Randomized Population with Taguchi's Method for Multi-objective Optimization

[1]Cheng-Yuan Tang, [2]Yi-Leh Wu, [1]Chien-Chin Peng,
[1]Chun-Chan Lin, [3]Chia-Chen Chen, [3]Hsien-Chang Lin

[1] *Department of Information Management, Huafan University, Taipei, Taiwan, ROC*
[2] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan*
[3] *Electronics and Optoelectronics Research Laboratories, Hsinchu, Taiwan*
e-mail: cytang@cc.hfu.edu.tw

ABSTRACT

Genetic algorithms can be divided into two categories: single objective and multiple objectives. With single objective, we introduce two modified genetic algorithms: the orthogonal genetic algorithm with quantization (OGA/Q), which utilizes the orthogonal design and quantization technique, and the Hybrid Taguchi Genetic Algorithm (HTGA), which utilizes the Taguchi's method. Because of the multiple objective functions, the design of the multi-objective genetic algorithms focuses on the fitness assignment, the diversity preservation, and the addition of an elite set.

In this paper, we propose to include an additional random population besides the original initial population. In each generation we replace the random population and select only the non-dominated individuals into the elite set. The proposed method can explore more general solution space and can locate better solutions. We then apply Taguchi's method to generate better individuals in the additional random population.[1]

1. INTRODUCTION

When the number of variables increases, optimization problems become extremely complex. In the past, people are at a loss how to conquer this kind of problems. With the introduction of the computers, people now can develop effective algorithms and perform simulations on the computers thus make the answer of optimization problems more feasible. For this reason, a number of stochastic search strategies such as evolutionary algorithms, table search, simulated annealing, and ant colony optimization have been developed [14].

Generally speaking, a stochastic search algorithm consists of three parts: 1) a working memory that contains the currently considered solution candidates, 2) a selection module, and 3) a variation module as depicted in Figure 1 [14].

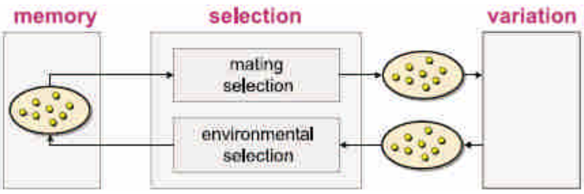

Figure 1. Components of a general stochastic search algorithm.

As to selection, one can distinguish between mating and environmental selection. Mating selection aims at picking promising solutions for variation and usually is performed in a randomized fashion. In contrast, environmental selection determines which of the previously stored solutions and the newly created ones are kept in the internal memory. The variation module takes a set of solutions and systematically or randomly modifies these solutions to generate potentially better solutions.

We can divide the evolutionary algorithms into single-objective evolutionary algorithms [4][5][8][10][11][15][18] and multi-objective evolutionary algorithms [6][13][14][16][17] by their number of objective functions. In single-objective evolutionary algorithms, the objective function is the fitness function. Because the number of objective functions in multi-objective evolutionary algorithms is equal or more than two, the fitness functions must have some special arrangements.

2. ORTHOGONAL ARRAY

In the whole factor design, when the number of factors increases, the required number of experiments will increase thereupon. Taguchi's method utilizes orthogonal array to collect the materials directly, make us obtain more reliable factor result estimator with less experiments. It is an important skill that a robust design utilizes the orthogonal array and carries on the experiments directly.

A general orthogonal array is defined as follows:

[1] This work was supported in part by the National Science Council, Republic of China under Contract No. 94-2213-E-211-012 and 95-2221-E-011-059.

$L_a(b^c)$
Where
a : number of experimental runs;
b : number of levels for each factor;
c : number of columns in the orthogonal array.

An example $L_8(2^4)$ orthogonal array is shown in Table 1.

