

# A Swarm Art Based on Evolvable Boids with Genetic Programming

Tae Jong Choi and Chang Wook Ahn

Department of Computer Engineering, Sungkyunkwan University 2066, Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do, Republic of Korea

Email: {gry17, cwan}@skku.edu

**Abstract**—In this paper, we proposed a swarm art system. The proposed system receives an image and embellishes it to present an abstract and complex image. This system consists of a famous swarm behavior model, Boids and an evolutionary algorithm, Genetic Programming (GP). So, by combining the swarm behavior with the evolutionary principle, the proposed system may have a better-enhanced emergent property. Therefore, the proposed system can produce different and diverse images for each execution thanks to the emergent property. To best of our knowledge, this is the first approach to combine Boids model with GP to make a system that can produce vast abstract and complex art pieces from one input image. This system may be utilized in creating art as well as in a variety of entertainment area such as games, movies.

**Index Terms**—artificial life, evolutionary algorithm, swarm art, Boids, genetic programming

## I. INTRODUCTION

Generative art is computer based art that in some part of an artwork is designed by a non-human autonomous system. Here, an autonomous system usually refers to one that can independently create some characteristics of artwork. The important feature of Generative art is that it can produce abstract art pieces thanks to its dynamic and complex properties. In other words, Generative art can generate unpredictable art pieces that may expand human creativity.

Craig Reynolds introduced a simple yet elegant swarm behavior model called Boids [1-3]. Boids model contains a set of agents and each agent maneuver through simple behavior rules with the locations and velocities of its nearby agents. Although these behavior rules are fairly simple, Boids model is capable of displaying amazing and unpredictable group movements like living creatures. This property had a profound impact on not only computer scientists but also many different area practitioners including artists.

Since the beginning of the 21st century, artworks called Swarm art that refers to the combined Swarm intelligence and Generative art [4,5] have emerged. Swarm art is characterized by the use of the extended version of Boids model as its creative procedure. Art

pieces called Swarm art possess some interesting properties such as self-organization, communication, behavior, and interaction, that is, to say emergence [6]. The articles [4,5,7-15] introduced some famous Swarm art examples and Fig. 1 shows two of them [4,5].

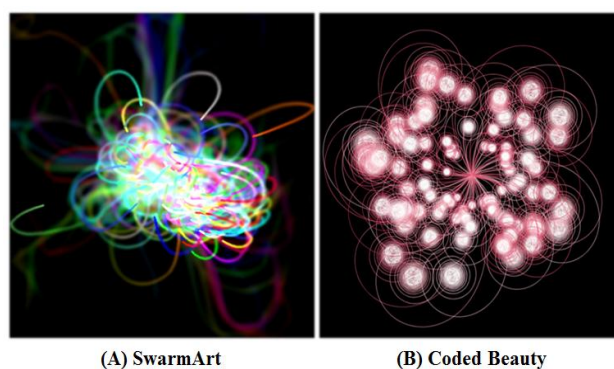


Figure 1. Two Swarm art examples: (A) SwarmArt (2007) [4] and (B) Coded Beauty (2008) [5]

In this paper, we proposed a swarm art system. The proposed system receives an image and embellishes it to provide an abstract and complex image. The proposed system is designed to use a set of evolvable Boids agents. Here, each agent has three different Genetic Programming (GP) trees, which represents an RGB color. So, each agent has its unique color and gets around the canvas environment that has the same size (width and height) as an input image. If an agent has a better color to represent the corresponding position of the input image than the color that environment has, the agent put a dot with its color to the position. The proposed system checks the agents every predetermined period, and if an agent is not satisfied with the number of left colors above a certain threshold, it is discarded from the population. After that, new agents are generated through the evolution of remained agents. As a result, by combining the swarm behavior with the evolutionary principle, the proposed system may have a better-enhanced emergent property, which makes it possible to produce different and diverse images for each execution. To best of our knowledge, this is the first approach to combine Boids model with Genetic Programming to make a system that can produce vast abstract and complex art pieces.

