GAMBLER'S RUIN MODEL AND GA

Z. Konfršt

Dept. of Cybernetics, FEE, Czech Tech. University

Abstract

In order to estimate the correct size of population, estimations of population sizing have been used in the genetic algorithms (GAs). The estimation considers a test function being optimized, a representation of individuals and a character of used operators. By means of the estimation model the right population size with taking into account the final overall quality of individuals is identified. This article extends the Gambler's ruin model (GRM) by a new equation for convergence time.

1 Introduction

There have been several significant attempts on the population sizing. Holland [4, 5] idealized the process in a GA as a cluster of parallel and interconnected 2^k-armed bandit problems. The Holland's bandit problem was extended by De Jong [6] in the equation for population sizing. It was representing the basic noise-to-signal equation. Although the equation was not extended in the De Jong's dissertation, it gave a first assessment on the population sizing problem.

In [1], the statistical decision theory was exercised. The authors were modeling a GA run as competitions between the best and the second best BBs (building blocks). Every BB was built-up from m partitions of order k. Order k means how many cells are in an every single partition. Two BBs were represented with their mean fitnesses and fitness variances. One of the outputs of the research [1] was that the probability of the right choice in a single trial of a problem (with m equally sized partitions) is

$$p = \Phi(\frac{d}{\sqrt{2m'}\sigma_{bb}}), \tag{1}$$

where Φ is the cumulative normal distribution function (CDF), σ_{bb}^2 is the average BB variance of the partition that is being considered, m' is a number of competing partitions ($m' = m-1$) and d is the fitness difference between the best and the second best BBs.

In [2], the authors used the well-known the gambler's ruin problem (also one-dimensional random walk) with absorbing barriers at $x = 0$ and $x = \beta$ representing convergence to the wrong (0 represents gambler's bankruptcy) and the right solutions ($\beta, \beta > 0$ representing a win of all opponent's money), respectively. Variables p and $1-p$ are probabilities that the best BB takes over the second best or vice versa. An initial seed defined as $x_0 = \frac{n}{2^k}$ (of k order) was used, and the functional analogy between GAs and the gambler's ruin problem supposed several conditions as follows: First, the competition takes place between the best and the second best BBs in a partition. Second, crossover and mutation do not destroy significant number of BBs. Third, boundaries of the random walk are absorbing. The well-known equation[1] [3] was employed to get the quality of a solution as the number of $X \in m$ partitions converged to the right BBs.

$$P_{bb} = \frac{1-(\frac{1-p}{p})^{n/2^k}}{1-(\frac{1-p}{p})^n}, \tag{2}$$

[1]A result from the theory of random walk is that a particle will be eventually captured by the absorbing barrier at $x = \beta$. The captured particle in the theory of random walk that reflects the partition of interest contains β copies of the correct BBs in the GA world.

where P_{bb} is a percentage of well-converged partitions. The initial population size is $x_0 = n/2^k$, because it represents expected number of copies of the correct BB in a randomly generated population n. Probability p is defined above. It holds $p > 0.5$ while the mean fitness of the best BB is greater than the second best BB. The probability p must be obtained from the equation (1).

We simplify here the equation (2) above under conditions $n \gg x_0$ and $p > 0.5$. Under these conditions and while n grows the denominator goes to 1 and one simplifies the equation to

$$P_{bb} \approx 1 - (\frac{1-p}{p})^{n/2^k}. \quad (3)$$

Comparison between the original and simplified versions of the P_{bb} equations is depicted in Figure 1. We show that there are sizable discrepancies for small p, k and population sizes n. On the other hand, the discrepancies disappear when the parameters p and k are bigger as it is demonstrated. The simplified version of the equation holds for $\{n \geq 25,\ p \geq 0.520,\ k = 1;\ n \geq 15,\ p \geq 0.585,\ k = 2;\ n \geq 5,\ p \geq 0.65,\ k \geq 3\}$, sufficiently. In the other cases, one must use the original version (2).

