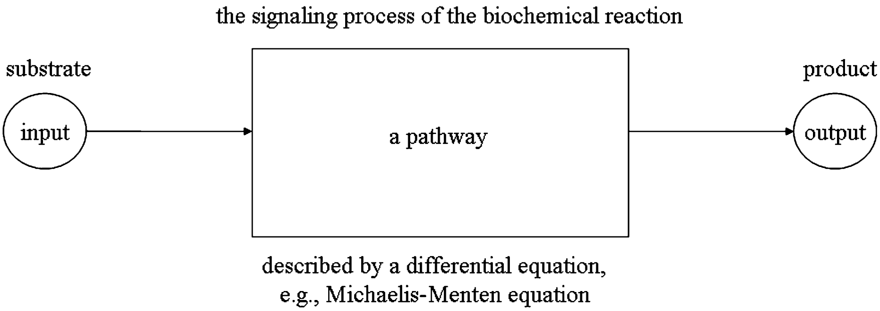

the signaling process of the biochemical reaction
substrate
input
a pathway
product
output
described by a differential equation,
e.g., Michaelis-Menten equation

Fig. 4.2 The concept of pathway as a system

The actual phosphorylation process follows the so-called *Michaelis–Menten equation* [6], which appears in the graph as a label on the edge between the substrate-vertex and the product-vertex. The Michaelis–Menten equation is described as follows. Let the reactant denote the input to the pathway, then the product is calculated by the Michaelis–Menten equation as

$$\mathrm{d}/\mathrm{d}t(\mathrm{product}(t)) = k_3 * \mathrm{enzyme} * \mathrm{substrate}(t)/(\mathrm{substrate}(t) + k_m)$$

where product(t) is the product concentration at time t, substrate(t) is the substrate concentration, enzyme is the enzyme concentration, and k_1, k_2, and k_3 are the coefficients of the biochemical reaction, wherein $k_{\mathrm{m}} = k_2/k_1(k_3 \ll k_2)$.

The above formalism is used for describing an individual pathway (Cf. Fig. 4.2). If a pathway cannot be divided into any other pathways, then the pathway is called *indivisible* or *atomic*. Atomic pathways are the building blocks from which more complex pathway networks are constructed. Such complex pathways are called *interacted pathways*. The entire pathway network will be constructed by these building blocks. In case of phosphorylation and dephosphorylation states, for example, it is possible to switch between these states via the interacted pathways that are regulated by kinases and phosphatases.

The interaction of different states needs to be investigated considering their influence on biological functions of cells. MAPK cascade is one of the important pathways with such features.

Column 2: Signaling Pathway Network

Functions of a Signaling Pathway Network

A *signaling pathway* refers to a biochemical reaction network representing interactions between the biomolecules in a living cell. Biologically, the functions associated with this pathway keep the cell alive. A biological signal is represented by intercellular or intracellular molecular concentrations. Besides, this signal is represented by the chemical concentration, it is also represented by the name of a molecule when the molecule serves as the signal transmission media. Since interactions among these molecules lead to biochemical reactions, they influence cell conditions biochemically and allow the cell to fulfill its functions. Intermolecular interactions are expressed by a series of corresponding biochemical reactions, and the biochemical reaction processes show the characteristics of temporal dynamics. When we focus on the direction of the biochemical reactions, biochemical reaction processes related to the intermolecular interactions suggest the concept of the *pathway*. A molecular set in the initial state of the cell, for example, changes to a new molecular set due to biochemical reactions generated by intermolecular interactions. If we define a cell as a system, the initial molecular set is the input and the final molecular set is the output.

Structure of Signaling Pathway Network

The structure of this pathway network is basically represented by a graph (see Fig. 4.1 in the main text). With recent technological developments in molecular biology, people began to analyze cell functions corresponding to genome sequences, and knowledge of biomolecular interactions in cells is being accumulated. The quantitative analysis of signaling pathway networks based on experimental biology will become possible in the future.

Column 3: Michaelis–Menten Kinetics

Analyzing and investigating a signaling pathway network can be done based on abstract models in view of computer science; in other words, graphic analysis. The biochemical processes of biomolecules for the pathway can be represented as nodes and links. As a simple example, consider the process of phosphorylation. In the graph, the node set consists of substrate and product nodes, while the pathway is represented by a label on the edge between a substrate node and a product node. This is a modeling tool in discrete mathematics. The knowledge of biochemical reactions is important for understanding the functions of biological systems. The Michaelis–Menten kinetics provides a quantitative method for studying these reactions.

In the graph, these reactions are represented by a label on the edge between a substrate node and a product node. With the substrate posited as the input to the pathway, the product is calculated by the Michaelis–Menten equation given below.

$$\mathrm{d}/\mathrm{d}t(\mathrm{product}(t)) = k_3 * \mathrm{enzyme} * \mathrm{substrate}(t)/(\mathrm{substrate}(t) + k_m)$$

Here, *product*(t) is the concentration of the product at time t; *substrate*(t) is the concentration of the substrate at time t; *enzyme* is the concentration of the enzyme; k_1, k_2, and k_3 are the coefficients of the biochemical reactions; and $k_\mathrm{m} = k_1/k_2$. The representation above is used to describe each pathway. (See Fig. 4.2 of the main text.) If a pathway cannot be divided into smaller pathways, then it is said to be *indivisible* or *atomic*. An atomic pathway is the smallest unit of network construction, i.e., a building block. Complex pathway networks, called *interaction pathways*, are constructed from atomic pathways. The entire pathway network consists of these units. For example, “phosphorylation” and “dephosphorylation” states can be switched through an interaction pathway regulated by “kinase” and “phosphatase.”

When analyzing interactions between different states, we need to consider the effects of the interactions of different states on the biological functions of the cell. The MAPK cascade is one of the important pathways that offer such a feature. It is appropriate to designate phosphorylation, dephosphorylation, and MAPK cascade as network motifs in a signaling pathway network and this will be useful in analyzing large-scale networks.

4.2.3 Example of Pathway: The MAPK Cascade

A MAPK cascade is an important pathway that is at the base of many biological functions in cells, such as in a phosphorylation process. Involved in a MAPK cascade is a number of kinases with the following names:

MAPK:	Mitogen activating protein kinase,
MAPKK:	Mitogen activating protein kinase kinase,
MAPKKK:	Mitogen activating protein kinase kinase kinase,
MAPKKKK:	Mitogen activating protein kinase kinase kinase kinase,

The resulting pathway is called *k*-layered, with *k* being an integer denoting the number of stages in the cascade. The structure of MAPK cascade is illustrated in Fig. 4.3. From top to down in the order going from upstream to down stream, MAPKKKK phosphorylates MAPKKK, MAPKKK phosphorylates MAPKK, MAPKK phosphorylates MAPK, which effectuates a phosphorylated protein as output to the process.

When building information models of signal transduction, it is important to structurally analyze the functionality of a signaling pathway network. As an example of a structural model, we show the MAPP cascade of budding yeast in Fig. 4.4, which involves the kinases Ste20, Ste11, Ste7, Kss1/Fus3, and Far1/Ste12. In this cascade, MAPKKKK is Ste20, MAPKKK is Ste11, MAPKK is Ste7, and there are two MAPK factors, Kss1 and Fus3. Furthermore, there are proteins at the bottom of the MAPK cascade, called Far1 and Ste12; these proteins, which form the output of the cascade, play an active role in the reproduction of cells. In technical terms, Far1 is a CDK (Cyclin-dependent Kinase) inhibitor, and Ste2 is a TF (Transcription Factor) Ste12. Other examples of signal pathways can be found in KEGG [7].

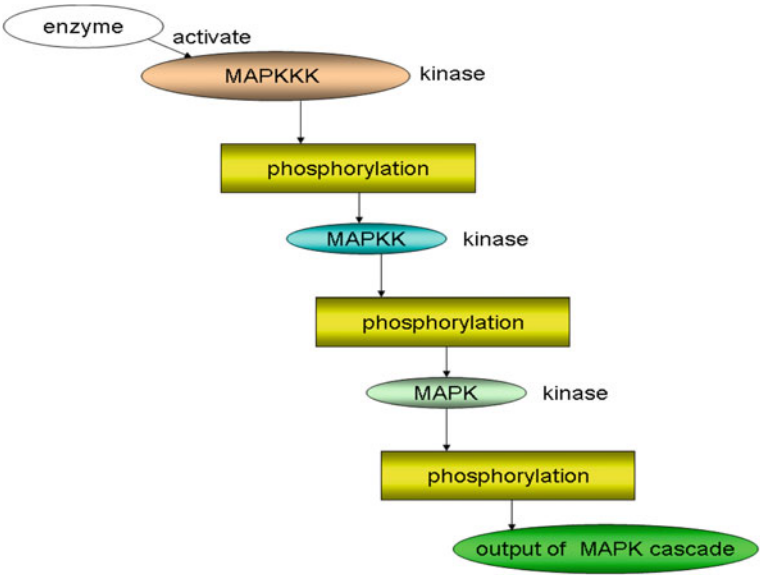


Fig. 4.3 The hierarchical structure of MAPK cascade

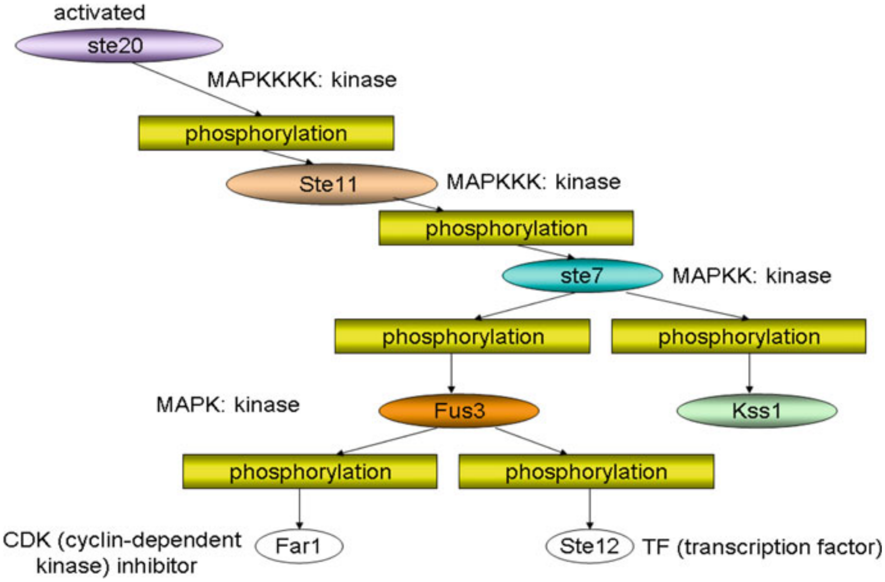


Fig. 4.4 An example of a MAPK cascade

Up to now, we have witnessed that the objects—nodes and links of pathway networks—can be represented by the vertexes and edges of graphs. Based on these, the dynamic features of networks can be investigated based on the computational formulation given above.

4.3 Dynamical Analysis of Signal Transduction Networks

4.3.1 Temporal Dynamics of Signal Transduction Networks

Based on the graphic structure, we formulated in previous sections, we will discuss the dynamics features of signal transduction networks to systematically understand their information processing mechanism. Basically, the dynamics of signal transduction can be investigated by two major aspects: *spatial dynamics* and *temporal dynamics* of signal transduction. In signal transduction, the spatial dynamics is mainly reflected in diffusion processes. Kholodenko's review article [8] on spatial dynamics uses the diffusion equation with polar coordinates to formulate the concentration values of kinases in the MAPK cascade constrained by the distance of diffusion.

This chapter focuses mostly on the information processing aspect of signal transduction networks, which clearly have a temporal character, and so we limit our discussion to the temporal dynamics of signal transduction. Such dynamics is usually formulated by differential equations, among which the Michaelis–Menten equation mentioned in Sect. 4.2 is the most fundamental. The Michaelis–Menten equation describes the biochemical reaction among molecules used for signaling in cells:

A *substrate* X is the input to the pathway under the regulation of an *enzyme* E, and *product* Y is the output of the pathway. The enzyme plays the role of catalyzer, which triggers the biochemical reaction and transforms the initial substrate into the resulting product.

$$\mathrm{S} + \mathrm{E} \xrightarrow{k_1} \mathrm{S} \cdot \mathrm{E} \xrightarrow{k_3} \mathrm{P} + \mathrm{E}$$

$$\mathrm{S} \cdot \mathrm{E} \xrightarrow{k_2} \mathrm{S} + \mathrm{E}$$

The parameters k_1, k_2, and k_3 describe the conversion rates and

$$k_{\mathrm{m}} = (k_2 + k_3)/k_1$$

which is called the Michaelis constant. In the following, we will denote the concentration of any chemical X as [X], describing the number of molecules per unit volume.

We can obtain a simple differential equation system for the kinetic dynamics of the product P, where E_0 is the total concentration of the enzyme.

$$\mathrm{d/dt}[\mathrm{P}] = k_3[\mathrm{E_0}] * [\mathrm{X}]/(k_\mathrm{m} + [\mathrm{X}]).$$

The above formulation, however, is only valid under quasi steady-state conditions, i.e., conditions in which the concentration of the substrate-bound enzyme changes at a much slower rate than those of the product and substrate. This allows the enzyme to be treated as constant, which is $\mathrm{E_0}$ in the formula.

A question arising naturally from here is how to use the above form to explain the MAPK cascade we already mentioned before. Therefore, we reformulate it as a set of coupled differential equations in which each stage in the cascade corresponds to one equation:

$$\begin{aligned}\mathrm{d/dt}[\mathrm{MAPKKK}] = {} & k_3(\mathrm{MAPKKK})[\mathrm{MAPKKKK}(0)] \\ & * [\mathrm{MAPKKK}]/(k_\mathrm{m}(\mathrm{MAPKKK}) + [\mathrm{MAPKKK}]),\end{aligned}$$

$$\begin{aligned}\mathrm{d/dt}[\mathrm{MAPKK}] = {} & k_3(\mathrm{MAPKK})[\mathrm{MAPKKK}(0)] \\ & * [\mathrm{MAPKK}]/(k_\mathrm{m}(\mathrm{MAPKK}) + [\mathrm{MAPKK}]),\end{aligned}$$

$$\mathrm{d/dt}[\mathrm{MAPK}] = k_3(\mathrm{MAPK})[\mathrm{MAPKK}(0)] * [\mathrm{MAPK}]/(k_\mathrm{m}(\mathrm{MAPK}) + [\mathrm{MAPK}]),$$

$$\begin{aligned}&\mathrm{d/dt}\,[\text{output-of-MAPKcascade}] = k_3\left(\text{output-of-MAPKcascade}\right)\left[\mathrm{MAPK}\left(0\right)\right] * \\ &\quad \left[\text{output-of-MAPKcascade}\right]/\left(k_\mathrm{m}\left(\text{output-of-MAPKcascade}\right)\right. \\ &\qquad \left. + \left[\text{output-of-MAPKcascade}\right]\right),\end{aligned}$$

With the appropriate parameters, usually obtained from empirical observations, the system MAPK cascade can be numerically calculated. Basically, the general behavior can be described as an amplification of the initial substrate concentration S(0), resulting in an enhanced signal with higher concentration of the resulting product P(final moment).

One of the well-known functions of the MAPK cascade in cellular signal transduction networks is to act as an amplifier for intracellular signaling processes. However, an unexpected phenomenon—a fixed point that occurs at a four-layered MAPK cascade where a feedback is embedded—is observed by simulation [9], which shows that in theory the second messengers' signals can be kept at a constant value during their relay processes within cells.

4.3.2 Fixed Point for Pathways with Feedbacks

Nonlinear dynamics features, such as bifurcation [3], have been reported in signal transduction networks. In order to study the computational and communication capacity of signal transduction networks, it is necessary to make sure to analyze how the cellular signals are controlled so that the information flow can be quantitatively measured. This framework forms the basis of the architectural design and performance analysis of engineered ICT systems inspired by signal transduction networks in cells. In this section, we take the fixed-point phenomena as an instance to demonstrate the nonlinear phenomena of signal transduction networks, as reported in [9].

