TURING 图灵原版计算机科学系列

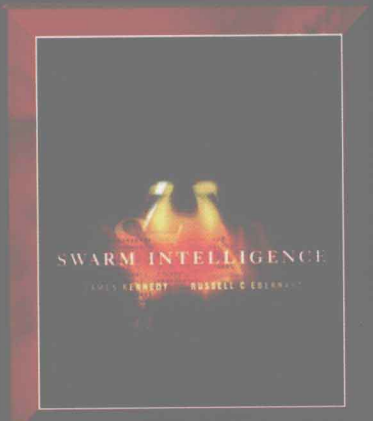
ELSEVIER 爱思唯尔

Swarm Intelligence

群体智能

（英文版）

James Kennedy
[美] Russell C. Eberhart 著
Yuhui Shi



人民邮电出版社
POSTS & TELECOM PRESS

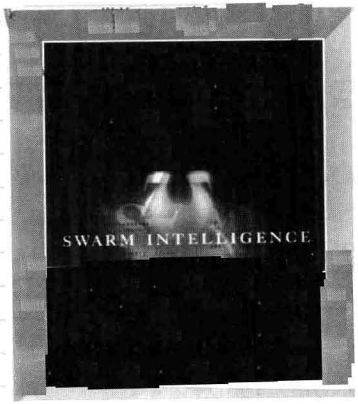
TURING 图灵原版计算机科学系列

Swarm Intelligence

群体智能

（英文版）

James Kennedy
[美] Russell C. Eberhart 著
Yuhui Shi



人民邮电出版社
北　京

图书在版编目（CIP）数据

群体智能＝Swarm Intelligence, First Edition：英文/（美）肯尼迪（Kennedy, J.），（美）埃伯哈特（Eberhart, R. C.），史玉回著. —北京：人民邮电出版社，2009.2
（图灵原版计算机科学系列）
ISBN 978-7-115-19550-0

I. 群… II. ①肯… ②埃… ③史… III. 人工智能－英文 IV. TP18

中国版本图书馆CIP数据核字（2008）第204824号

内 容 提 要

群体智能是通过模拟自然界生物群体行为来实现人工智能的一种方法。本书综合运用认知科学、社会心理学、人工智能和演化计算等学科知识，提供了一些非常有价值的新见解，并将这些见解加以应用，以解决困难的工程问题。书中首先探讨了基础理论，然后详尽展示如何将这些理论和模型应用于新的计算智能方法（粒子群）中，以适应智能系统的行为，最后描述了应用粒子群优化算法的好处，提供了强有力的优化、学习和问题解决的方法。

本书主要面向计算机相关学科的高年级本科生或研究生以及相关领域的研究与开发技术人员。

图灵原版计算机科学系列

群体智能（英文版）

- 著　[美] James Kennedy Russell C. Eberhart Yuhui Shi
  责任编辑　杨海玲
- 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
  邮编　100061　电子函件　315@ptpress.com.cn
  网址　http://www.ptpress.com.cn
  北京顺义振华印刷厂印刷
- 开本：800×1000　1/16
  印张：33.75
  字数：648千字　2009年2月第1版
  印数：1-2 000册　2009年2月北京第1次印刷

著作权合同登记号　图字：01-2008-5828号

ISBN 978-7-115-19550-0/TP

定价：75.00元

读者服务热线：（010）88593802　印装质量热线：（010）67129223
反盗版热线：（010）67171154

Preface

At this moment, a half-dozen astronauts are assembling a new space station hundreds of miles above the surface of the earth. Thousands of sailors live and work under the sea in submarines. Incas jog through the Andes. Nomads roam the Arabian sands. *Homo sapiens*—literally, "intelligent man"—has adapted to nearly every environment on the face of the earth, below it, and as far above it as we can propel ourselves. We must be doing something right.

In this book we argue that what we do right is related to our sociality. We will investigate that elusive quality known as intelligence, which is considered first of all a trait of humans and second as something that might be created in a computer, and our conclusion will be that whatever this "intelligence" is, it arises from interactions among individuals. We humans are the most social of animals: we live together in families, tribes, cities, nations, behaving and thinking according to the rules and norms of our communities, adopting the customs of our fellows, including the facts they believe and the explanations they use to tie those facts together. Even when we are alone, we think about other people, and even when we think about inanimate things, we think using language—the medium of interpersonal communication.

