'Small Models': A Methodology for Designing and Investigating Adaptive Systems

JASON BROWNLEE

Technical Report 070326A

Complex Intelligent Systems Laboratory, Centre for Information Technology Research,
Faculty of Information and Communication Technologies, Swinburne University of Technology
Melbourne, Australia
jbrownlee@ict.swin.edu.au

***Abstract*-Complex and adaptive systems are difficult to design and investigate, thus the need for a coherent and proven methodology for such activities is paramount. Inspired by an interpretation of the achievements of the Wright brothers in achieving the first powered flight, this work summarises a methodology proposed by Goldberg as applied to the investigation and design of genetic algorithms.**

Keywords- Methodology, Genetic Algorithms, Small Models, Decomposition, Complex Adaptive System

I. INTRODUCTION

The design and investigation of complex and adaptive systems are difficult tasks and the application of an effective methodology is critical. Given his extensive and seminal contributions to the field of genetic algorithms, Goldberg [1,19] has used this experience to elucidate and apply an engineering-inspired methodology for investigating genetic algorithms (GA's) and designing what he calls 'competent genetic algorithms'.

Goldberg began to elucidate his ponderings on a methodology in the early 1990's ([3-6]) inspired by an account by Bradshaw and Lienert [23] on the innovation methodology used by the Wright brothers. Over the course of the subsequent 13-14 years Goldberg has released a series of papers on the topic which one may divide into two main thrusts: (1) *methodology* for the design and investigation of genetic algorithms (e.g. see [7,9-11]), and (2) genetic algorithms as a *computational mechanism for innovation* (e.g. [8,12-14]). This division is arbitrary because the presentation of these themes is tightly integrated, for example the results of the decomposition process of the methodology are used in arguments for genetic algorithms being considered an innovation mechanism. Goldberg's recent book [20] on the 'design of innovation' is a culmination and most recent revision of these themes.

This work provides a terse summary of Goldberg's methodology for investigating and designing what he calls 'conceptual machines', genetic algorithms being the primary example. It is the intent that this methodology may be adopted in the investigation and design of artificial immune systems (AIS), specifically the clonal selection adaptive plan and resultant models as proposed by Brownlee in [26,27]. Section II presents sound foundational concepts for the methodology, such as the Wright brothers example, and Goldberg's insightful comments on the experimentalist-theoretician debate. Section III describes the methodological elements themselves, specifically; decomposition, 'small models', and integration. Finally, section IV uses genetic algorithms to provide examples of the application of the methodology, which includes Goldberg's genetic algorithm design theory for the construction of competent genetic algorithms.

II. FOUNDATION

This section provides the context for the methodology presented in section III and discussed in the context of genetic algorithms in section IV. The context includes the so-called innovation methodology employed by the Wright brothers in their success of the first powered flight, and Goldberg's comments on the experimentalist (inventor-engineer) versus theoretician (mathematician-scientist) debate as it pertains to the development of a functional methodology for investigating and designing adaptive systems.

A. *The Wright Brothers*

A seminal influence on Goldberg articulating his methodology was the account by Bradshaw and Lienert [23] on the success of Wright brothers in achieving the first powered flight where so many others had failed. In their account, they rejected common reasons such as they were better craftsmen, or they had more time to focus their efforts given their bachelor status. Bradshaw and Lienert proposed that they were simply better inventers than their peers, extracting a methodology for invention used for their airplane.

This extracted methodology is as follows:

1. The Wright brothers decomposed the problem of powered flight into sub-problems
2. They solved the sub-problems using facet-wise models, empirical calibration, and dimensional analysis
3. They assembled the sub-solutions into an integrated solution, focused on results rather than elegance or mathematical niceties

This general approach resonated with Goldberg [6,7] particularly in his focus on the need for and defence of 'small models' in the investigation of genetic algorithms.

B. *Engineers and Mathematicians*

The idea of 'small models' is the core of the methodology and thus required defence. Goldberg describes the airplane and other products of engineering as 'material machines', and distinguishes them from the engineering of genetic algorithms and other adaptive systems as 'conceptual machines'. He argues the methodological distinction between the two is counterproductive and harmful from the perspective of conceptual machines, that the methodology of the material is equally applicable to that of the conceptual.

He argues:

"*I believe the unquestioning adoption of the methods and values of science and mathematics for the design of conceptual machines is slowing progress by demanding mathematical and experimental rigor without a corresponding payoff in the marginal advance of the technology*" [11] (page 3).