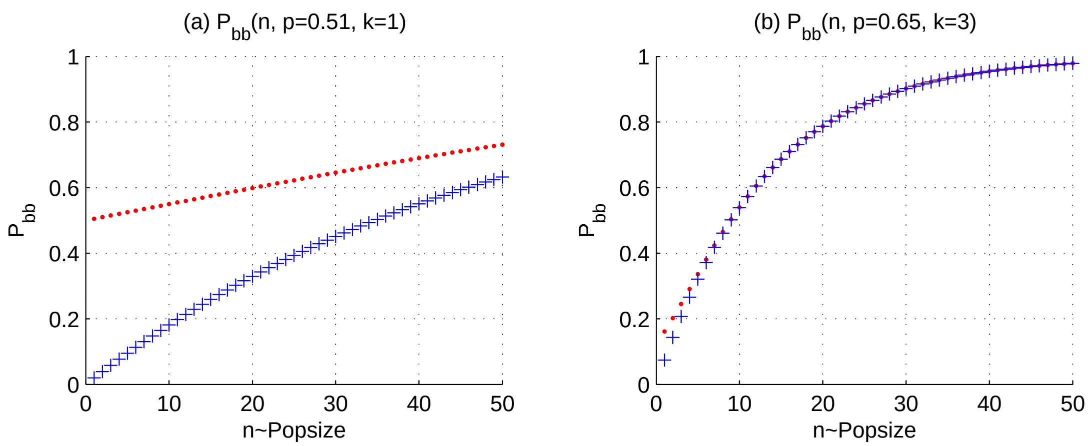

Figure 1: Figures depicts variance between the original P_{bb} and the simplified P_{bb} for two cases. The original one is represented with dots and the simplified one with crosses. One can see that figures differ in the order of BB k and the probability p. The simplified equation holds for $n \geq 15$, $p \geq 0.520$ and $k \geq 1$ fairly well.

From this simplified equation one can derive an equation for the population size (n) such as

$$n = \frac{2^k . \ln(1 - P_{bb})}{\ln(\frac{1-p}{p})}. \quad (4)$$

The equation (4) estimates the population size (n) for a GA problem. The probability p from the equation (1) has been dependant on the number competing partitions (m'), a problem being optimized (σ_{bb}) and the fitness difference (d). Different population sizes (n) would reflect different proportions of BBs (P_{bb}). Three characteristics which demonstrate the dependence given by the equation above are in Figure 2. There is shown the dependency of estimated population size on three parameters such as p, k and P_{bb}.

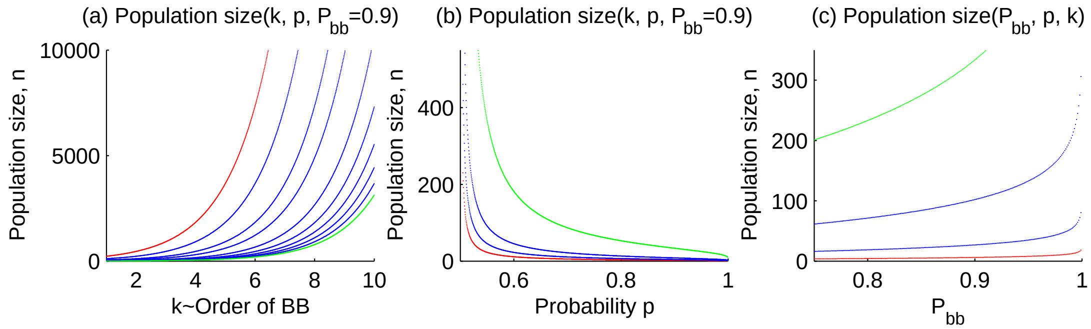

Figure 2: Figures depicts the behavior of the population sizing equation (4) in several views. Red and green lines are the most simple and most complex problems, respectively. Blue lines show problems of complexity somewhere between these two extremes. One can see that the estimate of the population size is not linear in any case. The three graphs show reliance of the Popsize on order of BB, probability p and the percentage of well-converged partitions P_{bb} (from left to right). The first graph shows increasing p as k order of BB increases. The second graph depicts curves related to decreasing k. The third graph illustrates decreasing k and p.

