■ 629

An Improved Robot Path Planning Algorithm

Xuesong Yan[1], Qinghua Wu[2,3], Hammin Liu[1,4]

[1]School of Computer Science, China University of Geosciences, Wuhan, China
[2]Hubei Provincial Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, China
[3]School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan, China
[4]Wuhan Institute of Shipbuilding Technology, Wuhan, China
e-mail:yanxs1999@126.com

Abstrak

Robot path planning adalah masalah NP.Metode optimasi tradisional tidak efektif untuk mengatasinya. Algoritma genetika tradisional terjebak ke dalam minimum lokal dengan mudah. Oleh karena itu, didasarkan pada algoritma genetika yang sederhana dan menggabungkan ideologi dasar metode desain orthogonal kemudian diterapkan ke inisialisasi populasi, menggunakan mekanisme elit antargenerasi, serta pengenalan operator pencarian adaptif lokal untuk mencegah terjebak ke dalam minimum lokal dan meningkatkan kecepatan konvergensi untuk membentuk algoritma genetik baru. Melalui serangkaian percobaan numerik, algoritma baru telah terbukti efisien. Kami juga menggunakan algoritma yang diusulkan untuk memecahkan masalah robot path planning dan hasil penelitian menunjukkan bahwa algoritma baru terbuktiefisien untuk memecahkan masalah robot path planning dan jalan terbaik biasanya dapat ditemukan.

Kata kunci*: optimalisasi fungsi, algoritma genetika, inisialisasi ortogonal, perencanaan robot jalan*

Abstract

Robot path planning is a NP problem. Traditionaloptimization methods are not very effective to solve it. Traditional genetic algorithm trapped into the local minimum easily. Therefore, based on a simple genetic algorithm and combine the base ideology of orthogonal design method then applied it to the population initialization, using the intergenerational elite mechanism, as well as the introduction of adaptive local search operator to prevent trapped into the local minimum and improvethe convergence speed to form a new genetic algorithm. Through the series of numerical experiments, the new algorithm has been proved to be efficiency.We also use the proposed algorithm to solve the robot path planning problem and the experiment results indicated that the new algorithm is efficiency for solving the robot path planning problems and the best path usually can be found.

Keywords*: function optimization, genetic algorithm, orthogonal initialization, robot path planning*

1. Introduction

Autonomous robots are developed to perform high level task without further humanoperation. To accomplish these tasks the robots need to move in the real world. Inconsequence, one of the more important problems to be solved in the design ofautonomous robots is the path planning. These planning problems can involvegeometric workspace, constraints and additional complex features such as incompleteknowledge, moving obstacles, sensing and model uncertainties, unpredictability,kinematics, multiple robots and goals. In the last decade, path planning has receivedconsiderable attention from robotic communities since this fundamental operationrequires the solution of a variety of theoretical and practical problems.

Until now, the research in robot path planning has focused on findingoptimal routes from start to goal. The optimality is usually measured in terms oftraveled distances [1]. Other measures are also used, e.g. confidence value [2]. Forplanetary rovers the efficiency of path is often expressed in terms of slope orroughness of the surface [3, 4].

Robot path planning is a NP problem[5], traditionaloptimization methods are not very effective to it,which are easy to plunge into local minimum. A lot of algorithms have been proposed to solve NP problem. Some of them (based on dynamic programming or branch and bound methods) provide the global optimum solution. Other algorithms are heuristic ones, which are much faster, but they do not guarantee the optimal solutions.The most popular evolutionary

model used in the current research is Genetic Algorithms (GA), originally developed by John Holland[6]. The GA reproduction operators, such as recombination and mutation, are considered analogous to the biological process of mutation and crossover respectively in population genetics. The recombination operator is traditionally used as the primary search operator in GA while the mutation operator is considered to be a background operator, which is applied with a small probability.