4.3.2.1 The Model for the Simulation

As shown in Figs. 4.5 and 4.6, the structure of a MAPK cascade is layered, with feedback being embedded into each layer.

The set of coupled differential equations, in case feedback is present, then becomes

$$\begin{aligned}&\mathrm{d/dt[phosphor\text{-}protein\ in\ MAPKKK\ layer]}\\&\quad = -k_3[\mathrm{MAPKKK}][\mathrm{MAPKKKK}(0)]\\&\qquad * [\mathrm{MAPKKK}]/(k_\mathrm{m}(\mathrm{MAPKKK}) + [\mathrm{MAPKKK}]) - \int [\mathrm{MAPKK}]\mathrm{dt},\end{aligned}$$

$$\begin{aligned}\mathrm{d/dt[phosphor\text{-}protein\,in\,MAPKK\,layer]} = &-k_3(\mathrm{MAPKK})[\mathrm{MAPKKK}(0)]\\&* [\mathrm{MAPKK}]/(k_\mathrm{m}(\mathrm{MAPKK}) + [\mathrm{MAPKK}])\\&- \int [\mathrm{MAPK}]\mathrm{dt},\end{aligned}$$

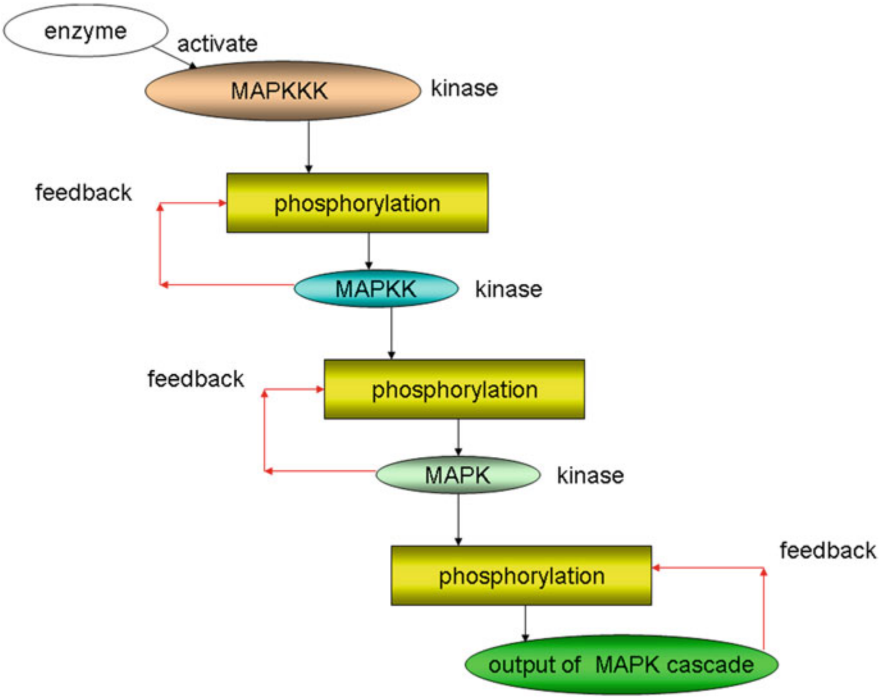


Fig. 4.5 Three-layered MAPK cascade with feedback

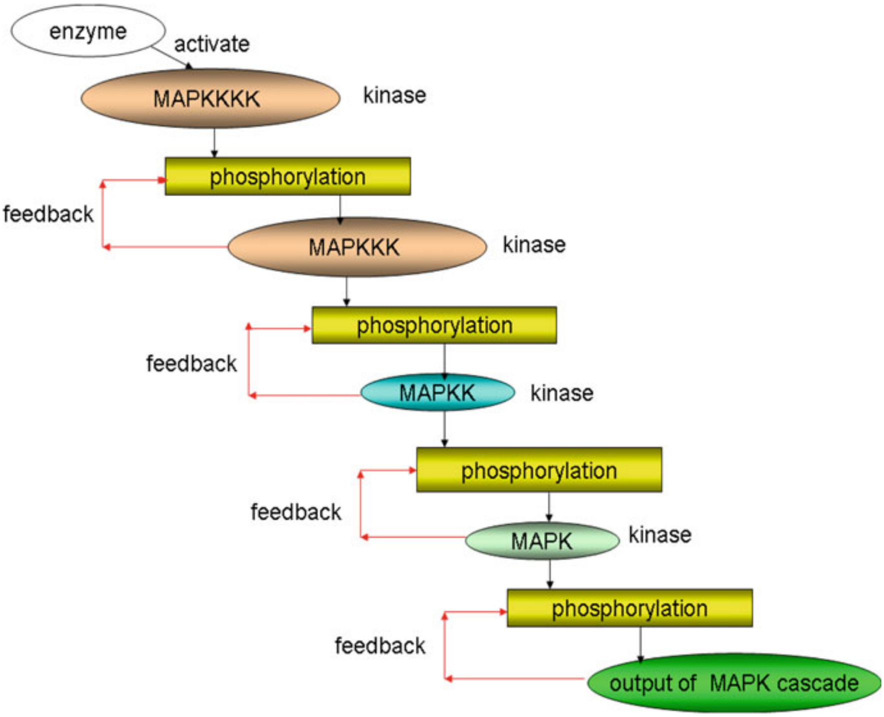


Fig. 4.6 Four-layered MAPK cascade with feedback

$$\begin{aligned}\mathrm{d/dt[phosphor\text{-}protein\ in\ MAPK\ layer]} = {} & -k_3(\mathrm{MAPK})[\mathrm{MAPKK}(0)] \\ & * [\mathrm{MAPK}]/(k_\mathrm{m}(\mathrm{MAPK}) + [\mathrm{MAPK}]) \\ & - \int [\mathrm{output\text{-}of\text{-}MAPK\text{-}cascade}]\mathrm{dt},\end{aligned}$$

$$\begin{aligned}\mathrm{d/dt[MAPKKK]} = {} & k_3(\mathrm{MAPKKK})[\mathrm{MAPKKKK}(0)] \\ & * [\mathrm{MAPKKK}]/(k_\mathrm{m}(\mathrm{MAPKKK}) + [\mathrm{MAPKKK}]),\end{aligned}$$

$$\begin{aligned}\mathrm{d/dt[MAPKK]} = {} & k_3(\mathrm{MAPKK})[\mathrm{MAPKKK}(0)] \\ & * [\mathrm{MAPKK}]/(k_\mathrm{m}(\mathrm{MAPKK}) + [\mathrm{MAPKK}]),\end{aligned}$$

$$\mathrm{d/dt[MAPK]} = k_3(\mathrm{MAPK})[\mathrm{MAPKK}(0)] * [\mathrm{MAPK}]/(k_\mathrm{m}(\mathrm{MAPK}) + [\mathrm{MAPK}]),$$

$$\begin{aligned}\mathrm{d/dt[output\text{-}of\text{-}MAPKcascade]} = {} & k_3(\mathrm{output\text{-}of\text{-}MAPKcascade})[\mathrm{MAPKK}(0)] \\ & * [\mathrm{output\text{-}of\text{-}MAPKcascade}]/(k_\mathrm{m}(\mathrm{output\text{-}of\text{-}MAPKcascade}) \\ & + [\mathrm{output\text{-}of\text{-}MAPKcascade}]),\end{aligned}$$

where the integral part is calculated from initial time 0 to the current time t.

4.3.2.2 Simulation

Because the essential process of intracellular communication exhibits nonlinear dynamics behaviors in a biochemistry framework in the model above, it is possible to figure out what kind of parameters of Michaelis–Menten kinetics behind the fixed-point phenomena is used by the molecular mechanism of signal transduction networks through the MAPK cascade.

The conditions are set as follows:

- The initial concentration of the enzyme = 0.45.
- In the MAPKKKK, MAPKKK, MAPKK, MAPK layer, the product acts as the kinase for the succeeding MAPKKK, MAPKK, MAPK layers and output of the entire MAPK cascade at its bottom.
- In each layer of MAPK cascade $k_{\mathrm{m}} = 0.1$ and $k_3 = 0.01$.
- The initial concentration of substrates (phospho-proteins) in all the pathway-like units is set as 0.45.
- The step/sample number is 10.
- The initial concentration of product is set as 0.001.

Now follows a simulation of a four-layered MAPK cascade.

Let $y = f(x)$ denote the signaling process from MAPK denoted as x to protein of the entire MAPK cascade y; we observed that

$$f(0.0001030) = 0.0001030.$$

where $x = y = 0.0001030$.

This is a fixed-point-like phenomenon. The value 0.0001030 determines the crucial point of phase transition between the monotonic decreasing mode and the fixed-point-like mode.

Column 4: What is a Fixed-Point Phenomenon?

A nonlinear system is associated with a number of curious phenomena. We will investigate one of these. Here, we focus on the *fixed-point phenomenon*, which demonstrates nonlinear behavior in a signaling pathway network, as discussed in Ref. [9] of the main text.

A fixed point is also known as an invariant point. If we denote the input by x and the output by y, then we can express the system as the function $y = f(x)$. The fixed point is defined mathematically in the equation given below:

$$x = f(x)$$

In this phenomenon, input x and output y of the system have equal values ($y = x$).

Owing to dynamic non-linear behavior in the basic processes of biochemical reactions in intercellular communication, the molecular signaling mechanism of the MAPK cascade network is described by the Michaelis–Menten kinetics. From the simulation for the above-mentioned pathway, the fixed point phenomenon is observed under the condition of certain parameters selected empirically.

Here, we set the following conditions:

$$\text{Initial enzyme concentration} = 0.45$$

The products in the MAPKKKK, MAPKKK, MAPKK, and MAPK layers function as kinase for subsequent MAPKKK, MAPKK, MAPK layers, and for the final output of the entire MAPK cascade.

In each layer of the MAPK cascade, $k_{\mathrm{m}} = 0.1$ and $k_3 = 0.01$

In all pathway-type units, the initial concentration of the substrate (phospho-protein) is set to 0.45.

The number of steps/samples is set to 10.

The initial concentration of the product is set to 0.001.

Under the above conditions, we perform a four-layer MAPK cascade simulation. Formulating the signaling process from the MAPK (denoted by x) to the protein (denoted by y) throughout the MAPK cascade by $y = f(x)$, we obtain the result shown below:

$$f(0.0001030) = 0.0001030$$

Here, $x = y = 0.0001030$. This cascade exhibits the behavior similar to the fixed point. The value 0.0001030 determines an important point in the phase transition between a monotonically decreasing mode and a mode similar to the fixed point.

According to the Michaelis–Menten kinetics, the representation for the product is given as follows.

$$\mathrm{d}/\mathrm{d}t(\text{product}(t)) = k_3 * \text{enzyme} * \text{substrate}(t)/(\text{substrate}(t) + k_m)$$

Here, *product*(t) is the concentration of the product at time t; *substrate*(t) is the concentration of the substrate at time t; *enzyme* is the concentration of the enzyme; the parameter setting for k_1, k_2, and k_3 has influence on the stable and unstable states of the corresponding pathway network in simulation.

4.3.3 Robustness

At the system level of a pathway network, dynamics features of pathway networks are the key to understand the cellular signaling mechanism. One of the most important dynamics features of pathway networks is robustness. The robustness of a pathway network can be investigated through different means: for example, stability analysis is an efficient one in the case of the Mos-p MAPK cascade pathway, where Mos-p is a kinase/protein that is in the phosphorylation state, whereas Mos is the same protein in the dephosphorylation state.

The cell has high robustness against external disturbances. As Kitano [10] points out, cancer is an example of robustness in the cell. In contrast with the robustness of pathway networks in the cell, the communication network is very fragile: for example, the Achelles' heel phenomenon in internet networks occurs when failures occur. This contrast motivates us to quantitatively describe the biological robustness [11] of pathway networks to investigate the possibility of applying the knowledge of the biological robustness in pathway networks to the design of information networks in the future.

In the previous sections, we discussed the graphic structure that is usual in computer science. It is obvious that the robustness of cellular signaling processes described in terms of nonlinear dynamics is tightly connected with the "dynamical" graphic structure of pathway networks. In general, the parameters of the differential equations that describe the biochemical reactions of pathway networks can be defined as labels that correspond to the graph, where the vertices and edges of the graph are defined for the pathway network in previous section. These topics belong to the field called *Dynamical Networks*, which refers to the integration of nonlinear dynamics and graph theory. In this section, we define nonlinear dynamics systems in a matrix formulation that corresponds to the graph of the pathway network.

Based on these schemes, we formulate the robustness mechanism of cellular signaling pathway networks.

4.3.3.1 Basic Concepts

The concept of robustness is defined as follows.

Definition of Robustness. Robustness refers to a mechanism that can guarantee and realize the state transition of a (usually dynamic) system from vulnerable and unstable states to sustainable (stable) states when the system suffers from disturbance that is outside the environment and that is unexpected in most cases [12].

Let us define a nonlinear system W for describing the cellular signaling mechanism in cells:

$$W(X, S, U, Y, Q)$$

where X is the input to the system, Y the output of the system, S the state of the system, U the feedback, and Q parameter vector.

The system description is given as follows:

$$\mathrm{d}/\mathrm{d}t(X) = AX + BS$$

$$Y = CS$$

where A, B, are C are matrices.

All the above-mentioned variables are vectors.

We focus on the definition of robustness in terms of the system state. Then, we define the robustness by the function $G(\cdot)$ satisfying the condition that

$$S = G(S, E)$$

when the system suffered disturbance E.

The robustness feature of pathway is expressed in several aspects of pathway network. The stability is one among them. The above-mentioned formulation makes it possible to use the states of the system for describing the robustness where the robustness mechanism is interpreted as the mechanism for providing the steady states. The Mos-p MAPK pathway [13] is an example explaining/analyzing the pathway stability for the robustness of the corresponding pathway network. In this pathway, the term "stable steady states" denoted as SS is used to describe the state transition of the pathway network within the certain domain [13]. The Mos-p MAPK pathway includes a MAPK cascade. As Huang et al. [14] reported, the nonlinear dynamics feature of different phosphorylation processes in a MAPK cascade varies, i.e., the phosphorylation concentration versus time curve is different for each layer of a MAPK cascade. The significance of this phenomenon is obvious if we reveal the fixed point of the MAPK cascade presented in the previous section.

4.3.3.2 Stability Analysis: From an Example of the Mos-p MAPK Pathway for Explanation of Stability

The Mos-p MAPK pathway that demonstrates the transition between the stable/unstable steady states in cells is reported in [13]. In order to make the formulation of the corresponding model, we need the *Hill coefficient* and the *Hill equation*.

The Hill equation is given as

$$\delta = ([L]^n)/(K_\mathrm{d} + [L]^n) = [L]^n/\left(K_\mathrm{A}^n + [L]^n\right),$$

where δ is the concentration of the phosphorylated protein, $[L]$ the ligand concentration, K_d the equilibrium dissociation constant, K_A is the ligand concentration occupying half of the binding sites, and n is the Hill coefficient describing the cooperativity of binding.

The cooperativity means the degree of the biochemical reactions for binding the substrate and enzyme. Here, ligand means an enzyme protein that can bind with another molecule. In Mos-p MAPK pathway, Mos-p is the ligand.

The coefficient n mentioned above is normally denoted as n_Hill; $n_\mathrm{Hill} = 1$ refers to the case of Michaelis–Menten kinetics, which corresponds to the case of completely independent binding, regardless of how many additional ligands are already bound, $n_\mathrm{Hill} \geq 1$ shows the case of positive cooperativity and $n_\mathrm{Hill} \leq 1$ shows the case of negative cooperativity.

The Hill coefficient describes the nonlinear degree of the product's response to the ligand.

By this way, the SS state can be quantitatively described. When the SS state is achieved, the activation degree (concentration) of Mos-p can be formulated as a fixed point of the following form:

$$\text{Mos-p} = f(\text{Mos-p}),$$

where $f(\cdot)$ refers to the Mos-p MAPK cascade pathway.

This shows the steady state of Mos-p, which is consistent with the phenomena reported by Ferrell et al. [13].

Based on the ultrasensitivity quantization, we have a closer look at the Mos-p MAPK pathway from two viewpoints.

