Evolutionary Multi-Objective Optimisation with a Hybrid Representation

Tatsuya Okabe
Honda Research Institute Europe
Carl-Legien Strasse 30,
63073 Offenbach/M, Germany
tatsuya.okabe@honda-ri.de

Yaochu Jin
Honda Research Institute Europe
Carl-Legien Strasse 30,
63073 Offenbach/M, Germany
yaochu.jin@honda-ri.de

Bernhard Sendhoff
Honda Research Institute Europe
Carl-Legien Strasse 30,
63073 Offenbach/M, Germany
bernhard.sendhoff@honda-ri.de

Abstract- For tackling multi-objective optimisation (MOO) problem, many methods are available in the field of evolutionary computation (EC). To use the proposed method(s), the choice of the representation should be considered first. In EC, often binary representation and real-valued representation are used. In this paper, we propose a hybrid representation, composed of binary and real-valued representations for multi-objective optimisation problems. Several issues such as discretisation error in the binary representation, self-adaptation of strategy parameters and adaptive switching of representations are addressed. Experiments are conducted on five test functions using six different performance indices, which shows that the hybrid representation exhibits better and more stable performance than the single binary or real-valued representation.

1 Introduction

In order to solve multi-objective optimisation problems a wide range of methods have been proposed in the field of evolutionary computation (EC); examples are multi-objective GA (MOGA), non-dominated sorting GA (NSGA), fast elitist NSGA (NSGA-II), Pareto archived ES (PAES), see [Coe01, Deb01] for a comprehensive list.

Usually all of these algorithms have a specific representation which is used to encode the design parameters of a particular problem, although some can be used in conjunction with a variety of different representations (See [Rot02] for representation). In evolutionary computation two types of representations are widely used: the binary representation, which is motivated by the building block hypothesis in genetic algorithms (GA) [Gol89] and the real-valued representation which is motivated by the idea to choose the natural problem representation especially in the context of evolution strategies (ES) [Rec94]. In particular the real-valued representation has frequently been used also with genetic algorithms and the separation between GA (binary encoding[1]) and ES (real-valued encoding) is certainly not valid any longer [Deb95, Ono97, Esh93, Deb02b]. On the contrary, the pragmatic view to choose the representation and the variation operators which empirically perform best is widely accepted, not least due to the lack of rigorous theoretical results.

Hybrid representations have been used before in single objective optimisation either because the optimisation problem was mixed continuous and discrete or because the hybrid representation was used to improve the search process. Barbulescu et al. [Bar00] used a dynamic Gray code. They mentioned that a local optimum under one representation may not be a local optimum under another. They propose the "shifting" method to add a bias to the binary representation. Shifting helps the evolutionary algorithms to escape local optima. Schnier and Yao [Sch00] used a Cartesian representation in conjunction with a pseudo-polar representation. On several test functions, the mixed representation showed good performance. However, the results were also strongly dependent on the variation operators that were employed.

In this paper, we propose a hybrid representation consisting of a binary representation and a real-valued representation for multi-objective optimisation problems. Our original motivation was to exploit the different dynamics of the populations during the search process. This dynamics must be analysed in both the parameter as well as in the fitness space [Oka02]; it is strongly influenced by the choice of the representation and the variation operators.

We will discuss the framework in which we use the hybrid representation in Section 3 and we will compare the hybrid algorithm to two standard methods on a variety of different test functions and performance indices in Section 4. In Section 5, we discuss our results and we conclude in Section 6. As a starting point, we will briefly summarise the performance indices and test functions that will be used in this paper in Section 2.

2 Performance Indices and Test Functions

Judging the performance of multi-objective optimisation algorithms is not an easy task; this is reflected by the large number of performance indices[2] that can be found in the lit-

[1] Note that we use the terms representation and encoding synonymously.

[2] Often performance measures are referred to as metrics without proving that the necessary conditions are fulfilled, therefore, we will only denote

erature [Coe01, Deb01, Kno02, Zit02, Zit03, Oka03]. Since no single performance index is sufficient for comparing two algorithms, see [Oka03] for a criticism of performance indices, it seems most appropriate to use a portfolio of different performance indices (PIs).

We can group PIs into five different classes: (1) distance-based accuracy, (2) volume-based accuracy, (3) distance-based distribution, (4) niche-based distribution, and (5) spread-based measures (See [Oka03]). In this paper, we select one PI from each class. Additionally, we also use the $R1_R$ index proposed by Hansen [Han98]:

- **Deb's γ index in (1)** [Deb01, Deb02] **:** This PI is based on the distance from the solution set, S, to the Pareto optimal solution set, P, to measure the accuracy.
- **Zitzler's H index in (2)** [Zit00] **:** This PI is based on the size of the area that is dominated by S to measure the accuracy.
- **Deb's Δ' index in (3)** [Deb00, Deb01] **:** This PI is for measuring the distribution of S. The Euclidean distances between consecutive solutions in S are used.
- **Zitzler's $\mathcal{M}_2$ index in (4)** [Zit00] **:** This PI is based on the concept of niching for measuring the distribution.
- **Zitzler's $\mathcal{M}_3$ index in (5)** [Zit00] **:** This PI is for measuring the spread of S. The distances between the boundary solutions are calculated.
- **Hansen's $R1_R$ index** [Han98] **:** This PI is based on a decision maker's (DM) preference. The solution set which is often selected by the DM is the better one.

For each of the listed indices, we calculate the rank of the algorithms, i.e., *Rank 1* refers to the best optimiser, *Rank 2* to the second best and so on. Thereafter, we average all six ranks of the six performance indices ending up with one scalar performance measure for each algorithm.