The obsession of mathematical rigor of computer science, although extremely valuable is not effective in the investigation of adaptive systems given their complexity. He sights ([10,11]) the airplane as an example where the engineering invention is used and trusted without a formal proof that the invention works (that an airplane can fly[1]).

This defence leads to what Goldberg refers to the 'economy of design' which is demonstrated with a trade-off that distinguishes *model description* (mathematician-scientists) that is concerned with model fidelity, and *model prescription* (engineer-inventor) that is concerned with a working product. In descriptive modelling 'the model is the thing' [of interest] whereas in prescriptive modelling, 'the object is the thing' [of interest]. In the latter, the model (and thus its utility) serves the object, in the former model accuracy may be of primary concern.

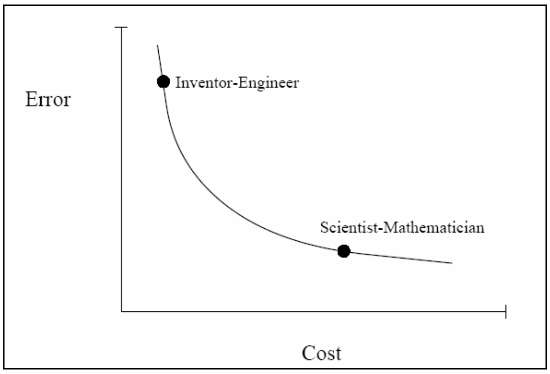

Figure 1 - Hypothetical Pareto front of an error-cost plane showing the relative positions of inventor-engineers and scientist-mathematicians, taken from [10]

This economy of modelling provides a perspective that distinguishes the needs of the prescriptive and descriptive fields of investigation. The mathematician-scientist is interested in increasing model accuracy at the expense of the speed (slow), where as the engineer may require a marginally predictive (inaccurate) model relatively quickly. This trade-off between high-cost high-accuracy models and low-cost low-fidelity models is what may be referred to as the 'modelling spectrum' that assists in selecting an appropriate level of modelling.

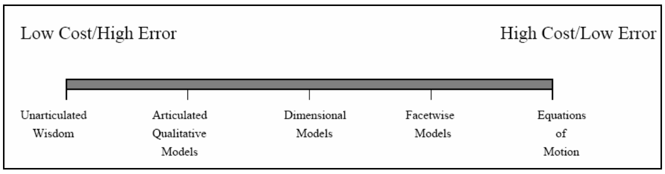

Figure 2 – 'Modelling Spectrum' taken from [10]

Goldberg proposes that the field of genetic algorithms expend too much effort at either ends of this spectrum. There is much work where there is an obsession with blind-prototyping many different tweaks in the hope of striking it lucky with the 'right' mechanism, operator, or parameter. Alternatively, there is also an obsession with detailed mathematical models such as full-blown differential equations and Markov chains. The middle ground of the spectrum, what Goldberg refers to as 'little models' is a valuable economic modelling consideration for the investigation of conceptual machines to 'do good science through good engineering'.

III. Methodology

The previous section highlighted the necessity of economic modelling. This section provides a summary of the methodology, of which 'little models' are the focus. The methodology has been referred to as post-modern systems engineering [14], and has been referred to as a methodology of innovation. A recapitulation of the methodology extracted from an analysis of the Wright brothers is presented (Figure 3).

Methodology Summary

1) Decompose the large problem approximately and intuitively, breaking into quasi-separate sub-problems

2) Investigate each sub problem separately (or as separate as possible) using empirical testing coupled with adequately predictive, low-cost models

3) Assemble the sub-solutions and test the overall invention, paying attention to unforseen interactions between the sub-problems

Figure 3 - Summary of the methodology from [10], inspired by an assessment of the Wright brothers in [23]

It is important to clarify that it is not so much that the methodology is novel or novel in the investigation of adaptive systems, rather it is the specific application as proposed by Goldberg that is the primary concern. This section discusses each step in turn, with a focus on the application of the economy of modelling and small models in the second step.

B. *Decomposition*

Problem decomposition and decomposition design is an axiom of reductionist-based research and is at the very heart of problem solving in computer science. Thus, it is not worth dwelling on the topic other than to comment that as to its meaning within the context of adaptive systems. One may consider the base or medium on which the system is performing its computation mechanisms, the so-called building blocks of

[1] Goldberg is quick to point out that sets of equations do exist for various aspects of flight, although no integrated mathematical proof for airplane flight exists

information processing. A structural decomposition may involve the architecture and data structures of the system. Finally, one may also consider a functional breakdown of mechanisms such as the operators applied at each discrete step of an algorithmic process or mechanisms. The reductions achieved provide the basis of investigation and modelling by 'small models'.

