

# Population Sizing for the Redundant Trivial Voting Mapping

Franz Rothlauf

Department of Information Systems 1  
University of Mannheim  
68131 Mannheim, Germany  
franz@rothlauf.com

**Abstract.** This paper investigates how the use of the trivial voting (TV) mapping influences the performance of genetic algorithms (GAs). The TV mapping is a redundant representation for binary phenotypes. A population sizing model is presented that quantitatively predicts the influence of the TV mapping and variants of this encoding on the performance of GAs. The results indicate that when using this encoding GA performance depends on the influence of the representation on the initial supply of building blocks. Therefore, GA performance remains unchanged if the TV mapping is uniformly redundant that means on average a phenotype is represented by the same number of genotypes. If the optimal solution is overrepresented, GA performance increases, whereas it decreases if the optimal solution is underrepresented. The results show that redundant representations like the TV mapping do not increase GA performance in general. Higher performance can only be achieved if there is specific knowledge about the structure of the optimal solution that can beneficially be used by the redundant representation.

## 1 Introduction

Over the last few years there has been an increased interest in using redundant representations for evolutionary algorithms (EAs) (Banzhaf, 1994; Dasgupta, 1995; Barnett, 1997; Shipman, 1999; Shackleton et al., 2000; Yu & Miller, 2001; Toussaint & Igel, 2002). It was recognized that redundant representations increase the evolvability of EAs (Shackleton et al., 2000; Ebner et al., 2001; Smith et al., 2001; Yu & Miller, 2001) and there is hope that such representations can increase the performance of evolutionary search. However, recent work (Knowles & Watson, 2002) indicated that redundant representations do not increase EA performance. More so, in most of the problems investigated (NK-landscapes, H-IFF, and MAX-SAT) redundant representations appeared to reduce EA performance.

The goal of this paper is to investigate how the redundancy of the trivial voting (TV) mapping, that is a redundant representation for binary phenotypes, influences the performance of genetic algorithms (GA). The developed population sizing model for the TV mapping is based on previous work (Rothlauf, 2002,

sect. 3.1) that shows that the population size  $N$  that is necessary to find the optimal solution is proportional to  $O(2^{k_r}/r)$ , where  $k_r$  is the order of redundancy and  $r$  is the number of genotypic building blocks (BBs) that represent the optimal phenotypic BB. The results show that uniformly redundant representations do not result in a better performance of GAs. Only if the good solutions are overrepresented by the TV mapping does GA performance increase. In contrast, if the good solutions are underrepresented GA performance decreases. Therefore, the redundant TV mapping can only be used beneficially for GA search if knowledge about the structure of the optimal solution exists.

The paper is structured as follows. In the following section we review some results of previous work (Rothlauf, 2002) that presented a model on how redundant representations influence GA performance. Section 3 describes the TV mapping. We discuss the properties of the representation and formulate a population sizing model. In Sect. 4 the paper presents experimental results for one-max and deceptive trap problems. The paper ends with concluding remarks.

## 2 Redundant Representations

The following section reviews the population sizing model presented in Rothlauf (2002, Sect. 3.1). In particular, Sect. 2.1 reviews characteristics of redundant representations and Sect. 2.2 presents the population sizing model for redundant representations.

### 2.1 Characteristics of Redundant Representations

In this subsection we introduce some characteristics of redundant representations based on Rothlauf (2002, Sect. 3.1).

In general, a representation  $f_g$  assigns genotypes  $\mathbf{x}_g \in \Phi_g$  to phenotypes  $\mathbf{x}_p \in \Phi_p$ .  $\Phi_g$ , respectively  $\Phi_p$  are the genotypic and phenotypic search spaces. A representation is redundant if the number of genotypes  $|\Phi_g|$  exceeds the number of phenotypes  $|\Phi_p|$ . The order of redundancy  $k_r$  is defined as  $\log(|\Phi_g|)/\log(|\Phi_p|)$  and measures the amount of redundant information in the encoding. When using binary genotypes and binary phenotypes, the order of redundancy can be calculated as

$$k_r = \frac{\log(2^{l_g})}{\log(2^{l_p})}, \quad (1)$$

where  $l_g$  is the length of the binary genotype and  $l_p$  is the length of the binary phenotype. When using a non-redundant representation, the number of genotypes equals the number of phenotypes and  $k_r = 1$ .

Furthermore, we have to describe how a representation over- or underrepresents specific phenotypes. Therefore, we introduce  $r$  as the number of genotypes that represent the one phenotype that has the highest fitness (we assume that there is only one global optimal solution). When using non-redundant representations, every phenotype is assigned to exactly one genotype and  $r = 1$ . However, in general,  $1 \leq r \leq |\Phi_g| - |\Phi_p| + 1$ .