We use five different test functions, i.e., SCH1, ZDT1, ZDT2, ZDT3 and FON2 from [Deb01]. The details are shown in Table 1. For all of these functions, the true Pareto front, PF_{true}, can be determined analytically [Oka02]. We use the test functions with the following dimensions: $n = 2$, $n = 20$ and $n = 50$.

Table 1: Test functions.

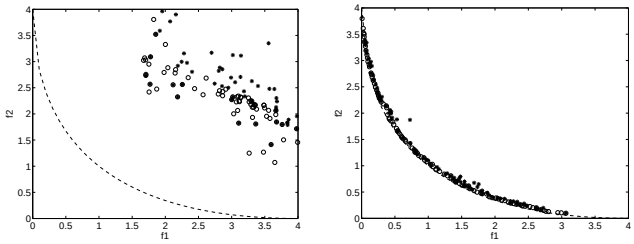
SCH1	$f_1 = \frac{1}{n}\sum_{i=1}^{n} x_i^2$ $f_2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - 2.0)^2$ $-4 \leq x_i \leq 4$
ZDT1	$f_1 = x_1$ $g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $f_2 = g \times (1.0 - \sqrt{\frac{f_1}{g}})$ $0 \leq x_i \leq 1$
ZDT2	$f_1 = x_1$ $g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $f_2 = g \times (1.0 - (\frac{f_1}{g})^2)$ $0 \leq x_i \leq 1$
ZDT3	$f_1 = x_1$ $g(x_2, ..., x_n) = 1.0 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $f_2 = g \times (1.0 - \sqrt{\frac{f_1}{g}} - (\frac{f_1}{g})sin(10\pi f_1))$ $0 \leq x_i \leq 1$
FON2	$f_1 = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right)$ $f_2 = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$ $-2 \leq x_i \leq 2$

3 Hybrid Representation and Genetic Operators

3.1 Motivations

Our original motivation for using hybrid representations was to exploit the different dynamics of populations based on different representations during the search process. In [Oka02], the dynamics of evolution strategies (ES) based on the real-valued representation were observed by projecting the normal distribution onto the fitness space. Since the ES-mutation is carried out by adding a normally distributed random value to the current parent, a more local search can be realised (we omitted the recombination operator from this analysis). In genetic algorithms with binary encoding the main search operator is crossover. By exchanging parts of the chromosomes between several parents, new offspring are generated. It is intuitive that the resulting offspring distribution and therefore, the dynamics will be very different from the one generated by ES-mutation.

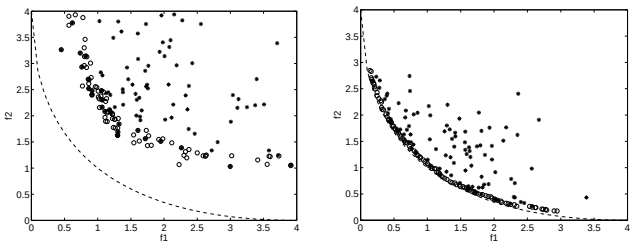
We show two typical snapshots of the different dynamics on test function SCH1 (20 dimensions) in Figure 1 and 2. In Table 2, the parameters are shown.



(a) Generation 25 (b) Generation 200

Figure 1: Snapshot of the distribution of parents (circles) and offspring (dots) at generation 25 and 200 using real-valued representation and ES-mutation.

all measures as performance indices.



(a) Generation 25 (b) Generation 200

Figure 2: Snapshot of the distribution of parents (circles) and offspring (dots) at generation 25 and 200 using binary encoding and crossover

Table 2: Parameters for investigating different dynamics.

Figure 1	Representation	Real-valued representation
	Population size	100
	Mutation	ES-mutation See Section 3.3
	Self-adaptation	Also see Section 3.3
	Recombination	Not used
	Initial step size	$[0, 1]$
	Lower step size	$0.004 \times \lfloor x_i \rfloor$
	Selection	Crowded tournament selection by Deb [Deb02]
Figure 2	Representation	Binary encoding
	Population size	100
	Bits per one design parameter	20
	Mutation	Bit flip
	Mutation Rate	0.01
	Crossover	One-point crossover
	Crossover Rate	0.9
	Coding	Gray coding
	Selection	Crowded tournament selection by Deb [Deb02]

In general in the early generations (exploration phase), the distribution should be wider. Following this argument, the distribution of offsprings in Figure 2 is better. However, in later generations (exploitation phase), the distribution should be concentrated near the Pareto front. In this case, the distribution of offsprings in Figure 1 seems to be better.

Our basic idea is to exploit both dynamics. Using the binary representation we want to realize a wider distribution of offsprings in the early stage and using the real-valued representation with ES-mutation we favour a concentrated distribution of offsprings in the later stage to facilitate efficient local search.

3.2 Hybrid Representation

In the hybrid representation, each individual has two chromosomes. One is the binary representation and the other is the real-valued representation of the design parameters. Additionally, each individual has one special chromosome with two alleles that indicate which representation is to be used, i.e. allele *B* specifies the binary representation (and respective operators) and *R* the real-valued representation (and ES-mutation).

The hybrid representation is shown in Figure 3. Since both parameter chromosomes have the same value the encoding is redundant and synchronisation is necessary. Note that for each individual either the binary or the real-valued representation is used while the other representation is only updated. Although the analogy with dominant and recessive alleles in biological systems can be seen, it is not entirely correct. In our system the value of the design parameter is always identical (thus the alleles are always the same), it is the representation of the parameters that changes, which would be like changing the genetic composition of the allele without changing its actual value.

