The Clever Algorithms Project: Algorithm Selection*

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

January 12, 2010
Technical Report: CA-TR-20100112-1

Abstract

The Clever Algorithms project proposes to describe a large number of algorithms from the fields of Computational Intelligence and Biologically Inspired Computation in a complete, consistent, and centralized manner in order to improve the accessibility of the approaches. This report proposes a data-driven methodology for selecting algorithms to describe in the Clever Algorithms project based on objective algorithm popularity in search engines across a number of domains. An automated script is used to evaluate a prepared listing of unconventional optimization algorithms against six different search domains. The collected measures are used to produce a scoring for each algorithm, and the algorithms are ranked. Although the designed methodology produced reasonable and intuitive results, a number of areas in the process are highlighted as opportunities for future improvement such as the use of algorithm names to compare techniques, the assignment of algorithms to taxonomic kingdoms, and the introduction of biasing measures based on their source in the scoring of algorithms. A diverse and interesting selection of 50 algorithms is made from the results and proposed as a first draft for the listing of algorithms to be described by the Clever Algorithms project.

Keywords: `Clever, Algorithms, Selection, Methodology, Results`

1 Introduction

The Clever Algorithms project aims to describe a large number of algorithms from the field of Artificial Intelligence in a complete, consistent, and centralized way to improve the accessibility of the methods [1]. This report focuses on the methodology for selecting algorithms that appear in the project and provides a preliminary suggested listing of algorithms to describe. Section 2 describes a data-driven methodology for organizing, evaluating, and selecting algorithms. Section 3 summarizes the raw results of applying the methodology in terms of the ranked algorithms. Section 4 analyzes the presented results and comments on areas for improvement in the methodology. Section 5 provides a preliminary listing of algorithms selected to be described in the Clever Algorithms project. Section 6 reviews the findings of the process, and highlights some areas for future consideration.

2 Methodology

This section describes the methodology for the selection of algorithms for inclusion in the Clever Algorithms project. This methodology and its results are based on a post on the blog 'Never Read Passively' by Jason Brownlee (this author) on August 2nd 2009 entitled "What is a good optimization algorithm? A data-driven method for algorithm selection"[1]. The method presented in this previous work was based on the question: *Is it possible to select a diverse, interesting, and useful set of inspired algorithms using a simple data driven method?*. The methodology presented in this section is updated to include more sources used for algorithm comparison and uses a weighted sum of normalized measures when calculating an algorithms score. The application of this methodology was updated to use an automated script and the listing of algorithms for evaluation was updated and refined. The methodology is presented in terms of the three core tasks in the process: algorithm list, algorithm ranking, and algorithm selection.

2.1 Algorithm List

The first task involves the preparation of a large listing of candidate algorithms. The algorithms may be drawn from the an array of fields under or related to the fields of Biologically Inspired Computation and Computational Intelligence, such as: Meta-heuristics, Hyper-heuristics, Natural Computation, Bio-memetics, Swarm Intelligence, Collective Intelligence, and Evolutionary Computation.

The sources for algorithms names should be diverse and may include books, articles, papers, magazines, websites, and software. Algorithm name standardization is an important aspect of the methodology. Algorithms should be listed using their canonical (most common) name, normalized, without acronyms (wherever reasonably possible), and in their full unabbreviated form. The language used to to list algorithm names should be standardized, such as to American English.

Each algorithm should be assigned a kingdom for taxonomic reasons. The kingdom is used to group algorithms which defines the scope of competition that the algorithms must survive to be included in the project. Kingdoms may be superficial names (taxonomic names the algorithms may be assigned based on general observation and intuition), such as: evolutionary, immunological, swarm, physical, probabilistic, and stochastic.

2.2 Algorithm Rankings

Each algorithm in the prepared listing must be evaluated, assigned a scoring, and ranked to allow selection decisions to be made. A data-driven popularity-based approach is proposed for algorithm evaluation. Each algorithm name is submitted to an array of different domain specific search engines and the total approximated number of results reported by the search services are taken as measures. The submission of the algorithm name is standardized such that it is surrounded by quotes to ensure that the search is specific and the algorithm name is converted to lower case.

