MEDAL

Missouri Estimation of Distribution Algorithms Laboratory

Image Segmentation using a Genetic Algorithm and Hierarchical Local Search

Mark Hauschild, Sanjiv Bhatia and Martin Pelikan

MEDAL Report No. 2012002

February 2012

Abstract

This paper proposes a hybrid genetic algorithm to perform image segmentation based on applying the q-state Potts spin glass model to a grayscale image. First, the image is converted to a set of weights for a q-state spin glass and then a steady-state genetic algorithm is used to evolve candidate segmented images until a suitable candidate solution is found. To speed up the convergence to an adequate solution, hierarchical local search is used on each evaluated solution. The results show that the hybrid genetic algorithm with hierarchical local search is able to efficiently perform image segmentation. The necessity of hierarchical search for these types of problems is also clearly demonstrated.

Keywords

genetic algorithms, hybrid genetic algorithms, image segmentation, local search

Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Department of Mathematics and Computer Science
University of Missouri–St. Louis
One University Blvd., St. Louis, MO 63121
E-mail: medal@cs.umsl.edu
WWW: http://medal.cs.umsl.edu/

Image Segmentation using a Genetic Algorithm and Hierarchical Local Search

Mark Hauschild
Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Dept. of Math and Computer Science, 320 CCB
University of Missouri at St. Louis
One University Blvd., St. Louis, MO 63121
mwh308@umsl.edu

Sanjiv Bhatia
Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Dept. of Math and Computer Science, 320 CCB
University of Missouri at St. Louis
One University Blvd., St. Louis, MO 63121
sanjiv@acm.org

Martin Pelikan
Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Dept. of Math and Computer Science, 320 CCB
University of Missouri at St. Louis
One University Blvd., St. Louis, MO 63121
pelikan@cs.umsl.edu

Abstract

This paper proposes a hybrid genetic algorithm to perform image segmentation based on applying the q-state Potts spin glass model to a grayscale image. First, the image is converted to a set of weights for a q-state spin glass and then a steady-state genetic algorithm is used to evolve candidate segmented images until a suitable candidate solution is found. To speed up the convergence to an adequate solution, hierarchical local search is used on each evaluated solution. The results show that the hybrid genetic algorithm with hierarchical local search is able to efficiently perform image segmentation. The necessity of hierarchical search for these types of problems is also clearly demonstrated.

Keywords: genetic algorithms, hybrid genetic algorithms, image segmentation, local search

1 Introduction

Image segmentation is the process of dividing an image into multiple distinct segments. It is one of the most important applications in computer vision and image processing. Good image segmentation can be used to help emphasize boundaries and locate distinct objects in images and is often used as a preliminary step in computer vision. Unfortunately, segmentation is also one of the most difficult tasks in image processing.

This paper focuses on image segmentation of grayscale images using the q-state Potts model. While this model has been used in the past to perform image segmentation, the previous methods either used a Monte Carlo approach that takes exponential time as image size increases (Descombes, Moctezuma, Maître, & Rudant, 1996; Neirotti, Kurcbart, & Caticha, 2003; Peng, Urbanc, Cruc, Hyman, & Stanley, 2003; Tanaka, 2002) or directly applied the q-state model to the original image itself (Bentrem, 2009). In this paper, the input image is used to generate a set of weights for a q-state spin glass. Then candidate solutions are evolved based on these set of weights using a genetic algorithm (GA). The candidate solution that results in the lowest spin glass energy is the final image segmentation returned by the algorithm.

While GAs have been used on an almost endless variety of problems (Goldberg, 1989), using a GA for image segmentation presents an interesting challenge. Even relatively small images have a large number of data points compared to most GA applications. This can lead to long convergence times as well as an increased cost to crossover and fitness functions. To help alleviate these problems, it is helpful to use as small a population as possible. We achieve this by using two methods. First, a steady-state genetic algorithm is used to help preserve diversity. Second, a powerful local search is needed to allow a small population size, both to ensure a small initial population size is sufficient and to decrease the convergence time to an adequate solution. A hierarchical local search is used with the steady-state GA to create a powerful hybrid GA capable of performing image segmentation. In addition, we note that it is possible for two segmented images to have the same physical regions. To overcome this problem, a transformation is used to map different candidate solutions to the same region assignments when performing crossover.

The paper is organized as follows. Section 2 describes the q-state Potts model and how we transform the input image into a set of weights. Section 3 decribes the hybrid GA used to evolve candidate image segmentations. Section 4 covers the experimental setup as well as the experimental results. Finally, section 5 summarizes and concludes the paper.

2 Q-state Potts Model

Spin glasses are prototypical models for disordered systems and have played a central role in statistical physics during the last three decades (Binder & Young, 1986; Fischer & Hertz, 1991; Mezard, Parisi, & Virasoro, 1987; Young, 1998). A simple model to describe a finite-dimensional Potts spin glass is typically arranged on a regular 2D or 3D grid where each node i corresponds to a spin s_i and each edge $\langle i, j \rangle$ corresponds to a coupling between two spins s_i and s_j. Each edge has a real value $J_{i,j}$ associated with it that defines the relationship between the two connected spins. In general, the computational task is to find an assignment of spins such that it minimizes the total energy of the system.

In this paper, the first step to our goal is to map a grayscale image to a set of spin glass weights. First we convert the grayscale image into a set of weights using a method similar to reference (Peng, Urbanc, Cruc, Hyman, & Stanley, 2003). The weight $J_{i,j}$ between neighboring pixels at locations i and j (spins) is given by:

$$J_{i,j} = 1 - \frac{\Delta_{i,j}}{\theta \bar{\Delta}} \, , \tag{1}$$

where $\Delta_{i,j}$ is the absolute grayscale difference between the neighboring pixels in the original image and $\bar{\Delta}$ is the average difference between neighboring pixels in the entire image. From this equation, we see that if the difference between pixels i and j is large, then the corresponding weight parameter $J_{i,j}$ will be low. On the other hand, the closer the values of pixels i and j, the closer the weight will be to 1.