Almost as soon as the electronic computer was invented (or, we could point out, more than a century earlier, when Babbage's mechanical analytical engine was first conceived), philosophers and scientists began to ask questions about the similarities between computer programs and minds. Computers can process symbolic information, can derive conclusions from premises, can store information and recall it when it is appropriate, and so on—all things that minds do. If minds can be intelligent, those thinkers reasoned, there was no reason that computers could not be. And thus was born the great experiment of artificial intelligence.

To the early AI researchers, the mark of intelligence was the ability to solve large problems quickly. A problem might have a huge number of

possible solutions, most of which are not very good, some of which are passable, and a very few of which are the best. Given the huge number of possible ways to solve a problem, how would an intelligent computer program find the best choice, or at least a very good one? AI researchers thought up a number of clever methods for sorting through the possibilities, and shortcuts, called *heuristics,* to speed up the process. Since logical principles are universal, a logical method could be developed for one problem and used for another. For instance, it is not hard to see that strings of logical premises and conclusions are very similar to tours through cities. You can put facts together to draw conclusions in the same way that you can plan routes among a number of locations. Thus, programs that search a geographical map can be easily adapted to explore deductive threads in other domains. By the mid-1950s, programs already existed that could prove mathematical theorems and solve problems that were hard even for a human. The promise of these programs was staggering: if computers could be programmed to solve hard problems on their own, then it should only be a short time until they were able to converse with us and perform all the functions that we the living found tiresome or uninteresting.

But it was quickly found that, while the computer could perform superhuman feats of calculation and memory, it was very poor—a complete failure—at the simple things. No AI program could recognize a face, for instance, or carry on a simple conversation. These "brilliant" machines weren't very good at solving problems having to do with real people and real business and things with moving parts. It seemed that no matter how many variables were added to the decision process, there was always something else. Systems didn't work the same when they were hot, or cold, or stressed, or dirty, or cranky, or in the light, or in the dark, or when two things went wrong at the same time. There was always something else.

The early AI researchers had made an important assumption, so fundamental that it was never stated explicitly nor consciously acknowledged. They assumed that cognition is something inside an individual's head. An AI program was modeled on the vision of a single disconnected person, processing information inside his or her brain, turning the problem this way and that, rationally and coolly. Indeed, this is the way we experience our own thinking, as if we hear private voices and see private visions. But this experience can lead us to overlook what should be our most noticeable quality as a species: our tendency to associate with one another, to socialize. If you want to model human intelligence, we argue here, then you should do it by modeling individuals in a social context, interacting with one another.

In this regard it will be made clear that we do not mean the kinds of interaction typically seen in multiagent systems, where autonomous subroutines perform specialized functions. Agent subroutines may pass information back and forth, but subroutines are not changed as a result of the interaction, as people are. In real social interaction, information is exchanged, but also something else, perhaps more important: individuals exchange rules, tips, and beliefs about how to process the information. Thus a social interaction typically results in a change in the thinking processes—not just the contents—of the participants.

It is obvious that sexually reproducing animals must interact occasionally, at least, in order to make babies. It is equally obvious that most species interact far more often than that biological bottom line. Fish school, birds flock, bugs swarm—not just so they can mate, but for reasons extending above and beyond that. For instance, schools of fish have an advantage in escaping predators, as each individual fish can be a kind of lookout for the whole group. It is like having a thousand eyes. Herding animals also have an advantage in finding food: if one animal finds something to eat, the others will watch and follow. Social behavior helps individual species members adapt to their environment, especially by providing individuals with more information than their own senses can gather. You sniff the air and detect the scent of a predator; I, seeing you tense in anticipation, tense also, and grow suspicious. There are numerous other advantages as well that give social animals a survival advantage, to make social behavior the norm throughout the animal kingdom.

What is the relationship between adaptation and intelligence? Some writers have argued that in fact there is no difference, that intelligence *is* the ability to adapt (for instance, Fogel, 1995). We are not in a hurry to take on the fearsome task of battling this particular dragon at the moment and will leave the topic for now, but not without asserting that there is a relationship between adaptability and intelligence, and noting that social behavior greatly increases the ability of organisms to adapt.

