A Template for Standardized Algorithm Descriptions*

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

January 7, 2010
Technical Report: CA-TR-20100107-1

Abstract

The focus of the Clever Algorithms project is to provide a consistent and complete description of a number of algorithms from the fields of Computational Intelligence and Biologically Inspired Computation. For this effort to be successful a standardized algorithm description is required. A series of elements that may be used in a standardized description are proposed, each of which includes an expectation as to the elements contents, the form in which it should be written, and motivating questions to assist an implementor. A standardized algorithm template is proposed for the Clever Algorithms project that is comprised of a subset of the prospective elements is proposed. Usage recommendations are provided for the template that assist a would-be author with scope and presentation structure. Strategies for addressing redundancy and ambiguity when writing algorithm descriptions, as well as maintenance advice for refining the proposed template in the future. Three examples are provided of applying the template to popular algorithmic techniques, that, although are not complete, provide a suggestion as to how the template may be adopted. The area of algorithm selection criteria for inclusion in the project and algorithm name and scope recommendations are proposed as areas for future investigation.

Keywords: `Clever, Algorithms, Standard, Description, Template, Elements`

1 Introduction

The Clever Algorithm Project is concerned with the complete, consistent, and centralized description of algorithms from the fields of Computational Intelligence and Biologically Inspired Computation to ensure that they are accessible, usable, and understandable [1]. This report provides an exploration and definition of the standardized structure for algorithm descriptions in the project.

Section 2 provides a summary of algorithm description elements that may be used in a standardized algorithm description template. Section 3 proposes a template with specific description elements and expectations as to how the template should be adopted and refined. Section 4 demonstrates the usage of the proposed standardized algorithm description template against three popular algorithmic techniques.

2 Descriptive Elements

This section provides a summary of descriptive elements that may form a part of a standardized algorithm description. Each element is i) defined, ii) specifies the expected form of the description (such as size, scope and form), and iii) specifies two questions that motivate the content for the element.

2.1 Name

The algorithm name defines the canonical name used to refer to the technique, in addition to common aliases, abbreviations, and acronyms. The name is used in terms of the heading and sub-headings of an algorithm description.

- *What is the canonical name and common aliases for a technique?*
- *What are the common abbreviations and acronyms for a technique?*

2.2 Taxonomy: Lineage and locality

The algorithm taxonomy defines where a techniques fits into the field, both the specific subfields of Computational Intelligence and Biologically Inspired Computation as well as the broader field of Artificial Intelligence. The taxonomy also provides a context for determining the relationships between algorithms. The taxonomy may be described in terms of a series of relationship statements or pictorially as a venn diagram or a graph with hierarchical structure.

- *To what fields of study does a technique belong?*
- *What are the closely related approaches to a technique?*

2.3 Inspiration: Motivating system

The inspiration describes the specific system or process that provoked the inception of the algorithm. The inspiring system may non-exclusively be natural, biological, physical, or social. The description of the inspiring system may include relevant domain specific theory, observation, nomenclature, and most important must include those salient attributes of the system that are somehow abstractly or conceptually manifest in the technique. The inspiration is described textually with citations and may include diagrams to highlight features and relationships within the inspiring system.

- *What is the system or process that motivated the development of a technique?*
- *Which features of the motivating system are relevant to a technique?*

2.4 Metaphor: Explanation via analogy

The metaphor is a description of the technique in the context of the inspiring system or a different suitable system. The features of the technique are made apparent through an analogous description of the features of the inspiring system. The explanation through analogy is not expected to be literal scientific truth, rather the method is used as an allegorical communication tool. The inspiring system is not explicitly described, this is the role of the 'inspiration' element, which represents a loose dependency for this element. The explanation is textual and uses the nomenclature of the metaphorical system.

- *What is the explanation of a technique in the context of the inspiring system?*
- *What are the functionalities inferred for a technique from the analogous inspiring system?*

2.5 Strategy: Problem solving plan

The strategy is an abstract description of the computational model. The strategy describes the information processing actions a technique shall take in order to achieve an objective. The strategy provides a logical separation between a computational realization (procedure) and a analogous system (metaphor). A given problem solving strategy may be realized as one of a number specific algorithms or problem solving systems. The strategy description is textual using information processing and algorithmic terminology.