The overall goal is to minimize the energy of the q-state spin glass. Using the Kronecker δ function (which is 1 if the values are equal and 0 othrwise), energy is defined as

$$E = -\sum_{\langle i,j \rangle} J_{i,j} \delta_{\sigma_i \sigma_j} \ , \qquad (2)$$

The strength of the interaction is given by the weight $J_{i,j}$. However, the two spins only interact if they are nearest neighbors and are in the same spin state since $\delta_{\sigma_i,\sigma_j}$ is the Kronecker δ function.

Once given a set of weights, it is now possible to segment our image by assigning each pixel in a new image to one of a range of possible grayscale colors corresponding to unique regions and then, finding the set of assignments to such regions that minimize the energy E. This works since using the set of weights and trying to minimize the overall energy of the system yields a high probability that neighboring pixels that are nearly the same shade of grayscale values in the input image will be assigned to the same region in the segmented image. On the other hand, if two neighboring pixels are of very different gray scale levels they will most likely be placed into different regions in the segmented image result.

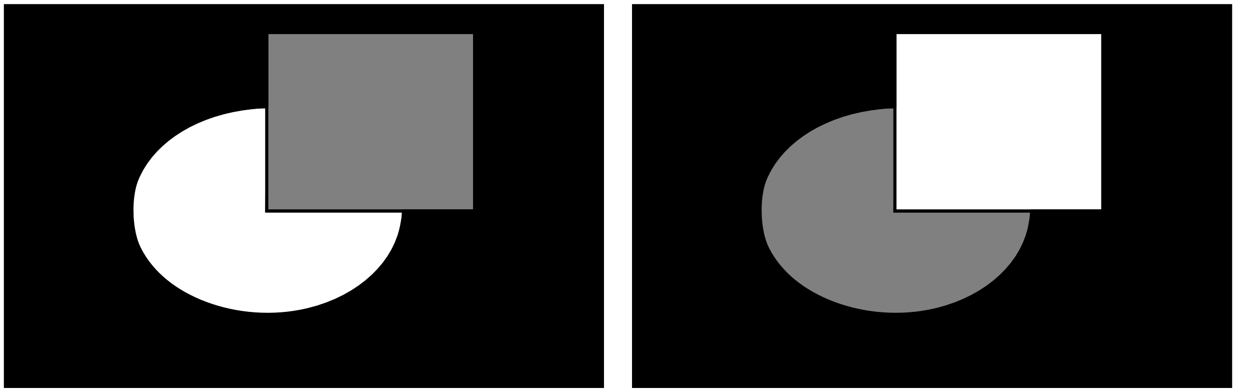
3 Algorithm

The steady-state genetic algorithm (GA) (Goldberg, 1989; Holland, 1975) evolves a population of candidate solutions typically represented by strings of a fixed length. The initial population is generated at random according to a uniform distribution over all of those strings. Each iteration starts by randomly selecting two parents and then variation operators such as crossover is used to create a new solution. The new candidate solution is then incorporated into the population using a replacement operator. The run is terminated when some termination criteria has been met. The basic steady-state GA procedure is laid out in Algorithm 1.

Algorithm 1 Steady-state GA pseudocode

```
$g \leftarrow 0$
generate initial population $P(0)$
while (not done) do
  select two parents $a$ and $b$ randomly from the population
  generate new individual $c$ by crossover of $a$ and $b$
  perform mutation on new individual
  if Fitness($b$) > Fitness($a$) then
    Swap($a$,$b$)
  end if
  if Fitness($c$) > Fitness($b$) then
    $b \leftarrow c$
  end if
  $g \leftarrow g + 1$
end while
```

In this work, we have used two different variation operators. The first, the *crossover* operator, described in Section 3.1, swaps a rectangular section between the two images with a transformation applied before crossover to ensure that the segment color schemes are consistent between the two parents. Then, we apply the *mutation* operator, with each character of the solution string having a probability of $1/n$ (where n is the number of pixels in the image) to be assigned randomly to one



(a) Image a

(b) Image b

Figure 1: Two images with identical segmented regions but different region assignments.

of the regions. Finally, a hierarchical local searcher is used on each solution when it is evaluated to improve performance. If the new candidate solution is of higher fitness than the least fit parent solution, it replaces the least fit parent solution in the population.

3.1 Crossover Operator

The goal of our crossover operator is to take parent images a and b and create a new image c that incorporates elements of both the parent images. First, we randomly select a pivot pixel as the pixel that will be in the center of the rectangular region. For an image of size x columns and y rows, a rectangular region of size $\sqrt{x/2}$ columns and $\sqrt{y/2}$ rows is used, centered on the pivot pixel. This region size was used to ensure that the crossed over region was relatively small as image size increased, to minimize disruption during crossover. Then, from image a, add those pixels in the rectangular region to image c and all other pixels in c are set to the values in image b.

The application of this crossover operator without any transformation leads to a poor quality to the new child. This is because the individual segmented images can be essentially segmented the same but be done using different numbered assignments. Figure 1 shows two simple segmented images, segmented between black, white, and gray. It is clear that using a rectangular crossover between these two images would result in a poor child solution. To attempt to alleviate this problem, we perform a transformation on the rectangular region from image a to keep its region assignments as similar as possible to b before adding it to the new individual c. This transformation is created using the possible region assignments $q \in \{0 \dots r\}$. For each pixel in the entire images a and b at each stage, we calculate the conditional probability that given a source pixel in a that is assigned to a state q_m, the probability that in b it is assigned to a state q_n. An example of one of these conditional probability tables is given in Table 1. In this table, the columns represent assignments in image a and the rows represent an assignment in image b, with the values in the table representing the conditional probability that a pixel in image a is of value p in image b.

