

The Clever Algorithms Project: Inspiring Works*

Jason Brownlee
jasonb@CleverAlgorithms.com
The Clever Algorithms Project
<http://www.CleverAlgorithms.com>

January 9, 2010
Technical Report: CA-TR-20100109-1

Abstract

The Clever Algorithms project aims to provide a complete, consistent, and centralized description of a large number of Artificial Intelligence algorithms. The project was inspired by a number of book and software projects in the same and related domains, some of which provide desirable models for the project. This report reviews the inspiring books and software projects for the Clever Algorithms project highlighting the properties of each work that the project would like to emulate or adopt. Inspirations include classical Artificial Intelligence books such as Minsky's 'The Society of Mind', books on algorithms such as Charras and Lecroq's 'Handbook of Exact String Matching Algorithms', science fiction works released under permissive licenses such as Cory Doctorow's novels, and programming recipe works such as O'Reilly Media's Cookbook series. The desirable properties from all of the inspiring works are distilled into a set of principles that are proposed as guidelines for the ongoing content development and the production of outcomes for the project.

Keywords: Clever, Algorithms, Inspiration, Works, Motivation, Principles

1 Introduction

The Clever Algorithms project seeks to describe a large number of algorithms from the fields of Computational Intelligence and Biologically Inspired Computation in a complete, consistent, and centralized manner [6]. The project was not devised in isolation, it was inspired and influenced by a diverse collection of books (Section 2) and software (Section 3). This technical report provides an overview of the works that influenced the inception and development of the Clever Algorithms project and distills the specific features from the inspiring works into a set of principles and actions to help guide the project (Section 4).

2 Books

This section summarizes specific books that influenced the Clever Algorithms project.

*© Copyright 2010 Jason Brownlee. Some Rights Reserved. This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 2.5 Australia License.

2.1 The Society of Mind

The Society of Mind by Marvin Minsky [14] proposes a society of processes referred to as agents in a theory that seeks to explain how the mind works. The theory is presented as a series of approximately 270 one-page thought experiments organized into 30 chapters (and an appendix) each representing a specific feature or artifact of the mind.

Minsky also proposed the book as a society of many small ideas, that collectively propose the theory. Throughout the book, Minsky acknowledges comments and influence from students that helped refine specific elements, suggesting that the content for the book was developed over an extended period and presented for feedback. Each essay explores a concept with a thought experiment or explanation that is used as a building block in one or more other essays. As a result there are many cross-connections between the one page essays. In the prolog section, Minsky highlights the cross-connections between the topics as a necessary artifact of his construction, like the cross-connections in the mind itself. In the postscript and acknowledgements section, Minsky mentions that he tried to re-write the content ‘several times’ but found that a more traditional narrative could not accommodate all of the ideas effectively.

Minsky’s book is an exemplar for the content development model proposed for the Clever Algorithms project. Additionally, the book itself is an example of how the product of such a content development model may be usefully stitched together and collectively presented to an audience as a series of discrete (separable), semi-independent (some cross-references) structure of essays.

2.2 Evolutionary Computation

The Handbook of Evolutionary Computation [1] is a compendium (presented as a folder) of articles written by experts in the field of Evolutionary Computation generally about the experts own specialty sub-field. The objective of the handbook was to present complete, clear, and accessible information to thoroughly describe the state-of-the-art evolutionary computation research and applications in a comprehensive style. The handbook presents 8 parts that cover a range of topics (introduction, fundamental concepts, computational models, hybrid approaches, implementations, applications, case studies, and research), each of which is partitioned into chapters that presents a series of independent articles authored by various different experts.

This project was later updated resulting in the release of two books:

- Evolutionary Computation 1: Basic Algorithms and Operations [2]
- Evolutionary Computation 2: Advanced Algorithms and Operations [3]

The two volumes present similar material as the handbook although updated, and under a broader structure that presents increased detail. Volume one presents a series of six parts that cover major topics on evolutionary algorithms (why evolutionary computation, standard algorithms, representations, selection, and search) with 34 chapters, each chapter presenting anywhere between one and ten articles by specialists in the field. The second volume presents a similar structure (parts, chapters, articles) and focuses on more advanced topics in the field (fitness evaluation, constraint handling techniques, population structures, advanced techniques, and implementation). The articles are like short journal pieces with independent authorship and isolated structure (title, content, and references).

