Complementary selection and variation for an efficient multiobjective optimization of complex systems

Benoit Bagot[1], Hartmut Pohlheim[2]

[1] ZF Getriebe GmbH, Entwicklung Steuerungen PKW AT, Applikation Automatikgetriebe
PF 2567, 88015 Friedrichshafen, Germany
benoit.bagot@zf.com
[2] DaimlerChrysler AG, Research and Technology, Methods and Tools
Alt-Moabit 96a, 10559 Berlin, Germany
hartmut.pohlheim@daimlerchrysler.com

Abstract. Real-world applications generally distinguish themselves from theoretical developments in that they are much more complex and varied. As a consequence, better models require more details, new methods and, finally, more complexity. By confronting a benchmark evolutionary algorithm with an automotive gearbox with hundreds of parameters to optimize, we were able to observe new requirements which led us to an additional procedure.

Keywords: real-world application, multi-objective ranking, scalability, dominance, sexual selection, automobile industry, gearbox

1 Introduction

Roughly two years ago we were working on the optimization of a complex system (the parameterization of an automotive gearbox). A large number of parameters (variables) had to be set which, for their part, affected several evaluation criteria (objective values). Using evolutionary algorithms (EA) proved to be an effective optimization procedure which also allowed multi-objective optimization.

In the course of our work we examined the optimization runs more closely and were able to establish that in parts of the system good solutions had been abandoned by the optimization. As each individual optimization took many hours, even more preparation time and required the deployment of substantial resources on the engine test bench, we decided to look for the cause of the problem.

An in-depth analysis showed that the good partial solutions were being covered by other not-so-good partial solutions in the overall picture – their effect was being suppressed.

In this paper we shall present our reflections on how to avoid this phenomenon as well as parallels from the area of biology/genetics which served as the starting point for further work, see section 2.

Based on these reflections we looked for a use and implementation of these principles. The first domain which we identified as promising was the connection between the incomplete coupling of variables which were to be optimized and the system's objective values. In section 4 we describe an implementation of the principles outlined in sections 2 and 3. We have called this extension complementary selection and variation. It became apparent that this extension can easily be incorporated into the established procedure without changes having to be made to the existing structure.

The application of complementary selection and variation is shown in section 5 using examples constructed from well-known benchmark functions. This should make the principles presented as well as their positive effects easy to understand. We shall also provide the results of our real-world application of the gearbox optimization. Unfortunately, we cannot give a full account of this as not all details have been approved for publication.

Finally, in section 5, we show how the principle presented noticeably improves the scalability of evolutionary algorithms for these kinds of incompletely coupled systems, making it possible to solve more complex problems with the computing technology available today or rather to do so in less time or with less effort. In the example of gearbox optimization mentioned earlier, one can now, with the same amount of effort, simultaneously parameterize the gearbox for a greater number of working points than was possible at the beginning of our optimization work, thus achieving higher quality. This is quite apart from the fact that the use of evolutionary algorithms clearly facilitates a task which is monotonous and prone to error.

2 Biological Concepts and Complexity

If there is a growing number of variables and features as well as a lot of interactions between these, the complexity of the system under investigation rises considerably. One can try to simplify these complex systems by compartmentalizing them. In this way, the complex system can be seen as a number of simpler systems. These can, on the one hand, be analyzed more easily and then, on the other, be optimized more easily. It is quite possible for features to exist which have only been generated with the information from one gene. There are also features generated with the information from several genes which lie on one and the same chromosome. This is shown in a simplified diagram in figure 1.

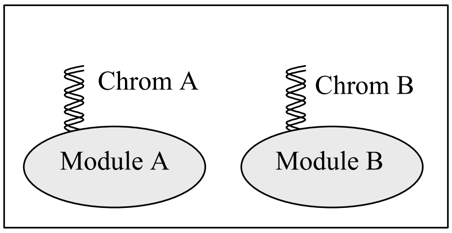


Figure 1. No coupling between chromosomes/modules during feature generation

This kind of compartmentalization can, however, only be carried out in the rarest of cases. Most features are formed with the information from several genes which are distributed over various chromosomes. The same genes can simultaneously contribute their information in order to generate several features. Figure 2 shows this coupling between the genes belonging to different chromosomes in order to generate features.

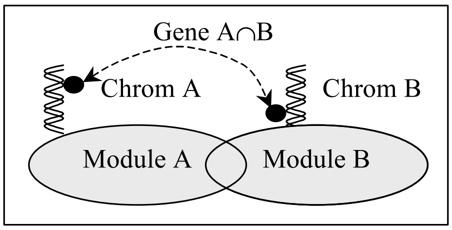


Figure 2. Partial coupling between chromosomes/modules during feature generation

This constitutes a borderline situation: we cannot separate parts of the system into parts running strictly parallel to each other, nor are we able (or do we want) to mix all the parts with each other. In figure 3 we have tried to provide an idea of what coupling between different genes/chromosomes and their dependent features could look like.

