대 한 조 선 학 회 논 문 집
제46권 제1호 2009년 2월

Journal of the Society of Naval Architects of Korea
Vol. 46, No. 1, pp. 87-95, February 2009
DOI: 10.3744/SNAK.2009.46.1.087

유전자 알고리즘을 이용한 B-spline 곡면 피팅

Tat-Hien Le*, 김동준†*, 민경철**, 표상우***

부경대학교 조선해양시스템공학과*
부경대학교 조선해양시스템공학연구소**
리얼웹***

B-spline Surface Fitting using Genetic Algorithm

Tat-Hien Le*, Dong-Joon Kim†*, Kyong-Cheol Min** and Sang-Woo Pyo***

Dept. of Naval Architecture and Marine Systems Engineering, Pukyong National University*
Naval Architecture and Marine Systems Engineering Research Center, Pukyong National University**
Realweb Co., Ltd.***

Abstract

The applicability of optimization techniques for hull surface fitting has been important in the ship design process. In this research, the Genetic Algorithm has been used as a searching technique for solving surface fitting problem and minimizing errors between B-spline surface and the ship's offset data. The encoded design variables are the location of the vertex points and parametric values. The sufficient accuracy in surface fitting implies not only various techniques for computer-aided design, but also the future production design.

※Keywords: Vertex point(조절점), Surface fitting(곡면 피팅), Optimization technique(최적화 기법), Genetic algorithm(유전자 알고리즘), Encoded design value(설계값 부호화)

1. Introduction

In ship design process, a hull form is designed by cross sectional curves at the basic design stage. The hull surface reconstruction from these offset points is more important than before, because the accurate modeling of hull surface has been considered as important for the effective support of the ship production as well as the numerical ship performance analysis. The approach of B-spline surface construction also has received lots of attention in recent years because NURBS are established as a standard in CAD program.

접수일: 2008년 8월 7일, 승인일: 2009년 1월 2일
†교신저자: djkim@pknu.ac.kr, 051-629-6614

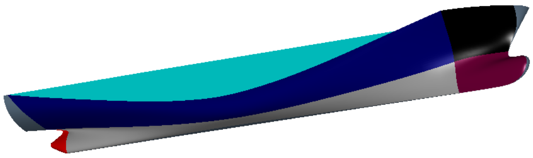


Fig. 1 The whole ship hull with many patches

In some researches, in order to ensure the surface quality of complicated shape, the ship hull has to be divided into some patches (Westgaard and Nowacki 2001, Lee and Kim 2004), as shown in Fig. 1. However, disadvantages are caused by the continuing patches problem, inconvenient handling in later design(CAE), and difficulties in automatic fairing. Instead of subdividing a hull form into patches, it is advantageous to work with the only one B-spline surface for ship hull. In this research, the reconstruction of B-spline surface from the given data points is considered.

For B-spline surface fitting method, there are two approaches, matrices inversion and vertex point estimation for surface approximation. In matrices inversion, many conventional methods have been proposed(Choi 1991, Yoon et al. 1985). However, the matrices inversion gets the ill conditioned problem, and it also must avoid round off error magnifications in back-substitution calculation and large storage capacity(Mathews and Fink 1992). Another approach, surface approximation method is developed by Birmingham and Smith(1998).

In this research, B-spline surface is generated in a fully automatic approach of the location of the vertex point and the proper parameters by using the Genetic Algorithm (GA). We assume that the preliminary hull surface is a gene type and GA technique can be applied for the convergence toward a good solution. This procedure improves the stability and makes less error than other methods.

2. Overview of the fitting algorithm

2.1 Surface fitting basis

The idea of using B-spline surface in shipbuilding is initiated by Rogers(1977). Mathematically, the shape of the B-spline surface is defined by knot vector, degree, B-spline weight value, and vertex points (in common case : weight = 1 and degree = 3 for cubic spline). At this point, the discussion is confined to the regular mesh. Fig. 2 illustrates the relationship between surface data points and vertex points B_{ij} in 3D coordinates.

