WSN coverage hierarchical optimization method based on the improved MOEA/D

Xianhao Shen, Jun Li, Qi Zhang

*College of Information Science and Engineering,
Guilin University of Technology, Guilin 541004, China*

Abstract
Coverage is a fundamental problem in wireless sensor network (WSN), which is defined as the measurement of the quality of surveillance of sensing function. The concerns of coverage optimization are the maximize coverage rate and the minimize energy consumption. In this paper, we proposed the multi-objective evolutionary algorithm based on decomposition with particle swarm optimization (MOEA/D-PSO). Through preserving the high-quality individuals in present generation, improving the local optimization solution set in evolutionary search direction and the search progress, eventually propose to make up for the shortcomings of multi-objective evolutionary algorithm based on decomposition (MOEA/D). Compared with MOEA/D and non-dominated sorting genetic algorithm-II (NSGA-II), the results show that the MOEA/D-PSO is closer to Pareto optimal surface, the performs better in uniformity and diversity of solution set distribution. WSN has a broader coverage and consumes less energy.
Key words: COVERAGE OPTIMIZATION, COVERAGE RATE, MOEA/D-PSO, ENERGY CONSUMPTION, PARETO OPTIMAL SURFACE

1. Introduction

In recent years, multi-objective evolutionary algorithm (MOEA) has gradually become a hot topic, such as, non-dominated sorting genetic algorithm-II [1] (NSGA-II) and strength Pareto evolutionary algorithm 2 (SPEA2) [2], etc. The Pareto optimal solution of multi-objective optimization problem (MOP) could be an optimal solution of the weighting scalar optimization problem. Therefore, the whole Pareto front could be approximately divided into a number of single objective optimization sub-problems. Zhang and Li et al proposed a multi-objective evolutionary algorithm based on decomposition (MOEA/D) [3]. MOEA/D is a new framework of MOEA [4]. Through setting a group weight vector of the uniform and continuous distribution, converting MOP to a number of single objective sub-problems. By choosing the weight vector, it controls the individual's search direction and trajectory in the population [5]. MOEA/D has attracted to a growing number of researcher's interests, and many follow-up studies. Krzysztof [6] described the effects that cause the MOEA/D algorithm to converge asymmetrically. This was undesirable, because a multiobjective optimizer should explore the solution space evenly. Then the method gave some guidelines on how to avoid such asymmetries. Carvalho, Saldanha and Gomes et al [7] proposed a multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas. The algorithm was applied to the design of optimal 3 to 10 elements Yagi-Uda antennas, whose optimal Pareto fronts are provided in a single picture. It is solved by differential evolution (DE) and Gaussian mutation operators in order to provide a better approximation of the Pareto front. Ma and Liu et al [8] proposed approach is called opposition-based learning MOEA/D (MOEA/D-OBL). Compared with MOEA/D, MOEA/D-OBL uses an opposition-based initial population and opposition-based learning strategy to generate offspring during the evolutionary process. Muhammad and Rashida [9] proposed a study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D. In this method, the extended/modified versions of Stochastic ranking (SR) and constraint-domination principle (CDP) are implemented for the first time in the multiobjective evolutionary algorithm based on decomposition (MOEA/D) framework. Tan, Jiao and Li et al [10] proposed a new version of MOEA/D with uniform design, named the uniform design multiobjective evolutionary algorithm based on decomposition (UMOEA/D). UMOEA/D adopted the uniform design method to set the aggregation coefficient vectors of the sub-problems. MOEA/D provided a new idea for solving multi-objective optimization problem, which could directly introduce fitness assignment of solving single objective optimization problem, providing diversity maintenance strategy of multi-objective problem solution, effectively deal with the problem of high-dimensional and decision space discontinuity [11].

Since the MOEA/D had two problems, which were the lack of preserving high-quality individuals of present generation and individuals rarely in optimal solutions, we propose multi-objective evolutionary algorithm based on decomposition with particle swarm optimization (MOEA/D-PSO) in this paper. By preserving the high-quality individuals of present generation, improving local optimization solution set in evolutionary search direction and the search for progress, eventually make up for the shortcomings of t/he original MOEA/D. The aim is improving the convergence efficiency of the algorithm and providing more accurate solutions for decision-makers. Thereby, it increases the coverage rate and reduces the energy consumption.

