

Study on Multi-objective Optimization Problem of Multi-source Image Fusion

Yang Xinfeng^{*,1} and Liu Zhiyuan²

¹School of Computer and Information Engineering, Nanyang Institute of Technology, Henan, Nanyang, 473004, P.R. China; ²Department of Urban Rail Translation, ZhengZhou Technical College, Henan, ZhengZhou, 450121, P.R. China

Abstract: Multi-source image fusion integrates multiple images derived from the same scene or target collected into a new image to obtain more accurate and more complete description about the scene or target. The multi-objective optimization problem of multi-source image fusion is researched in the transform domain. Based on the analysis of multi-objective optimization theory and algorithms, an adaptive differential evolution algorithm is proposed. With adaptive variance factor, dynamical crossover probability function and optimal elite ordering strategy, the algorithm reflects not only good search capability but also good convergence. When applied to multi-objective optimization of multi-source image fusion of transform domain, it will be an effective solution to the comprehensive evaluation in the image fusion process.

Keywords: Adaptive differential evolution algorithm, Multi-objective optimization, Multi-source image fusion.

1. INTRODUCTION

Multi-objective optimization problem (MOPs) is one of the theoretical researches and engineering practice prevalent form of multi-objective optimization problem solving, seeking to meet all the objectives and requirements of the optimal solution [1]. In the most practical problems, among competing objectives, this tradeoff, a goal to improve performance, often reduces the performance of other more target expenses. Unable to competitively find all the targets to achieve a unique solution to optimize, the targets must be weighted for collection of an optimal solution.

With the rapid development of sensor and computer technology, the need for acquisition, image transmission and processing of information is growing and more and more types of image fusion technology requirements are also increasing. In digital image processing system, different contents of image processing with different processing models and procedures are stored in different storage systems, which not only takes up more storage resources, but there is no unified perception and understanding of the standard [2]. If a different image fusion task relatively unified multi-objective optimization system is set, it will be able to effectively reduce the waste of a variety of software and hardware resources, to achieve the best results.

From the perspective of control theory, multi-source image fusion is a source of many images for process optimization. Due to the effect of image fusion evaluation on diversity, the relationship between the complex and even contradictory indicators, the current emphasis on fusion method only requires some indicators, which have the difficulty to fully play a role. In this sense, multi-source image fusion of multi-objective optimization problem should be explored and solved, and also an image processing technology can

promote social progress and development of meaningful research.

2. MULTI-OBJECTIVE OPTIMIZATION ALGORITHM

2.1. Multi-Objective Optimization Problem

The traditional method for solving multi-objective optimization problem involves converting a multi-objective problem into several single objective problems, re-using single-objective optimization algorithm to find a solution [3]. Single-objective optimization algorithm can solve to some extent the multi-objective optimization problem, but the solution is likely a non-optimal solution. Multi-objective weighting method requires prior knowledge or set of weights for each goal, as there are a lot of subjective preferences being experience-dependent [4]. In the fusion process, multiple evaluation is not a simple linear relationship between the weights, so it is necessary to introduce multi-objective optimization algorithm to obtain optimal image fusion results.

In the single-objective optimization problem, the optimal solution is often the only identified, but in a multi-objective optimization problem, there is one solution for a goal which might be better, or even worse for the other goal which is not ideal, as generally, globally optimal solution does not exist for each target. Optimal multi-objective optimization problem is usually just a "compromise" set of the optimal solution [5], the collection is called the Pareto optimal solution set.

The key for solving multi-objective optimization problem is obtained as the approximate Pareto optimal solution set frontier, and a set of solutions [6] evenly distributed. To minimize the problem, for example, multi-objective optimization problem can be expressed as:

$$\begin{cases} \min y = f(X) = \{f_1(X), \dots, f_i(X), \dots, f_n(X)\} \\ X = (x_1, \dots, x_j, \dots, x_m) \\ x_{j\min} \leq x_j \leq x_{j\max}, j = 1, 2, \dots, m \end{cases} \quad (1)$$

*Address correspondence to this author at the School of Computer and Information Engineering, Nanyang Institute of Technology, Nanyang, 473004, P.R. China; Tel: 13803770071; E-mail: ywind2005@163.com

wherein: X is an m -dimensional decision variable; $f(X)$ is function-dimensional target vector, $f_i(X)$ ($i=1,2,\dots,n$) for the i -th objective function; x_j ($j=1,2,\dots,m$) for the j -th component of x , *i.e.* the j -th decision variables, $x_{j\min}$ and $x_{j\max}$ its minimum and maximum values, respectively.