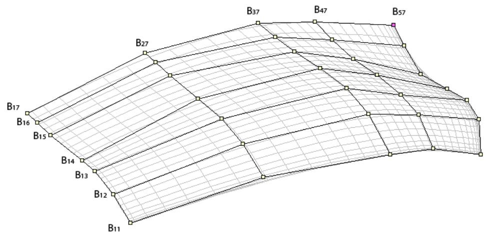


Fig. 2 The relationship between surface data points and vertex points

B-spline surface point D, moving in space with 2 degrees of freedom, u and w, is given by Eq. (1).

$$D(u,w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,l}(w) \quad (1)$$

where $N_{i,k}(u)$ and $M_{j,l}(w)$ are the basis functions of B-spline surface, and $B_{i,j}$ are the vertex points of surface.

Eq. (1) shows that the surface can be generated if the location of the vertex point is changed. In the GA process, the shape of the surface is changed automatically in the following way.

$$N_{i,k}(u) = \begin{cases} 1 & \text{if } x_i \le u < x_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,k}(u) = \frac{(u-x_i)N_{i,k-1}(u)}{x_{i+k-1}-x_i} + \frac{(x_{i+k}-u)N_{i+1,k-1}(u)}{x_{i+k}-x_{i+1}}$$

$$M_{j,l}(w) = \begin{cases} 1 & \text{if } y_i \le w < y_{j+1} \\ 0 & otherwise \end{cases}$$

$$M_{j,l}(w) = \frac{(w-y_j)M_{j,l-1}(w)}{y_{j+l-1}-y_j} + \frac{(y_{j+l}-w)M_{j+1,l-1}(w)}{y_{j+l}-y_{j+1}} \quad (2)$$

In Eq. (2), x_i and y_j are the knot values and $B_{i,j}$ are the vertex points of B-spline for surface fitting. The values of x_i, y_j are elements of the knot vector satisfying the relation $x_i \le x_{i+1}$.

When the vertex points are relocated in B-spline surface by using GA technique, the parametric values also need to be managed. Therefore, the parameterization method is proposed to change the parametric values proportional to the chord distances between vertex points.

2.2 The procedure of finding the nearest point

In order to get a good fitted surface, the error between the given data point and fitted surface should be minimized as small as possible. In this case it is very important to find the corresponding point on the fitted surface to the given data point. For this corresponding point, we consider the nearest point on the surface D(u,w) to the given data point G. The value of D(u,w) can be obtained by Jacobian inversion method(Choi 1991). Through the Jacobian inversion process, we can find out the parametric value u, w.

First, we assume the initial parametric value $r_0(u,w)$ for the parametric surface and G is the given data point. From an initial parameter value $r_0(u,w)$, Newton-Raphson iteration can be used to reduce the error between a proper parametric value $r^*(u,w)$ and a given data point G. The Jacobian is followed by Choi (1991) (see Fig. 3).

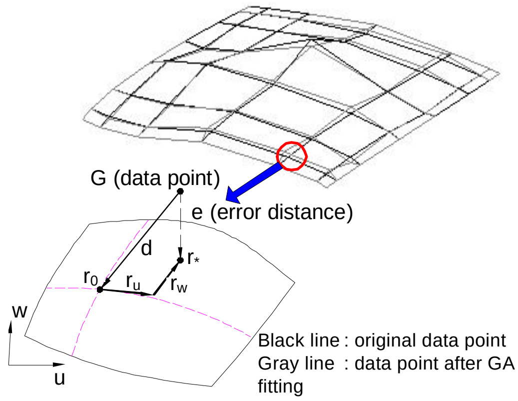


Fig. 3 Jacobian Inversion between the given data point G and closest point r* on GA surface

2.3 Fitness function

The fitness function is the normalized root mean squared error Q (the difference between the nearest point D(u,w) on the fitted surface and the given data point G in a non-dimensional form). The procedure continues until the convergence criteria are satisfied. The fitness function is given by Eq. (3).

$$Q = \frac{\sum_{i=1}^{N} \sqrt{\{D(u,w) - G\}^2}}{|G_{\max} - G_{\min}|} \approx precision \quad (3)$$

where, $|G_{\max} - G_{\min}|$ is a range of observed values.

3. Genetic algorithm procedure

3.1 Encoding

In the GA mechanism, each individual is considered as a B-spline surface created randomly in an initial population. The evolution process describes the surface generations which are encoded by design variables. Fig. 4 shows the B-spline surface generated by vertex points and the encoding process.