2 Time convergence of GA

In the previous section, the estimation of the correct population size of GA was derived. We keep following an analogy between Gambler's ruin problem (GRP) and GA as well for the time convergence of GA. If we can say that we have some estimation when a game terminates in GRP, we can say that this holds for a GA when the GA converges.

In [3], there is an equation that describes the expected duration of the game D_z in the classical Gambler's ruin problem (5),

$$D_z = \Big(\frac{z}{q-p} - \frac{a}{q-p}\Big).\Big(\frac{1-(q/p)^z}{1-(q/p)^a}\Big), \tag{5}$$

where z is the initial capital of a gambler, a gain of an opponent capital is $a-z$, the total capital involved is a, a gambler wins one game with the probability q and losses with the probability p. And it holds $p+q=1$. If $p>q$, the game is favourable to the gambler and vice versa.

We take the equation (5) and we change it for GA. The number of correct BBs (the initial capital) is $z \sim n/2^k$ and all converged BBs is $a \sim n$. The duration of the game D_z is replaced by the number of generations T_{conv} in the GA case. Under the condition $p \neq 0.5$ and after simple algebraic adjustments we receive the equation (6). The equation describes the expected number of generation till a GA convergence.

$$T_{conv} = \Big(\frac{n/2^k - n}{1-2p}\Big).\Big(\frac{1-((1-p)/p)^{n/2^k}}{1-((1-p)/p)^n}\Big) \tag{6}$$

As one can easily see the second part of the equation (6) equals (2), so we can simplify P_{bb} as previously in (3),

$$T_{conv} = \Big(\frac{n/2^k - n}{1-2p}\Big).\Big(1-(\frac{1-p}{p})^{n/2^k}\Big). \tag{7}$$

The second part of the equation (7) is replaced by P_{bb}, so we get

$$T_{conv} = A_c.P_{bb}, \quad (8)$$

where $A_c = \frac{(n/2^k - n)}{1-2p}$ and $P_{bb} = 1 - (\frac{1-p}{p})^{n/2^k}$, n is the population size, k is order of BB and p is the probability of the right choice. This equation (8) gives the time estimation when a GA converges.

A problem to be optimized by a GA has to be somehow encoded into the GA. The probability p, the proportion of correct BBs P_{bb} and the order of BBs k are input variables to reach the estimate of correct population size n for the optimized problem in the equation (4). Similarly, based on the previous input variables and the estimate of the population size, the time to convergence of a GA T_{conv} can be calculated from equation (7).

3 Experiments

The experiments utilized the Genetic Algorithm and Direct Search toolbox, which is a dedicated GA related toolbox in Matlab R2006a. The toolbox is a GA toolkit to analyze, test and examine a GA and its parameters. The toolbox also helps to visualize the results and various data conversion between Matlab and other calculation software. The experiments were run on a commodity PC computer with Intel Celeron 2.8GHz, 1GB RAM and Windows 2000 SP4 (EN).

3.1 GA parameters

Table 1: Tabulated GA parameters for 5 test cases (TC). The cases vary in the parameters such as Population representation (PopType), Selection, Crossover and Mutation. The parameters (or their values in brackets), which are not in this table, used predefined default values. As far as the numeric values are concerned, tournament(4) means a tournament size of 4, heuristic(1.2) is a default ratio of 1.2, gauss(0.5/0.75)–the default values of scale and shrink parameters are 0.5 and 0.75 and finally, uniform(0.01) is the probability rate of being mutated 0.01.