On this basis, multi-objective optimization is listed in several important definitions:

Definition 1 (advantages and disadvantages): If u section is less than v , that is $\forall i \in \{1,2,\dots,n\}, (u_i \leq v_i) \wedge (\exists i, u_i < v_i)$, it can be claimed that the objective function vector $u = (u_1, \dots, u_j, \dots, u_n)$ has gifted $v = (v_1, \dots, v_i, \dots, v_n)$, or v is inferior to u .

Definition 2 (p disposable): If the decision variables X_1 and X_2 are in feasible domain, only if $f(X_1)$ is better than $f(X_2)$, then it can be said that solution X_1 Pareto dominates X_2 , or X_2 is Pareto dominated by solution X_1 .

Definition 3 (Pareto optimal): If a feasible decision variable X_3 is not Pareto dominated by any solution in viable region, X_3 can be called Pareto optimal for multi-objective optimization problem.

Pareto optimal solution set of all Pareto optimal compositions is called multi-objective optimization problem, denoted by P_S ; objective function vector corresponding Pareto optimal solution set constitutes a multi-objective problem of non-Pareto, denoted by P_F . Multi-objective optimization problem by specific objective function is a constant determined by the collection. In a multi-objective optimization problem Pareto optimal set of P_S , there is no better solution than the optimal solution Pareto.

2.2. Adaptive Differential Evolution Algorithm

As a class of evolutionary algorithms heuristic search algorithm, by maintaining the composition of the population of potential solutions between generations to achieve global search, which from population to population optimization method for multi-objective optimization problem of searching optimal solution p set is very effective. Because of the evolutionary algorithm having higher parallelism, in solving multi-objective optimization problem, multiple p optimal solutions can be obtained, with unparalleled advantages of traditional algorithms, which has gained academic attention [7, 8].

The typical evolutionary algorithms NSGA, NSGA-II, SPEA, etc., to solve the multi-objective optimization problem have currently become a hot topic in the intelligent computing industry [9-11]. Differential evolution algorithm is a random parallel direct global search algorithm newly

proposed by Storn and Price [12], which uses real component synthesis parameter vector; the evolutionary process is a less controlled parameter, which is easy to understand and implement. It has shown to be highly efficient, convergent and robust, and has been used in many complex optimization problems [12, 13].

Similar to the GA, PSO, ACA, AI and other evolutionary algorithms, differential evolution algorithm is based on the population of each individual genetic operations (crossover and mutation) to achieve the group's evolution; biological evolution is mathematical "micro" level of modeling. It has good local optimization ability, but the ability of global search is often relatively poor [14]. In order to make the front of the group inclined towards the direction of convergence, uniformly distributed in the entire front end and not falling into the local front-end, it is necessary and timely to adjust the differential evolution algorithm global search capability and control algorithm convergence speed [15].

In evolutionary multi-objective optimization algorithm, to generate new individual mechanisms undoubtedly has a significant impact on the performance of the algorithm, while the majority of evolutionary algorithms are still currently using the traditional crossover and mutation to generate new individuals. Based on the above analysis, we have proposed a multi-objective optimization problem solving adaptive differential evolution algorithm. In the course of evolution, mutation and crossover operations were introduced as adaptive concave exponential factor and dynamic cross mutation probability function. By controlling the convergence speed, it has been observed that the algorithm has strong global convergence capability at the start.

2.2.1. Variation

In the differential evolution algorithm, NP parameter vector $X_i = (x_{i1}, \dots, x_{in})$ ($i=1,2,\dots, NP$) forms a generation, optimizing in the search space. Its composition is the parent of the variation of the different vectors, including the parent of two different individuals' each differential vector. Mathematical expression of mutation is:

$$X_i^{g+1} = X_i^g + F * (X_{r1}^g - X_{r2}^g) \quad (2)$$

Where, g is the current iteration number; r_1, r_2 are integers different from each other, taken from a random population set $(1,2,\dots, NP)$; variability factor F is usually taken to be a constant within the scope of $[0,1.2]$.

The value of F determines the size of the convergence rate and has an important impact on the performance of the algorithm. In the early evolution, in order to ensure that the solution can be effective for space exploration, a large variation in the value factor can slow down the convergence speed appropriate to strengthen the global search ability of the algorithm; in the evolutionary process, if variability factor is gradually reduced, it can speed up the convergence, strengthening the local search capabilities.