Table 1 $L_8(2^4)$ Orthogonal Array

Run \ Factor	A	B	C	D
1	**1**	**1**	**1**	**1**
2	**1**	**1**	**2**	**2**
3	**1**	**2**	**1**	**2**
4	**1**	**2**	**2**	**1**
5	**2**	**1**	**1**	**2**
6	**2**	**1**	**2**	**1**
7	**2**	**2**	**1**	**1**
8	**2**	**2**	**2**	**2**

2.1. TAGUCHI'S METHOD

Although Taguchi's parameter design method is seldom applied in the field of computer science, it is an important tool for robust design. Taguchi's method is often regarded as an engineering methodology to optimize products and process conditions which are minimally sensitive to the causes of variations, and which produce high-quality products with low development and manufacturing costs. Orthogonal array and the SNR (Signal-to-Noise Ratio) are two major tools used in Taguchi's method.

SNR in Taguchi's method is used to assess each level in the contribution degree to the object function of each factor. Formulation of the SNR is derived from the unbiasedness in statistics. It is an estimate of how samples deviate from the center of population. The general formulation of the SNR is as follows:

$$(\bar{y}-m)^2 - S^2$$

Where $\bar{y}$ the mean of sample
m the mean of object
S the standard deviation of sample

$$SNR = -10\log\left[(\bar{y}-m)^2 - S^2\right] = -10\log\left[\frac{\sum_{i=1}^{n}(y_i - m)^2}{n}\right]$$

3. THE INTELLIGENT MULTIOBJECTIVE EVOLUTIONARY ALGORITHM (IMOEA)

The IMOEA [6] was proposed by Ho et al. and its purpose is to optimize multiple objective functions simultaneously in order to achieve the optimal solution. In the IMOEA, the Intelligent Gene Collector (IGC) is a main phase and the generalized pareto-based scale-independent fitness function (GPSIFF) is the fitness assignment strategy.

.

3.1 The IGC

The IGC[6] uses a divide-and-conquer approach, which consists of three parts: 1. the dividing part: divide large chromosomes into an adaptive number of gene segments; 2. the conquering part: identify potentially good gene segments such that each gene segment can potentially be part of an optimal solution; and 3. the combination part: combine the potentially better gene segments of their parents to produce a potentially good approximation to the best one of all combinations of gene segments.

3.2 The Fitness function GPSIFF

The fitness assignment strategy is an important issue in solving multi-objective optimization problems. The IMOEA employs the generalized pareto-based scale-independent fitness function (GPSIFF) to quantify the fitness performances in the objective space for both dominated and non-dominated individuals. Let the fitness value of an individual X be a score obtained from all participated individuals by the following function:

$$\mathrm{GPSIFF}(X) = p - q + c$$

where p is the number of individuals which can be dominated by X, q is the number of individuals which can dominate X in the objective space, and c is a constant. Following is an example of the dominate relationship:

$$\left(\mathbf{X}_1 = \begin{Bmatrix}5\\3\\2\\6\end{Bmatrix}\right) \prec \left(\mathbf{X}_2 = \begin{Bmatrix}6\\4\\3\\7\end{Bmatrix}\right), \quad \left(\mathbf{X}_3 = \begin{Bmatrix}7\\5\\3\\5\end{Bmatrix}\right) \prec \left(\mathbf{X}_4 = \begin{Bmatrix}7\\5\\3\\7\end{Bmatrix}\right) \quad \text{dimension is 4}$$

The definition of dominate is that $\mathbf{X}_i$ must be greater than or equal to all $\mathbf{X}_j$, with at least one $\mathbf{X}_{ik}$ greater than one $\mathbf{X}_{jk}$. The dominate relationship does not exist in the following example:

$$\left(\mathbf{X}_5 = \begin{Bmatrix}5\\3\\2\\6\end{Bmatrix}\right) \text{non - dominate} \left(\mathbf{X}_6 = \begin{Bmatrix}5\\3\\2\\6\end{Bmatrix}\right), \left(\mathbf{X}_7 = \begin{Bmatrix}7\\5\\3\\5\end{Bmatrix}\right) \text{non - dominate} \left(\mathbf{X}_8 = \begin{Bmatrix}8\\2\\3\\5\end{Bmatrix}\right) \quad \text{dimension is 4}$$

4. THE FUNDAMENTAL MATRIX AND THE EIGHT-POINT ALGOTITHM

Given two images in 3D computer vision systems, to establish a general relationship between the two sets of image coordinates which expresses the constraints that the corresponding rays through the two camera centers must intersect in space [1]. When the intrinsic parameters of the cameras are known, the epipolar constraint can be represented algebraically by a 3x3 matrix, called the essential matrix or the fundamental matrix, F. One can compute the fundamental matrix ***F*** with the eight-point algorithm stated below.