A Brief Look at the System. We consider the input of the system as the proteins Mos-P and malE-Mos, and the output as the protein MAPK. Here, two signals concerning MAPK are involved—activated MAPK and phosphorylated MAPK (phos·MAPK for short).

Based on the results from the experiment reported in [13], we can use the Michaelis–Menten equation to obtain that the response of Mek is monotonic to Mos-P, to malE-Mos, or to Mos-P and malE-Mos. In addition, the response of activate·MAPK or phos·MAPK is monotonic to the systems' inputs Mos-P or to malE-Mos or to Mos-P and malE-Mos, under the condition that there is no feedback in the pathway networks, which are branched at the routes from Mos-P and malE-Mos to Mek. The phosphorylation effect can be witnessed at Mek, MAPK, and Mos-P.

A graphic description for this pathway network is given in Fig. 4.7.

Modeling the Temporal Dynamics of the System. Considering the robustness again, we realize that we need the feedback (Cf. Fig. 4.7) and related nonlinear cellular signaling mechanism to help us understand the robustness within this pathway network.

Let us use the Hill-coefficient-based formula to describe the temporal behavior of cellular signaling. Assuming the time series of malE-Mos as

$$\mathrm{d}/\mathrm{d}t[\text{moleE-Mos}] = f_1([\text{molE-Mos}],\ \text{coefficients})$$

where $f_1(\ )$ takes the form of a monotonic function with a decreasing order,

we obtain

$$[\text{moleE-Mos}](n+1) - [\text{molE-Mos}](n) = \text{coefficient}[\text{molE-Mos}](t)$$

This is the time difference equation used for numerical calculation.

Let m denote the step, where m is an integer, and the [molE-Mos] is set as 0.

Then, we obtain

$$\begin{aligned}[\text{active MAPK tot}](m+1) = {} & ([\text{Mos-P}](m) + 1000)^H \\ & / \left(\text{EC50}^{\text{H}} + ([\text{Mos-P}](m) + 1000)^{\text{H}}\right)\end{aligned}$$

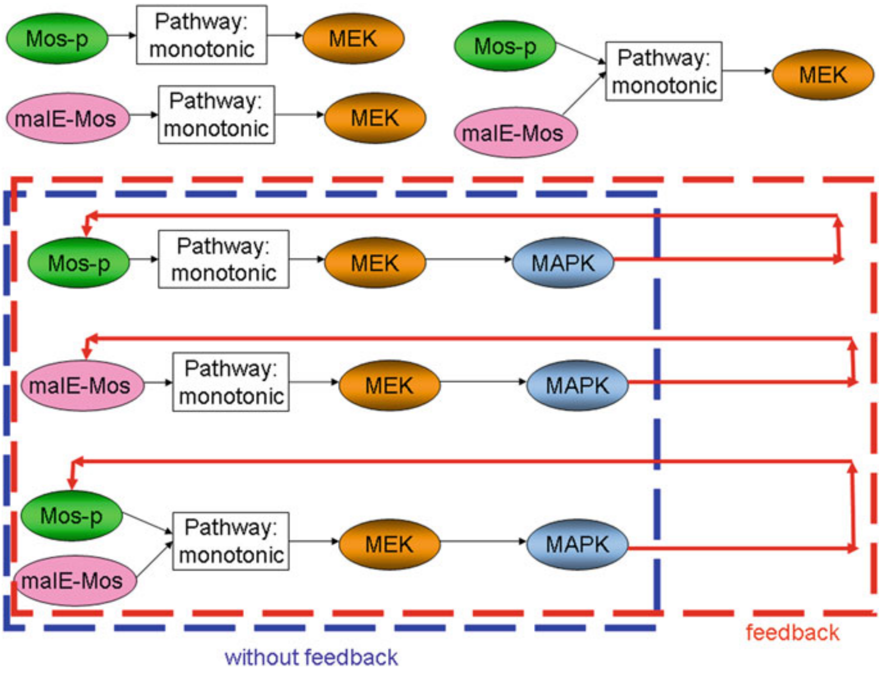


Fig. 4.7 Cascade without feedback and with feedback

where H refers to the Hill coefficient $n_{\rm H} = 5$, EC50 = 20 (nM), [molE-Mos]($t = 0$) is set as 1,000.

Then

$$\begin{aligned}[\text{Mos-P}](m+2) &= 0.5 * [\text{active MAPK}](m+1) \\ &= 0.5 * [\text{MAPK(tot)}]([\text{Mos-P}](m) + 1000)^{\rm H} \\ &\quad /\left(\text{EC50}^{\rm H} + ([\text{Mos-P}](m) + 1000)^{\rm H}\right)\end{aligned}$$

Now assume that [Mos-P](m + 3) = [Mos-P](m + 2) + [Mos-P](m). Let the moment $n + 1$ correspond to $m + 3$, n correspond to m when we set the time reference of Mos-P according to the initial time of going through the pathway and the final time. The moments of $m + 1$ and $m + 2$ refer to the internal states of the pathway network as a system. Consequently, we have

$$\begin{aligned}[\text{Mos-P}](n+1) = 0.5 * ([\text{Mos-P}](n) + 1000)^{\rm H} \\ /\left(\text{EC50}^{\rm H} + ([\text{Mos-P}](n) + 1000)^{\rm H}\right) \\ + [\text{Mos-P}](n)\end{aligned}$$

The Mos-P MAPK pathway derived computing process then becomes

$$[x(n+1)] - [x(n)] = 0.5 * ([x(n)] + 1000)^5 \Big/ \left(20^5 + ([x(n)] + 1000)^5\right) - [x(n)]$$

where x is an integer ≥ 0 and $[x(0)]$ is set as 5.

$$\text{Let}\, f((x(t)) = 0.5 * (x(t) + 1000)^5 \Big/ \left(20^5 + (x(t) + 1000)^5\right) - x(t)$$

$$\mathrm{d}/\mathrm{d}t(x(t)) = x(t+1) - x(t) = f((x(t))$$

We define a function in general:

$$\mathrm{d}/\mathrm{d}t(x(t)) = f((x(t))$$

when [molE-Mos] is treated as a constant.

It is obvious that the above system is still nonlinear even though the control input [molE-Mos] is a constant.

Considering the dynamics feature of control input [molE-Mos], we can have that

$$\mathrm{d}/\mathrm{d}t(x(t)) = 0.5 * (x(t) + u(t))^5 \Big/ \left(20^5 + (x(t) + u(t))^5\right) - x(t)$$

where x(t) = Mos-p that refers to the state of the system, and u(t) = [molE-Mos] that refers to the control input.

Since this is a coupled system, it is necessary to decouple the different signals to efficiently control the system.

The description of the pathway network as a system is normally established by a differential equation. Denote the system by the following equations:

$$\mathrm{d}[X(t)]/\mathrm{d}t = f(t, X(t), U(t))$$

$$X(t) = g(t, X(t))$$

where t is time, *U(t)* is the input, *X(t)* is the state, and *Y(t)* is the output.

This is a general form that includes the case of nonlinear systems.

Based on the instance of Mos-p MAPK pathway we discussed before, the cellular pathway network is modeled as a controller-centered system where feedback is embedded. Different constraints can be used to formulate specific objects in applications, e.g., a matrix-based representation could be

$$\mathrm{d}X(t)/\mathrm{d}t = A(t)X(t) + B(t)U(t)$$

$$Y(t) = C(t)X(t)$$

where *A(t)*, *B(t)*, and *C(t)* are matrices.

In this instance, the variable is one-dimensional, and a nonlinear relation exists between state transitions. Therefore, we have that

$$\mathrm{d}/\mathrm{d}t(x(t)) = f(x(t), u(t))$$

where *x(t)* = [Mos-p]—this is the state; *u(t)* = [molE-Mos]—this is the control input; this equation is established under the steady states of the Mos-p MAPK pathway.

The output for detecting the signals of phosphorylation is given as follows:

$$y(t) = \mathrm{phos} \cdot [\mathrm{MAPK}] = g((x(t), u(t))$$

where

$$\begin{aligned} g((x(t), u(t)) = 0.5 * (x(t) + u(t))^3 \\ / \left(20^5 + (x(t) + u(t))^3\right) \end{aligned}$$

under the condition that the Mos-p MAPK pathway is with the steady state.

From the above discussion of pathway systems, we have presented a dynamics-based representation for formulating the robustness of a dynamic system, which is motivated by the biological robustness in cellular pathway networks, but which can be modified and extended to any abstract dynamic system where feedback is embedded.

4.4 Error-Correcting Codes for Cellular Signaling Pathways

How can we carry out reliable information processing by pathway networks with dynamics features? An important element underlying cellular signaling is the robustness of molecular pathways. Mechanisms such as those resembling error-correcting codes may play an essential role in this framework.

Column 5: Error-Correcting Code

Claude Shannon's information theory is important to study efficient information processing models and corresponding information processing algorithms based on signal encoding for telecommunications. Today, they serve as a framework for applying information theory to various fields, including those outside electronic media, like the fields discussed in the main text. Here, we discuss information theory with respect to the information observed at the level of biomolecules. Molecular signals, of course, can be represented by a chemical form and studied using experimental data such as concentrations. However, in formulating these signals by mathematical

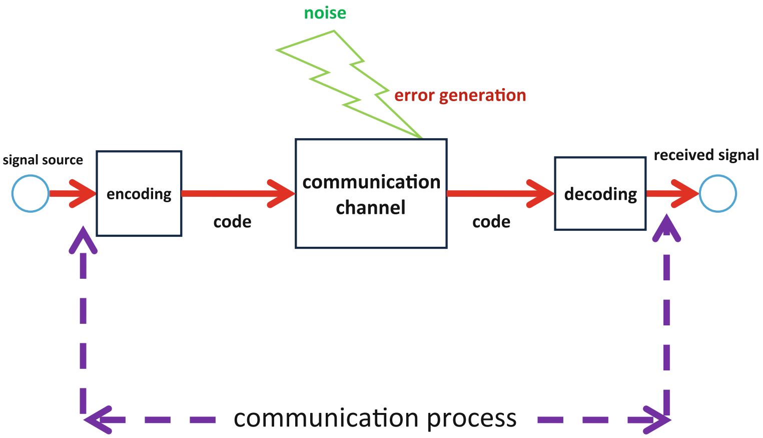


Fig. 4.8 Conceptual framework of the communication process in information theory

methods, we find, to our surprise, that such phenomena can be addressed by Shannon's information theory (see Fig. 4.8 in this column).

In communication systems, the signal source provides the input (to the encoding unit), while the output (of the decoding unit) is delivered to the receiver. Input and output correspond to transmitter and receiver, respectively.

The path connecting the transmitter and the receiver is defined as a communication channel. Analysis is easy for an ideal system free of noise. However, we cannot completely eliminate noise, since the thermal motion of molecules causes certain kind of noise. Signals at the receiver are subject to losses compared to the original signals transmitted. Many techniques have been developed to minimize such losses. With respect to the technology to improve signal-to-noise ratios (SNRs), the directly related topic discussed in this chapter is error correction.

SNR is defined by the expression below.

$$\mathrm{SNR} = 20 \cdot \log_{10}[\text{amplitude of signal/amplitude of noise}]\ [\mathrm{dB}]$$

SNR can be improved by increasing the amount of the received signal, reducing noise, or improving SNR by balancing both aspects.

A specific example of codes will help us understand this concept. From the perspective of information theory, codes can be classified as either of two types: source code and channel code. The former contributes directly to efficient information processing. The latter is designed to account for the noise encountered in the communication channel. In studies of molecular communications, codes designed for phosphorylation and dephosphorylation correspond to source code. Molecular codes with error-correcting capacity based on pathway mechanisms are proper to be considered as channel code. Codes used to recover signals containing errors caused by noise are defined by error-correcting codes. The codes studied in the field of information communication include the following:

Turbo code
Low-density parity-check code (LDPC)

In bioinformatics on cells, noise and fluctuations on biomolecules can be formulated using error-correcting codes under the condition of changes of the molecular concentration in the environment. For example, LDPC for a pathway network is useful in constructing a theory of molecular communication. The cell signaling pathway network is essential for implementing error correcting codes at the molecular level in molecular communications.

4.4.1 Molecular Coding from Molecular Communication

A reversible molecular switch is the basis of information representation in cellular informatics. Two kinds of reversible molecular switches exist in cells.

The molecular switch of phosphorylation and dephosphorylation The phosphorylation state of a signaling protein is defined as 1, whereas the dephosphorylation state of a signaling protein is defined as 0. The phosphorylation process is regulated by kinase, while the dephosphorylation process is regulated by phosphatase.

The switch of GTP-bound and GDP-bound states As shown in Fig. 4.9, the GTP-bound state of GTPase set by the so-called GEF and GEF pathway is defined

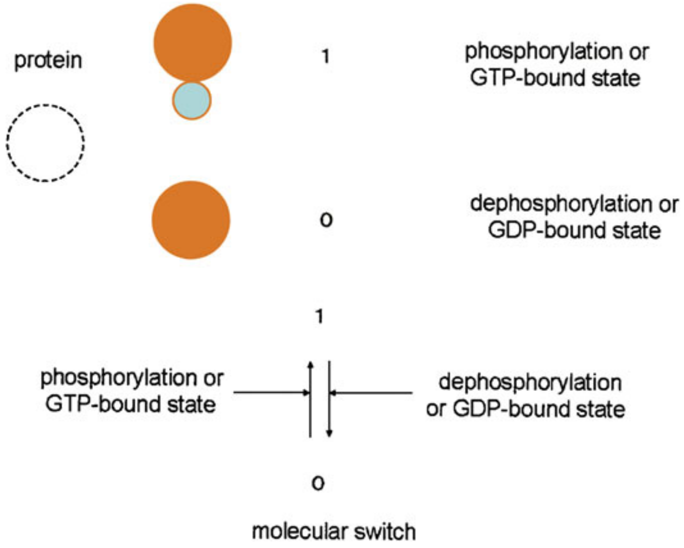

protein
1
phosphorylation or GTP-bound state
0
dephosphorylation or GDP-bound state
1
phosphorylation or GTP-bound state
dephosphorylation or GDP-bound state
0
molecular switch

Fig. 4.9 Molecular switch for binary codes (bits)

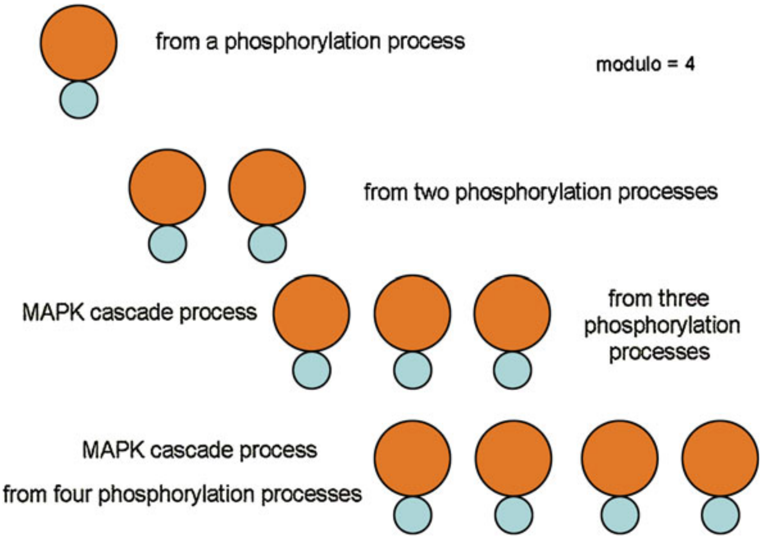

from a phosphorylation process
modulo = 4
from two phosphorylation processes
MAPK cascade process
from three phosphorylation processes
MAPK cascade process
from four phosphorylation processes

Fig. 4.10 Four bits generated from a MAPK cascade

as 1 and the GDP-bound state of GTPase set by the so-called GAP and GAP pathway is defined as 0.

The MAPK cascade consists of several phosphorylation processes. Therefore. multiple binary codes can be generated, e.g., four bits generated by four-layered MAPK cascade (see Fig. 4.10).

The above molecular switches allow us to formulate a mathematical model for information processing based on an abstraction of the data structure corresponding to the signaling pathways.

