最適化アルゴリズムを用いたQRコード装飾

小野 智司† 森永 健介† 中山 茂†

† 鹿児島大学工学部情報工学科 〒890–0065 鹿児島市郡元1–21–40
E-mail: †{ono,sc104069,shignaka}@ics.kagoshima-u.ac.jp

あらまし 本稿では，QRコード上にイラスト，写真，絵，記号などを配置することで，可読性を損ねることなくQRコードの装飾を行う方式を提案する．提案する方式は，イラスト等を配置する位置，サイズ，傾きを決定するタスクを最適化問題として定式化し，実数遺伝子表現の遺伝的アルゴリズムを用いて解を発見する．また，複数のソフトウェアデコーダを利用することで，装飾されたQRコードの可読性をより正確に検証する．通常のQRコードに加え，提案する方式は，QRコード上でイラストが移動や変形を行う動画QRコードの生成に利用が可能である．動画QRコードの生成もまた最適化問題として定式化され，あらかじめデザイナによって指定された移動軌跡に近く，可読性を損ねない解を山登り法によって発見する．

キーワード QRコード，動画QRコード，バーコード装飾，最適化，遺伝的アルゴリズム

QR Code Decoration Using Optimization Algorithms

Satoshi ONO†, Kensuke MORINAGA†, and Shigeru NAKAYAMA†

† Department of Information and Computer Scince, Faculty of Engineering, Kagoshima University
Korimoto 1–21–40, Kagoshima, 105–0123 Japan
E-mail: †{ono,sc104069,shignaka}@ics.kagoshima-u.ac.jp

Abstract This paper proposes a method for decorating 2-dimensional barcode with some illustrations inside the code without detracting machine-readability. The proposed method formulates the task of finding appropriate positions, scales, and angles of illustrations, and solves the task by using real-coded genetic algorithm. The proposed method also uses multiple barcode decoder with the aim of improving decode feasibility of the decorated barcode. In addition to still QR code images with illustrations, the proposed method can generate animated QR codes with illustrations, which attracts the attention of the people more than the still QR codes. The animated QR code generation is formulated and solved by hill-climbing algorithm so that illustration positions are adjusted according to positions and movement specified by user.

Key words QR code, animated QR code, barcode decoration, optimization, genetic algorithm

1. Introduction

QR code (Quick Response code)(注1), a kind of two-dimensional barcode investigated by Denso Wave as shown in Fig. 1, is currently used in Japan as a so-called shortcut to get a URL, e-mail address, phone number and so on. Most Japanese mobile phones have cameras and QR code scanner. By holding a mobile phone over QR codes printed on papers, billboards, or television screens, users can get decoded information and browse Web sites, or send e-mails without typing URLs or e-mail addresses on their mobile phones. This act of linking from physical world object is known as a hardlink or physical world hyperlinks [1].

Although QR code can involve various information such as URL, e-mail address, short sound, and so on, users cannot know what kind of information is implanted in it (from QR code itself). In addition, QR code is unattractive and occupies an area in limited, worthy space of papers, billboards, or other media.

QR code has an error correction function which can supplement at most 30% data loss. An illustration therefore can be put inside a QR code. QR code decorated with an illustration is more attractive than a general, plain QR code. In addition, it can let humans know what kind of information

(注1)：QR code is trademarked by Denso Wave, inc.

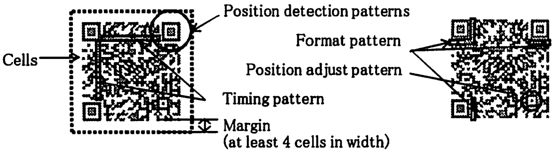


Fig. 1 QR code.

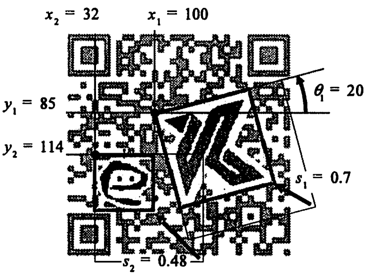


Fig. 2 Chromosome representation.

is implanted in the code.

However, putting an illustration into QR code often damages implanted information. It is difficult to find an appropriate place in order to let the error correction work well and keep machine-readability. Larger illustrations or more images make this problem more difficult.