In evolutionary algorithms, mutation operators are not only for the decision variables (individual) change, but also

for every decision to change the range of variables [16, 17]. In order to ensure that the beginning of the algorithm may change the coverage of each individual and then become non-linear with time to reduce its coverage area, this paper has introduced an adaptive concave exponential factor variation and mutation operations improved mathematical expression, that is:

$$F^g = \frac{1}{e-1} [F_{\min} e^{-F_{\max}} + (F_{\max} - F_{\min}) \exp(1 - \frac{g}{G_{\max}})] \quad (3)$$

Wherein: F_{\max} , F_{\min} are the variation factor and the lower limit value; G_{\max} is the maximum number of iterations.

2.2.2. Cross

In order to increase the diversity of the new population, the crossover operation is introduced after the variation operation. For each generation of populations, the i -th individuals X_i^g and X_i^{g+1} correspond to the variation of individual cross operation and generated cross individual $C_i = (c_{i1}, \dots, c_{in})$ ($i=1, 2, \dots, NP$). To ensure individual X_i^g evolution, first random selection is done, such that C_i least one contribution is from X_i^g , then $P_c \in [0, 1]$ crossover probability factor f is re-used of other bits which determine C_i contribution from X_i^{g+1} , X_i^g .

Mathematical expression of crossover operation is:

$$C_{in} = \begin{cases} x_{in}^{g+1} & \text{rand}() \leq P_c \\ x_{in}^g & \text{rand}() > P_c \end{cases} \quad (4)$$

Where, $\text{rand}()$ is a random number uniformly distributed between $[0, 1]$.

Formula (4) shows that if P is larger, the X_i^g 's contribution to C_i is more conducive to local search which may accelerate the convergence rate; if P_c is smaller, the X_i^{g+1} 's contribution to C_i is more conducive to maintaining populations' diversity and global search.

To avoid falling into local optimal algorithm, so that the optimal solution is evenly distributed in the front as possible, we have designed a dynamic crossover probability function; improved cross-operating mathematical expression is:

$$P_c = P_{c_{\min}} + (P_{c_{\max}} - P_{c_{\min}}) (1 - \frac{g}{G_{\max}})^n \quad (5)$$

Wherein: $P_{c_{\max}}$, $P_{c_{\min}}$ are for the crossover probability and the lower limit value; n is a cross index. P_c increases with the increasing number of iterations while it decreases gradually when different values of n occur. P_c of the curve is shown in Fig. (1).

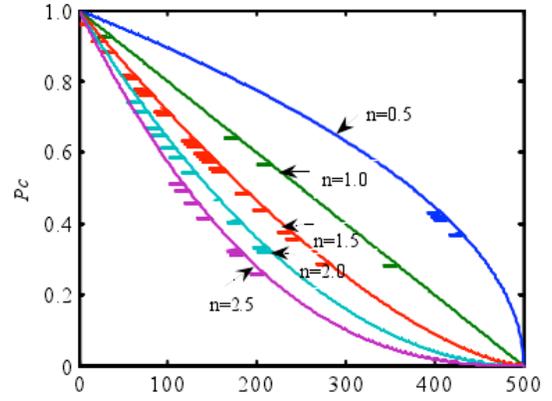


Fig. (1). Curve of crossover probability.

2.2.3. Selection

After mutation and crossover operations generate C_i test individual competition with X_i^g , when the fitness is better than X_i^g then C_i is selected as the offspring, or directly to X_i^g as a child. Options mathematical expression is:

$$X_i^{g+1} = \begin{cases} C_i & f(C_i) \leq f(X_i^g) \\ X_i^g & f(C_i) > f(X_i^g) \end{cases} \quad (6)$$

In a multi-objective optimization problem, if there is no absolute optimal solution which can only define satisfactory Pareto optimal, the collections consisting of Pareto optimal solution set are considered the optimal solution. Due to the current population of a Pareto optimal solution, which cannot necessarily be found in the future evolution of the population, each generation always maintains optimal use of external memory [7] to save the evolutionary process of finding the optimal solution p . In order to ensure the optimal diversity into the memory, according to the elite optimal strategy [14], objective function value is sorted.

If the individual after X_i^g through mutation, Get the crossover operation after the new individual C_i better than X_i^g , the individual and the memory already Pareto optimal solution set to compare; otherwise, indicating that the individual has not been evolutionary, not to be compared. (AU-THOR: The highlighted is vague and must be re-phrased) If C_i has dominant P^* in some individuals, it will be deleted from dominant individual in P^* , and will be added to the P^* ; if C_i and P^* have the absence of individual dominance relationship, it is thought that C_i is the optimal solution of Pareto.