2. The basic principle of MOEA/D

MOEA/D can convert a complex MOP to a set of simple single objective sub-problems. The goal of each single objective sub-problem is a weighted set of each objective. Then synchronously optimize these sub-problems by evolution algorithm (EA). Neighborhood relationships in these sub-problems are defined by the Euclidean distance among their weight vectors [12]. MOEA/D only uses the neighboring sub-problem's information to solve the individual sub-problem. MOP can be defined as follows:

$$\max F(x) = (f_1(x), ..., f_m(x))^T \quad (1)$$

where $F(x)$ is the set of multiple objective function, which consists of m sub-objective function. Due to the Tchebycheff approach [13] could handle MOP with non-convex Pareto front, it is adopted as the decomposition method. After converting MOP problem by Tchebycheff approach, the j-th question is defined as follows:

$$\min g^{te}(x \mid \lambda^j, z^*) = \max_{1 \le i \le m}\{\lambda_i^j \mid f_i(x) - z_i^* \mid\} \quad (2)$$

where x belong to the space of decision variables, $\lambda^j = (\lambda_1^j, ...\lambda_m^j)^T$, $\lambda^1, ..., \lambda^N$ is a set of uniformly distributed weight vectors. z^* is a reference point, which is the optimal value of the objective function. Since g^{te} continuous about λ, when λ^i and λ^j close to each other, the optimal solution of $g^{te}(x \mid \lambda^i, z^*)$ is similar to $g^{te}(x \mid \lambda^j, z^*)$. Therefore, any information about g^{te} of the each neighbor or close to λ^i, could be used to optimize the objective function $g^{te}(x \mid \lambda^i, z^*)$. MOEA/D makes every generation of population to preserve several parameters, includes the population sub-problems number $x^1, ..., x^N$. x^i represent the solution of i-th sub-problem. F value of x^i is FV^i, and the range is $FV^1, ..., FV^N$. z_i is the optimal value of sub-objective function f_i, and $z = (z_1, ..., z_m)^T$. External population preserves the searched non-dominated solutions [14]. The process of algorithm framework is defined as follows:

Step 1 - Initialization: set $EP := \varphi$.

Step 1.1 - To calculate the Euclidean distance between any two weights, summarize the nearest T weights point of every weights.

Step 1.2 - For $i = 1, ..., n$, let $B(i) = \{i_1, ..., i_T\}$, where $\lambda^{i_1}, ..., \lambda^{i_T}$ are the N nearest weights vectors to λ^i.

Step 1.3 - Population initializing $x^1, ..., x^N$ by a specific multi-objective problems, to make $FV^i = F(x^i)$.

Step 1.4 - By a specific multi-objective problem initializes the optimal Parameter Standard $z = (z_1, ..., z_m)^T$.

Step 2 - Evolution: For $i = 1, ..., n$.

Step 2.1 - From the $B(i)$ selects randomly two sub-problems, the subscripts of sub-problems are k and l. Trough EA to integrate x^k and x^l, it generates new population individual y.

Step 2.2 - Improved: Depend on the problem needs, modify y to y' in characteristics and preferences.

Step 2.3 - Update z: For $j = i,....,m$, if $z_j < f_j(y')$ then $z_j = f_j(y')$.

Step 2.4 - Update neighbor: For $j \in B(i)$, if $g^{te}(y' \mid \lambda^j, z) \le g^{te}(x^j \mid \lambda^j, z)$ then $x^j = y'$, $F(x^j) = F(y')$.

Step 2.5 - Update EP: From EP remove vectors, which is dominated by the $F(y')$. If there is no population individuals or vectors can dominate $F(y')$ in EP, then $F(y')$ joins the EP.

Step 3 - Stopping Criteria: If the search process of the optimal solution meets the stop standard of setting previously, exit the search and output population, otherwise go to step 2.

3. Improved MOEA/D algorithm

During MOEA/D searches the population high-quality individuals, it lacks of preserving the high-quality individuals of present generation, eliminates the high-quality individuals of present generation, and decreases the algorithm's converges efficiency. Furthermore, due to the shared information of the neighborhood sub-problems in algorithm, the evolution individual is usually better than the neighbor individual. The several neighborhood individuals are replaced by new individuals. Resulting in finally obtaining few individuals in optimal solution set. It cannot provide more specific and precise solution for decision-makers.