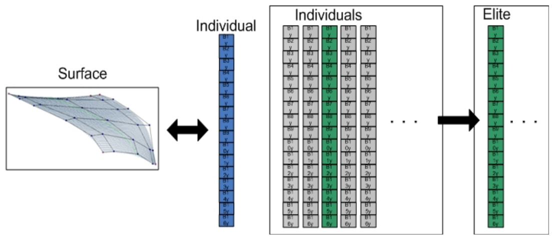


Fig. 4 The vertex point of B-spline surface encoding procedure

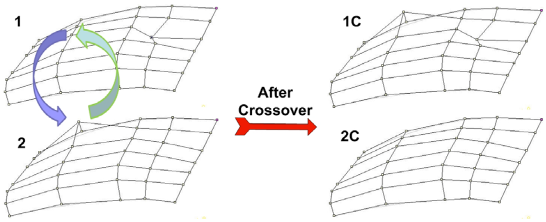
3.2 Reproduction process

In reproduction process, individuals are chosen for the next generation of the population based on its fitness value. Three types of reproduction are roulette wheel selection, ranking-based selection, and tournament selection(Goldberg 1989).

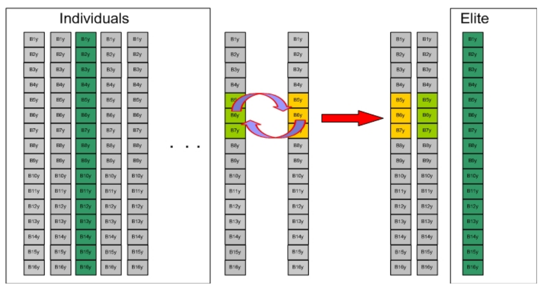
The steady-state selection is not particular method for selecting(Whitley and Kauth 1988). Each individual in population has a fitness value. First, the individuals are sorted from lowest to highest fitness value. Main idea of this selection is that the good 80% individuals in population(good group) are selected to produce and the rest(bad group) are discarded. Also, in the good group, a few best individuals will be retained at each generation by Elitism method. We recommended that the number of elite individuals is 5% the number of individuals in good group. In addition, the bad group in current population will be replaced by the individuals, which are executed by the crossover and mutation process, in the good group of current population. A replacement /deletion strategy is more practical than other methods because some characters of good individuals can be lost if they are not selected to produce or if they are destroyed by crossover or mutation.

3.3 Crossover

Crossover is explorative. The encoded design variables are arranged in a dimensional form as a "string"(individual). The crossover process means that the individuals divide a string into sub-strings and swap the sub-strings between individuals randomly to create new individuals. Fig. 5 shows that the individual 1 received a new sub-string from individual 2 to generate a new individual in next population.



(a) Vertex points location crossover



(b) Individuals crossover in GA

Fig. 5 Crossover encoding procedure

We set p_c as a probability of Crossover of each two individuals at the generation. Then :

$$\begin{cases} p_c = (f' - f_{\min})/(\bar{f} - f_{\min}),\ f' \le \bar{f} \\ p_c = 1,\ f' > \bar{f} \end{cases} \tag{4}$$

Here : $f_{\min}$ is the minimum fitness in the population,

$\bar{f}$ is the average of the fitness values in the population.

f' is the smaller of the fitness values of the individuals to be crossed.

Crossover process occurs in some crossover probabilities p_c. The higher the values of p_c, the quicker new solutions are introduced into population. As p_c increase ; however, solutions can be disrupted faster. We can adaptively control the probability of crossover (Srinivas and Patnaik 1998). Eq. (4) ensures that all solutions having random numbers smaller than p_c will be swapped.

