PoD Can Mutate: A Simple Dynamic Directed Mutation Approach for Genetic Algorithms

Adam Berry
School of Computing
University of Tasmania
Sandy Bay, Tasmania, Australia
Adam.Berry@utas.edu.au

Peter Vamplew
School of Computing
University of Tasmania
Sandy Bay, Tasmania, Australia
Peter.Vamplew@utas.edu.au

ABSTRACT

The incorporation of contemporary directed mutation techniques into traditional genetic algorithms has shown significant potential and garnered many promising results. While such findings are important, the utility of directed mutation in a generic sense is limited by either a reliance on domain specific knowledge or an increase in complexity with respect to both parameter size and the functions required for random number generation. Thus, this paper presents a new approach to mutation that utilises a co-evolving set of simple directions for each chromosome that dictate the result of gene mutations. In addition to biasing mutation towards promising areas of the search space, the Pointed Directed (PoD) mutation technique requires no increase in a priori information about the domain, nor does it come at a significant cost to system complexity. Moreover, results show that PoD mutation offers significant advantages over gaussian mutation across a range of complex problem areas, maximising performance by utilising adaptation to tune both directions and emergent system momentum.

1. INTRODUCTION

Given that mutation is a fundamental operator in genetic algorithms, it is not surprising that significant work has gone into subverting the natural randomness of the process with synthetic additions charged with guiding or improving the search. While approaches to this amendment are varied, amongst the most promising and frequently pursued areas is directed mutation. By biasing the operator to guide exploration towards important areas of the search space, directed mutation aims to improve both the quality of results garnered and the speed at which they are achieved.

Conventionally, the directed technique biases the mutation operator via the modelling of key components derived from pre-existing problem domain knowledge [1-7]. Often this approach requires an expert to determine weaknesses in prior runs and address these in such a way that solutions will match an expected result. While such directed mutation has proven successful, particularly with respect to engineering and timetabling problems, it is not without significant, and extremely limiting, deficiencies. Chief amongst these restrictions is that the reliance on domain knowledge precludes any generality in the directed approach – every problem will require restructuring of mutation and frequent experimentation until a suitable balance is met. Furthermore, by influencing the search space with domain knowledge, the process is made more susceptible to human error and becomes unfairly bound by human expectation – if the expert is continuously adjusting mutation to achieve an expected outcome, then the utility of the algorithm is limited only to achieving that outcome. Thus, the domain-guided form of directed mutation is overly restricted and unreasonably inflexible, when viewed from a generalist context.

Partially to address such inherent limitations in traditional directed mutation, more recent literature moves from problem-specific biasing to adaptive approaches [8-13]. The central goal in this dynamic technique is to shift the responsibility for establishing the direction of mutation away from the expert and to allow these to evolve from within the system. While results are limited, due to the fledgling nature of the procedure, they show significant promise and carry greater general applicability because of their detachment from domain knowledge. However, such generality comes at the cost of complexity – requiring an increase in the number of system parameters (from linear to quadratic, depending on approach) and a significantly more elaborate random-number generator.

To address the limitations of both domain-specific and adaptive directed mutation, this paper proposes a highly simplified co-evolutionary approach that avoids the complexity of the dynamic solution, while maintaining its generality. By tightly coupling the literal direction of mutation to each gene in a chromosome and allowing this direction to mutate and evolve across the run, the proposed Pointed Directed (PoD) mutation facilitates guiding of the search with minimal change to both the structure and operation of the standard generic Genetic Algorithm. Moreover, the use of non-transient, though evolving, directions should imbue the system with a form of momentum that will not only accelerate solutions to important areas of the search space, but also aid in escaping local optima.