The population sizing model presented in Rothlauf (2002) is valid for selectorecombinative GAs. Selectorecombinative GAs use crossover as the main search operator and mutation only serves as a background operator. When using selectorecombinative GAs we implicitly assume that there are building blocks (BBs) and that the GA process schemata. Consequently, we must define how  $k_r$  and  $r$  depend on the properties of the BBs.

In general, when looking at BBs of size  $k$  there are  $2^k$  different phenotypic BBs which are represented by  $2^{k k_r}$  different genotypic BBs. Therefore,

$$k_r = \frac{k_g}{k_p}, \quad (2)$$

where  $k_g$  denotes the genotypic size of a BB and  $k_p$  the size of the corresponding phenotypic BB. As before, a representation is redundant if  $k_r > 1$ . The size of the genotypic BBs is  $k_r$  times larger than the size of the phenotypic BB. Furthermore,  $r$  is defined as the number of genotypic BBs of length  $k k_r$  that represent the best phenotypic BB of size  $k$ . Therefore, in general,

$$r \in \{1, 2, \dots, 2^{k k_r} - 2^k + 1\}.$$

In contrast to  $k_r$  which is determined by the representation used,  $r$  depends not only on the used representation, but also on the specific problem that should be solved. Different instances of a problem result in different values of  $r$ . If we assume that  $k_r$  is an integer (each phenotypic allele is represented by  $k_r$  genotypic alleles) the possible values of the number of genotypic BBs that represent the optimal phenotypic BB can be calculated as

$$r = i^k, \text{ with } i \in [1, 2, \dots, 2^{k_r} - 1]. \quad (3)$$

A representation is uniformly redundant if all phenotypes are represented by the same number of different genotypes. Therefore, when using an uniformly redundant representation every phenotypic BB of size  $k = k_p$  is represented by

$$r = 2^{k(k_r-1)} \quad (4)$$

different genotypic BBs.

## 2.2 Influence of Redundant Representations on GA Performance

This subsection reviews the population sizing model presented in Rothlauf (2002). This population sizing model assumes that the redundancy of a representation influences the initial supply of BBs.

Earlier work (Harik, Cantú-Paz, Goldberg, & Miller, 1999) has presented a population sizing model for selectorecombinative GAs. The probability of failure  $\alpha$  of a GA can be calculated as

$$\alpha = 1 - \frac{1 - (q/p)^{x_0}}{1 - (q/p)^N}. \quad (5)$$

where  $x_0$  is the expected number of copies of the best BB in the randomly initialized population,  $q = 1 - p$ , and  $p$  is the probability of making the right choice between a single sample of each BB

$$p = \mathbb{N} \left( \frac{d}{\sqrt{2m'}\sigma_{BB}} \right). \quad (6)$$

$d$  is the signal difference between the best BB and its strongest competitor,  $m' = m - 1$  with  $m$  is the number of BBs in the problem,  $\sigma_{BB}^2$  is the variance of a BB, and  $q = 1 - p$  is the probability of making the wrong decision between two competing BBs. Therefore, we get for the population size that is necessary to solve a problem with probability  $1 - \alpha$ :

$$N = \frac{\log \left( 1 - \left( \frac{1-(q/p)^{x_0}}{1-\alpha} \right) \right)}{\log(q/p)}.$$

Rothlauf (2002) extended the work presented in Harik, Cantú-Paz, Goldberg, and Miller (1999) and assumed that redundant representations change the initial supply of BBs:

$$x_0 = N \frac{r}{2^{kk_r}}. \quad (7)$$

After some approximations (compare Rothlauf (2002, Sect. 3.1)) we get for the population size  $N$ :

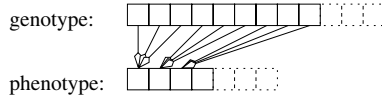
$$N \approx -\frac{2^{k_r k - 1}}{r} \ln(\alpha) \frac{\sigma_{BB} \sqrt{\pi m'}}{d}. \quad (8)$$

The population size  $N$  goes with  $O\left(\frac{2^{k_r}}{r}\right)$ . With increasing  $r$  the number of individuals that are necessary to solve a problem decreases. Using a uniformly redundant representation, where  $r = 2^{k(k_r-1)}$ , does not change the population size  $N$  in comparison to non-redundant representations.

### 3 The Trivial Voting Mapping

In the following subsection we give a short introduction into the trivial voting (TV) mapping.