Some example search domains to which algorithm names may be submitted include:

- *Google Web Search*: Search of webpage on the internet index by Google.
- *Google Book Search*: Search of books index by Google.
- *Google Scholar Search*: Search of the academic publications and patents index by Google.
- *Springer Article Search*: Search of the articles and papers published by Springer.
- *Scirus Article Search*: Search of the articles and papers indexed by Scirus.

[1] Online: http://www.neverreadpassively.com/2009/08/what-is-good-optimization-algorithm.html

- *IEEE Article Search*: Search of the articles and papers published by IEEE.
- *JSTOR Article Search*: Search of the articles and papers scanned and indexed by JSTOR.
- *CiteSeer Article Search*: Search of the articles and papers indexed by CiteSeer.
- *ACM Article Search*: Search of the articles and papers indexed by ACM Digital Library.
- *Amazon Book Search*: Search of the books indexed and sold by Amazon.

The collected measures are representative of how often an algorithm name appears in a particular search domain, and inferred to represent the popularity to which a given algorithm is written about and studied. This data-driven algorithm evaluation strategy may be simply stated as: "*An algorithm or method is as good its name, as useful and interesting as it is abundant*", or more specifically, an algorithm that is used (written about) is an algorithm that is useful (worth knowing about).

The collected measures are then combined to produce a scoring for a given algorithm name. The combination may be a weighted sum of the measures, where different weighting factors may be used to increase the influence of a particular search domain. For example an algorithms occurrence in published books may have more importance than an algorithms occurrence on webpages.

Finally, the calculated algorithms scores are used to compare algorithms. Algorithm comparisons may be direct, or may be scoped, such as by using an algorithms allocated taxonomic kingdom.

2.3 Algorithm Selection

The selection of the algorithms that are considered in the project is as important, if not more important, than the descriptions of the algorithms themselves. The particular mixture of algorithms must include popular techniques that readers are expected to lookup, although the pool must be diverse enough to be interesting and promote discovery. As such, the selection of algorithms may not rely upon the popularity-based ranking of algorithms alone. Algorithms may be sorted by ranking and kingdom which provide an indication of algorithm popularity. The sorted ranked algorithms may then be filtered using using indicators for algorithm interestingness and diversity. Such measures are difficult to quantify and automate and so may rely on the subjective human assessments of an expert.

3 Results

This section summarizes the results of assessing algorithms for inclusion in the Clever Algorithms project.

3.1 Configuration

This section describes the configuration under which the data was collected, as follows:

- The algorithm list was prepared with 113 algorithms collected primarily for optimization rather than model building. These algorithms are referred to as 'unconventional optimization algorithms', and subfields that were intentionally neglected include artificial neural networks, fuzzy logic, and statistical machine learning subjects such as kernels.
- The algorithms were divided into 6 taxonomic kingdoms: evolutionary (36), immunological (6), swarm (22), physical (9), probabilistic (17), and stochastic (23).

- The search engines used as algorithm measures were: Google Web, Google Book, Google Scholar, Springer Article, Scirus Article, and IEEE Article Search.
- The data collection process was automated with a ruby script that performed the searches, collated the results, and generated the final statistics and ordered listings (the Ruby script is available from http://www.CleverAlgorithms.com).
- The measures from the search domains were combined into the algorithm scoring by first normalizing the results for ah measure and summing the measures together. No weighting factors were used in the calculation of the algorithm scores.

3.2 Raw Data

This section described the raw data collected as a result of following the experimental methodology. The results were collected from the specified search engines on January 11th 2010. The results show the collected measures (unnormalized), and the score (rounded to three decimal places) calculated as the sum of the normalized measures.

Table 1 highlights the top 10 most popular algorithms overall. The remaining tables present the raw results for each kingdom.