This paper proposes a method for generating two-dimensional barcode incorporated with some illustrations inside the code without affecting machine-readability and stored information. The method formulates the task of finding appropriate position, scale, and angle of an illustration, photo, logo or other image item put into QR code as an optimization problem. By using Real-Coded Genetic Algorithms (RCGA) [], the method finds positions in which a given image item can be merged without damaging machine-readability and stored information.

A simple animated QR code generation method is also proposed in this paper. An animated QR code involves illustrations moving and/or transforming inside the code. The proposed method for generating an animated QR code is an improved one from previous work [2], [3] in terms of formulation, optimization operator, and quality evaluation of generated QR code. The proposed method formulates an animation QR code generation problem as just one optimization problem instead of a set of still QR code decoration problems. The proposed method also uses optimization operators designed for this problem, and quality evaluation functions to maintain natural, smooth illustration movement.

Experiments have shown that our method can generate still QR codes containing at most 10 illustrations and animated QR codes involving at most 8 illustrations and at most 32 frames with keeping natural illustration movements and machine readability.

2. QR-code decoration method

2.1 Basic idea

The principles of the QR code decoration method with illustrations [2], [4] are as follows:

1) Formulating the task of decorating QR codes by placing illustrations into the code as an optimization problem.

Deciding the positions of illustrations requires iteration of trial production of the barcode with the illustrations and test for decoding practicability. This is because we cannot know the appropriate places where the illustrations should be placed in the barcode in advance. We therefore formulate this task as an optimization problem. The objective function is composed of feasibility of decoding evaluated by QR code decoders, and adequacy of positions and other parameters of the incorporated image based on human-readability.

2) Using real-coded genetic algorithm to find an appropriate image position.

Genetic algorithm (GA) [5] is a well-known meta-heuristics [6] for optimization inspired by evolutionary biology. GA can produce better solutions by recombining good "building blocks" [5]. Traditionally, genes are represented in binary, but real values and other encodings have also become possible in recent years. Because positions and other parameters providing a barcode with illustrations should be real numbers instead of sequences of binary variables, the method utilizes real-coded GA [7–9].

3) Evaluating solution candidates by multiple QR code decoders.

Most recent mobile phones on the market in Japan involve a QR code decoder software developed by mobile phone manufacturers or other subcontract software developers. Although it is hard to prepare and use the same QR code decoders in our method as mobile phones use, we try to ensure the robustness of generated QR codes by using multiple decoders. The more QR code decoders the method uses, the more mobile phones can read the generated codes.

2.2 Chromosome representation

QR code design problem involves continuous variables of which the number is $4 \times N_I + 1$, where N_I indicates the number of illustrations: x- and y-axis positions x_i, y_i, an inclination θ_i, and a scale s_i for each illustration i, and QR code direction θ_{QR}. A chromosome, i.e. genotypic representation, consists of the variables coded as integers and real values. Positions and an inclination are coded as integers, scales as real values, and QR code direction as four discrete values: 0, 90, 180, or 270 degrees. The phenotypic represen-

```
begin
  make individuals C_i by random
  repeat
    generate decorated QR code images (PTYPE) from C_i
    apply QR code decoders to the decorated QR code images and calculate P(C_i)
    calculate Q(C_i) and fitness F(C_i)
    select elites to preserve
    crossover by BLX-α
    mutation
  until the stop condition is satisfied
end
```

Fig. 3 Algorithm for decorating still QR code by GA.

tation is formed by placing illustrations along the variables, as shown in Fig. 2.

2.3 Process flow

The inputs are a general QR code and illustrations. The QR code can be easily generated by using open software or services on various web sites.

The QR code decoration method iterates generation of decorated QR codes and evaluation of the QR codes by using software decoders. Real-coded GA [7–9] is adopted to optimize the illustration placement, as shown in Fig. 3. The method utilizes blend crossover (BLX-α), a simple crossover operator for real-coded GA proposed by Eshelman [9]. The algorithm shown in Fig. 3 stops when fitness of the best solution reaches 1.0, or the number of generations reaches the pre-defined limit.