Randomly selecting a target function $f_j (j=1, 2, \dots, k)$, $f_i(C_i)$ and P^* in the objective function value correspond to a large ascending sort order to calculate $f_j(c_i)$ sort of back and forth corresponding distance between the individual objective function value. If the minimum distance is smaller

than the set threshold, then C_i does not join P^* but if it is greater than or equal to threshold, then C_i joins P^* , maintaining the diversity of Pareto optimal solution set. All new solutions are not dominant and are related to the memory of a new population of individuals; when the memory reaches its maximum storage capacity, the objective function value is calculated according to the individual degree of congestion. If the crowd is small, indicating that the individual is in dense area, it will not enter the memory.

The major steps of differential evolutionary multi-objective optimization algorithm are as follows:

(1) For random initial population in the search space, the initialization parameters and differential evolution of the maximum number of iterations are set;

(2) The value of the initial population of each individual function optimization goals is calculated according to Pareto optimality theory to identify the dominant individual; the individual is then set for the initial population of Pareto optimal individual;

(3) In the population, randomly three different individuals are selected according to equation (3) of adaptive mutation to generate individual variation;

(4) According to equation (5), the individual computing function values can be obtained corresponding to each of the two optimization goals;

(5) For objective function value for the individual cross-ordering, if the individual meets the diversity of conditions, it is decided whether to add Pareto optimal solution set or not to join;

(6) In the next iteration, steps (3) to (5) are repeated;

(7) It is determined whether the maximum number of iterations is reached at the end of the evolutionary process; if not, then the process is returned to step (3);

2.3. Simulation

To test the performance of the algorithm proposed in this paper, selected four typical benchmark functions ZDT1, ZDT2, ZDT3 and ZDT6 are tested [9]. Where, ZDT1 and ZDT2 are convex and non-convex continuous functions, ZDT3 is for convex discontinuous functions; ZDT6 compared with non-convex function is affected by the sine function; their expression is as follows:

$$(1) \text{ ZDT1: } \begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X) \times [1 - \sqrt{x_1/g(X)}] \\ g(X) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1) \\ n = 30, x_i \in [0, 1], i = 1, 2, \dots, n \end{cases} \quad (7)$$

(2) ZDT2:

$$\begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X) \times [1 - (\sqrt{x_1/g(X)})^2] \\ g(X) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1) \\ n = 30, x_i \in [0, 1], i = 1, 2, \dots, n \end{cases} \quad (8)$$

(3) ZDT3:

$$\begin{cases} f_1(X) = x_1 \\ f_2(X) = g(X) \times [1 - \sqrt{x_1/g(X)} - \frac{x_1}{g(X)} \sin(10\pi x_1)] \\ g(X) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1) \\ n = 30, x_i \in [0, 1], i = 1, 2, \dots, n \end{cases} \quad (9)$$

(4) ZDT6:

$$\begin{cases} f_1(X) = x_1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ f_2(X) = g(X) \times [1 - (\sqrt{x_1/g(X)})^2] \\ g(X) = 1 + 9 \times \left[\left(\sum_{i=2}^n x_i \right) / (n-1) \right]^{0.25} \\ n = 10, x_i \in [0, 1], i = 1, 2, \dots, n \end{cases} \quad (10)$$

Experimental parameters differential evolution algorithm is set to: $NP = 100$, $G_{\max} = 300$, $F_{\max} = 0.8$, $F_{\min} = 0.3$, $P_{c\max} = 0.6$, $P_{c\min} = 0.2$, $n = 1.5$.

These four test functions of the Pareto optimal solution set are shown in Fig. (2a-d) below.

From the figure, it can be found that the algorithm on the distribution of the four benchmark functions has better convergence. But susceptible to subjective perception of psychological factors, the need for quantitative evaluation by means of comparison with other optimization methods is difficult to judge just from the visual merits of the algorithm level.

Being different from the single-objective optimization problem, multi-objective optimization requires not only convergence to the true Pareto front end, but also to ensure uniform distribution of the solution in the Pareto front end. In order to objectively evaluate the effect of multi-objective optimization, this paper introduced error rate (ER) and pitch (SP) [18] results of the two indicators to compare different algorithms obtained:

Pareto optimal solution represents the error rate obtained in the number of solutions focused on the non-Pareto proportion reflecting the convergence of the algorithm. If the error rate value is smaller, it results in the Pareto optimal solution set being closer to the true front.