C. *Small Models*

Given the economy of modelling presented as a spectrum in Figure 2, one may extend the description of each of the five presented model types:

1. **Unarticulated Wisdom** (*low-cost, high-error*)
   - Intuition, what is used when there is nothing else
2. **Articulated Qualitative Models**
   - Descriptions of mechanisms
   - Graphical representations of processes and or relationships
   - Empirical observation or statistical data collection and analysis
3. **Dimensional Models**
4. **Facet-wise Models**
5. **Equations of Motion** (*high-cost, low-error*)
   - Differential Equations (deterministic)
   - Markov Chains (stochastic)

Figure 4 - Summary of the modelling spectrum

'Small models' refers to the middle of the spectrum, specifically to the application of dimensional models and facet-wise models. These are mid-range quantitative models, which make accurate prediction over a limited range of states at moderate cost. Once derived, this class of models generally requires a small amount of formal manipulation and large amounts of data for calibration and verification. See [11] for a concise treatment of this form of modelling.

Facet-wise models are an exercise in simple mathematics that may be used to investigate a decomposition element of a model in relative isolation. They are based on the idea of 'bracketing high-order phenomena' by simplifying or making assumptions about the state of the system. An example from fluid mechanics used by Goldberg is a series of equations that simplify the model by assuming that a fluid or gas has no viscosity, which matches no known substance. A common criticism of this modelling approach is "*system X doesn't work like that, the model is unrealistic*". The source of such concerns with adaptive systems is that their interactions are typically high-dimensional and nonlinear. Goldberg's response is that for a given poorly understood area of research, any useful model is better than no model. Dimensional analysis or the so-called dimensional reasoning and scaling laws are another common conceptual tool in engineering and the sciences. Such models may be used to investigate dimensionless parameters of the system, which may be considered the formalisation of the systemic behaviours.

D. *Integration*

Integration is a unification process of combining the findings of various models together to form a patch quilt coherent theory of the system. Integration obviously is not limited to holistic unification, one may address specific hypothesis regarding the system resulting in conclusions about existing systems, and design decisions pertaining to the next generation of systems.

E. *Application*

In addition to elucidating the methodology, Goldberg specifies a series of five useful heuristics for the application of the methodology as follows (taken from [11], page 8):

1. *Keep the goal of a working conceptual machine in mind.* Experiments commonly get side tracked by experimental design and statistical verification; theoreticians get side tracked with notions of mathematical rigor and model fidelity.
2. *Decompose the design ruthlessly.* One cannot address the analytical analysis of a system like a genetic algorithm in 'one big gulp'.
3. *Use facet-wise models with almost reckless abandon.* One should build easy models that can be solved by bracketing everything that gets in the way.
4. *Integrate facet-wise models using dimensional arguments.* One can combine many small models together in a patch quilt manner and defend the results of such models using dimensional analysis.
5. *Build high-order models when small models become inadequate.* Add complexity to models as complexity is needed (economy of modelling).

IV. Genetic Algorithms

Genetic algorithms are conceptual machines inspired by neo-Darwinian evolutionary theory. They are algorithms that are non-linear, large-memory, stochastic and operate on problems of infinite variety, high dimensionality, and complexity. This section discusses how the methodology was developed for the investigation and design of second-generation (competent) genetic algorithms. This section also provides examples of designing and investigating small models, and summarises Goldberg's theory that genetic algorithms are an example of a computational model of innovation.

A. *Examples*

The small models methodology was forged through its application to genetic algorithms. This section recounts some of the small model application examples.

The primary example provided by Goldberg is that of the analysis of 'takeover time' in the context of selection operators in [17] based on early work in [2]. This example is further simplified in an example of a facet-wise model presented in [11]. Takeover time refers how long it takes for a 'good' individual solution to dominate a population (to take-over the resources of the population). This is combined with a conception called 'innovation time' taken from a investigations of the crossover operator (innovation operator) from [15] to form a dimensionality analysis of the ratio of innovation time to take over time. This was also presented as a mixing dimensionality model [21]. Another interrelated example is that of investigations of population sizing with theoretical expectations and empirical observations [16,22].