2.4 Fitness calculation with QR code decoders

The QR code decoration method uses more than one QR code decoder to evaluate individuals, which are QR codes decorated with illustrations inside, for the sake of enhancing robustness of the decoding feasibility of the decorated code. Fitness $F(C)$ of an individual C is a product of success rate in decoding $P(C)$ and appropriateness of illustrations' positions $Q(C)$:

$$F(C) = P(C)^{w^{(p)}} \times Q(C)^{w^{(q)}} \quad (1)$$

$P(C)$ indicates how C can be decoded properly and is calculated as following equation:

$$P(C) = \frac{\sum_k p_k(C) \times w_k^{(p)}}{\sum_k w_k^{(p)}} \quad (2)$$

$p_k(C)$ is calculated from decode result of a decoder k; $p_k(C) = 1$ if a decoder succeeds in extracting information. In the case that a decoder k does not produce any data except the fact that the decoder failed to decode, $p_k(C) = 0$ when the decoder failed. In the case that a decoder k can present decode error amount, $p_k(C)$ is calculated by subtracting error rate from 1.0. $w_k^{(p)}$ is a weight parameter.

The method uses Psytec QR Code Decode Library (注2) (Psytec decoder) and Open Source QR Code Decode Library (注3) (open source decoder). Psytec library outputs information embedded in a given QR code only when the code is decodable, and does not produce any information when the code cannot be decoded. Open Source decode library outputs information embedded in the code even when the library cannot decode the code entirely. Thereby the decode error rate can be calculated by comparing the library's output and embedded information.

$Q(C)$ indicates how C places illustrations appropriately in its QR code, and calculated as follows:

$$Q(C) = \prod_l q_l(C)^{w_l^{(q)}} \quad (3)$$

The previous method uses following four functions $q_1(C)$ through $q_4(C)$ to calculate $Q(C)$: how illustrations are overlapped each other ($q_1(C)$), how illustrations are placed appropriately inside of QR code ($q_2(C)$), how large illustrations are magnified ($q_3(C)$), and how illustrations are magnified or reduced with the same scale between illustrations($q_4(C)$).

Additional three functions $q_5(C)$, $q_6(C)$, and $q_7(C)$ are newly proposed in this paper in order to enhance macine-readability of decorated QR codes. $q_5(C)$, $q_6(C)$, and $q_7(C)$ indicate how the position detection patterns, the timing patterns, and the format patterns are covered by illustrations.

3. Extension of the proposed method for generating animated QR codes

3.1 Basic idea

The proposed method in this paper generates animated QR codes, in which illustrations move and/or transform inside the QR code. The differences between generating still and animated QR codes are as follows:

1) Letting users specify the positions of illustrations in advance.

The proposed method requires rough positions of illustrations for each frame. In case that illustrations placed at the specified positions damages implanted information and makes mobile phones hard to decode the code, the proposed method tries to find appropriate positions which avoids damaging the implanted information and as near as the specified position.

2) Generating animated QR code by decorating some still QR codes with illustrations and playing

(注2) : http://www.psytec.co.jp/product/03/
(注3) : http://sourceforge.jp/projects/qrcode

```
1: begin
2:   initialize solution candidate C
3:   evaluate C
4:   repeat
5:     repeat
6:       select an illustration i at random
7:       select a frame f at random
8:     until i is permitted to adjust positions and s̄_i^(f) ≠ 0
9:     for n = 1 to N_n do
10:      select an operator at random
11:      make C'_n by the selected operator
12:    end for
13:    generate a QR code image of C'_n involving illustrations
       for each frame
14:    apply QR code decoders to frame images in C'_n and
       calculate P(C'_n)
15:    calculate Q(C'_n) and F(C'_n)
16:    select C'_h in C'_n whose F(C'_n) is the highest
17:    if F(C'_h) ≥ F(C) then
18:      C ← C'_h
19:    end if
20:  until the stop condition is satisfied
21: end
```

Fig. 4 Algorithm for decorating and generating animated QR code by HC.

them as movie frames.

The previous method [3] formulates an animation QR code generation problem as a set of optimization problems which are still QR code decoration problems. The proposed method in this paper formulates the animation QR code generation problem as just an optimization problem instead of the set of still QR code decoration problems; all frames are generated simultaneously in order to maintain the continuity of illustration movements.

3) Adopting Hill-Climbing (HC) algorithm to optimize illustration positions. Simple HC algorithm is sufficient to find appropriate illustration positions because desirable positions are given and local search around the given positions is particularly effective.

The proposed method uses some specific operators designed for this problem in addition to previous work [3]. The added operator allows revising illustration positions with minimal effects on illustration trajectory.

4) Quality evaluation:

The proposed method uses additional quality evaluation functions to previous work [3], allowing to maintain the given illustration trajectory.