## II. RELATED WORK

### A. Boids Model

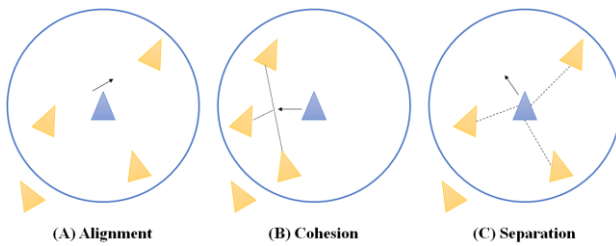


Figure 2. Three behavior rules of Boids model

Boids is an artificial life program that simulates swarm behavior such as bird flocking, animal herding, and fish schooling [1-3]. This model has an interesting property called emergence, which is that the complexity of the model occurs from the interaction of agents adhering to simple behavior rules [1-3]. These behavior rules that applied in agents are as follows:

- Alignment: each agent steers toward the average heading of its local agents
- Cohesion: each agent steers toward the average position of its local agents
- Separation: each agent steers to avoid crowding of its local agents

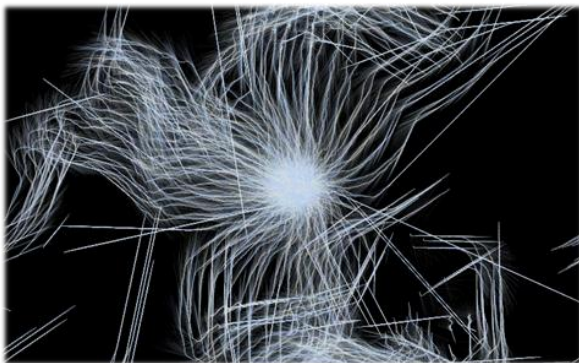


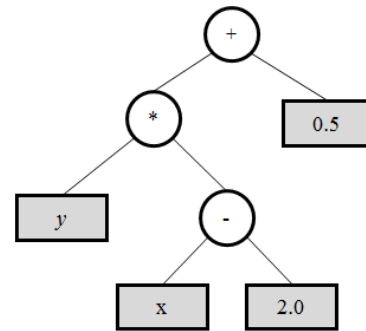
Figure 3. Swarm behavior of Boids model

More behavior rules such as avoiding obstacles and seeking a goal can be added. The stirring movement of Boids agents can be described as either chaotic or orderly [3]. Also, unpredictable behavior like separating and reorganizing flocks can be considered emergent behavior [3]. As a result, Boids model has been widely applied in computer graphics to represent swarm behavior such as bird flocking, animal herding, and fish schooling. Fig. 3 shows an example of Boids model.

### B. Genetic Programming

GP [16-19] is an attempt to deal with a question that how a computer can learn to solve problems without being explicitly programmed [20]. GP is an extension of an evolutionary algorithm called Genetic Algorithm (GA). While the expression of the genotype of GA is an array, GP uses a tree structure genotype. Thus, GP can express a formula or a program that GA may not. Fig. 4 shows how

the expression of the genotype of GP can represent a formula.



$$A(x, y) = y * (x - 2.0) + 0.5$$

Figure 4. A formula represented by GP

GP consists of three major operators, selection, crossover, and mutation. The flow of GP is similar to GA. The four steps that GP proceed to solve problems are as follows:

1. Generate an initial population
2. Execute each individual and evaluate its fitness
3. Create a new population
  - A. Select a randomly selected parent individual based on fitness and copy it
  - B. Create a child individual by crossover with two randomly selected parent individuals based on fitness
  - C. Create a child individual by mutation with a randomly selected parent individual based on fitness
4. If the best-so-far GP program satisfies a termination criterion, return the program. Otherwise, go back to step 3.

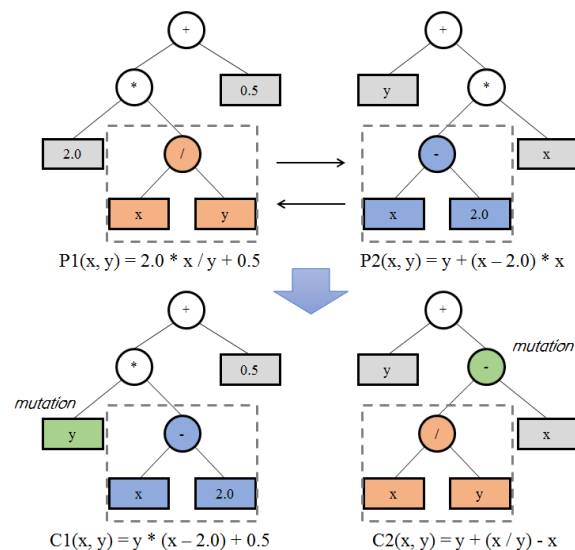


Figure 5. The crossover and mutation results of GP

In GP, crossover and mutation operators are just as important as GA because these operators act to increase the diversity of the evolutionary algorithms. Since the representation of the genotype of GP is a tree structure, the methods for crossover and mutation slightly differ

