

Domino Convergence, Drift, and the Temporal-Salience Structure of Problems

Dirk Thierens¹, David E. Goldberg², Ângela Guimarães Pereira³

Abstract— The convergence speed of building blocks depends on their marginal fitness contribution or on the salience structure of the problem. We use a sequential parameterization approach to build models of the differential convergence behavior, and derive time complexities for the boundary case which is obtained with an exponentially scaled problem (BinInt). We show that this domino convergence time complexity is linear in the number of building blocks ($O(l)$) for selection algorithms with constant selection intensity (such as tournament selection and (μ, λ) or truncation selection), and exponential ($O(2^l)$) for proportionate selection. These complexities should be compared with the convergence speed for uniformly salient problems which are respectively ($O(\sqrt{l})$) and ($O(l \ln l)$). In addition we relate this facetwise model to a genetic drift model, and identify where and when the stochastic fluctuations due to drift overwhelms the domino convergence, resulting in drift stall. The combined models interrelate the strong convergence of salient building blocks and the stochastic drift of less salient ones.

Keywords— Convergence time complexity, domino convergence, drift stall, sequential parameterization modeling.

I. INTRODUCTION

The novice evolutionary algorithmist often notices an interesting fact about the problems he or she tries to solve. Building blocks with higher marginal fitness contribution—*salient* building blocks—converge before those with lower marginal fitness. Moreover, there comes a point in time—and fitness scale—when strong convergence runs out of steam, and low salience building blocks *drift* to substantial convergence at random with a resulting high probability of error. Yet, while such experiences are commonplace for even the rank beginner, little modeling has taken place to understand the details of the timing and quality of convergence in such cases. Moreover, the lack of analytical understanding has slowed or hampered the design to overcome the nemesis of drift *stall*.

In this paper, we partially remedy this unsatisfactory state of affairs by first creating a simple model of *domino convergence* that accounts for the theoretical time to correctly converge a building block of specified salience. In particular, we analyze a problem with exponential scaling (the binary integer or BinInt problem), although the method and parameterization adopted may be applied to problems of arbitrary fitness distribution. We then use this ideal model, a model of drift convergence times, and dimensional reasoning to predict the point at which drift

stall should occur. Computational experiments verify these modeling efforts to within a calibration constant of the predicted values. Extensions to the work and ramifications of the modeling for GA design are considered.

II. BACKGROUND

The BinInt problem has an exponential scaled salience or fitness structure, and is defined as:

$$f(x) = \sum_{i=1}^l x_i 2^{l-i} \quad x_i \in \{1, 0\},$$

where l is the string length and x_i the alleles. The salience of one particular gene is higher than the combined marginal fitness contributions of all the following genes. The outcome of a competition between two arbitrary strings is thus uniquely determined at the most significant locus where the two strings have opposite alleles. For instance if we have two strings i1: 11110000 and i2: 11001100 and the fitness contributions decrease from left to right, then the competition is fully determined at the third position where i1 has allele 1 and i2 has allele 0. As a consequence, we should expect the genes to converge sequentially starting those with highest salience and finishing with those of lowest salience. The fitness signal of the less salient genes is so low as compared to the signal of the more salient genes, that they can only feel any selection pressure when the more salient ones are (almost) fully converged. On the contrary the more salient genes are not hampered much by the signals of the low salience genes, so the collateral noise for them is very low and they converge very fast.

The BinInt problem was introduced in (Rudnick, 1992) and he called the sequential convergence phenomenon *domino convergence* due to its resemblance with a falling row of domino stones. Rudnick's study looked at two things: first he showed the existence of a convergence window. This window is a set of contiguous genes that have started to converge but have not yet fully converged. The more salient genes are already fully converged while the genes at the other side of the convergence window did not start to converge yet. Figure 1 shows a plot of this convergence behavior. The second result of Rudnick's study was the analysis of the convergence stall due to mutation: when using mutation at a fairly high rate the lower salient genes do not fully converge since the selective pressure acting on these genes is counterbalanced by the disruptive effect of the mutation operator.