Traditionally, GA uses a binary string representation of chromosomes with concentration on the notion of 'schemata'. A schema is a template that allows exploring the similarity among chromosomes. Genetic Algorithms model evolution as a search for structures or building blocks that perform well in a given environment. Therefore, the recombination and mutation operators focus on an individual's structure, not the structure's interpretation. The results of applying reproduction operation in GA generate solutions that share structural similarities with their parents but may have significantly different interpretations. However, many recent applications of GA have used other representation such as graphs, Lisp expressions, ordered list, and red-valued vectors.

```
t=0;
initialize P(t)
evaluate srructures in P(t)
repeat
t= t+1
select- reproduction C(t)from: P(t-1)
combine and mutate structures in C(t)forming C'(t);
evaluate structures in C'(t)
select-replace P(t) from C '(t) and P(t+ 1);
Until ( termination condition satisfied ).
```

The above code gives the basic algorithmic steps for GA. After the initial population of individuals is generated (usually randomly) and individuals' structures are evaluated, the loop is entered. Then a selection buffer C(t) is created to accommodate the selected copies from P(t-l), "select-reproduction". In the Holland original GA, individuals are selected probabilistically by assigning each individual a probability proportional to its structural fitness. Thus, better individuals are given more opportunity to produce offspring. Next the variation operators (mutation and crossover) are applied to the individuals in C(t) buffer producing offspring C(t). After evaluating the structural fitness of C(t), the selection method is applied to select replacement for P(t) from C'(t) and P(t-1).

In general, genetic algorithms are usually used to solve problems withlittle or no domain knowledge, NP-complete problems, and problems for which nearoptimum solution is sufficient. The GA methods can be applied only if there exist areasonable time and space for evolution to take place.

2. Improved Genetic Algorithm

In our algorithm, the code representation we use the real coding, the real coding method has advantages compared with the binary coding in the function optimization problems, because the realcoding can solve the "Hamming cliff"problem which the binary coding has no idea to solve it, and then the encoding, the decodingand the calculation of the fitness function is more convenient when we use the realcoding.

2.1 Initialize Population

The traditional method of genetic algorithm is randomly initialized population, that is, generate a series of random numbers in the solution space of the question. Design the new algorithm, we using the orthogonal initialization[7] in the initialization phase. For the general condition, before seeking out the optimal solution the location of the global optimal solutionis impossible to know, for some high-dimensional and multi-mode functions to optimize, the function itself has a lot of poles, and the global optimum location of the function is unknown. If the initial population of chromosomes can be evenly distributed in the feasible solution space, the algorithm can evenly search in the solution space for the global optimum. Orthogonal initialization is to use the orthogonal table has the dispersion and uniformity comparable; the

individual will be initialized uniformly dispersed into the search space, so the orthogonal design method can be used to generate uniformly distributed initial population.

2.2 Intergenerational Elite Mechanism

Genetic algorithm is usually complete the selection operation based on the individual's fitness value, in the mechanism of intergenerational elite,the population of the front generation mixed with the new population which generate through crossover and mutation operations, in the mixed population select the optimum individuals according to a certain probability.The specific procedure is as follows:

Step 1 : Using crossover and mutation operations for population P1 which size is N then generatingthe next generation of sub-populations P2;

Step 2 : The current population P1 and the next generation of sub-populations P2 mixed together form a temporary population;

Step 3 : Temporary population according to fitness values in descending order, to retain the best N individuals to form new populations P1.

The characteristic of this mechanism is mainly in the following aspects. First is robust, because of using this selection strategy, even when the crossover and mutation operations to produce more inferior individuals, as the resultsof the majority of individual residues of the original population, does not cause lower the fitness value of the individual.The second is in genetic diversity maintaining, the operation of large populations, you can better maintain the genetic diversity of the population evolution process.Third is in the sorting method, it is good to overcome proportional to adapt to the calculation of scale.

2.3 Adaptive Local Search Operator

Local search operator has a strong local search ability, and then can solve the shortcomings of genetic algorithm has the weak ability for the local search. And the population according to the current state of adaptive evolution of the local search space adaptive local search operator will undoubtedly greatly enhance the ability of local search. In the initial stage of the evolution, the current optimal solution from the global optimum region is still relatively far away, this time the adaptive local search operator to require search a large neighborhood space to find more optimal solution, it can maintain the population diversity. When the population has evolved to the region containing the global optimum, the adaptive local search operator to require a relatively small area to search in order to improve the accuracy of the global optimal solution.

In our algorithm, the adaptive local search operator is the adaptive orthogonal local search operator. Adaptive orthogonal local search operator is aimed at the neighborhood of a point to search, so the key point is to identify a point as the center of the hypercube, the hypercube in the orthogonal test, expect to be better solution.