from GA. The crossover of GP is applied to an individual by exchanging one of its nodes with another node from another individual. In a tree structure like GP, to switch a node means that to switch a branch. So, crossover results are quite different from their parent individuals. Also, the mutation of GP is applied to an individual by replacing one of its nodes with a new node or a new branch. Fig. 5 shows an example of the crossover and mutation results of GP.

To sum up, GP is a powerful evolutionary algorithm at solving any problems that can be tractable to machine learning and has been successful at evolving new and unpredictable ways of solving problems [21].

### III. THE PROPOSED SYSTEM

In this section, we introduce the proposed Swarm art system. The proposed system is designed to develop a method that can provide abstractly and complexly modified images in each execution with a given input image. We designed the proposed system with following key properties:

- An image drawing method with different patterns in each execution
- A method of embellishing a given art piece in abstract and complex ways

To embed these properties, we designed the proposed system to use a set of evolvable Boids agents. Evolvable Boids is a combination of the swarm behavior with the evolutionary principle, which moves around the environment like a Boids agent and leaves its unique color that generated by its GP programs. Because crossover and mutation operators can evolve the color of a Boids agent, we named it evolvable Boids.

The proposed system uses Boids model to draw images with different patterns in each execution. As we described in Section II, Boids model can show unpredictable group behavior based on simple behavior rules. Therefore, by using Boids model, the proposed system can draw images with the emergence property. From Algorithm 1 to Algorithm 4 shows our implementation of Boids model based on articles [1-3].

#### Algorithm 1. Boids model: Flock

---

```

Algorithm flock()
  For each timeStep Do
    Vector vel (0, 0)
    For each boid in Boids Do
      vel.add (alignment (boid))
      vel.add (cohesion (boid))
      vel.add (separation (boid))
    End
    For each boid in Boids Do
      If vel >= MAX_SPEED Then
        vel = MAX_SPEED
      End
      boid.location.add (vel)
      draw (boid)
    End
  End

```

---

#### Algorithm 2. Boids model: Alignment

---

```

Algorithm alignment (boid)
  neighborRange = 50
  Vector sum (0, 0)
  count = 0
  For each neighbor in Boids Do
    d = boid.location - neighbor.location
    If d > 0 and d < neighborRange Then
      sum.add (neighbor.velocity)
      count++
    End
  End
  Vector steer (0, 0)
  If count > 0 Then
    sum.mult (MAX_SPEED / count)
    steer = sum - boids.velocity ()
    steer.limit (MAX_FORCE)
  End
  Return steer

```

---

#### Algorithm 3. Boids model: Cohesion

---

```

Algorithm cohesion (boid)
  neighborRange = 50
  Vector sum (0, 0)
  count = 0
  For each neighbor in Boids Do
    d = boid.location - neighbor.location
    If d > 0 and d < neighborRange Then
      sum.add (neighbor.location)
      count++
    End
  End
  Vector steer (0, 0)
  If count > 0 Then
    sum.div (count)
    steer = seek (sum)
  End
  Return steer

```

---

#### Algorithm 4. Boids model: Separation

---

```

Algorithm separation (boid)
  neighborRange = 25
  Vector steer (0, 0)
  count = 0
  For each neighbor in Boids Do
    d = boid.location - neighbor.location
    If d > 0 and d < neighborRange Then
      Vector diff = boid.location - neighbor.location
      diff.normalize ()
      diff.div (d)
      steer.add (diff)
      count++
    End
  End
  If count > 0 Then
    steer.div (count)
  End
  If steer.length () > 0 Then
    steer.normalize ()
    steer.mult (MAX_SPEED)
    steer.sub (boid.velocity)
    steer.limit (MAX_FORCE)
  End
  Return steer

```

---

GP is usually applied to solve some optimization problems such as classification or regression, but there is some work that applied it to create artistic and interesting images which are quite difficult to be made by human beings [22-28]. In the proposed system, each agent has three different GP trees, which represents an RGB color. So, each agent has its unique color and gets around the canvas environment that has the same size (width and height) as an input image. If an agent has a better color to represent the corresponding position of the input image than the color that environment has, the agent put a dot with its color to the position. Fig. 6 shows the example of two evolvable Boids agents on the proposed system.