Kingdom	Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
evolutionary	genetic algorithm	282000	4149	380000	18352	179189	16411	6.0
physical	simulated annealing	132000	3680	307000	10493	135676	4191	3.747
stochastic	random search	161000	1509	32400	2855	121802	405	1.88
evolutionary	genetic programming	64000	1469	38700	4699	67457	1223	1.39
stochastic	tabu search	42800	1399	35400	4208	36146	1035	1.076
swarm	particle swarm optimization	49200	722	24600	2151	15566	3999	0.861
stochastic	reactive search optimization	53	3543	4	0	45	0	0.854
evolutionary	evolutionary programming	19400	974	17500	1933	15421	706	0.584
probabilistic	expectation-maximization algorithm	13700	727	14400	1182	15345	1460	0.501
evolutionary	differential evolution	17400	868	11300	934	10562	803	0.459

Table 1: The top 10 highest ranking algorithms.

Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
simulated annealing	132000	3680	307000	10493	135676	4191	3.747
memetic algorithm	4070	416	2850	601	2963	182	0.182
adaptive simulated annealing	2770	589	1640	147	2493	51	0.181
extremal optimization	1290	242	658	103	729	19	0.075
cultural algorithm	1090	188	559	72	329	76	0.061
quantum annealing	2660	154	415	36	1007	2	0.055
simplex-simulated annealing	405	84	299	17	234	8	0.025
harmony search algorithm	884	50	266	44	150	21	0.02
stochastic tunneling algorithm	42	3	15	3	9	0	0.001

Table 2: Raw results for the 'physical algorithms' taxonomic kingdom, ordered by score, descending.

Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
clonal selection algorithm	2120	183	1260	209	668	132	0.078
negative selection algorithm	1320	173	937	168	563	74	0.065
artificial immune recognition system	580	53	331	35	352	18	0.02
immune network algorithm	378	41	167	29	115	20	0.015
b-cell algorithm	102	29	60	20	70	4	0.009
dendritic cell algorithm	241	16	98	31	211	2	0.008

Table 3: Raw results for the 'immune algorithms' taxonomic kingdom, ordered by score, descending.

Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
expectation-maximization algorithm	13700	727	14400	1182	15345	1460	0.501
population-based incremental learning	1510	294	1110	190	1157	60	0.099
cross-entropy method	1840	293	1110	113	1505	49	0.097
bayesian optimization algorithm	1780	251	1060	207	1599	24	0.091
compact genetic algorithm	1310	145	726	118	1576	45	0.059
univariate marginal distribution algorithm	484	117	378	86	364	15	0.038
probabilistic incremental program evolution	357	84	189	54	293	10	0.027
hierarchical bayesian optimization algorithm	431	69	215	64	430	5	0.025
extended compact genetic algorithm	447	52	249	44	1093	8	0.024
bivariate marginal distribution algorithm	211	55	172	33	157	2	0.017
gaussian adaptation	175	33	41	4	27	2	0.009
mutual information maximization for input clustering	52	23	67	12	56	1	0.007
stochastic hill climbing with learning by vectors of normal distributions	45	20	39	12	29	0	0.006
estimation of bayesian networks algorithm	44	15	44	14	34	0	0.005
graduated optimization	110	5	24	2	30	2	0.002
estimation of multivariate normal algorithm	47	8	30	3	33	0	0.002
estimation of gaussian networks algorithm	17	9	30	5	8	0	0.002

Table 4: Raw results for the 'probabilistic algorithms' taxonomic kingdom, ordered by score, descending.

Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
genetic algorithm	282000	4149	380000	18352	179189	16411	6.0
genetic programming	64000	1469	38700	4699	67457	1223	1.39
evolutionary programming	19400	974	17500	1933	15421	706	0.584
differential evolution	17400	868	11300	934	10562	803	0.459
evolution strategies	12300	785	11800	1629	10995	288	0.431
simple genetic algorithm	5760	711	6920	747	4717	235	0.291
learning classifier system	2520	485	1700	146	2238	87	0.156
grammatical evolution	2830	473	1020	209	1661	33	0.149
non-dominated sorting genetic algorithm	2170	403	2540	419	1332	78	0.146
strength pareto evolutionary algorithm	1580	370	1980	412	973	63	0.131
genitor algorithm	949	372	958	99	791	0	0.105
messy genetic algorithm	982	346	826	93	719	20	0.099
gene expression programming	3240	211	1100	139	1257	98	0.086
interactive genetic algorithm	2500	246	889	118	681	52	0.084
sequential niching genetic algorithm	0	296	0	0	1	0	0.071
vector evaluated genetic algorithm	569	233	758	123	356	4	0.069
shuffled complex evolution	1120	139	1210	26	928	8	0.048
cellular genetic algorithm	515	132	302	69	343	21	0.041
double diploid genetic algorithm	0	166	0	0	0	0	0.04
coevolutionary genetic algorithm	662	103	312	58	188	37	0.034
restricted tournament selection	277	84	252	50	211	2	0.026
niching genetic algorithm	470	49	322	33	256	31	0.019
linkage learning genetic algorithm	219	50	139	31	201	4	0.016
covariance matrix adaptation evolution strategy	273	35	210	45	197	18	0.014
island genetic algorithm	319	39	218	24	171	9	0.014
crowding genetic algorithm	136	38	56	12	72	1	0.011
cooperative coevolutionary genetic algorithm	256	22	79	17	27	8	0.008
indicator based evolutionary algorithm	62	19	60	26	21	1	0.006
deterministic crowding algorithm	31	20	34	7	31	1	0.005
chc genetic algorithm	61	13	49	7	36	4	0.004
genetic algorithm with punctuated equilibria	11	15	19	1	8	1	0.004
dependency structure matrix driven genetic algorithm	35	15	36	10	61	0	0.004
fitness sharing genetic algorithm	62	3	20	4	5	2	0.001
speciation genetic algorithm	50	1	1	0	6	1	0.0
probabilistic crowding genetic algorithm	3	1	3	2	2	0	0.0
nested evolution strategy	12	2	4	1	6	1	0.0

Table 5: Raw results for the 'evolutionary algorithms' taxonomic kingdom, ordered by score, descending.

Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
particle swarm optimization	49200	722	24600	2151	15566	3999	0.861
ant system	9470	536	7400	940	4903	155	0.27
ant colony system	6620	566	4260	609	2587	201	0.231
antnet	4060	457	1320	176	1185	29	0.146
max-min ant system	1880	256	1650	282	990	51	0.096
multiple ant colony system	501	122	451	81	307	2	0.038
pareto ant colony optimization	171	30	127	35	73	1	0.01
rank-based ant system	200	31	142	18	61	7	0.01
bees algorithm	870	13	163	10	214	10	0.009
population-based ant colony optimization	137	20	64	12	311	4	0.008
bee colony optimization	379	17	115	19	52	15	0.008
elitist ant system	144	13	101	11	61	2	0.005
bacterial foraging optimization algorithm	120	6	51	7	31	5	0.003
waves of swarm particles	20	5	13	7	16	2	0.002
virtual bee algorithm	20	7	18	7	3	1	0.002
firefly algorithm	126	7	42	4	38	1	0.002
honey bee algorithm	73	8	27	5	7	0	0.002
repulsive particle swarm optimization	154	6	32	2	39	3	0.002
artificial bee colony optimization	23	6	14	1	6	2	0.002
crowding population-based ant colony optimization	3	1	2	0	0	0	0.0
monaco: multi-objective network optimization based on aco	1	2	0	0	0	0	0.0
wasp swarm algorithm	10	2	4	2	6	0	0.0

Table 6: Raw results for the 'swarm algorithms' taxonomic kingdom, ordered by score, descending.

Name	g-web	g-book	g-scho	Springer	Scirus	IEEE	Score
random search	161000	1509	32400	2855	121802	405	1.88
tabu search	42800	1399	35400	4208	36146	1035	1.076
reactive search optimization	53	3543	4	0	45	0	0.854
hill climbing search	2340	648	2950	320	2869	10	0.206
scatter search	4290	450	3300	544	2987	59	0.182
adaptive random search	928	627	888	70	653	17	0.165
variable neighborhood search	3150	420	2750	457	3090	41	0.164
reactive tabu search	1640	431	1390	186	1362	44	0.134
greedy randomized adaptive search procedure	1400	398	1730	279	1203	28	0.129
guided local search	1700	320	1320	242	1987	12	0.111
iterated local search	2050	287	1650	318	1730	28	0.109
variable depth search	563	212	418	64	417	5	0.06
directed random search	377	222	394	46	231	7	0.06
stochastic gradient descent algorithm	508	204	419	38	403	25	0.058
random-mutation hill climbing	434	138	414	47	366	6	0.041
random restart hill climbing	257	35	106	6	250	0	0.011
hill climbing with learning	108	30	73	14	80	0	0.009
stochastic diffusion search	286	15	95	9	549	2	0.008
random bit climbing	32	21	25	3	12	0	0.005
mutation hill climbing algorithm	42	16	34	7	15	1	0.004
parallel hill-climbing algorithm	67	6	26	5	36	3	0.002
localized random search	36	9	25	3	14	1	0.002
multiple restart algorithm	2	1	2	1	1	0	0.0