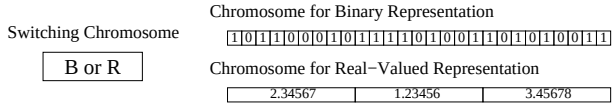


Figure 3: The hybrid representation using three chromosomes.

To make sure that the binary and the real-valued chromsomes always encode the same parameter value they have to be synchronised:

- If the switching chromosome reads *B*, the data in the binary representation are copied to the real-valued representation.
- If the switching chromosome reads *R*, the data in the real-valued representation are copied to the binary representation. If the value in the real-valued representation is out of the encoding range, the value is set to the nearest boundary.

In step 2, a discretisation error will occur. Furthermore, the values which are out of the encoding range are shifted. On some test functions, this discretisation error and the shifting shows significant influence. Both are very important when we compare the performance of the real-valued representation and the binary representation. We will discuss this topic in more detail in Section 5.

3.3 Genetic Operators

Associated with the binary representation are the one-point crossover and GA-type mutation, i.e. flipping bits with probability p_m. The crossover operator relies on a sufficient number of parents to act upon, therefore we apply the crossover operator to the whole population irrespective whether the first chromosome indicates *B* or *R*. Therefore,

parents with the value *R* in their switching chromosome also join the crossover operation, see Figure 4. GA-mutation is only applied to individuals with active binary encoding, i.e. with the allele *B* in their switching chromosome.

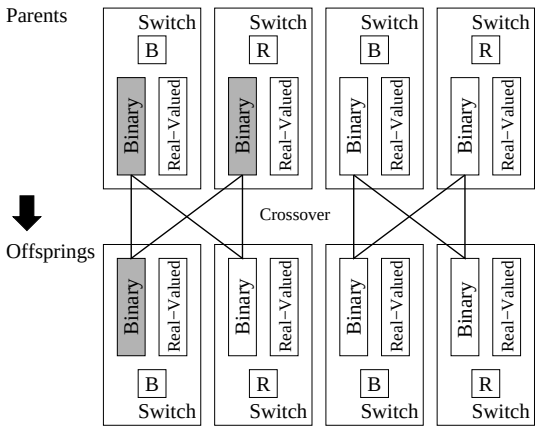


Figure 4: Crossover in the hybrid representation. Even parents with allele *R* take part in the crossover operation.

ES-mutation is applied to individuals with active real-valued representation, thus with the allele *R* in the switching chromosome. Furthermore, following the concept of mutative self-adaptation, the standard deviations of the normal distribution, which are called step-sizes or strategy parameters, are also mutated (dimension n):

$$\sigma_i(t) = \sigma_i(t-1)\exp(\tau' z)\exp(\tau z_i)\,;\quad i = 1,\ldots,n, \quad (1)$$

where z and z_i are $\mathcal{N}(0,1)$ distributed random values and the parameters, τ and τ' have the standard values

$$\tau = \frac{1}{\sqrt{2n}}, \quad \tau' = \frac{1}{\sqrt{2\sqrt{n}}}. \quad (2)$$

However, the problem with the self-adaptation is that in the case when the active encoding is the binary one, the parameter value is changed without any changes in the strategy parameter. If at a later stage the representation is switched (back) to the real-valued one, the setting of strategy parameters is likely to be inappropriate.

Although this problem cannot be fully solved, we use the following approximate method to adapt the strategy parameters even during binary representation based search. If an offspring is generated by the crossover operator, we measure its distance to the parent and directly use this distance to adjust the strategy parameters σ_i along each coordinate axis, see Figure 5. Since the strategy parameters tend to converge to small values rather quickly the overestimation of the size that might occur using the above outlined method is not harmful to the overall search process.

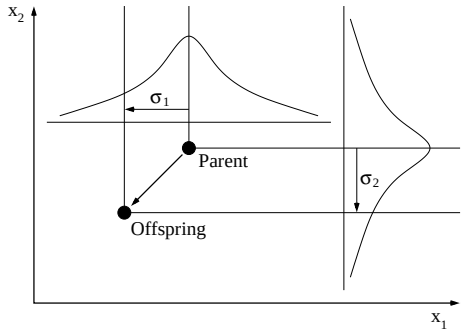


Figure 5: Self-adaptation of the strategy parameters of ES-mutation during the search process with active binary representation.

3.4 Switching Representations

We initialise the switching chromosome of the population with equal probabilities for the alleles *B* and *R*, i.e. on average half of the individuals will start with an active binary representation and half with an active real-valued representation.

During the optimisation the offspring inherits the switching chromosome with equal probability from either of its parents. In addition, the chromosome is mutated or flipped (from *B* to *R* and *R* to *B*) with a probability ($p_{m,sc} = 0.01$) In order to avoid extinction of any of the alleles, at least 5% of the whole population is forced into having an active binary or real-valued representation.

4 Results

On the five test functions, we run optimisations with the binary representation, with the real-valued representation, and with the hybrid representation. Table 3 shows the parameter settings.

Table 3: Parameters used in the experiments.

Population Size	100
Generations	500
Crossover Points	1
Crossover Rate	0.9
Mutation Rate (Binary)	0.01
Bits per one design parameter	20
Coding	Gray coding
Initial Step Size	[0, 1] for SCH1, FON2 [0, 0.01] for ZDT1, 2, 3
Lower Step Size	$0.004 \times \lvert x_i \rvert$

As the selection operator, we use the crowded tournament selection proposed by Deb [Deb02]. The following

steps are carried out: (1) sort offspring by their rank, (2) sort offspring by the crowded distance within the same rank, and (3) select the best offspring deterministically.