In this paper we analyze the convergence time complexity of the BinInt problem because it contrasts nicely with

¹Department of Computer Science, Utrecht University, The Netherlands

²Department of General Engineering, University of Illinois at Urbana-Champaign, USA

³College of Science and Technology, New University of Lisbon, Portugal

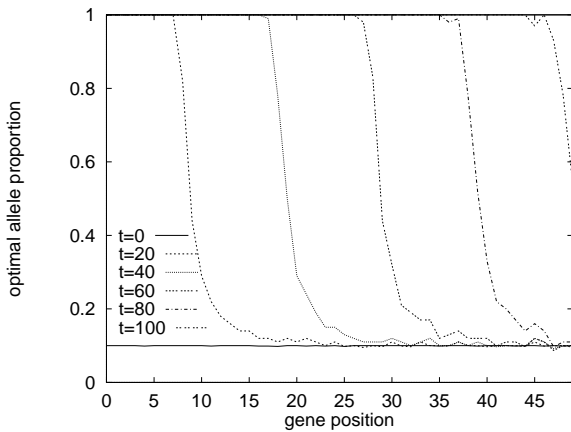


Fig. 1. Domino convergence window: the proportion of optimal allele values at each gene shown at different generations t . Experiment used tournament selection, uniform crossover, a string length 50, and initial optimal alleles proportion of $p_0 = 0.1$.

the convergence behavior of the OneMax or BitCounting problem. In the OneMax problem all genes are equally salient and the convergence is uniform for all genes. Analysis has shown that the convergence complexity on the OneMax problem for constant selection intensity algorithms is of order $(O(\sqrt{l}))$, and of order $(O(l \ln l))$ for proportionate selection (Mühlenbein and Schlierkamp-Voosen, 1993; Thierens and Goldberg, 1994; Miller and Goldberg, 1995; Bäck, 1995; Blicke and Thiele, 1995). The convergence complexity on the problems with power-law distributed salience structure are somewhere in between these two extremes, but their modeling is left for further work.

The convergence models of the OneMax problem were based on the computation of the selection differential $S(t)$ for different selection algorithms. The selection differential $S(t)$ is the difference between the mean fitness of the parent population at generation t and the population mean fitness at generation t . The selection intensity $I(t)$ is defined as the scaled selection differential where the scale factor is the standard deviation $\sigma(t)$ of the population fitness.

$$I(t) = \frac{S(t)}{\sigma(t)} = \frac{\overline{f^s(t)} - \overline{f(t)}}{\sigma(t)}.$$

In the OneMax problem all genes are equally salient and the expected proportion of optimal alleles at each generation is equal for the entire string, which allows a straightforward modeling of the convergence time complexity. However for problems with a non-uniform salience structure we need to incorporate the sequential convergence characteristics. In the next section we discuss such a modeling approach.

III. MODELS

A key factor in the convergence of problems with a non-uniform distribution of building block fitness contributions is the sequential population takeover by the building blocks in domino order from most salient to least salient. To model this we introduce a sequential parameter $\lambda \in [0, \ell]$

that defines the dividing line between the building blocks that have been properly converged and those that have not. Thus, when $\lambda = 0$ all the bits are assumed to be at their initial random state, when $\lambda = \ell$ all bits are assumed to be converged to the correct value, and when λ is at some intermediate value the λ most salient values are assumed to be converged correctly, and the remaining $\ell - \lambda$ values are assumed to be at initial values.

In general the λ parameter specifies a certain property that becomes fulfilled by the building blocks in a sequential way: the more salient a building block is, the faster it will fulfil the property. In this paper we let λ represent the fully converged genes, but one should notice that it is possible to choose other sequential properties: for instance in niched-based GAs one could choose to let λ represent the building blocks that have attained a proportion equal to n_c/c , where n_c is the number of niches, and n the population size.

Here we build a model for the BinInt problem by sequential parameterizing the genes that are fully converged at generation t . Using this parameterization we then calculate the increase in population mean fitness from one generation to the next as a function of our parameter λ .

The domino convergence specifies three different regions of the chromosome: the fully converged region, the convergence window, and the region where genes have not felt any selection pressure yet. In order to simplify the modeling we only consider the two major regions: the fully converged part and the part where no convergence has taken place yet. Figure 1 plots experimental domino convergence values showing that the convergence window is only a few genes wide so its impact on the calculation of the mean and variance of the population fitness is very small, and will certainly not offset our results concerning the convergence time complexities.