In [12], Goldberg describes the use of control maps (the results of sensitivity analysis) in locating the 'sweet spot', that is investigating where genetic algorithms perform well. The example discussed comes from [15,21] of crossover pressure versus selective pressure, measuring the systems success on a trivial problem domain. The map provides a qualitative understanding of the key points made by the above facet-wise and dimensional models, highlighting meaningful performance boundaries such as drift (randomness), mixing (premature-convergence and steady-state evolution) and cross-competition (selective pressure being too high). See Figure 5 for an example of the crossover-selective pressure control map.

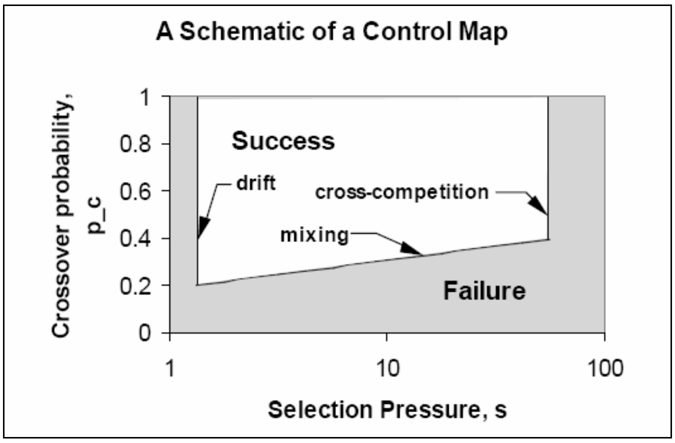

Figure 5 - Example control map taken from [12] (page 7)

B. *Design Theory*

Through experience in applying the methodology, specifically in the decomposition and small modelling of genetic algorithms, Goldberg proposes a theory for the design of genetic algorithms. The facets of the theory represent sub-problems of the genetic algorithm, which may be modelled in a quasi-independent manner. These facets evolved along with Goldberg's presentation of the theory, see Figure 6.

The Facets of Genetic Algorithm Design Theory

Building Blocks – The medium that genetic algorithms process, they work through mechanisms of quasi-decomposition and recomposition. Holland's schemata theorem provides the starting point of a mathematical analysis of these building blocks, suggesting that genetic algorithms work through a process of implicitly defining building blocks or subassemblies of good solutions, and recombine different subassemblies to form very high performance solutions.

Challenges – Identify and address the problems faced by building blocks. These challenges may be exploited by developing adversarial test problems. These are problems for which the algorithm finds it difficult to identify and acquire building blocks.

Supply – Ensure the initial supply of building blocks is adequate for the problem instance, particularly the number of decision variables in the problem. Likely related to the configured population size.

Growth – Allow good building blocks to gain an increased market share of the resources, permitting a so-called 'economy of ideas'.

Analysis – Measure the growth rate. Perform an analysis of properties such as takeover, drift, and convergence. The growth rate that cannot be too fast or too slow for finding good solutions.

Decisions – Understanding the decision making process within the system such that parameters may be meaningfully adjusted, for example increasing the population size to increase the likelihood of making better or the best possible decisions.

Mixing – The identification and the exchange of building blocks resulting in the recombination (innovation) of building blocks in novel ways.

Figure 6 - Summary of the elements of GA design theory, taken from [20] (page 52)

C. *Competent Genetic Algorithms*

The application of the methodology, the use of small models, and the genetic algorithm design theory (Figure 6) has lead to both (1) an understanding why simple genetic algorithms are servery limited on hard problems, and (2) to designing around those limitations lead to the second-generation (competent) genetic algorithms.

Competent Genetic Algorithm: A genetic algorithm that can scale efficiently with problem size and difficulty. A competent genetic algorithm can (1) address hard problems, it can do so (2) quickly, (3) accurately, and (4) reliably. Examples of competent genetic algorithms include: the fast messy genetic algorithm, the gene expression messy genetic algorithm, the linkage learning genetic algorithm, and the Bayesian optimization algorithm.