The above framework only involves switches, and it does not include the redundancy typically associated with error-correcting codes. If we intend to include such codes, then the resulting mechanisms should be compatible with the mechanisms encountered in cell communication.

4.4.2 LDPC Coding for Pathways

Error-correcting codes are mathematical constructs, and their design involves a different philosophy from the way in which biological mechanisms have evolved, which includes an element of randomness. Fortunately, there is an error-correcting code, of which the design—though mathematically well founded—involves a random element as well. Called *Low Density Parity Code (LDPC)*, this code is, surprisingly, very efficient, in the sense that it allows transmission of information at rates very close to the theoretical maximum. These codes were developed in 1960 by Gallager, and they had been long forgotten due to the perception of them

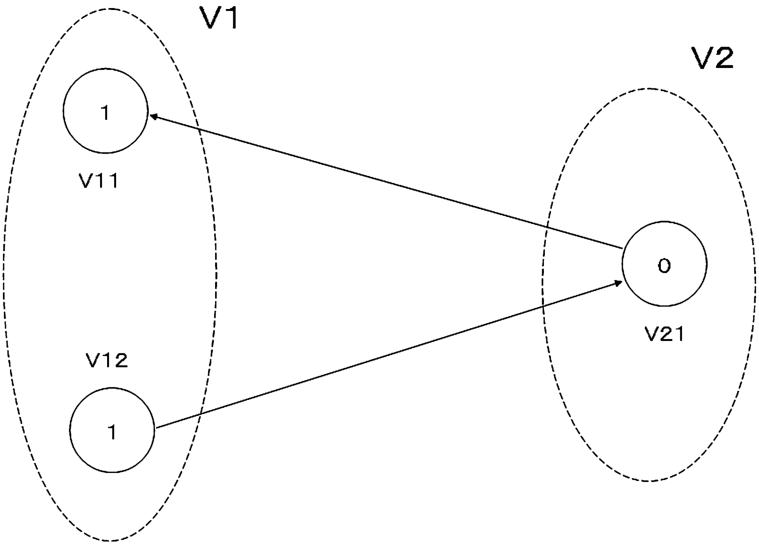


Fig. 4.11 A simple example of an LDPC encoding

being impractical, only to be rediscovered by MacKay in 1996. Ironically, the randomness in LDPC codes is very attractive in the framework of biologic systems, and this is the reason why we describe them in more detail.

The LDPC code is defined by a partite graph BG = $\langle V, E\rangle$, where the vertex set $V = V_1 \cup V_2$. V_1 is an ordered set of nodes, each of which is labeled by a binary number. The set V_1 thus denotes a binary code word. V_2 is a set of which each node is labeled by a binary number equaling the parity value of the sum of the labels of the nodes in V_1 that are connected to the node in V_2 by an edge in E.

As shown in Fig. 4.11, at first, let V_{12} be 1 and V_{21} be 0, so that the parity summation should be 0, and V_{11} should accordingly be assigned the value 1. This is the encoding scheme. The decoding is carried out in a reverse way, as shown in Fig. 4.12.

Assume that V_{11} is lost during information transmission, then to restore the value of V_{11}, we need to use the information of V_{12} and V_{21}. Because V_{12} is 1 and the parity summation is 0, we can infer that V_{11} should be 1.

Figure 4.13 gives the encoding process of 5 bits in V_1 (labeled as L_1), where the known bits are put into the condition part of an “IF THEN” rule and the unknown bits are put into the conclusion part of this rule.

The nodes in V_1 are called L_1-units and the ones in V_2 are called L_2-units, owing to the fact that they are described by two separated vertex sets in a bipartite graph.

Within the signal transduction model followed in this chapter, the two bipartite parts associated with the generation of an LDPC can be designed as shown in Fig. 4.14, where the information processing units corresponding to the phosphorylation/dephosphorylation pathway and GEF/GAP pathway are denoted as L_1-unit and L_2-unit, respectively.

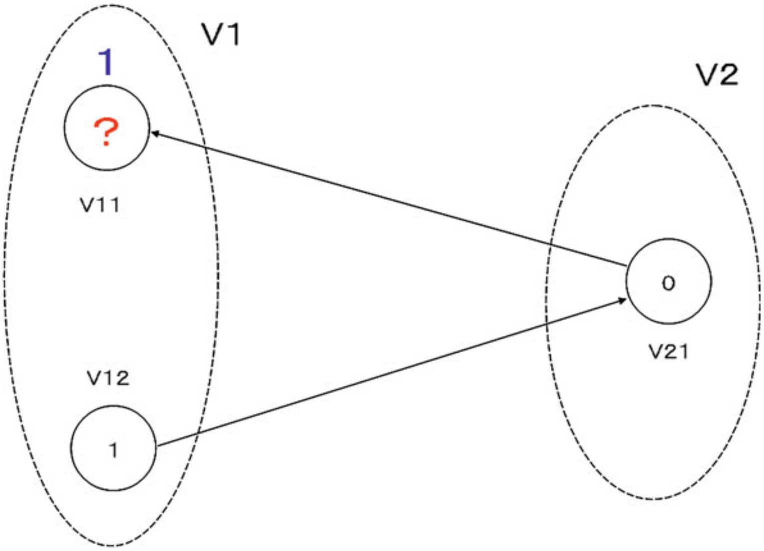


Fig. 4.12 A simple example of an LDPC decoding

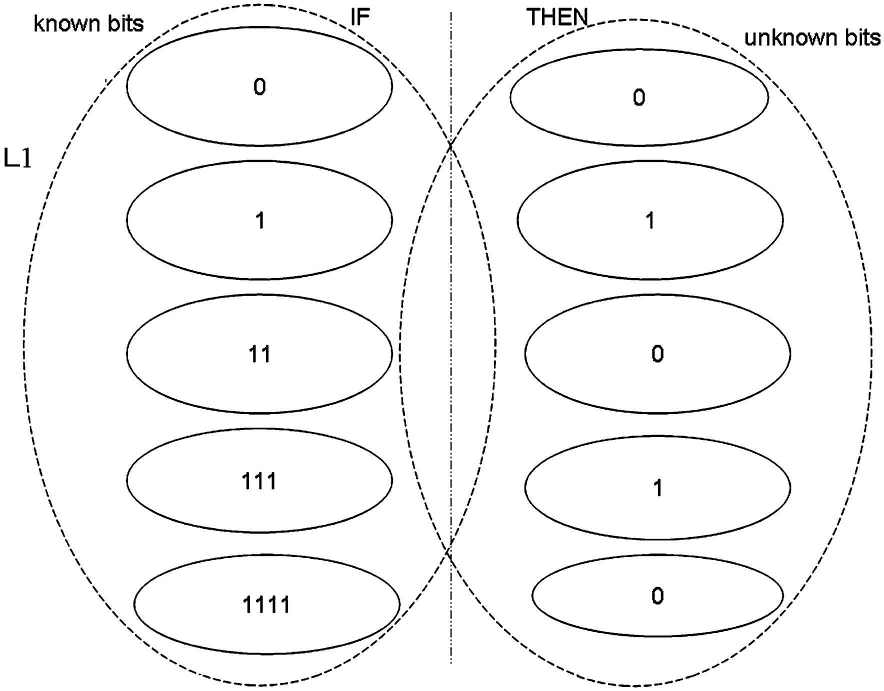


Fig. 4.13 The rules for generating words

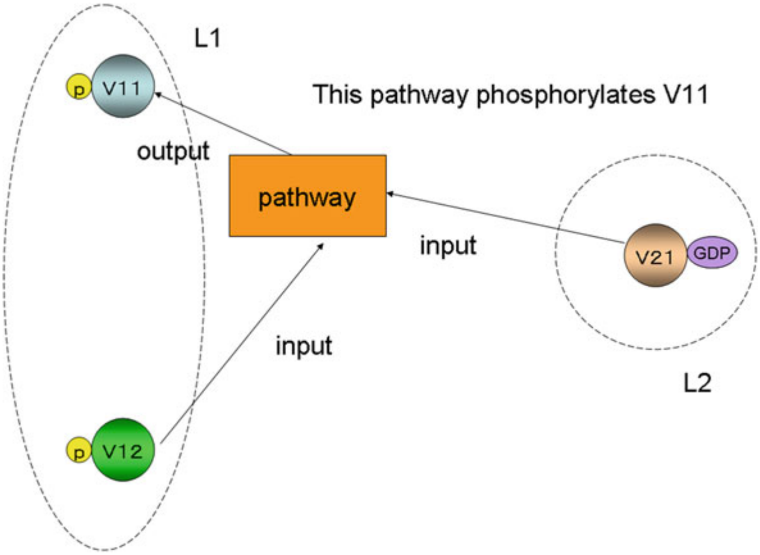


Fig. 4.14 An example of a pathway that is mapped to the model in Fig. 4.12

As shown in the above figure, the bipartite graph to encode an LDPC code is mapped from a pathway whose input is V_{12} when in the phosphorylation state and V_{21} when in the GTP-bound state and whose output is $_{11}$ when in the phosphorylation state. This pathway corresponds to the graph in Figs. 4.11 and 4.12 for encoding and decoding an LDPC code.

The above model allows us to encode phosphorylation pathways and the dephosphorylation pathways in terms of an LDPC code, which is capable of approaching the Shannon limit. The derived encoding/decoding model is bidirectional, symmetrical and "implicitly binary" (i.e., its binary form can be formulated in terms of n bits of the L_1-units of the model), which differs from the previous model given in [15] that is one-directional, asymmetrical (from phosphorylation/dephosphorylation to GTP-bound/GDP-bound states) and "explicitly binary" (i.e., the phosphorylation/dephosphorylation mechanism directly encodes the code z).

Column 6: Low-Density Parity-Check Code (LDPC)

LDPC is an abbreviation for *low-density parity-check code*. First proposed by Robert G. Gallager in the 1960s, LDPC did not get practical application as a communication technology for quite some time. With advances in computing technologies, they now serve as an important error-correcting encoding method alongside turbo code. For example, this approach has

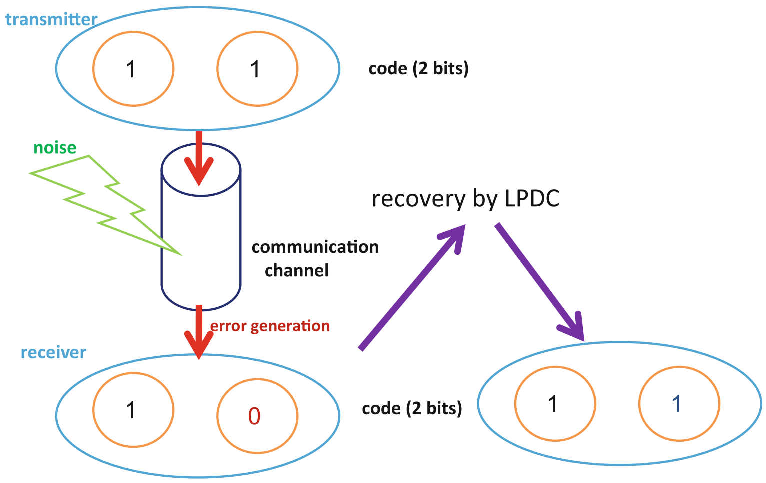


Fig. 4.15 Encoding by LDPC

served as the standard error-correcting method for the satellite communication of digital television since 2003. Figure 4.15 in this column illustrates the basic principles of LDPC.

We begin with a two-bit code and encode it by LDPC rules. (These rules are relatively simple.) The number of zeros in the code is not an important factor. However, the number of bits containing 1 must be even. According to the concept, the total number of all bits containing 1 in the code must be even.

As in Fig. 4.15, assume that the transmitter sends the code “11” and that noise in the communication channel changes this to “10” by the time, it reaches the receiver side. The transmitted code is shown below.

$$\text{Code C1} = \text{“11”} \rightarrow \text{The number of bits containing “1” is even}(2).$$

The received code is as given below.

$$\text{Code C2} = \text{“10”} \rightarrow \text{The number of bits containing “1” is odd}(1), \text{indicating an error}.$$

To correct the code, LPDC changes the “0” in the code received to “1.”

4.5 Summary

In this chapter, we have described pathways in biological organisms that behave de-facto like communication systems on cellular scales. The robustness of biological communication systems offers an important lesson for the design of man-made communication systems: key concepts in this context are parallelism, adaptability, and structural stability. We have also described LDPC codes, which have much in common—due to their randomness—with the unstructured character of biological systems. This may suggest that LDPC codes form an important avenue of research in the realization of communication systems that include the above key concepts. Cellular signal transduction networks form a rich source of inspiration in the design of next-generation communication systems, which will have greater robustness, greater capacity, and greater adaptability, but which will also be less visible to its users, forming a transparent ever-present environment, such as the *overlay-network* in [16].

References

1. A.L. Barabasi, E. Bonabeau, Scale-free networks. Sci. Am. **288**(5), 60–69 (2003)
2. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan et al., Network motifs: simple building blocks of complex networks. Science **296**, 910–913 (2002)
3. H. Kitano, T. Azuma, Systems biology and control. ISCIE J. Syst. Control Inf. **48**, 104–111 (2004). (in Japanese)
4. J. Leonardi, J.A. Box, J.T. Bunch, P. Baumann, TER1, the RNA subunit of fission yeast telomerase. Nat. Struct. Mol. Biol. **15**, 26–33 (2008). (Published online: 23 December 2007)
5. B. Alberts et al., *The Molecular Biology of the Cell*, 4th edn. (Garland Science, New York, 2002)
6. U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits* (Chapman Hill/CRC, Boca Raton, 2006)
7. KEGG: Kyoto Encyclopedia of Genes and Genomes. http://www.kegg.jp/
8. B.N. Kholodenko, Cell signaling dynamics in time space. Nat. Rev. Mol. Cell Biol. **7**, 165–176 (2006)
9. J.-Q. Liu, On quantitative aspect of an information processing model inspired by signaling pathways in cells: an empirical study. IPSJ SIG Technical Report, 2007-MPS-43, pp. 21–24 (2007)
10. H. Kitano, Cancer as a robust system: implications for anticancer therapy. Nat. Rev. Cancer **4**, 227–235 (2004)
11. H. Kitano, Biological robustness. Nat. Rev. Genet. **5**, 826–837 (2004)
12. J.-Q. Liu, K. Leibnitz, Modeling the dynamics of cellular signaling for communication networks, in *Bio-Inspired Computing and Communication Networks*, ed. by Y. Xiao, F. Hu (Auerbach Publications/CRC Press, Boca Raton, in press, 457–480
13. J.E. Ferrell Jr., E.M. Machleder, The biochemical basis of an all-or-none cell fate switch in Xenopus oocytes. Science **280**, 895–898 (1998)
14. C.-Y.F. Huang, J.E. Ferrell Jr., Ultrasensitivity in the mitogen-activated protein kinase cascade. Proc. Natl Acad. Sci. **93**, 10078–10083 (1996)
15. J.-Q. Liu, K. Shimohara, *Biomolecular Computation for Bionanotechnology* (Artech House, Boston/London, 2007)
16. Z. Li, P. Mohapatra, QRON: QoS-aware routing in overlay networks. IEEE J. Sel. Areas Commun. **22**, 29–40 (2004)

Chapter 5
For Deeper Understanding

Hidefumi Sawai

Abstract In this book, we have explained so far the importance of application of mechanism in life for designing the advanced ICT system by referring to the representative algorithms and information processing models. In this last chapter, we would like to briefly introduce the thoughts of Pascal, Darwin, and Einstein who have inspired us to aim at and study about nature and life. Science has been developed by many ancestors who were inspired by nature and life. We will also be able to explore a hint for new ideas from their thoughts. As an example peculiar to the real world, we take up the trends in the latest complex networks and the application to social networks.