The handbook and the updated book volumes are seminal references for practitioners and researches in the field of Evolutionary Computation. The content in a given article is suitable to gain a basic understanding of an approach or operation, the dearth of articles means that all major subfields and niches in the area of research are covered, and the compactness of each article means that a number of articles can be read in a given sitting. A reader can skim across topics by flicking through the table of contents and dive into a specific sub-field in a random

access manner, both of which are properties of good reference text. The articles are independent, with few (if any) explicit cross-connections, although any required prior knowledge is located in one or more other articles in the collection.

The handbook and volumes cover a field a number of algorithms and related topics that are expected to be covered in the Clever Algorithms project. The handbook and volumes do present algorithms with pseudo code, but they are presented as areas of research, and each algorithm is presented differently. The Clever Algorithms project is different, it only focuses on the presentation of algorithms rather than on the myriad of sub-fields, and all algorithms will be presented using the same structure. The Evolutionary Computation books are complementary to the Clever Algorithms project, providing a specific and more detailed reference (although, at least ten years out of date at the time of writing) to accompany the presentation of most of the evolutionary algorithms in the project.

2.3 Handbook of Exact String Matching Algorithms

The Handbook of Exact String Matching Algorithms by Christian Charras and Thierry Lecroq [9] is a technical book that presents 34 string matching algorithms. The book starts with a single chapter providing some context for the field of text processing and the importance of string matching with some conventions. It then proceeds to present algorithms, one per-chapter. The description of each algorithm is systematic, including the following sections:

- *Title*: The name of the algorithm used as the chapter title.
- *Main features*: A short bullet-point listing of the computational complexity (efficiency) and data structures used in the algorithm.
- *Description*: A terse textual description of the distinctiveness of the algorithm, sometimes with bullet pointed features, diagrams, and mathematical equations.
- *The C code*: A concise listing of algorithm's source code in the C programming language, sometimes with a brief textual description.
- *The example*: Commonly a listing of the matching procedure on a specific problem instance, depicting the algorithms behavior graphically for each iteration until the example problem is solved.
- *References*: A bullet point listing of references suitable for understanding the presented algorithm.

Both PDF and Postscript versions of the book are available for free from Lecroq's website <http://www-igm.univ-mlv.fr/~lecroq/livres.html>, and the book can be purchased traditionally. It is unclear whether the published version of the text resembles the free version reviewed. The website for the author also shows four other books on string matching, highlighting the authors interest and dedication to the field. Again, it is unknown whether the other published volumes adopt a similar book structure and algorithm presentation methodology.

The book is an excellent model for the Clever Algorithms project. It is a text that focuses on the presentation of a large corpus of algorithms (I had no idea there were 34 ways to test-match strings), in a complete (diverse presentation forms) and consistent (fixed structure) way. The inclusion of a 'main features' element in the description is something that may be considered in the Clever Algorithms standardized algorithm description template [7]. The focused domain (string matching) allows for a consistent graphical motif to be used for algorithm descriptions that is both effective and meaningful within the domain. Generally, the algorithm descriptions are terse, providing sufficient information to understand the approach without overwhelming the reader, whilst providing a listing of seminal references for those readers interested to acquire a deeper understanding.