For two defects in MOEA/D above, we propose an improved MOEA/D with particle swarm optimization (MOEA/D-PSO). On the one hand it preserves the high-quality individuals of present generation to improve the convergence and acquire the non-dominated solution set of Pareto optimal surface; On the other hand due to the weight of the each sub-problem in original algorithm has been fixed, the search direction has been identified to some extent. For further optimizing individual, within the search range of the local high-quality solution, MOEA/D-PSO tries to change the search direction and the search progress of the temporary optimal solution, depth optimization of the currently non-dominated solutions, so that each sub-problem can get better objective value. Furthermore, because the solution of MOEA/D-PSO is better, neighbor nodes have more high-quality solutions. Therefore, the new solutions replace few neighbor solutions, provide better and more accurate solutions for users. During these search process, MOEA/D-PSO conducts the appropriate operation according to individual fitness function value. Individuals previously search experience and past history search experience of groups, as well once before the search progress closely linked the current search progress of each individual. Improved part of the algorithm is as follows:

Step 1 - Preserve the original high-quality individuals: for $j = i,....,m$, if $z_j < f_j(y') < f_j(y)$ then $z_j = f_j(y)$;

Step 2 - Search for better population individuals, and update the search speed, $k = 1,...,\text{iter}$

$$v_j^{k+1} = r_1 v_j^k + c_1 r_2 (p_j - y_j^k) + c_2 r_3 (p_g - y_j^k) \quad (3)$$

$$y_j^{k+1} = y_j^k + v_j^{k+1} \quad (4)$$

Step 3 - Preserve individuals and sub-problem objective function value of optimal population: if $z_j < f_j(y^{iter})$ then $z_j = f_j(y^{iter})$;

where g represents the number of iterations. v_j is the search speed of j-th individual in the search space. y_j is j-th high-quality individual. p_j preserves j-th population individual's a historical optimal value. p_g preserves the entire population individuals' the historical optimal value in the iterative process. k is the current iteration number. $iter$ is setting historical maximum iterations number. r_1, r_2, r_3 are random number in the range [0,1]. c_1 and c_2 are accelerating factor.

4. WSN regional coverage and energy consumption model

4.1 Calculation principle of coverage

Assumed the monitoring area in two-dimensional plane, high-density randomly spill sensor nodes, these densely distributed nodes have same radius in perception and communication [15]. The total number of similar nodes are N in layout structure, and nodes position perceive through by its own equipment. Assumed perception radius is r_s, communication radius is r_c, when $r_c = 2r_s$ can better ensure the connectivity of WSN, and decrease the interference of external factors such as wireless. $c_i = \{x_i, y_i, r\}$ expresses the center coordinates in plane (x_i, y_i) and r is the perception radius of the sensor node. Assumed (x, y) is pixel point coordinate. The distance between sensor nodes and the objective pixel is calculated as follows:

$$d(c_i, p) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (5)$$

Therefore, probability of the sensor node C_i covered pixel (x, y) is calculated as follows:

$$p_{\text{cov}}(x,y,c_i)=\begin{cases}1 & \text{if } d(c_i,p)\le r_s-r_e\\ e^{\frac{-a_1\lambda_1\beta_1}{a_2\beta_2+\lambda_2}} & \text{if } r_s-r_e<d(c_i,p)<r_s+r_e\\ 0 & \text{otherwise}\end{cases} \quad (6)$$

where a_1 and a_2 are defined as follows:

$$a_1=r_e-r_s+d(c_i,p) \quad (7)$$

$$a_2=r_e+r_s-d(c_i,p) \quad (8)$$

Using Eq. (6, 7, 8) calculates the coverage rate that each sensor node covers one pixel point in the regional.

The joint coverage rate of every sensor node for this pixel point is calculated as follows:

$$P_{\text{cov}}(C_{con})=1-\prod_{c_i\in C_{con}}(1-p_{\text{cov}}(x,y,c_i)) \quad (9)$$

The maximum coverage rate in the monitored area is calculated as follows:

$$p_{area}=\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}P_{\text{cov}}(C_{con})}{m\times n} \quad (10)$$

Where r_e ($0<r_e<r_s$) is reliability parameters of the sensor measurement, λ_1, λ_2, β_1, β_2 are measurement parameters associated with the node's own characteristics.

4.2 Consumption analysis of node energy

Energy consumption in wireless sensor networks mainly focused on two aspects, energy consumption of communication and energy consumption of perception [16].Through a lot of theoretical and experimental studies show that only when $r_c>2r_s$ could guarantee the whole network node is connected. This paper with this topology, limit energy consumption of sending k bit data is $kE_{elec}+4k\varepsilon_{fs}R_s^2$. Energy consumption of accepting k bit data is kE_{elec}.