Error rate expression is given by:

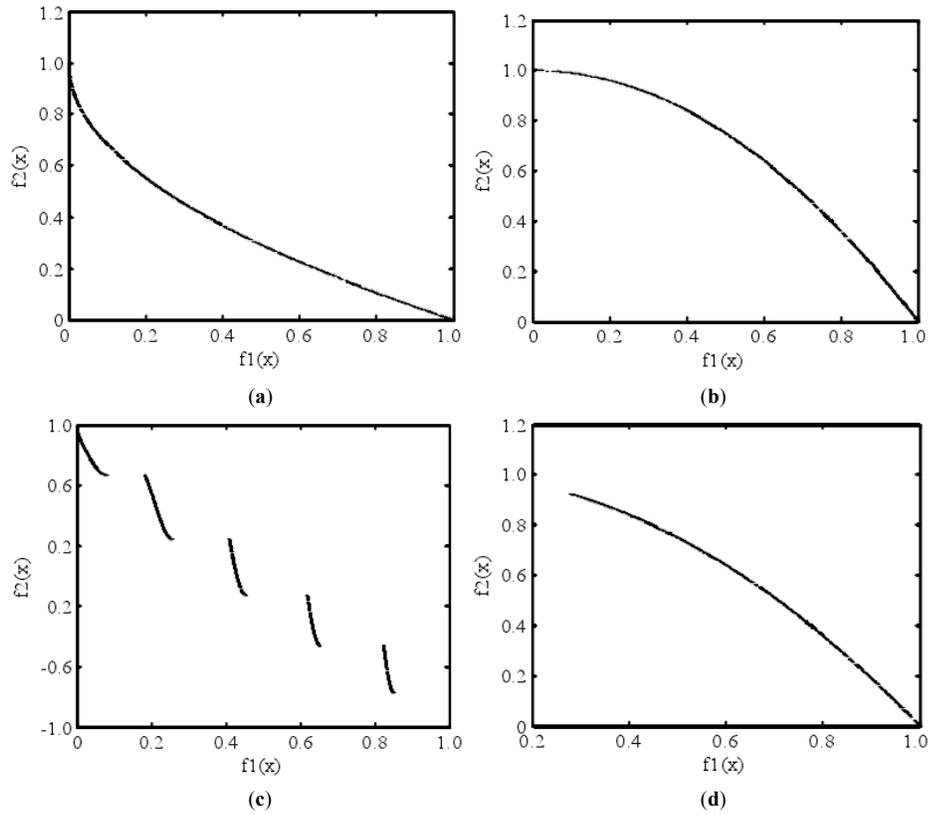


Fig. (2). Pareto optimal sets: (a) ZDT1; (b) ZDT2; (c) ZDT3; (d) ZDT6.

Table 1. Means of ER and SP.

Optimization	ZDT1		ZDT2		ZDT3		ZDT4	
	Error Rate	Spacing						
NSGA-II	0.03101	0.39290	0.07050	0.33293	0.10468	0.01714	0.30346	0.60610
DE	0.00472	0.00585	0.00867	0.00994	0.01214	0.00450	0.01577	0.00249
ALGORITHM	0.00039	0.00065	0.00014	0.00099	0.00101	0.00082	0.00113	0.00040

$$ER = \frac{1}{N} \sum_{i=1}^N e_i \tag{11}$$

Where, N is the number of Pareto optimal centralized solution. When Pareto approximate solution is a Pareto optimal solution, $e_i = 0$, otherwise $e_i = 1$.

Spacing is mainly used to measure income distribution Pareto optimal solution set of the Pareto front end, and its expression is:

$$SP = \sqrt{\frac{1}{i-1} \sum_{i=1}^N (\bar{d} - d_i)^2} \tag{12}$$

In the formula, \bar{d} is the average value of all d_i ,

$$d_i = \min_j \left\{ \sum_{i=1}^k |f_m^i(x) - f_m^j(x)| \right\} \quad i, j = 1, 2, \dots, N \tag{13}$$

If $SP = 0$, it shows that the resulting Pareto optimal solution in the Pareto front end is equally spaced; the smaller the spacing values of the resulting optimal solution set, the more evenly distributed these are on the Pareto front end.

Table 1 shows the NSGA-II, DE and algorithm for ZDT1, ZDT2, ZDT3 and ZDT6 independent operation and the mean distance between the mean error rates as 30 times respectively. In the NSGA-II, the distribution parameters are $\eta_c = 20$, $\eta_m = 20$, run algebra 500; differential evolution mutation factor $F = 0.35$, crossover probability $P_c = 0.30$, 500 iterations.