Table 7: Raw results for the 'stochastic algorithms' taxonomic kingdom, ordered by score, descending.

4 Analysis

This section analyses the results of the assessed algorithms for inclusion in the Clever Algorithms project. The analysis focuses on the limitations of the adopted algorithm selection strategy, and the identification of potential areas for improvement.

4.1 Algorithm Names

Not all algorithm names are created equal, and as such, using algorithm names alone to compare the popularity of algorithmic techniques provides an opportunity for improvement. For some algorithm names, such as the 'genetic algorithm', the name represents a base name (and typically a base technique) for a family techniques. This results in artificially inflated popularity scorings. Additionally, some techniques are renamed one or more times throughout their life, and some techniques are better known by abbreviations or acronyms then their full unabbreviated names.

Names are commonly used to delineate areas of research, not necessarily specific algorithmic techniques intended for application. For example, there is no 'canonical genetic algorithm', there is instead a general procedure and a number of variant operators that may be applied at each step. Although, in this case, there does happen to be a 'simple genetic algorithm' that could be considered canonical, and separately, a 'canonical genetic algorithm' may be deduced from the best practice realization of each operator in the procedure. This is an exemplar case that highlights both the confusion in the field and an opportunity that that Clever Algorithms project may help address.

Finally, in the same manner that the popularity of algorithms are tested by assessing their names, the popular association of an algorithm and a field or taxonomic kingdom may be determined using a data-driven method. A set of field names may be defined and the popularity of each algorithm name with each field name may be calculated, scored, and ranked. A similar method may be used to select one algorithm name for a given technique from a collection of aliases and abbreviations.

4.2 Algorithm Scoring

The number of results returned from each search engine is an approximation with an unknown level of error, both with regard to the service reporting hits from its known index, and to the extent that index is not complete. Another area of concern with the collected measures is that it does not take into account when algorithms were released. For example, those techniques released many years ago will likely be more popular than those released very recently. This is likely a desirable effect of the method, although may mask new algorithms whose popularity is rising and old algorithms whose popularity is falling. This may be addressed by collecting algorithm popularity trends over an extended period of time (years).

Finally, the measures are absolute quantities that are not likely to be comparable between search domains. This was tentatively addressed by normalizing the results for each domain before combining them into a single score. This may be extended through the introduction of weighting factors in the calculation of the algorithm scoring, such as giving additional weight to the popularity of an algorithm name in published book search domains.

Ideally, popularity would not be the basis for algorithm comparison. Some more useful measures may include efficiency, efficacy, quality of research, and scope of innovation. These are all properties that are difficult to quantify and ultimately difficult to measure using objective and automated means.

4.3 Algorithm Selection

Algorithm popularity may be useful for determining the ordering by which algorithms are described in the project (for marketing and discoverability purposes for example), although may

not be appropriate for the ordering by which algorithms are listed in deliverable outcomes of the project, such as a book or website. A more appropriate ordering in these cases may be a logical progression through algorithm complexity, determined by difficulty (time) required to understand and/or implement a given algorithm.

5 Selected Algorithms

This section provides a listing of 50 algorithms selected for description in the Clever Algorithms Project. The presentation of this list of algorithms is partitioned into taxonomic kingdoms. The algorithms are presented in a suggested ordering, both with regard to the kingdoms and the algorithms within each kingdom (roughly in order of increasing complexity).