The obtained solution sets are shown in the appendix. Using the six performance indices (PIs) outlined in Section 2, we determine the rank of each algorithm. The parameters of all PIs are shown in Table 6 in the appendix and the ranks for each PI are shown in the appendix as the median over 30 runs.

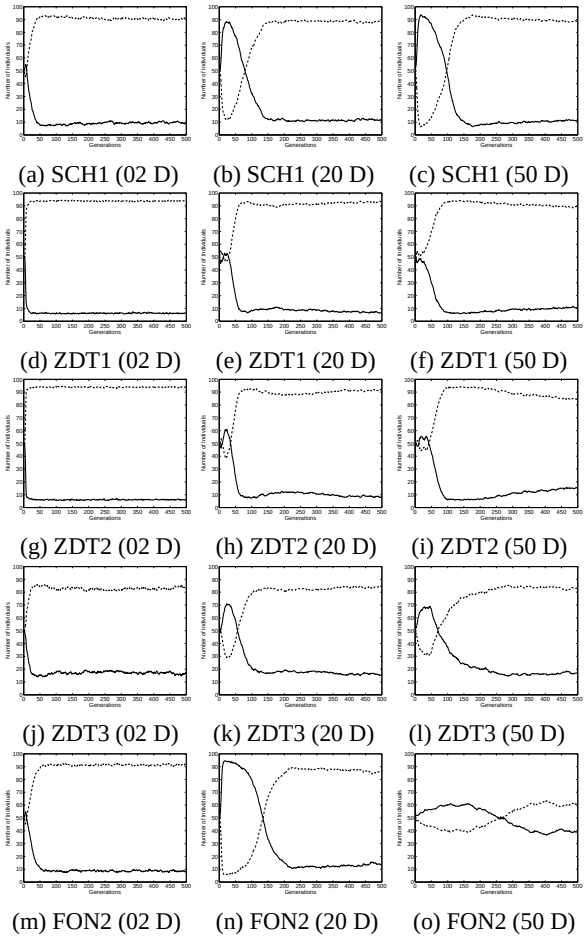
The overall rank, i.e. the average over the six performance indices (which are the median over 30 runs), is shown in Table 4 for each algorithm for each test functions and for each of the three different dimensions (2, 20, 50). The raw ranks can be seen in Table 7 in the appendix.

Table 4: Averaged ranks for each test function.

Function	Binary	Real-Valued	Hybrid
SCH1 $n = 2$	2.67	1.67	1.67
SCH1 $n = 20$	2.33	1.50	1.83
SCH1 $n = 50$	2.67	1.83	1.00
ZDT1 $n = 2$	2.33	2.00	1.50
ZDT1 $n = 20$	2.00	2.67	1.33
ZDT1 $n = 50$	2.00	2.50	1.33
ZDT2 $n = 2$	2.33	2.00	1.50
ZDT2 $n = 20$	2.17	2.50	1.17
ZDT2 $n = 50$	2.00	2.67	1.17
ZDT3 $n = 2$	2.33	2.00	1.50
ZDT3 $n = 20$	2.00	2.67	1.33
ZDT3 $n = 50$	1.83	2.67	1.33
FON2 $n = 2$	2.83	1.50	1.67
FON2 $n = 20$	2.50	1.33	2.00
FON2 $n = 50$	1.33	2.17	1.83

First, we compare the binary representation and the real-valued representation. The real-valued representation shows good performance on SCH1, FON2 (except $n = 50$) test functions and for all low dimensional cases, i.e., ZDT1 ($n = 2$), ZDT2 ($n = 2$) and ZDT3 ($n = 2$). On the other hand, the binary representation shows good performance on the high dimensional cases of ZDT1, ZDT2 and ZDT3 ($n = 20, 50$) and on FON2 ($n = 50$). Apparently, we can classify the five test functions into two classes. The first class is the group of SCH1 and FON2, and the second one is the group of ZDT1, ZDT2 and ZDT3. We will discuss the characteristics of the two classes in the next section.

Second, we compare the hybrid representation with both other representations. The averaged ranks are less than or equal to 2. Compared to the binary and the real-valued representations, the high performance of the hybrid representation is very stable, i.e., it is the best or close to the best algorithm for all test cases. The history of the active representation during the optimisation is shown in Figure 6. In order to minimise the stochastic influence as well as the genetic drift on finite populations when there is little selective pressure, we show the average of 30 runs.



(a) SCH1 (02 D) (b) SCH1 (20 D) (c) SCH1 (50 D)

(d) ZDT1 (02 D) (e) ZDT1 (20 D) (f) ZDT1 (50 D)

(g) ZDT2 (02 D) (h) ZDT2 (20 D) (i) ZDT2 (50 D)

(j) ZDT3 (02 D) (k) ZDT3 (20 D) (l) ZDT3 (50 D)

(m) FON2 (02 D) (n) FON2 (20 D) (o) FON2 (50 D)

Figure 6: The history of the allele of the switching chromosome. The average of 30 runs are shown. The x-axis labels the generations and the y-axis the number of individuals in the population with binary representation (solid line) and with real-valued representation (dotted line).

The results for 2 dimensions show the clear tendency that the real-valued representation is better than the binary representation. This tendency is in good agreement with the better performance of the real-valued representation in these cases.