Parameter	*TC1*	*TC2*	*TC3*	*TC4*	*TC5*
PopType	doubleVector	bitString	bitString	doubleVector	bitString
Selection	stochuniform	roulette	remainder	tournament(4)	uniform
Crossover	scattered	singlepoint	twopoint	heuristic(1.2)	intermediate
Mutation	gauss(0.5/0.75)	uniform(0.01)	adaptfeas	uniform(0.01)	gauss(0.5/0.75)

In Table 1, five test cases were defined – TC1, TC2, TC3, TC4 and TC5. The cases vary in Population representation (PopType), Selection, Crossover and Mutation. More possible combinations of test cases were possible but these had been selected. Five selection schemes and crossover operators were chosen. Three mutation operators with non-zero probability were set in all experiments. In Table 2, there are GA parameters that did not change during the experiments at all. The GA parameters in both Tables are quite self-descriptive as well as well-known in the GA community and would not be explained in detail.

3.2 Experimental setup

The GRM was parameterized that the source of diversity is the initial random population. It was used the Matlab's *Rand(n)* function to generate uniformly distributed random numbers.

Every single run of GA used a new initial random population to evolve the solution. Termination of each run was achieved when the population converges completely or in the more difficult fitness function after bound on the number of generations is reached. Due to no mutation, the complete convergence was possible. All the results are the averages of 3 independent runs of a simple GA.

We made use of several test functions with various degrees of computational complexity. In Table 3 table below shows seven test functions for GA.

3.3 Population sizing

The population sizing problem has been investigated and experimented with in various scientific papers such as [1, 2].

Fig. 3 confirms the population sizing theory. The theory says there exists a population size that helps to solve the optimized problem sufficiently. Notice that from a "critical" population size $n_{crit} \sim 51$, the curve approaches optimum very quickly. The critical population size is estimated by the population sizing models.

3.4 Time convergence of GA

In the theoretical part, we had proposed the equation (7) that gives an estimate of the time convergence for GA (T_{conv}). In next subsection, we identify if the equation, under several conditions, reflects the GA run. Experiments were executed to reveal prospective flaws and inconsistencies.

3.5 Results

In Figure 4, one can see the T_{conv} volatility in the GA run such as short and narrow peaks. The largest peaks are at $n = 130$ and $n = 260$. Variances of the time convergence are skewed to upside. The trend of the curve is asymptotically robust.

In Figures 5-9, the time convergence for GA T_{conv} has been shown based on population size n for five test cases each. Every test case corresponds to two figures (left $n \leq 300$, right

Table 2: Tabulated GA default parameters for 5 test cases (TC). These parameters were default for all GA experiments.

Parameter	*Value*
FitnessScale	rank
CreationFcn	uniform
PopulationSize	$5 - 1000$
EliteCount	2
CrossoverFrac	0.8
Generations	300
StallGenLimit	50
StallTimeLimit	50
InitialPenalty	10
PenaltyFactor	100

Table 3: Selected GA test functions and their parameters for experimental verifications. The name, equation, definition interval **x** and the global minimum min(**f**(.)) of GA test functions are tabulated.

f(.)	*Name*	*Equation*	**x**	min(**f(.)**)
f1	Zeromin	$f(\mathbf{x}) = \sum_{i=1}^{100} x_i$	$< 0, 1 >^{100}$	0
f2	DeJong's1	$f(\mathbf{x}) = \sum_{i=1}^{3} x_i^2$	$< -5.12, 5.12 >^3$	0
f3	DeJong's2	$f(\mathbf{x}) = 100.(x_1^2 - x_2)^2 + (1 - x_1)^2$	$< -2.048, 2.048 >^2$	0
f4	Rastrigin's	$f(\mathbf{x}) = 10.n + \sum_{i=1}^{n}(x_i^2 - 10.\cos(2.\Pi.x_i))$	$< -5.12, 5.12 >^2$	0
f5	Ackley's	$f(\mathbf{x}) = 20 + e^{x_1} - 20.e^{-0.2.\sqrt{((1/n).\sum_{i=1}^{n} x_i^2)}}$ $-e^{1/n}.\sum_{i=1}^{n}\cos(2.\Pi.x_i)$	$< -32.768, 32.768 >^2$	~ 2.98
f6	Eggholder	N/A	$< 0, 80 >^2$	-33
f7	Shu	N/A	$< -2.0, 2.0 >^2$	~ -184