- *What is the information processing objective of a technique?*
- *What is a techniques plan of action?*

2.6 Procedure: Abstract computation

The algorithmic procedure summarizes the specifics of realizing a strategy as a systemized and parameterized computation. It outlines how the algorithm is organized in terms of the data structures and representations. The procedure may be described in terms of software engineering and computer science artifacts such as pseudo code, design diagrams, and relevant mathematical equations.

- *What is the computational recipe for a technique?*
- *What are the data structures and representations used in a technique?*

2.7 Heuristics: Usage guidelines

The heuristics element describe the commonsense, best practice, and demonstrated rules for applying and configuring a parameterized algorithm. The heuristics relate to the technical details of the techniques procedure and data structures for general classes of application (neither specific implementations not specific problem instances). The heuristics are described textually, such as a series of guidelines in a bullet-point structure.

- *What are the suggested configurations for a technique?*
- *What are the guidelines for the application of a technique to a problem instance?*

2.8 Applications: Demonstrated usage

The applications element describes the general problem classes to which the technique has been applied or to which the technique is well suited. The description may also include descriptions of specific problem instances that represent archetypal applications of the technique. The description of the applications are textual, likely with supporting references and potentially in a table structure.

- *What are the suggested classes of problems for which a technique is suited?*
- *What are some exemplar problem instances to which a technique has been applied?*

2.9 History: Inception and development

The history describes the inception of the technique including the original sources and the authors and contributors. The description also covers the subsequent and/or parallel development of the technique and the relevant original sources and contributors. The description may be textual with citations, and a timeline structure may be appropriate.

- *Under what conditions was a technique originally proposed?*
- *What were the significant milestones in the development of a technique?*

2.10 Code: Operational realization

The code description provides a minimal but functional version of the technique implemented with a programming language. The code description must be able to be typed into an appropriate computer, compiled or interpreted as need be, and provide a working execution of the technique. The technique implementation also includes a minimal problem instance to which it is applied, and both the problem and algorithm implementations are complete enough to demonstrate the techniques procedure. The description is presented as a programming source code listing.

- *How is a technique implemented as an executable program?*
- *How is a technique applied to a concrete problem instance?*

2.11 Tutorial: Guided implementation

The tutorial description provides a guide to realizing the technique using a programming language. The result of completing the tutorial is a minimal yet complete implementation of the technique applied to a problem, similar or the same to the source code description. The tutorial description provides explanations as to the design decisions and rationale for the way the technique is implemented. The tutorial description is textual, providing a narrative with an objective, a series of steps, and outcome that may be interspersed with source code examples.

- *What is the rationale when implementing a technique as an executable program?*
- *What is the rationale when applying a technique to a concrete problem instance?*

2.12 References: Deeper understanding

The references element description includes a listing of both primary sources of information about the technique as well as useful introductory sources for novices to gain a deeper understanding of the theory and application of the technique. The description consists of hand-selected reference material including books, peer reviewed conference papers, journal articles, and potentially websites. A bullet-pointed structure is suggested.

- *What are the primary sources for a technique?*
- *What are the suggested reference sources for learning more about a technique?*

3 Standardized Description

This section proposes a standardized algorithm template that includes some of the elements drawn from Section 2. The template is separate from the elements listed in the previous section to allow for independent on going refinement and modification.

3.1 The Template

A technique is more than just a procedure or code listing. Each approach is an island of research and the meta-information that define the context of a technique are just as important to understanding and application as abstract recipes and concrete implementations. A standardized algorithm description template shall be a suitable mixture of softer narrative descriptions,

programmatic descriptions both abstract and concrete, and most importantly useful sources for finding out more information about the technique.