Figure 7 - Definition of a competent genetic algorithm

A competent genetic algorithm is able to address hard problems, quickly, accurately and reliably. This may be achieved through an effective balance between the selection and the crossover operators which is the critical design difference between so called first-generation and competent genetic algorithms, these second generation algorithms were designed to identify building blocks before deciding amongst and combining them. Goldberg gives four examples of competitive genetic algorithms [20] (chapter 12) as follows:

1. **Fast Messy Genetic Algorithm** (fmGA) [18]
2. **Gene Expression Messy Genetic Algorithm** (gemGA) [25]
3. **Linkage-Learning Genetic Algorithm** (LLGA) [24]
4. **Bayesian Optimization Algorithm** (BOA) [28,29]

D. *Innovation*

The methodology of innovation is taken a step further, where Goldberg claims that genetic algorithms themselves employ a process of innovation. The GA procedure may be considered a computational model for innovation, and the design of better GA's may be seen as the design of improved innovating conceptual machines – the so-called 'design of innovation'. The genetic algorithm is decomposed in the context of innovation, what is called a 'fundamental intuition of genetic algorithms' or the 'innovation intuition' (see [13] for a concise summary). In this decomposition, different combinations of operators may be observed as different forms of innovation.

Innovation Intuition: Goldberg's decomposition of the genetic algorithm in the context of innovation. The decomposition involves (1) continual improvement, and (2) cross-fertilisation. Continual improvement is provided by selection+mutation and provides a hill-climbing effect that is limited in scope. Cross-fertilisation is provided by selection+crossover that provides long jumps or new combinations of ideas. He refers to this decomposition as a 'fundamental intuition of genetic algorithms' or 'innovation intuition' for short.

Figure 8 - Summary of Goldberg's innovation intuition

Selection+Mutation=Continual Improvement. Here mutation provides minor variations on a theme providing a hill-climbing like effect of continual improvements. Such improvements are likely to be limited in scope, climbing to local optima, unless some ability of jumping to new areas is introduced.

Selection+Crossover=Innovation. The crossover operator provides the jumping mechanism. The recombination properties of the operator provide the cross-fertilisation of ideas, the mixing of components in novel arrangements that result in an innovation effect.

Goldberg suggests that this computational model of innovation may be transferable to other domains such as human innovation endeavours. He suggests that it may be used as a guiding philosophy for innovation. Taken from [13], Goldberg provides a number of lessons from the design of innovative conceptual machines (genetic algorithms) that may be applicable to the 'real-world':

- There are many models of the innovation process and the genetic algorithm demonstrates some of its different facets
- Respect the wisdom of the population. The population provides a testing ground, confirming the best notations are indeed the best. It is a place where one may try out new ideas (fail so that we may succeed)
- Innovation has a sweet spot (like a tennis racket). Innovation is played well when a number of variables are property aligned. Ensure alternatives are not eliminated too quickly with respect to the generation of ideas.
- Innovation depends on exchanging the right stuff. Genetic algorithms teach us that exchange if done property is the difference between finding and not finding a solution. For innovation one needs mechanisms for trying different decompositions of the problem was well as different classes of solutions.
- Creativity is more powerful than innovation. Innovation is used as a catch all for invention and improvement. Creativity may be introduced by remapping the primitives (changing the representation), and metaphorical transfer (transfer a solution from a different better-understood domain).

V. Discussion

The core of Goldberg's methodology is the prescription, advocation, and defence of using small models. Small models are economic to use in terms of cost-benefit and they have been demonstrated to be effective with genetic algorithms. In particular, the methodology is credited to assisting in the identification of the limitation of the first generation, and facilitating the design (transition) to the second generation of genetic algorithms. Ultimately, Goldberg's maxim that any useful model is better than no model may be sufficiently convincing to investigators of complex and adaptive systems who may be unsatisfied with qualitative results.

Acknowledgments

Tim Hendtlass for his patience and for providing useful feedback on drafts of this paper

References

[1] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, USA, Canada: Addison Wesley Publishing Company, Inc., 1989.

[2] David E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," *Proceedings of the third international conference on Genetic algorithms,* George Mason University, United States , pp. 70-79, 1989.

[3] David E. Goldberg, Genetic algorithms as a computational theory of conceptual design *Applications of Artificial Intelligence in Engineering*, vol. 6, pp. 3-16, 1991.

[4] David E. Goldberg, Making genetic algorithms fly: a lesson from the Wright brothers *Advanced Technology For Developers*, vol. 2, pp. 1-8, 1993.

[5] David E. Goldberg, The Wright brothers, genetic algorithms, and the design of complex systems *SYNAPSE '93 Japan*, vol. pp. 1-7, 1993.

[6] David E. Goldberg, A Wright brothers theory of genetic algorithm flight *Journal of the Institute of Systems, Control, and Information Engineers*, vol. 37, pp. 450-458, 1993.

[7] David E. Goldberg, First Flights at Genetic-Algorithm Kitty Hawk pp. 71994.