When using the TV mapping, a set of mostly consecutive, genotypic alleles is relevant for the value of one allele in the phenotype. Each allele in the genotype can only influence the value of one allele in the phenotype. The value of the phenotypic allele is determined by the majority of the values in the genotypic alleles. In general, the different sets of alleles in the genotype defining one phenotypic allele have the same size. Furthermore, all genotypes that represent the same phenotype are very similar to one another. A mutation in a genotype results either in the same corresponding phenotype, or in one of its neighbors.



**Fig. 1.** The trivial voting mapping

This is an important aspect of the TV mapping as the population sizing model presented in Rothlauf (2002) is only valid for such representations.

The TV mapping can easily be characterized using the representation parameters defined in Sect. 2.1. The order of redundancy  $k_r$  is simply the number of genotypic alleles that determine the value of one phenotypic allele. As the representation is uniformly redundant,  $r = 2^{k(k_r-1)}$  (Eq. (4)). Figure 1 gives an example for the TV mapping.

Shackleton, Shipman, and Ebner (2000) applied the TV mapping to binary strings illustrating that the use of redundant representations increases the evolvability of EAs. When used for binary strings, binary genotypes  $x^g \in \mathbb{B}^{l_g}$  are assigned to binary phenotypes  $x^p \in \mathbb{B}^{l_p}$ . The length of a genotype is larger than the length of a phenotype,  $l_g > l_p$ . The value of one phenotypic bit is determined by the majority of the values in the corresponding genotypic bits (majority vote). However, if  $k_r$  is even then the number of ones could equal the number of zeros. Therefore, half the cases that result in a tie should encode a one in the corresponding phenotypic allele, and half the cases should represent a zero. For example, for  $k_r = 4$  the genotypic BBs 1100, 1010, and 1001 represent a 1 and the phenotypic BBs 0011, 0101, 0110 represent a zero.

Because the majority of the votes determines the values of the corresponding phenotypic allele, the TV mapping is a uniformly redundant representation. Each phenotypic BB is represented by the same number of genotypic BBs which is  $2^{k(k_r-1)}$ , where  $k$  is the size of the phenotypic BB.

As we are not only interested in uniformly redundant representations, but also want to know how non-uniformly redundant representations influence GA performance, we extend the TV mapping to allow the encoding to overrepresent some individuals. Therefore, we want to assume that if the number of ones in the  $k_r$  genotypic alleles  $x_{k_r i+j}^g$ , where  $j \in \{0, \dots, k_r - 1\}$ , is larger or equal than a constant  $u$  then the value of the phenotypic allele  $x_i^p$  is set to one ( $i \in \{0, \dots, l_p - 1\}$ ). Similarly, the phenotypic allele  $x_i^p$  is set to zero if less than  $u$  of the corresponding genotypic alleles are set to one.  $x_i^g$  respectively  $x_i^p$  denotes the  $i$ th allele of the genotype respectively phenotype. Therefore,

$$x_i^p = \begin{cases} 0 & \text{if } \sum_{j=0}^{k_r-1} x_{k_r i+j}^g < u \\ 1 & \text{if } \sum_{j=0}^{k_r-1} x_{k_r i+j}^g \geq u, \end{cases}$$

where  $u \in \{1, \dots, k_r\}$ .  $u$  can be interpreted as the number of genotypic alleles that must be set to one to encode a one in the corresponding phenotypic allele. We denote this representation the *extended trivial voting (eTV) mapping*. For  $u = (k_r + 1)/2$  ( $k_r$  must be odd) we get the original TV mapping. Extending

the TV mapping in the proposed way allows us to investigate how non-uniform redundancy influences the performance of GAs.

When using the eTV mapping, the number  $r$  of genotypic BBs that can represent the optimal phenotypic BB depends on the number of ones in the genotypic alleles that determine the value of the corresponding phenotypic allele. Considering Eq. (3) we get