Keywords Paradigm shift · Scientific and technological revolution · Hierarchy in nature and society · Complex networks

5.1 Paradigm Shifts in Scientific and Technological Revolution

> *Possibly we shall know a little more than we do now. But the real nature of things, that we shall never know, never.*
>
> —Albert Einstein

H. Sawai (✉)
Kobe Research Laboratories, National Institute of Information and Communications Technology, 588-2, Iwaoka, Nishi-ku, Kobe, 651-2492, Japan
e-mail: sawai@nict.go.jp

H. Sawai (ed.), *Biological Functions for Information and Communication Technologies,* Studies in Computational Intelligence, 320,
DOI: 10.1007/978-3-642-15102-6_5, © Springer-Verlag Berlin Heidelberg 2011

Let us examine how scientific revolutions can be viewed as paradigm shifts, drawing on Thomas Kuhn's *The Structure of Scientific Revolutions* [1]. According to Kuhn's definition, "paradigms" are "universally recognized scientific achievements that for a time provide model problems and solutions to a community of practitioners." As a prime example of a "scientific revolution" he gives the discovery of the general theory of relativity by Albert Einstein to explain the nature of space and time—a theory that completely overturned the concept of absolute space and time presented by Newton [2]. Newtonian dynamics failed to explain Mercury's perihelion movement (precession), the gravitational red-shift of the light spectrum, or the gravitational diffraction of light, all phenomena successfully and quantitatively explained for the first time by *Einstein's general theory of relativity*. This example illustrates how scientific revolutions involve a shift from an old paradigm to a new one. But what does the creation of a new "inspired by life" *paradigm* mean? To understand this, we must first learn from the *theory of Darwin*, the father of evolution.

Column 1: Hierarchy in Nature and Society and Pascal's "Pensées" [3, 4]

Figure 5.1 shows the hierarchy in nature and human society. "Cleopatra's nose: had it been shorter, the whole face of the world would have been changed." This famous remark is by Blaise Pascal (1623–1662, see Fig. 5.2)

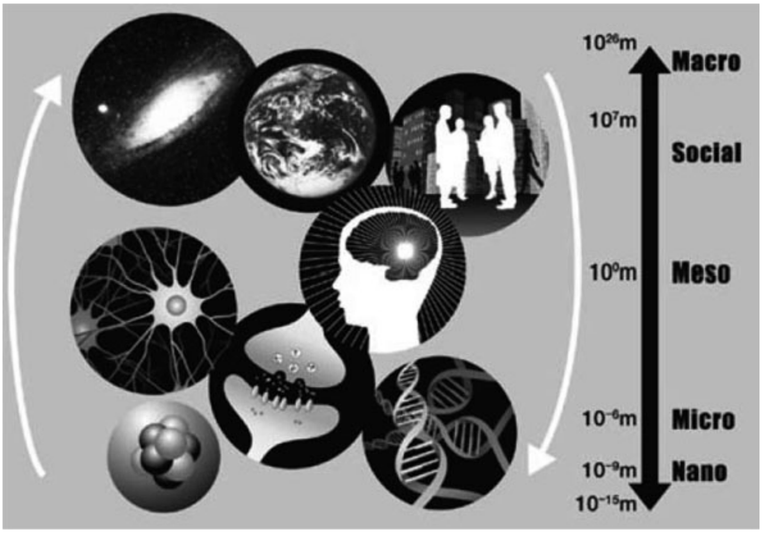


Fig. 5.1 Hierarchy in nature and society

Fig. 5.2 Portrait of Pascal (Photo: Popperfoto/Getty Images/AFLO)

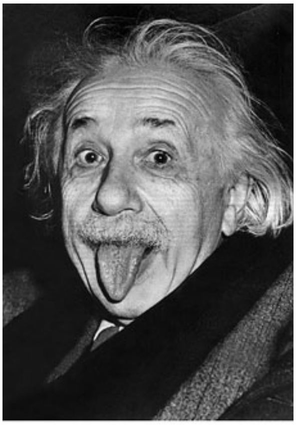
from his "Pensées." Here, let us quote aphorisms related to this hierarchy, from "Pensées." (The quoted text is given according to the interpretation of the author.)

> Man is but a reed, the most feeble thing in nature; but he is a thinking reed. … But, if the universe were to crush him, man would still be more noble than that which killed him, because he knows that he dies and the advantage which the universe has over him; the universe knows nothing of this.

As shown in Fig. 5.1, the natural world is structured hierarchically, from the minute 10^{-15} m (femtometer) scale of elementary particles to the 10^{-9} m (nanometer) scale of atoms and molecules, the 10^{-6} m (micrometer) scale of living cells, the 10^{0} m (=1) scale of human proportions (in other words, human beings are mesoscopic beings), the 10^{3} m (kilometer) to 10^{6} m (megameter) scale of human populations on the Earth, and the 10^{26} m scale of the size of the universe. Viewing the world from the human scale, considering that the constituents of the infinitely large universe are the elementary particles, we understand that this hierarchy we see in nature reflects a mode of observation unique to human beings. "For in fact what is man in nature? A Nothing in comparison with the Infinite, an All in comparison with the Nothing, a mean between nothing and everything. Since he is infinitely removed from comprehending the extremes, the end of things and their beginning are hopelessly hidden from him in an impenetrable secret," and "The eternal silence of these infinite spaces frightens me."

Column 2: Einstein and the Theory of Relativity [2, 5–9]

Albert Einstein (1879–1955, Refer to Fig. 5.3) was unable to find a job for a period after graduating from the Swiss Federal Institute of Technology Zurich. Later, while officially employed as an officer at the Federal Office for



1879-1955

Fig. 5.3 Portrait of Einstein (Photo: UPI = Kyodo)

Intellectual Property, he did work on the special theory of relativity and published it as an article in the German Academic Journal, *Annalen der Physik*, in 1905, when he was 26 years old. In the same year, he published five Nobel-class papers in rapid succession, including the light quantum hypothesis and the theory of Brownian motion. This year was later referred to as the "Annus Mirabilis" (miraculous year) in the history of physics.

In 1922 (*Taisho* 11), Einstein was on his way to Japan on board the Kitano-maru of Nippon Yusen Kaisha for a lecture tour, invited by a publisher by the name of Kaizosha, when he received news that he had won the Nobel Prize in physics. Reports indicate his excitement. After landing at Kobe harbor on November 17 of the same year, he gave lectures on the theory of relativity at universities and research institutions over a course of 43 days in various parts of Japan. Although he had already published work on the general theory of relativity by 1916 for the world's appraisal, the Nobel Prize recognized not this work, but his discovery of photoelectric effects. Max Planck (awarded the Nobel Prize in Physics in 1918), a pioneering figure in quantum theory who was familiar with both Einstein and his work, is said to have remarked: "Even if Einstein had not written a single line about the theory of relativity, he still would have undoubtedly been noted as one of the greatest physicists of the twentieth century." Numerous ceremonies and events were held in 2005 to mark the 100th anniversary of the birth of the theory of relativity.

Derived from the "principle of the constancy of the speed of light" and the "special principle of relativity," the special theory of relativity expresses the relationship between physical quantities (such as position, time, velocity, and mass) in a stationary system and in an inertial system moving at constant velocity, leading to the famous equation $E = mc^2$ linking energy and mass. The general theory of relativity expands the special theory of relativity to accelerated systems (fields with gravity). It is derived from two principles: the equivalence principle, which states that inertial mass and gravitational mass are the same; and the principle of general covariance, which states that a law of physics must take the same form in any coordinate system (including accelerated systems). The equation is expressed as follows.

$$G^{\mu\nu} = R^{\mu\nu} - 1/2g^{\mu\nu}R = \kappa T^{\mu\nu} \quad (\mu, \nu = 0, 1, 2, 3)$$

Here, $G^{\mu\nu}$ is the Einstein tensor, $g^{\mu\nu}$ is the metric tensor, $R^{\mu\nu}$ is the Ricci tensor, $T^{\mu\nu}$ is the energy-momentum tensor, $\kappa = 8\pi G/c^4$ is the Einstein's constant (where G is the Newton's gravitational constant and c is the speed of light), and μ, $\nu = 0$, 1, 2, 3 corresponds to the dimension of time (t) and space (x, y, z), in the sequence given.

This equation also plays an important role in modern cosmology. Solving this equation makes it possible to predict the movement of Mercury's perihelion, the deflection of light by gravity (the gravitational lensing effect), gravitational redshifts, and the existence of black holes and gravitational waves (tensor waves). "The value of a theory lies in its ability to make predictions," said Richard Feynman, awarded the 1965 Nobel Prize in Physics with Julian Schwinger and Shinichiro Tomonaga. The everyday applications of the Global Positioning System (GPS) used for car navigation and in mobile phones, the time and spatial calibrations for which are based on the general theory of relativity, suggest the scale of this theory's success.

Here, we briefly summarize several episodes related to Einstein.

After being awarded the Nobel Prize in Physics, Einstein remarked on the praise from Germans, Swiss, and Italians; "The Germans say that it honors all Germans, the Swiss say that I am an honorable Swiss citizen, and the Italians say that I am Italian because I lived in Italy when I was a child. However, if the theory of relativity should turn out to be wrong, the Italians would call me a Swiss citizen, the Swiss would call me a German, and the Germans would call me a Jew."

Meeting by chance on a busy thoroughfare in New York, Charlie Chaplin, the king of comedy, and Einstein had the following exchange as they were surrounded by a crowd of people.

Einstein: "Mr. Chaplin, your movies are so popular because anyone in the world can easily understand the wonderful stories!"

Chaplin: "My goodness, Mr. Einstein, I can't beat you. Few people in the world can understand your theory of relativity, but you are so popular!"

Episode when Bernard Shaw (awarded the Nobel Prize in Literature in 1925), the Irish writer and ironist, introduced Einstein at a ceremony and compared him to Newton.

"Now, I would like to introduce Dr. Einstein, a genius physicist of the twentieth century. He is a great scientist who pointed out mistakes in the Newtonian dynamics, which have been believed true for over 200 years. However, I am not sure how long his theory will remain true."

Einstein: "…" (wry grin)

Regarding his views of nature and the world, his doubts regarding quantum mechanics are well known, as expressed in the phrase: "God does not play dice."

When asked what he thought of death, he replied, "One can no longer listen to Mozart," a beautiful remark indicating his love of Mozart's music.

5.1.1 Darwin's Theory of Evolution Still Surviving Today

> *The most incomprehensible thing about the world is that it is at all comprehensible.*
>
> —Albert Einstein

The year of 2009 is a memorial year of the 200th anniversary since Darwin's birth in 1809 as well as the 150th anniversary since the first issue of "The Origin of Species" published in 1859. The keywords in his theory of evolution are "Natural selection" and "Survival of the fittest" [10, 11].

Richard Leakey, an evolutionary biologist, claims that Darwin was able to present a mechanism that convincingly explained how changes occur in species—namely, Natural Selection. Darwin first conceived the idea of natural selection in 1838, inspired by *An Essay on the Principle of Population* by Thomas Robert Malthus, a parson and socio-economist of the early nineteenth century. While Malthus was primarily interested in population, his essay points out that the number of offspring produced by an organism is greater than the number expected to live to reproductive age, identifying this as a general principle in nature [11].

Based on Darwin's theory of evolution, John Holland in the 1960s advanced the idea of genetic algorithms (GA) as a computational model [12]. Genetic algorithms and their evolutionary forms (e.g., genetic programming and evolutionary computation) were discussed in greater detail in Chap. 1.

Column 3: Darwin's Theory of Evolution and Motoo Kimura's Neutral Theory of Evolution

Published November 24, 1859, *The Origin of Species* [10] was a contemporary bestseller. The 250 copies of the initial edition are said to have sold

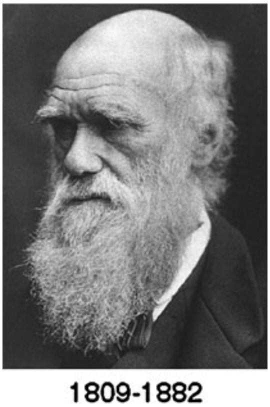


Fig. 5.4 Portrait of Darwin (Photo: Kyodo News)

out the day it was published. Generations of people have continued to read it. It is a long-term bestseller that has significantly influenced human thought and philosophy (Fig. 5.4).

The terms "natural selection" and "survival of the fittest" refer to the concept that the individuals of a given population (species) best-suited to a given natural environment survive, while others disappear. Here, the species is not invariant, but changes (evolves) in response to the environment. Darwin conceived of the idea of the theory of natural selection while on board the *Beagle* on a voyage around the world, including the Galapagos Islands, during a 5-year period from December 1831 to October 1836. (*The Voyage of the Beagle*) [14]

Darwin had the following thoughts regarding the driving force and mechanisms that change species: Even organisms belonging to the same species are not identical in form, but have slight individual differences. These differences include characteristics resulting from genetic mutations as well as acquired characteristics; the former are passed on from parents to offspring. Organisms seek to produce as many offspring as possible, which perpetuates the species. However, limits on food and resources within the natural environment result in a struggle for these resources and for survival, wherein individuals with characteristics advantageous to survival survive and pass on their traits, while others do not. Darwin owed the germ of this idea to Thomas Malthus's "An Essay on the Principles of Population," [15] wherein Malthus observed that the capacity to produce food for human consumption increased arithmetically, while populations grew geometrically and exceeded the capacity to cultivate food.

In contrast, according to Motoo Kimura's neutral theory of evolution [13], "Mutation and genetic drift at the molecular level are the primary factors driving evolution; change is neither advantageous nor disadvantageous (i.e., is neutral) with respect to natural selection." When first proposed in the late 1960s to the early 1970s, the neutral theory of evolution generated heated discussion regarding potential conflicts with Darwin's theory of natural selection. The current consensus is that the neutral theory of evolution predominates at the level of the genotype (molecular evolution), while natural selection predominates at the level of the phenotype (changes in the characteristics of individuals).

5.1.2 3.8 Billion Years' Stream of Life

> *The distinction between past, present, and future is only a stubbornly persistent illusion.*
>
> —Albert Einstein

In 1837, Darwin realized that the evolutionary genealogy between organisms could be more clearly represented using a tree diagram. (Figure 5.5 is a schematic drawing of a phylogenetic tree.)

Over 10 million rich and diverse species populate the globe today, all descended from a single primordial cell existing some 4 billion years ago. Advances in fossil survey and DNA analysis techniques have made it possible to put the time at which humans diverged from orangutans at some 13 million years ago; from gorillas at some 6.5 million years ago; and from chimpanzees at some 5 million years ago. Gradual accumulations of small differences in genotype created by mutations eventually result in significant differences in phenotype, in due course leading to brains and intelligent organisms capable of adapting to their environment. The term

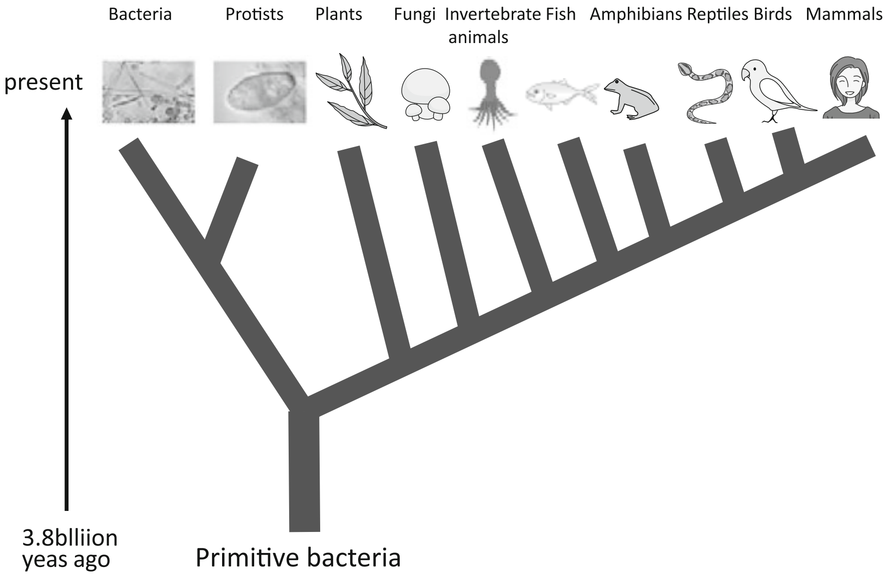


Fig. 5.5 Schematic diagram of a phylogenetic tree

"intelligence" as used here will refer both to the adaptive functions of an organism that affect its survival and the exquisite functions emergent from the brain.