2.4 Cory Doctorow Novels

Cory Doctorow, among other professions, is a science fiction writer and proponent of the Creative Commons organization. Doctorow publishes his novels traditionally and releases his novels both under permissive licenses, allowing the works to be re-worked by fans. I have read two of Doctorow's novels, specifically:

- *Down and Out in the Magic Kingdom* [11]
- *Someone Comes to Town, Someone Leaves Town* [12]

The novel 'Down and Out in the Magic Kingdom' has a dedicated webpage <http://craphound.com/down> that provides a free copy of the novel in multiple formats under the Creative Commons Attribution-Noncommercial-No Derivative Works 1.0 Generic License. The novel 'Someone Comes to Town, Someone Leaves Town' also has a dedicated webpage <http://craphound.com/someone> that provides free downloads of the book released under the Creative Commons Attribution-Noncommercial-No Derivative Works 2.0 License. Generally, the reader can reproduce the work (different file formats, languages, publications, re-arrangements of the text, and even audiobooks), although cannot modify it (without permission), make money from it, and must provide the appropriate attribution. These books are popular because they are both great stories, although the adopted license has allowed a community to spring up around the respective books. Each book's webpage provides a blog that tracks updates to the unconventional publication and distribution of the works, encouraging the community to go even further. As of writing, the download page for 'Down and Out' shows 36 different versions of the book and 'Someone Comes to Town' shows 31 versions, presumably mostly prepared by fans.

Doctorow's works, specifically these two books provide great examples of what may be achieved by distributing a book traditionally and releasing its content under a permissive license (assuming a high-quality base product). These cases highlight that active community discussion and encouragement of derivative works are desirable properties for reaching a wider audience (generally, an author's biggest failure is not being read). Science fiction is a domain that may be more amenable than science to engendering passion and fans, and so ambitions of community may be less transferable to the Clever Algorithms project. The tools of facilitation appear to include a permissive license such as Creative Commons, hosting free copies of the book on a dedicated website, the encouragement of the reader-base to generate derivative works, and the promotion of created derivative works on the books webpage.

2.5 Clonal Selection as an Inspiration for Adaptive and Distributed Information Processing

The PhD dissertation in the field of Artificial Immune Systems by Jason Brownlee [5] (the current author). This thesis presents a series of adaptive information processing models inspired by specific structures and functions of the acquired immune system. The content for the thesis was developed initially as a series of nearly 50 discrete, semi-independent technical reports over a period of approximately six-months, available online from <http://www.it.swin.edu.au/personal/jbrownlee>. A total of 11 clonal selection algorithms were described in a semi-structured format motivated by the following descriptive elements:

- *Inspiration*: A summary of the motivating structure and/or functions of the acquired immune system for the proposed algorithm or system.
- *Strategy*: An abstraction of the inspiring system as a description of the computational processes, data structure, and architecture (if appropriate).

- *Empirical Assessment*: One or a series of structured experiments with the algorithm applied to a problem instance in order to demonstrate and empirically confirm the presence of expected systemic information processing properties. The description of empirical experiments was structured, including the following named elements:
 - *Aim*: The objective of the experiment.
 - *Method*: A declaration of the problem instance, algorithm procedure in pseudo code, and experimental procedure.
 - *Results*: A presented summary of the results
 - *Analysis*: A textual analysis of the presented results and their potential impact.
 - *Conclusions*: A textual summary of the findings of the experiment and their relationship to the stated objectives.

The algorithm description was somewhat structured, although without explicitly named description elements to encourage presentation consistency, although the motivating nature of the descriptions was formative for the clever algorithms project. A fixed and named structure was adopted for the description of empirical experiments, although generally, the presentation of each algorithm was too verbose compared to the concise descriptions discussed in Section 2.3. The content development methodology adopted for the Clever Algorithms project is patterned after the method used in the development of this thesis. A series of discrete, semi-independent (cross-referenced) technical reports were prepared on topics expected to be in and/or related to the topics presented in the thesis. The reports were prepared first and then reproduced with modification as the thesis in a patch work fashion, with gaps being filled as required.

This process will be adopted with some changes in the preparation of the book deliverable for the Clever Algorithms project. A series of technical reports will be prepared for topics expected to be in or related to the content of the front and back matter of the book, although a more structured approach will be adopted for algorithms where one report will be written for each algorithm that appears in the book. Additionally, the content of the book (and potentially other deliverables such as a website) are proposed to be developed in parallel with the technical reports in a series of discrete milestones.