We use a greedy algorithm to create the transformation from image a to b. All the pixels taken from the target rectangular region in a are mapped to the highest conditional probabilities of their region assignment in b before being added to the new individual c. Referring back to Figure 1, it is clear that the best transformation from image a to b would be to map all white pixels to gray and all gray pixels to white. In this way, the disruption to similar regions in a and b is

	$p = 0$	$p = 1$	$p = 2$	$p = 3$
$p = 0$	0.7902	0.0022	0.0989	0.1998
$p = 1$	0.0828	0.0162	0.0603	0.0532
$p = 2$	0.0505	0.6261	0.4766	0.6950
$p = 3$	0.0765	0.3554	0.3641	0.0520

Table 1: Example conditional probability table generated during crossover that shows the probability that a pixel p in image a is assigned to a different state in image b.

minimized when combined in c. For a more complex example,using the data in Table 1, the highest conditional probability results in assigning segment 0 of image a to segment 0 of image b. The next highest assigns segment 2 of image a to segment 3 of b. Continuing on, the final transform array is $T = [0, 1, 3, 2]$.

3.2 Hierarchical Local Search

Due to the large number of variables in each solution string, it is important that a local search be used with the GA to allow us to use a smaller population and to ensure quick convergence to an adequate solution. Initially, a simple deterministic hill climbing (DHC) local search was used. This hill climbing algorithm would find the largest possible fitness gain by changing any single variable. This process is repeated until no single variable change would lead to an improvement.

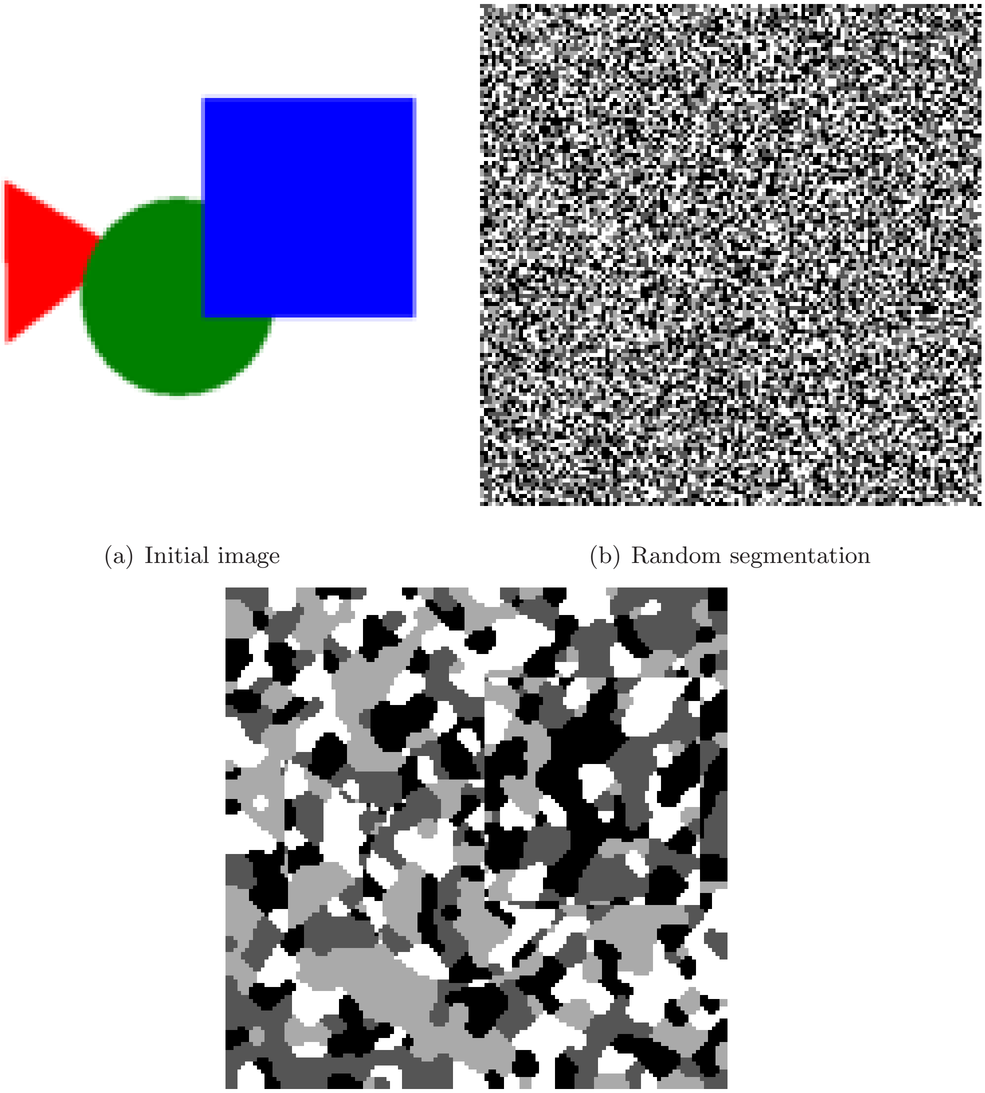
From other works on local search in spin glasses (Pelikan, 2005), a dramatic increase in segmented image quality was expected. For example, it might be expected that local search would expand a region to cover an entire monocolored region of an image. Unfortunately, this was not the case. Instead, the local search repeatedly got stuck in many poor local optima.

Figure 2a shows a simple example image of only 3 shapes with the exact same color in all shapes. Figure 2b shows the segment assignment after an individual is randomly initialized. Figure 2c shows the segmentation after the application of DHC. We see in this image that instead of assigning the monocolored area to the same region, the local search gets stuck and the final assignments are many different patchy areas that are of the same color.