2. DOMAIN-GUIDED DIRECTED MUTATION

Domain-specific directed mutations find their greatest utility in constraint-based problems, where conventional random mutation may disrupt feasible solutions. Indeed, the stochastic nature of mutation can result in previously good solutions being cast into infeasible regions of the search space, particularly in problems where constraints infer a high interconnectedness between genes within a single chromosome. Such a deficiency is a particular problem in fields such as timetabling, where the success of a solution is directly related to the relationship between alleles – a property which conventional mutation is incapable of incorporating into operation. Thus, domain-guided mutation finds an important niche in timetabling problems [1-4], using domain knowledge to guide the search away from infeasible regions of the search space by identifying disruptive time-periods [2], problematic classes [3] or ways to replace classes in domain-sensible ways [4]. The results of such niche-studies indicate that directed mutation, in constraint-oriented problems, provide significant performance increases over conventional genetic algorithms, particularly with respect to the efficiency of the search [2].

While constraint-based problems account for a significant amount of directed mutation research, the application of domain-specific operators is considerably more diverse and ranges from musical composition [5] to velocity inversion models for seismic refractions [6]. Indeed, where constraint satisfaction requires the avoidance of infeasible regions, the broader, though less focussed, area of directed mutation research resides in the improvement of results – specifically, the improvement of results so that they better match human expectation. In these cases, directed mutation is generally applied after an initial run indicates particular flaws in a solution that can be recognised and understood by a human-expert. This is particularly true of engineering design problems, where inadequacies of resulting designs are typically remedied by incorporating domain-specific guidance that will reduce the likelihood of practical deficiencies (as an example, [7] uses smoothing mutation operators to remove holes in a voxel representation of an I-beam structure). Although such approaches have proven successful, the incorporation of domain-knowledge specifically to remedy problems and thus influence the search, indelibly imprints evolution with human-expectation – the outcome of which is results that better match those expectations, but are ultimately limited by them.

Despite the applicability and relative success of domain-guided directed mutation in both constraint-based and more general problems, the fact remains that this form of heuristic requires significant restructuring of the standard genetic algorithm for each new problem to be solved. Such a lack of generality makes directed mutation expensive to incorporate, particularly given the typical need for a domain-expert to recognise the problem areas of greatest influence and importance. Moreover, the inclusion of domain knowledge will always bias the search process, be it intentional or otherwise, making it difficult to validate whether achieved results are maximal, or simply maximal within the constraints of the included influences. Finally, and perhaps most significantly, genetic algorithms are frequently used in areas where domain knowledge is low, which precludes the use of domain-specific genetic operators without thorough investigation and makes domain-guided directed mutation infeasible as a generic enhancement to evolutionary algorithms.

3. DYNAMIC DIRECTED MUTATION

Dynamic generic directed mutation is amongst the most niche and contemporary areas of Evolutionary Algorithms research. Where a-priori techniques are intrinsically bound to the problem at hand, Dynamic Directed Mutation (DDM) notionally builds on the concepts introduced in adaptive parameter control – namely, the adjustment of system properties according to current population states. Thus, DDM influences the nature of mutation based on the emergent behaviour of the population, rather than on preconceived conceptions about the specific problem domain.

Presently, the majority of DDM research takes place within the field of Evolutionary Strategies (ES). Given that traditional ES favours the use of mutation over crossover [14], it is not surprising that work has examined enhancements to gaussian mutation which are charged with directing the search. Principally, the techniques are self-adaptive and biased towards areas of improvement within the local search space, utilising a co-variance matrix with rotation angles in correlated mutation [8] or an adaptive skew-normal distribution in asymmetrical mutation [9-12]. While results have shown the utility of directed mutation in an ES framework, they are subject to, and limited by, significant disadvantages inherent in each approach. Correlated mutation is an extremely computationally expensive technique due to its reliance on matrices – requiring $N*D$ standard deviation parameters and $0.5ND(D-1)$ rotation parameters for a population of size N with D decision variables. Though asymmetrical mutation requires fewer parameters (reducing population-tied parameter growth from quadratic to linear [10]), its performance has not been satisfactorily investigated on multi-modal or real-world problems (where stagnation may cause problems). Moreover, both approaches are subject to diminishing selection pressure with respect to strategy parameters – a good individual carrying poor parameters will be more likely to propagate their directional cues than a poor solution with the perfect choice of parameters [14].