2.6 A Field Programmers Guide to Genetic Programming

A Field Programmers Guide to Genetic Programming is a book by Riccardo Poli, William Langdon, and Nicholas McPhee [15]. The book is about the field of Genetic Programming, a sub-field of Evolutionary Computation. The book is self-published using the LuLu service (<http://www.lulu.com>) providing both a dead tree version for just over \$22USD with delivery (at the time of writing) and an eBook version for free released under the Creative Commons Attribution-Noncommercial-No Derivative Works 2.0 UK: England and Wales License.

The book has a dedicated website in the form of a blog <http://www.gp-field-guide.org.uk>, as well as a user group for discussing the topics and highlighting errata in the content. The books website links to the two published versions of the book, as well as the books presence on other online retailers such as Amazon. The blog discusses topics such as errata, positive mentions and reviews, as well as tracking sales volumes and site visit statistics. The books subject matter and its free availability have made it popular in googles ranking (when searching for ‘genetic programming’) as well as in online technical communities.

The book is in a field related to the Clever Algorithms project, although focuses on one technique (the genetic programming algorithms) and its variants, extensions, and applications in great detail. The self-publishing service LuLu handles both the on-demand publishing of the book as a paperback, and a hosting facility for the eBook, providing the tracking of both sales and downloads. The free status, quality content, and seniority of the authors in the field have

strongly contributed to the success (tracked downloads) of the book. The self-published status suggest the intent of the project was readership, although the popularity of the book may also suggest that some reasonable revenue may have also been generated (split between the three authors).

2.7 Pro Git

Pro Git is a book published by Scott Chacon [8] about the distributed version control system git. The book has a dedicated website <http://progit.org> that provides the content of the book for free in HTML released under a permissive Creative Commons Attribution-Non Commercial-Share Alike 3.0 license. Unlike the license used to distributed Doctorow's books in Section 2.4, this license allows the derivative works to be modified, although they cannot be commercial and must include appropriate attribution. The books website also promotes the dead-tree version, and provides a blog with topics related to the broader uptake of the book as well as translations of the books contents in other languages (prepared by readers).

The source code for the examples presented in the book are available as a publicly readable project on the Github website <http://github.com/progit/book-examples>. Interestingly, the HTML version (specifically markdown that is rendered as HTML) of the book is also made available as a publicly readable project on Github <http://github.com/progit/progit>. It appears that both of these projects were created after the release of the book and are intended to be used to track errata. The book project has been forked and translated, the results of which have been pulled back (included) into the original project, allowing moderated public contributions to the project. Viewing the commit log shows that contributions to the project are frequent and that development (translations) are still active at the time of writing.

The book is not about algorithms and does not contain algorithm descriptions, although it is a model for using a publicly accessible version control system for a book project, and for web-based dissemination of the content. Github is a social version control platform that encourages collaboration. The usage of Github to track errata for the examples source code and the book content is an excellent idea that can be exploited further. The book project on github does provide a README providing a terse context for the effort, although it does not (at the time of writing) explain the directory structure of the project, how users may contribute, or how or why the Github project came into existence. Github is a platform for collaboration, informing users how they may collaborate on a project that clearly has community interest (493 project followers and 133 forks at the time of writing) is expected to be a priority.

It is likely that the book content was not released publicly under a permissive license until after the publication of the dead-tree version, likely because of contractual constraints of the publisher. No such constraints are imposed on the Clever Algorithms project. As such, the content can be developed in a publicly readable repository allowing ad hoc contributions from any user who wishes to contribute to the project. Such contributions are not required to complete the project, and are not generally expected given the limited size of the target audience (even compared to git users), although the opportunity for such contributions is facilitated.

2.8 Global Optimization Algorithms - Theory and Applications

Global Optimization Algorithms - Theory and Applications by Thomas Weise [16] is about the broader field of optimization techniques in Artificial Intelligence. The book is release for free as a PDF on Weise's website <http://www.it-weise.de> under the GNU Free Documentation License, Version 1.2.