We argue here against the view, widely held in cognitive science, of the individual as an isolated information-processing entity. We wish to write computer programs that simulate societies of individuals, each working on a problem and at the same time perceiving the problem-solving endeavors of its neighbors, and being influenced by those neighbors' successes. What would such programs look like?

In this book we explore ideas about intelligence arising in social contexts. Sometimes we talk about people and other living—carbon-based—organisms, and at other times we talk about silicon-based entities, existing in computer programs. To us, a mind is a mind, whether embodied in protoplasm or semiconductors, and intelligence is intelligence. The

important thing is that minds arise from interaction with other minds. That is not to say that we will dismiss the question casually. The interesting relationship between human minds and simulated minds will keep us on our toes through much of the book; there is more to it than meets the eye.

In the title of this book, and throughout it, we use the word *swarm* to describe a certain family of social processes. In its common usage, "swarm" refers to a disorganized cluster of moving things, usually insects, moving irregularly, chaotically, somehow staying together even while all of them move in apparently random directions. This is a good visual image of what we talk about, though we won't try to convince you that gnats possess some little-known intelligence that we have discovered. As you will see, an insect swarm is a three-dimensional version of something that can take place in a space of many dimensions—a space of ideas, beliefs, attitudes, behaviors, and the other things that minds are concerned with, and in spaces of high-dimensional mathematical systems like those computer scientists and engineers may be interested in.

We implement our swarms in computer programs. Sometimes the emphasis is on understanding intelligence and aspects of culture. Other times, we use our swarms for optimization, showing how to solve hard engineering problems. The social-science and computer-science questions are so interrelated here that it seems they require the same answers. On the one hand, the psychologist wants to know, how do minds work and why do people act the way they do? On the other, the engineer wants to know, what kinds of programs can I write that will help me solve extremely difficult real-world problems? It seems to us that if you knew the answer to the first question, you would know the answer to the second one. The half-century's drive to make computers intelligent has been largely an endeavor in simulated thinking, trying to understand how people arrive at their answers, so that powerful electronic computational devices can be programmed to do the hard work. But it seems researchers have not understood minds well enough to program one. In this volume we propose a view of mind, and we propose a way to implement that view in computer programs—programs that are able to solve very hard mathematical problems.

In *The Computer and the Brain,* John von Neumann (1958) wrote, "I suspect that a deeper mathematical study of the nervous system . . . will affect our understanding of the aspects of mathematics itself that are involved. In fact, it may alter the way in which we look on mathematics and logics proper." This is just one of the prescient von Neumann's predictions that has turned out to be correct; the study of neural systems has

opened up new perspectives for understanding complex systems of all sorts. In this volume we emphasize that neural systems of the intelligent kind are embedded in sociocultural systems of separate but connected nervous systems. Deeper computational studies of biological and cultural phenomena are affecting our understanding of many aspects of computing itself and are altering the way in which we perceive computing proper. We hope that this book is one step along the way toward that understanding and perception.

A Thumbnail Sketch of Particle Swarm Optimization

The field of evolutionary computation is often considered to comprise four major paradigms: genetic algorithms, evolutionary programming, evolution strategies, and genetic programming (Eberhart, Simpson, and Dobbins, 1996). (Genetic programming is sometimes categorized as a subfield of genetic algorithms.) As is the case with these evolutionary computation paradigms, particle swarm optimization utilizes a "population" of candidate solutions to evolve an optimal or near-optimal solution to a problem. The degree of optimality is measured by a fitness function defined by the user.

Particle swarm optimization, which has roots in artificial life and social psychology as well as engineering and computer science, differs from evolutionary computation methods in that the population members, called *particles,* are flown through the problem hyperspace. When the population is initialized, in addition to the variables being given random values, they are stochastically assigned velocities. Each iteration, each particle's velocity is stochastically accelerated toward its previous best position (where it had its highest fitness value) and toward a neighborhood best position (the position of highest fitness by any particle in its neighborhood).