For the 20 dimensional cases, the binary representation is selected in the early generations but the real-valued representation is selected in the later generations. Although we did not encourage this behaviour, it corresponds to our initial motivation outlined in Section 3.

In the 50 dimensional cases, the binary representation is also selected in the early generation and the real-valued

representation in the late generation. However, the tendency in FON2 ($n = 50$) is not so clear.

5 Discussions

5.1 Self-adaptation in the Binary Representation

In order to realise strategy parameter adaptation with the hybrid representation, we proposed self-adaptation in the framework of the binary representation in Section 3.3. To see whether this self-adaptation works or not, we fix the switching chromosome to *B* for all generations and record the history of step sizes in the chromosomes, which is shown in Figure 7 for the ZDT1 ($n = 50$) test function. The plot shows all 50 strategy parameters averaged over 30 runs. It is evident that a "typical" convergence behaviour of the step sizes is realized.

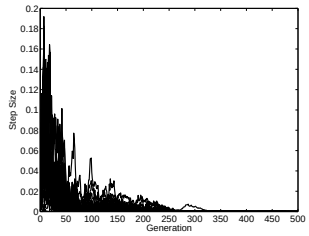


Figure 7: The history of step sizes on ZDT1 ($n = 50$). The switching chromosome is fixed to *B*. The average values of 30 runs are shown.

5.2 Characteristic of Test Functions

As explained before, the binary representation seems to be particularly suitable for the higher dimensional cases of ZDT1, ZDT2, ZDT3 and FON2. On the other hand, the real-valued representation is suitable for the low dimensional cases and for the test functions SCH1 and FON2 (except 50 dimensions). In previous sections we proposed to group the test functions into two classes and observe their characteristics. If we analyse the Pareto fronts in parameter space, we get the following equations:

SCH1, FON2

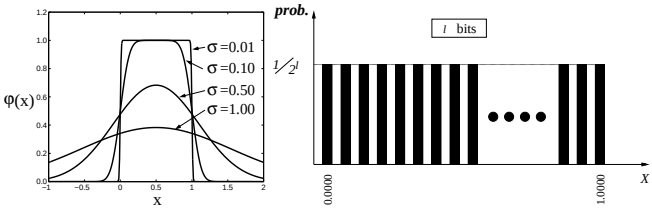
$$\begin{aligned} & x_1 = x_2 = ... = x_n \\ & 0.0 \leq x_i \leq 2.0 \quad \text{for SCH1} \\ & -1/\sqrt{n} \leq x_i \leq 1/\sqrt{n} \quad \text{for FON2} \end{aligned} \tag{3}$$

ZDT1, ZDT2, ZDT3

$$\begin{aligned} & 0 \leq x_1 \leq 1 \\ & x_i = 0 \qquad (i = 2, ..., n). \end{aligned} \tag{4}$$

Apparently, SCH1 and FON2 have solutions which do not lie on a boundary, whereas the solutions for ZDT1, ZDT2 and ZDT3 all lie on a boundary.

For uniformly distributed parents in $[0, 1]$, we calculate the distribution of offsprings theoretically for the binary and the real-valued representation with crossover and ES-mutation, respectively. The results are shown in Figure 8. We note that the distribution of the binary representation is shown by the respective probabilities whereas for the real-valued representation the probability density function (pdf) is shown.



(a) Real-valued Representation (b) Binary Representation

Figure 8: The offspring distribution under the assumption of uniformly distributed parents.

The distribution of the offspring is very different. The distribution of the binary representation is still uniform but the distribution of the real-valued representation is not uniform. The probability at the centre is higher than at the boundaries. We may expect that the relation between the shape of the offspring distribution and the nature of the Pareto front in parameter space explains why some representations are more suitable for a particular problem class. However, further research is necessary to be able to draw some grounded conclusions.

5.3 On the Comparison of Binary and Real-valued Representations

In this paper, we compare binary, real-valued and hybrid representations. However, the real-valued representation apparently has disadvantages caused by the search space. The binary representation can search only discrete points within in the coding range. By setting the number of bits and the coding range, we restrict the search space. If the restricted search space includes the optimal solutions, the binary representation has a strong priori advantage. On the other hand, the real-valued representation is able to search the continuous space without any limitations. If the optimal solutions are not in the coding range for the binary representation, they cannot be identified.

Furthermore, for a fair comparison we have to take the previously mentioned discretisation error into account. In order to do this, we add the discretisation error in the real-valued representation. After ES-mutation, we convert the

real-valued representation to the binary representation and back. This way, we artificially discretise and restrict the search space also for the real-valued representation. The results are shown in Table 5. The results with conversion are denoted as *real-valued (E)*.

Table 5: Averaged ranks on each test function.

Function	Binary	Real-Valued (E)	Hybrid
SCH1 $n = 2$	2.67	1.33	2.00
SCH1 $n = 20$	2.33	1.83	1.67
SCH1 $n = 50$	2.67	1.50	1.17
ZDT1 $n = 2$	2.50	1.67	1.33
ZDT1 $n = 20$	2.67	1.67	1.67
ZDT1 $n = 50$	2.33	1.50	1.50
ZDT2 $n = 2$	2.50	1.67	1.33
ZDT2 $n = 20$	3.00	1.67	1.33
ZDT2 $n = 50$	2.67	1.50	1.33
ZDT3 $n = 2$	2.50	1.67	1.33
ZDT3 $n = 20$	2.67	2.00	1.33
ZDT3 $n = 50$	2.33	1.83	1.33
FON2 $n = 2$	2.83	1.33	1.67
FON2 $n = 20$	2.67	1.50	1.83
FON2 $n = 50$	1.33	2.17	1.83

The performance of the binary representation is the worst except for FON2 ($n = 50$). In all other cases the real-valued representation with conversion is better. The hybrid representation still exhibits stable superior performance.