A simple high-level filtering method was applied, where those kingdoms with ≤ 10 algorithms were constrained to 5, and those with > 10 algorithms were constrained to 10 algorithms.

This listing of algorithms is not final, rather, it is a preliminary draft based on the best information available at the time of writing. It is expected that the specific algorithms described in the Clever Algorithms project will differ as the project unfolds. This preliminary listing is suggested as a starting point for breakdown for the project deliverable outcomes, such as book chapters and sections.

5.1 Stochastic Algorithms

The algorithms are presented from random search, through variations with neighborhoods and adaptation.

1. Random Search
2. Adaptive Random Search
3. Hill Climbing Search
4. Guided Local Search
5. Variable Neighborhood Search
6. Reactive search optimization
7. Greedy Randomized Adaptive Search
8. Scatter Search
9. Tabu Search
10. Reactive Tabu Search

5.2 Physical Algorithms

A focus on algorithms inspired by physical and social systems with an upfront focus on the simulated annealing algorithm.

1. Simulated Annealing
2. Adaptive Simulated Annealing
3. Memetic Algorithm
4. Extremal Optimization
5. Cultural Algorithm

5.3 Evolutionary Algorithms

A focus on the classical evolutionary algorithms, followed by some newer and popular approaches and some for multiple objective optimization. I'd really like to present many more algorithms from this kingdom such as gene expression, messy, cellular, and niching approaches.

1. Genetic Algorithm
2. Genetic Programming
3. Evolutionary Programming
4. Evolution Strategies
5. Learning Classifier System
6. Differential Evolution
7. Grammatical Evolution
8. Non-dominated Sorting Genetic Algorithm
9. Strength Pareto Evolutionary Algorithm
10. Island Population Genetic Algorithm

5.4 Probabilistic Algorithms

Those techniques concerned with managing a probabilistic model of a domain, most of which are extensions of algorithms from evolutionary computation.

1. Cross-entropy method
2. Population-based incremental learning
3. Probabilistic Incremental Program Evolution
4. Compact Genetic Algorithm
5. Extended Compact Genetic Algorithm
6. Bayesian Optimization Algorithm
7. Hierarchical Bayesian Optimization Algorithm
8. Univariate Marginal Distribution Algorithm
9. Bivariate Marginal Distribution Algorithm
10. Gaussian Adaptation

5.5 Swarm Algorithms

Approaches related to the principle of collective intelligence such as particle swarm, ant colony, bee, wasp, and bacterial algorithms.

1. Particle Swarm Optimization
2. AntNet
3. Ant System
4. MAX-MIN Ant System
5. Rank-Based Ant System
6. Ant Colony System
7. Multiple Ant Colony System
8. Population-based Ant Colony Optimization
9. Bees Algorithms
10. Bacterial Foraging Optimization Algorithm

5.6 Immune Algorithms

Algorithms from the field of Artificial Immune Systems.

1. Clonal Selection Algorithm
2. Negative Selection Algorithm
3. Artificial Immune Recognition System
4. Immune Network Algorithm
5. Dendritic Cell Algorithm

6 Conclusions

This report proposed a data-driven methodology for selecting algorithms based on popularity. This methodology was applied for the proposes of selecting algorithms to describes in the Clever Algorithms project. The raw results were presented for posterity, and a preliminary set of 50 algorithms were selected for the project. This listing of algorithms is not expected to be maintained within this document, although is expected to represent a first draft for breakdown for the deliverable outcomes of the Clever Algorithms project, such as a book and website.

There is desire to describe more than 50 algorithms, especially including more approaches from the evolutionary and swarm kingdoms. Such additions may be made after the first 50 algorithms are completed and/or after the first draft of the outcome deliverables (book and website) are completed. For example, a second edition may be released with an additional 20 algorithm descriptions as well as fixed errata from the first edition.

The listing focused on so-called 'unconventional optimization algorithms', intentionally neglecting model-based algorithms such as artificial neural networks and fuzzy logic systems. If the Clever Algorithms project proves successful, the effort may consider the preparation of a second volume (book) that shifts the focus of the project from optimization algorithms to model-creating algorithms and address neural network, fuzzy logic, and some statistical machine learning algorithms that fit within the scope of the project.

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.