4.1 The Eight-Point Algorithm

The eight-point algorithm is linear; hence the fundamental matrix ***F*** ca be computed fast and easily. The eight-point algorithm for computing the essential matrix was introduced by Longuet-Higgins [1]. The coefficients in the fundamental matrix are in a nine-vector which constitutes a linear system.

$$(u,v,1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad (4\text{-}1)$$

. Because the coefficients of the fundamental matrix is homogeneous, we therefore decide F_{33}=1 and let it not zero. Using the eight-points algorithm $(p_i, p_i'), i = 1,\cdots,8$ make equation (4-1) to rewrite the 8×8 linear equation:

$$\begin{bmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' \end{bmatrix}\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{bmatrix} = -\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The great advantage of the eight-point algorithm is that it is linear. If the eight pairs of corresponding points are known, we can compute the parameters of the fundamental matrix. With more than eight points, a linear least squares minimization problem must be solved.

4.2 The Three-Axis Bucket

Traditional bucketized algorithms divide the plane into some non-overlapping areas, and then select one point from each area. We propose to use a 3-axis bucket to select points which is distributed over the domain. But in reality, we do not have information about the third axis. When we obtain the 3D models of scenes from two images, the disparity is larger when object is closer to the camera. So we assume that the disparity can be the third axis in our bucket selection scheme. .

Given the coordinates of a pair of correspondence points are (x_1, y_1) and (x_2, y_2), we have three ways to compute the disparity of correspondence points, as describes below:

1. $\begin{pmatrix} x_1 - x_2 \\ y_1 - y_2 \end{pmatrix}$
2. $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
3. When the Y-axis values of the correspondence points are the same, we use just $(x_1 - x_2)$.

In this paper, we use the second method above to compute the disparity of correspondence points.

The method of selecting correspondence points:

1. Compute the disparity (d) of correspondence points.
2. Find the largest and smallest value on X-axis and Y-axis and 1/d of correspondence points, and equally divide them into Q parts.
3. Use the orthogonal array $L_{Q^2}(Q^3)$ $Q \geq 2$ we select the Q^2 orthogonal areas.
4. Select eight areas from the Q^2 orthogonal areas and then select one point form each area to form a chromosome.

For example, we first compute the disparity of all correspondence points with $Q = 3$. We can then divide all correspondence points into 27 areas as shown in Figure 2 and Figure 3. Figure 2 depicts the original date points and Figure 3 depicts the points with added third axis to form the 3-axis bucket.

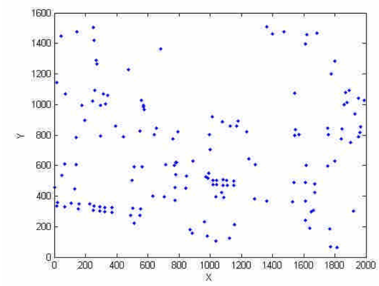

Figure 2. The 2D coordinates of correspondence points.

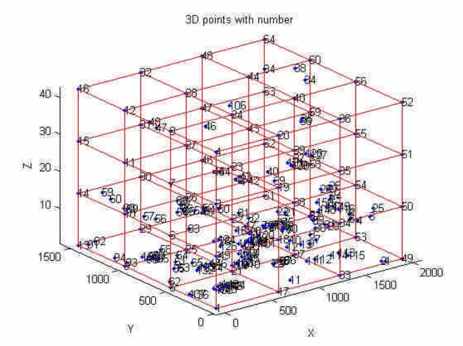

Figure 3. The 3D coordinates of correspondence points in the 3-axis bucket.

5. THE ADDITION OF RANDOM POPULATION

After the initial population is generated in IMOEA, all individuals' genes are fixed if there is no mutation operation. When the search space is fixed, we can only find local optima. We propose a new method which includes a randomized population with all individuals is regenerated again with generations. We only include non-dominated individuals in our randomized population to the elite set and exclude other individuals to participate in later evolutions. The proposed method which includes a randomized population can search more feasible solution space and can include better individuals than the ones in the initial population.