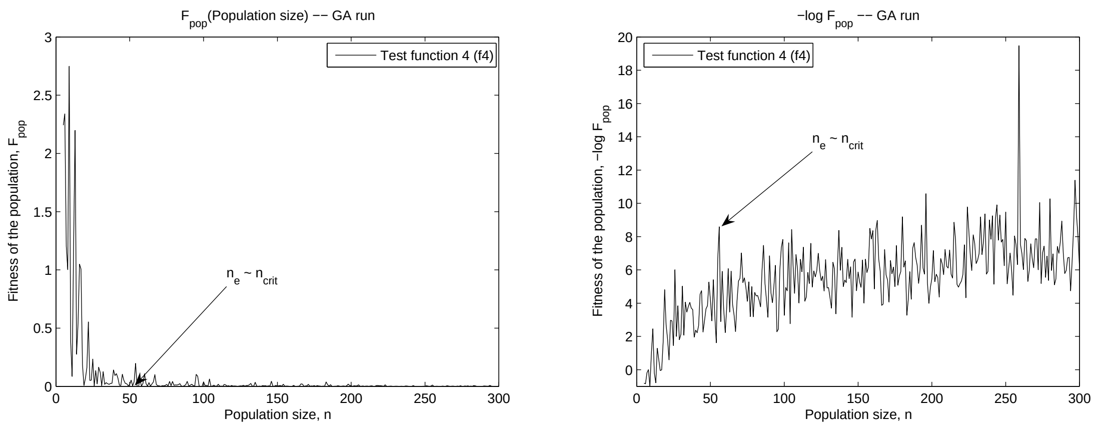

Figure 3: The fitness of population F_{pop}. Both figures show the same F_{pop} based on population sizes but with different y-axes. Figures are F_{pop} (left) and $-log(F_{pop})$ (right). It is easy to see that there exists a "critical" population size $n_{crit} \sim 51$, which gives a very favorable result. The test function was **f4.**

$n \leq 1000$). The functions are of various complexities, therefore they show variations and some do fluctuate very wildly.

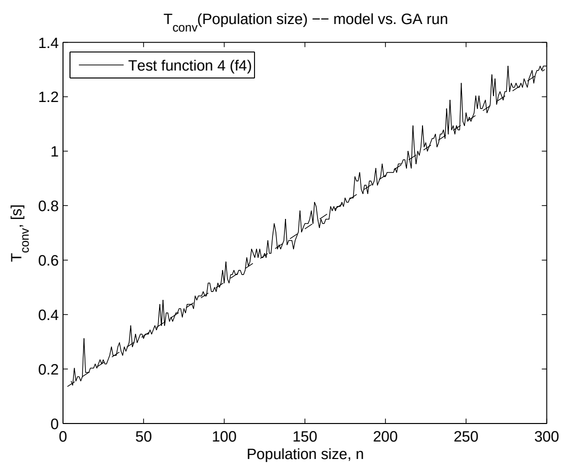

Figure 4: The T_{conv} model vs. the volatility of a GA run. Figure shows T_{conv} based on population sizes. The test function was f4. In detail, one can see small peaks and discrepancies in the graph instead of expected linearity. GA parameters were set for TC1.

4 Discussion

In Figure 4, one see small spikes of various sizes. The spikes are more above the line than below. This figure shows the real experimental data after one independent GA run. When one adds more GA runs and calculates averages, the depicted results are far more smooth and close to the line generated by the model.

The convergence time T_{conv} was tested in a GA domain. However, we experienced several tiny discrepancies, the experiments provided fair robust support for the convergence time equation. Some functions also showed increased variance and volatility, but we see the results very satisfactory.