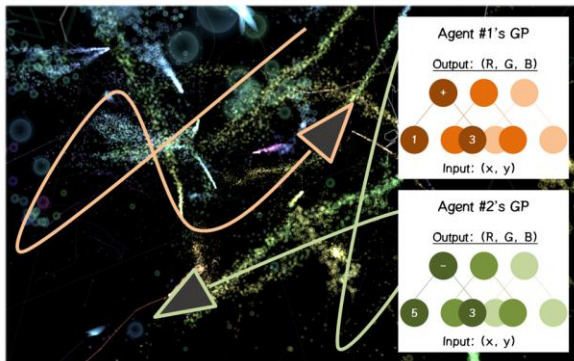


Figure 6. The proposed system: Boids agent with GP

The terminal nodes of GP are  $\{X, Y, R\}$ , where  $X$  and  $Y$  are the locations of an agent and  $R$  is a random real number. The function nodes of GP are as follows:

1. Bit-wise AND:  $X \& Y$  (or  $R$ )
2. Bit-wise OR:  $X | Y$  (or  $R$ )
3. Bit-wise XOR:  $X \wedge Y$  (or  $R$ )
4. NOT:  $\neg X$
5. Left-shift:  $X \ll 2$  (multiplied by 2)
6. Right-shift:  $X \gg 2$  (divided by 2)
7. ADD:  $X + Y$  (or  $R$ )
8. SUB:  $X - Y$  (or  $R$ )
9. MUL:  $X * Y$  (or  $R$ )
10. DIV:  $X / Y$  (or  $R$ ) if  $Y > 0.0001$
11. SIN:  $\sin(X) * 255$
12. COS:  $\cos(X) * 255$
13. TAN:  $\tan(X) * 255$

The proposed system evaluates Boids agents every predetermined period. If an agent is not satisfied with the number of left colors above a certain threshold, it is discarded from the population. Here, each Boids agents moves for every 0.001 seconds, and the evaluation procedure occurs for every second. Also, we choose the threshold as 150. So, if a Boids agent leaves its color above 150, it will be survived, and otherwise, removed. After that, new agents are generated through the evolution of remained agents. Each survived agent produce one child agent by using the mutation operator of GP. So the child agent will possess slightly different RGB trees and be placed in the same location of its parent. This process is repeated until there is no agents exist in the canvas. As we described earlier, art pieces based on GP can show abstract and complex patterns. Therefore, by using GP with Boids model, the proposed system can

embellish a given input image to produce interesting images. Also, the modifying image process of the system can also be interesting to users.

As a result, the proposed system can produce emergently modified images and display their processes on a given input image.

#### IV. RESULTS

In this section, we introduce a set of the results of the proposed system. Fig. 7 to Fig. 10 are the first result of the proposed system. Fig. 7 shows the original image and from Fig. 8 and Fig. 9 show some of the results. And Fig. 10 shows a set of results.