$$r = \left( \sum_{j=u}^{k_r} \binom{k_r}{j} \right)^k, \quad (9)$$

where  $u \in \{1, \dots, k_r\}$ .  $k$  denotes the size of the phenotypic BB. We want to give a short illustration. We use a redundant representation with  $k_r = 3$ ,  $k = 1$ , and the optimal BB is  $x_i^p = 1$  (compare Fig. 1). Because  $u \in \{1, \dots, k_r\}$  there are three different values possible for  $r$ . For  $u = 1$  the phenotypic allele  $x_i^p$  is set to one if at least one of the three corresponding genotypic alleles  $x_{ik_r}^g$ ,  $x_{ik_r+1}^g$ , or  $x_{ik_r+2}^g$  is set to one. Therefore, a one in the phenotype is represented by  $r = \sum_{j=1}^3 \binom{k_r}{j} = 7$  different genotypic BBs (111, 110, 101, 011, 100, 010, and 001). For  $u = 2$ , the optimal phenotypic BB  $x_i^p = 1$  is represented by  $r = \sum_{j=2}^3 \binom{k_r}{j} = 4$  different genotypic BBs (111, 110, 101, and 011) and the representation is uniformly redundant. For  $u = 2$  we get the original TV mapping. For  $u = 3$ , the optimal phenotypic BB is represented only by one genotypic BB (111).

Finally, we can formulate the population sizing model for the eTV mapping combining Eq. (5), 7, and 9. The probability of failure can be calculated as

$$\alpha = 1 - \frac{1 - (q/p)^{\left(\frac{N}{2^{k_r}} \left(\sum_{j=u}^{k_r} \binom{k_r}{j}\right)^k\right)}}{1 - (q/p)^N}, \quad (10)$$

where  $k_r$  is the number of genotypic bit that represent one phenotypic bit,  $k$  is the size of the phenotypic BBs,  $q = 1 - p$ , and  $p$  is the probability of making the right choice between a single sample of each BB (Eq. (6)).

## 4 Experiments and Empirical Results

We present empirical results when using the TV and eTV mapping for the one-max problem and the concatenated deceptive trap problem.

### 4.1 One-Max Problem

The first test example for our empirical investigation is the one-max problem. This problem is very easy to solve for GAs as the fitness of an individual is simply the number of ones in the binary phenotype. To ensure that recombination results in a proper mixing of the BBs, we use uniform crossover for all experiments with the one-max problem. As we focus on selectorecombinative GAs we use no mutation. Furthermore, in all runs we use tournament selection without

**Table 1.** The trivial voting mapping for  $k_r = 3$ 

$x_i^p$	$x_{3i}^g x_{3i+1}^g x_{3i+2}^g$ (with $k_r = 3$ )			
	extended TV			original TV
	$u = 1$	$u = 2$	$u = 3$	$u = 2$
	$r = 7$	$r = 4$	$r = 1$	$r = 4$
0	000	001, 010, 100, 000	111, 110, 101, 011, 001, 010, 100	001, 010, 100, 000
1	111, 110, 101, 011, 001, 010, 100	111, 110, 101, 011	000	111, 110, 101, 011

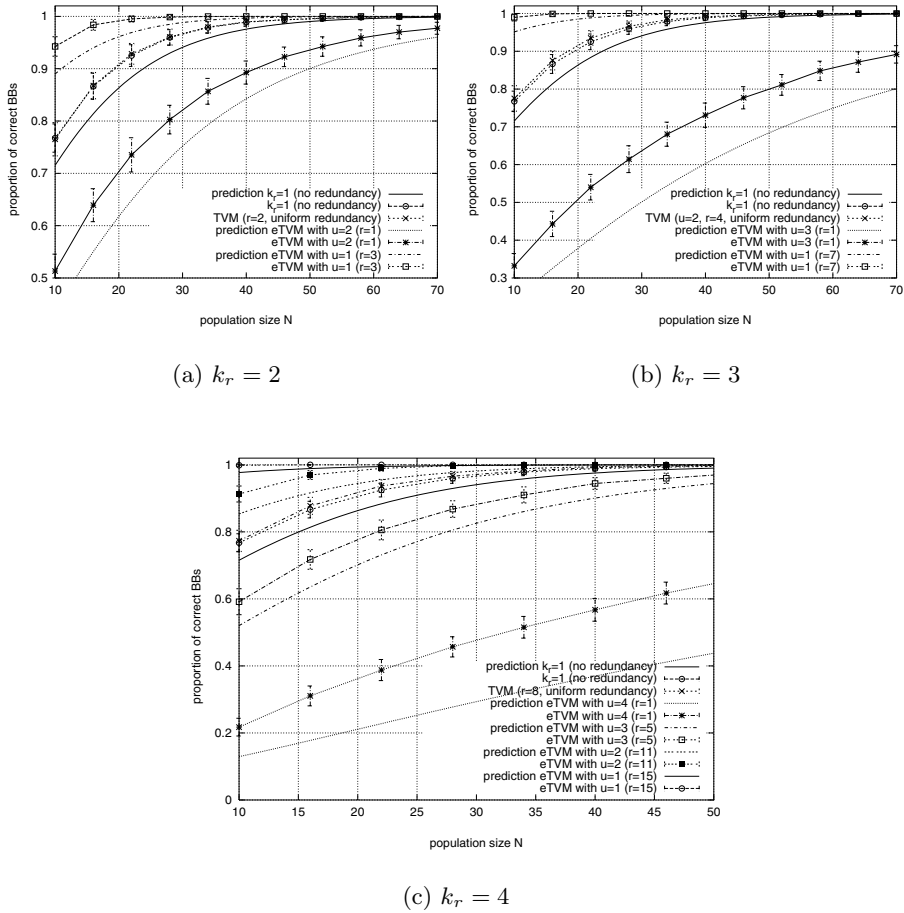
replacement and a tournament size of 2. For the one-max function the signal difference  $d$  equals 1, the size  $k$  of the building blocks is 1, and the variance of a building block  $\sigma_{BB}^2 = 0.25$ .