In Table 4, a short summary of our experimental results is scored. In Figures 5-9, several variables were evaluated. The variables were adequacy to the model ("model"), successful run ("succ"), trend following ("trend"), volatility ("vol") and sequence order of the curves ("order").

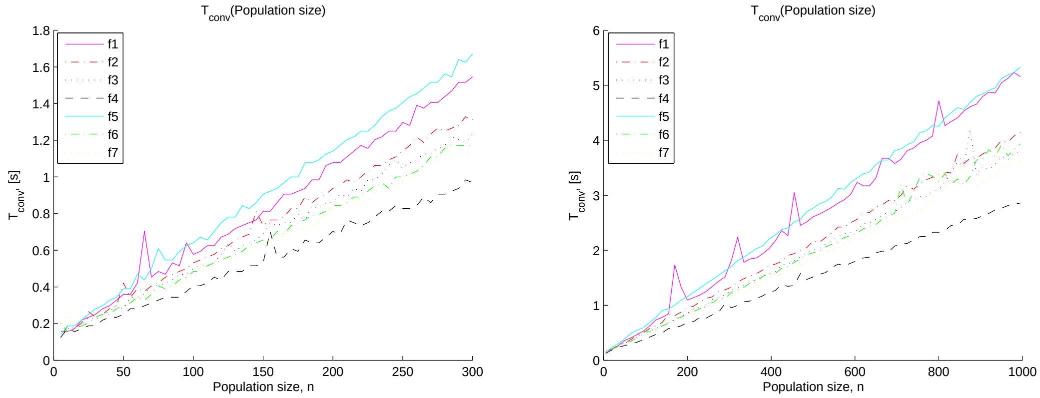

Figure 5: Test case one – TC1. Figure depicts the functional dependency of T_{conv} on the population sizes $n = 300$ (left) and 1000 (right) for TC1. The test functions were f1-f7.

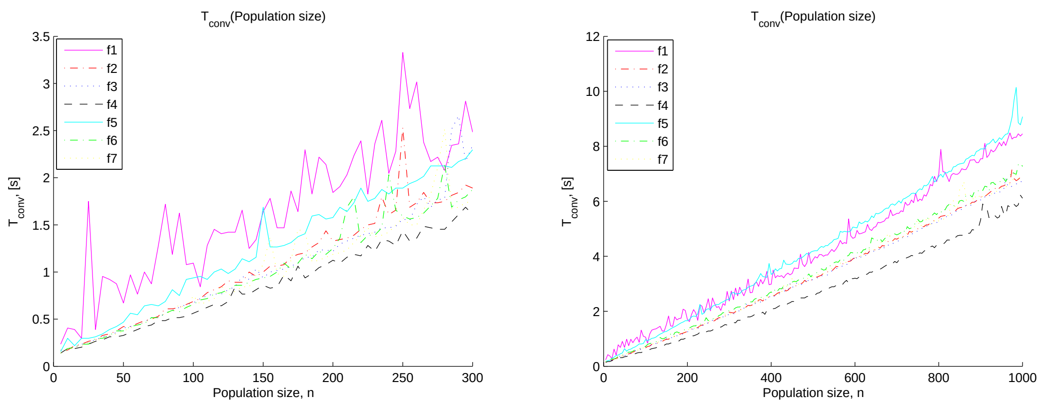

Figure 6: Test case two – TC2. This figure depicts the functional dependency of T_{conv} on the population sizes $n = 300$ (left) and 1000 (right) for TC2. The test functions were f1-f7.