Suppose we have a string of length l where the λ most significant genes are fully converged and the remaining $l - \lambda$ genes have not been subjected to any selection pressure yet. Starting from a randomly initialized population - and ignoring genetic drift for the moment - the proportion of optimal alleles in the non-converged region is simply 0.5.

The mean fitness $\mu(\lambda)$ is thus given by:

$$\begin{aligned} \mu(\lambda) &= \sum_{i=1}^{\lambda} 2^{l-i} + \sum_{i=\lambda+1}^l \frac{1}{2} 2^{l-i} \\ &= \sum_{j=0}^{l-1} 2^j - \sum_{j=0}^{l-\lambda-1} 2^j + \frac{1}{2} \sum_{j=0}^{l-\lambda-1} 2^j \\ &= 2^{l-1} (2 - 2^{-\lambda} - 2^{-l}) \end{aligned}$$

where we have substituted $l - i$ by j and made use of the equation $\sum_{j=0}^{n-1} 2^j = 2^n - 1$.

The fitness variance when λ genes are converged can be calculated as:

$$\sigma^2(\lambda) = \overline{f^2(\lambda)} - \overline{f(\lambda)}^2$$

$$\begin{aligned}
&= \frac{1}{2^{l-\lambda}} \sum_{j=0}^{2^{l-\lambda}-1} j^2 - \left(\frac{1}{2^{l-\lambda}} \sum_{j=0}^{2^{l-\lambda}-1} j \right)^2 \\
&= \frac{1}{2^{l-\lambda}} \frac{(2^{l-\lambda}-1)(2^{l-\lambda})(2^{l-\lambda}-1)+1}{6} \\
&\quad - \left(\frac{1}{2^{l-\lambda}} \frac{(2^{l-\lambda}-1)(2^{l-\lambda})}{2} \right)^2 \\
&= \frac{1}{12} (2^{2(l-\lambda)} - 1) \\
&\approx \frac{1}{12} 2^{2(l-\lambda)}
\end{aligned}$$

where we have made use of the equations $\sum_{j=0}^n j = \frac{n(n+1)}{2}$ and $\sum_{j=0}^n j^2 = \frac{n(n+1)(2n+1)}{6}$. The second step is obtained by recognizing that the variance is only determined by the non-converged region which has an equal proportion of alleles 0 and 1. The fitness function is just the binary integer representation so we simply have to take the average of all (squares of the) integers from 0 to $2^{l-\lambda} - 1$.

The calculations of the mean and variance of the fitness as a function of the number of converged genes λ allows us to derive dynamical models of the convergence process under different selection schemes. First we will consider selection schemes with a constant selection intensity (e.g. tournament selection, (μ, λ) or truncation selection, and rank based selection). Thereafter we will look at proportionate selection which has a variable selection intensity.

A. Constant selection intensity I

Selection schemes with constant selection intensity I cause an increase of the population mean fitness from one generation to the next which is equal to the product of the selection intensity with the standard deviation of the population fitness:

$$\mu_{t+1}(\lambda) - \mu_t(\lambda) = \sigma_t(\lambda) I.$$

Plugging in the above computed expressions for the fitness mean and variance results into:

$$-2^{l-1} (2^{-\lambda_{t+1}} - 2^{-\lambda_t}) = \frac{2^{l-\lambda_t}}{2\sqrt{3}} I$$

or

$$2^{-\lambda_{t+1}} = \left(1 - \frac{I}{\sqrt{3}}\right) 2^{-\lambda_t}.$$

Initially ($t = 0$) none of the genes are converged, so $\lambda_0 = 0$ and therefore:

$$2^{-\lambda_t} = \left[1 - \frac{I}{\sqrt{3}}\right]^t.$$

Calculating t in function of λ_t finally gives us:

$$t = \frac{-\ln 2}{\ln \left[1 - \frac{I}{\sqrt{3}}\right]} \lambda_t.$$

Clearly the number of generations t until convergence is a linear function of the gene position in the string. For instance in case of tournament selection with a tournament size two the selection intensity is $I = 1/\sqrt{\pi}$ and the expected number of generations until the entire string would be converged ($\lambda_t = l$) is:

$$t = \frac{-\ln 2}{\ln \left[1 - \frac{1}{\sqrt{3\pi}}\right]} l = 1.76 l.$$

So if the length of the string is 50 we expect complete convergence after approximately 88 generations. Note that the constant factor should not be taken too literally: this kind of modeling gives us the dimensional relations, and constant factors can be calibrated by experimental results. The model does show however that the convergence time for constant selection intensity algorithms is linear in the problem length.