3.2 Process flow

Fig. 4 shows the process flow of animated QR code generation by the proposed method. The proposed method generates all frames simultaneously whereas the previous method individually generates a still QR code with illustrations for each frame. HC is a single point search algorithm and has a solution candidate C. In each iteration of HC, neighbors C'_n $(n = 1, \ldots, N_n)$ of C are generated and the best solution candidate C'_h in C'_n is chosen. If $F(C'_h)$ is equal or higher than $F(C)$, substitute C'_h for C. The process stops when the iteration number reaches the limit.

The neighbors C'_n is made in line 5 through 12. First, an illustration i and a frame f is selected at random, and then an operator is selected. If a user specifies that illustration i should not move, or that the size of i at frame f should be zero, then i and f must be selected again. The proposed method uses the following four operators:

a) Translocation: This operator changes i's positions with the same amount determined randomly at all frames to keep i's movement parallel to the specified trajectory.

b) Rough adjustment: This operator changes i's positions at all frames within P_l [%] of the search range.

c) Fine adjustment: This operator changes i's position at frame f within P_l [%] of the search range.

d) QR code rotation: This operator changes background QR code direction at random.

The above operators are selected with the possibility $P_{trans}, P_{rough}, P_{fine}$, and P_{rotate}, respectively. One of the three operators, translocation, rough adjust, and fine adjustment, is selected first, and then QR code rotation operator is applied with the possibility P_{rotate}.

3.3 Evaluation of solution candidates

Evaluation of solution candidates is conducted using the following equation:

$$F(C) = P_A(C)^{w^{(p)}} \times Q_A(C)^{w^{(q)}}, \tag{4}$$

where $P_A(C)$ and $Q_A(C)$ are the decode possibility and the decoration quality, respectively. $w^{(p)}$ and $w^{(q)}$ are weight parameters.

Most of mobile phones try to decode QR code continuously like taking a movie, so not all of animation QR code's frame must be decoded. Decode possibility $P_A(C)$ is hence calculated by the following equation instead of eq. (2):

$$P_A(C) = \prod_{f=1}^{N_F - N_F^{ignore}} \frac{\sum_k p_k(C^{(\mathrm{best}f)}) \times w_k^{(p)}}{\sum_k w_k^{(p)}}, \tag{5}$$

where N_F indicates the number of frames, and $p_k(C^{(\mathrm{best}f)})$ is the f-th frame's decode possibility from the best one in solution candidate C in order of the decode possibilibty. N_F^{ignore} indicates the tolerance number of frames which cannot be

Table 1 Parameter configuration for generating still QR codes.

Parameter	Value
Number of individuals	10
Number of elites	2
Crossover	BLX-α
α	0.2
R	0.3
Crossover rate	0.8
Mutation rate	0.03
Generations	$N_I \times 50$
$(w^{(p)}, w^{(q)})$	(2.0, 1.0)
$(w_1^{(p)}, w_2^{(p)})$	(0.8, 0.2)
$(w_1^{(q)}, w_2^{(q)}, w_3^{(q)}, w_4^{(q)}, w_5^{(q)}, w_6^{(q)}, w_7^{(q)})$	(0.5, 0.5, 1.0, 0.5, 0.2, 0.1, 0.1)

decoded well. The lower N_F^{ignore} allows mobile phones to scan the QR code more quickly, but the problem becomes more difficult to solve.

The proposed method uses the following equations to calculate decoration quality $Q_A(C)$ instead of eq. (3).

$$Q_A(C) = \frac{1}{N_F} \sum_{f=1}^{N_F} Q(C^{(f)}) \quad (6)$$

$C^{(f)}$ is a subsolution of frame f which corresponds to a solution of still QR code decoration problem. $Q(C^{(f)})$ can therefore be evaluated by eq. (3).