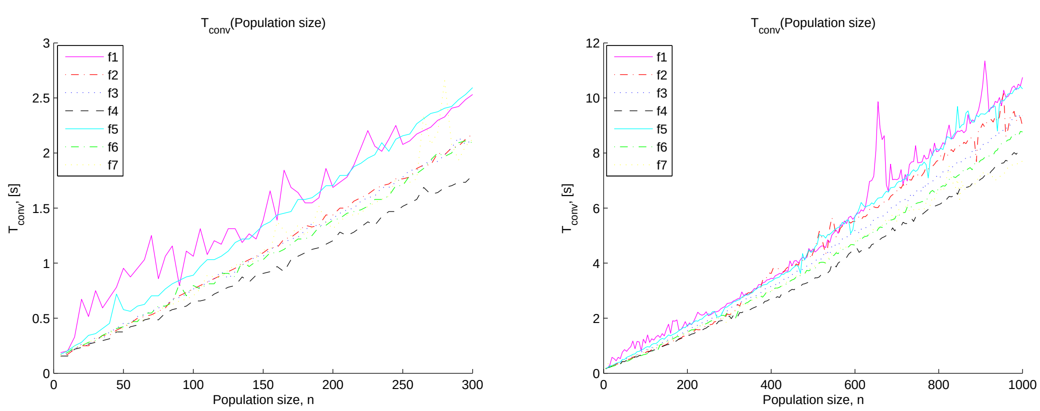

Figure 7: Test case three – TC3. Figures depicts the functional dependency of T_{conv} on the population sizes $n = 300$ (left) and 1000 (right) for TC3. The test functions were f1-f7.

The volatility is the only variable when a lower result value is better (negative variable). For the other variables (positive ones), it is quite opposite, higher values mean more favorable results. Evaluation scores were only 0 (negative) or 1 (positive) for each test function in each test case and these scores were summarized for each test case.

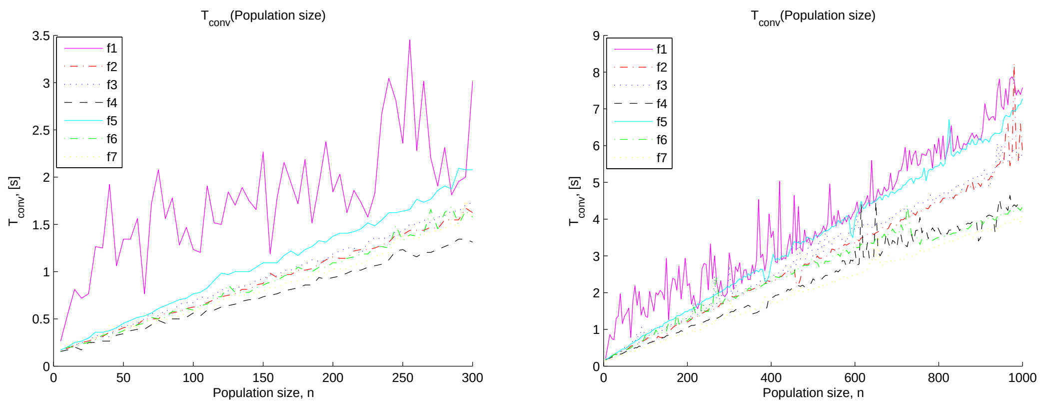

Figure 8: Test case four – TC4. Figures depicts the functional dependency of T_{conv} on the population sizes $n = 300$ (left) and 1000 (right) for TC4. The test functions were f1-f7.