When using the binary TV mapping for the one-max problem each bit of a phenotype  $x^p$  is represented by  $k_r$  bits of the genotype  $x^g$ . The string length of a genotype  $x^g$  is  $l_g = k_r * l_p$  and the size of the genotypic search space is  $|\Phi_g| = 2^{k_r l_p}$ . Table 1 illustrates for  $k_r = 3$  the three possibilities of assigning genotypic BBs {000, 001, 010, 100, 110, 101, 011, 111} to one of the phenotypic BBs {0, 1} when using the extended TV mapping described in the previous paragraphs. With denoting  $x_i^p$  the value of the  $i$ th bit in the phenotype, the  $3i$ th,  $(3i + 1)$ th, and  $(3i + 2)$ th bit of a genotype determine  $x_i^p$ . Because the size of the BBs  $k = 1$ , the number of genotypic BBs that represent the optimal phenotypic BB is either  $r = 1$ ,  $r = 4$ , or  $r = 7$  (compare Eq. (9)).

In Fig. 2(a) ( $k_r = 2$ ), 2(b) ( $k_r = 3$ ), and 2(c) ( $k_r = 4$ ) the proportion of correct BBs at the end of a run for a 150 bit one-max problem using the TV and eTV mapping is shown. For this problem  $2^{150}$  different phenotypes are represented by either  $2^{300}$  ( $k_r = 2$ ),  $2^{450}$  ( $k_r = 3$ ), or  $2^{600}$  ( $k_r = 4$ ) different genotypes. If we use the eTV mapping (indicated in the plots as eTVM) we can set  $u$  either to 1 or 2 ( $k_r = 2$ ) or to 1, 2, or 3 ( $k_r = 3$ ), or to 1, 2, 3, or 4 ( $k_r = 4$ ). The corresponding values for  $r$  which can be calculated according to Eq. (9) as well as  $x_0/N$  are shown in Table 2.  $x_0$  is the expected number of copies of the best BB in the initial population and  $N$  is the population size. Furthermore, the figures show the results when using the original, uniformly redundant TV mapping, and when using the non-redundant representation with  $k_r = 1$ .

The lines without line points show the theoretical predictions from Eq. (10), and the lines with line points show the empirical results which are averaged over 250 runs. The error bars indicate the standard deviation.

The results show that for the uniformly redundant TV mapping,  $r = 2$  ( $k_r = 2$ ),  $r = 4$  ( $k_r = 3$ ), or  $r = 8$  ( $k_r = 4$ ) we get the same performance as for using the non-redundant representation ( $k_r = 1$ ). As in the original model proposed by Harik, Cantú-Paz, Goldberg, and Miller (1999) the theoretical model slightly underestimates GA performance. As predicted by the model described in Sect. 2.2, GA performance does not change when using a uniformly redundant representation. Furthermore, we can see that if the optimal BB is underrepresented GA performance decreases. Equation (10) gives us a good prediction for



**Fig. 2.** Experimental and theoretical results of the proportion of correct BBs on a 150-bit one-max problem using the trivial voting mapping for  $k_r = 2$  (a),  $k_r = 3$  (b), and  $k_r = 4$  (c). The lines without line points show the theoretical predictions. When using non-uniformly redundant representations, GA performance is changed with respect to the overrepresentation or underrepresentation of the high-quality BBs.

the expected solution quality if we consider that the non-uniform redundancy of the representation changes the initial BB supply according to Eq. (7). If the optimal solution is overrepresented GA performance increases. Again the theoretical models give a good prediction for the expected proportion of correct BBs.