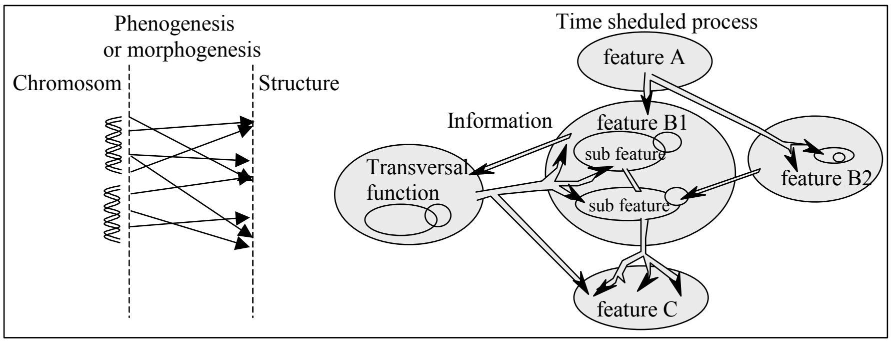


Figure 3. Complex interaction between chromosomes/modules and features/sub features

If, for example, we try to consider the human genome with its 46 units, it soon becomes clear that this division is only an extremely rough approximation of its complexity. Consequently, other strategies and data structures must be employed in order to reflect the system's complexity within the evolutionary process. The features that appear in figure 3 can be considered directly as objective (or fitness) values. We can also decompose the relations between genes and fitness values into two steps, firstly morphogenesis which leads to the constitution of structures characterizing individuals, and secondly evaluation (testing) which relates structures to fitness values, see figure 4. This whole process is called phenogenesis. A similar description of mapping between "evolved representations" (genes) and "representation that is evaluated by the fitness function", as result of the morphogenetic process, is given by Angeline [2].

The complexity of the system can be divided into

- the complexity of the data structure and
- the complexity of the procedure.

However, this division cannot be taken too strictly. On the one hand, a new procedure can necessitate the storing of additional information. On the other, complex data structuring can make new administrative functions necessary.

In complementary selection and variation, which we would like to propose as a means of faster optimization, particular use is made of additional information. An evolutionary algorithm works with this information in a way which is not only purely stochastic but also knowledge-based. The aim is to use the available knowledge concerning phenogenesis in the evolution procedures. We are still working on extending this principle so that it can be applied to new procedures.

3 Complementary Selection and Variation

The basic idea behind complementary crossover is very simple. First, form parent couples which complement each other well in their respective strengths. Then take the best gene from each parent in order to create a new individual.

We are not making a case for artificial gene selection (gene manipulation) or eugenism. Quite the opposite, we have attempted to find out how this selection could take place naturally. Each person has different character traits with different advantages which can complement each other well. This more constructive point of view can be seen as the counterpart to the competition and struggle for life or "for the possession of the other sex" pointed out by Darwin and Spencer, or as the transposition of social (or ecological) complementarity to the level of the genome. Meanwhile, we take Darwin's definition for use on the level of organs or genes: "this preservation of favourable individual differences and variations, and the destruction of those which are injurious, I have called Natural Selection or the Survival of the Fittest" [3].

We have tried to challenge some earlier assertions. For example, Monod [11] claims that "the selection operates on the macroscopic scale, that of the organism". We, in contrast, have selected on gene level. F. Jacob [6] assumes that "the program does not learn its lesson from experience". However, we found that this would be an unnecessary restriction for efficient optimization. That is why we wanted to feed our algorithm with knowledge derived from experience. We reconsidered Mayr's [10] words: "There is no link between a molecular event and its potential significance. It's the same for the mix of chromosomes (crossing over) or their segregation, the selection of gametes or of sexual partners".

At the same time, these authors and scientific papers (such as Losos' [9]) confirmed, to some extent, our idea that genes could be involved in a lot of intelligent processes of evolution. As Losos puts it: "Phenotic plasticity: an organism ("phenotype") is not only formed by its genes (its "genotype") but also by its environment. […] For a few years now, evolutionary scientists have increasingly been discussing the question of whether such a plasticity could possibly also be of importance to evolution". If this biological analogy were transferred to evolutionary algorithms, for example, this might lead to the inclusion of gradient-based procedures in hybrid evolutionary algorithms.

Even though neither the examples mentioned nor the textbooks (such as Lodish's [8] or Reinhard's [15]), or the specific literature in the field of EAs referred directly to complementary selection and variation, they have inspired us strongly. Schaffer (quoted by Zitzler [18]), for example, carries out selection for each objective separately. Complementary recombination with the use of dominance and diploidy has been studied frequently, see for example [7]. Learning from experience has been used in different forms, such as in Rasheed's search control [14]. Note that the Lamarckian learning that we use is applied to the additional information "dominance", which does not code directly (intergenic regions, introns or pseudogenes), see Angeline [2] and Singh [16]. Rules for the viability of mating are mentioned in Fonseca [4]. We gathered and adapted these ideas in the new procedure. Even if the procedure cannot be found in nature, this does not keep us from working with it and the evolutionary algorithms may well emancipate themselves from this paradigm. In any case, we agree with Zwirn [19] that biology will continue to be an important source of inspiration for work with complex systems.