B. Variable selection intensity I

Contrary to the order-based selection schemes, proportionate selection has a variable selection intensity I and the mean fitness increase after one generation is given by:

$$\mu_{t+1}(\lambda) - \mu_t(\lambda) = \frac{\sigma_t^2(\lambda)}{\mu_t(\lambda)}.$$

Plugging in our equations for the mean and variance now results into:

$$-2^{l-1} (2^{-\lambda_{t+1}} - 2^{-\lambda_t}) = \frac{2^{2(l-\lambda)}}{12 \cdot 2^{l-1} (2 - 2^{-\lambda} - 2^{-l})}$$

or

$$2^{-\lambda_{t+1}} - 2^{-\lambda_t} = -\frac{2^{-2\lambda}}{3(2 - 2^{-\lambda})}$$

where we have dropped the term -2^{-l} in the denominator. For ease of notation we let $2^{-\lambda} = u$:

$$u_{t+1} - u_t = -\frac{1}{3} \frac{u_t^2}{2 - u_t}.$$

This difference equation can be approximated by the corresponding differential equation:

$$\frac{du_t}{dt} = -\frac{1}{3} \frac{u_t^2}{2 - u_t}$$

or

$$\frac{t}{3} = \frac{2}{u_t} + \ln u_t + C.$$

The boundary condition is given by $t = 0$, $\lambda_0 = 0$ and $u_0 = 0$ so the integration constant becomes $C = -2$, or:

$$t = 3 (2^{\lambda_{t+1}} - \lambda_t \ln 2 - 2).$$

Recognizing that the constant and linear term are very small compared to the exponential term we finally get:

$$t = 6 \cdot 2^{\lambda_t}.$$

The convergence time for proportionate selection is thus an exponential function of the string length as opposed to the linear law for the constant selection intensity algorithms !

In the previous section we have build a model for the domino convergence of different selection algorithms. The model is a facetwise model in the sense that it only looked at the impact of exponentially scaled gene fitness contributions on the convergence time complexity. By definition, facetwise models only look at a limited number of aspects or facets of a problem, and we therefore expect their range of application to be limited. In the domino convergence the less salient genes do not feel any selection pressure for a number of generations, and the above model simply assumes that their allele proportion will remain unchanged. However in a finite-sized population the proportion of alleles fluctuates due to stochastic sampling errors, so even in the absence of any selective pressure, the genes will eventually become fixated at one particular allele. This phenomenon is called genetic drift and has been studied extensively in the population genetics literature (see for instance Crow & Kimura, 1970), and also in the context of genetic algorithms (Goldberg & Segrest, 1987; Asoh & Mühlenbein, 1994). Those studies showed that the expected time for a gene to converge due to genetic drift - using random sampling with replacement - is proportional to the population size N :

$$t_{drift} = c N$$

where the constant c depends on the initial allele proportion. For instance a gene with two alleles with initial proportion equal to 0.5 has an expected convergence time t_{drift} equal to $1.4 N$. Note that this is the expected or mean convergence time and a single trial can show a significantly different value because the variance in the drift process is quite large.

In domino convergence the genes that are not yet reached by the convergence window do not feel any selective pressure, and their allele proportions are changing solely by genetic drift. If we want to converge to the optimal string with high probability then the domino convergence has to reach the least salient genes before they converge due to drift. If on the other hand the convergence time $t_{convergence}$ as computed in the domino model is larger than the drift time t_{drift} , then the stochastic fluctuations overtake the convergence due to selection, and a drift stall will occur.

We can predict where in the string the drift stall will start by equating the time models for the domino convergence and the drift model, or $t_{drift} \approx t_{convergence}$. For proportionate selection this leads to:

$$1.4N \approx 6 \cdot 2^{\lambda^*}$$

or

$$\lambda^* \approx -2.1 + 1.44 \ln N,$$

while for the constant selection intensity algorithms, we have:

$$1.4N \approx \frac{-\ln 2}{\ln \left[1 - \frac{I}{\sqrt{3}} \right]} \lambda^*$$

