2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

A Self-Adaptive Hybrid Genetic Algorithm for Color Clustering

Tarek El-Mihoub, Lars Nolle, Gerald Schaefer, Tomoharu Nakashima and Adrian Hopgood

Abstract — **Color palettes are inherent to color quantized images and represent the range of possible colors in such images. When converting full true color images to palletized counterparts, the color palette should be chosen so as to minimize the resulting distortion compared to the original. In this paper, we show that in contrast to previous approaches on color quantization, which rely on either heuristics or clustering techniques, a generic optimization algorithm such as a self-adaptive hybrid genetic algorithm can be employed to generate a palette of high quality. Experiments on a set of standard test images using a novel self-adaptive hybrid genetic algorithm show that this approach is capable of outperforming several conventional color quantization algorithms and provide superior image quality.**

I. Introduction

True color images typically use 24 bits per pixel which results in an overall gamut of 2^{24} i.e. more than 16.8 million different colors. While nowadays most images are captured and stored in that format, in certain applications (for example display of images on limited hardware such as mobile devices and for compression and retrieval of images [1]) it is advantageous to limit the range of possible colors to fewer entries whose ensemble are known as a color palette. Color quantization is the process of generating a suitable palette (usually of size between 8 and 256) where suitable is often defined as introducing as little distortion as possible, or equivalently, as maintaining the best possible image quality.

In this paper we apply a self-adaptive hybrid genetic algorithm (SAHGA) as a standard black-box optimization approach to the color quantization problem. The main advantage of black-box optimization algorithms is that they do not require any domain specific knowledge yet are able to provide a near optimal solution. We evaluate the effectiveness of our approach by comparing its performance to the results obtained by several purpose built color quantization algorithms [2-4]. The results obtained show that even without any domain specific knowledge our SAHGA based algorithm is able to outperform standard quantization algorithms and hence to provide palletized images with superior image quality.

The rest of the paper is organized as follows. The next section provides a formal definition of the color quantization problem. Section III provides the background for optimization based on self-adaptive hybrid genetic algorithm. Section IV explains our application of SAHGA, a modified HGA algorithm, to the color quantization problem. Section V presents experimental results based on a set of standard test images while Section VI concludes the paper.

Tarek El-Mihoub, Lars Nolle, Gerald Schaefer, and Adrian Hopgood are with the School of Computing and Informatics, Nottingham Trent University, Nottingham, UK.

Tomoharu Nakashima is with the College of Engineering, Osaka Prefecture University, Osaka, Japan.

II. Color Quantization

Color quantization produces a color palette that contains only a small number of colors (usually between 8 and 256); pixel data are then stored as indices to this palette. Clearly, the choice of colors that make up the palette has a crucial influence on the image quality of the quantized image. Formally, given an image quality metric which assigns $d(I_1(x,y),I_2(x,y))$ as the distance (or difference) between two pixels at location (x,y) in images I_1 and I_2, an n x m original image $O = \{o_i=\{R_i,G_i,B_i\}, i = 1\ldots n \times m\}$, a palette of size N, $P = \{p_j = \{R_j,G_j,B_j\}, j = 1\ldots N\}$, P is optimal iff

$$\begin{aligned}&\neg\exists\, \overline{P} = \{\overline{p}_k = \{R_k,G_k,B_k\}, k = 1...N\}\\ &\text{so that } D(O,q(O,\overline{P})) < E(O,q(O,P))\end{aligned} \tag{1}$$

with E (the error between two images) defined as

$$E(I_1,I_2) = \sum_{x=1}^{n}\sum_{y=1}^{m} d(I_1(x,y),I_2(x,y)) \tag{2}$$

and q (the result of the quantization process)

$$q(O,P) = \{q_r = p_s, r = 1...n\cdot m / d(o_r,p_s) < d(o_r,p_t)\ \forall\, t \neq s\} \tag{3}$$

However, the selection of the optimal color palette is known to be an np-hard problem [4]. In the image processing literature many different algorithms have been introduced that aim to find a palette that allows for good image quality of the quantized image. In general these can be divided into heuristic techniques such as the popularity algorithm [4] and clustering-based algorithms such as the median cut approach [4].

III. Hybrid Genetic Algorithm

Genetic algorithms [5] and other search methods can be seen as complementary tools that can be brought together to achieve an optimization goal. In these hybrids, a genetic algorithm incorporates one or more methods to improve the performance of the genetic search. There are several ways in which a search or optimization technique can complement the genetic search [6-10].