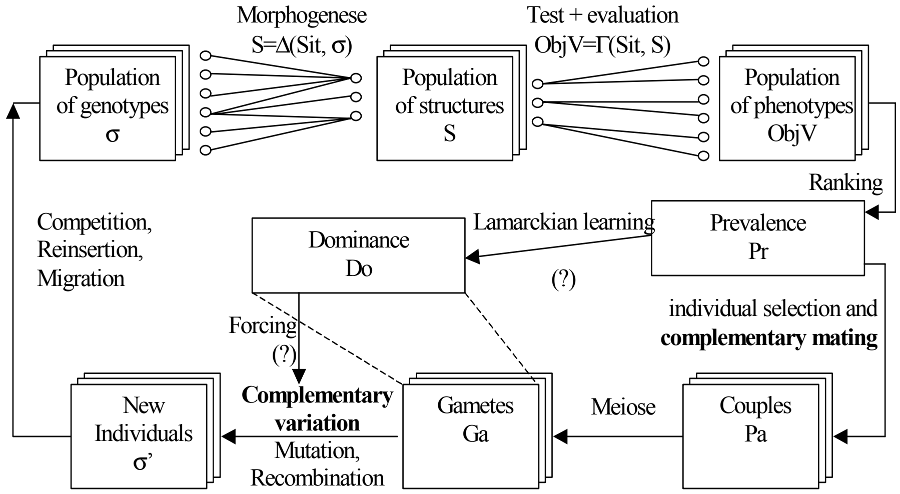
The essential reason for biologists' rejection of Lamarckian evolution (heredity of learned characters) comes from molecular biology and is based on the absence of observation: there is no known molecular mechanism that can explain a feed back to the genes, the decoding of genetic information is a one-way process AND->ARN->Protein. Furthermore, real observations of the evolution process can be explained by Darwinian selection alone and attempts to prove the heredity of learned characters failed. A few recent discoveries seem to question this central dogma, e.g. repair mechanisms, regulation mechanisms, genetic conversion, transformation ARN->ADN in a virus. To the second argument, we can counter that a Lamarckian evolutionary process should be robust in the face of the hazards of life and work very slowly because it takes time to gather solid knowledge. Therefore, its effects will be difficult to recognize, especially in complex organisms. It would be interesting to accelerate this learning process in an artificial system, depending on environmental noise.

During our work, we considered several ways to ascertain the information "dominance". The first possibility is through slow mutation and selection in parts of the genome that carry this information. Or we could use a retarded Lamarckian mechanism by which, for example, the corresponding genes of two alleles are alternatively compared during individuals' lives. We chose the third possibility, which would be quite incredible in the nature but is easy to realize in a program: direct Lamarckian learning. The genes of two parent individuals are directly compared assuming an information exchange on this level. Dominance would only be an advantage in a stable environment or if cyclical changes occur, in which "learning" makes sense. This is a common situation in industrial applications.

Perhaps Lamarckian heredity is not necessary for the existence of complementary selection and variation in nature. However, we thought it necessary to point out its role because of its didactic effect on our work and its effective use in our algorithms.

3.1 Procedure on the basis of a simple example

In figure 4, we represent complementary selection and variation within the whole schema of iterative evolution. We then show the single steps which led us to the global procedure.


Morphogenese
S=Δ(Sit, σ)
Test + evaluation
ObjV=Γ(Sit, S)
Population of genotypes σ
Population of structures S
Population of phenotypes ObjV
Ranking
Prevalence Pr
Lamarckian learning
(?)
Dominance Do
Competition, Reinsertion, Migration
individual selection and **complementary mating**
Forcing
(?)
Complementary variation
Mutation, Recombination
New Individuals σ'
Gametes Ga
Meiose
Couples Pa

σ = structured data:
- coding genes
- loci (seen as information)
- enhancer, homeobox
- dominance modifier

S = structured body
ObjV = structured fitness values

Sit = time, situations (changing environment parameters), society, parents, hard and smooth constraints.

Figure 4. Complementary selection and variation as part of the whole iterative evolutionary process

To illustrate the procedure, let us look at an example of five individuals which possess two genes each, whose expression is identifiable in two features.

Gene A finds its expression in the f1 feature; Gene B finds its expression in the f2 feature.

	Gene A	Gene B	f1	f2
Ind 1	A1	B1	1	9
Ind 2	A2	B2	3	3
Ind 3	A3	B3	2	7
Ind 4	A4	B4	8	1
Ind 5	A5	B5	6	2

A PARETO ranking which has $f1 < 7$ and $f2 < 8$ as its goals would exclude the individuals Ind 4 and Ind 1 from further optimization loops and thus give away valuable potential of the genes A1 and B4. Here we can see that in the case of genes (and features) being assigned absolutely separately, the best solution would combine Gene A1-Gene B4.