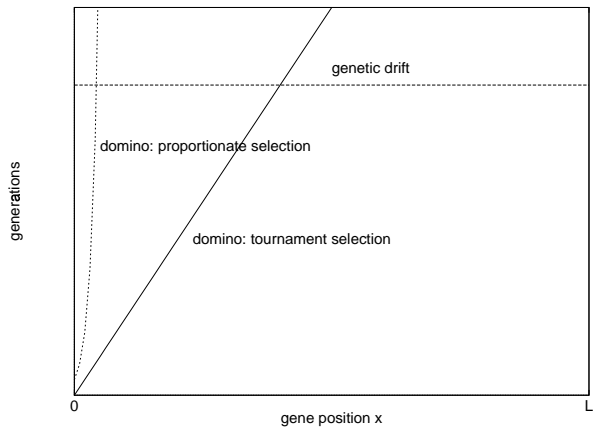


Fig. 2. Dimensional predictions made by the models: domino convergence is linear (resp. exponential) in the number of genes for tournament (resp. proportionate) selection, while the drift model predicts a constant generations upper boundary.

or

$$\lambda^* \approx -2 \ln \left[1 - \frac{I}{\sqrt{3}} \right] \cdot N.$$

The domino convergence complexity models predict a linear convergence speed for the constant selection intensity methods and an exponential rate for proportionate selection. At the same time the drift model predicts that there is an upper boundary at the convergence time for all genes that have not felt any selection pressure by then. This boundary is proportional to the population size and independent of the gene position. Figure 2 plots the model predictions, and in the next section we will experimentally verify them.

V. EXPERIMENTAL VERIFICATION

To validate the predictions made by the facetwise models we have counted the number of generations it takes before each of the genes fully converges. Results are averaged over 50 independent runs. First we look at proportionate selection which has a variable selection intensity I , then we use tournament selection as the representative of the constant selection intensity algorithms.

1. Proportionate selection.

We have implemented proportionate selection with roulette wheel selection: a parent is picked with probability f_i/\bar{F} , copied into the mating pool, and replaced into the original population. Figure 3 shows the gene convergence for a string of length $l = 20$, and a population size of $N = 250$. The domino convergence model predicts an exponential convergence curve, $t_{convergence} = 6 * 2^x$, while the drift model predicts an average convergence by drift after about $t_{drift} = 1.4 N = 350$ generations. Combining the two models we expect that domino convergence is valid for the first $x = \log_2 \frac{350}{6} = 5.9$ genes, and from there on the drift model specifies the convergence time.

It is clear from figure 3 that the predictions by the models and the experimental values coincide very well.

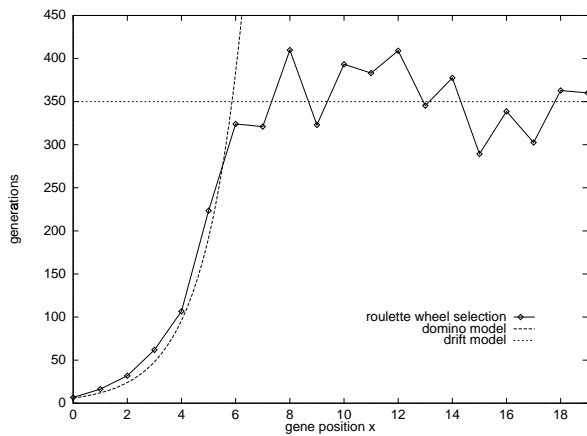


Fig. 3. Domino convergence model, drift model, and roulette wheel selection results.

2. Tournament selection.

In a second set of experiments we applied tournament selection (with replacement) on a string of length $l = 200$ with a population size $N = 50$. Figure 4 shows the results for tournament sizes $s = 2$ and $s = 4$. The model predictions are again found in the experiments: a linear convergence complexity of the most significant genes, and a constant convergence time due to genetic drift.