[8] David E. Goldberg, "The existential pleasures of Genetic Algorithms," *Genetic Algorithms in Engineering and Computer Science,* pp. 23–31, 1995.

[9] David E. Goldberg, "The design of innovating machines: Lessons from genetic algorithms," *Computational Methods in Applied Sciences '96,* Paris, France, pp. 100-104, 1996.

[10] David E. Goldberg, Toward a mechanics of conceptual machines: Developments in theoretical and applied mechanics *Proceedings of the Eighteenth Southeastern Conference on Theoretical and Applied Mechanics*, vol. pp. 1-9, 1996.

[11] David E. Goldberg, From Genetic and Evolutionary Optimization to the Design of Conceptual Machines *Evolutionary Optimization*, vol. 1, pp. 1-12, 1999.

[12] David E. Goldberg. The Race, the Hurdle, and the Sweet Spot: Lessons from Genetic Algorithms for the Automation of Design Innovation and Creativity. In: *Evolutionary Design by Computers*, ed. Peter J. Bentley. USA: Morgan Kaufmann, 1999.pp. 105

[13] David E. Goldberg, The Design of Innovation - Lessons from Genetic Algorithms, Lessons for the Real World *Technological Forecasting and Social Change*, vol. 64, pp. 7-12, 2000.

[14] David E. Goldberg, "The Design of Innovating Machines: A Fundamental Discipline for a Postmodern Systems Engineering," *Engineering Systems Symposium,* Cambridge, MA, USA, 2004.

[15] David E. Goldberg, K. Deb, and D. Theirens, Toward a better understanding of mixing in genetic algorithms *Journal of the Society of Instrument and Control Engineers*, vol. 32, pp. 10-16, 1993.

[16] David E. Goldberg, K. Deb, and J. H. Clark, Genetic Algorithms, Noise, and the Sizing of Populations *Complex Systems*, vol. 6, pp. 333-362, 1992.

[17] David E. Goldberg and Kalyanmoy Deb: "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," *Proceedings of the First Workshop on Foundations of Genetic Algorithms,* Bloomington, Indiana, USA, pp. 69-93, 1991.

[18] David E. Goldberg, Kalyanmoy Deb, Hillol Kargupta, and Georges Harik, "Rapid accurate optimization of difficult problems using fast messy genetic algorithms," *Proceedings of the 5th International Conference on Genetic Algorithms,* pp. 56-64, 1993.

[19] David Edward Goldberg, Computer-Aided GAS Pipeline Operation Using Genetic Algorithms and Rule Learning 1983. University of Michigan.

[20] David Edward Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, USA: Kluwer Academic Publishers, 2002.

[21] Dirk Thierens and David E. Goldberg, "Mixing in Genetic Algorithms," *Proceedings of the 5th International Conference on Genetic Algorithms,* pp. 38-47, 1993.

[22] G. Harik, E. Cantu-Paz, David E. Goldberg, and B. L. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," *IEEE International Conference on Evolutionary Computation,* Indianapolis, IN, USA, pp. 7-12, 1997.

[23] G. L. Bradshaw and M. Lienert, The invention of the airplane *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, vol. pp. 605-610, 1991.

[24] Georges R. Harik and David E. Goldberg, "Linkage Learning," *Foundations of Genetic Algorithms 4,* pp. 247-262, 1997.

[25] H. Kargupta , "The Gene Expression Messy Genetic Algorithm," *Proceedings of IEEE International Conference on Evolutionary Computation,* Nagoya, Japan, pp. 814-819, 1996.

[26] Jason Brownlee, "A Series of Adaptive Models Inspired by the Acquired Immune System," Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, Victoria, Australia, Technical Report ID: 070302A, Mar 2007.

[27] Jason Brownlee, "A Series of Discrete Repertoire Models Inspired by Lymphocyte Migration," Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, Victoria, Australia, Technical Report ID: 070320A , Mar 2007.

[28] M. Pelikan, David E. Goldberg, and E. Cantu-Paz, "Linkage Problem, Distribution Estimation, and Bayesian Networks," Illinois Genetic Algorithm Laboratory, University of Illinois at Urbana-Campaign, USA, IlliGAL Report Number 98013, 1998.

[29] Martin Pelikan, David E. Goldberg, and Erick Cantu-Paz, "BOA: The Bayesian Optimization Algorithm," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO -99),* Orlando, FL, pp. 525-532, 1999.