Column 4: Phylogenetic Tree and Molecular Clocks

As Fig. 5.6 shows, we know that a common ancestor species arising some 3.8 billion years ago branched into the Archaea, Bacteria, and Eukarya, and that the Eukarya evolved into animals, plants, and fungi (molds and mushrooms). Somewhat to our surprise, Darwin drew just such a phylogenetic tree in his personal notebook [16], long before genes had been discovered. As shown in Fig. 5.6 of the main text, over 10 million species populate the globe today, including Bacteria, Protoctista, plants, fungi, invertebrates, fish, amphibians, reptiles, birds, and mammals, creating a rich diversity of living forms. We can estimate at what point in time these species diverged during the process of evolution based on the so-called molecular clock (the rate of base substitution in the intron, believed to be near-constant), based on Motoo Kimura's neutral theory of evolution.

Column 5: Hypothesis for Mechanism Underlying Inspiration (the Aha! Experience)

Let us describe how human inspiration—or the aha! sensation—works, using the example of a degraded image (Fig. 5.7 in this column). I doubt this picture would make much sense to anyone seeing it for the first time. This is, in fact, a

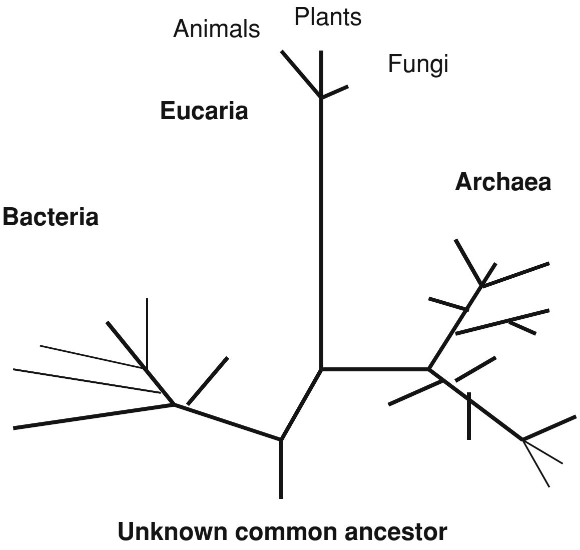
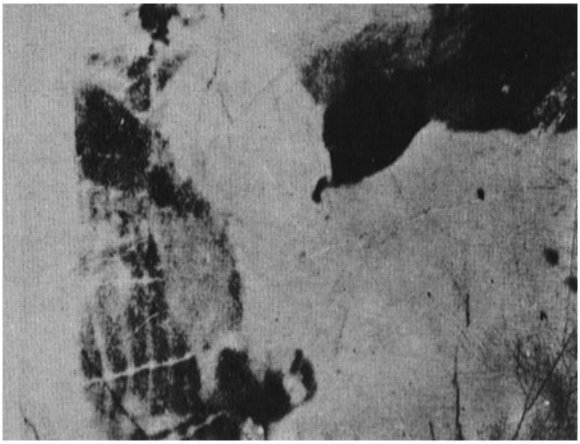


Fig. 5.6 Phylogenetic tree of life on earth

Fig. 5.7 Long-known example of degraded image (Dallenbach’s cow) (Reproduced from [51])



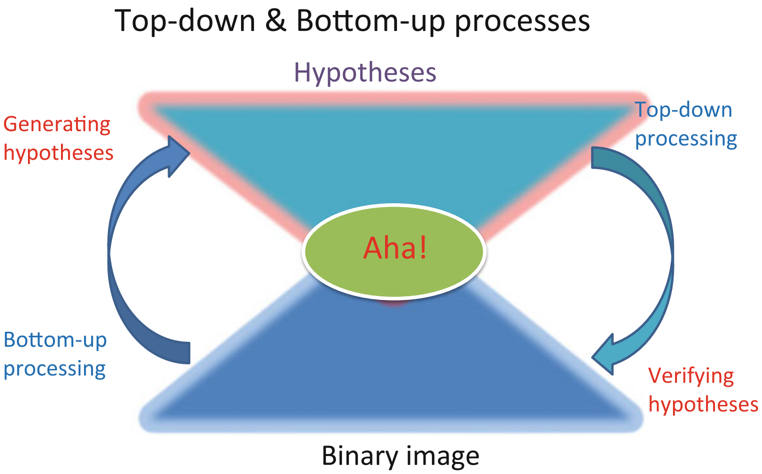


Fig. 5.8 The mechanism underlying inspiration (schematic diagram)

monochromatic image of a cow viewed from the front. The eyes, ears, and nose are painted black. Once we know this, we recognize the entire image of the cow: We recognize the outline of the face of the cow, which we had not seen before, and the outline on the right-hand side emerges naturally, though it is not drawn in a manner that makes the representation explicit. In short, once we know this is a picture of a cow, it cannot look like anything but a cow.

Now, consider what goes on in the brain up to the moment of inspiration (aha! experience). Figure 5.8 is a schematic diagram of the mechanism of inspiration; Fig. 5.9 gives the corresponding flowchart.

When we first see the image in Fig. 5.7, we think: What is this? Rifling through our memories, we then create various hypotheses. This is the bottom-up hypothesis-generating process illustrated in Fig. 5.9(1). We then try

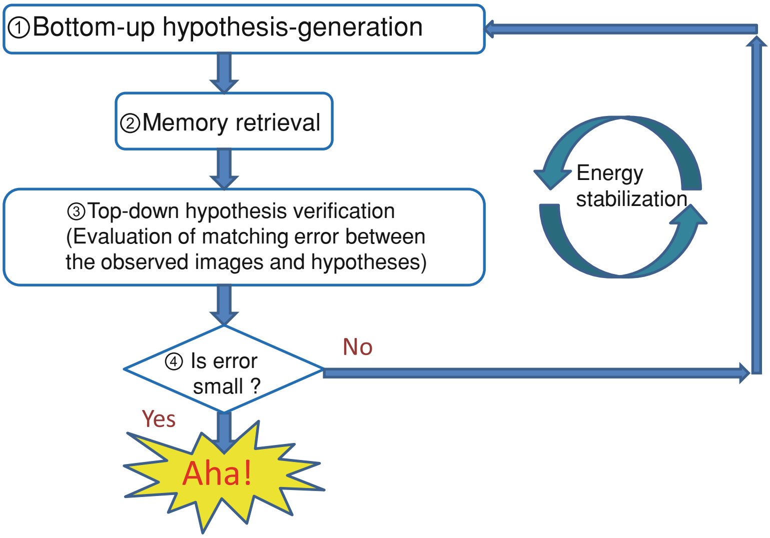

①Bottom-up hypothesis-generation
②Memory retrieval
③Top-down hypothesis verification (Evaluation of matching error between the observed images and hypotheses)
④ Is error small ?
No
Yes
Aha!
Energy stabilization

Fig. 5.9 The mechanism underlying inspiration (flowchart)

to understand it by searching past memories in the storage area in the brain. This is the memory retrieval process shown in Fig. 5.9(2). In the memory retrieval process, we recollect and trace various memory patterns accumulated in the brain (in other words, we chaotically progress toward an attractor corresponding to the memory patterns [17, 18]) and pattern-match one retrieved image after another stored in the brain against the degraded input image shown in Fig. 5.7. This is the top-down processing of hypothesis verification shown in Fig. 5.9(3). Here, the brain evaluates the matching error between the two images. If this error is small in the judgment made at Step (4), inspiration strikes: "Aha! A cow!" If the matching fails and the error is large, we generate new hypotheses via bottom-up processing based on memory retrieval, repeating the series of processes from Steps (1) to (4). Eventually, the brain comes across a suitable memorized image, establishes the match against the observed image of Fig. 5.7 (i.e., the matching error becomes suitably small), the energy condition in the brain stabilizes, and we experience inspiration (the aha! sensation).

5.2 Solution by Complex Networks Toward the Problems in the Real World

Man is, at one and the same time, a solitary being and a social being.

—Albert Einstein

The information and communication systems permeating today's society grow in *sophistication* and *complexity* seemingly day by day, while assuming greater roles in our lives at the same apparent pace. But the same information and communication systems have been shown to be fragile under conditions such as accidents and disasters. Developing technologies that will overcome such *fragility* has become essential. To resolve these problems, by focusing on network dynamics, we examined basic technologies for securing self-organized functional network structures that do not rely on an infrastructure and remain operational even in dynamically changing environments. Our goal is to propose a proto-model for a next-generation information and communications system that is highly reliable and has high affinity for human needs and modes of interaction. Here, we examined research trends in complex network sciences and statistical physics, with a special focus on networks with *self-repairing* functions and requiring only localized information for adaptive evolution, and on the design principles of *self-organizing* networks that feature node deployment and load distribution functions according to population distribution and economic activity levels and network traffic management functions designed to resolve traffic congestion. In addition, we also examined the spontaneous social network structures created by relationships between real people and considered a network society desirable in the near future.

5.2.1 What are Complex Networks? [19–36]

First, we will explain the "small-world phenomenon," "scale-free network," "network analysis indicators," and "general trends in research" on complex network science.

5.2.1.1 Phenomena, Structures, Laws, and Characteristics in Complex Networks

- *Small-world phenomenon* [21, 22]

A network phenomenon emerging from the rewiring of a few edges in a regular network; one in which the average path length (described later) is significantly reduced. This is known in common parlance as "six-degrees of separation." Observations of the phenomenon date from experiments performed in the U.S. by the social psychologist Stanley Milgram. From left to right in Fig. 5.10 are a regular network, a small-world network, and a random network. ρ represents network rewiring probability; higher ρ values indicate higher randomness. The figure shows how randomness increases from regular networks to small-world networks and finally to random networks.

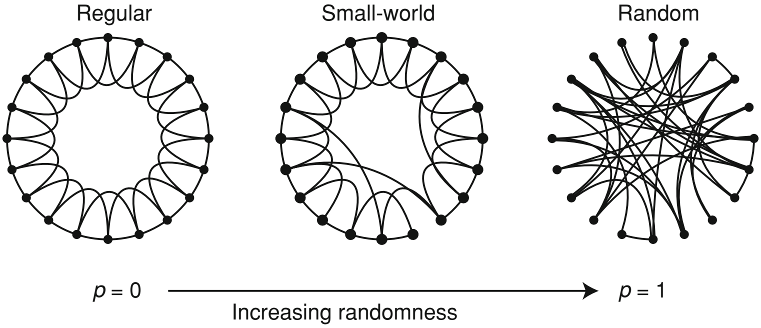
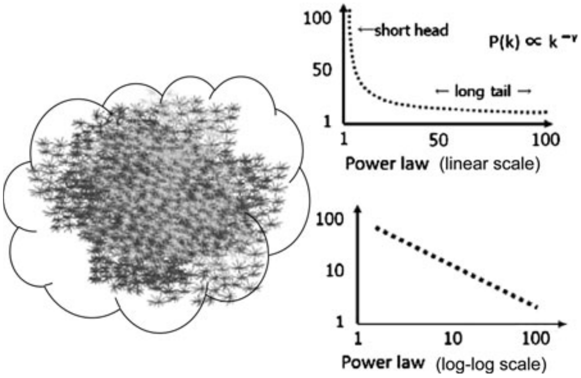


Fig. 5.10 Small-world network (Reproduced from [52])

Fig. 5.11 Scale-free network (for Internet)



- *Scale-free network*

A network whose relationship between the degree (k_i) of a node *i* in the network and degree distribution $P(k_i)$ follows the *power law* $P(k_i) \propto k_i^{-\gamma}$. Figure 5.11 shows the Internet in a scale-free condition, with relatively small numbers of hubs having nodes with large degrees, and with numerous nodes having small degrees. The upper right-hand panel in the figure shows the degree and degree distribution, displaying what is known as the long-tail phenomenon (resembling the *long tail* of a dinosaur). The lower right-hand panel shows the same data as the upper right-hand panel, but in a log–log plot. The inclination of the linear line here is equal to $-\gamma$. Various scale-free networks in nature and society have their own unique values.

- *Pareto's law*

A law proposed by the Italian economist Vilfredo Pareto, popularly known as the "80:20 rule." It states that the significant 20% of the element dominates 80% of the whole. It is deeply associated with the relationship between degree $\mathrm{k_i}$ and degree distribution $P(k_i)$ in the above scale-free network. Namely, a few hubs (corresponding to the significant 20%) play an important role as the function of the whole network, dominating the other sub-networks (corresponding to 80%) connected to the hubs.

- *Robustness*

Adaptability and/or fault tolerance of life or systems to environmental change.

- *Fragility, vulnerability*

The opposite of robustness; the fragility of life or systems (weakness) in the face of environmental change.

In *Persistent Life* by Hiroaki Kitano and Kaoru Takeuchi (Diamond Inc., 2007), the trade-off between robustness and fragility is explained by taking as examples the 2003 New York City blackout, the beef bowl business strategy of Yoshinoya, and diseases such as diabetes and cancer [35].

5.2.1.2 Network Analysis Indicators

- *Network G*

Set $G = \{V, E\}$ consisting of the set of nodes $V = \{v_1, v_2, \ldots, v_n\}$ and set of edges $E = \{e_1, e_2, \ldots, e_m\}$.

- *Degree*

Degree k_i of node v_i refers to the number of edges branching out of node v_i.

- *Average path length/average distance between nodes L(ρ)*

For two arbitrarily chosen points v_i and v_j, the average of the minimum number of edges that must be crossed to reach each other.

- *Clustering coefficient C(ρ)*

An index that takes a value between 0 and 1, which quantifies the degree of concentration of edges constituting network G.

- *Network centrality*

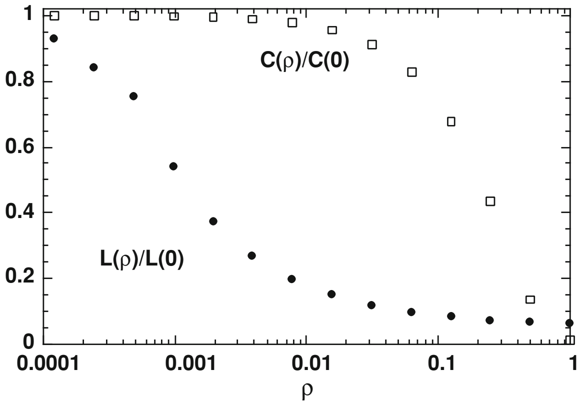
The network property that plays the central role. When nodes having higher degrees are more central, the network is assumed to have "degree centrality." The nodes clustered in the highest concentration along the shortest path between two node pairs are described as having "betweenness centrality." The index was proposed by the sociologist Linton Freeman as an index for social network analysis.

- *Network density*

The higher the number of edges between nodes, the higher the network density.

Figure 5.12 is a graph showing the relationship between the clustering coefficient $C(\rho)$ and average path length $L(\rho)$ (where ρ is the network rewiring rate).

Fig. 5.12 Clustering coefficient $C(\rho)$ and average path length $L(\rho)$ (Reproduced from [52])



For small values of ρ, both the clustering coefficient $C(\rho)$ and the average path length $L(\rho)$ are large. But as ρ increases, they decrease. At intermediate values of ρ (around 0.01), $C(\rho)$ is large while $L(\rho)$ becomes small, at which point the small-world phenomenon emerges.

5.2.1.3 A Network Architecture Resistant to Both Random Failure and Targeted Attack [36–38]

Although the question, "Do such networks actually exist?" comes to mind, according to "What Kind of Strong Wiring Design Would be Resistant to Both Failure and Attack?" in "Semi-Special Edition: Spread of Complex Network Science" by Toshihiro Tanizawa (pp. 282–289, *Information Processing*, March 2008 issue, No. 3, Vol. 49), the answer is a bimodal distribution network architecture in which only two types of nodes are present: a small number of large-degree (k_2) hubs and large numbers of small-degree (k_1, where $k_1 \ll k_2$) nodes.

5.2.1.4 General Trends in Research on Network Science

- In *Network Science* by the U.S. National Research Council (http://www.nap.edu/catalog/11516.html), the research topics are reported to be priority areas.

Based on this table, we find that modeling the simulation of very large networks is considered critical in the development of real-world tool sets and that human performance enhancement is the objective of studies of swarming behavior in the development of self-organizing systems and studies on metabolic and gene expression networks. The selection of biological mechanisms and human swarm behavior as research areas for achieving these research objectives is noteworthy.