6 Conclusion

In this paper, we suggested and analysed a hybrid representation consisting of a binary representation and a real-valued representation of which only one is active in each individual in each generation. We tested the hybrid representation on five different functions for different problem space dimensions and measured the performance averaged over six different performance indices for multi-objective optimisation. First of all the pragmatic observation can be made that the hybrid representation exhibits stable superior performance compared to the other two encodings. There are different reasons. Our initial motivation for the hybrid representation was the observation of different dynamics both theoretically as well as empirically for different encodings. Our intuitive idea to combine a wider search at early generations (binary representation) with a more restricted search at later generations (real-valued representation) was confirmed by the empirical results of the adaptation of the representation shown in Figure 6. At the same time, the discussion in the previous section shows, that the comparison has to be made with great care, because the discretisation error combined with specific properties of test functions can produce unexpected effects.

Acknowledgement

The authors would like to thank E. Körner, A. Richter and M. Olhofer for their support and discussions.

Bibliography

[Bar00] Laura Barbulescu and Jean-Paul Watson and Darrell Whitley (2000) "Dynamic Representations and Escaping Local Optima: Improving Genetic Algorithms and Local Search", AAAI/IAAI 2000, p.p. 879-884.

[Coe01] Carlos A. Coello Coello, David A. Van Veldhuizen and Gary B. Lamont (2001) "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer Academic Publishers.

[Deb95] Kalyanmoy Deb, Ram B. Agrawal (1995) "Simulated Binary Crossover for Continuous Search Space", Complex Systems, vol. 9, p.p. 115-148.

[Deb00] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap and T. Meyarivan (2000) "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II", Proceedings of the Parallel Problem Solving from Nature VI - PPSN VI, p.p. 849-858.

[Deb01] Kalynmoy Deb (2001) "Multi-Objective Optimization using Evolutionary Algorithms", John Wiley & Sons, LTD.

[Deb02] Kalyanmoy Deb, Amrit Pratap, Sameer Agrawal and T. Meyarivan (2002) "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, p.p. 182-197.

[Deb02b] Kalyanmoy Deb, Ashish Anand and Dhiraj Joshi (2002) "A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization", Evolutionary Computation, vol. 10, no. 4, p.p. 371-395.

[Esh93] Larry J. Eshelman and David J. Schaffer (1993) "Real-coded Genetic Algorithms and Interval Schemata", Foundations of Genetic Algorithms, p.p. 187-202.

[Gol89] David E. Goldberg (1989) "Genetic Algorithms in Search, Optimization, and Machine Learning", MA: Addison-Wesley.

[Han98] Michael Pilegaard Hansen and Andrzej Jaszkiewicz (1998) "Evaluating the quality of approximations to the non-dominated set", Institute of Mathematical Modeling, Technical University of Denmark, Denmark, IMM Technical Report IMM-REP-1998-7.

[Kno02] Joshua Knowles and David Corne (2002) "On Metrics for Comparing Nondominated Sets", Proceedings of Congress on Evolutionary Computation - CEC 2002, p.p. 711-716.

[Oka02] Tatsuya Okabe, Yaochu Jin and Bernhard Sendhoff (2002) "On the Dynamics of Evolutionary Multi-Objecitve Optimisation", Proceedings of the Genetic and Evolutionary Computation Conference - GECOO 2002, p.p. 247-255.

[Oka03] Tatsuya Okabe, Yaochu Jin and Bernhard Sendhoff (2003) "A Critical Survey of Performance Indices for Multi-Objective Optimisation", Proceedings of IEEE Congress on Evolutionary Computation - CEC 2003 (Accepted).

[Ono97] Isao Ono and Shigenobu Kobayashi (1997) "A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover", Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-7), p.p. 246-253.

[Rec94] Ingo Rechenberg (1994) "Evolutionsstrategie '94 (German)", Frommann-holzboog.

[Rot02] Franz Rothlauf (2002) "Representations for Genetic and Evolutionary Algorithms", Springer.

[Sch00] Thorsten Schnier and Xin Yao (2000) "Using Multiple Representations in Evolutionary Algorithms", Proceedings of the 2000 Congress on Evolutionary Computation, p.p. 479-486.

[Zit00] Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele (2000) "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", Evolutionary Computation, vol. 8, no. 2, p.p. 173-195.