Figure 7. The original image



Figure 8. One of the results of Figure 7 #1

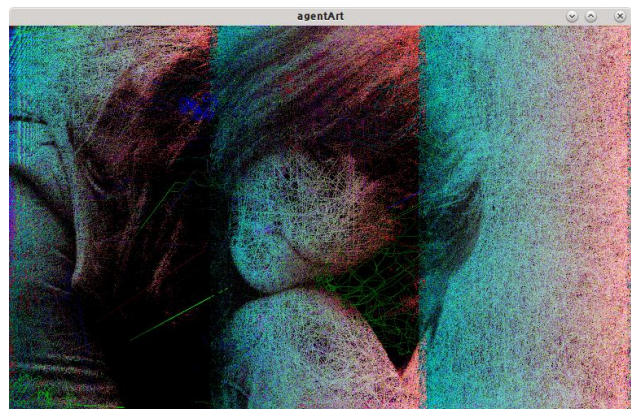


Figure 9. One of the results of Figure 7 #2

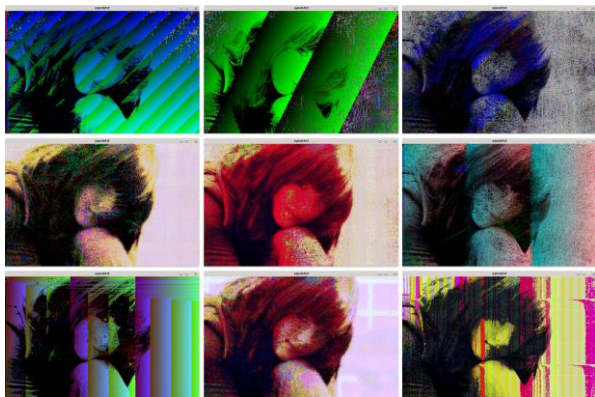


Figure 10. A set of the proposed system results of Figure 7

As we can see, the proposed system can embellish a given image in abstract and complex ways in each execution. Fig. 11 to Fig. 12 are the second result of the proposed system. Fig. 11 shows the original image and Fig. 12 shows the progress of creating the art piece of the proposed system.



Figure 11. The original image

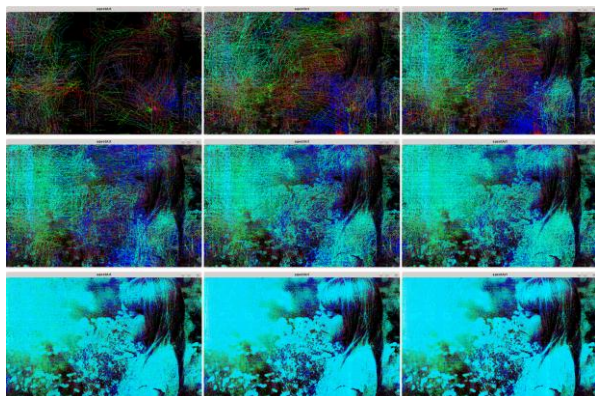


Figure 12. The creating image process of the proposed system

As we can see, the proposed system can draw images in swarm behavior thanks to the emergent property. Therefore, the processing of creating an art piece is also may present artistic and interesting to users.

## V. CONCLUSION

In this paper, we proposed a system for generating abstract and complex images based on the given input image. The proposed system can produce artistic and

complex art pieces, and the progress of creating an image is also dynamic and interesting to users. The proposed system can be described as a fusion of Swarm Intelligence and Generative Art. We introduced evolvable Boids model that is the combination of Boids model and GP. Boids is an artificial life program that simulates swarm behavior, and this model has an interesting property called emergence, which is that the complexity occurs from the interaction of agents adhering to simple behavior rules. GP is an extension of GA and can produce artistic and interesting images as well as to solve some optimization problems. The proposed system is designed to maximize the emergent feature by using these approaches, and hence, can take advantage of these systems to create art pieces. The proposed system may be utilized Art as well as in a variety of entertainment such as games, movies.

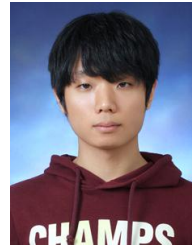
## ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2015R1D1A1A02062017).