How do we arrive at this solution? Firstly, all existing features are sorted by using a selective, or rather complementary, partner search. In the example we sort the solutions simultaneously according to f1 and f2:

Sorting according to:	f1	/	f2
	f1-Ind1	-	f2-Ind4
	f1-Ind3	-	f2-Ind5
	f1-Ind2	-	f2-Ind2
	f1-Ind5	-	f2-Ind3
	f1-Ind4	-	f2-Ind1

In this way, pairs are created which contain a complementary potential in their genes. Actually, neither individuals nor pairs are sorted here, but rather features, since Ind 1-Ind 4 can be seen as the best and worst result at the same time.

If we take the pair Ind 1-Ind 4, we have four possible descendants. Gene A1-Gene B1, Gene A1-Gene B4, Gene A4-Gene B1 and Gene A4-Gene B4.

How can the best solution Gene A1-Gene B4 prevail? In the course of the next generations after a new selection? This seems to be very inefficient. The use of dominance, which has been implemented in many EAs with di- or polyploidy, can be reemployed here.

The traits which have received a better rank with the corresponding feature will be assigned greater prevalence (see Allchin [1]) which we transform into dominance using previous knowledge concerning the correlations between gene-traits. Prevalence is the fitness of traits considered separately and ranked with their respective equivalent in the population. It is related to the whole population and not limited to the comparison between two alleles (or more) in diploid (or polyploid) chromosomes. We have called any process (and corresponding information) that leads to a stronger or favored expression (all-or-none or intermediate) or more frequent apparition and heredity of a gene or phene 'forcing'. While dominance tends to be attached to alleles and prevalence tends to refer to traits in a population, both contribute to forcing solutions which we suppose to be better; note that it could be interesting to introduce a third term matching the structure evaluated. Here, Gene A1 has the highest rank among all f1-rankings and Gene B4 ranks highest among all f2-rankings. Gene A1 and Gene B4 become dominant.

During crossover, or genetic conversion, or gene expression, only those genes are accepted which have the greatest dominance. "The set without any defects assumes leadership and hence is able to compensate the defects of the other set" [17]. Recessive genes should disappear gradually. Dominance will stay with the gene during the next generations and is increased or reduced with every new generation. Thus, a knowledge base is transferred from generation to generation and is possibly updated on an ongoing basis.

With continuous values it is possible to realize the recombination in a "soft" way, as figure 5 shows.

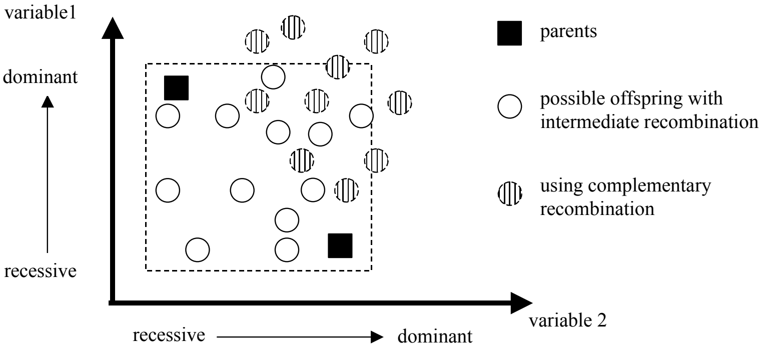
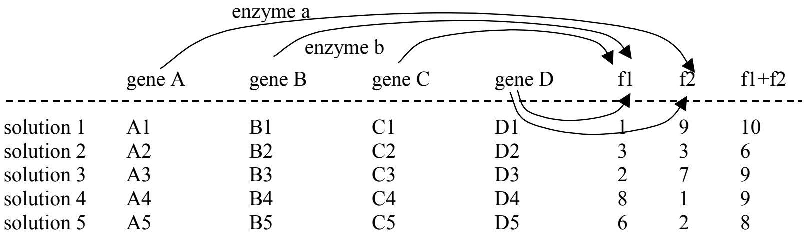

variable1
dominant
recessive
variable 2
recessive
dominant
parents
possible offspring with intermediate recombination
using complementary recombination

Figure 5. Effect of complementary intermediate recombination (principle) compared to intermediate recombination ("soft" recombination of dominant / recessive genes)

3.2 Extension of the procedure - knowledge-based or knowledge-building

In our procedure, the connections between the gene and the feature are formed by polygenia (a feature is influenced by several genes) and pleiotropy (one gene is involved in the formation of several features). Figure 6 tries to give an impression of these connections.



	gene A	gene B	gene C	gene D	f1	f2	f1+f2
solution 1	A1	B1	C1	D1	1	9	10
solution 2	A2	B2	C2	D2	3	3	6
solution 3	A3	B3	C3	D3	2	7	9
solution 4	A4	B4	C4	D4	8	1	9
solution 5	A5	B5	C5	D5	6	2	8

Figure 6. Polygenia and pleiotropy – the connection between gene and feature

The allocation of gene to feature is complex. This allocation can be predefined if it is already known. In this case, a knowledge basis will be defined before optimization is started (our implementation).