If a genetic algorithm is combined with a fast converging local search methods [11] the resulting hybrid can often outperform the algorithms [12]. Hybridizing a local search method provides the global genetic search algorithm with some local knowledge that can guide and may accelerate the search to the global optimum [13]. Figure 1 shows a flowchart of the basic hybrid genetic algorithm. As it can be seen from the figure, after the genetic operators are applied in order to generate a new generation, each of the individuals in the new generation undergo optimization using a local search method.

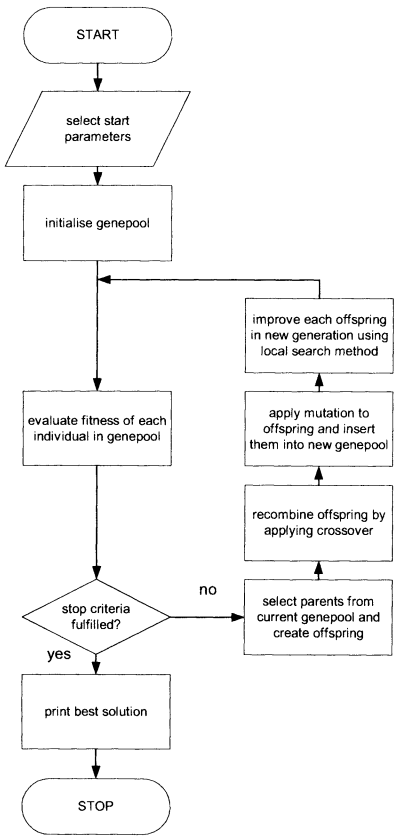

Figure 1 – Flowchart of hybrid Genetic Algorithm.

In this research, a new self-adaptive hybrid genetic algorithm (SAHGA) has been used, which employs a novel local search method, which is described in the following section.

A. Local Search Method used

The new local search algorithm used in this research is a probabilistic method that works on the genotype space by using a sub-group of the current population of solutions to optimize the structures of each solution present in the genepool. In this way, it aims to make use of some of the valuable genetic search information. It also aims to avoid disrupting the genetic schema processing by improving the solution in accordance with the global genetic search.

The modification of the initial solution based on a group of solutions of the genetic population can provide the local search method with a partial global view of the problem at hand. Based on this view, the search method can produce a solution in the context of global view captured by the genetic algorithm. This form of search can minimize any conflict with the global genetic search. The partial global aspect of the search method can be controlled by the group size and the mechanism of selecting the group members. This method is also characterized by its low costs. Its costs are equal to the costs of evolving a solution for a single iteration of the genetic search (i.e. one function evaluation per solution). This can help to minimize the loss of the hybrid's time in the case of any undesirable interference between the two search methods. Figure 2 provides pseudo code for the algorithm.

```
Procedure LocalSearch
Begin
  For each individual s0 in genepool
  Begin
    Select randomly group of individuals
    s1,s2,s3,s4;
    For each gene g in s0
    Begin
      If g = 1
        Set probability p(g) to 1.0;
      Else
        Set probability p(g) to 0.0;
    End
    For each individual s in group
    Begin
      Calculate probability value v as the absolute
      fitness difference between s0 and s,
      normalized by dividing it by the sum
      of differences between each group member
      and s0;
      For each allele a in s
      Begin
        If a equal to corresponding allele in s0
          Set corresponding probability in
          probability vector for s to 0.0;
        Else if a less than allele in s0
          Set probability vector for s to +v;
        Else
          Set probability vector for s to -v;
      End
    End
    Add all probability vectors for s0,s1,s2,s3,s4;
    Generate random vector r;
    For each element in random vector
    Begin
      If probability vector > random element
        Set corresponding gene in new solution to
        1;
      Else
        Set corresponding gene in new solution to
        0;
    End
    Evaluate new solution;
    If fitness of new solution > fitness of s0
      Replace s0 with new solution;
  End LocalSearch
```

Figure 2 – Pseudo code of local search method.

The algorithm assumes that each gene contributes uniformly to the fitness of the solution. Based on this assumption, the search method compares the genetic structure and the fitness of the solution to be improved with the structures and the fitness of a group of solutions selected from the current genetic population. Depending on the differences in both the structure and the fitness between this solution and the group members, the solution structure is modified in the direction of improving its fitness score. The new solution is evaluated and then inserted back into the population if it shows an improvement in its fitness.

