"Success Effort" for Performance Comparisons

Matthew Walker, Howard Edwards, and Chris Messom
Massey University, Auckland, New Zealand
{m.g.walker, h.edwards, c.h.messom}@massey.ac.nz

ABSTRACT

This paper looks at the production of a confidence interval for a statistic we rename *success effort.*

Categories and Subject Descriptors: I.2.2 [Artificial Intelligence]: Automatic Programming

General Terms: Experimentation, Measurement, Performance

Keywords: Genetic Programming, Performance Statistics, Confidence Intervals, Success Effort, Hit Effort

1. DEFINITION

"Success effort" was a measure introduced by Miller and Thomson [3] under the name "hit effort". They defined a hit as a run that found a solution. Given that Koza had earlier defined the term "hit" as a run having success in a *portion* of the given problem [1], we felt "hit effort" was a somewhat confusing name. Consequently, we have renamed Miller and Thomson's measure as "success effort".

Success effort measures the expected number of generations before a solution will be found. It may be calculated from a collection of runs by $\frac{\text{mean}(g)}{p}$, where g is the vector of generations at which the runs terminated (generations-to-termination) and p is the proportion of runs that found a solution.

Koza's minimum computational effort [1] can be seen to answer the question "what is the minimum number of generations that must be executed for a solution to be found 99% of the time, if the optimal number of runs are performed concurrently?". Success effort answers the question "given a sequence of runs, what is the average number of generations that will be executed before a solution will be found?".

2. CONFIDENCE INTERVALS

Table 1 introduces a method based on the simulation of the two likelihood functions for the true mean generation and the true probability of success.[1] We have shown the method to be: quick to evaluate; to have good coverage; and to frequently produced tighter confidence intervals than Koza's minimum computational effort statistic. It

[1]Code for the implementation of this algorithm and complete results for all experiments are available at `www.massey.ac.nz/~mgwalker/CompEffort`.

1. Obtain n independent runs. Count the number of successes n_s, the number of failures n_f. $p = \frac{n_s}{n}$. Extract the vector of generations for the successful runs g_s, and the vector for the failures g_f.
2. Produce $B = 10,000$ random variables that are normally distributed with mean equal to mean(g_s) and with a standard deviation of $\frac{\text{sd}(g_s)}{\sqrt{n_s}}$. Label these G_s.
3. Produce B random variables that are normally distributed with mean equal to mean(g_f) and with a standard deviation of $\frac{\text{sd}(g_f)}{\sqrt{n_f}}$. Label these G_f.
4. $G = pG_s + (1-p)G_f$
5. Produce B random variables that follow a beta distribution with parameters $\alpha = pn+1$ and $\beta = (1-p)n+1$. Label these P.
6. Find the 2.5% and 97.5% quantiles of $\frac{G}{P}$. These are the limits of a 95% confidence interval for the true value of the success effort statistic.

Table 1: Simulation algorithm to produce 95% confidence intervals for the success effort statistic.

is a method that corresponds to a Bayesian statistical approach [2].

The simulation algorithm works correctly even if g_f is comprised of multiple instances of a single number, as would be obtained if the traditional GP approach were used where the maximum generation was set to 50 generations. If only one success or one failure was observed, then the standard deviation of that observation should be considered to be zero. If zero failures were observed, then the cut-off generation should be used in place of mean(g_f) and zero should be used in place of sd(g_f). Finally, if zero successes were observed, the use of this statistic should not even be considered!

3. REFERENCES

[1] J. R. Koza. *Genetic Programming.* MIT Press, 1992.
[2] P. M. Lee. *Bayesian Statistics: An Introduction.* Arnold, 2nd edition, 1997.
[3] J. F. Miller and P. Thomson. Cartesian genetic programming. In *Proceedings of EuroGP*, 2000.