While use of DDM in traditional Genetic Algorithms has thus far been sparse, the few studies currently available warrant some discussion given the almost diametrically opposed nature of the technique in a GA context. Where Evolutionary Strategies use directed mutation to focus the search around areas of interest, existing GA approaches manipulate mutation to encourage population diversity – thus potentially driving solutions away from high-yield areas. Such diversity-guided directed mutation typically operates by ensuring that mutations move chromosomes away from the population centre of gravity (as in [13]), resulting in a greater exploration of the search space. While the utility of diversity in Genetic Algorithms has long been held, diversity-guided mutation may lead to significant disruption of useful solutions, a reduced emphasis on fine-grained local-exploration (particularly important as solutions near optimal states) and will require some form of inherently problem-dependent scaling. Moreover, it is inappropriate for those Genetic Algorithms which are mutation-centric, due to the divergent nature of an operator based on diversity, and carries the additional computational burden of performing population analyses to locate the centre of gravity for the system.

4. PoD Mutation

Pointed Directed (PoD) Mutation represents an inexpensive and simple new form of dynamic directed mutation designed for, though not limited to, use in generic Genetic Algorithms. In the PoD scheme, each gene is tightly coupled with a single bit that dictates the direction of mutation that the gene may follow (or more strictly – which half-gaussian distribution the mutation will utilise: positive or negative). Thus the coupled directions are representative of the evolutionary history of the chromosome, unlike the self-adaptive strategy parameters included in contemporary DDM, which represent only the likely history. In PoD mutation, if a gene is carrying a successful value and a positive inclination, then that gene can only have arrived at that value via a positive mutation, thus reducing the poor selection pressures associated with pre-existing DDM strategy parameters.

Since directions are represented as a single binary value[1] for each gene, mutation is straightforward (see Figure 1) and computationally inexpensive, facilitating rapid directional change via simple bit-flipping. Such malleability is in direct contrast to conventional asymmetrical approaches to DDM, where highly skewed populations are slow to reverse direction and escape evolutionary dead-ends. Thus, the consequence of the reinforced biasing seen in skew-normal approaches is that they are best suited to unimodal problem domains, while PoD should find utility in a broader range of areas.

While PoD mutation is simple and powerful, the highly coupled nature of the gene and direction means that independent direction perturbation is infeasible – it would introduce significant noise into the system and consequently reduce the corresponding selection pressures. Thus, directions are varied via coupled-mutation – where any change in direction leads to an immediate gene mutation in the newly defined inclination (see Figure 1d). The distribution between coupled- and gene-mutations represents the level of implicit momentum in a PoD system and dictates how aggressively the system will follow (and continue to follow) previously successfully directions. While it is possible to pre-define a correct level for such distributions as a system-wide parameter, it is fundamentally tied to the problem domain and would require significant tuning to achieve an acceptable balance. Subsequently, this paper utilises an elementary and low-cost adaptive scheme as defined in Equation 1, where s_t is the percentage of mutations that led to an improved solution over t generations; and c is a small predefined constant ($0 < c < 1$).

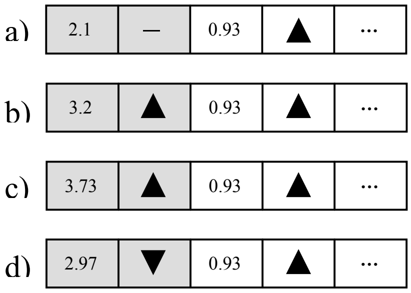


Figure 1 – Examples of PoD Mutation

a) The first gene-direction pair is in the initial state, with undefined direction

b) Random mutation of the initial-state pair in an upward direction

c) Mutation of the first allele, following the prescribed positive half-gaussian mutation

d) Mutation of the first direction leading to a negative mutation in the corresponding allele

$$\text{if } \left(s_{t_{couple}} > s_{t_{gene}} \right) \text{ then } \Delta\mu_{couple} = c\mu_{couple}$$
$$\text{if } \left(s_{t_{couple}} < s_{t_{gene}} \right) \text{ then } \Delta\mu_{couple} = -c\mu_{couple}$$
$$\mu_{gene} = \mu_{system} - \mu_{couple}$$

Equation 1 – Self-Adaptive Mutation Distributions

In addition to the cost savings afforded a self-adaptive mutation distribution, the technique maximises the efficiency of the directed mutation process. Driven by self-adaptation, PoD systems can move from high-momentum biases, which rapidly traverse linear descent portions of the problem space, to low-momentum configurations charged with finer grained search for end-of-run convergence. Under a static system parameter, such flexibility must be exchanged for an acceptable middle ground, which may be difficult to achieve in complex problem spaces.