In The Large Parameter Optimization Problems (LPOPs), individuals in the randomized population are generated in one of the following two ways: 1. Randomly generated individuals: We randomly generate a random population of K individuals, and it is the same way as generating the initial population. 2. Orthogonal quantization generated individuals: It employs the quantization method that originated from the OGA/Q to quantize the domain of every gene, so the individuals can be distributed over the entire domain. We then use Taguchi's method to all individuals that are generated by method 1 or 2. Those result individuals are put in the randomized population.

In the fundamental matrix optimization problems, individuals in the randomized population are generated in two ways: 1. randomly generates individuals and 2. use 3-axis bucket to generate individuals. We then employ Taguchi's method on the result individuals. Those chromosomes that are selected by the above operations have to go through the following operations, as shown below, before include them in the randomized population.

1. Randomly select two chromosomes that are generated by the 3-axis bucket and put them into Taguchi' s method.
2. Pu the individuals generated by 1 into the randomized population.
3. Repeat 1 and 2, until the desired number of individuals in the population is met.

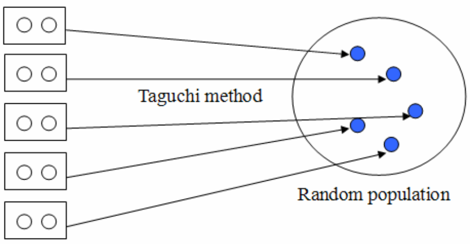

Figure 4. Producing individuals in randomized population.

The proposed modification of IMOEA with additional randomized population is described as follows:

1. Initialization: randomly generate an initial population of N_{pop} individuals and create an empty elite set e and an empty temporary elite set e'.
2. Randomized population: use the method proposed in section 7.1 to generate K individuals of random population.
3. Evaluation: compute two distances (geometric distance and algebraic distance) and transform them into the fitness values of each individual in the populations. Assign each individual a fitness value by using GPSIFF.
4. Update elite sets: add the non-dominated individuals in both the population, random population and e' to e and empty e'. Considering all individuals in e, then remove the dominated ones. If the number N_E of non-dominated individuals in e is greater than N_{Emax}, then randomly discard excess individuals
5. Selection: Select N_{pop}-N_{ps} individuals from the population using the binary tournament selection and randomly select N_{ps} individuals from e to form a new population, where $N_{ps} = N_{pop} \cdot P_s$. If $N_{ps} > N_{ps}$, let $N_{ps} = N_e$.
6. Recombination: perform the IGC operations for $N_{pop} \cdot p_c$ selected parents. For each IGC operation, add non-dominated individuals derived from by-products OA combinations (by-products) and two children to e'.
7. Mutation: apply the conventional mutation operation with P_M to the population.
8. Termination test: if a stopping condition is satisfied, stop the algorithm. Otherwise, go to Step 2).

6. EXPERIMENTS

In this section, we perform experiments that employ the additional random population on top of the IMOEA to solve the Large Parameter Optimization Problems (LPOP) and the estimation of fundamental matrix problems.

6.1 The Large Parameter Optimization Problems

While single objective genetic problems can optimize a single function, the multi-objective genetic problem can optimize two or more objective functions in parallel. Our first experiment try to minimize $\sum_{i=1}^{D}(x_i+0.5)^2$ and $\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)+10\right]$ test functions together [6], where $-5.12 \le x_i \le 5.12$ and D=8.

Table 2 LPOP experimental settings.

algorithm	IMOEA	IMOEA with Orthogonal random population	IMOEA with Random population
Size of population	200	200	200
# of generations	1000	1000	1000
Pc	0.8	0.8	0.8
P_M	0.02	0.02	0.02

Table 3 LPOP experiment results.

algorithm	IMOEA	IMOEA with Orthogonal random population	IMOEA with Random population
time (in sec)	6554	9070	8869

The non-dominated individuals in all experiments are depicted in Figure 5 and Figure 6.