Figure 3 shows the reason for the local search getting stuck in poor local optima. Figure 3 shows a subset of a segmented image where the original image is all of the same color. This implies that assignments to adjacent segments that are the same have zero cost and all others have the same penalty. The local search has assigned the lower diagonal part of the image to gray. However, at this point, the local search gets stuck. This is due to a having 3 neighbors that give penalties (due to being of different segment assignments when the underlying color was the same). If a is assigned to gray, those 3 penalties go away but an additional 5 extra penalty terms are added. With different segmented solutions containing many patches assigned to various regions even in same color areas, this resulted in the crossover operator having poor performance.

Examining Figure 2a led us to consider a hierarchical local search. The hierarchical local search algorithm works by initially performing the simple DHC. Then, all contiguous colored regions are identified and treated as a single region assignment. The local search is then performed again, this time attempting to assign each region to a new color and seeing the fitness gain. The assignment that leads to the highest fitness gain is then taken, which usually leads to some regions being merged together. This process is then repeated with the new set of regions. The overall process terminates when no region, when fully assigned to a new color, leads to a fitness gain.

Figure 4 shows several steps of the hierarchical search in action. We see that over time, the



(a) Initial image

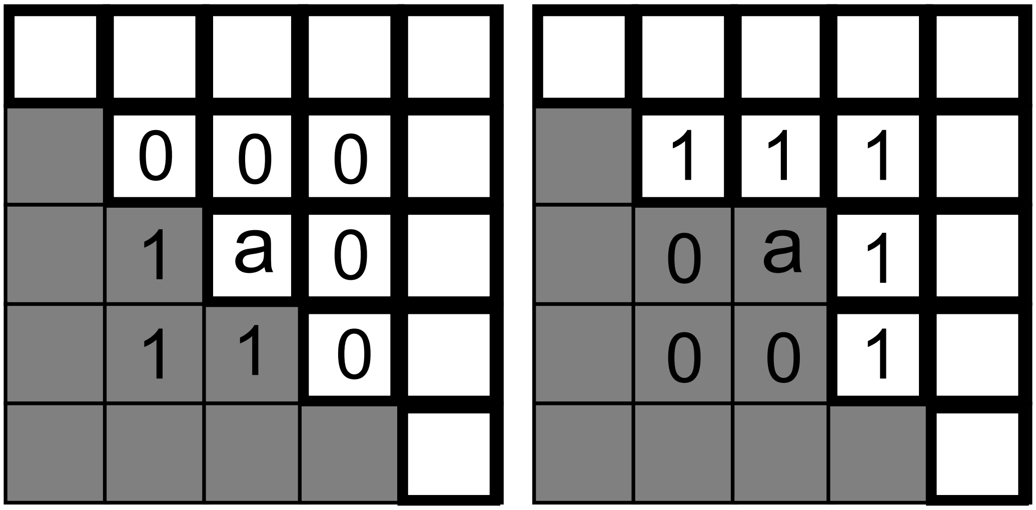
(b) Random segmentation

(c) Segmentation after DHC

Figure 2: An image containing 3 monocolor shapes and its corresponding randomly initialized segmentation and the segmentation result after DHC. Note that local search gets stuck in many poor local optima.

hierarchical search merges regions together and eventually, leads to the same color areas being completely covered by the same region assignment as we might have expected initially from just using DHC. In Figure 4a, it is hard to perceive the original shapes. However, by *step* = 50 in Figure 4b the shapes begin to become clear. After the final hierarchical search step, the original image is much more noticeable but clearly more work needs to be done than simply the hierarchical local search.

To further emphasize the necessity of local search, the GA was run on the 3 shapes image with different local search methods with the population set to 50 initial randomly generated solutions. Figures 5 shows the resulting image segmentations at generation $g = 1$, $g = 100$ and $g = 1000$ using



(a) Before a

(b) After a

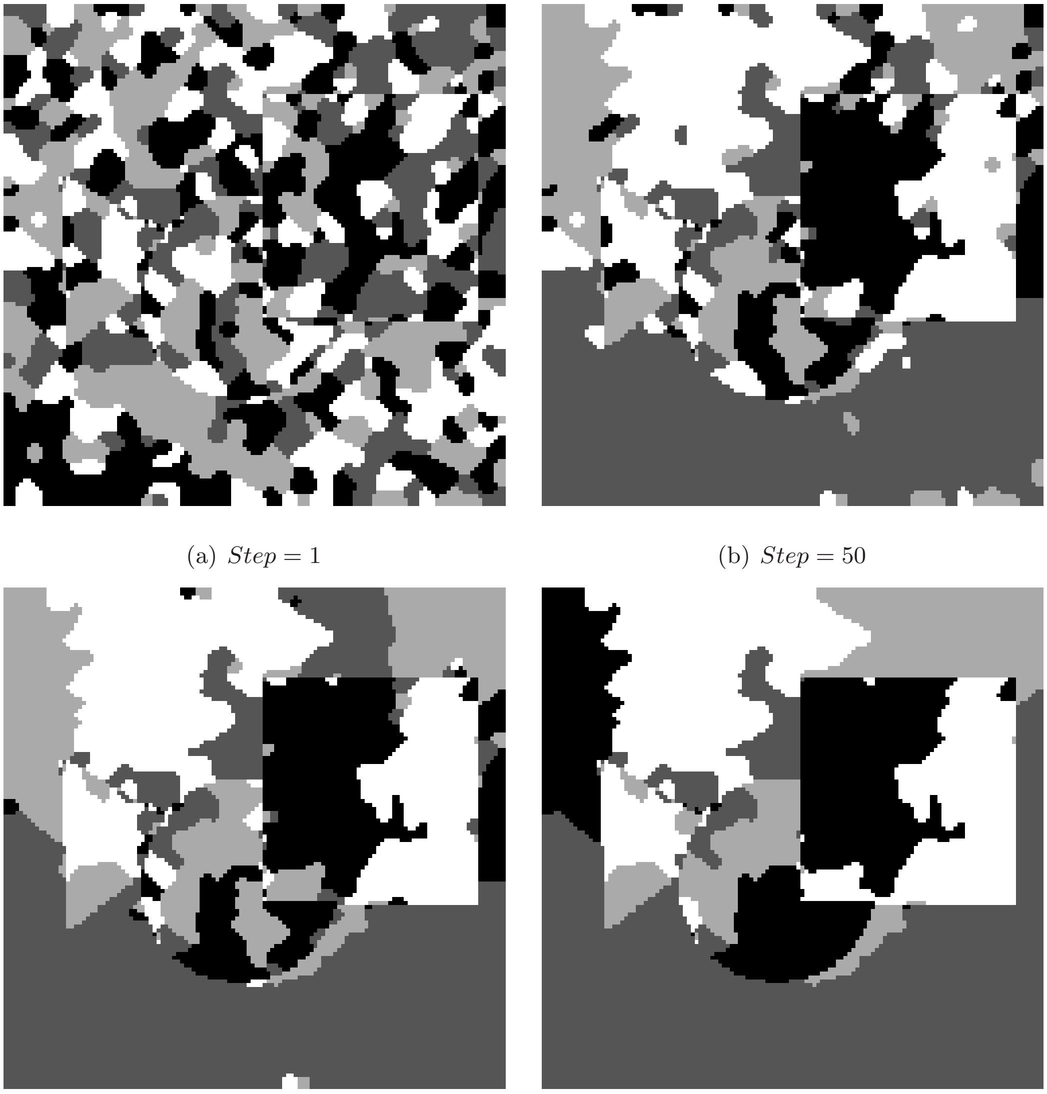
Figure 3: A subset of a segmented image with associated penalties to fitness before and after a is switched between regions.