The particle swarms we will be describing are closely related to *cellular automata* (CA), which are used for self-generating computer graphics movies, simulating biological systems and physical phenomena, designing massively parallel computers, and most importantly for basic research into the characteristics of complex dynamic systems. According to mathematician Rudy Rucker, CAs have three main attributes: (1) individual cell updates are done in parallel, (2) each new cell value depends only on the old values of the cell and its neighbors, and (3) all cells are updated using the same rules (Rucker, 1999). Individuals in a particle

swarm population can be conceptualized as cells in a CA, whose states change in many dimensions simultaneously.

Particle swarm optimization is powerful, easy to understand, easy to implement, and computationally efficient. The central algorithm comprises just two lines of computer code and is often at least an order of magnitude faster than other evolutionary algorithms on benchmark functions. It is extremely resistant to being trapped in local optima.

As an engineering methodology, particle swarm optimization has been applied to fields as diverse as electric/hybrid vehicle battery pack state of charge, human performance assessment, and human tremor diagnosis. Particle swarm optimization also provides evidence for theoretical perspectives on mind, consciousness, and intelligence. These theoretical views, in addition to the implications and applications for engineering and computer science, are discussed in this book.

What This Book Is, and Is Not, About

Let's start with what it's not about. This book is not a cookbook or a how-to book. In this volume we will tell you about some exciting research that you may not have heard about—since it covers recent findings in both psychology and computer science, we expect most readers will find something here that is new to them. If you are interested in trying out some of these ideas, you will either find enough information to get started or we will show you where to go for the information.

This book is not a list of facts. Unfortunately, too much science, and especially science education, today has become a simple listing of research findings presented as absolute truths. All the research described in this volume is ongoing, not only ours but others' as well, and all conclusions are subject to interpretation. We tend to focus on issues; accomplishments and failures in science point the way to larger theoretical truths, which are what we really want. We will occasionally make statements that are controversial, hoping not to hurt anyone's feelings but to incite our readers to think about the topics, even if it means disagreeing with us.

This book is about *emergent behavior (self-organization),* about simple processes leading to complex results. It's about the whole being more than the sum of its parts. In the words of one eminent mathematician, Stephen Wolfram: "It is possible to make things of great complexity out of things that are very simple. There is no conservation of simplicity."

We are not the first to publish a book with the words "swarm intelligence" in the title, but we do have a significantly distinct viewpoint from some others who use the term. For example, in *Swarm Intelligence: From Natural to Artificial Systems,* by Bonabeau, Dorigo, and Theraulaz (1999), which focuses on the modeling of social insect (primarily ant) behavior, page 7 states:

> It is, however, fair to say that very few applications of swarm intelligence have been developed. One of the main reasons for this relative lack of success resides in the fact that swarm-intelligent systems are hard to "program," because the paths to problem solving are not predefined but emergent in these systems and result from interactions among individuals and between individuals and their environment as much as from the behaviors of the individuals themselves. Therefore, using a swarm-intelligent system to solve a problem requires a thorough knowledge not only of what individual behaviors must be implemented but also of what interactions are needed to produce such or such global behavior.

It is our observation that quite a few applications of swarm intelligence (at least our brand of it) have been developed, that swarm intelligent systems are quite easy to program, and that a knowledge of individual behaviors and interactions is not needed. Rather, these behaviors and interactions emerge from very simple rules.

Bonabeau et al. define swarm intelligence as "the emergent collective intelligence of groups of simple agents." We agree with the spirit of this definition, but prefer not to tie swarm intelligence to the concept of "agents." Members of a swarm seem to us to fall short of the usual qualifications for something to be called an "agent," notably autonomy and specialization. Swarm members tend to be homogeneous and follow their programs explicitly. It may be politically incorrect for us to fail to align ourselves with the popular paradigm, given the current hype surrounding anything to do with agents. We just don't think it is the best fit.

So why, after all, did we call our paradigm a "particle *swarm?*" Well, to tell the truth, our very first programs were intended to model the coordinated movements of bird flocks and schools of fish. As the programs evolved from modeling social behavior to doing optimization, at some point the two-dimensional plots we used to watch the algorithms perform ceased to look much like bird flocks or fish schools and started looking more like swarms of mosquitoes. The name came as simply as that.