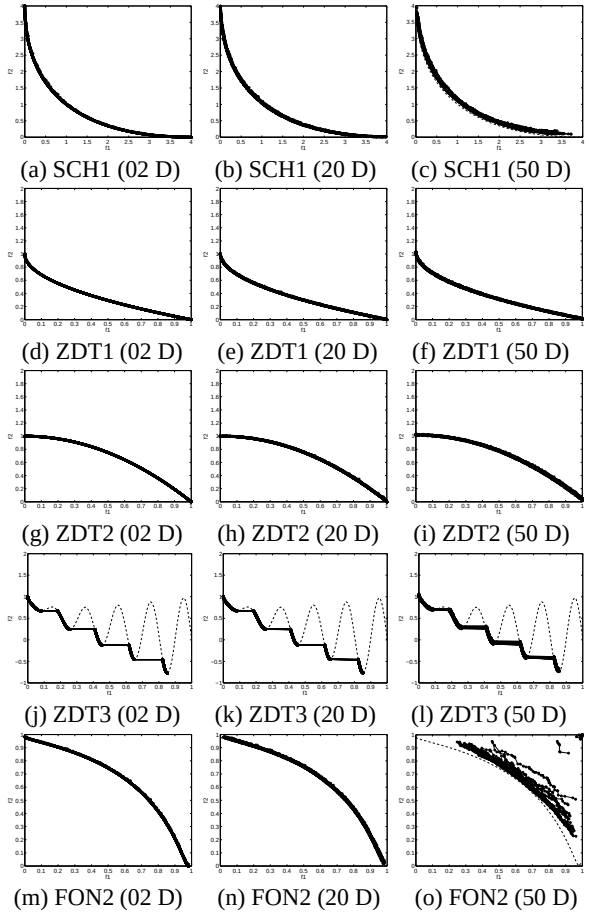
[Zit02] Eckart Zitzler, Marco Laumanns, Lothar Thiele, Carlos M. Fonseca and Viviane Grunert da Fonseca (2002) "Why Quality Assessment of Multiobjective Optimizers Is Difficult", Proceedings of the Genetic and Evolutionary Computation Conference - GECCO 2002, p.p. 666-673.

[Zit03] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca and Viviane Grunert da Fonseca (2003) "Performance Assessment of Multiobjective Optimizers: An Analysis and Review", IEEE Transactions on Evolutionary Computation 7(2), p.p. 117-132.

Appendix

Table 6: Parameter settings for PIs.

Pareto Solution Set for γ	500 solutions
Ref. Solution Set for $R1_R$	100 solutions
Origin for H	(5.0, 5.0) for SCH1
	(1.1, 6.0) for ZDT1, 2, 3
	(1.1, 1.1) for FON2
Niche Radius for $\mathcal{M}_2$	0.03250 for SCH1
	0.00741 for ZDT1
	0.00746 for ZDT2
	0.00920 for ZDT3
	0.00712 for FON2



(a) SCH1 (02 D) (b) SCH1 (20 D) (c) SCH1 (50 D)

(d) ZDT1 (02 D) (e) ZDT1 (20 D) (f) ZDT1 (50 D)

(g) ZDT2 (02 D) (h) ZDT2 (20 D) (i) ZDT2 (50 D)

(j) ZDT3 (02 D) (k) ZDT3 (20 D) (l) ZDT3 (50 D)

(m) FON2 (02 D) (n) FON2 (20 D) (o) FON2 (50 D)

Figure 9: Obtained solutions of hybrid representation (connected by a line). Dotted curve shows the boundary of a feasible region.

Table 7: Values of each PI and its ranks.