The book provides a long preface that describes the work as a the outcome of an extended project by Weise investigating the field. It also mentions that the book is not currently complete, with the incomplete sections marked with a 'TODO' statement in red colored text, sometimes in a large sized font. The book was started in 2006, is up to it's second revision (as of writing),

although is periodically updated and made available online. Differentiation between the many versions of the book are made through the issue number, and many reference style formats for the book are provided on the website. The preface also credits ‘a large number of people’ as providing corrections and contributions to the book over its lifetime.

The book is through and enormous, totaling 820 pages at the time of writing (Version: 2009-06-26). It covers a large number of topics, such as specific Computational Intelligence optimization algorithms and algorithm families, each of which are allocated a chapter for discussion. Algorithm descriptions are detailed including a narrative of background, inspiration, equations, pseudo code, application areas, books, and configuration. Aspects of a given technique may also be described at great length including variations, proofs, and applications.

The book covers the broader field directly related to the Clever Algorithms project and includes many of the techniques expected to be described by the project. Additionally, the book is the product of a side project of a practitioner and expert in the field, very much like the Clever Algorithms project, although Weise appears a lot more distinguished. The book is detailed and may provide an excellent reference text, although it is too detailed compared to the vision for the Clever Algorithms book deliverable. Additionally, the book project is in its fourth year (of a likely discontinuous development) and its large size may be an artifact of project scope creep.

The project involves the periodic self-publishing of an incomplete work. The process has resulted in a lot of useful feedback according to the author that contribute to the quality of the work, and this is an advantage of using an agile or iterative content development methodology (release early, release often). Reading a self-published book in an incomplete state is expected to damage the reputation and the quality of the project as it is a discomfort to the reader. It is proposed that an incomplete book is different to an incomplete piece of software, the latter is functional whereas the former is not. This problem may be addressed through the proposed technical report content development methodology for the Clever Algorithms project (early and complete sub-releases), and through a commitment to only releasing completed versions of the book deliverable.

2.9 O’Reilly Cookbooks

O’Reilly Cookbooks are a series of books published by O’Reilly Media that present recipes or quick fixes to common and specific programming problems. A typical book is partitioned into chapters of major topics and each chapter is partitioned into specific recipes. The recipes are presented in a consistent format (problem, solution, discussion, and see also), where the solution is most commonly a sequence of actions or snippet of code that can be directly used (in a target programming language, an executable form, and an exemplar and best practices solution to the presented problem).

O’Reilly has a website dedicated to the series <http://oreilly.com/store/series/cookbooks.csp> that provides an effective description of these books, as follows: “*Each cookbook contains hundreds of programming recipes presented in a special Problem/Solution/Discussion format, and includes hundreds of scripts, programs, and command sequences you can use to solve specific problems*” The source code from each cookbook is provided on the O’Reilly website and can be downloaded and copied directly into user applications, presumably released under a permissive license. The recipe structure taken from the O’Reilly website is as follows:

- *Problem*: Each problem is clearly stated, specific, and practical.
- *Solution*: The solution is easy to understand and implement.
- *Discussion*: The discussion clarifies and explains the context of the Problem and Solution.
- *See Also*: Directs you to additional information related to the topic.

The O'Reilly Cookbooks webpage lists approximately 50 books in the series (at the time of writing), each on a specific subject such as a programming language, environment, or development platform. Perhaps the most famous book in the series is the Perl Cookbook by Tom Christiansen and Nathan Torkington [10] that may have started the series. The source code solutions from the book have subsequently been ported to a number of other programming languages and released under the GNU Free Documentation License on a website dedicated to the maintenance of the examples <http://pleac.sourceforge.net>.

Although not about algorithms specifically, the cookbook format from O'Reilly is an excellent model for the presentation of content in the Clever Algorithms project. Essentially, each book is a compendium of discrete, independent, terse problem-solution pairs with executable programming code presented as the solutions. The intent of the Clever Algorithms project is to produce a reference text with the same general properties, although on a more complex subject matter, the details of which are drawn from a dispersed and elusive corpus of papers, articles, books, websites, and source code.