Mark Millonas (1994), at Santa Fe Institute, who develops his kind of swarm models for applications in artificial life, has articulated five basic principles of *swarm intelligence:*

- The *proximity* principle: The population should be able to carry out simple space and time computations.
- The *quality* principle: The population should be able to respond to quality factors in the environment.
- The principle of *diverse response:* The population should not commit its activity along excessively narrow channels.
- The principle of *stability:* The population should not change its mode of behavior every time the environment changes.
- The principle of *adaptability:* The population must be able to change behavior mode when it's worth the computational price.

(Note that stability and adaptability are the opposite sides of the same coin.) All five of Millonas' principles seem to describe particle swarms; we'll keep the name.

As for the term *particle,* population members are massless and volumeless mathematical abstractions and would be called "points" if they stayed still; velocities and accelerations are more appropriately applied to particles, even if each is defined to have arbitrarily small mass and volume. Reeves (1983) discusses *particle systems* consisting of clouds of primitive particles as models of diffuse objects such as clouds, fire, and smoke within a computer graphics framework. Thus, the label we chose to represent the concept is *particle swarm.*

Assertions

The discussions in this book center around two fundamental assertions and the corollaries that follow from them. The assertions emerge from the interdisciplinary nature of this research; they may seem like strange bedfellows, but they work together to provide insights for both social and computer scientists.

I. *Mind is social.* We reject the cognitivistic perspective of mind as an internal, private thing or process and argue instead that both

function and phenomenon derive from the interactions of individuals in a social world. Though it is mainstream social science, the statement needs to be made explicit in this age where the cognitivistic view dominates popular as well as scientific thought.

A. *Human intelligence results from social interaction.* Evaluating, comparing, and imitating one another, learning from experience and emulating the successful behaviors of others, people are able to adapt to complex environments through the discovery of relatively optimal patterns of attitudes, beliefs, and behaviors. Our species' predilection for a certain kind of social interaction has resulted in the development of the inherent intelligence of humans.

B. *Culture and cognition are inseparable consequences of human sociality.* Culture emerges as individuals become more similar through mutual social learning. The sweep of culture moves individuals toward more adaptive patterns of thought and behavior. The emergent and immergent phenomena occur simultaneously and inseparably.

II. *Particle swarms are a useful computational intelligence (soft computing) methodology.* There are a number of definitions of "computational intelligence" and "soft computing." Computational intelligence and soft computing both include hybrids of evolutionary computation, fuzzy logic, neural networks, and artificial life. Central to the concept of computational intelligence is system adaptation that enables or facilitates intelligent behavior in complex and changing environments. Included in soft computing is the softening "parameterization" of operations such as AND, OR, and NOT.

A. *Swarm intelligence provides a useful paradigm for implementing adaptive systems.* In this sense, it is an extension of evolutionary computation. Included application areas are simulation, control, and diagnostic systems in engineering and computer science.

B. *Particle swarm optimization is an extension of, and potentially important new incarnation of, cellular automata.* We speak of course of topologically structured systems in which the members' topological positions do not vary. Each cell, or location, performs only very simple calculations.

Organization of the Book

This book is intended for researchers; senior undergraduate and graduate students with a social science, cognitive science, engineering, or computer science background; and those with a keen interest in this quickly evolving "interdiscipline." It is also written for what is referred to in the business as the "intelligent layperson." You shouldn't need a Ph.D. to read this book; a driving curiosity and interest in the current state of science should be enough. The sections on application of the swarm algorithm principles will be especially helpful to those researchers and engineers who are concerned with getting something that *works*. It is helpful to understand the basic concepts of classical (two-valued) logic and elementary statistics. Familiarity with personal computers is also helpful, but not required. We will occasionally wade into some mathematical equations, but only an elementary knowledge of mathematics should be necessary for understanding the concepts discussed here.

Part I lays the groundwork for our journey into the world of particle swarms and swarm intelligence that occurs later in the book. We visit big topics such as life, intelligence, optimization, adaptation, simulation, and modeling.

Chapter 1, Models and Concepts of Life and Intelligence, first looks at what kinds of phenomena can be included under these terms. What is life? This is an important question of our historical era, as there are many ambiguous cases. Can life be created by humans? What is the role of adaptation in life and thought? And why do so many natural adaptive systems seem to rely on randomness?