Summarizing the results, we can see that using the uniformly redundant TV mapping does not change GA performance in comparison to using the non-redundant representation. Only if we overrepresent the optimal phenotypic BB, does GA performance increase; likewise, if we underrepresent the optimal BB,



**Table 2.** Properties of the different TV mappings for the one-max problem ( $k = 1$ )

		extended TV mapping				original TV mapping
		$u = 1$	$u = 2$	$u = 3$	$u = 4$	
$k_r = 2$	$r$	3	1	-	-	2
	$x_0/N$	3/4	1/4	-	-	$2/4 = 1/2$
$k_r = 3$	$r$	7	4	1	-	4
	$x_0/N$	7/8	$4/8 = 1/2$	1/8	-	$4/8 = 1/2$
$k_r = 4$	$r$	15	11	5	1	8
	$x_0/N$	15/16	11/16	5/16	1/16	$8/16 = 1/2$

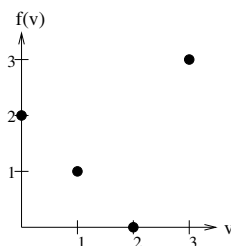
GA performance drops. The derived model is able to make accurate predictions for the expected solution quality.

## 4.2 Concatenated Deceptive Trap Problem

Our second test example uses deceptive trap functions.

Traps were first used by Ackley (1987) and investigations into the deceptive character of these functions were provided by Deb and Goldberg (1993). Figure 3 depicts a 3-bit deceptive trap problem where the size of a BB is  $k = 3$ . The fitness value of a phenotype  $x^p$  depends on the number of ones  $v$  in the string of length  $l$ . The best BB is a string of  $l$  ones which has fitness  $l$ . Standard EAs are misled to the deceptive attractor which has fitness  $l - 1$ . For the 3-bit deceptive trap the signal difference  $d$  is 1, and the fitness variance equals  $\sigma_{BB}^2 = 0.75$ . We construct a test problem for our investigation by concatenating  $m = 10$  of the 3-bit traps so we get a 30-bit problem. The fitness of an individual  $x$  is calculated as  $f(x) = \sum_{i=0}^{m-1} f_i(v)$ , where  $f_i(v)$  is the fitness of the  $i$ th 3-bit trap function from Fig. 3. Although this function is difficult for GAs it can be solved with proper population size  $N$ .

For deceptive traps of size  $k = 3$  we can calculate the number  $r$  of genotypic BBs that represent the optimal genotypic BBs according to Eq. (9). Table 3 summarizes for the modified TV mapping how  $r$  and  $x_0/N$  depends on  $u$ , that describes how many of the genotypic alleles must be set to 1 to encode a 1 in the phenotype.  $x_0$  is the expected number of copies of the best BB in the initial


**Fig. 3.** A 3-bit deceptive trap problem

**Table 3.** Properties of the different TV mappings for the deceptive trap of size  $k = 3$

		extended TV mapping				original TV mapping
		$u = 1$	$u = 2$	$u = 3$	$u = 4$	
$k_r = 2$	$r$	$3^3 = 27$	$1^3 = 1$	-	-	$2^3 = 8$
	$x_0/N$	$27/64$	$1/64$	-	-	$8/64 = 1/8$
$k_r = 3$	$r$	$7^3 = 343$	$4^3 = 64$	$1^3 = 1$	-	$4^3 = 64$
	$x_0/N$	$343/512$	$64/512 = 1/8$	$1/512$	-	$64/512 = 1/8$
$k_r = 4$	$r$	$15^3 = 3375$	$11^3 = 1331$	$5^3 = 125$	$1^3 = 1$	$8^3 = 2048$
	$x_0/N$	$3375/4096$	$1331/4096$	$125/4096$	$1/4096$	$512/4096 = 1/8$