2.4　Framework of Algorithm

In this paper, we design the improved genetic algorithm with the real encoding, orthogonal initialization, combined with the roulette selection and the adaptive local search operator, using the intergenerational elitist mechanism.The framework of the algorithm as follows:

Step 1 : Initialize a population P using the orthogonal initialization method, the scale is N, and evaluate the individuals in this population;

Step 2 : Using the roulette selection method, select two parent individuals from the population P respectively then using the crossover operation to get an optimal individual, then using the intergenerational elitist mechanism for this individual generate a sub-individual, and bring this individual into the new population pool P', repeat the step2 until the new population pool P'size is N;

Step 3 : Evaluate the new population P', then combined the new population P' with the populations P and form a temporary population;

Step 4 : Sort the temporary population according to the fitness value of the individuals in descending order; retain the best N individuals to form the next generation population P;

Step 5 : If not satisfy the termination condition, switch to step2, otherwise, end of the algorithm.

2.5 Experiment Verify

In order to verifythe improvement of the new algorithm, we use benchmarks function to test the algorithm's performance and compare the results with traditional genetic algorithm. Specific details of the test function see Table 1. In the table 1, n behalf of the dimension number of the function,S behalf of the range of variables, f_{min} behalf of the minimization of the function.

Table1. Test Function

Test Function	n	S	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	(-100,100)	0
$f_2(x) = 6 \cdot \sum_{i=1}^{5} \lfloor x_i \rfloor$	30	(-5.12, 5.12)	0
$f_3(x) = \sum_{i=1}^{n} i \cdot x_i^4 + U(0,1)$	30	(-1.28,1.28)	0
$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n}(x_i-100)^2 - \prod_{i=1}^{n}\cos(\frac{x_i-100}{\sqrt{i}}) + 1$	30	(-300.0,300.0)	0
$f_5(x) = -20\cdot\exp(-0.2\sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}x_i^2}) - \exp(\frac{1}{n}\cdot\sum_{i=1}^{n}\cos(2\pi\cdot x_i)) + 20 + e$	30	(-32.0,32.0)	0
$f_6(x) = \sum_{i=1}^{n}100((x_{i+1}-x_i^2)^2 + (x_i-1)^2)$	30	(-2.048,2.048)	0
$f_7(x) = \sum_{i=1}^{n}-x_i\sin(\sqrt{|x_i|})$	30	(-500,500)	-12569.5

Table 2. New algorithm and GA experiment

Function	Algorithm	Best Value	Worst Value	Standard deviation
$f_1(x)$	GA	1495.71	7032.89	201.038
	Improved GA	4.13731E-29	1.0882E-24	2.28015E-26
$f_2(x)$	GA	0	0	0
	Improved GA	0	0	0
$f_3(x)$	GA	0.00177094	0.00833963	0.000210055
	Improved GA	0.00193565	0.0103595	0.000259903
$f_4(x)$	GA	72.5069	123.954	1.52289
	Improved GA	2.18559E-12	8.63194E-25	0.00177512
$f_5(x)$	GA	-3.19744E-14	4.4229	0.148418
	Improved GA	-3.19744E-14	1.50229	0.0749509
$f_6(x)$	GA	1.84889E-28	8.63194E-25	1.83991E-26
	Improved GA	2.55147E-28	1.20401E-23	2.41678E-25
$f_7(x)$	GA	-5038.62	-3233.13	54.0123
	Improved GA	-9535.19	-8203.56	45.1661

The two algorithms of the same experimental parameters set. Each function in Table 1 is run 50 times with the two algorithms, their experimental results such as Table 2. By analyzing the experimental results we know, in solving function f1, f4 and f7, use the genetic algorithm is easily into local optimum, but use the new algorithm, convergence soon, and find a better solution, the average fitness and the best fitness was both superior to genetic algorithm. For the function f2, the new algorithm and GA all can find the global optimal, these two algorithm for this test function is very effective. For function f5, the two algorithms can find the best solutions are the same (see Table 2), and the new algorithm to get the best value of the average is better than GA algorithm.

In sum, we can see that in solving function f1, f4, f5 and f7, the new algorithm more efficient than GA algorithm,in solving other function, the performance almost same of the two algorithms. In short, this new algorithm has the following value: better global search capability.