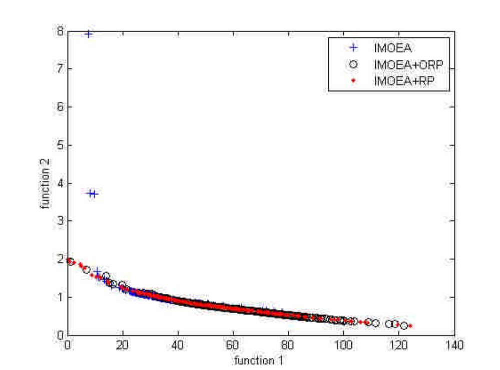

Figure 5. IMOEA(+)、IMOEA+rand. pop.(.) and IMOEA+orthogonal rand. pop.(o) in LPOP.

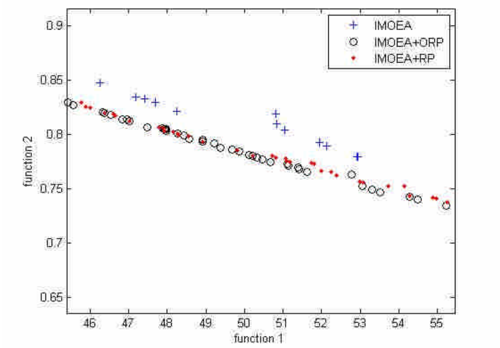

Figure 6. A zoom in picture of Figure 5.

In the LOPO experiments, the original IMOEA, the IMOEA with random population and the IMOEA with orthogonal random population methods can all minimize the two objective functions in parallel. In Figure 6, we show that by adding the randomized population we can find better solutions than the original IMOEA method. And the IMOEA with orthogonal random population outperforms the IMOEA with orthogonal random population method.

6.2 The Fundamental Matrix Optimization Problems

We employ the eight-point algorithm to estimate the fundamental matrices with two objective functions. The first function is the geometric distance that measures the distance of all corresponding points to their epipolar lines in image. The second function is the algebraic distance of all corresponding points in the fundamental matrix. We want to minimize both functions and together we quantize the performance of the estimated fundamental matrices. In this experiment, Function 1 (F1) is geometric distance and Function 2 (F2) is the algebraic distance. We take the 884 corresponding point pairs (depicted in red as shown in Figure 7) as our experiment data. Table 4 shows the experimental settings.

Figure 7. Corresponding Points.

Table 4. Experimental settings (fundamental matrix estimation).

algorithm	IMOEA with 2D bucket	IMOEA with Random population	IMOEA with 3-axis bucket and random population
Size of population	200	200	200
# of generations	500	500	500
Pc	0.8	0.8	0.8
PM	0.02	0.02	0.02
# test data points	884	884	884

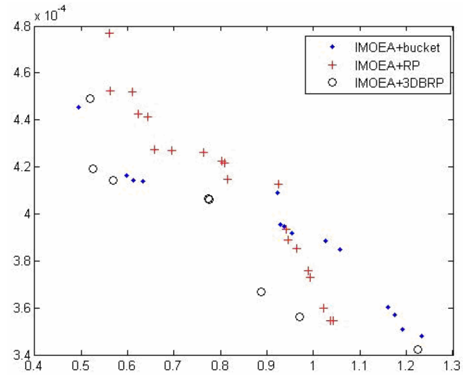

Figure 8. IMOEA+bucket(.)、IMOEA+ rand. pop.(+) and IMOEA+ 3-axis bucket random population (o) – fundamental matrix estimation results.

As shown in Figure 8, IMOEA with the randomized population can find better solutions than the original IMOEA and sometimes even outperforms the IMOEA with bucketization. The domain quantization with the orthogonal array method can generate more representative individuals than the pure random generation and the populations are more evenly distributed in the solution domain. However, both the pure random and bucketization can populate more evenly distributed individual in the solution space and the best solutions found by both methods can dominate the solutions generated by the traditional IMOEA. We conclude that by adding the randomized population, we can improve the chance of finding optimal solutions in the IMOEA method.