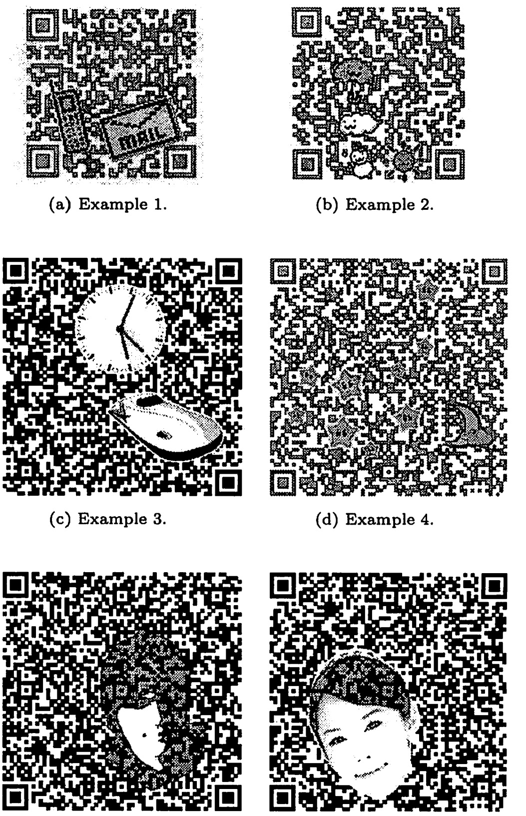
Although q_1 through q_7 can evaluate each frame quality of animated QR code, the proposed method uses following functions q_8 through q_{12} instead of q_1 through q_7 in order that placing illustrations as near as the specified positions takes first priority and keep illustration movements the same as the specified ones.

$q_8(C^{(f)})$ is a function which evaluates the difference between the specified positions of illustrations and the positions in which the illustrations are placed in $C^{(f)}$. $q_9(C^{(f)})$ and $q_{10}(C^{(f)})$ are functions which evaluate the difference between the specified scales and angles of illustrations, and scales and angles in which the illustrations are placed in $C^{(f)}$, respectively.

$q_{11}(C^{(f)})$ and $q_{12}(C^{(f)})$ are new functions proposed in this paper. $q_{11}(C^{(f)})$ evaluates how the illustrations move in accrdance with the specified trajectories. $q_{12}(C^{(f)})$ evaluates how the illustration scale changes in accordance with the specified scale changes.

4. Example outputs

In this section, we demonstrate some examples of decorated still QR codes by combining illustrations as described in Section 2.. Parameters were configured as shown in Table 1. The proposed method uses 10 individuals, which is



(a) Example 1. (b) Example 2.

(c) Example 3. (d) Example 4.

(e) Example 7. (f) Example 8.

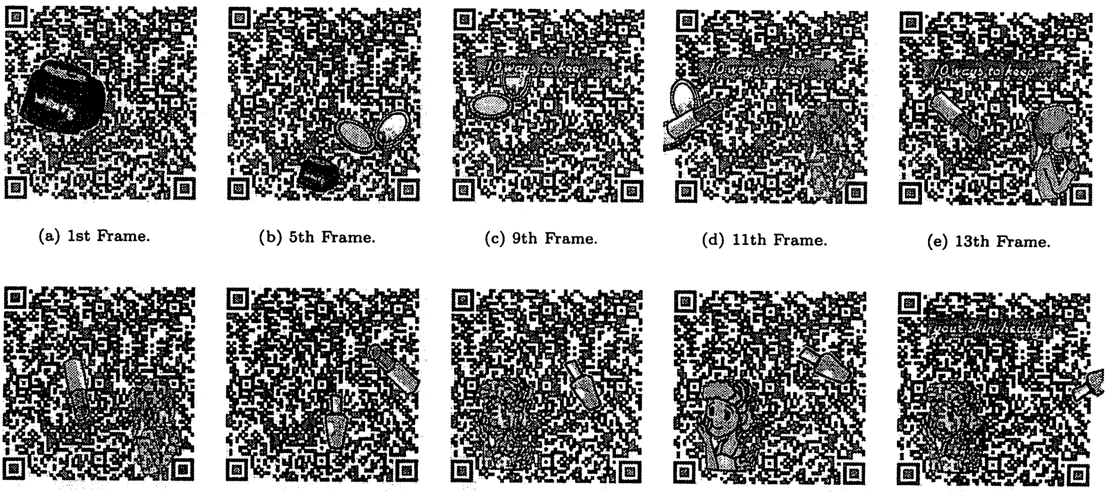
Fig. 5 Example still QR codes.

Table 2 Parameter configuration for generating animated QR code.

Parameter	Value
Number of neighbors at each iteration N_n	10
Iteration limit	200
Search range of operators	0.2
$(P_{trans}, P_{rough}, P_{fine}, P_{rotate})$	(0.7, 0.1, 0.2, 0.1)
N_F^{ignore}	0.1
$(w^{(p)}, w^{(q)})$	(2.0, 1.0)
$(w_1^{(p)}, w_2^{(p)})$	(0.8, 0.2)
$(w_8^{(q)}, w_9^{(q)}, w_{10}^{(q)}, w_{11}^{(q)}, w_{12}^{(q)})$	(0.01, 2.0, 2.0, 2.0, 3.0)

quite smaller than general GA configuration, because only one semi-optimal solution is sufficient in this experiment.