4.1 PoD in a GA Context

Beyond the changes required by the inclusion of a self-adaptive scheme, the incorporation of PoD into a canonical real-valued GA simply involves the replacement of conventional (typically gaussian) mutation and subtle amendments to both the initialisation and crossover procedures. PoD-based chromosomes are initialised with random alleles and undefined directions, with the first mutation on the gene dictating the direction it will carry (see Figure 1a and 1b). Note that the inclusion of undefined directions is significant, as successful initial solutions should not infer any advantage to a particular direction (since no direction has been used to garner such an advantage).

With respect to crossover, since direction and allele values are tightly coupled, the gene-direction pair is treated as an indivisible unit. Given that the direction is meaningful only to the particular value with which it is associated, allowing independent crossover would be unreasonable – particularly since it would disrupt selection pressure on directions.

[1] Strictly, the coupled direction is a trinary value (due to the inclusion of an undefined state for initialisation purposes) – though throughout the majority of the run it can be treated as a simple binary.

Simple Unimodal: n=3, xᵢ∈[-51.2,51.2]		
F1	$\sum_{i=1}^{n} x_i^{\,2}$	DeJong's Sphere [15]
Complex Unimodal: n=2, xᵢ∈[-51.2,51.2]		
F2	$f(x)=\sum_{i=1}^{n} 100\left(x_{i+1}-x_i^{2}\right)^2+x_{i-1}^{\,2}$	Generalised Rosenbrock [15]
Flat Search Space Features: n=5, xᵢ∈[-51.2,51.2]		
F3	$f(x)=6n+\sum_{i=1}^{n}\lfloor x_i\rfloor$	DeJong's Step [15]
Noise: n=30, xᵢ∈[-12.8,12.8], R=uniform noise		
F4	$f(x)=\sum_{i=1}^{n} x_i^{\,4}(i+1)+R$	Modified DeJong Quartic [15]
Multimodal: n=10, xᵢ∈[-50,50]		
F5	$10n+\sum_{i=1}^{n} x_i^{\,2}-10\cos\left(2\pi x_i\right)$	Modified Rastrigin's Function [16]
Multimodal, with 25 foxholes: n=2, xᵢ∈[-655.36,655.36]		
F6	$f(x)=\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{n}\left(x_i-a_{ij}\right)^6}$ $a_{1j}=16\left(j \bmod 5-2\right)$ $a_{2j}=16\left(\lfloor j\div 5-2\rfloor\right)$	Shekel's Foxholes [15]

Figure 2 – Test Functions

5. Results

5.1 Test Functions

In order for PoD mutation to be accepted as a generic improvement over conventional mutation, it is necessary to verify, at least in a preliminary way, that it outperforms conventional gaussian mutation across a broad range of problem areas. That is to say, it is inadequate to perform well simply on unimodal problems (where directed mutation has previously established the greatest advantage), as these form only a subset of the search-space features found in real-world tasks. Thus, this paper investigates PoD behaviour across a wide range of well-recognised test functions that aim to summarise the general problem characteristics indicative of real-world optimisation (see Figure 2).