The cookbook format is very similar to the article approach used in the Evolutionary Computation books discussed in Section 2.2, although with a fixed presentation structure and more independence between the sections. A reader may search the table of contents (or the content electronically) and locate a solution to the general class to their specific problem. Alternatively, a reader may browse the text, and randomly access knowledge on an array of specific best practice programming idioms. This experience may be recreated in the Clever Algorithms project for Artificial Intelligence practitioners seeking general algorithmic solutions suited to difficult problems, and a corpus of such solutions that may be browsed for inspiration.

3 Software

This section summarizes the specific software that influenced the Clever Algorithms project.

3.1 Waikato Environment for Knowledge Analysis

The Waikato Environment for Knowledge Analysis (WEKA) is a machine learning workbench for exploring and experimenting with machine learning algorithms and data mining processes [13]. It is an open source software platform developed by the University of Waikato written in Java and released under the GNU General Public License (GPL). The project is used in university level machine learning and data mining courses, is a platform for scientific enterprise, has a dedicated website for the project <http://www.cs.waikato.ac.nz/~ml/weka> and at least one book written by the main contributors [17] (now in at a second edition).

The description from the WEKA webpage is as follows: “*Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.*”

The WEKA platform is host to a large number of standardized algorithm implementations for general classes of problems such as regression, classification, feature selection and clustering. The platform is concerned with the fields of data mining and machine learning that may have some overlap with the Clever Algorithms project. The number of algorithms is always increasing as third party developers implement algorithms, and researches implement and investigate their own approaches. I have been interested in the platform for a long time and have contributed to the effort by writing a series of algorithms for the platform, located at: <http://weka.classalgos.sourceforge.net>.

The software specifies programatic contracts (interfaces, base types, and abstract methods) that must be implemented for an algorithm to be included in the platform, enforces standardiza-

tion. This contract also includes referencing primary sources, parameterization documentation, and default algorithm configuration. These three pieces of information are invaluable to a practitioner for understanding where an algorithm came from, how to configure it, and provides a starting point for blind application (bad practice, but a typical use case). What the algorithms are missing are more detailed descriptions. This is a point that the associated WEKA book aims to address, although it departs from the structured standardized presentation style used in the software.

The large number of algorithms, and the high quality implementations has attracted a lot of attention resulting in it being considered a premiere platform for research in applied machine learning and data mining with many papers and articles referencing the software and the project. This supposition highlights that a positive feedback loop may be created for such a practical effort to the point where the platform becomes a de-facto standard in the field.

3.2 Optimization Algorithm Toolkit

The Optimization Algorithm Toolkit (OAT) is a software platform created by Jason Brownlee (this author) in Java for exploring and experimenting with optimization algorithms from the fields of Computational Intelligence and Biologically Inspired Computation [4]. It was developed between 2006 and 2008, was extended for use in Brownlee’s dissertation work (Section 2.5), and is released as open source under the GNU Lesser General Public License (LGPL) version 3. The project has a dedicated website <http://optalgtoolkit.sourceforge.net> that describes the general use cases, the target audience, and provides support information.

A description of the project from the projects website is as follows: “*The Optimization Algorithm Toolkit (OAT) is a workbench and toolkit for developing, evaluating, experimenting, and playing with classical and state-of-the-art optimization algorithms on standard benchmark problem domains. The software includes reference algorithm implementations, graphing, visualizations, and much more.*”

The project was inspired by the WEKA platform although targeted optimization algorithms, providing a large number of problem and algorithm instances for a handful of domains (such as continuous function optimization, binary function optimization, traveling salesman problem, and graph coloring). Algorithms are drawn from many fields of study such as Evolutionary Computation, Swarm Intelligence, Ant Colony Optimization, and Stochastic Optimization. Like the WEKA project, the OAT library enforces algorithm implementation standardization through the use of programming artifacts (interfaces, base classes, and abstract methods).