Fig. 5 shows output examples with various illustrations. Examples 1, 2, 3, and 4 shown in Fig. 5 involve two, four, two, and ten illustrations, respectively. Example 1 implies that this QR code involves an e-mail address of someone's mobile phone. Example 2 and 3 are appropriate for implanting web site addresses in which users can acquire weather forecast and a train timetable, respectively. Translucent colors would be promising to enlarge illustration sizes as shown



(a) 1st Frame. (b) 5th Frame. (c) 9th Frame. (d) 11th Frame. (e) 13th Frame.

(f) 14th Frame. (g) 18th Frame. (h) 20th Frame. (i) 22th Frame. (j) 26th Frame.

Fig. 6 Example animated QR code generated by the proposed method.

in Fig. 5(e) and (f).

Next, we tried to decorate and generate animated QR code involving 8 illustrations and 32 frames. Fig. 6 shows the generated QR code. The animated QR code is generated for implanting URL of a cosmetics manufacturer's web site. It must attract the attention of the people to show the animated QR code on monitors installed in trains, displayed on cosmetics counter in department stores, and so on.

5. Adopting rosubt optimization

As a future work, it must be a good idea to introduce robust optimization [10], [11] into the proposed method in this paper. Robust optimization technique is applicable to improve robustness againsft partial damage or small stain on a decorated QR code. Additional machine-readability evaluation would be conducted for decorated QR code images which are damaged or stained with random noise.

6. Conclusions

Proposed in this paper is a method for generating an animated QR code in which illustrations, photographs, logo or similar move and/or transform. Animated QR codes attract people's attention more effectively than general QR code. The proposed method uses real-coded GA to optimize illustration positions, scales, and angles at all frames simultaneously. The proposed method also uses operators and quality evaluation functions to keep illustration movement natural and smooth.

In the future, we plan to combine interactive evolutionary computation [12], and apply the combined model to the proposed QR code generation method.

References

[1] M. Rohs: "Real-world interaction with camera phones", Proc. Int'l Sympo. Ubiquitous Computing Systems 2004, pp. 74–89 (2004).

[2] S. Ono, K. Morinaga and S. Nakayama: "Two-dimensional barcode decoration based on real-coded genetic algorithm", Proc. IEEE World Congress on Computational Intelligence (WCCI 2008), pp. 1068–1073 (2008).

[3] S. Ono, K. Morinaga and S. Nakayama: "Animated two-dimensional barcode generation using optimization algorithms", Proc. Joint Int'l Conf. Soft Computing and Intelligent Systems and Int'l Sympo. advanced Intelligent Systems (SCIS&ISIS 2008), pp. 1232–1237 (2008).

[4] S. Ono, K. Morinaga and S. Nakayama: "Barcode design by evolutionary computation", Artificial Life and Robotics, **13**, 1, pp. 238–241 (2008).

[5] D. E. Goldberg: "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley, Reading (1989).

[6] C. Blum and A. Roli: "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", ACM Computing Surveys, **35**, 3, pp. 268–308 (2003).

[7] A. H. Wright: "Genetic algorithms for real parameter optimization", Foundations of genetic algorithms (Ed. by G. J. Rawlins), Morgan Kaufmann, San Mateo, CA, pp. 205–218 (1991).

[8] L. J. Eshelman and J. D. Schaffer: "Real-coded genetic algorithms and interval-schemata", Foundations of Genetic Algorithms, Vol. 2, pp. 187–202 (1993).

[9] L. J. Eshelman, K. E. Mathis and J. D. Schaffer: "Crossover operator biases: Exploiting the population distribution" (1997).

[10] J. Branke: "Creating robust solutions by means of evolutionary algorithms", Lecture Notes in Computer Science, **1498**, pp. 119–128 (1998).

[11] Y. Jin and J. Branke: "Evolutionary optimization in uncertain environments — a survey", IEEE T. Evolutionary Computation, **9**, 3, pp. 303–317 (2005).

[12] H. Takagi: "Interactive evolutionary computation – fusion of the capabilities of ec optimization and human evaluation", Proc. IEEE, Vol. 89, pp. 1275–1296 (2001).