Population	$n=25$
Mutation Rate	$\mu\in\{0.005,0.01,0.05,0.1\}$
Adaptive PoD Mutation Parameters	$0.5\mu\le\mu_{gene}\le 0.85\mu$; $\mu_{gene_0}=0.65\mu$; $c=0.05\Rightarrow\Delta\mu_{gene}=\pm 0.05\mu_{gene}$; $t=10$
Adaptive Std. Dev. Parameters	$\left(x_{\max}-x_{\min}\right)\le\sigma\le\left(x_{\max}-x_{\min}\right)\times 10^{-5}$ $\sigma_0=0.1\left(x_{\max}-x_{\min}\right)$; $\Delta\sigma=\pm 0.2\sigma$; if $\sigma_{t..10t}=\left(x_{\max}-x_{\min}\right)$ then $\sigma_{11t}=\sigma_0$
Selection	Asexual Tournament (size = 2)

Figure 3 – System Parameters

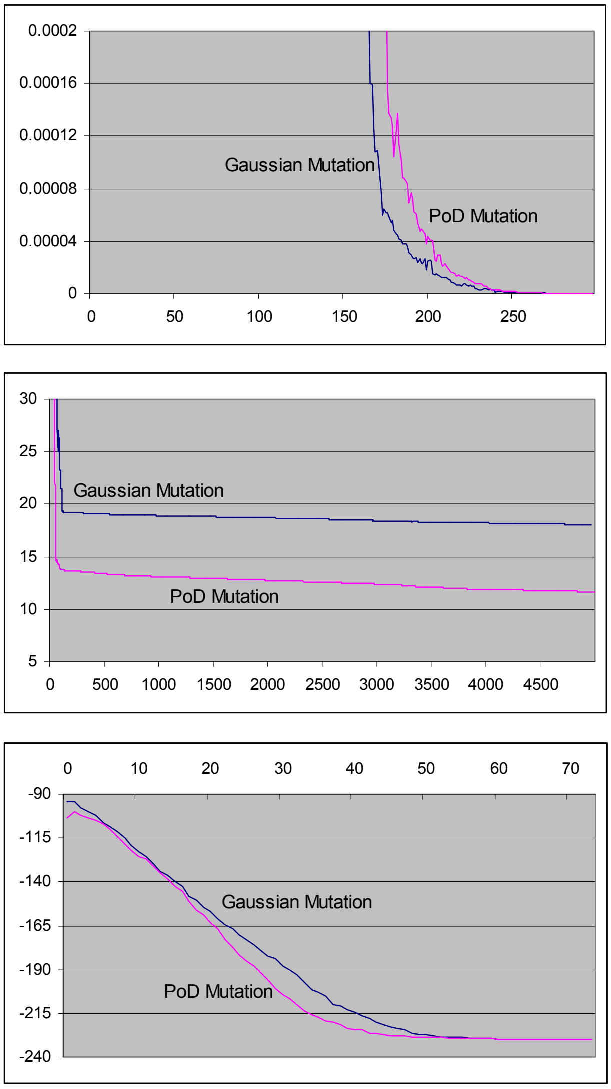
5.2 System Settings

The base system upon which the comparative tests are performed is an asexual[2] canonical genetic algorithm with system parameters defined in Figure 3. Note that since both gaussian and PoD mutation methods are based on normal distributions, an adaptive standard deviation is used (according to the 1/5 success rule [17], with reset to σ_0 on stalling) to ensure that incorrect step-sizes do not produce misleading outcomes.

All tests include twenty-five runs for a given function with a population size of twenty-five[2]. For each problem, systems use empirically observed optimal mutation rates, selected from the range outlined in Figure 3.