Two things about the drift boundary might seem surprising: first the average number of generations is substantially lower than the value predicted by the drift model ($t_{drift} = 1.4 N = 70$ generations), and second the drift time is different when using different selection intensities. A plausible cause for this might be that the least salient genes are simply hitchhiking with the more salient ones. To see the effect of this we run a set of experiments where no longer entire strings are selected to enter a tournament, but now we generate each time a new string by randomly copying an allele from the population at the corresponding gene position. So the created strings will get an allele-1 with probability equal to the proportion of alleles-1 in the population at that gene position. This way the alleles are selected independently from each other and less salient genes cannot simply hitchhike with more salient ones. Figure 5 plots the results for tournament size 2 together with experimental drift values. These drift values are obtained by generating strings from the population as above but no tournament selection is used afterwards. We see that the convergence time and drift stall without hitchhiking is higher and closer to the theoretical and experimental drift values. Even without hitchhiking the number of generations to convergence for the least salient genes is still somewhat below the pure drift values. A closer look at the experiments revealed that even for these long strings ($l=200$) a small fraction of tournaments still gets decided at the least salient genes, resulting in a lower average number of generations to convergence.

Note that for proportionate selection the hitchhiking effect does not show up because the convergence speed is an order of magnitude slower.

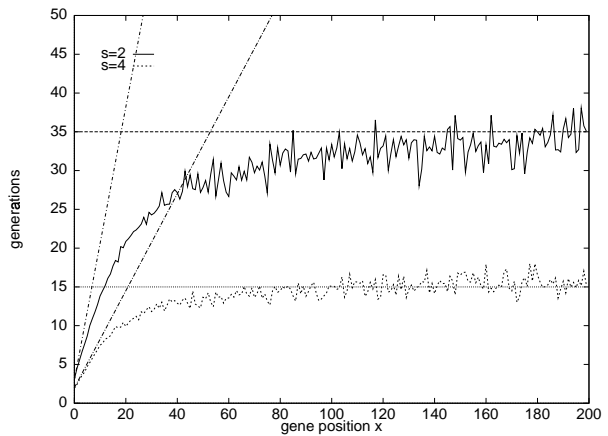


Fig. 4. Domino convergence model, drift model, and tournament selection results for tournament sizes $s = 2$ and $s = 4$.

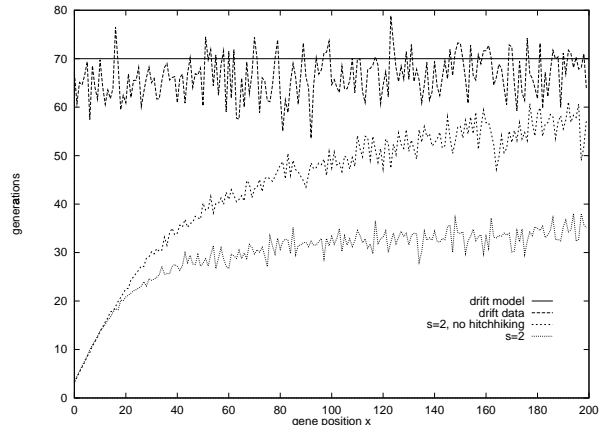


Fig. 5. Domino convergence model, drift model, and tournament selection ($s = 2$) with uniform and population based crossover.

VI. EXTENSIONS

This paper has given for the first time a simple model of the interrelation between convergence sequence, drift stall, time complexities, and the effect of the salience distribution of problems. We have limited the full analysis to the BinInt problem but it should be clear that the sequential parameterization modeling approach can also be applied to problems with other salience distributions. One immediate extension we are pursuing is to model the convergence time complexities for problems with a power-law fitness scale. Those models should not give any surprising results as their time complexities are bounded between the times given here ($O(l)$ for constant selection intensity algorithms, and $O(2^l)$ for proportionate selection), and the convergence times for the uniformly salient problems like the OneMax problem (resp. $O(\sqrt{l})$ and $O(l \ln l)$).

The modeling approach taken here is to build simple facetwise models and then interrelate them in a dimensionless way (Goldberg, Deb, & Thierens, 1993). An important extension is to include the effect of building block disruption: both the BinInt and the OneMax allows us to study the effects of the fitness or salience distribution on

the convergence speed, while bracketing the influence of building block survival and disruption. Incorporating this in the work done so far requires a straightforward facetwise model of building block propagation and mixing, and then interrelating this with the models given here.

A complementary approach might look at less facetwise modeling and try to build more integrated and accurate models. These more sophisticated models should be seen as an additional tool in understanding the complex behavior of our evolutionary algorithms. While one expects them to give more accurate predictions, they also might be less intuitively clear, so whatever model the evolutionary algorithmist uses will depend on his goals for using them in the first place.

