Genet Program Evolvable Mach (2011) 12:329–331
DOI 10.1007/s10710-011-9135-4

SOFTWARE REVIEW

Open BEAGLE: a generic framework for evolutionary computations

Dmitry Batenkov

Published online: 29 March 2011

1 Introduction and history

Software frameworks for evolutionary computation are very important both for developing applications using known algorithms and also for writing new algorithms. One such framework is Open BEAGLE [5, 6]. Below I give a general overview of this framework.

The project's leader (and principal developer) is Christian Cagné from Université Laval, Canada. Development started in 1999 and the last stable version (3.0.3) was released in November 2007. Since February 2009 an alpha release of the next major version has been available.

The name of the framework refers to the famous HMS Beagle, on which Charles Darwin made his voyage around the world [2]. Therefore, it is not surprising to find other software systems using this name both in genetic programming [3, 4] and in other fields [1].

2 Features

Open BEAGLE is a fully extensible generic framework which should in principle support any kind of evolutionary computation. This is achieved by a very clean object-oriented design. Out of the box, Open BEAGLE provides support for several flavours of genetic algorithms (bit-string, integer-valued vector, real-valued vector), as well as evolutionary strategies (including CMA-ES), tree-based genetic programming, co-evolution and multi-objective optimization. Unfortunately it does not support other GP flavours, such as linear GP. It implements a rich subset of

D. Batenkov (✉)
Weizmann Institute of Science, Rehovot, Israel
e-mail: dima.batenkov@gmail.com

crossover and mutation operators, as well as several initialization strategies, termination criteria and selection rules. It also supports strongly-typed GP, Automatically Defined Functions and ephemeral random constants. The object-oriented design enables one to add new components and to change the existing ones.

C++ implementation should be a general advantage in terms of performance, compared to frameworks in other languages (such as Java-based ECJ). Still, Open BEAGLE's dynamic polymorphism might slow things down. I was not able to find any objective performance comparison. A separate package called Distributed BEAGLE [7] is available for parallel computations, but unfortunately it is compatible only with version 2.x of Open BEAGLE.

Almost every parameter of the computation can be specified via configuration files without recompiling the application. Logging to the console and/or XML files is supported.

The user guide and manual, both available online, are good starting points. An architecture overview is also available, although it is somewhat outdated. The framework comes with a dozen clearly written code examples. In many cases these follow the standard textbooks in the field, such as [8]. The Doxygen documentation is extensive, containing in some places tutorial-style entries.

Only minimal theoretical knowledge of genetic programming and moderate object-oriented programming skills (say, of a 2nd year computer science undergraduate) should be sufficient in order to master the basics of Open BEAGLE. However, more advanced tasks (e.g. writing a new mutation operator) require slightly better programming skills, plus a good understanding of the inner architecture of the framework. This makes the learning curve somewhat steep.

The development cycle of the framework is slow, which is an obvious drawback. It seems that there is a great need for more contributors. As a compensation, Cagné and his collaborators are very committed to the project, providing good support through forums and mailing lists.

The framework is cross-platform. (It runs at least on Windows, Linux and Mac OS X). It is released under the GNU Lesser General Public License, which makes it available both for free and commercial applications.

3 Upcoming release

Although version 4 has been available for more than 2 years, it is still at the alpha stage and so should not be used in production-level code.

Additional genetic operators (e.g. insert mutation) have been implemented. Support for parallel (Open MP) and distributed (MPI) computations has been merged into the main code base from other sources (such as Distributed BEAGLE).

Compilation will become much easier with migration to CMake.

One important version 4 change is to Open BEAGLE's high-level architecture. While this change simplifies the whole implementation, it requires users of earlier versions to invest some time in order to adapt to it. However, all the code examples have been kept up-to-date, so this shouldn't be difficult.

4 Conclusion

Open BEAGLE is a mature, feature-rich, well-supported framework. Obviously a lot of thought and programming effort went into its design and implementation. I would definitely recommend considering it both for real GP projects and also for research and education.

Deciding which version to use is non-trivial. For basic usage (i.e. just using out-of-the-box functionality) and production code, I would recommend version 3. For distributed and parallel computations, version 4 seems to be the only viable option. If you plan to develop your own extensions, such as new crossover operators, I would recommend version 4.

Acknowledgments The author wishes to thank William B. Langdon for his support and useful suggestions.

References

1. L.B. Browning, Beagle: Genetic analysis software package. http://faculty.washington.edu/browning/beagle/beagle.html
2. C. Darwin, Voyage of the Beagle, 1989 edn. (penguin classics, Henry Colburn, London, 1839)
3. R. Feldt, M. O'Neill, C. Ryan, P. Nordin, W.B. Langdon, in *GP-Beagle: A Benchmarking Problem Repository for the Genetic Programming Community*, ed. by D. Whitley, Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference. (Las Vegas, Nevada, USA, 2000) pp. 90–97
4. F. Richard, BEAGLE A Darwinian approach to pattern recognition. Kybernetes **10**(3), 159–166 (1981)
5. C. Gagné, M. Parizeau, Open BEAGLE: A new versatile C++ framework for evolutionary computation. In *Late Breaking Papers at the Genetic and Evolutionary Computation Conference (GECCO-2002)*, ed. by E. Cantú-Paz (New York, NY, 2002) pp. 161–168. (AAAI)
6. C. Gagné, M. Parizeau, Open BEAGLE A C++ framework for your favorite evolutionary algorithm. SIGEvolution **1**(1), 12–15 (2006)
7. C. Gagne, M. Parizeau, M. Dubreuil, Distributed BEAGLE: An Environment for Parallel and Distributed Evolutionary Computations. In *Procceedings of the 17th Annual International Symposium on High Performance Computing Systems and Applications (HPCS) 2003*, Sherbrooke (QC), 11–14 May 2003
8. J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. (MIT Press, Cambridge MA USA, 1992)