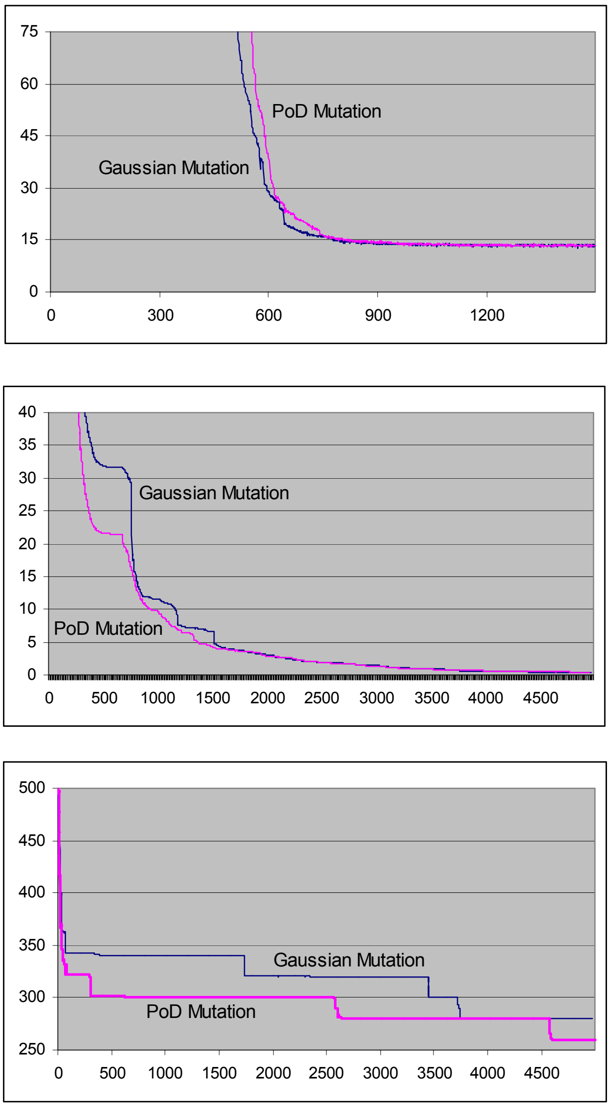
5.3 Function Performance

As is the convention for performance comparisons in Genetic Algorithms, results are illustrated graphically (see Figures 4 – 9), with the average lowest results



Figures 4, 5 and 6 – Performance of PoD Mutation against gaussian Mutation (F1:µ=0.1; F2:µ=0.05; F3:µ=0.1). X-axis: generation, Y-axis: lowest average result.

[2] Similar results for larger population sizes and sexual GAs will be presented in an upcoming Tech Report.



Figures 7, 8 and 9 – Performance of PoD Mutation against gaussian Mutation (F4:µ=0.005; F5:µ=0.01; F6:µ=0.05). X-axis: generation, Y-axis: average best result.

across the runs displayed. To further elucidate the utility of each approach, a table summarises the lowest averaged result obtained for each function (see Table 1).

In all cases, PoD is comparable and often better than the conventional gaussian mutation. For those cases where gaussian mutation offers greater efficiency, the difference is negligible and typically related to very fine-grained optimality convergence. It is likely that this degradation in performance is due to the pre-defined lower-boundary for standard deviations – if the boundary is set too high then mutation may consistently step over the optimal region, thus making it difficult to correctly deduce directions. Investigations into the influence of step-size on PoD mutation should be addressed in future work.

It is interesting to note that PoD performs worst (though again, the margin of difference is small) on F4. The per-gene noise in this function makes it difficult to correctly correlate the utility of directions and the success of the allele – the worth of the allele is not simply affected by the direction of mutation in this case, but also by the amount of noise it is subject to. In essence, the noise weakens (and even subverts) the selection pressure applied to directional choice.

	PoD Mutation	gaussian Mutation
F1	$3.57\text{x}10^{-8}$	$1.06\text{x}10^{-8}$
F2	11.665	18.051
F3	-230 (61 generations)	-230 (66 generations)
F4	13.281*	13.211*
F5	0.331	0.318
F6	260	280

Table 1 – Averaged lowest result for PoD and gaussian Mutation for all test functions. Note that since F4 is a noisy function, the displayed result is an average of the optimal points found from generation 1200 onwards.

While gaussian mutation is competitive on functions F1, F4 and F5, PoD mutation finds significant improvements on the remaining functions. Fundamentally, the efficiency increases are caused by the implicit, emergent momentum present in PoD mutation (a side-effect of high retention rates for previously successful directions). Such momentum accounts for speed improvements for traversal of flat portions of a search space (as in F3), as there is a reduced likelihood for oscillation in large basins. Furthermore, and somewhat counter-intuitively, momentum aids in the escape of local optima (as in F6) – particularly when coupled with higher standard deviations. Essentially, directions act as a form of attractor for solutions – it is only when a local optima provides a stronger influence that a PoD population member will become drawn in.