B. The Self-Adaptive Hybrid Genetic Algorithm (SAHGA)

The success of such a hybrid algorithm in solving a given problem efficiently depends on its success in achieving a balance between exploration and exploitation [12, 13]. Among the factors that affect this balance is the duration of local search [14], which is defined as the number of the consecutive local search iterations that is performed on a solution before terminating a local search procedure. This control parameter can be used to adapt the hybrid on-line to a specific problem.

In the proposed hybrid algorithm, the number of local search iterations is incorporated into the representation of an individual. Through this parameter, the duration of a local search is controlled. It defines the number of local iterations that should be performed by the associated individual. The global genetic algorithm evolves the number of local search iterations parameter while the hybrid is using that control parameter to optimize the fitness function variables. Through adopting the evolutionary self-adaptation metaphor, the algorithm allows the global genetic algorithm to dynamically decide on the individuals that should perform a local search. It also decides on the duration of the local search method through modifying the number of local iterations as it co-operates with the local search to solve a given problem. This can facilitate the adaptation of number of local search iterations control parameter without exogenous control.

In general, the control parameters in the evolutionary self-adaptive algorithm can be adapted either at the individual level (i.e. local level) or at the population level (i.e. global level). In the local adaptation, the control parameter is applied to the associated solution only. In contrast, the control parameter in the global adaptation is tied to the population as a whole and not to a particular solution. The number of local iterations of an individual is computed by taking the average of the number of local iterations of the individuals of the whole population. Local adaptation is used in the proposed algorithm because it is reasonable to assume that different individuals are following different paths through the search space. It is also proven that local adaptation outperforms global adaptation [15].

In the proposed self-adaptive hybrid algorithm, after performing a genetic iteration, the number of local iterations associated with each solution is extracted from the chromosome's structure. Depending on the value of that parameter, a number of local search iterations are performed on that solution. If the value of that parameter is zero, no local search iteration will be performed. Otherwise the specified number of local iterations will be performed consecutively. Using the learning strategy specified by the algorithm, the resulting solution is mapped back to the mating pool.

The maximum value of the number of local iterations was set to three. The reason for selecting this value is the expected benefits of using small durations of local search to fight the hindering effect problem. The algorithm also makes use of the number of local iteration control parameter, which already exists within the chromosome, to discriminate between innate and acquired fitness. In a case of an equal fitness, the algorithm chooses the individual with the smaller value of local search iterations since its acquired fitness is closer to the innate one. This can help to alleviate the consequences of the hindering effect problem [11] associated with the Baldwinian approach.

IV. SAHGA FOR COLOR QUANTIZATION

In this paper we apply the SAHGA algorithm described above as a black box optimization algorithm to the color quantization problem. For color quantization the objective is to minimize the total error introduced through the application of a color palette. The color palette P for an image O, a codebook of N color vectors, should then be chosen so as to minimize the error function E

$$E(P,O) = \frac{1}{\sum_{j=1}^{N} l_j} \sum_{i=1}^{k} \sum_{j=1}^{l_j} \left\| P_i - O_j \right\| + c(P,O) \qquad (4)$$

with

$$c(P,O) = \sum_{i=1}^{k} \delta a_i, \quad a_i = \begin{cases} 1 & \text{if } l_i = 0 \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

where l_i is the number of pixels O_j represented by color P_i of the palette, $\|.\|$ is the Euclidean distance in RGB space, and δ is a constant (δ=10 in our experiments). The objective function $E(P,O)$ used is hence a combination of the mean Euclidean distance and a penalty function. The penalty function $c(P,O)$ is integrated in order to avoid unused palette colors by adding a constant penalty value to the error for each entry in the codebook that is not used in the resulting picture. As can be seen from Equation 4 the objective function is highly non-linear, i.e. it has a high degree of epistasis [16].

For our color quantization algorithm we employ the SAHGA algorithm with a population size of 100. For all experiments, binary tournament selection, single-point crossover, and simple mutation were used with a crossover probability of 0.6 and a mutation probability of 0.01. Each experiment was repeated 19 times and the codebook with the median error was used for comparison.

V. EXPERIMENTAL RESULTS

In order to evaluate our new method for color quantization, we have taken a set of three standard images commonly used in the color quantization literature, *Lenna*, *Pool*, and *Airplane*, and applied our optimization scheme to generate quantized images with a palette of 16 colors.

To put the results we obtain into context, we have also implemented four popular color quantization algorithms to generate corresponding quantized images with palette size 16. The algorithms we have tested were:

- Popularity algorithm [4]: Following a uniform quantisation to 5 bits per channel the n colours that are represented most often form the colour palette.