If the allocation between gene and feature is not known in advance, it can be decided for each generation, with the help of a correlation analysis for instance. In this second case, which we have not looked at more closely in our program realization yet, this knowledge builds up in the course of the optimization. Dominance is then a product between the correlation factor and the ranking value. When the rank is high (good), the ranking value is positive; when the rank is low, the ranking value is negative. This produces the following results:

- high correlation * positive ranking value -> dominant,
- high correlation * negative ranking value -> recessive,
- low correlation -> neutral.

The relation between gene and feature can be described with the help of a soft definition. The connection between country and language or language and culture is an example of this. In this case, we speak of "the sphere of influence". Genetic interference (silencing) and differentiated gene expression (few genes find strong expression; most of the genes find weak expression or none at all) show that the allocation of gene to feature is not at all trivial. It is quite certain that new functional patterns may be found here.

4 Programming

In technical practice a large number of dynamic systems exist which are evaluated by simulating the system for an extended time period and analyzing the system reaction, see figure 7. The system can be controlled via the input variables (variables for optimization). The problem-specific evaluation of the output signals usually results in several objective values rather than just one. Consequently, a multi-objective optimization of the system must be carried out.

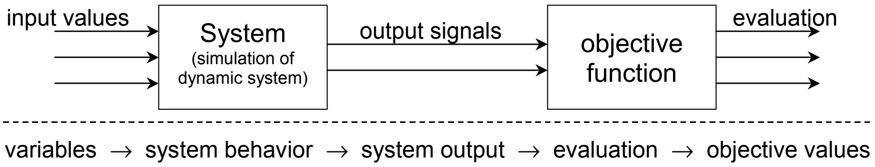


Figure 7. Structure of control and evaluation of a dynamic system

Real-world systems cannot be evaluated comprehensively if only one simulation is carried out but rather a greater number is necessary. In the case of a gearbox, several working points have to be investigated. Naturally, the number depends on the complexity of the system as well as the qualitative requirements on the result.

Unfortunately, this increases the number of variables which have to be optimized, the number of objective values and also the total simulation time. In short, the complexity of the optimization problem which is to be solved becomes noticeably greater.

However, the addition of further working points results in a situation in which not all the variables affect all the objective values. It is often the case that few objective values are dependent on a larger number of variables. Other objective values, in contrast, depend only on a small number of variables (for instance, parameters in a set of characteristic curves controlling a certain field of activity). It is often known in advance which of these variables contribute to which objective values.

In our first implementation of the principles discussed in section 3 we took advantage of this knowledge available to the engineer/optimizer concerning the inner couplings between variables/parameters and objective values present in the system to be optimized.

Here, we are working on the precept that the user knows which variables `Var` (termed genes in section 2) affect which objective values `ObjV` (termed features in section 2). In this way, it is possible to state exactly that `Var1` affects `ObjV1` and that `ObjV2`, `Var2` affects `ObjV2` and `ObjV4`, etc.

In this procedure a variable can affect one or several objective values. In the same way, an objective value can be affected by several variables. As regards all real-world applications, we can assume that the user knows this allocation.

By the way, this procedure also covers for the eventuality that all the variables affect all the objective values (the most 'general special case', the only one which has been taken into account in all MO applications so far). This really is a special case, however, which very seldom occurs in practice when optimizing complex technical systems.

In order to define the assignment between variables and objective values we have employed a cell array. (As we carried out our work using Matlab we have used the Matlab notation here. The principles can, however, be transferred to any other kind of implementation.) The first column contains the indices of variables `Var`, each of which affect the same group of objective values `ObjV`. The second column contains these corresponding objective values.

```
% Assignment Var to ObjV
Var2ObjV = { [1],    [1 2];
             [2],    [2 4];
             [3 4], [3 5 6]
           };
```

This information allows us to break open the normal process of ranking, selection, recombination and mutation which, up till now, had been carried out for all variables and objective values simultaneously. With the new procedure, a separate loop is run through for each row in `Var2ObjV`, see figure 8. This is always done for exactly each of those variables and objective values that belong together. In the beginning, the individuals are divided into separate variable groups. New sub-individuals are generated for these groups, which are joined together again at the end. As a variable can only ever occur in one variable group, the variables do not overlap. Nevertheless, it is possible for an objective value to appear in one or all variable groups.