The template proposed in this section should be considered a draft of a standard that is expected to be refined through adoption and use throughout the execution of the Clever Algorithms project. This report shall be updated to reflect the changes to the evolving algorithm description template, and the technical report number (currently CA-TR-20100107-1) and date shall be updated to reflect these updates. The final version of this standard to which this project converges will be used for all algorithms described in the projects compendium in order to ensure the objective of consistency is met. This constraint is suggested, but not required for the algorithms described in technical reports during content development.

Table 1 contains the standardized algorithm description template to be completed by all algorithms described in the Clever Algorithms project. The table describes the specific elements included in the template as well as their suggested ordering and their inclusion state as either required or optional.

Element	**Inclusion**	**Description**
Name	Required	The heading and alternate headings for the algorithm description.
Taxonomy	Required	A small tree diagram showing related fields and algorithms.
Inspiration	Optional	A textual description of the inspiring system.
Metaphor	Optional	A textual description of the algorithm by analogy.
Strategy	Required	A textual description of the information processing strategy.
Procedure	Required	A pseudo code description of the algorithms procedure.
Heuristics	Required	A bullet-point listing of best practice usage.
Tutorial	Required	A textural narrative for realizing the algorithm with complete source code.
References	Required	An bullet-point annotated reference list of primary sources and useful resources.

Table 1: Standardized algorithm description template.

The template describes the intended usage of each descriptive element and the arrangements of the elements. It is suggested that each element represent a heading in a complete description of an algorithm, except the name element that becomes the name of the description itself. For textual descriptions such as the inspiration, metaphor, and strategy elements, a small number of paragraphs (1-4) are suggested as suitable size and scope for a description.

Very few algorithms will have the information required for all of the elements in a ready-to-use format. Elements such as the metaphor and strategy will have to be deduced from the primary sources. The procedure will have to be mapped onto a common pseudo code standard adopted for all algorithms described in the project to ensure the consistency objective is observed. The tutorial element will also have to be prepared anew for each technique, ensuring that a minimal although complete implementation is prepared and explained. It is suggested that standalone source code is prepared and tested separate from the tutorial elements in the description to ensure that program correctness can be maintained. These code listing may be associated with the project but will not directly appear as an element in the description.

For many techniques, the inspiration and the metaphor will be similar or even identical, with differentiation between the techniques occurring at the strategy or maybe even the procedure elements. In such cases, these algorithms are likely to represent candidates for siblings in the taxonomy description. The selection criteria of what constitutes an algorithm as opposed to

a class of algorithms are not always well defined. This is a subject for future consideration, although in such a case where an algorithm name represents a family of specialization's that may or may not have their own differentiable names, a canonical or best practices form of the technique may be presented and the potential for variations discussed.

4 Examples

This section supports the standardized algorithm description template presented in Section 3 by briefly summarizing how three popular algorithms from the fields of Computational Intelligence and Biologically Inspired Computation may be described. The presented algorithm descriptions are not complete, rather they suggest at the content that may be provided in each description element for a given specific algorithm.

4.1 Genetic Algorithm

An example of the standardized algorithm description template applied to the genetic algorithm.

- *Name*: Genetic Algorithm (GA, Simple Genetic Algorithm, SGA, Canonical Genetic Algorithm)
- *Taxonomy*: An evolutionary algorithm, belongs to evolutionary computation. Sibling of genetic programming, evolution strategies, and evolutionary programming, parent of many techniques.
- *Inspiration*: The genetic understanding of evolutionary theory, the so-called new or modern synthesis of evolutionary genetics.
- *Metaphor*: Natural selection of better individuals who mate and propagate their species through a process of crossing the selected parents genetic material with the introduction of copying errors as mutation. This process results in an improved adaptive-fit between the species and the environment.
- *Strategy*: Management of a population of samples in a search space that are probabilistically selected based on 'goodness of fit' in the problem domain, the representations of which are recombined using a crossover operator and mutated to produce new samples in the in the search space each iteration. The process seeks to improve the goodness of the population in the search space against a cost function from the problem domain.
- *Heuristics*: Very low probability of mutation to preserve representations although introduce variation, very high to complete probability of crossover to recombine representations of fit parents, etc.
- *Procedure*: A procedural presentation of the algorithm's run loop in pseudo code.
- *Tutorial*: A guided example of implementing a simple genetic algorithm for a binary optimization problem such as the one max problem instance using a procedural programming language structure.
- *References*: Listing of Holland and Goldberg as primary and seminal sources, additionally introductory articles on the algorithm and popular books on the field.