3. Robot Path Planning Algorithm Based on Improved GA

We have chosen genetic algorithms to generate robot path mainly for tworeasons: firstly, they are a powerful tool for searching in high dimensional spaces,like in this case. Secondly, the method imposes few mathematical constraints inthe shape of the function to optimize. In this way, this method is applicable to thegeneration of a wide range of behaviors (obstacle avoiding, wall following, etc.).

3.1 Robot Path Planning Problem

Robot path planning can generally be considered as a search in a configuration space defined in what is known as the configuration space formulation [8-10]: Let A be a single rigid object, moving in a Euclidean space $W = \Re^N$, N = 2 or 3. Let O_1 ,… ,O_n be fixed rigid objects distributed in W. The O_i´s are called obstacles. If A is described as a compact subset of W, and the obstacles O_1 ,… ,O_n are closed subsets of W, a configuration of A is a specification of the position of every point in A with respect to F_W , where F_W is a Cartesian coordinate system. The configuration space of A is the space denoted by C, with all possible configurations of A. The subset of W occupied by A at configuration q is denoted by A(q) . A path from an initial configuration q_{init} to a goal configuration q_{goal} is a continuous map $\tau : [0,1] \rightarrow C$ with $\tau(0) = q_{init}$ and $\tau(1) = q_{goal}$.

The workspace contains a finite number of obstacles denoted by O_i with i=1,…,n . Each obstacle O_i , maps in C to a region $C(B_i) = \{q \in C | A(q) \cap O_i \neq 0\}$ which is called C obstacle The union of all the C obstacle is the C obstacle region, $\overset{n}{\underset{i=1}{U}} C(O_i)$ and the set $C_{free} = c \setminus \overset{n}{\underset{i=1}{U}} C(O_i) = \left\{q \in C \setminus A(q) \cap (\overset{n}{\underset{i=1}{U}} O_i = 0\right\}$ is called the free space. A free collision path between two configurations is any continuous path $\tau : [0,1] \rightarrow C_{free}$.

The configuration space is a powerful conceptual tool because it seems to be the natural space where the path planning problem "lives". This is mainly because any transformation of a rigid or articulated body becomes a point in the configuration space.

From mathematical point of view[11], robot path planning problem can be expressed as a find the optimal value of the goal function, the goal function is the cost of the path planning, constraints condition is to avoid collision with the obstacles, so the mathematic model of the robot path planning can expression in the following equation.

$$\min f(X), \quad X \in R^n$$
$$s.t. g_i(X) \le b_i, \quad i = 1,2,\cdots p \tag{1}$$

where the $f(X)$ is the goal function, $g_i(X)$ is the constraints condition and p denotes the numbers of the constraints inequation.

3.2 Path Representation

In our algorithm, to facilitate the coding of individual and genetic operation, the head point of the obstacles, the start point and the goal point are all called head point, give these head points the Number, though a path can be denoted a head points sequence, which the start head point is v_1, and the goal head point is v_k.

Each individual in the population represents a path starting from the initialpoint and trying to reach the final point.In this algorithm, we use list denote the path chromosome which can facilitate the insert and delete operation in the process of the path evolution. A single list denote a path, each list has a head point, all of the head point form the head array. So, the gene of the chromosome(the element of the list) denotes the point of the path. The Figure 1 is the chromosomerepresentation of our algorithm.
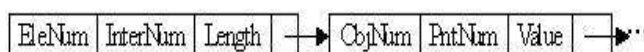

Figure 1. Chromosomerepresentation

Where, in the head point, EleNum denotes the number of the gene in the chromosome, InterNum denotes the number of the cross about the chromosome and the obstacles, Length denotes the length of the chromosome; where in each gene, ObjNum denotes the serial

numberof the obstacle, PntNum denotes the number of the gene in the obstacle ObjNum, Value denotes the location of the gene in the head point sequence.

3.3 Fitness Function

Find fitness function is an important part of genetic algorithms.For the robot path planning problem,if the fitness functiononly consider the length of the individual path is not enough, taking into account the situation of the obstacles cross with individual paths. For this reason, in our algorithm fitness function measures the optimality of each path considering two factors:the distance between the final robot position and the goal point and the cross situation which the individual have no cross with the obstacles. Fitness function equation as shoen in following equation.

$$f(S_i) = \frac{1}{\sum_{i=d1}^{d2-1} d(v_i, v_{i+1})}$$

(2)

where v_{d1}, v_{d2} is the location of the initial point s and the goal point g in the chromosome Si.