- Median cut quantisation [4]: This algorithm starts by computing the box that encompasses all colours present in the image. The box is then split (orthogonal to the colour axis) at the median value into two sub-cubes. The larger remaining sub- cube is then again divided at its median point and this process is repeated until n colour boxes have been found.
- Octree quantisation [3]: The colour space is represented as an octree where the root node corresponds to the whole colour space, the nodes at the next level the eight sub-cubes that are obtained by dividing each colour axis into two equal halves, and so on. In a first pass the sub-tree that represents the colours present in the image is built and in a second pass, starting at the bottom of the tree, nodes are successively merged until a tree of n colours is reached.
- Neuquant [2]: A one-dimensional self-organising Kohonen neural network is trained to generate the colour map. The Kohonen network defines a mapping from the colour values in the image to an index representing the palette entries. The weights of the network are updated based on the image data to ensure an optimal palette with good image quality.

For all algorithms, pixels in the quantised images were assigned to their nearest neighbours in the colour palette to provide the best possible image quality.

The results are listed in Tables 1 and 2, expressed in terms of mean-squared-error (MSE) and peak-signal-to-noise-ratio (PSNR) defined as

$$\mathrm{MSE}(I_1,I_2) = \frac{1}{3mn}[(R_1(i,j)-R_2(i,j))^2 + (G_1(i,j)-G_2(i,j))^2 + (B_1(i,j)-B_2(i,j))^2] \quad (6)$$

and

$$\mathrm{PSNR}(I_1,I_2) = 10\log_{10}\frac{255^2}{\mathrm{MSE}(I_1,I_2)} \quad (7)$$

where $R(i,j)$, $G(i,j)$, and $B(i,j)$ are the red, green, and blue pixel values at location (i,j) and n and m are the dimensions of the images.

As MSE and PSNR are not necessarily the best quality indicators, the results are also provided in terms of S-CIELAB [17]. This is an image quality metric based on uniform colour spaces but it also takes into account the spatial interaction between neighbouring pixels based on a blurring effect derived from psychophysical experiments. S-CIELAB results, expressed in terms of ΔE differences between original and quantised images are provided in Table 3.

	Popalg	Medct	Octree	Neuqu.	SAHGA
Lenna	388.1	271.4	117.0	107.4	**93.4**
Pool	669.9	226.9	74.9	127.2	**60.3**
Airplane	1668.1	240.7	86.5	97.6	**72.8**
all	908.7	246.3	92.8	110.7	**75.5**

Table 1. Quantization results, given in terms of MSE.

	Popalg	Medct	Octree	Neuqu.	SAHGA
Lenna	22.24	23.79	27.45	27.82	**28.43**
Pool	15.91	24.32	28.77	28.24	**30.53**
Airplane	19.87	24.57	29.39	27.08	**29.51**
all	19.34	24.23	28.54	27.71	**29.45**

Table 2. Quantization results, given in terms of PSNR [dB].

	Popalg	Medct	Octree	Neuqu.	SAHGA
Lenna	11.92	21.81	20.54	41.03	**9.31**
Pool	10.34	8.89	10.66	10.10	**7.73**
Airplane	7.20	7.92	9.36	6.66	**4.85**
all	19.34	24.23	28.54	27.71	**29.45**

Table 3. Quantization results, given in terms of S-CIELAB ΔE.

From Tables 1 to 3 we can see our self-adaptive hybrid genetic algorithm approach to colour quantisation obtains clearly the best results for all three images. Overall a mean PSNR (MSE) of 29.45 dB (75.5) is achieved which is significantly better than the 28.54 dB (92.8) and 27.71 dB (110.7) obtained by Octree and Neuquant, the two next best algorithms.

In terms of S-CIELAB, the hybrid algorithm provides image quality that is more than 2 ΔE units lower than the next best approaches. Considering that a difference of 1 ΔE unit is perceptually visible this indeed indicates a significant improvement.

An example of the performance of the different algorithms is provided in Figure 3, which shows the *Pool* image together with the quantization results from all algorithms. Figure 4 provides error images of the quantised images from Figure 1 compared to the original.

It is clear that the popularity algorithm performs poorly on this image and assigns virtually all of the colors in the palette to green and achromatic colors. Median cut is better but still provides fairly poor color reproduction; most of the colors in the quantized image are fairly different from the original. The same holds true for the images produced by Neuquant. Here the most obvious artifact is the absence of an appropriate red color in the color palette. A far better result is achieved by the Octree algorithm, although here also the red is not very accurate and the color of the cue is greenish instead of brown.