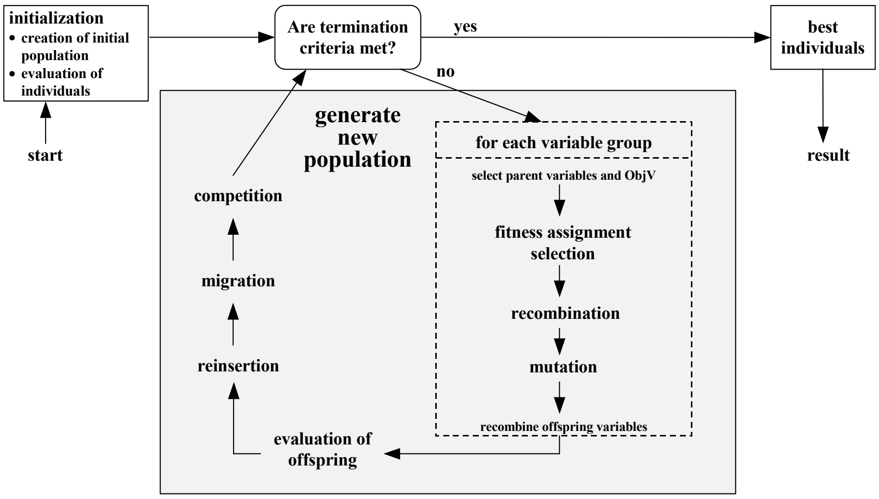

initialization
• creation of initial population
• evaluation of individuals
start
Are termination criteria met?
yes
no
best individuals
result
generate new population
for each variable group
select parent variables and ObjV
fitness assignment selection
recombination
mutation
recombine offspring variables
evaluation of offspring
reinsertion
migration
competition

Figure 8. Structure of the extended evolutionary algorithm employing complementary selection and variation

By using this procedure we are able to make sure that one variable is influenced by only one or a few objective values. That means that during ranking (and the subsequent selection and production of offspring), only those objective values are used which influence this variable. This allows the locally good features/variables to assert themselves relatively easily, rather than being suppressed by other bad variables in the same individual.

5 Application of Complementary Selection and Variation

For all our experiments we used an extended evolutionary algorithm previously employed successfully in many of our other real-world applications (see for instance [12]). The structure and the main operators are shown in figure 8. We employed the following evolutionary parameters:

- discrete recombination, recombination rate of 1,
- real valued mutation with medium sized mutation steps (mutation range of 0.03, mutation precision of 20),
- multi-objective ranking using PARETO ranking and goal attainment, sharing between individuals, selection pressure of 1.7, stochastic universal sampling and elitist selection by employing a generation gap of 0.9, an archive for good solutions found (currently without reinsertion into the population).

Normally, we use multiple subpopulations with migration, different strategies for each subpopulation and competition between subpopulations. However, the experiments for this paper involved a very small number of individuals (25-35 individuals) and it would make no sense to use multiple subpopulations for such a small number of individuals.

5.1 Example using extended benchmark functions

We constructed an example function using a number of standard benchmark functions. Each of the multiple objective values is calculated by one well-known test function. The special characteristic is that only a few of the variables are used for the calculation of one objective value. In addition, some variables are used for the calculation of more than one objective value.

In this example we employed the following two test functions: hypersphere function (DeJong's function 1) and 'Moved axis parallel hyper-ellipsoid 1c' (the minimum of this function is located at `Var = [5, 10, 15, 20, 25, 30, ...]`). We used 10 variables calculating 2 objective values. `ObjV1` is calculated from variables 1 to 5, `ObjV2` from variables 4 to 10. This results in the following assignment between variables and objective values:

```
% Assignment Var      to      ObjV
Var2ObjV = {[1, 2, 3],        [1]   ...
          ; [4, 5],           [1 2] ...
          ; [6, 7, 8, 9, 10],   [2] };
```

Our optimization runs were terminated after 30 generations. This is quite early and the optimization has not really converged to the PARETO front. We are, however, looking for a quick return. Our real-world application cannot be run for a longer time (the current application must use less generation because of running time and available resources). Thus, it would not make sense to compare long running optimizations.

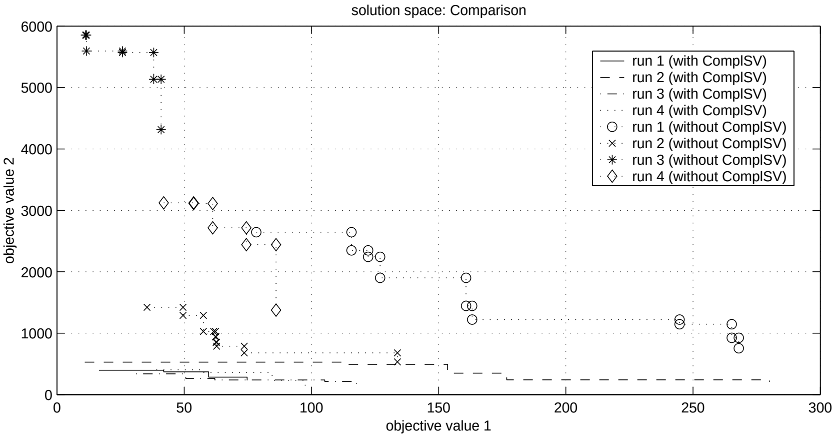


Figure 9. Comparison of PARETO front (formed from non-dominated individuals) of different optimization runs over 30 generations employing complementary selection and variation (labeled 'with ComplSV') or without complementary selection and variation (labeled 'without ComplSV')

The results presented compare representative runs for two types of optimization, which are identical except that some used and others did not use our new principle: complementary selection and variation as presented in section 4. The runs with complementary selection and variation are labeled `'with ComplSV'`, the others are labeled `'without ComplSV'`.