3.4 Algorithm Framework

Following provides a more detailed description of the whole robot path planningalgorithm base on improved genetic algorithm.

```
Random initialization of the population P
initialize S(t) ;  S(t) = {S₁(t), S₂(t),⋯, Sᵢ(t)}
evaluate S(t) ;  f(S(t)) = {f(S₁(t)), f(S₂(t)),⋯, f(Sᵢ(t))}
While (not satisfied termination condition) do
i=0;
begin
  for each individual Sᵢ ∈ P  do
  begin
S' ← Sᵢ
Select (randomly) a Node C from S'
repeat
{begin
    if (rand() ≤ p)
      select the node C' from the remaining nodes in S'
    else
      {Select (randomly) an individual in P
assign to C' the next node to the node C in the select individual
      }
    if (the next node or the previous node of node C in S' is C')
    exit repeat loop
    invert the gene node from the next node of C to the node C' in S'
C ← C'
}
end
  if (eval(S') ≤ eval(Sᵢ))
S_i ← S'
  end
    i=i+1;
      end
DO
  PRINT X_{best}, f(X_{best}) ;
```

3.5 Experiment Result

We simulated the robot path planning algorithm with computer, and compared the planned path with the tradition method: visibility graph.Forthe two algorithms, we generated the obstacles randomly. Figure 2 isthe visibility graph result, left corner is start point and rightcorner is intention point, the optimum path is shownwith green line.Figure 3 is our algorithm result. From the result figures, the results indicated thatour algorithm is efficiency for solving the robot path planning problems andit can found the right and optimalpath. We also compared the calculation time with the two method, our algorithm is faster than the visibility graph .
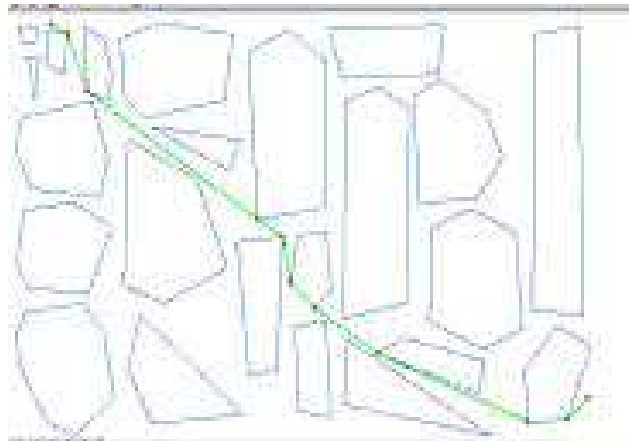

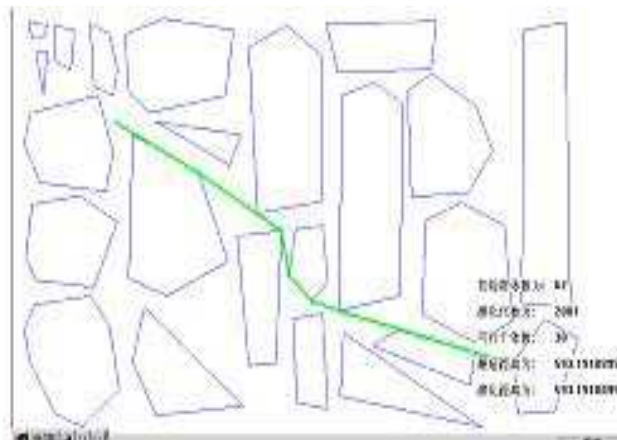
Figure 2. The visibility graph result

Figure 3. Our algorithm result

4. Conclusion

This paper introduces a new algorithm based on the traditional genetic algorithm, for the traditional GA algorithm the new algorithm has done some improvements: By introducing genetic selection strategy, decreased the possibility of being trapped into a local optimum. Compared the traditional genetic algorithm, the new algorithm enlarges the searching space and the complexity is not high. By analyzing the testing results of nine benchmarks functions optimization, we reach the conclusion: in the optimization precision and the optimization speed, the new algorithm is efficiency than the traditional genetic algorithm and the new algorithm is more efficient than traditional genetic algorithm in coping with the function optimization problems.We also use this proposed algorithm for solving robot path planning problem and compared the experiment result with traditional method, the experiment results indicated that the new algorithm is efficiency for solving the robot path planning problems and the best path usually can be found.