(1) Energy consumption of perception

In the monitoring area, sensor node's energy consumption of perception is usually proportional to the covered area, where m is a proportionality factor, and energy consumption of perception is calculated as follows:

$$E_s=m\pi R_s^2 \quad (11)$$

(2) Energy consumption of communication

In the actual application and operating environment, during sending the data, various reasons cause data conflicts and inevitably re-transmission operation, therefore we should consider the data sensor node's average re-transmission times. If p is the sensor node's data packet loss probability, n is the number of nodes, then the energy consumption of communication is calculated as follows:

$$E_c=E_T+E_R=\frac{1}{(1-p)^{n-1}}[2kE_{elec}+4k\varepsilon_{fs}R_s^2] \quad (12)$$

(3) In the network the minimum total energy consumption is the sum of perception and communication:

$$E_{total}=\sum_{i=1}^{n}(E_s^i+E_c^i)=\frac{2nkE_{elec}}{(1-p)^{n-1}}+(m\pi+\frac{4k\varepsilon_{fs}}{(1-p)^{n-1}})\times\sum_{i=1}^{n}(R_s^i)^2 \quad (13)$$

From these formula, the sensor node's packet loss probability p and perception radius decide total energy consumption. Solving multi-objective problem is pursuing to maximize regional coverage and minimize energy consumption. To convenient unified computing, multi-objective problems convert to maximization. Therefore, the energy consumption minimum value convert to the energy consumption maximization negative value, which is defined as followed $\max(-E_{total})$.

5. Experimental studies

In this part, by Matlab we compare the performance of MOEA/D-PSO with MOEA/D and NSGA-II in coverage and energy consumption, in order to study the coverage optimization of the MOEA/D-PSO.

5.1 Experimental setting

Energy parameters: wireless communication consumes energy E_{elec} = 20*nJ/bit*, free-space model ε_{fs} = 10*pJ/bit/m*^{2}, k = 4000bit, the best packet loss probability p = 0.05, set the energy threshold 1200*J* (40 nodes). Coverage parameters: assume the size of two-dimensional monitoring region is 25 m * 25 m, the sensor's coordinate system is (*x*, *y*) (1≦x≦25,1≦y≦25). Perception radius r_s =3 m, r_e = 0.5*r_s = 1.5 m. The initial stage in the monitoring area randomly select large number of redundant nodes, detect model parameters λ_1 = 1, λ_2 = 0, β_1 = 1, β_2 =1.5. Neighbor nodes T = 4. Particle swarm iterations Gp = 30. The node's number is expressed as N. The size of population expresses as *POP*. Regional node density expresses as De and population iteration times expresses as Ge [17]. The setting of network parameters on the Net1- Net4 show in Table 1:

Table 1. Network parameter settings

Sequence number	*N*	*POP*	*De*	*Ge*
Net1	18	100	0.0288	110
Net2	26	140	0.0416	110
Net3	33	170	0.0528	180

Net4	40	200	0.0640	180

5.2 Performance metrics

Performance metrics usually include the IGD-metric, non-dominated solutions, the Δ-metric, C-metric etc. Our simulation experiments will be mainly through the following three metrics to evaluate the performance:

Δ-Metric: In the Pareto-optimal set solutions, Δ measures the degree of spread of the diversity. In the two objectives, the value of a set of candidate solutions A is defined as follows:

$$\Delta(A) = \frac{d_h + d_l + \sum\left|d_d - \bar{d}\right|}{d_h + d_l + |A|\bar{d}} \qquad (14)$$

where d_h and d_l are two extreme Pareto optimal solutions in the target space, d_d is the distance between two neighboring solutions and $\bar{d}$ is the average of all the distribution. When $\Delta(A)$ =0, solutions spread uniformly in the target space, so $\Delta(A)$ is the lower the better.

Coverage (C)-metric: The metric is used to compare two sets of non-dominated solutions M and N The $C(M, N)$ is often considered as a quality metric, calculates the ratio of the non-dominated solutions in N dominated by the non-dominated solutions in M. $C(M, N)$ is defined as follows:

$$C(M,N) = \frac{\left|x \in N \mid \exists y \in M : y \succ x\right|}{|N|} \qquad (15)$$

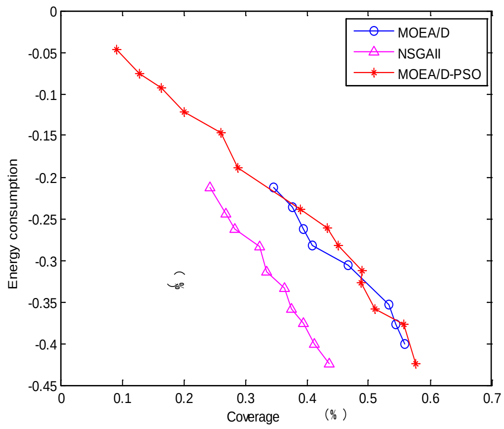
when $C(M,N)$=1 means that non-dominated solutions in N are completely dominated by the non-dominated solutions in M.