Table 1 shows that in the proposed adaptive differential evolution algorithm after 30 operations, the error rate is significantly less than the average that reached NSGA-II and differential evolution algorithm calculated value. This shows that the non-inferiority of the algorithm obtained Pareto solution proportion of true Pareto optimal solution is much high-

er than the other two algorithms, so the Pareto set gets closer to the true Pareto front end. At the same time, the average value of the new algorithm obtained spacing than the other two algorithms calculated value is several times lower, even more than 100 times, proving that the Pareto optimal solution obtained solution has better uniform distribution characteristics than the other two algorithms on the Pareto front end. (AUTHOR: Please review the highlighted corrections for confirmation)

Simulation results show that compared with other multi-objective evolutionary algorithms, this algorithm converges faster and there is a uniform distribution of non-dominated solutions obtained, reflecting the good global and local search capability of the adaptive differential evolution algorithm, and verifying the algorithm's superiority in solving multi-objective optimization problem.



Fig. (3). Multi-focus images: (a) Focus on the left; (b) Focus on the right.

3. EXPERIMENTS AND RESULTS ANALYSIS

3.1. Multi-Focus Image Fusion

In order to verify the effectiveness of the proposed algorithm, a set of multi-focus image fusion experiments is first used. Fig. (3a) is a left side focus of Lab images, Fig. (3b) is Lab focused image on the right, the size of the source images A and B is 620×460 , and has registration before fusion.

According to the characteristics of multi-focus image fusion, the paper selected four kinds of evaluation, according to the priority order: mutual information > Entropy > average gradient > MSE. Multi-objective optimization process and conditions set initialization parameters are: population size $NP = 100$, maximum number of iterations $G_{max} = 200$.

In NSCT domain generally, the pixel-level fusion method, NSGA-II, multi-objective optimization differential evolution (DEMO) and the Adaptive Differential evolutionary

multi-objective optimization (ADEMO) method are used for multi-focus image fusion. For comparison, in the non-subsampled Contourlet transform domain, respectively, Chapter III is used of the general pixel-level fusion method and NSGA-II, DEMO and RDEMO fusion method. NSCT nonsubsampling filter banks are used as Maxflat and Dmaxflat7 pyramid filters and directional filter, and the low-frequency coefficients are used in the weighted average, high-frequency coefficients neighborhood whichever is greater contrast fusion rules. (AUTHOR: Te highlighted is vague and must be re-phrased)

Fused image obtained by the method is shown in Fig. (4a-d); the corresponding objective evaluation is shown in Table 2.

The fusion results of data can be seen in Table 2:

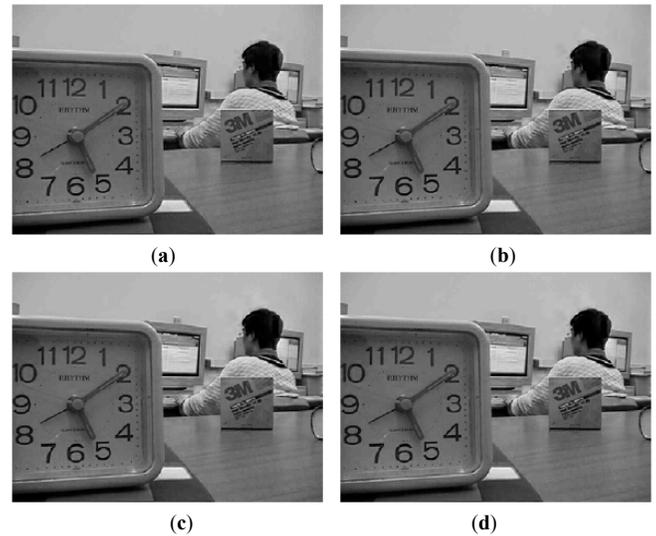


Fig. (4). Multi-focus image fusion results: (a) NSCT+Pixel; (b) NSGA-II; (c) DEMO; (d) RDEMO.

(1) In this paper, the fusion method was used to obtain relatively best fusion results; the overall visual image fusion desired effect can better retain the multi-focus image edge information and details, and verify the effectiveness of adaptive differential evolution algorithm.

(2) In the objective evaluation of the results, reflecting the fusion process to optimize the overall cross-entropy primary objective characteristics reflects the evaluation of prioritized superiority.