Column 6: A Network Architecture that Resists Both Failures and Attacks

The Internet as currently configured has numerous hub nodes with diverse degree k and a scale-free property, as discussed in the main text. For this reason, attacks targeting only a few percent of the hub nodes can damage the links (communication channels) of the entire network and disable communications between nodes (computers). On the other hand, the Internet will survive random failures of individual nodes and links. What network architecture can resist both failures and attacks? The answer is a bimodal distribution network (also discussed in the main text) [37]. Figure 5.13 in this column shows an example of a bimodal distribution network with number of nodes $N = 100$, average degree $\langle k \rangle = 2.1$, $r = 0.1$, and 10 ($r \cdot N = 10$) hubs of degree 12. In this network, most nodes feature only degree $\langle k_1 \rangle$, close to average; however, a small number of hub nodes have a large degree $\langle k_2 \rangle$. For this reason, the network is resistant to random failures. Since it contains few hubs, it can maintain connectivity through the remaining nodes even when subject to targeted attacks. We can compare this to "a network structure that consists of individual actors, each having adequate abilities and collaborating with each other, and a small number of leaders organizing the actors" [36].

5.2.2 Application Fields of Complex Networks

One field of application of Complex Network Science is "congestion studies." *Congestion Studies* by Katsuhiro Nishinari (Shincho Sensho, 2006) [41] presents a

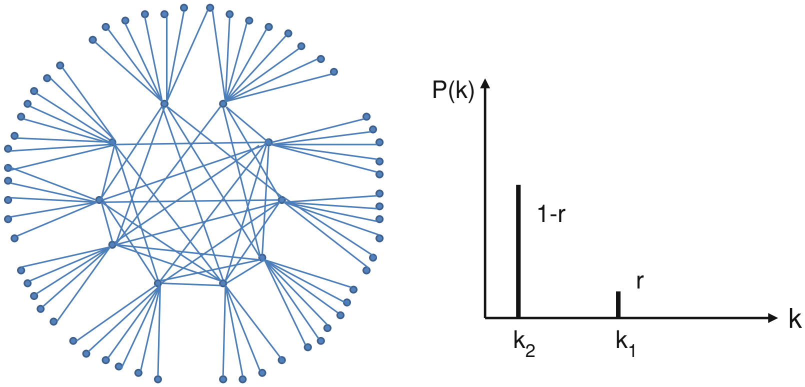


Fig. 5.13 Bimodal distribution network resistant to both failure and attack (reproduced from Fig. 5 of Ref. [37])

unified discussion of the phenomenon of congestion in information transferred over the Internet (packets) and of traffic caused by human and automobile transportation from the perspective of networks. He introduces technology for relieving traffic congestion while explaining how congestion can be actively directed to benefit humans (for example, by preventing the spread of wildfires or viral infections).

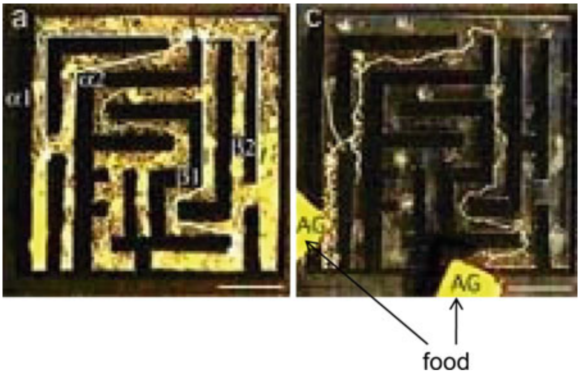
One of the other applications would be the application of a "small-world," as introduced in Column 8. If we could develop a high-speed retrieval and visualizing system that enables to know your acquaintances only through the "six-degrees of separation" after registering all people in the world (about 6.8 billion people in 2010) on a link database, the system would be extremely useful academically and in practice.

Column 7: Application of Slime Mold to Maze Search

Often encountered in gardens, forests, and cultivated fields, slime mold (*Physarum polycephalum*) was studied intensively in the past by naturalist and folklorist Kumagusu Minakata. Slime molds are classified into two types: true slime molds and cellular slime molds. True slime mold is also referred to as Myxomycete. It has certain animal-like features in that the trophozoite (referred to as plasmodium) moves and eats food such as microorganisms, and certain plant-like features in that it forms fruiting bodies and propagates by spores . Plasmodium is a mass of protoplasm that exhibits ameboid movements. It has various interesting properties, such as internal cytoplasmic streaming and reciprocating movements (oscillation). Researchers are currently exploring various applications of these properties to solve combinatorial optimization problems, including the traveling salesman problem (TSP) and the route searching problem for transport networks. In an article in which he describes how slime mold solves a maze, Toshiyuki Nakagaki states, "This remarkable process of cellular computation (of slime mold) implies that cellular materials can show a primitive intelligence" [39]. Recently, slime molds have been applied to the problem of finding optimum (shortest) routes in transport networks such as the Japan Railway Company in the Kanto area of Japan, and the Interstate system (highways connecting states) in the continental United States [40]. Here, the slime mold itself looks for an alternative route if accidents block the first route.

These studies provide important clues for developing functions in the Internet to search for bypasses in a self-organizing manner when links disconnect or node PCs fail. Dr. Nakagaki was awarded an Ig Nobel "Cognitive Science Prize" in 2008, and "Transportation Planning Prize" in 2010 at Harvard University. (See Fig. 5.14)

Fig. 5.14 Slime mold working on maze-solving problem (Reproduced from Fig. 1 of Ref. [39])



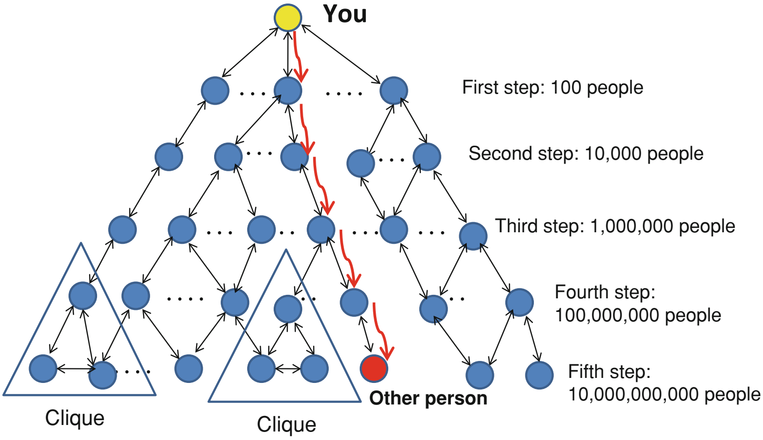
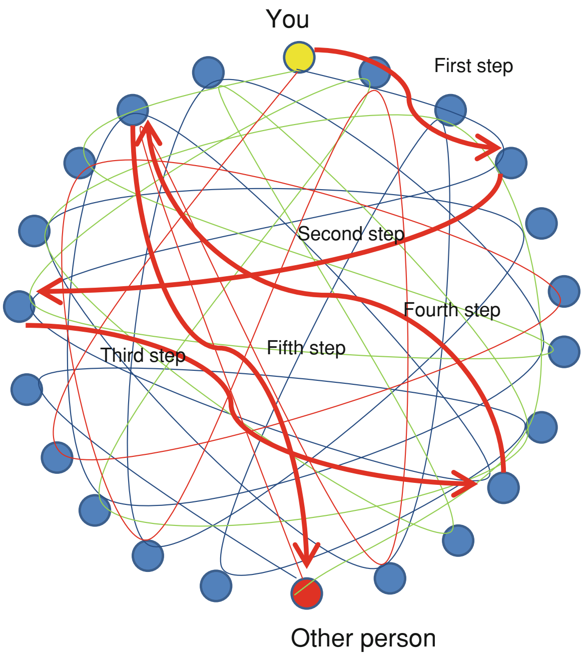


Fig. 5.15 Is it a small world?

Column 8: Applying Complex Networks [19–22, 31]

Is everyone in the world really separated by just six degrees of separation from everyone else? Although the experiment performed by sociologist Stanley Milgram has been subject to various criticisms, including potential biases in friend selection, the phrase "It's a small world!" rings true. Assume you have 100 friends, each of whom has 100 friends, and so forth. Let us track these friends through five subsequent steps. If we assume, as shown in Fig. 5.15 in this column, that friends who independently know others within the group (indicated by two-headed arrows), forming collective groups called cliques, account for 30% of all those in the group (or, conversely, 70% of the individuals in the group do not know each other directly), the number of all friends is $100^5 \times 0.7 = 7$ billion, or approximately equal to the world's current population (6.8 billion as of 2010).

Fig. 5.16 Visualization and the search for a global human network



However, how can we find a network demonstrating this idea of “six degrees of separation”? In principle, if everyone in the world registered 100 of his or her friends to construct a “database of friend links” and if the database were analyzed, we could determine whether the proposition “two people arbitrarily selected from among the world population are linked by six intermediate people, including themselves (we can find a network structure)” is true or false. Can we solve this problem analytically using a random graph? The problem can be stated as follows: “How many (k) average friends for each person and how many (m) steps do we need to link all the people ($N =$ approximately 6.8 billion) in the world?” How can this problem be applied to human networks with “scale-free structures” like the Internet? These problems posit a human network in the real world. As shown in Fig. 5.16, this is the problem of “seeking and visualizing the shortest path” from you to an arbitrary other person. The problem is of both real-world and academic interest.

Other examples of the “small world” phenomenon include the following.

The relationship between the “small world” and fireflies: Why and how do fireflies emit light simultaneously in perfect synchronization?

The network of actors who had appeared in movies with actors who in turn had appeared with other actors in other movies (known as the “Kevin Bacon game”) and the Erdös number, used to characterize the citation network of mathematical papers are also examples of the “small world” phenomenon.

Clarifying the P53 protein, which the tumor suppressor gene (P53 gene) produces, and the topology of its interaction network will contribute to the treatment of cancer.

Other examples of scale-free structures include the relationship between the frequency of occurrence (f) of a word and its ranking within a list of frequency of occurrence (R), which is known to satisfy $f \propto 1/R$ (the first-order power law) (Zipf's law).

5.2.3 Trends of Social Network-Associated Fields [42–47]

In this subsection, we will explain the research trends on social networks and their current applications to the real world. Please refer to the following books introduced later in more detail.

5.2.3.1 Organization Network Strategy

- According to *Global Neighborhoods—Strategies of Successful Organizational Networks* by Toshio Nishiguchi (NTT Publishing Co., Ltd. 2008) [42], a strong and effective human network may be constructed by combining long-distance relationships (=use of small-world by distant but dear acquaintances) and neighborhood socializing (=*rewiring* of human networks). Examples given in the book include the robust human network created overseas by former residents of Wenzhou, China, and the miraculous recovery of Aisin Seiki Co., Ltd., an affiliate subsidiary of Toyota, after a fire (self-organizing human network when faced with emergency), as well as the examples of defense procurement by the British Ministry of Defense and by Japan. The book presents actual examples of several interesting network topologies. The book concludes by summarizing such networks as follows: The secret to successfully managing a complex *social network* that far surpasses human cognitive limits is to establish a neighborhood social network and to weave into it long-distance relationships by some appropriate rewiring at the edges to transform the entire system into a *small-world*, and to take the best parts of both. Supporting this network is the *social capital* that forms the foundation of the relationship of mutual trust and ease with which structural rewiring can be performed.
- *The Science of Building a Network of Human Connections—Probing the Power Hidden in Human–Human Relationships* by Yuki Yasuda (Nikkei Inc., 2004) [46].

"Distant people" are more beneficial. The quality of the links is more important than the number. We turn more often to those we seldom see for support than frequent friends. Using *The Strength of Weak Ties* by Mark Granovetter and the "small world theory" by Duncan J. Watts as the basis of her arguments, she reveals the most efficient method for constructing human connections by network analysis.

- *Network Analysis—What Determines Human Actions* by Yuki Yasuda (Shinyosha, 1997) [47].

Covering human relationships from lovers to organization and international networks, Yasuda searches for the common pattern in human–human and human–society relationships, then explains how network analysis techniques can be used to explore factors that determine human actions.

5.2.3.2 SNS Field

- The current status of social network services such as Mixi, Twitter, etc.

Typical examples of such networks include *MySpace* in the U.S., *Mixi* in Japan, and Twitter in all over the world. These networks have enjoyed widespread success by providing services that let users create blogs, send short messages, and make connections to friends and other community members on the Internet.

5.2.3.3 Information Search

- Google’s business strategy

Based on the hypothesis that “quality sites are sites supported by many other sites”—based on how documents on the WWW are linked together—Google has successfully applied a business strategy that presents the search results for a search term using an algorithm called *PageRank*™.

5.2.3.4 Distribution Field

- Amazon’s business strategy

Amazon has applied the *long-tail phenomenon* (power law = scale-free property) in book marketing, creating a database that lets customers search for and purchase on the Internet, books ranking low on the sales list, and has succeeded in making enormous profits off such books, going against the conventional trend in which best-sellers and long-sellers were the primary source of revenue for booksellers.

Column 9: Hierarchy in Nature and Society and Emergent Networks

In Column 1, discussed earlier, we describe the hierarchies observed in nature and society, grouping them into microscopic, mesoscopic, and macroscopic

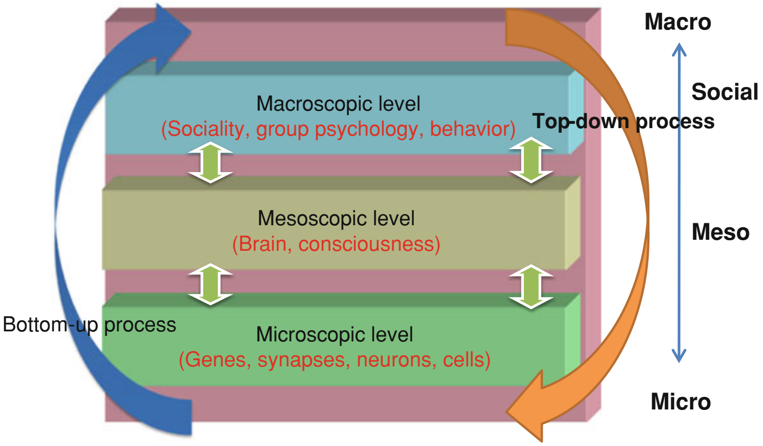


Fig. 5.17 Emergent network dynamics

levels. Here, we describe the relationship between this hierarchy and emergent networks. Originally, as already discussed, this hierarchy reflects the mode of observation of human beings as mesoscopic beings. Elementary particles, atoms, molecules, genes as polymers (DNA), proteins made of amino acid sequences expressing these genes, and cells (including the neurons composed of these proteins) are at the microscopic level. Consciousness, a mesoscopic phenomenon, is believed to emerge from the brain, a gigantic network of neurons and synapses. Individual animals and humans have brains, and both humans and animals interact in populations to form societies. At this macroscopic level, the behavior observed reflects the psychology of a population rather than an individual. Figure 5.17 schematically illustrates the microscopic, mesoscopic, and macroscopic levels of the hierarchy and the interrelationships between these levels. Special interactions exist between the levels of the hierarchy. Column 5 describes the mechanism underlying inspiration as the circulation of the bottom-up process of hypothesis generation and the top-down process of verifying these hypotheses. A similar mechanism is believed to be at work between the different levels of the hierarchy in nature and in society, although the functional scales differ significantly. This is attributable to the fractal (self-similar) nature of nature.

In Column 10, we consider the mechanism of the interactions between the various levels of this hierarchy.

Column 10: Emergent Network Mechanism [48–50]

Figure 5.18 in this column schematically illustrates the "emergent network mechanism" to describe interactions between the levels of a hierarchy in nature and in society.