Non-dominated solutions (*NDS(S)*): This metric is generally considered discrete optimization problems, where represents the number of non-dominated solution in set S. *NDS(S)* is defined as follows:

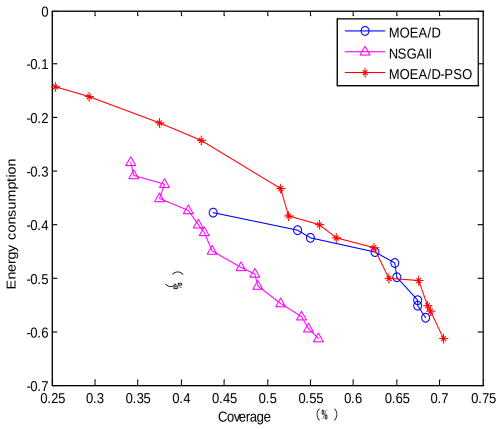
$$NDS(S) = |S| \qquad (16)$$

In order to provide more Pareto optimal solutions, it is better to obtain a high number of *NDS(S)*. However, a high number of *NDS* is not ideal in continuous optimization because the decision procedure becomes more complicated and more time consuming.

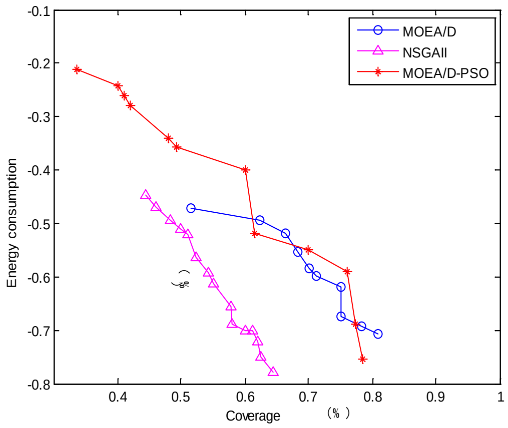
5.3 Simulation results



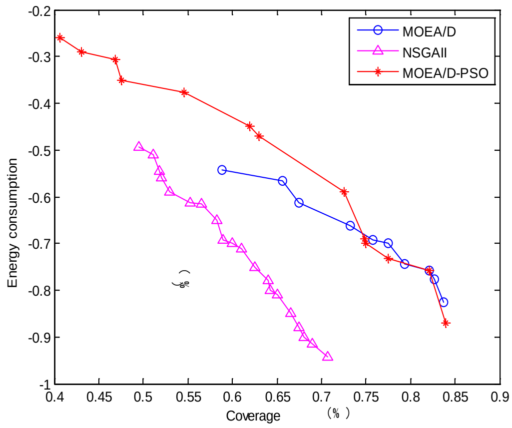
(Net1)



(Net2)



(Net3)



(Net4)

Figure 1. Performance of MOEA/D-PSO and MOEA/D and NSGA-II on Net1-Net4

Net1-Net4 respectively represent the performance of MOEA/D-PSO and MOEA/D and NSGA-II in different network environments parameter settings. By preserving high-quality individual of present generation and depth optimization of the local searched non-dominated solutions, MOEA/D-PSO makes the obtained non-dominated solutions closer to the multi-objective values. By Net1 and Net2 show that as increasing the number of sensor nodes, coverage rate is also increasing, simultaneously coverage rate will be stabilized. As increasing the number of population iterations, the numbers of non-dominated solutions are increasing. By Net3 and Net4 show that when the number of population iterations is increasing, the non-dominated solutions are evolved and decrease the number. When the number of iterations is too high, it could be also affecting the algorithm running time.