## REFERENCES

- [1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 25-34, 1987, August, ACM.
- [2] C. Reynolds, (2001). Boids: Background and update. URL: <www.red3d.com/cwr/boids>. (Accessed 1 February 2016).
- [3] R Wikipedia contributors. (n.d.). Boids. URL: <en.wikipedia.org/wiki/Boids>. (Accessed 1 February 2016).
- [4] C. J. Jacob, G. Hushlak, J. E. Boyd, P. Nuytten, M. Sayles, and M. P. Swarmart, "Interactive art from swarm intelligence. Leonardo," vol. 40, no. 3, pp. 248-254, 2007.
- [5] T. Kräftner, "Coded beauty-die generative methode als ästhetische praxis der computergrafik," PhD thesis, FH Joanneum - Fachhochschulstudiengang Informations-Design, 2008.
- [6] G. Greenfield and P. Machado, *Swarm Art*, Leonardo, vol. 47, no.1, pp. 5-7, 2014.
- [7] S. Aupetit, V. Bordeau, N. Monmarché M. Slimane, and G. Venturini, "Interactive evolution of ant paintings," in *Proc. The 2003 Congress on Evolutionary Computation*, vol. 2, 2003, December, pp. 1376-1383.
- [8] G. Greenfield, "Evolutionary methods for ant colony paintings," in *Proc. Applications of Evolutionary Computing*, Springer Berlin Heidelberg, 2005, pp. 478-487.
- [9] N. Monmarché I. Mahnich, M. and Slimane, "Artificial art made by artificial ants," in *The Art of Artificial Evolution*, Springer Berlin Heidelberg, 2008, pp. 227-247.
- [10] S. Schlechtweg, T. Germer, and T. Strothotte, "Renderbots-multi-agent systems for direct image generation," in *Computer Graphics Forum*, vol. 24, no. 2, pp. 137-148, Blackwell Publishing Ltd, 2005, June.
- [11] A. Dorin, "The virtual ecosystem as generative electronic art," in *Proc. EvoWorkshops*, pp. 467-476.
- [12] A. Dorin, "A survey of virtual ecosystems in generative electronic art," in *The Art of Artificial Evolution*, Springer Berlin Heidelberg, 2008, pp. 289-309.
- [13] J. McCormack, "Artificial ecosystems for creative discovery," in *Proc. The 9th Annual Conference on Genetic And Evolutionary Computation*, ACM, pp. 301-307, 2007, July.
- [14] J. McCormack, "The evolution of sonic ecosystems.,," in *Artificial Life Models in Software*, Springer London, pp. 393-414, 2009.
- [15] J. McCormack, "Enhancing creativity with niche construction," in *ALIFE*, pp. 525-532, 2010.
- [16] J. R. Koza, "Genetic programming: On the programming of computers by means of natural selection," vol. 1, MIT Press, 1992.

- [17] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Subprograms*. Cambridge, MA, USA, 1994.
- [18] J. R. Koza, "Genetic programming iii: Darwinian invention and problem solving," vol. 3, Morgan Kaufmann, 1999.
- [19] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza, "Genetic programming IV: Routine human-competitive machine intelligence," vol. 5, *Springer Science & Business Media*, 2006.
- [20] J. R. Koza, "Survey of genetic algorithms and genetic programming," in *Proc. Wescon Conference Record*, Western Periodicals Company, pp. 589-594, 1995, November.
- [21] R. Poli, W. B. Langdon, N. F. McPhee, and J. R. Koza, *A Field Guide to Genetic Programming*, Lulu. Com, 2008.
- [22] S. Das, T. Frangiadakis, M. E. Papka, T. A. DeFanti, and D. J. Sandin, "A Genetic programming application in virtual reality," *International Conference on Evolutionary Computation*, pp. 480-484.
- [23] E. J. Bedwell and D. S. Ebert, "Artificial evolution of algebraic surfaces," in *Proc. Implicit Surfaces '99*.
- [24] M. W. Jones, "Direct surface rendering of general and genetically bred implicit surfaces," in *Proc. 17th Ann. Conf. of Eurographics* (UK Chapter) pp. 37-46.
- [25] U. M. O'Reilly and G. Ramachandran, "A preliminary investigation of evolution as a form design strategy," in *Artificial Life VI*, pp. 443-447, Los Angeles, CA, UAS.
- [26] P. Testa, U. M. O'Reilly, and S. Greenwold, "Agent-based genetic programming for spatial exploration," in *Proc. ACSA*.
- [27] M. Hemberg, U. M. O'Reilly, and P. Nordin, "GENR8-A design tool for surface generation," *GECCO Late Breaking Papers*, pp. 160-167, San Francisco, 9-11.
- [28] T. Unemi, "Embedding movie into SBART-breeding deformed movies," in *Proc. 2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 6, pp. 5760-5764.



**Tae Jong Choi** received M.S from the Department of Computer Engineering, Ajou University, Republic of Korea. He is currently working toward the Ph.D. degree at the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. His research interests include machine learning, evolutionary algorithms, and the applications of machine learning and evolutionary techniques.



**Chang Wook Ahn** received the Ph.D. degree from the Department of Information and Communications, Gwang-Ju Institute of Science & Technology (GIST), Republic of Korea. From 2007 to 2008, he was a Research Professor in GIST. He is currently an Assistant Professor at Sungkyunkwan University (SKKU), Republic of Korea. His research interests include genetic algorithms, multiobjective optimization, neural networks, and the applications of evolutionary techniques to wireless networks.