In figure 9, four different runs are presented for each type of optimization. We selected the non-dominated individual from each run and calculated the (limited) PARETO front formed by these individuals (in the solution space). These fronts are plotted on the same diagram.

We used minimization for both objective values. Thus, solutions which are more to the left and lower down represent better solutions.

When analyzing the diagram we can see that the runs using complementary selection and variation performed better than the runs which did not use this principle. Unfortunately, we do not have a statistical evaluation or performance assessment of these results at hand. This will be part of our further work.

Our main goal with these (simple) experiments was to open up the discussion on the principle presented, using examples accessible for everyone. We see this as only the beginning of many more experiments using other (more complex) functions, gauging the influence of the many parameters and operators used in evolutionary algorithms and implementing performance assessment for a more rigorous comparison of the optimization results.

5.2 Optimization of the automotive gearbox

Depicting complexity is no easy matter and examples from industrial practice are particularly difficult to compare because they cannot be reproduced exactly. Hard facts (in a mathematical sense) cannot be shown. Many more tests will be necessary in order to be able to better quantify the advantage of the new procedure in technical applications.

With regard to our long-term objective of optimizing several thousand parameters, we will have to make compromises and limit the scope. When carrying out experiments which require a day’s preparation on expensive equipment and in the case of calculations which take 10 to 20 seconds per individual, experience can only be gathered very slowly.

After successfully completing first, basic tests with 12 variables two years ago, we became braver and today are able to observe a total of around a hundred values, distributed over 6 chromosomes with the aid of complementary selection and mutation (within the chromosomes). We also use many attributes to compare the results and fix the optimization targets.

The effort needed to check the optimization methods grows disproportionately. Currently we tend to use samples to assess the behavior of the optimization during an optimization run. The final results are used to check the overall quality of the runs.

The illustration in figure 10 below shows that with procedures becoming increasingly complex, extended visualization methods must be employed.

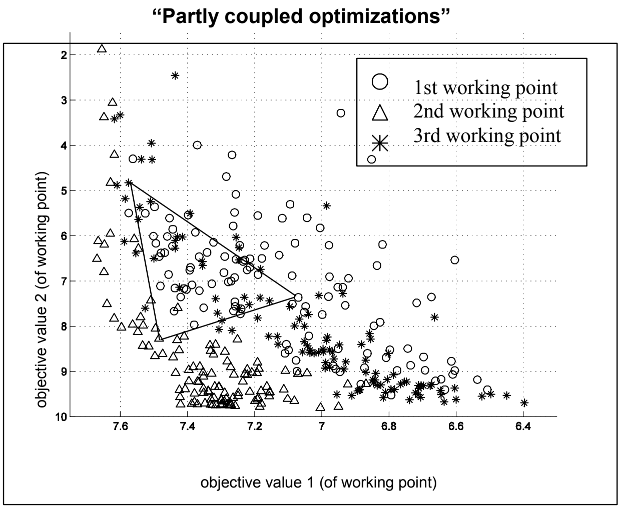


Figure 10. Depicting working points of the gearbox application looking for the goodness in each working point; each 'dot' represents a working point consisting of 2 objective values (x and y axis respectively), one individual consists of 3 working points/6 objective values (an allocation of the working points of each individual was not carried out in this diagram). The individual (triangle) outlined is dominated if we consider only one working point (2 obj. val.) but non-dominated if we take all of them into account.

If we consider only two or very few objectives, an optimization along the Pareto front as proposed in many EA makes sense. When the number of objectives increases (see figure 11), the number of dominated individuals drops and individuals that were considered dominated become non-dominated. In other words, there is a high probability of finding all the individuals on the non-dominated Pareto front, which therefore loses its significance. There is no longer any possible differentiation; all individuals have the same rank. In this case, selection without pressure is equivalent to a random selection. For this reason it is important for complex real-world applications to have extended MO strategies.

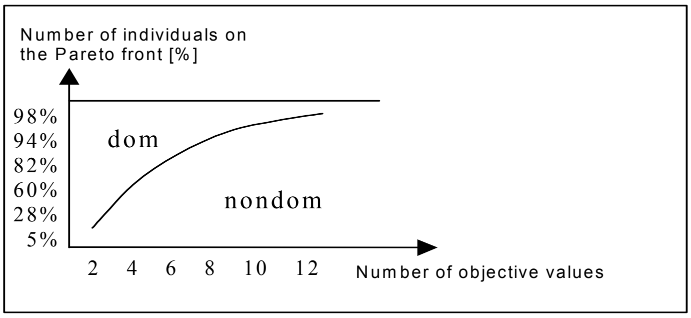


Figure 11. By increasing the number of objective values, the Pareto front loses its selective significance

6 Conclusion

Scientists have repeatedly observed that natural adjustments to changed living conditions are made more quickly in nature than can be explained by known evolutionary procedures. These kinds of observation support creationism which radically questions Darwinian evolutionary theory [5]. One can also, however, ask oneself which additional intelligent optimization procedures and "random rules" (which is, of course, a contradiction in terms) are employed in nature, which could explain the discrepancy between theory and observation.