Is cultural evolution Darwinian? Some think so; the question of evolution in culture is central to this volume. The Game of Life and cellular automata in general are computational examples of emergence, which seems to be fundamental to life and intelligence, and some artificial life paradigms are introduced. The chapter begins to inquire about the nature of intelligence and reviews some of the ways that researchers have tried to model human thought. We conclude that intelligence just means "the qualities of a good mind," which of course might not be defined the same by everybody.

Chapter 2, Symbols, Connections, and Optimization by Trial and Error, is intended to provide a background that will make the later chapters meaningful. What is optimization and what does it have to do with minds? We describe aspects of complex fitness landscapes and some methods that are used to find optimal regions on them. Minds can be

thought of as points in high-dimensional space: what would be needed to optimize them? Symbols as discrete packages of meaning are contrasted to the connectionist approach where meaning is distributed across a network. Some issues are discussed having to do with numeric representations of cognitive variables and mathematical problems.

Chapter 3, On Our Nonexistence as Entities: The Social Organism, considers the various zoom angles that can be used to look at living and thinking things. Though we tend to think of ourselves as autonomous beings, we can be considered as macroentities hosting multitudes of cellular or even subcellular guests, or as microentities inhabiting a planet that is alive. The chapter addresses some issues about social behavior. Why do animals live in groups? How do the social insects manage to build arches, organize cemeteries, stack woodchips? How do bird flocks and fish schools stay together? And what in the world could any of this have to do with human intelligence? (Hint: It has a lot to do with it.)

Some interesting questions have had to be answered before robots could do anything on their own. Rodney Brooks' subsumption architecture builds apparently goal-directed behavior out of modules. And what's the difference between a simulated robot and an agent? Finally, Chapter 3 looks at computer programs that can converse with people. How do they do it? Usually by exploiting the shallowness or mindlessness of most conversation.

Chapter 4, Evolutionary Computation Theory and Paradigms, describes in some detail the four major computational paradigms that use evolutionary theory for problem solving. The fitness of potential problem solutions is calculated, and the survival of the fittest allows better solutions to reproduce. These powerful methods are known as the "second-best way" to solve any problem.

Chapter 5, Humans—Actual, Imagined, and Implied, starts off musing on language as a bottom-up phenomenon. The chapter goes on to review the downfall of behavioristic psychology and the rise of cognitivism, with social psychology simmering in the background. Clearly there is a relationship between culture and mind, and a number of researchers have tried to write computer programs based on that relationship. As we review various paradigms, it becomes apparent that a lot of people think that culture must be similar to Darwinistic evolution. Are they the same? How are they different?

Chapter 6, Thinking Is Social, eases us into our own research on social models of optimization. The adaptive culture model is based on Axelrod's culture model—in fact, it is exactly like it except for one little thing: individuals imitate their neighbors, not on the basis of similarity,

but on the basis of their performance. If your neighbor has a better solution to the problem than you do, you try to be more like them. It is a very simple algorithm with big implications.

Part II focuses on our particle swarm paradigm and the collective and individual intelligence that arises within the swarm. We first introduce the conceptually simplest version of particle swarms, binary particle swarms, and then discuss the "workhorse" of particle swarms, the real-valued version. Variations on the basic algorithm and the performance of the particle swarm on benchmark functions precede a review of a few applications.

Chapter 7, The Particle Swarm, begins by suggesting that the same simple processes that underlie cultural adaptation can be incorporated into a computational paradigm. Multivariate decision making is reflected in a binary particle swarm. The performance of binary particle swarms is then evaluated on a number of benchmarks.

The chapter then describes the real-valued particle swarm optimization paradigm. Individuals are depicted as points in a shared high-dimensional space. The influence of each individual's successes and those of neighbors is similar to the binary version, but change is now portrayed as movement rather than probability. The chapter concludes with a description of the use of particle swarm optimization to find the weights in a simple neural network.

Chapter 8, Variations and Comparisons, is a somewhat more technical look at what various researchers have done with the basic particle swarm algorithm. We first look at the effects of the algorithm's main parameters and at a couple of techniques for improving performance. Are particle swarms actually just another kind of evolutionary algorithm? There are reasons to think so, and reasons not to. Considering the similarities and differences between evolution and culture can help us understand the algorithm and possible things to try with it.