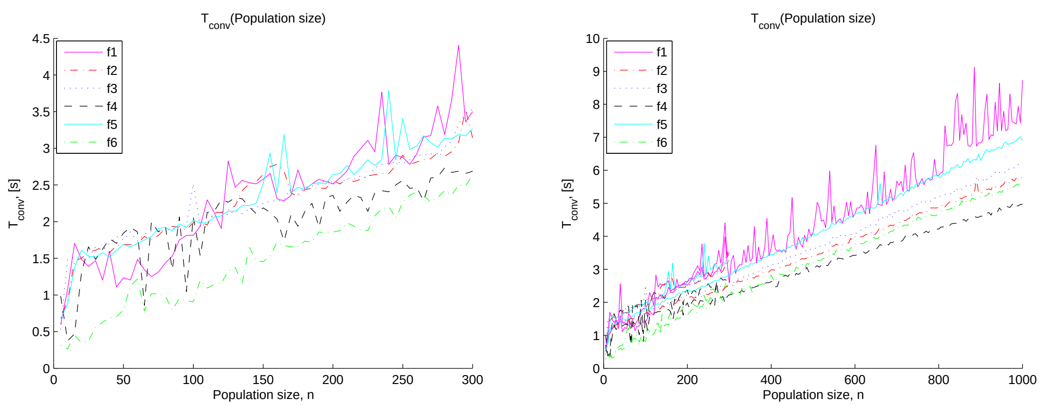

Figure 9: Test case five – TC5. Figures depicts the functional dependency of T_{conv} on the population sizes $n = 300$ (left) and 1000 (right) for TC5. The test functions were f1-f6. The test function f7 did not run correctly and was not included.

We argue that experimental results were quite positive. All positive variables scored $S_\epsilon \geq 91.18$ and better. The only negative one, volatility, was under $S_\epsilon \leq 26.47$. That means

Table 4: A short summary of experimental results. In the figures, several variables were evaluated. The variables were adequacy to the model ("model"), successful run ("succ"), trend following ("trend"), volatility ("vol") and sequence order of the curves ("order"). Only volatility is a negative variable, others are positive. Evaluation scores were only 0 or 1 and these were summarized for each test case. S_ϵ is calculated as $Score/S_{max}$.

TC	*model*	*succ*	*trend*	*vol*	*order*
TC1	7/7	7/7	7/7	1/7	7/7
TC2	7/7	7/7	7/7	2/7	7/7
TC3	6/7	7/7	7/7	1/7	6/7
TC4	5/7	6/7	6/7	3/7	6/7
TC5	6/6	6/6	6/6	2/6	5/6
$Score/S_{max}$	31/34	33/34	33/34	9/34	31/34
S_ϵ[%]	91.18	97.06	97.06	26.47	91.18

that the volatility of test results was higher in 26.5% cases. One of the main reasons of higher volatility results is that only 3 independent runs of GA were exploited. When we averaged the same results over 5, 10 and 15 independent runs of GA, the volatility decreased gradually, but calculations took very long. Nevertheless, the quality of experimental results with 3 independent runs was good enough and sufficient in our opinion.

The experiments evidenced that there is a sufficiently strong linkage between the duration of the game D_z in the Gambler's Ruin Problem and the convergence time T_{conv} in GA. The linkage can be further developed and exploited in the GA domain.

5 Conclusion

The paper explains basics of the Gambler's ruin model (GRM). GRM is a model to propose the estimate of the correct population size for an optimized problem. This model was extended by a new equation (7) for the estimate of convergence time. So now, the model proposes not only the estimate of population sizes but also the estimates of convergence time for GA measured as the number of generations needed.

The model was examined under several GA test functions, variously parameterized GA and the model showed good robustness. The comparison between the extended model and the verification runs of GA as well as the results from the experiments reflected closeness to the theoretical proposal. From the practical point of view, the extended GRM provides the estimate of a population size and of time convergence for GA now. The estimates can be calculated one after the other or quite independently. With extended GRM, GA practitioners have one tool to design and to parameterize efficient GAs.

References

[1] D. E. Goldberg et al. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

[2] G. Harik et al. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Proc. of the IEEE International Conference on Evolutionary Computation*, pages 118–125, 1997.

[3] W. Feller. *An introduction to probability theory and it's applications*, volume 1. Wiley, Essex, U.K., second edition, 1966.

[4] J. H. Holland. Genetic algorithms and the optimal allocations of trials. *SIAM Journal of Computing*, 2(2):88–105, 1973.

[5] J. H. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, U.S.A., 1975.

[6] K. A. De Jong. *An analysis of the behaviour of a class of genetic adaptive systems.* University of Michigan, Ann Arbor, U.S.A., 1975.

Zdeněk Konfršt
Email: zdenek.konfrst@seznam.cz
Address: Dept. of Cybernetics, FEE, Czech Tech. University, Technická 2, 166 27 Prague 6