The OAT provides a large number of algorithms that are expected to appear in the Clever Algorithms project. In fact, the Clever Algorithms project is an effort to re-creation and improved version of the OAT, with the objective to be more accessible. OAT was broadly targeted to scientists, developers, and amateurs, although was only really usable by programmers that were also domain experts and likely research scientists. Additionally, a failure of the OAT project was that it did not provide sufficient support or documentation for the implemented algorithms. The Clever Algorithms project addresses these needs through focusing on algorithm descriptions, where demonstrative implementation are only one part of the description. The medium of technical reports, books, and website based content are also expected to be more accessible than a software workbench.

3.3 The Watchmaker Framework for Evolutionary Computation

The Watchmaker Framework for Evolutionary Computation is a software platform written in Java by Daniel Dyer for solving problems using evolutionary algorithms. It has a dedicated website <http://watchmaker.uncommons.org> and the software is released under the Apache Software License version 2.0.

The focus for the platform is not to implement a large number of evolutionary algorithms (like OAT in Section 3.2), but rather to provide a high-quality evolutionary engine and API to aid developers in applying the techniques to specific problem instances. The software platform has been relatively widely adopted, a highlight of which is the use of the library in the Apache Mahout project for distributed and scalable machine learning (see <http://lucene.apache.org/mahout>).

It is a good library for evolutionary computation and is related to a field that will be covered by the Clever Algorithms project, but the community engagement behaviors of Dyer the important observation. An ad hoc search of social news sites such as Hacker News <http://news.ycombinator.com> or Reddit <http://reddit.com> or programmer question answer sites such as Stack Overflow <http://stackoverflow.com> for the phrase ‘genetic algorithm’ shows Dyer’s presence. He has actively engaged the community by answering questions, providing support, and participated in discussions in the general field of his platform (evolutionary computation) for an extended period. More often than not, a comment and link to his Watchmaker framework is provided in the discussion.

This is not selfish marketing, and whether intentional or not, it is an excellent example of the benefits of actively engaging the target audience, demonstrating expertise and a reasonable adoption a viral marketing strategy. It is expected that the popularity of the project is due, in no small part, to this strategy of directly engaging the target audience.

4 Findings

The Clever Algorithms project is inspired by a large and diverse set of books and software projects. This section attempts to synthesize a set of desirable features distilled from the inspiring works that may be used as guiding principles for the project. These principles are partitioned into content development and outcome deliverables (book and website).

4.1 Content Principles

This section describes a set of principles to guide the Clever Algorithms project regarding the development of content.

- *Discrete Development*: The content shall be developed as a series of discrete pieces (such as technical reports and/or blog posts). The content may be released under a license similar to that used for the outcomes and may be distributed publicly for the purposes of refinement, collaboration, and marketing (google index and sites such as <http://scribd.com>).
- *Deliverable Milestones*: The end product shall be produce as a series of discrete and deliverable milestones. The development of content should be driven by the production of milestones. A version of the outcome (book and website) may be released for each milestone for comment.
- *Compact Content*: The content shall be minimal such as it is suitable to be useful but does not overwhelm the reader.
- *Fixed Structure*: Algorithms shall be described using a fixed and consistent structure. A template has already been defined [7] that shall be refined through application and experience.
- *Publicly Developed*: Work product shall be released publicly and be amenable to comment and refinement. The development process shall be public (repository in a publicly readable version control system) allowing attribution to be recorded, progress to be tracked and graphed, hosted, backed up, and facilitate contributions from interested readers with moderated inclusion into the main project.

- *Algorithm Focus*: The project shall remain algorithm focused, limiting effort into related topics such as proposed front matter (background) and back matter (advanced topics).
- *Clear and Fixed Scope*: The project shall define and adopt a scope early and refine it over the course of the project to mitigate scope creep. In the case of the book and website outcomes, a structure and algorithm list shall be defined and adopted early in the project (first milestone).
- *Algorithm Independence*: There shall be few explicit dependencies (cross-connections) between the algorithm descriptions, and descriptions shall assume a reasonable level of prior knowledge from the audience.
- *Algorithm Parameterization*: Algorithm descriptions shall include a reasonable description of parameters, their effect, how they should be configured, and recommended default configuration.