no local search, DHC and the hierarchical local searcher. Figures 5a-c are the results using no local search and even after 1000 generations there is little to distinguish the final segmentation from the initial segmentation result. Using DHC in Figures 5d-f, we do see that as the generations increase, the segmented regions are getting larger, so the image is slowly converging to the solution. Even for $g = 1000$ however, the segmentation is of poor quality and it is very hard to distinguish the shapes in the initial image. When the GA is run with the hierarchical search as in Figures 5g-i, we see a dramatic improvement, with the GA able to find the optimal solution in 100 generations. We also explored performance with and without region maps and while we would expect the lack of region mapping to hurt good segmentations, the hierarchical local searcher seems to be more important than a crossover that minimizes disruption.

This hierarchical search process is considerably more computationally expensive than simple DHC. However, when determining the gain or loss in fitness from switching the color assignment of a region, it is only necessary to check the gain along the border between different regions. This is because internal to the same region, switching it entirely to a different color does not change the overall fitness value for that region. In addition, as the hierarchical local search progresses and changes in some region's color result in merged regions, some boundary pixels become internal pixels and are no longer considered. Also of note is that only the initial hierarchical local searches require many steps since the individual regions become much simpler after the initial search.

4 Experiments

This section covers both the experimental setup used for the hybrid GA and the resulting segmented images.