4.2 Simulated Annealing

An example of the standardized algorithm description template applied to the simulated annealing algorithm.

- *Name*: Simulated Annealing (SA)
- *Taxonomy*: Stochastic global optimization algorithm, algorithms inspired by physical processes, a sibling of extremal optimization, a parent of many variants.
- *Inspiration*: Annealing process in metallurgy.
- *Metaphor*: A metal is heated and slowly cooled, causing the internal structure to seek a low energy state (configuration of atoms) that improves the strength and quality of the metal.
- *Strategy*: Management of a sample in a search space that is iteratively resampled. A probabilistic acceptance criteria is used that uses the goodness of the sample in the problem domain to decide whether a new sample replaces the existing sample, which starts out permissive and progressively becomes restrictive over the number of samples taken.
- *Heuristics*: The longer the cooling schedule the more chance the procedure has to find the minimum energy state, etc.
- *Procedure*: A procedural presentation of the algorithm's run loop in pseudo code.
- *Tutorial*: A guided example of implementing the classical simulated annealing algorithm for a combinatorial function optimization problem such as a traveling salesman problem instance using a procedural programming language structure.
- *References*: Listing of Kirkpatrick et al. and Metropolis et al. as the primary and seminal sources, additionally introductory articles on the algorithm and popular books on the field.

4.3 Particle Swarm Optimization

An example of the standardized algorithm description template applied to the particle swarm optimization algorithm.

- *Name*: Particle Swarm Optimization (PSO)
- *Taxonomy*: Belongs to swarm intelligence and collective intelligence, a sibling of ant colony optimization, a parent of many variants.
- *Inspiration*: Inspired by the flocking behaviors of birds and the schooling behaviors of fish.
- *Metaphor*: A flock of creatures move through an environment in search of a desired location (such as a good location of food), sharing observations about the environment with each other in localized neighborhoods. Through the momentum of their movement and localized social interactions, the group locates a good location that meets their needs in the environment.
- *Strategy*: A population of particles with velocity (speed and direction) move through a multiple dimension search space, the positions and evaluation of which are sampled at discrete intervals. The velocity of the particles is also discretely updated based on the observations provided by a given particles history as well as its local neighbors. The process seeks to converge the positions of the particles to a shared understanding of a good location in the search space as evaluated by a cost function in the problem domain.

- *Heuristics*: The size of the neighborhood relative to the size of the population influences the amount of exploration or convergence of the population, etc.
- *Procedure*: A procedural presentation of the algorithm's run loop in pseudo code.
- *Tutorial*: A guided example of implementing the classical particle swarm optimization algorithm for a continuous function optimization problem instance using a procedural programming language structure.
- *References*: Listing of Kennedy and Eberhart as primary sources, additionally introductory articles on the algorithm and popular books on the field.

5 Conclusions

This report introduced a number of elements that may be used in a standardized algorithm description template, then proceeded to specify those elements that shall be adopted in a standardized template for use in the Clever Algorithms project. Importantly, the template was presented with practical suggestions for its usage including scope, organization, formulation recommendations for writing algorithm descriptions as well as maintenance advice for refining the proposed template in the future.

It is expected that the presented standardized template and/or the proposed elements for use in such a template may relevant beyond the Clever Algorithms project to the broader fields of Biologically Inspired Computation and Computational Intelligence. The proposed template (or a new formulation) may be used in other algorithm description projects such as books, programming libraries, dissertations, and perhaps articles and papers. An important area for future investigation are the criteria used for algorithm selection. Included with these criteria must be additional criteria for handling those algorithmic techniques that do not have a single well defined implementation, but instead consist of a family of variant techniques and operations.

References

[1] Jason Brownlee. The clever algorithms project: Overview. Technical Report CA-TR-20100105-1, The Clever Algorithms Project http://www.CleverAlgorithms.com, January 2010.