Table 2. Objective evaluation of fusion results

Fusion	Evaluation			
	Overall Cross-entropy	Entrop	Average Gradient	Standard Deviation
NSCT+Pixel	0.3072	7.0722	4.0832	9.3367
NSGA-II	0.2672	7.1015	4.0964	9.3592
DEMO	0.2231	7.1646	4.1031	9.3523
RDEMO	0.1983	7.1765	4.1030	9.3609

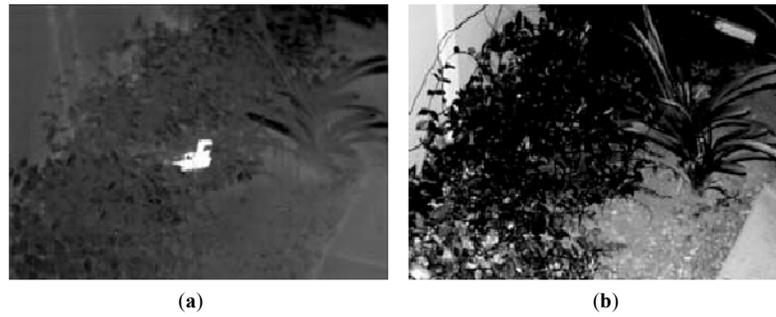


Fig. (5). Infrared and visible light images: (a) Infrared image; (b) Visible light image.

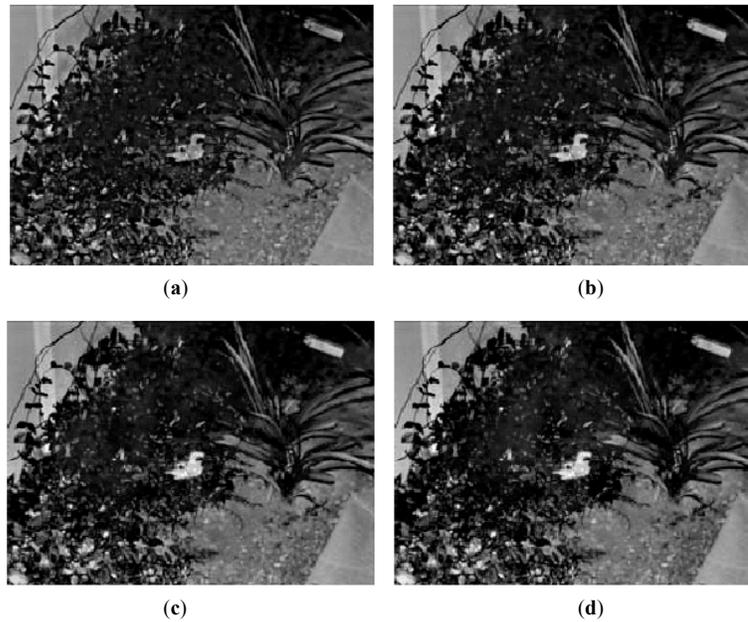


Fig. (6). Infrared and visible light image fusion results: (a) NSCT+Pixel; (b) NSGA-II; (c) DEMO; (d) RDEMO.

Table 3. Objective evaluation of fusion results.

Fusion	Evaluation			
	Edge Information Retention	Entrop	Average Gradient	Standard Deviation
NSCT+Pixel	0.6645	7.2671	13.6971	1.2354
NSGA-II	0.6910	7.2937	13.8156	1.1776
DEMO	0.6963	7.3265	13.8225	1.1087
RDEMO	0.6963	7.3290	13.7461	1.0577

3.2. Infrared and Visible Image Fusion

Using infrared and visible image fusion experiments, Fig. (5a) shows infrared camera collecting images of Bird. Fig. (5b) is visible light camera captured Bird image.

Infrared and visible image fusion based on the characteristics of the selected evaluation and prioritization is represented by: edge information retention> Entropy> average gradient> overall cross-entropy. Fusion results of generally pixel level fusion method and NSGA-II, DEMO, RDEMO method are shown in Fig. (6a-d); the corresponding evaluation is shown in Table 3.

From Fig. (6) and Table 3, it can be seen: Compared to Fig. (6a-c), Fig. (6d) not only showed a clear goal, a complete outline of the characteristics of the target but background surrounding the plant was more vivid, having distinct levels. Meanwhile, objective assessment of Fig. (6d) is also superior to other fused image.