(a) $Step = 1$

(b) $Step = 50$

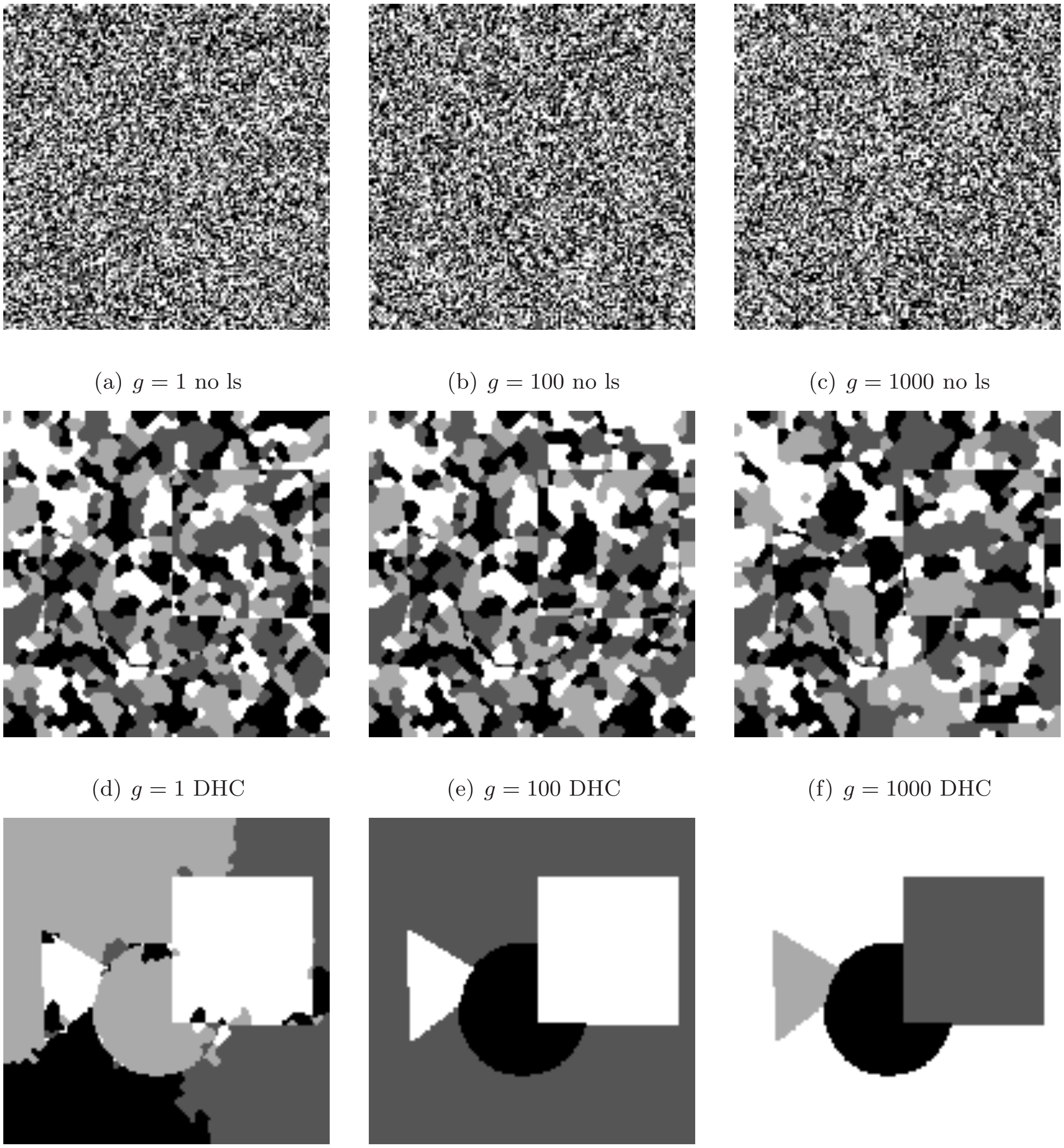
(c) $Step = 100$

(d) $Step = 143$

Figure 4: Several steps of the hierarchical local search as it merges regions together over time. $Step = 1$ is after the first merge of a region and $step = 143$ is after the final region merge with no other region merges giving any gains.

4.1 Experimental Setup

For an image of x columns and y rows, the GA evolves strings of size $n = xy$. The values of each individual variable in the string can be set to $q \in \{0 \ldots r - 1\}$, where r is the maximum number of different region assignments. In this paper, we have used the value $r = 4$, setting the cardinality of the variables in the solution strings to 4. The initial population for all experiments was set at $N = 50$, with all candidate solutions being initialized to random values. The hybrid GA was run for 1000 generations. All input images were converted to 8-bit grayscale to contain 256 distinct gray levels and this image was then used to generate a weights matrix for fitness evaluation.



(a) $g = 1$ no ls

(b) $g = 100$ no ls

(c) $g = 1000$ no ls

(d) $g = 1$ DHC

(e) $g = 100$ DHC

(f) $g = 1000$ DHC

(g) $g = 1$ HLS

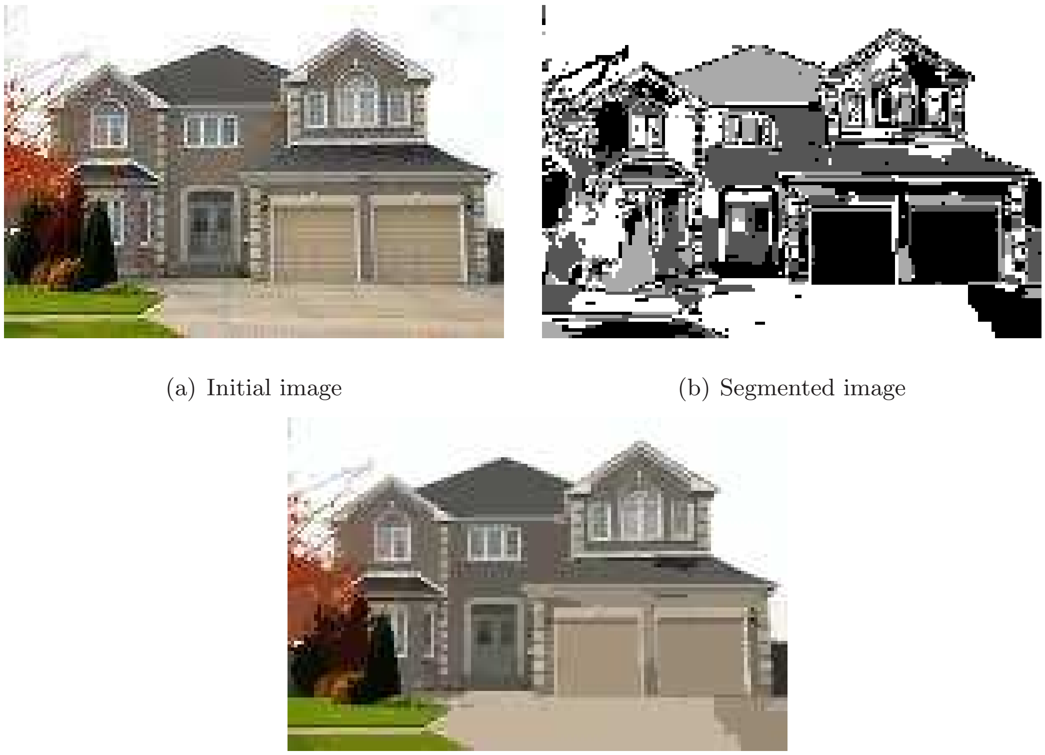
(h) $g = 100$ HLS

(i) $g = 1000$ HLS

Figure 5: Resulting best fitness segmentations on the 3 shapes image after runs of the GA to different numbers of generations using no local search, DHC and hierarchical local search (HLS).

4.2 Results

In this subsection, we will examine two different digital photographs and the resulting segmentations after running the hybrid GA on them. In addition, we will examine the segmented image after coloring the regions depending on the average color of that segment in the original image. Finally, we will examine the two images segmented using meanshift segmentation and compare the results to the hybrid GA segmentation.