VII. RAMIFICATIONS

In this paper we have identified and modeled the radical shift in convergence behavior when the reliable convergence of salient building blocks is overtaken by the stochastic fluctuations on the less salient building blocks. This understanding is very important to the GA designer as it clarifies what part of the marginal fitness contributions of his problem are processed fast and reliably in an implicit parallel way, and what part is under the nemesis of drift stall. In simple genetic algorithms the processing of salient building blocks is taken care of by the selection and recombination tandem, while the less salient building blocks have a high probability of being lost by the time selection and recombination can process them. In simple GAs we can only hope that mutation will bring them back, which is a slow serial process and the higher the building block order, the more time this will require.

The serial processing of mutation is much slower than the rapid implicit parallel processing power of selection and recombination. A number of extensions to the simple genetic algorithm can be devised to overcome the problem of drift stall. The key factor is to ensure that the diversity of the less salient building blocks is preserved long enough in order to buy time for the selection-recombination operators to reach this level of fitness scale. Different mechanisms for keeping the diversity are known: for instance dominance-diploidy, adaptive mutation, and recently a very interesting mechanism was introduced by Harik, called the probabilistic expression (Harik, 1997). Further work will need to study the convergence time complexities of these extended GAs.

VIII. CONCLUSIONS

Problems with a non-uniform distribution of the marginal fitness contribution of building blocks typically result in a temporal differentiated convergence behavior. The more salient building blocks converge rather fast, while the least salient ones only start to converge considerably when their 'big brothers' are almost done. An upper boundary case to this phenomenon occurs with an exponentially scaled salience structure, of which the BinInt problem is the prototypical example. We have applied a sequential parameterization approach to model this domino conver-

gence, and showed that the time complexity is linear in the number of building blocks ($O(l)$) for selection algorithms with constant selection intensity, and exponential ($O(2^l)$) for proportionate selection. These complexities should be compared with the convergence speed for uniformly salient problems which are respectively ($O(\sqrt{l})$) and ($O(l \ln l)$). In addition we have noticed that in a finite sized population the time for the less salient building blocks to wait to converge is limited because of genetic drift. Combining the domino convergence model and the genetic drift model by dimensional reasoning, we have identified where and when the stochastic fluctuations due to drift overwhelms the domino convergence, resulting in drift stall.

REFERENCES

- Asoh H., & Mühlenbein H. (1994). On the Mean Convergence Time of Evolutionary Algorithms without Selection and Mutation. *Lecture Notes in Computer Science, Vol. 866: PPSN-III*. eds. Y. Davidor, H.P. Schwefel, R. Männer. pp.98-107. Springer-Verlag.
- Bäck T. (1995b). Generalized Convergence Models for Tournament- and (μ, λ) -Selection. *Proceedings of the Sixth Int. Conf. Genetic Algorithms*, Morgan Kaufmann.
- Blickle and Thiele (1995). A Comparison of Selection Schemes used in Genetic Algorithms. *TIK-Report No. 11* Computer Engineering and Communication Networks Lab, ETH, Zurich, Switzerland.
- Crow J.F. & Kimura M. (1970). *An Introduction of Population Genetics Theory*. Harper And Row.
- Goldberg D.E., Deb K., & Thierens D. (1993). Toward a better understanding of mixing in genetic algorithms. *Journal of the Society for Instrumentation and Control Engineers, SICE Vol.32*, No.1 pp.10-16.
- Goldberg D.E., & Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1-8.
- Harik G. (1997). *Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms*. Doctoral disseration, University of Michigan.
- Miller B., & Goldberg D.E. (1996). Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise. *Evolutionary Computation 4(2)*, MIT Press.
- Mühlenbein H., & Schlierkamp-Voosen D. (1993). Predictive Models for the Breeder Genetic Algorithm. I. Continuous Parameter Optimization. *Evolutionary Computation 1(1):25-49*, MIT Press.
- Rudnick M. (1992). *Genetic Algorithms and Fitness Variance with an Application to the Automated Design of Artificial Neural Networks*. Unpublished doctoral disseration, Oregon Graduate Institute of Science and Technology.
- Thierens D., & Goldberg D.E. (1994). Convergence Models of Genetic Algorithm Selection Schemes. *Lecture Notes in Computer Science, Vol. 866: Parallel Problem Solving from Nature PPSN-III*. eds. Y. Davidor, H.P. Schwefel, R. Männer. pp.119-129. Springer-Verlag.