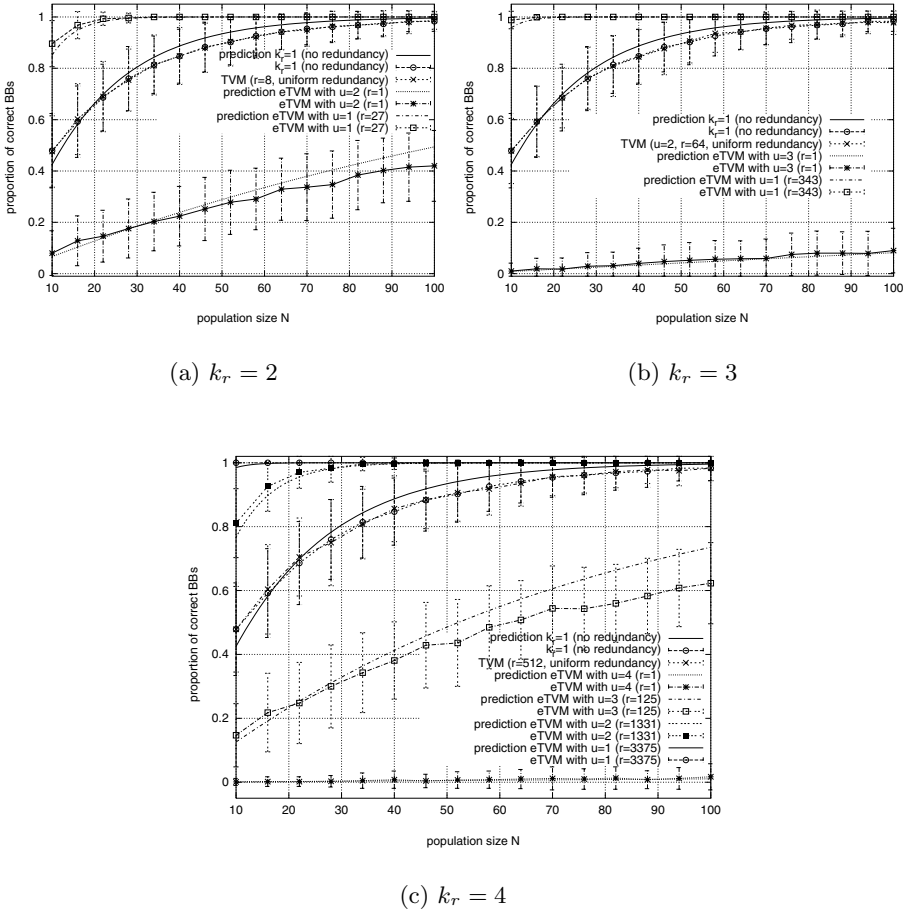
population and  $N$  is the population size. Furthermore, we list the properties of the original uniformly redundant TV mapping.

By analogy to the previous paragraphs, in Fig. 4(a) ( $k_r = 2$ ), Fig. 4(b) ( $k_r = 3$ ), and Fig. 4(c) ( $k_r = 4$ ) we show the proportion of correct BBs at the end of a run over different population sizes for ten concatenated 3-bit deceptive trap problems. In this problem,  $2^{30}$  different phenotypes are represented by either  $2^{60}$  ( $k_r = 2$ ),  $2^{90}$  ( $k_r = 3$ ), or  $2^{120}$  ( $k_r = 4$ ) different genotypes. As before, we use tournament selection of size two without replacement. In contrast to the one-max problem, two-point crossover was chosen for recombination. Uniform crossover would result in an improper mixing of the BBs because the genotypic BBs are either of length  $l_g = k_r l_p = 6$  ( $k_r = 2$ ), of length  $l_g = 9$  ( $k_r = 3$ ), or of length  $l_g = 12$  ( $k_r = 4$ ). Again, the lines without line points show the predictions of the proposed model for different  $r$ . Furthermore, empirical results that are averaged over 250 runs, are shown for various values of  $r$ . The results show that for the uniformly redundant TV mapping we get the same performance as when using the non-redundant representation ( $k_r = 1$ ). As in the experiments for the one-max problem the proposed model predicts the experimental results well if the eTV mapping is used and some BBs are underrepresented or overrepresented.

The presented results show that the influence of the redundant TV and eTV mapping on the performance of GAs can be explained well by the influence of the representation on the initial supply of high-quality BBs. If the eTV mapping favors high-quality BBs then the performance of GAs is increased. If good BBs are underrepresented the performance is reduced. If the representation is uniformly redundant, GAs show the same performance as when using the non-redundant encoding.

## 5 Summary and Conclusions

This paper presented a population sizing model for the trivial voting mapping and variants of this representation. The trivial voting mapping is a redundant representation for binary phenotypes. The presented population sizing model is based on previous work (Rothlauf, 2002) and assumes that redundant representations affect the initial supply of building blocks. The model was adapted to



**Fig. 4.** Experimental and theoretical results of the proportion of correct BBs for ten concatenated 3-bit deceptive traps. We show results for different variants of the TV mapping and  $k_r = 2$  (a),  $k_r = 3$  (b), and  $k_r = 4$  (c). The lines without line points show the theoretical predictions. As predicted, GA performance sharply decreases if the eTV mapping underrepresents the optimal BB.

the TV mapping and used for predicting the performance of genetic algorithms for one-max problems and deceptive trap problems.