Table 2. Performance Comparison(E-MOEA/D, S-NSGA-II, P-MOEA/D-PSO)

Sequence number	Net1	Net2	Net3	Net4
Δ(E)	0.8813	0.8669	0.8150	0.7918
Δ(S)	0.8334	0.8214	0.8130	0.6891
Δ(P)	0.6864	0.7148	0.6990	0.6542
C(E,P)	0.1432	0.2308	0.2727	0.1429
C(P,E)	0.6253	0.5556	0.6000	0.4000
C(S,P)	0.0	0.0	0.0	0.0
C(P,S)	1.0	1.0	1.0	1.0
NDS(E)	8	9	10	10
NDS(S)	10	15	15	20
NDS(P)	14	14	12	13

It can be seen from the performance comparison of Table 2, in different networks, *Δ(P)<Δ(E)* and *Δ(P)<Δ(S)*, show that Pareto front distribution of MOEA/D-PSO is more uniform. *C(P,E)>C(E,P)* and *C(P,S)>C(S,P)* show that Pareto solution of MOEA/D-PSO is better, and non-dominated solutions is more effective. *NDS(P)>NDS(E)* shows that MOEA/D-PSO has more non-dominated solutions. Although *NDS(S)* compared with the *NDS(P)* is almost the same or even more, solutions of NSGA-II is too dense. So the uniformity and ductility of Pareto front is not ideal. Based on the above, the non-dominated solutions of MOEA/D-PSO is better in Pareto front's distribution, and could provide more diversely and accurately solutions, and WSN could get wider coverage, and consume less energy.

6. Conclusions

MOEA/D is the new multi-objective evolutionary algorithm framework. Based on MOEA/D, we propose the multi-objective evolutionary algorithm based on decomposition with particle swarm optimization (MOEA/D-PSO) in this paper, analyze coverage rate and energy consumption. Compared with MOEA/D and NSGA-II, multi-objective solution sets of MOEA/D-PSO is better. The uniformity and diversity of solution sets are better. At the same time, coverage area and the effect of energy consumption are better ideal.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. E050603), Guangxi Universities Scientific Research Projects (No. YB2014157).

References

1. Li H, Zhang Q F. (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2), p.p.284-302.
2. Wu L G, Wang A, Xia M. (2014) Improved SPEA2 based on local search. *Application Research of Computers*, 31(9), p.p.2617-2619.
3. Zhang Q F, Li H. (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), p.p.712-731.
4. Zhang Y, Hu S B, Wu J L, et al. (2014) Multi-objective optimization of double suction centrifugal pump using Kriging metamodels. *Advances in Engineering Software*, 74, p.p.16-26.
5. Gong M G, Jiao L C, Yang D D, et al. (2009) Research on Evolutionary multi-objective Optimization Algorithm. *Journal of Software*, 20(2), p.p.271-289.
6. Krzysztof M. (2014) The effects of asymmetric neighborhood assignment in the MOEA/D algorithm. *Applied Soft Computing*, 25, p.p.97-106.

7. Carvalho R, Saldanha R R, Gomes B N, et al. (2012) A Multi-Objective Evolutionary Algorithm Based on Decomposition for Optimal Design of Yagi-Uda Antennas. *IEEE Journals & Magazines*, 48(2), p.p.803-806.
8. Ma X L, Liu F, Qi Y T, et al. (2014) MOEA/D with opposition-based learning for multiobjective optimization problem. *Neurocomputing*, 146, p.p.48-64.
9. Muhammad A J, Rashida A K. (2013) A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D. *Applied Soft Computing*, 13(1), p.p.128-148.
10. Tan Y Y, Jiao Y C, Li H et al. (2013) MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. Computers & Operations Research, 40(6), p.p.1648-1660.
11. Zhou A M, Qu B Y, Li H, et al. (2011) Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), p.p.32-49.
12. Wang Y N, Wu L H, Yuan X F. (2010) Multi-Objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Computing*, 14(3), p.p.193-209.
13. Ma X L, Liu F, Qi Y T, et al. (2014) MOEA/D with Baldwinian learning inspired by the regularity property of continuous multi-objective problem. *Neurocomputing*, 145, p.p.336-352.
14. Abido M A. (2009) Multiobjective particle swarm optimization for environmental/economic dispatch problem. *Electric Power Systems Research*, 79(7), p.p.1105-1113.
15. Shi C, Yan Z Y, Shi Z Z, et al. (2010) A fast multi-objective evolutionary algorithm based on a tree structure. *Applied Soft Computing*, 10(2), p.p.468-480.
16. Konstantinidis A,Yang K. (2012) Multi-objective energy-efficient dense deployment in Wireless Sensor Networks using a hybrid problem-specific MOEA/D. *Applied Soft Computing*, 12(7), p.p.1847-1864.
17. Liu Y M, Niu B, Zhao Q Z. (2011) Particle swarm optimizer simulation research of multi-objective optimization problems. *Application Research of Computers*, 28(2), p.p.457-459.