(a) Initial image

(b) Segmented image

(c) Image with each segment identified by its average color

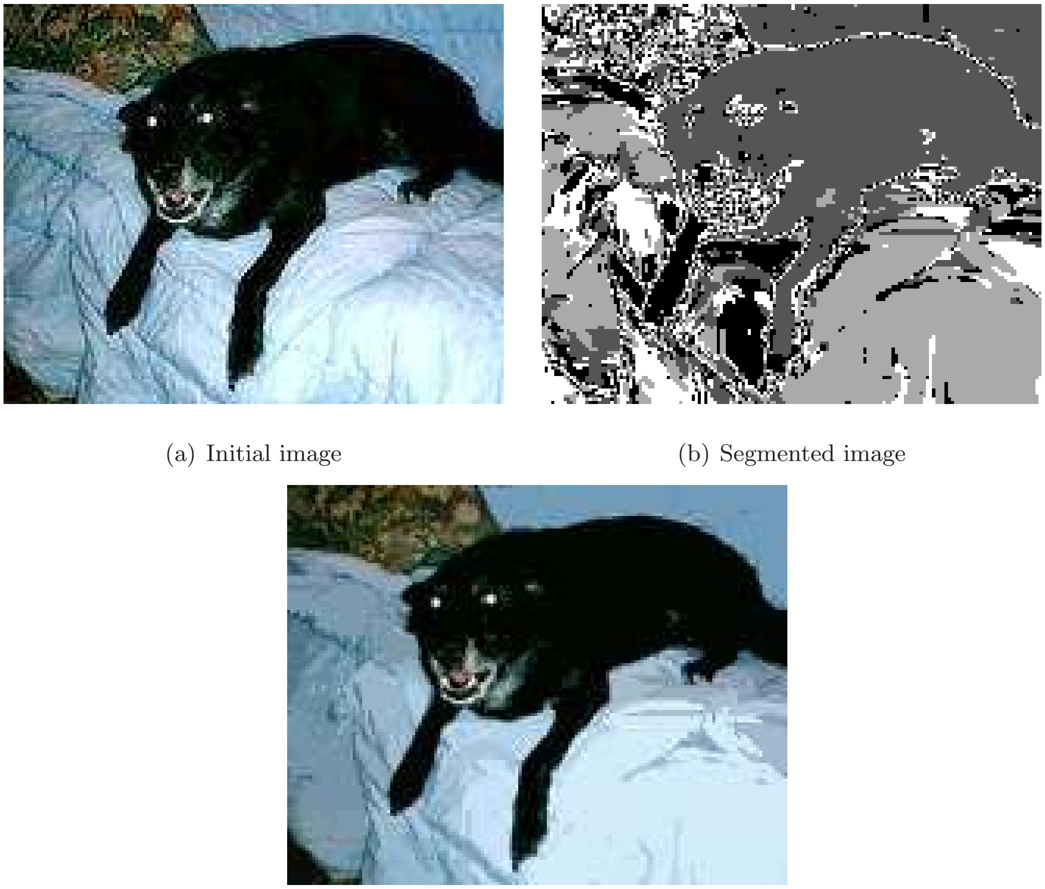
Figure 6: An image of a house, the resulting final segmentation using the hybrid GA, and then a segmented image with each segment colored by its average color in the original image.

Figure 6a shows a digital photograph of a house of size 150×100 pixels. The resulting segmented image found by the GA is shown in Figure 6b. In this image, we can see many distinct shapes corresponding to distinct regions in the original image. Finally, Figure 6c shows the segmented image with each segment colored by the average color of that segment in the original image. The average colored segmented image looks very similar to the original image except with much more clearly defined areas.

Figure 7 shows the digital image of a dog on a sofa of size 160 × 128 pixels. Figure 7b shows the resulting segmented image after running the GA. While this segmentation is not as clear as the results in the previous case, the averaged color image in Figure 7c shows more clearly defined edges than in the original image.

For comparison the two images were segmented using meanshift segmentation with the quantization-based method (Comaniciu & Meer, 1997), one of the current strongest segmentation algorithms. Figure 8a is intermediate result of the segmentation of the house image showing the borders between regions. Figure 8b is the final segmentation result with 17 colors. There is a strong similarity between the two segmentation results in Figure 7c and Figure 8b. Many large regions have been segmented in much the same way between algorithms. For example, the doors and driveway are easily identified in both, as well as different areas of the roof.

Figure 9 shows the results of meanshift segmentation on the dog on a couch image, with Figure 9a showing the region boundaries identified. Figure 9b shows the final results of the segmentation into