Function	PIs	Binary		Real-valued		Hybrid	
SCH1 $n = 2$	γ	0.00634	(3)	0.00518	(2)	0.00501	(1)
	H	22.23611	(3)	22.24827	(1)	22.24624	(2)
	Δ'	0.03096	(3)	0.02411	(1)	0.02736	(2)
	$\mathcal{M}_2$	96.65625	(3)	99.75758	(1)	99.64646	(2)
	$\mathcal{M}_3$	5.65681	(1)	5.65538	(3)	5.65677	(2)
	$R1_R$	0.31050	(3)	0.37850	(2)	0.38800	(1)
SCH1 $n = 20$	γ	0.03063	(3)	0.02869	(2)	0.02826	(1)
	H	21.71881	(3)	22.09109	(1)	22.08920	(2)
	Δ'	0.01454	(1)	0.02035	(2)	0.02057	(3)
	$\mathcal{M}_2$	99.60606	(3)	99.81818	(1)	99.81818	(1)
	$\mathcal{M}_3$	4.18587	(3)	5.53540	(1)	5.53528	(2)
	$R1_R$	0.02550	(1)	0.00550	(2)	0.00550	(2)
SCH1 $n = 50$	γ	0.17028	(3)	0.10788	(2)	0.08434	(1)
	H	19.79970	(3)	21.20691	(2)	21.53302	(1)
	Δ'	0.02421	(3)	0.01830	(2)	0.01747	(1)
	$\mathcal{M}_2$	94.98114	(3)	99.63636	(2)	99.69697	(1)
	$\mathcal{M}_3$	3.16285	(3)	4.57630	(2)	4.64827	(1)
	$R1_R$	0.00000	(1)	0.00000	(1)	0.00000	(1)
ZDT1 $n = 2$	γ	0.00086	(2)	0.00260	(3)	0.00073	(1)
	H	6.25978	(2)	6.25838	(3)	6.26041	(1)
	Δ'	0.00768	(3)	0.00446	(1)	0.00694	(2)
	$\mathcal{M}_2$	95.55789	(3)	99.86869	(1)	97.69072	(2)
	$\mathcal{M}_3$	1.41421	(2)	1.42385	(1)	1.41421	(2)
	$R1_R$	0.40425	(2)	0.26850	(3)	0.45000	(1)
ZDT1 $n = 20$	γ	0.02045	(2)	0.09802	(3)	0.00316	(1)
	H	6.23059	(2)	6.16139	(3)	6.25683	(1)
	Δ'	0.00540	(2)	0.02677	(3)	0.00475	(1)
	$\mathcal{M}_2$	99.79798	(2)	99.76768	(3)	99.83838	(1)
	$\mathcal{M}_3$	1.42885	(2)	2.60957	(1)	1.41421	(3)
	$R1_R$	0.00050	(2)	0.00000	(3)	0.19500	(1)
ZDT1 $n = 50$	γ	0.20261	(2)	0.45355	(3)	0.01380	(1)
	H	5.96233	(2)	5.69041	(3)	6.24074	(1)
	Δ'	0.01811	(2)	0.02841	(3)	0.00440	(1)
	$\mathcal{M}_2$	56.78571	(3)	64.73438	(2)	99.89899	(1)
	$\mathcal{M}_3$	1.55948	(2)	2.16096	(1)	1.42316	(3)
	$R1_R$	0.00050	(1)	0.00000	(3)	0.00050	(1)
ZDT2 $n = 2$	γ	0.00084	(2)	0.00349	(3)	0.00073	(1)
	H	5.92639	(2)	5.92492	(3)	5.92715	(1)
	Δ'	0.00723	(3)	0.00460	(1)	0.00618	(2)
	$\mathcal{M}_2$	95.60000	(3)	99.85859	(1)	97.69072	(2)
	$\mathcal{M}_3$	1.41421	(2)	1.45740	(1)	1.41421	(2)
	$R1_R$	0.39425	(2)	0.25100	(3)	0.43350	(1)
ZDT2 $n = 20$	γ	0.03158	(2)	0.19359	(3)	0.00286	(1)
	H	5.87376	(2)	5.56479	(3)	5.92440	(1)
	Δ'	0.00879	(2)	0.05428	(3)	0.00470	(1)
	$\mathcal{M}_2$	98.56122	(2)	37.29569	(3)	99.81818	(1)
	$\mathcal{M}_3$	1.38486	(3)	1.96969	(1)	1.41421	(2)
	$R1_R$	0.00050	(2)	0.00050	(2)	0.20350	(1)
ZDT2 $n = 50$	γ	0.33881	(2)	0.85850	(3)	0.01812	(1)
	H	5.41134	(2)	4.81161	(3)	5.89924	(1)
	Δ'	0.03257	(2)	0.08650	(3)	0.00454	(1)
	$\mathcal{M}_2$	29.62069	(2)	13.26923	(3)	99.87879	(1)
	$\mathcal{M}_3$	1.24585	(3)	1.51194	(1)	1.40644	(2)
	$R1_R$	0.00050	(1)	0.00000	(3)	0.00050	(1)
ZDT3 $n = 2$	γ	0.00101	(2)	0.00158	(3)	0.00099	(1)
	H	6.71830	(2)	6.71731	(3)	6.71884	(1)
	Δ'	0.01747	(3)	0.01213	(1)	0.01429	(2)
	$\mathcal{M}_2$	93.53763	(3)	99.82828	(1)	97.70103	(2)
	$\mathcal{M}_3$	1.96735	(2)	1.97834	(1)	1.96735	(2)
	$R1_R$	0.49350	(2)	0.21450	(3)	0.55000	(1)
ZDT3 $n = 20$	γ	0.00836	(2)	0.07717	(3)	0.00201	(1)
	H	6.67847	(2)	6.50971	(3)	6.71407	(1)
	Δ'	0.01332	(2)	0.02881	(3)	0.01249	(1)
	$\mathcal{M}_2$	99.75758	(2)	96.80681	(3)	99.76768	(1)
	$\mathcal{M}_3$	1.99387	(2)	2.63274	(1)	1.96463	(3)
	$R1_R$	0.00050	(2)	0.00000	(3)	0.12350	(1)
ZDT3 $n = 50$	γ	0.14768	(2)	0.49430	(3)	0.00987	(1)
	H	6.36853	(2)	5.73561	(3)	6.66769	(1)
	Δ'	0.02525	(2)	0.05105	(3)	0.01306	(1)
	$\mathcal{M}_2$	71.52113	(2)	55.05926	(3)	99.77778	(1)
	$\mathcal{M}_3$	2.16544	(2)	2.70820	(1)	1.94844	(3)
	$R1_R$	0.00050	(1)	0.00000	(3)	0.00050	(1)
FON2 $n = 2$	γ	0.00170	(3)	0.00133	(1)	0.00139	(2)
	H	0.54443	(3)	0.54550	(1)	0.54545	(2)
	Δ'	0.00628	(3)	0.00526	(2)	0.00523	(1)
	$\mathcal{M}_2$	97.67010	(3)	99.79798	(1)	99.77778	(2)
	$\mathcal{M}_3$	1.38831	(2)	1.38830	(3)	1.38832	(1)
	$R1_R$	0.28450	(3)	0.35850	(1)	0.34050	(2)
FON2 $n = 20$	γ	0.01394	(2)	0.00910	(1)	0.00941	(2)
	H	0.47350	(3)	0.52711	(1)	0.52546	(2)
	Δ'	0.00285	(1)	0.00406	(3)	0.00395	(2)
	$\mathcal{M}_2$	99.77778	(3)	99.89899	(1)	99.86869	(2)
	$\mathcal{M}_3$	1.04803	(3)	1.33726	(1)	1.32772	(2)
	$R1_R$	0.00000	(3)	0.00450	(1)	0.00400	(2)
FON2 $n = 50$	γ	0.32914	(1)	0.52026	(3)	0.49959	(2)
	H	0.09924	(1)	0.01000	(3)	0.01520	(2)
	Δ'	0.01223	(3)	0.00000	(1)	0.00236	(2)
	$\mathcal{M}_2$	14.21429	(1)	0.00000	(2)	0.00000	(2)
	$\mathcal{M}_3$	0.31290	(1)	0.00000	(3)	0.02348	(2)
	$R1_R$	0.00000	(1)	0.00000	(1)	0.00000	(1)