This does not in any way resolve the conflict between those who see chance as life-inducing and those who see a purpose behind every development from the beginning of time (was chance created by purpose?). In order to obtain useful results in practice, new patterns and rules will be looked for where it was assumed that chance was at work. Where fixed rules have been established, new deviations, sounds and other random phenomena will be searched for. The crossover which is considered random supplements the Mendelian laws, whilst complementary variation provides evolutionary chance with new deterministic rules. Simple models such as complementary selection are always particularly nice because the effort to gain ratio is right. Furthermore, it is easier to extend a basis which is simple.

When optimizing complex, real-world systems the question of the scalability of the optimization procedures used arises time and time again. It is often the case that no concrete statements exist regarding their scalability. Even at the speed of today's computers, time-consuming comparisons of these real-world systems cannot be made or are difficult to make due to extremely long computing times.

In order to manage large systems, therefore, an attempt is made to decompose them. Unfortunately, this can only be done easily in very few cases since interactions existing between different areas prevent complete decomposition. However, in many real-world applications there is no complete coupling of all sub areas. Thus, partial decomposition is possible.

We have presented an approach for this kind of partial decomposition. The utility of the method was demonstrated employing extended, standard benchmark functions. We have also presented an overview of our results from the parameter optimization of an automotive gearbox system. Given the limited resources available and the high cost of a full optimization, the application of complementary selection and variation produced notably better results than our (already very good) extended evolutionary algorithm [13]. In particular, the much improved scaling of the problem size regarding additional parameters and objective values (employing the simulation of additional working points) allowed us to derive better results in the very limited time available.

The further development of complementary crossover will connect the following fields of knowledge: behavioral psychology, genetics and bioinformatics. The latter can help to check the efficiency of hypothetical procedures before time-consuming trials or observations in other areas are undertaken.

References

[1] *Allchin, D.*: The Dilemma of Dominance.

[2] *Angeline, P. J.*: Morphogenic Evolutionary Computations: Introduction, Issues and Example. Evolutionary Programming, pp. 387-401, 1995.

[3] *Darwin, C.*: On the Origin of Species by mean of natural selection. first edition, 1859, Leipzig: Verlag Philipp Reclam, 1984.

[4] *Fonseca, C. M. and Fleming, P. J.*: An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation, 3(1), pp. 1-16, 1995.

[5] *Gitt, W.*: Am Anfang war die Information, Hänssler Verlag, 1994.

[6] *Jacob, F.*: La logique du vivant, Gallimard, 1976.

[7] *Lewis, J., Hart, E. and Ritchie, G.*: A Comparison of Dominance Mechanisms and Simple Mutation on Non-Stationary Problems. PPSN 1998, Lecture Notes in Computer Science, volume 1498, pp. 139-148, 1998.

[8] *Lodish, H.*: Molecular Cell Biology. 4th ed., W. H. Freeman and Company, 2001.

[9] *Losos J.*: Wenn die Evolution sich wiederholt, Spektrum der Wissenschaft, 5, 2001.

[10] *Mayr, E.*: Histoire de la Biologie. Diversité, évolution et hérédité (orig. The growth of biological thought. Diversity, evolution and inheritance), Librairie Arthème Fayard, 1989.

[11] *Monod*: Le hasard et la necessité. Essai sur la philosophie naturelle de la biologie moderne, Seuil, 1970.

[12] *Pohlheim, H.*: Evolutionäre Algorithmen - Verfahren, Operatoren, Hinweise aus der Praxis. Berlin, Heidelberg: Springer-Verlag, 1999. http://www.pohlheim.com/eavoh/

[13] *Pohlheim, H.*: *GEATbx* - Genetic and Evolutionary Algorithm Toolbox for *Matlab*. http://www.geatbx.com/, 1994-2004.

[14] *Rasheed, K. and Hirsh H.*: Learning to be Selective in Genetic-Algorithm-Based Design Optimization. Artificial Intelligence in Engineering, Design, Analysis and Manufacturing, 13, pp. 157-169, 1999

[15] *Reinhard, E.*: Pharmazeutische Biologie, Band 1, 5. Auflage, Wissenschaftliche Verlagsgesellschaft Stuttgart, 1995.

[16] *Singh, A.*: Giving Genes their Voice: a Survey of Information Expression Mechanisms in Genetic Algorithms. 2002.

[17] *Wickler, W.*: Die Evolution der Geschlechter. Spektrum der Wissenschaft Dossier, 5, 2003.

[18] *Zitzler, E. and Thiele, L.*: An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 1998.

[19] *Zwirn, H.*: La complexité. La science du XXIème siècle, 1980.