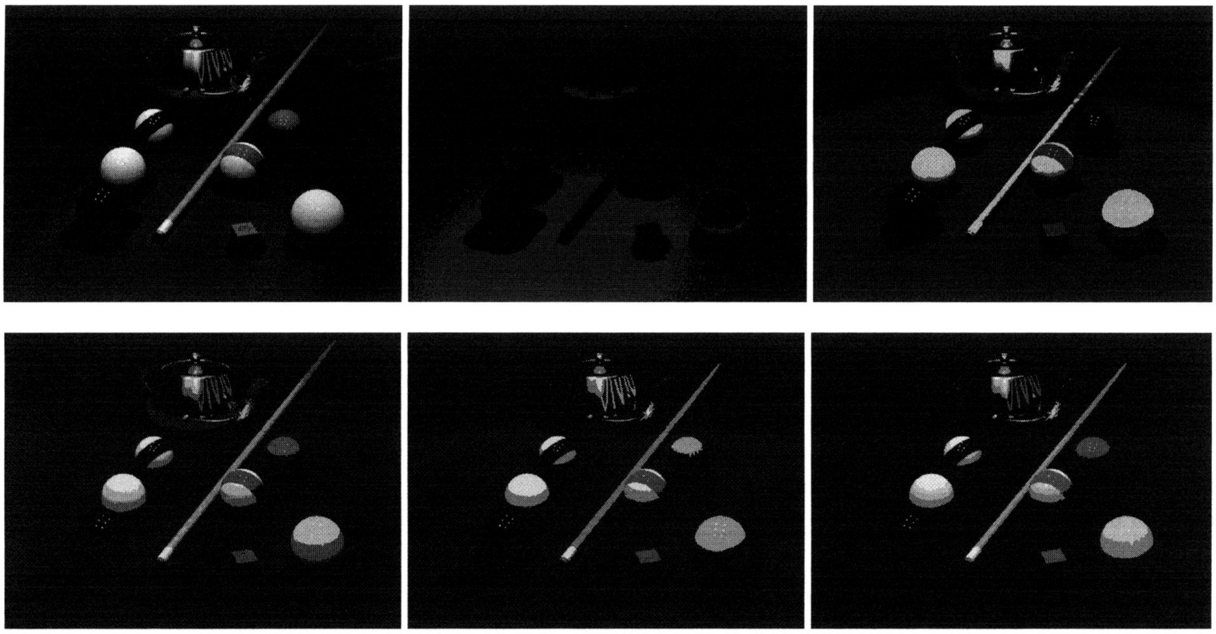

Figure 3 - Results of colour quantisation algorithms applied to *Pool* image (top left) after applying (from left to right, top to bottom) Popularity, Median cut, Octree, Neuquant, and SAHGA algorithms.