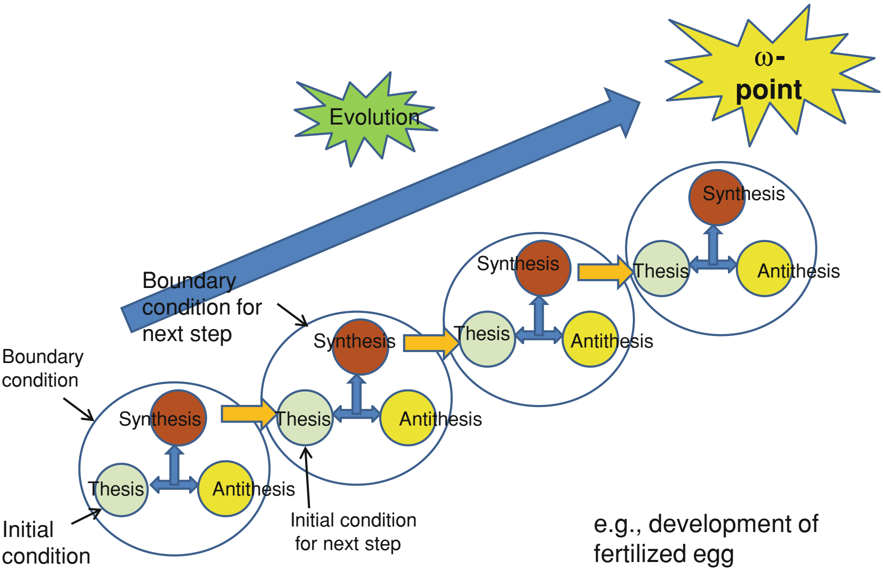


Fig. 5.18 Emergent network mechanism

As the first state of the bottom-up processing, an initial condition (the *thesis* of the dialectic) is given. The initial condition is not a completely free state, but is usually restricted by boundary conditions. In response to the *thesis*, a reaction—the *antithesis*—occurs. As a result, the initial condition is sublated (in a process called *Aufheben*) in the transition to a new state, the *synthesis*. These series of interactions and reactions occur under the boundary conditions discussed above. We must take care that these boundary conditions also change as the interactions progress. As a result, the system moves to a stable state (or an unstable state or chaotic state). If the system is dominated by external, dynamic fluctuations or environmental perturbations, the "synthesis" of the first stage becomes the new "thesis" of the second stage. The repetition of the "thesis" → "antithesis" → "synthesis" cycle pushes the state ever onward in stages. This is the mechanism of evolution in the biological world.

One example that might help clarify this idea is the development and differentiation of a fertilized egg. When a fertilized egg develops, the "initial condition" consists of the egg ("thesis") and the sperm ("antithesis"). These combine and form a fertilized egg ("synthesis"), which defines the "boundary conditions." Starting from these "boundary conditions," the egg iterates the process of cell division into halves, quarters, eighths, and so forth to form the embryo and, ultimately, the individual. Ontogeny recapitulates phylogeny, phylogeny being the history of the evolution of life [48]. The state transition of continuously climbing up the multi-stage hierarchy applies not just to the ontogeny of a fertilized egg, but is the mechanism of the processes whereby matter, via chemical evolution from the beginning of the universe proceeding through (non-living) matter produced primitive life, such as bacteria, and then the mechanism underlying the processes of the long evolution of life itself, starting with primitive life forms, into the rich ecosystem of 10 million or more species observed today (see the phylogenetic tree in Column 4, discussed earlier). This process will ultimately converge into the "ω-point" proposed by Teilhard de Chardin [49].

5.3 Summary

> *Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.*
>
> —Albert Einstein

In this chapter, we have discussed how information processing models "inspired by life" can be applied to solve various real-world problems and provided examples of how such models have actually been applied. We have witnessed dramatic progress in research that analyzes various natural and social phenomena from the perspective of networks. In addition, *Internet businesses* have emerged that skillfully apply the characteristics of the *small world phenomenon* and *scale-free networks*.

We see that *universal network principles* are in play at various hierarchical levels in both the *natural* world and *human* world (society). The field of complex network science has advanced rapidly from the end of the twentieth century to the beginning of the twenty-first century, and there are great expectations for its application to a wide range of academic and business fields. Our hope is that this field will enlighten all those living in the twenty-first century in the crucial field of knowledge called *network literacy*. One of the most important themes for the future is to apply this knowledge energetically not only in building future ICT infrastructures, such as *new-generation network architectures*, but in designing and constructing ideal social systems as well.

References

1. S.K. Thomas, *The Structure of Scientific Revolutions* (The University of Chicago Press, Chicago, 1962)
2. H. Yukawa, Readings from Einstein, in vol. 1: *Special Theory of Relativity, Quantum Theory and Brownian Motion*, vol. 2: *General Theory of Relativity and Unified Theory*, vol. 3: *Einstein and his Thoughts* (Kyoritsu Shuppan, Tokyo, 1970–1972)
3. K. Okouchi, Y. Maeda, *Chuko Bakkusu Sekai no Meicho 29 Pascal, Panse Shohinshu* [Chuko Backs, World Classics 29, Pascal, Pensées/sketchbook] (Chuokoron-Shinsha, Tokyo, 1978)

4. Pascal, Pensées, Volume 1 of the Japanese translation by Y. Maeda, K. Yuki, *Chuko Kurashikkusu Panse 1* [Chuko Classics, Pensées 1] (Chuokoron-Shinsha, Tokyo, 2008)
5. Nobel Foundation, http://nobelprize.org/
6. Edited by NHK Einstein Project, supervised by T. Kaneko, *Watashi wa Kami no Pazuru wo Tokitai—Einstein Dokyumento* [I Want to Solve the Puzzle of God—Einstein Document] (Tetsugaku Shobo, 1992)
7. J. Ishihara, I. Okamoto, *Einstein Koenroku (Proceedings of Einstein's Lectures)* (Tokyo Tosho Co., Ltd., Tokyo, 1991)
8. K. Yano, S. Bunko (Shincho Paperbacks), *Life Story of Einstein* (Shinchosha, Tokyo, 1997)
9. L.D. Landau, E.M. Lifshitz, Landau and Lifshitz course of theoretical physics, the classical theory of fields, 6th edn. Translated into Japanese by T. Tsunetou, *Landau=Lifshitz Riron Butsurigaku Kyotei, Ba no Kotenron, Gensho Dai 6 Han—Denkirikigaku, Tokushu oyobi Ippan Sotaisei Riron* [Electrodynamics and Theories of Special and General Relativity] (Tokyo Tosho Co., Ltd., Tokyo, 1978)
10. C. Darwin, *On the Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life* (John Murray, London, 1859)
11. Richard Leakey abridged and introduced "The Illustrated Origin of Species", The Origin of Species by Charles Darwin (The Rainbird Publishing Group Ltd., London, 1979)
12. J.H. Holland, *Adaptation in Natural and Artificial System* (The University of Michigan Press, Ann Arbor, 1975)
13. M. Kimura, T. Mukai, *Shinichi Kusakabe, Bunshi Sinka no Churitsusetsu* [Neutral Theory of Evolution for Molecular Evolution] (Kinokuniya Company Ltd., Tokyo, 1986)
14. C. Darwin, *Beagle-go Kokaiki* [The Voyage of the Beagle] (Iwanami Bunko, Iwanami Paperbacks, 1959)
15. J. Malthus, *An Essay on the Principle of Population* (Chuokoron-Shinsha, Inc., Tokyo, 1973)
16. The Complete Work of Charles Darwin Online, http://darwin-online.org.uk/
17. K. Kaneko, I. Tsuda, *Fukuzatsu-kei Sosho, Fukuzatsu-kei no Kaosu-teki Shinario* [Complex Systems Library, Chaotic Scenarios of Complex Systems] (Asakura Publishing, Tokyo, 1996)
18. I. Tsuda, *Kaosu-teki No-kan—No no Atarashii Moderu wo Mezashite* [Chaotic Views of the Brain: Toward a New Model of the Brain] (Saiensu-Sha Co., Tokyo, 1990)
19. A.-L. Barabasi, *Linked: The New Science of Networks* (Basic Books, New York, 2002)
20. M. Buchanan, *NEXUS: Small Worlds and the Groundbreaking Science of Networks* (W.W. Norton & Company, New York, 2002)
21. D.J. Watts, *Six Degrees, The New Science of a Connected Age* (Vintage, London, 2004)
22. D.J. Watts, *Small World—The Dynamics of Networks Between Orders and Randomness* (Princeton University Press, Princeton, 1999)
23. N. Masuda, N. Konno, *Introduction to Complex Networks* (Sangyo Tosho, Tokyo, 2005)
24. N. Masuda, N. Konno, *What is Complex Network?* (Kodansha, Tokyo, 2006)
25. N. Masuda, *How Do We Connect Each Other? Applying Network Science* (Chuko Shinsho, Tokyo, 2007)
26. Y. Hayashi (ed.), *How to Spread Rumor—Unraveling the World by Network Science*. DOJIN Sensho 009 (Kagaku Dojin, Tokyo, 2007)
27. Y. Hayashi (ed.), *Toolbox of Network Science—Untying the Hidden Phenomena of Connections* (Kindai Kagakusha, Tokyo, 2007)
28. Special Issue on Expansion of Complex Network Science, vol. 49, No. 3, *Proceedings of IPSJ*, March 2008
29. N. Kamibayashi, *From Network to Networking—Social and Academic Significance of Network Science*, vol. 51, No. 8 (System, Control and Information Society, 2007), pp. 348–353
30. The Forefront of Network Science. *Mathematical Science*, No. 520 (October 2006)
31. S. Strogatz, *SYNC—The Emerging Science of Spontaneous Order* (Hyperion, Philadelphia, 2003)

32. S. Johnson, *Emergence—The Connected Lives of Ants, Brains, Cities, and Software* (Scribner, New York, 2001)
33. H. Tanaka, *Life and Complex System* (Baihu-kan, Tokyo, 2002)
34. H. Tanaka, *Life—Evolutional Molecular Network—An Introduction to System Evolutionary Biology* (Personal Media, 2007)
35. H. Kitano, K. Takeuchi, *Sound Life* (Diamond-sha, Tokyo, 2007)
36. T. Tanizawa, How to connect networks robust for both random failure and targeted attack? in Special issue on expansion of complex network science, *Proceedings of the IPSJ*, vol. 49, no. 3 (2008), pp. 282–289
37. T. Tanizawa, G. Paul, R. Cohen, S. Havlin, H.E. Stanley, Optimization of network robustness to waves of targeted and random attacks. Phys. Rev. E **71**(4), 047101 (2005)
38. T. Tanizawa, G. Paul, S. Havlin, H.E. Stanley, Optimization of the robustness of multimodal networks. Phys. Rev. E **74**(1), 016125 (2006)
39. T. Nakagaki, H. Yamada, A. Toth, Intelligence: maze-solving by an amoeboid organism. Nature **407**, 470 (2000)
40. A. Tero, R. Kobayashi, T. Nakagaki, Physarum solver: a biologically inspired method of road-network navigation. Physica A **363**(1), 115–119 (2006)
41. K. Nishinari, *Study on Congestion* (Shincho Sensho, Tokyo, 2006)
42. T. Nishiguchi, *Companionship in the Long Distance and the Neighborhood—Successful Strategy of Organized Network* (NTT Shuppan, Tokyo, 2008)
43. J. Kanemitsu, *Foundation on the Analysis of Social Network—Toward the Social Relational Capital* (Keiso Shobo, Tokyo, 2003)
44. S. Nozawa, *Readings on the Theory of Network—Family, Community and Social Relational Capital* (Keiso Shobo, Tokyo, 2006)
45. H. Ohe, *Sustainable Community Network—Regional Management Strategy in the Information Society* (Nihon Chiiki Shakai Kenkyu-sho, Tokyo, 2007)
46. Y. Yasuda, *Human Network Science—Investigation into the Power Hidden in the Relation Between Humans* (Nihon Keizai Shinbun-sha, Tokyo, 2004)
47. Y. Yasuda, *Network Analysis—What Determine Human Actions?* (Shinyo-sha, Tokyo, 1997)
48. J. Maynard Smith, E. Szathmáry, The major transitions in evolution, translated into Japanese by K. Nagano, *Shinka suru Kaiso—Seimei no Hassei kara Gengo no Tanjo made* (Springer, Tokyo, 1997)
49. Edited by W. Hiromatsu et al., *Iwanami Tetsugaku Shiso Jiten* [Iwanami Encyclopedia of Philosophy and Thought] (Iwanami Shoten, Tokyo, 1998)
50. M. Polanyi, The tacit dimension, translated into Japanese by I. Takahashi, *Chikuma Gakugei Bunko*, Anmokuchi no Jigen (Chikumashobo Ltd., Chikuma Paperbacks of Arts, 2003)
51. K.M. Dallenbach, A puzzle-picture with a new principle of concealment. Am. J. Psychol. **64**(3), 431–433 (1951)
52. D.J. Watts, S.H. Strogatz, Collective dynamics of 'small world' networks. Nature **393**, 440 (1998)

Epilogue

Twenty years have passed since studies began in the late 1980s on information processing technologies "inspired by life." This book discusses various approaches to research that produce useful ideas "inspired by life," with the goal of designing and constructing future information communication systems.

Chapter 1, "Reconsidering information and communications technology from life," discusses our knowledge of brain functions and the evolution of life, mining this knowledge for insights into the superb functions of life forms and their possible application to the design of information processing systems and information communication systems capable of solving real-world problems. Chapter 2, "Research trends of molecular communication technology," argues for the possibility of information communication systems based on molecules, harnessing the inherent functions of life forms at the molecular level. Chapter 3, "Artificial chemistry and molecular networks," and Chap. 4, "Signal transduction in biological systems and its possible use in computation and communication systems," discuss the design of information and communication system based on molecules, and shows the possibilities applicable to real-world problems. Chapter 5, "For deeper understanding," shows the original ideas for the deeper understanding of life and nature, and explains the examples of application for real-world using complex networks.

As discussed in this book, research "inspired by life" represents a new paradigm, fully worthy of the term "paradigm shift" coined by Thomas Kuhn to describe revolutions in the history of science and technology and marking an epochal transition of the highest significance. If we penetrate into the process of "bio-evolution," which surpasses human wisdom (once again, let me remind readers that even human wisdom is a product of bio-evolution), we may develop a powerful wellspring of ideas for designing and constructing the future information and communication society.

Here's a prediction of how research on information and communications technologies inspired by life will proceed over the next several decades.

According to *The Singularity is Near: When Humans Transcend Biology*[1] by Ray Kurzweil, paradigm shifts (technological innovations) have emerged at accelerating rates in recent years, while information technology capabilities (cost performance, speed, capacity, bandwidth) grow at exponential rates, even faster than that forecast by Moore's Law.

The Moore's Law is an empirical law which Gordon Moore, who is a co-founder of Intel Corp., advocated in 1965 on the US Electronics Journal. That says, "integration density of transistors become twofold every 18–24 months. Even at present, this empirical law is still valid after more than forty years. However, he himself said, "this law would not last forever, and I think its limitation would come about ten or fifteen years later."

One of the ultimate goals of brain science is to create an "artificial brain." Forecasts call for the "reverse engineering of the brain" to result in human intelligence software by 2045 or so. Integrated with concurrent "GNR" (G: genetics, N: nanotechnology, R: robotics) breakthroughs taking place, these cutting-edge technologies will lead in several decades to revolutionary technologies well beyond our imagination. Kurzweil, an inventor and the well-known person who exactly predicted the age of Internet, explains the existence of "*The singularity* (technological innovations)," and pointed out in his book that the advent of the singularity will come within the 21st century, or in the year of 2050, after 40 years from now, at the earliest (See footnote 1).

In this future, humans will become qualitatively different entities, embarking on a new stage of human evolution. It requires a vivid imagination to envision the actual technologies that will be available, but in view of the scale and nature of the effects of scientific and technological revolutions on human civilization, these technologies will doubtlessly have both bright and dark aspects. We must consider not just scientific and technological progress, but the social impact of such progress, including ethical issues, as we pursue our research and development.

Lastly, we conclude this book by expecting the readers who read this book to produce many fruitful technologies which will contribute to the GNR or BIN (B: Biotechnology or Brain technology, I: Informatics, N: Nanotechnology and Network technology) breakthroughs for creating the 21st century.

[1] Ray Kurzweil, "The Singularity is Near: When Humans Transcend Biology," Viking Adult, 2005.

Index