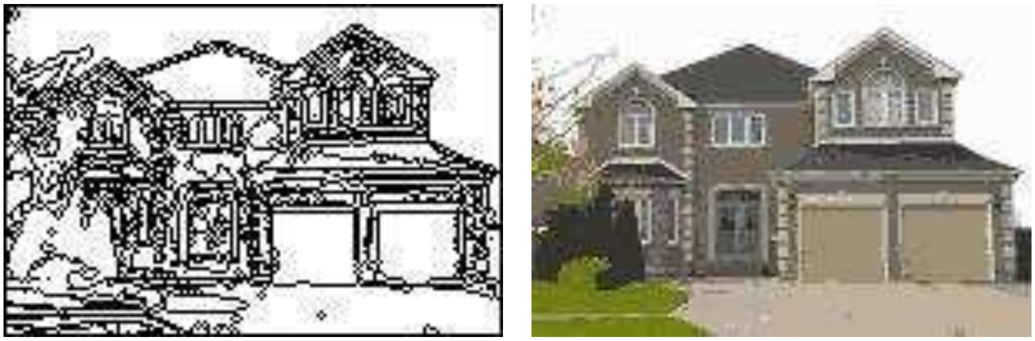


(a) Initial image

(b) Segmented image

(c) Image with each segment identified by its average color

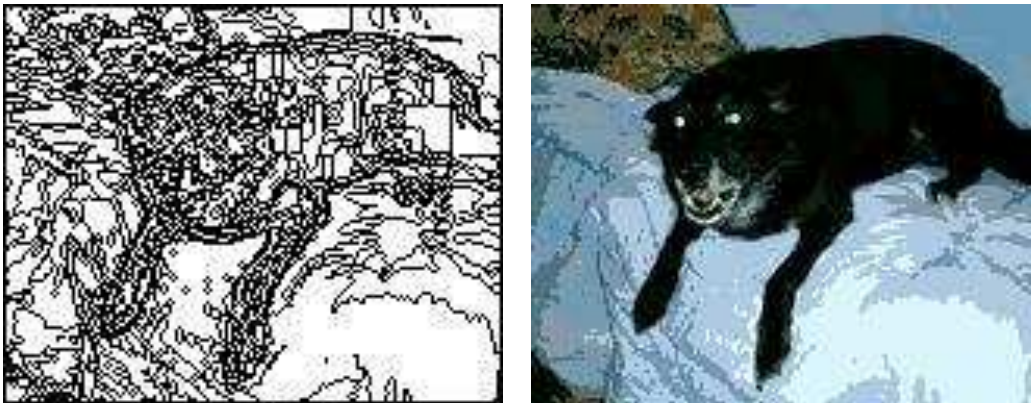
Figure 7: An image of a dog on a sofa, the resulting final segmentation using the hybrid GA, and a segmented image with each segment colored by its average color in the original image.



(a) Intermediate

(b) Final

Figure 8: Intermediate results and final results of meanshift segmentation on the house image segmented into 17 colors.



(a) Intermediate

(b) Final

Figure 9: Intermediate results and final results of meanshift segmentation on the dog on couch image segmented into 26 colors.

22 colors. One notable difference between the two algorithms can be seen in the intermediate results shown in Figure 7b and Figure 9a, with the dog divided into many more regions by the meanshift algorithm. Much as with the house image, the final results of segmentation are very similar, with major regions differentiated in both images.

5 Summary and Conclusions

This paper describes a hybrid GA to perform image segmentation on gray scale images. First, the images were transformed to a set of weights for the q-state Potts spin glass model, which can then be used to calculate the quality of an image segmentation of the original image. A hierarchical local search was used to improve solution quality upon each evaluation. A rectangular region crossover operator was used to generate new candidate segmentations, with the region transformed to ensure that the regions combined in the new candidate solution followed the same approximate region color scheme. The hybrid GA was then used to evolve candidate solutions of segmented images. The results show that the hybrid GA is able to efficiently generate segmented images of high quality.

This paper shows that while using a GA on images can be difficult due to the large number of variables and many local optima, it is still possible to overcome these problems using a powerful local search method such as the hierarchical local search used in this paper.

One of the key findings of this paper is the necessity of hierarchical local search on this type of problem. Since simple local search has often been shown to improve performance on many problems in GAs, a similar result was expected on image segmentation. However, the simple local search often led to a degradation of performance as monocolor regions were divided up into small regions instead of being fully assigned to one region. By moving to the hierarchical search that merged regions together, suddenly the local search resulted in dramatic improvements. This finding has implications beyond image segmentation, as simple local searches are used in many hybrid GAs and it is possible that they could benefit from a hierarchical local search. In particular, problems involving local search on images or spin glasses could be prime candidates for hierarchical local search.

There are several other key areas for future work. First, more powerful crossover operators should be examined and tested to see if they can improve the convergence time to a good solution. Then, different inhibition terms (Opara & Wörgötter, 1998; von Ferber & Wörgötter, 2000) could be used to emphasize contrast in the resulting segmented images. It is also important to study the effects of population size on the quality of the obtained image segmentation and the efficiency of the search.

Acknowledgments

This project was sponsored by the National Science Foundation under grants ECS-0547013 and IIS-1115352, and by the University of Missouri in St. Louis through the High Performance Computing Collaboratory sponsored by Information Technology Services. Most experiments were performed on the Beowulf cluster maintained by ITS at the University of Missouri in St. Louis and the HPC resources at the University of Missouri Bioinformatics Consortium. hBOA was developed at the Illinois Genetics Algorithm Laboratory at the University of Illinois at Urbana-Champaign. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bentrem, F. W. (2009, January 07). A Q-ising model application for linear-time image segmentation. *Central European Journal of Physics*, *8*, 689–698.

Binder, K., & Young, A. (1986). Spin-glasses: Experimental facts, theoretical concepts and open questions. *Rev. Mod. Phys.*, *58*, 801.

Comaniciu, D., & Meer, P. (1997, jun). Robust analysis of feature spaces: color image segmentation. In *CVPR '97: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* (pp. 750 –755).

Descombes, X., Moctezuma, M., Maître, H., & Rudant, J.-P. (1996, November). Coastline detection by a Markovian segmentation of SAR images. *Signal Process.*, *55*, 123–132.

Fischer, K., & Hertz, J. (1991). *Spin glasses.* Cambridge: Cambridge University Press.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Mezard, M., Parisi, G., & Virasoro, M. (1987). *Spin glass theory and beyond.* Singapore: World Scientific.

Neirotti, J. P., Kurcbart, S. M., & Caticha, N. (2003, Sep). Superparamagnetic segmentation by excitable neural systems. *Phys. Rev. E*, *68*, 031911.

Opara, R., & Wörgötter, F. (1998, August). A fast and robust cluster update algorithm for image segmentation in spin-lattice models without annealing – visual latencies revisited. *Neural Comput.*, *10*, 1547–1566.

Pelikan, M. (2005). *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms.* Springer-Verlag.

Peng, S., Urbanc, B., Cruc, L., Hyman, B. T., & Stanley, H. E. (2003). Neuron recognition by parallel potts segmentation. *Proceedings of the National Academy of Sciences*, *100*(7), 3847–3852.

Tanaka, K. (2002). Statistical-mechanical approach to image processing. *Journal of Physics A: Mathematical and General*, *35*(37), R81.

von Ferber, C., & Wörgötter, F. (2000, Aug). Cluster update algorithm and recognition. *Phys. Rev. E*, *62*, R1461–R1464.

Young, A. (Ed.) (1998). *Spin glasses and random fields*. Singapore: World Scientific.