Taking these two sets of results, the following has been found: In this paper, whether visual or objective evaluation, image fusion result showed that multi-objective optimization method is much better than several other fusion algorithms. This is mainly due to the optimization method discussed in

this paper, narrowing the gap between the initial state of the population and the corresponding group of Pareto optimal solution, in a limited number of iterations, quickly obtaining the optimal solution set. If the other two optimization methods are used, the performance of the integration process will be greatly reduced, or even can not be satisfied with the optimal solution set in a limited number of iterations. Subjective and objective evaluation results are reflected in the fused image adaptive differential evolution method which is applied to image fusion multi-objective optimization, obtaining good results.

4. CONCLUSION

For multi-source image fusion in multi-objective optimization problem, this paper introduced an adaptive differential image fusion of multi-objective optimization evolutionary algorithm optimization process. This algorithm allows to keep the diversity of the population, and to avoid premature convergence. In the multi-objective image fusion model non-subsampled Contourlet transform domain, respectively, using multi-focus image, infrared and visible images fusion with the proposed optimization method is validated.

Simulation and image fusion experiments show that Adaptive Differential Evolution Pareto optimal solution to maintain multi-objective optimization algorithm for image fusion provides better integration of the results of optimization than conventional frequency domain weighting and is generally multi-objective, so as to effectively solve the multi-source image fusion process of comprehensive evaluation.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the Henan Science and Technology Key Project (122102210563 and 132102210215).

REFERENCES

- [1] K. Li, H. Li, L. Kang, Z. Wu, "A multi-objective evolutionary algorithm based on transport theory", *The Computer Journal*, vol. 30, pp. 796-805, 2007.

- [2] Y. Niu, Y. Bo, and L. Shen, "Summary of application of image processing in multi-objective optimization", *Systems Engineering and Electronics*, vol. 30, pp. 1774-1780, 2008.
- [3] T. Back, D. Fogel, and Z. Michalewicz, *Evolutionary Computation 2-advanced Algorithms and Operators*, Philadelphia: Institute of Physic, 2000, pp. 65-98.
- [4] X. Cui, and C. Lin, "A genetic algorithm reconcile preference multi-objective", *Journal of Software*, vol. 16, pp. 761-770, 2005.
- [5] C.A. Coello, A.H. Aquirre, and E. Zitzler, "Evolutionary multi-objective optimization", *European Journal of Operational Research*, vol. 181, pp. 1617-1619, 2007.
- [6] D. Niu, F. Wang, and D. He, "Multi-objective chaotic differential evolution algorithm", *Control and Decision*, vol. 24, pp. 361-364, 2009.
- [7] C. A. Coello, "Evolutionary multi-objective optimization: a historical view of the field", *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28-36, 2006.
- [8] M. Gong, L. Jiao, and D. Yang, "Evolutionary multi-objective optimization algorithm", *Journal of Software*, vol. 20, pp. 271-289, 2009.
- [9] T. Lan, S. Liu, and X. Gu, "Based multi-objective evolutionary algorithm optimization method", *Control and Decision*, vol. 21, pp. 601-605, 2006.
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.
- [11] H. Li, J. Lu, W. Hou, and Y. Zhao, "Optimization based on hybrid genetic algorithm catalytic reforming process more objective", *Chemical Technology*, vol. 61, pp. 432-438, 2010.
- [12] X. Wang, M. Hao, Y. Cheng, and M. Li, "A multi-objective optimization problem mixing optimization algorithm", *Journal of System Simulation*, vol. 21, pp. 4980-4985, 2009.
- [13] K. Zielinski, and R. Laur, "Differential evolution with adaptive parameter setting for multi-objective optimization", In: *Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore: IEEEEXPLORE, 2007, pp. 3585-359.
- [14] L. Wu, Y. Wang, X. Yuan, and J. Zhang, "Multi-objective optimization problem differential evolution algorithm", *Journal of Hunan University (Natural Sciences)*, vol. 36, pp. 53-57, 2009.
- [15] B. Liu, L. Wang, and Y. Jin, "Research differential evolution algorithm", *Control and Decision*, vol. 22, pp. 721-729, 2007.
- [16] C.A. Coello, G.T. Pulido, and M.S. Lechuga, "Handling multiple objectives with particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 256-279, 2004.
- [17] Y. Niu, and L. Shen, "IMOPSO multi-objective algorithm based on multi-focus image fusion", *Journal of Electronics*, vol. 34, pp. 1578-1583, 2006.
- [18] C.A. Coello, and N.C. Cortes, "Solving multiobjective optimization problems using an artificial immune system", *Genetic Programming and Evolvable Machines*, vol. 6, pp. 163-190, 2005.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Xinfeng and Zhiyuan; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.