4.2 Outcome Principles

This section describes a set of principles to guide the Clever Algorithms project regarding the preparation of outcomes (such as a book and website).

- *Quantity*: The outcome shall contain a significant number (≥ 50) and diverse collection (computational models) of algorithm descriptions.
- *Quality*: The outcome shall be of a high-standard (publishable) having been through a process of peer review and code review before being officially released.
- *Permissive License*: The outcome shall be released under a permissive license (such as creative commons) that allows readers to choose the way in which they want to consume the product and create derivative works, although maintain appropriate attribution and copyright. The objective of the project is to have a readership (produce something useful) rather than monetary reimbursement (get paid).
- *Marketing*: The outcome shall have a dedicated webpage for the promotion of derivative works, and links to published and free versions of the product. The website may also encourage user reviews of the product, promote signed copies, and highlight the popularity of the work including sales figures, downloads, website visits, and citations.
- *Publishing*: The outcome shall be professionally published using a self-publishing company such as <http://lulu.com> providing a product listing, ISBN, print-on-demand, eBook hosting and download tracking.
- *Pricing*: The outcome shall be available for free, as well as for a small fee for a dead tree version. The product's set price shall be as low as possible (within reason).
- *Referencing*: The project shall provide examples of how to reference all produced work product including technical reports, milestones, and outcomes (book and website).
- *Complete Book*: Only complete versions of the book shall be released, no versions or incomplete states. Once complete, the book may be released, and updates may be realized through the release of subsequent revisions (a relatively slow frequency of 6 or 12 or 18 months). The content may be continually refined and reflected in the website version.
- *Foster Collaboration*: The project shall foster collaboration through distributed version control systems and clear explanation as to what types of contributions may be accepted and how they may be made. The project has a dictator in Jason Brownlee who manages the focus, direction, and moderation of the content for the outcomes.

References

- [1] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, 1997.
- [2] Thomas Bäck, D.B Fogel, and Z Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators*. IoP, 2000.
- [3] Thomas Bäck, D.B Fogel, and Z Michalewicz, editors. *Evolutionary Computation 2: Advanced Algorithms and Operations*. IoP, 2000.
- [4] Jason Brownlee. OAT: The optimization algorithm toolkit. Technical Report 20071220A, Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information and Communication Technologies, Swinburne University of Technology, December 2007.
- [5] Jason Brownlee. *Clonal Selection as an Inspiration for Adaptive and Distributed Information Processing*. PhD thesis, Complex Intelligent Systems Laboratory, Faculty of Information and Communication Technologies, Swinburne University of Technology, 2008.
- [6] Jason Brownlee. The clever algorithms project: Overview. Technical Report CA-TR-20100105-1, The Clever Algorithms Project <http://www.CleverAlgorithms.com>, January 2010.
- [7] Jason Brownlee. A template for standardized algorithm descriptions. Technical Report CA-TR-20100107-1, The Clever Algorithms Project <http://www.CleverAlgorithms.com>, January 2010.
- [8] Scott Chacon. *Pro Git*. Apress, 2009.
- [9] Christian Charras and Thierry Lecroq. *Handbook of Exact String Matching Algorithms*. King’s College Publications, 2004.
- [10] Tom Christiansen and Nathan Torkington. *Perl Cookbook*. O’Reilly Media, second edition, 2003.
- [11] Cory Doctorow. *Down and Out in the Magic Kingdom*. Tor Books, 2003.
- [12] Cory Doctorow. *Someone Comes to Town, Someone Leaves Town*. Tor Books, 2006.
- [13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.
- [14] Marvin Minsky. *The Society of Mind*. Simon and Schuster, 1988.
- [15] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A Field Programmers Guide to Genetic Programming*. Lulu Enterprises, 2008.
- [16] Thomas Weise. *Global Optimization Algorithms - Theory and Application*. Thomas Weise, 2009-06-26 edition, 2007.
- [17] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2000.