Chapter 9, Applications, reviews a few of the applications of particle swarm optimization. The use of particle swarm optimization to evolve artificial neural networks is presented first. Evolutionary computation techniques have most commonly been used to evolve neural network weights, but have sometimes been used to evolve neural network structure or the neural network learning algorithm. The strengths and weaknesses of these approaches are reviewed. The use of particle swarm optimization to replace the learning algorithm and evolve both the weights and structure of a neural network is described. An added benefit of this approach is that it makes scaling or normalization of input data

unnecessary. The classification of the Iris Data Set is used to illustrate the approach. Although a feedforward neural network is used as the example, the methodology is valid for practically any type of network.

Chapter 10, Implications and Speculations, reviews the implications of particle swarms for theorizing about psychology and computation. If social interaction provides the algorithm for optimizing minds, then what must that be like for the individual? Various social- and computer-science perspectives are brought to bear on the subject.

Chapter 11, And in Conclusion . . . , looks back at some of the motifs that were woven through the narrative.

Appendix A, Statistics for Swarmers, is where we review some methods for scientific experimental design and data analysis. The discussion is a high-level overview to help researchers design their investigations; you should be conversant with these tools if you're going to evaluate what you are doing with particle swarm optimization—or any other stochastic optimization, for that matter. Included are sections on descriptive and inferential statistics, confidence intervals, student's *t*-test, one-way analysis of variance, factorial and multivariate ANOVA, regression analysis, and the chi-square test of independence. The material in this appendix provides you with sufficient information to perform some of the simple statistical analyses.

Appendix B, Genetic Algorithm Implementation, explains how to use the genetic algorithm software distributed at the book's web site. The program, which includes the famous Fisher Iris Data Set, is set up to optimize weights in a neural network. You can experiment with various parameters described in Chapter 4 to see how they affect the ability of the algorithm to optimize the weights in the neural network, to accurately classify flowers according to several measurements taken on them. The source code is also available at the book's web site and can be edited to optimize any kind of function you might like to try.

Software

The software associated with this book can be found on the Internet at *www.engr.iupui.edu/~eberhart/web/PSObook.html*. The decision to use the Internet as the medium to distribute the software was made for two main reasons. First, by *not* including it with the book as, say, a CD-ROM, the cost of the book can be lower. And we hope more folks will read the book

as a result of the lower price. Second, we can update the software (and add new stuff) whenever we want—so we can actually do something about it when readers let us know about the (inevitable?) software critters known as bugs. Some of the software is designed to be run online from within your web browser; some of it is downloadable and executable in a Windows environment on your PC.

Definitions

A few terms that are used at multiple places in the book are defined in this section. These terms either do not have universally accepted definitions or their definitions are not widely known outside of the research community. Throughout the book, glossary terms are italicized and will be defined in the back of the book. Unless otherwise stated, the following definitions are to be used throughout the book:

Evolutionary computation comprises machine learning optimization and classification paradigms roughly based on mechanisms of evolution such as biological genetics and natural selection (Eberhart, Simpson, and Dobbins, 1996). The evolutionary computation field includes genetic algorithms, evolutionary programming, genetic programming, and evolution strategies, in addition to the new kid on the block: particle swarm optimization.

Mind is a term we use in the ordinary sense, which is of course not very well defined. Generally, mind is "that which thinks." David Chalmers helps us out by noting that the colloquial use of the concept of mind really contains two aspects, which he calls "phenomenological" and "psychological." The phenomenological aspect of mind has to do with the conscious experience of thinking, what it is like to think, while the psychological aspect (as Chalmers uses the term, perhaps many psychologists would disagree) has to do with the function of thinking, the information processing that results in observable behavior. The connection between conscious experience and cognitive function is neither simple nor obvious. Because consciousness is not observable, falsifiable, or provable, and we are talking in this book about computer programs that simulate human behavior, we mostly ignore the phenomenology of mind, except where it is relevant in explaining function. Sometimes the experience of being human makes it harder to perceive functional cognition objectively, and we feel responsible to note where first-person subjectivity steers the folk-psychologist away from a scientific view.