Acknowledgments

References

[1] A Yahja, S Singh, A Stentz. An Efficient on-line Path Planner for Outdoor Mobile Robots. *Robotics and Autonomous Systems*. Elsevier Science. 2000; 32: 129-143.
[2] UNehmzow, COwen. Robot Navigation in the Real World: Experiments with Manchester's Forty Two in Unmodified Large Environments.*Robotics andAutonomous Systems.*Elsevier Science.2000; 33: 223-242.
[3] A Howard, H Seraji. Vision-Based Terrain Characterization and TraversabilityAssessment.*Journal of Robotic Systems.* Wileyperiodicals.2001; 18(10): 577-587.
[4] DBGennery. Traversability Analysis and Path Planning for Planetary Rovers.*Autonomous Robots.* Kluwer, Academic Publishers.1999; 6: 131-146.
[5] G Liu, T Li, Y Peng, X Hou. *The ant algorithm for solving robot path planning problem.* 3rd International Conf on Information Technology and Applications. IEEE Press.2005; 2: 25-27.
[6] J Holland. Adaptation in natural and artificial systems. University of Michigan press.1975.
[7] Leung Yiu-Wing, Wang Yuping. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation*. 2001; 5(1): 41-53.
[8] Latombe JC. Robot Motion Planning. Kluwer Academic Publishers. Boston.1991.
[9] M Bracho de Rodríguez, J Ali Moreno. *Heuristic Algorithm for Robot Path Planning Based on Real Space Renormalization.* International Joint Conference, 7th Ibero-American Conference on AI: Advances in Artificial Intelligence table of contents, Lecture Notes in Computer Science, Springer-Verlag London, UK.2000: 379–388.
[10] Barraquand J, Langlois B,Latombe JC. Numerical Potential FieldTechniques for Robot Path Planning. *IEEE Transactions on System, Man and Cybernetic.* 1992; 22(2): 224-241.
[11] Durbin R, Willshaw D. An Analogue Approach to the Traveling Salesman Problem Using an Elastic NetApproach.*Nature.* 1987; 326(6114): 689-691.
[12] XSYan, LSKang et.al.*An Approach to Dynamic Traveling Salesman Problem.*3[rd]International Conference on Machine Learning and Cybernetics.IEEE Press, Shanghai, 2004;2418-2420.
[13] XSYan,AM Zhou, LS Kang et,al.*TSP Problem Based on Dynamic Environment.* 5[th]World Congress onIntelligent Control and Automation, IEEE press, Hangzhou,China.2004;2271-2274.
[14]   MJ. Rendas,WTetenoire.*Definition of exploratory trajectories in roboticsusing genetic algorithms.*10[th]International Conference in Industrial andEngineering Applications of Artificial Intelligence and Expert Systems.1997; 211-216.
[15] JXiao, Z Michalewicz, L Zhang, K Trojanowski. Adaptive evolutionaryplanner/navigator for robots.*IEEE Trans. on Evolutionary Computation.*1997; 1(1): 18-28.
[16] GengqianLiu, TiejunLi, YuqingPeng, XiangdanHou. *The Ant Algorithm for Solving Robot Path Planning Problem.*3[rd]International Conference on Information Technology and Applications, IEEE Press.2005; 2: 25-27.
[17] XSYan, Hui Li et.al.*A Fast Evolutionary Algorithm for Combinatorial Optimization Problems.*4th International Conference on Machine Learning and Cybernetics.IEEE Press.2005; 3288-3292.
[18] XSYan, Qing Hua Wu et.al.*A New Optimizaiton Algorithm for Function Optimization.*3rd International Symposium on Intelligence Computation & Applications, Lecture Notes in Computer Science.Springer Press.2009;144-150.
[19] XS Yan, QH Wu. Function Optimization Based on Cultural Algorithms. *Journal of Computer and Information Technology*. Academy Publish.2012; 2: 152-158.
[20] XS Yan, QH Wu, C Zhanget, al.An Improved Genetic Algorithm and Its Application.*TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(5): 1081-1086.
[21] QH Wu, HM Liu et, al. Research of Function Optimization Algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(4): 858-863.