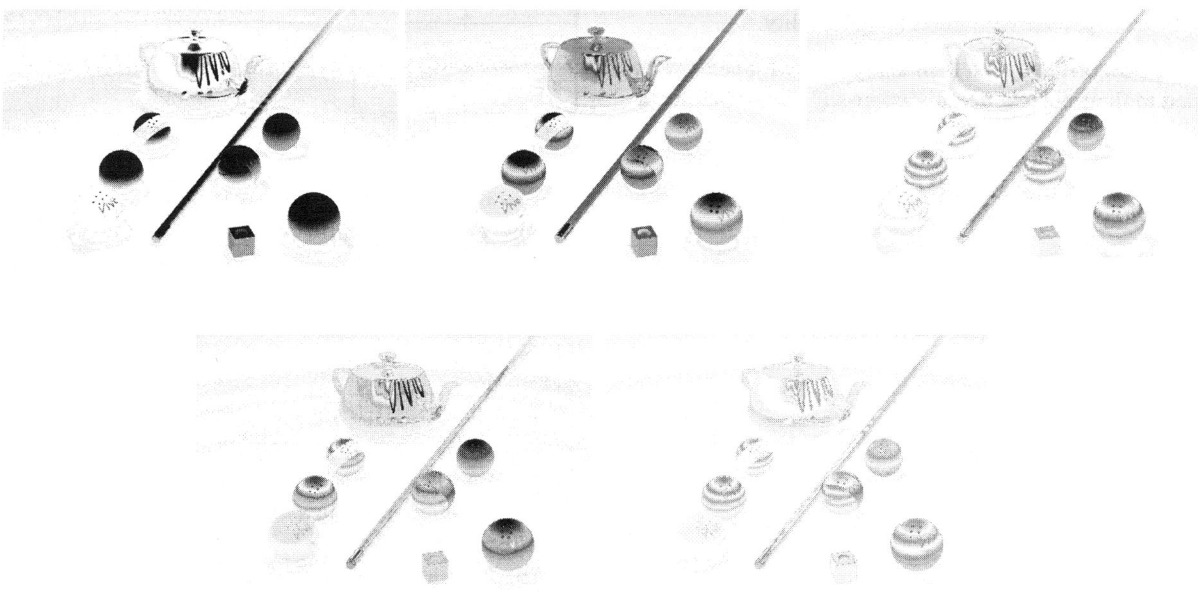

Figure 3 - Error images of the quantised images from Figure 1 for (from left to right, top to bottom) Popularity, Median cut, Octree, Neuquant, and SAHGA algorithms compared to the original.

Clearly the best image quality is maintained by applying our self-adaptive hybrid technique. Although the color palette has only 16 entries all colors of the original image are accurately presented including the red ball and the color of the billiard cue.

VI. Conclusion

In this work we have applied a novel self-adaptive hybrid genetic algorithm as a generic optimization algorithm to the color quantization problem. Experimental results obtained on a set of standard test images have demonstrated that this type of optimization techniques cannot only be effectively employed but is even able to outperform standard purpose built color quantization algorithms.

References

[1] G. Schaefer, G. Qiu, and G. Finlayson, "Retrieval of palettised colour images," in SPIE, vol. 3972 (Storage and Retrieval for Image and Video Databases VIII), 2000, pp. 483-493.

[2] A. H. Dekker, "Kohonen neural networks for optimal colour quantization," Network: Computation in Neural Systems, vol. 5, pp. 351-367, 1994.
[3] M. Gervautz and W. Purgathofer, "A simple Method for Color Quantization: Octree Quantization," in Graphics Gems, A. S. Glassner, Ed. San Diego, CA: Academic Press, 1998.
[4] P. S. Heckbert, "Color image quantization for frame buffer display," ACM Computer Graphics (ACM SIGGRAPH '82 Proceedings), vol. 16, pp. 297-307, 1982.
[5] J. Holland, Adaptation in Natural and Artificial Systems: The University of Michigan Press, 1975.
[6] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in the International Conference on Genetic Algorithms and their Applications. Hillsdale, USA: Lawrence Erlbaum, 1985, pp. 154-159.
[7] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," Soft Computing, vol. 9, pp. 3-12, 2005.
[8] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, "BOA: the Bayesian optimization algorithm," in the Genetic and Evolutionary Computation Conference. Orlando, USA: Morgan Kaufmann, 1999, pp. 525-532.
[9] X. Yao, "Evolving artificial neural networks," Proceedings of the IEEE, vol. 87, pp. 1423-1447, 1999.
[10] T. White, B. Pagurek, and F. Oppacher, " ASGA: improving the ant system by integration with genetic algorithms," in the third Conference on Genetic Programming (GP/SGA'98). Madison, USA, 1998, pp. 610-617.
[11] P. Turney, "Myths and legends of the Baldwin effect," in Workshop on Evolutionary Computation and Machine Learning at the 13th International Conference on Machine Learning. Bari, Italy, 1996, pp. 135-142.
[12] F. G. Lobo and D. E. Goldberg, "Decision making in a hybrid genetic algorithm," in IEEE International Conference on evolutionary Computation. Piscataway, USA: IEEE Press, 1997, pp. 122-125.
[13] W. E. Hart, "Adaptive global optimization with local search," Doctoral Dissertation. San Diego: University of California 1994.
[14] W. E. Hart, C. R. Rosin, R. K. Belew, and G. M. Morris, "Improved evolutionary hybrids for flexible ligand docking in AutoDock," in Optimization in Computational Chemistry and Molecular Biology, C. A. Floudas and P. M. Pardalos, Eds.: Springer 2000, pp. 209-230.
[15] W. M. Spears, "Adapting crossover in a genetic algorithm," in the Fifth Conference on Evolutionary Programming. San Diego, USA: MIT press, 1995, pp. 367-384.
[16] Y. Davidor, "Epistasis variance: suitability of a representation to genetic algorithms," Complex Systems, vol. 4, pp. 369-383, 1990.
[17] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color image reproduction," in SID Symposium Technical Digest, vol. 27, 1996, pp. 731-735.