7. CONCLUSION

In this paper, we propose to use an additional randomized population to expand the search space in genetic algorithms to produce better individuals and thus lead to better solutions. The randomized population is not just randomly generated but Taguchi's method is also employed to select more representative individuals in the populations. Our experiment results suggest that the proposed method is feasible and can generate better solution than the original IMOEA method. In 2D coordinate system the visual disparity is meaningful. But when we transform the visual disparity into a third axis, how to more evenly distribute the inverse of visual disparity onto the third Z-axis remains an important problem. Our future directions include the use of Principal Component Analysis (PCA) to determine main axis and better quantization on these main axes to locate more diverse corresponding point pairs and eventually lead to better solutions.

REFERENCES

[1] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in Computational Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25, No. 8, Aug. 2003, pp. 993-1008.

[2] M. Carcassoni, and E. R. Hancock, "Correspondence Matching with Modal Clusters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, Dec. 2003, pp. 1609-1615.

[3] J. Chai and S. D. Ma, "Robust Epipolar Geometry Estimation Using Genetic Algorithm," *Pattern Recognition Letters*, Vol. 19, 1998, pp. 829-838.

[4] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1996.

[5] Y.W. Leung and Y.Wang, "An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization," *IEEE Trans. Evolutionary Computation*, Vol.5, Feb. 2001, pp. 41–53.

[6] S.Y. Ho, L.S. Shu, and J.H. Chen, "Intelligent Evolutionary Algorithms for Large Parameter Optimization Problems," *IEEE Trans. Evolutionary Computation*, Vol.8, No.6, December 2004, pp. 522-541.

[7] T. Jones, *Evolutionary Algorithms, Fitness Landscapes and Search*, Ph.D Dissertation, University of New Mexico, May 1995.

[8] J. M. Renders and S. P. Flasse, "Hybrid Methods Using Genetic Algorithms for Global Optimization," *IEEE Trans. Systems, Man, and Cybernetics-part B: Cybernetics*, Vol. 26, No. 2, April 1996, pp. 243-258.

[9] C.-Y. Tang, H.-L. Chou, Y.-L. Wu and Y.-H. Ding, "Fast and Robust Algorithm Using Coplanar Constraints to Estimate Fundamental Matrices," *IEEE International Conf. Systems, Man and Cybernetics*, Taipei, Taiwan, 2006.

[10] J. T. Tsai, T. K. Liu, and J. H. Chou, "Hybrid Taguchi-Genetic Algorithm for Global Numerical Optimization," *IEEE Trans. Evolutionary Computation*, Vol. 8, No. 4, Aug. 2004, pp. 365-377.

[11] Z. Tu and Y. Lu, "A Robust Stochastic Genetic Algorithm (StGA) for Global Numerical Optimization," *IEEE Trans. Evolutionary Computation*, Vol. 8, No. 5, October 2004, pp. 456-470.

[12] W. Zhong, J. Liu, M. Xue, and L. Jiao, "A Multiagent Genetic Algorithm for Global Numerical Optimization," *IEEE Trans. Systems, Man, and Cybernetics—part B: Cybernetics*, vol. 34, no. 2, April 2004, pp. 1128-1141.

[13] Vlasis K. Koumousis and Christos P. Katsaras; "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," *IEEE Trans. Evolutionary Computation*, Vol. 10, No. 1, February 2006.

[14] E. Zitzler, M. Laumanns, and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization," *Metaheuristics for Multiobjective Optimization, Lecture Notes in Economics and Mathematical Systems*, Springer, vol. 535, 2004, pp. 3–37.

[15] Qingfu Zhang and Yiu-Wing Leung,; "An Orthogonal Genetic Algorithm for Multimedia Multicast Routing," *IEEE Trans. Evolutionary Computation*, Vol. 3, No. 1, Apr 1999

[16] Yimin Liu, Tansel Özyer, Reda Alhajj and Ken Barker "Integrating Multi-Objective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering," *Informatica* 29 (2005) 33–40E.

[17] Zitzler and L. Thiele. "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Trans. Evolutionary Computation*, Vol. 3, No. 4,257–271, 1999.

[18] Yueh-Hung Lai, *Research on Optimization Using Evolutionary Computation*. Master Thesis, Department of Information Management, Huafan University Taipei, Taiwan, May 2005.

[19] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[20] Yan-Hung Ding, *3D Reconstruction of Buildings from Uncalibrated Image Sequences*. Master Thesis, Department of Information Management, Huafan University, Taipei, Taiwan, May 2005.