The results show that the proposed population sizing model allows an accurate prediction of the influence of the redundant TV representation on GA performance. GA performance remains unchanged if the TV mapping is uniformly redundant that means each phenotype is represented on average by the same number of genotypes. Furthermore, the proposed population sizing model is able to give accurate quantitative predictions on the expected solution quality if variants of the TV mapping either overrepresent or underrepresent the optimal

solution. If the optimal BB is overrepresented GA performance increases, if it is underrepresented it decreases. The results reveal that in general the redundant TV mapping and variants of it do not increase GA performance. GA performance can only be increased if there is specific knowledge about the structure of the optimal solution and if the representation overrepresents the good solutions.

Previous work (for example Shipman et al. (2000), or Ebner et al. (2001)) noticed that redundant representations increase the evolvability of EAs and assumed that this may lead to increased EA performance. However, the results we present indicate that redundant representations like the TV mapping do not increase EA performance. The influence of redundant representation on the initial supply seems to be more relevant for EA performance than an increase of evolvability. Therefore, we encourage researchers to focus on the question of how redundant representations influence EA performance and to neglect their influence on evolvability. The influence on the evolvability of EAs might be an interesting partial aspect but more relevant is the question of whether we are able to construct EAs that allow us to solve relevant problems efficiently, fast, and reliably.

## References

- Ackley, D. H. (1987). *A connectionist machine for genetic hill climbing*. Boston: Kluwer Academic.
- Banzhaf, W. (1994). Genotype-phenotype-mapping and neutral variation – A case study in genetic programming. In Davidor, Y., Schwefel, H.-P., & Männer, R. (Eds.), *Parallel Problem Solving from Nature- PPSN III* (pp. 322–332). Berlin: Springer.
- Barnett, L. (1997). *Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality*. Master's thesis, School of Cognitive Sciences, University of East Sussex, Brighton.
- Dasgupta, D. (1995). Incorporating redundancy and gene activation mechanisms in genetic search for adapting to non-stationary environments. In Chambers, L. (Ed.), *Practical Handbook of Genetic Algorithms* (Chapter 13, pp. 303–316). CRC Press.
- Deb, K., & Goldberg, D.E. (1993). Analyzing deception in trap functions. In Whitley, L.D. (Ed.), *Foundations of Genetic Algorithms 2* (pp. 93–108). San Mateo, CA: Morgan Kaufmann.
- Ebner, M., Langguth, P., Albert, J., Shackleton, M., & Shipman, R. (2001, 27–30 May). On neutral networks and evolvability. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001* (pp. 1–8). COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea: IEEE Press.
- Harik, G., Cantú-Paz, E., Goldberg, D.E., & Miller, B.L. (1999). The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation*, 7(3), 231–253.
- Knowles, J.D., & Watson, R.A. (2002). On the utility of redundant encodings in mutation-based evolutionary search. In Merelo, J.J., Adamidis, P., Beyer, H.-G., Fernandez-Villacanas, J.-L., & Schwefel, H.-P. (Eds.), *Parallel Problem Solving from Nature, PPSN VII* (pp. 88–98). Berlin: Springer-Verlag.
- Rothlauf, F. (2002). *Representations for genetic and evolutionary algorithms*. Studies on Soft Computing and Fuzziness. Berlin: Springer Verlag.

- Shackleton, M., Shipman, R., & Ebner, M. (2000, 6-9 July). An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00* (pp. 493–500). La Jolla Marriott Hotel La Jolla, California, USA: IEEE Press.
- Shipman, R. (1999). Genetic redundancy: Desirable or problematic for evolutionary adaptation? In *Proceedings of the 4th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA)* (pp. 1–11). Springer Verlag.
- Shipman, R., Shackleton, M., & Harvey, L. (2000). The use of neutral genotype-phenotype mappings for improved evolutionary search. *British Telecom Technology Journal*, 18(4), 103–111.
- Smith, T., Husbands, P., & M., O. (2001). Neutral networks and evolvability with complex genotype-phenotype mapping. In *Proceedings of the European Conference on Artificial Life: ECAL2001* (pp. 272–281).
- Toussaint, M., & Igel, C. (2002). Neutrality: A necessity for self-adaptation. In Fogel, D.B., El-Sharkawi, M.A., Yao, X., Greenwood, G., Iba, H., Marrow, P., & Shackleton, M. (Eds.), *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002* (pp. 1354–1359). IEEE Press.
- Yu, T., & Miller, J. (2001). Neutrality and evolvability of Boolean function landscapes. In *Proceedings of the 4th European Conference on Genetic Programming (EuroGP)*, Volume LNCS 2038 (pp. 204–217). Springer.