6. Conclusions and Future Work

This paper has presented a new approach to dynamic directed mutation that focuses on simplicity and adaptability within a Genetic Algorithms context. Across a diverse range of problem areas, PoD mutation shows impressive utility, always achieving competitive results when compared with gaussian mutation and frequently achieving significantly better outcomes.

While results are certainly promising, future work must address weaknesses regarding performance on noisy problem-domains where directional selection pressure diminishes. Furthermore, studies should be made into the influence of standard deviation levels on convergence and divergence in dynamic directed mutation – perhaps with a view to developing an adaptive approach that is less susceptible to local optima than the 1/5 success rule. Finally, the results presented in this paper are of a preliminary nature – given the positive results, it will be necessary to validate performance of PoD mutation on real-world problem domains.

7. Acknowledgements

The authors would like to acknowledge the following people and institutions, without whom this paper would not have been possible: Michael Berry, Trixie Berry, Pauline Mak, Ian Lewis, David Benda, the School of Computing at the University of Tasmania and the GAI Research Group.

REFERENCES

[1] Craenen, B.G.W., *An Experimental Comparison of Three Different Heuristic GAs for Solving Constraint Satisfaction Problems*, in Internal Report 98-21, Department of Computer Science. 1998, Leiden University.

[2] Burke, E.K. and Petrovic, S., *Recent Research Directions in Automated Timetabling*. European Journal of Operational Research, 2002.

[3] Ross, P., Corne, D., and Fang, H., *Improving Evolutionary Timetabling with Delta Evaluation and Directed Mutation*. PPSN III, 1994.

[4] Paechter, B., Ranking, R.C., Cumming, A., Fogarty, T.C., *Timetabling the Classes of an Entire University with an Evolutionary Algorithm*. PPSN V, LNCS, 1998. 1498: p. 865-874.

[5] Wiggins, G.A., *The Use of Constraint Systems for Musical Compositions.* ECAI98 Workshop on Constraints and Artistic Applications, 1998.

[6] Li, L., Louis, S.J., and Brune, J.N., *Application of Genetic Algorithms to 2D Velocity Inversion of Seismic Refraction Data.* Proceedings of the Third Golden West International Conference on Intelligent Systems, 1995.

[7] Baron, P.J., Fisher, R.B., Mill, F., Sherlock, A., Tuson, A.L., *A Voxel-Based Representation for the Evolutionary Shape Optimisation of a Simplified Beam: A Case-Study of a Problem-Centred Approach to Genetic Operator Design.* Soft Computing in Engineering Design and Manufacturing, 1997.

[8] Schwefel, H.-P., *Evolution and Optimum Seeking*. 1994, New York: John Wiley & Sons.

[9] Berlik, S., *A Polymorphical Mutation Operator for Evolution Strategies.* In Proceedings of the International Conference EUSFLAT 2003, 2003: p. 502-505.

[10] Berlik, S., *A Directed Mutation Framework for Evolutionary Algorithms.* Proceedings of the International Conference on Soft Computing, MENDEL, 2004.

[11] Berlik, S., *Directed Mutation by Means of the Skew-Normal Distribution.* International Conference on Computational Intelligence, FUZZY DAYS, 2004 [accepted].

[12] Hilderbrand, L., *Asymmetrische Evolutionsstrategien*. PhD Thesis, Department of Computer Science, University of Dortmund. 2002

[13] Ursem, R.K., *Diversity-Guided Evolutionary Algorithms.* Proceedings of Parallel Problem Solving from Nature VII (PPSN VII), 2002.

[14] Whitley, D., *An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls.* Information and Software Technology, 2001. 43: p. 817-831.

[15] DeJong, K., *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. PhD Thesis, Department of Computer Science, University of Michigain, 1975.

[16] Mühlenbeim, H., Schomisch, M., and Born, J., *The Parallel Genetic Algorithm as a Function Optimizer*. Parallel Computing, 1991. 17: p. 619-632.

[17] Rechenberg, I., *Evolutionsstrategie '94*. 1994, Stuttgart: Frommann-Holzboog.