9-1-2007

Review of: A Java Library of Graph Algorithms and Optimization by Hang T. Lau

Robert A. Beezer
University of Puget Sound, beezer@pugetsound.edu

Citation
Beezer, Robert A. 2007. "A Java Library of Graph Algorithms and Optimization." Siam Review 49(3): 517-518.

the book the reader gains a good overview of the considerations to be made when selecting the sensors and actuators, while in the second part several robotic mechanics and applications are given. The last part refers to higher level aspects of autonomous robotics such as localization and navigation, and exploration with advanced concepts such as neural networks or genetic programming. It is appreciated that the author gives the book a practical slant by reporting technicalities useful for practical implementations, thus avoiding the use of heavy formalisms. Moreover, the correct emphasis is placed on the importance of hardware in the loop simulations as part of the robotic design.

In the reviewer's opinion, this book is suitable as a textbook for a laboratory class on robotics. A student will learn most of the capabilities required to select the components and build, program, and control an autonomous robotic system. Since it is obviously not easy to produce a self-contained book for such a wide topic, and in order to really appreciate the book's contents, the reader should have a basic background in control/robotics.

The book might also be of interest to Ph.D. students when first reading about autonomous robotics; however, in the reviewer's opinion the book would not be of special interest to researchers with a mature background in this topic. In fact, in several chapters the bibliography does not seem to address the relevant citations; as an example, Chapter 12 contains six citations and four of them are theses; the Peters citation in Chapter 7 is in German. Furthermore, there is a certain nonhomogeneity in the depth of the various chapters' bibliographies, with some containing seminal books and journals and others reporting only conference papers.

Another comment concerns the relative weights among the topics: it is the reviewer's opinion that maybe classical control approaches are a little under-evaluated compared to genetic or neural network approaches.

In conclusion, the book is very well organized and it is written in a pleasantly concise style. Undergraduate and graduate students and researchers interested in embedded robotics will find it useful and rich in valuable material.

GIANLUCA ANTONELLI
Università degli Studi di Cassino

A Java Library of Graph Algorithms and Optimization. *By Hang T. Lau.* Chapman & Hall/CRC, Boca Raton, FL, 2007. $99.95. x+386 pp., hardcover. ISBN 978-1-58488-718-8.

As the title suggests, this text describes a collection of routines in the Java programming language solving basic problems in graph theory, such as locating the connected components of a graph, and in graph optimization, such as finding a maximum network flow. Each of the 55 routines is preceded by an introduction about a page in length. This is split between a description of the problem to be solved, with some references to the literature, and a description of the parameters passed to the routine. This is followed by the verbatim listing of the code. The one-page conclusion of each routine's description is a small example, typically a graph on 8 or 10 vertices, and the program listing for a small driver program and its output.

So roughly 250 pages are given over to program listings. The use of the word "library" in the title would suggest some sort of uniformity to the routines and an interdependence, utilizing common data structures and exploiting the object-oriented features of Java for efficiency in use and presentation. This is not the case. There is not even a graph object. Typical data structures are arrays of integers, so graphs are implemented as paired one-dimensional arrays of endpoints of edges (`nodei[], nodej[]`) or as a two-dimensional array that is the adjacency matrix. Worse, the code takes no advantage of Java's strengths and is written in a style reminiscent of C, Pascal, or FORTRAN. Arguments to routines are often long lists of arrays, there are no objects in sight anywhere, and the output requires the user to interpret the contents of the arrays containing the results. Besides questions of style, the particular choice of algorithms also raises questions. The description of depth-first search ignores the possibility of a recursive approach. The long section on graph isomorphism makes no mention of McKay's algorithm (implemented as the nauty package and freely available for non-military use), and instead uses an algorithm from a FORTRAN90 library.

Of the roughly 100 pages that are not code, there is little explanation of the choice of algorithms or data structures for the problem at hand. A similar effort to this book is Skiena's *Implementing Discrete Mathematics* (Addison–Wesley, 1990), which describes the Combinatorica package for *Mathematica*. The reader who expects the work under review to match Skiena's careful attempts to teach and explain will be very disappointed. The book's introduction says, "The library of programs is intended to be used for educational and experimental purposes." While the use of such a library could be beneficial in an educational setting (or for rapid prototyping), the book itself makes very little effort to teach or inform.

Included inside the back cover is a copyrighted compact disc. There is no reference to this disc in the book, no description of its contents, no guidance on allowed uses. The closest thing to a mention of how the disc may be used is the standard boilerplate on the copyright page which says, "No part of this book may be utilized... in any information storage or retrieval system... without written permission from the publishers." Hazarding a copyright violation, an examination of the contents of the disc reveals it contains the class files of the routines. These are the intermediate, machine-readable files created by a Java compiler, and not the original source files as printed. So modifications to these files (perhaps for "experimental purposes") are not made easily. For a project of this nature, it would be more useful to release the source code in electronic form with a license explaining clearly what the purchaser is allowed to do with the code and with any programs that might incorporate it. A search uncovered no web locations that might also be hosting the source code. Using Skiena's work as an example again, consider that his *Algorithm Design Manual* (TELOS, 1998) is available online in a hypertext-linked edition and is supported by the comprehensive Stonybrook Algorithm Repository website.

There is little to recommend in this project. The text and the code are not instructive. Practical use of the class files on the compact disc for experimental purposes is limited at best, and prohibited as described in the front matter. Whether or not the algorithms are carefully chosen and implemented, whether or not the style of coding is appropriate or instructive, delivery as a printed book with an electronic supplement containing only machine-readable implementations seems ill-advised.

ROBERT A. BEEZER
University of Puget Sound

Handbook of Mathematics for Engineers and Scientists. *By Andrei D. Polyanin and Alexander V. Manzhirov.* Chapman and Hall/CRC, Boca Raton, FL, 2007. $99.95. xxxii+1509 pp., hardcover. ISBN 978-1-58488-502-3.

How many hours are there in a day for Andrei Polyanin? Given his production over the last decade, simply measured in terms of reference books [3, 4, 5, 6, 7], the answer has to exceed 24. And now arrives the most massive of them all, the 1500-page handbook that is the subject of this review. The book is coauthored with Alexander Manzhirov, who was also a collaborator on [5]. That previous book has a far more limited scope: it focuses on a single topic, broad as it may be, and presents a seemingly exhaustive list of solutions to different integral equations. In addition, the user (I had written "reader," but corrected myself) finds sections on the methods leading to those formulae. The other reference texts [3, 4, 6, 7] are in the same vein. The current book is different. It is by far the most classical, and there are plenty of books available for direct comparison (see [1, 2, 8], for instance). Of these, CRC's own [8] is perhaps the best known. The most accurate comparison is probably with [1]. Both books not only list lots of facts, but also discuss methods, give theorems, and even include the occasional proof. For reference books such as this, one's favorite will often be the one first used, as this is the book that allows the user to find what is needed the fastest. Therefore I cannot recommend that we all