3.4 Mutation

Mutation is exploitation; it creates a small deviation on the surface. The role of mutation in GA has been preventing premature convergence of the solution (see Fig. 6). Therefore, the surface quality improves a good fitting result during each time. As we discussed in section 2.1, the B-spline surface can be generated if the location of the vertex point is changed. In our method, the location of the new vertex point is created by moving in a circle of small radius δ around the old vertex point. We recommend that δ should be less than 5% of the distance between old vertex point and the nearest neighboring vertex point. The variable design has a deviation with the deviation size δ as given by eq. (5).

$$[B_{new}] = [B_{old}] + \delta \tag{5}$$

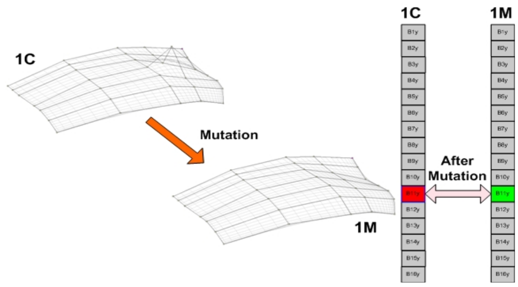


Fig. 6 Encoding procedure of vertex point location mutation and individual mutation in GA

The role of mutation in GA has been restoring lost genetic material into the population to prevent the convergence of the GA to suboptimal solutions. Mutation involves the modification of the value of each gene with some probabilities p_m. Mutation occurs only in some probability p_m(Srinivas and Patnaik 1998).

We assumed p_m as a probability of mutation of each individual at the generation. Then :

$$\begin{cases} p_m = 0.5(f - f_{\min})/(\bar{f} - f_{\min}),\ f \le \bar{f} \\ p_m = 0.5,\ f > \bar{f} \end{cases} \tag{6}$$

Here, f is the fitness value of an individual, f_{min} is the minimum fitness value in the population, $\bar{f}$ is the average of the fitness value in the population. Eq. (6) ensures that all of solutions having random number smaller than Mutation probability p_m be mutated.

In our approach, we applied the p_c and p_m adaptively in response to the fitness value. The value of p_c and p_m are increased when the solution get stuck at the local optima and are decreased when the solution is scattered in the population.

3.6 Genetic algorithm procedure

Fig. 7 shows the GA generation procedure.

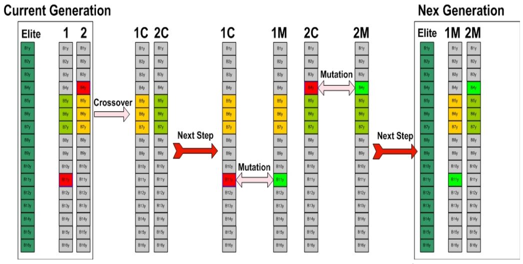


Fig. 7 Total encoding Procedure

Step 1. Input the 3D given data point

Step 2. Create an initial population by using uniform vertex point location

Step 3. For each individual in population, the surface points are computed based on vertex point above.

Step 4. Obtain the Fitness value and Do Reproduction process (Choose Reproduction rate : 80%)

Step 5. Test for convergence criteria (precision or generation times), if it's not satisfied, the GA process will continue. Otherwise, GA process will stop.

Step 6. Do Crossover in each 02 individuals

Step 7. Do Mutation (change the location of the vertex point) for each Individual and go back Step 4.

4. Experimental results

By using GA technique for surface fitting problem, we obtained the following control variables and experimental result.

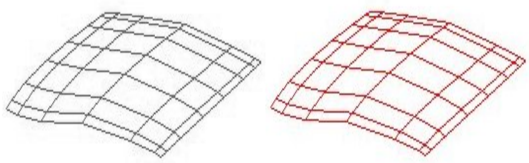
Control variables are population size, crossover probability, and mutation probability. When population size is increased, the diversity and computation time for each generation is also increased. Besides, if crossover probability value is increased, the opportunity for recombination is increased but also disruption of good combination might be occurred. In mutation step, if mutation probability value is increased, the randomly search will be closer and help to introduce new gene or reintroduce the lost gene.

Our GA technique was written in C++ and run on a 2.4 GHz processor. The simple surface, geometrically complicated shape and yacht surface without keel are presented for high accuracy measurement.

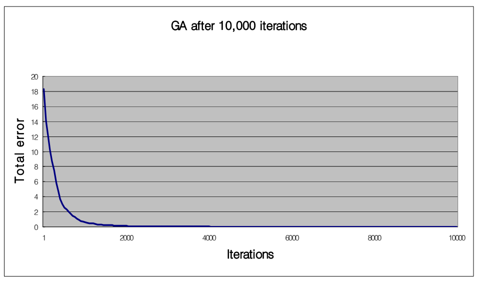
4.1 Simple surface

The mesh consist of 7×8 given data points. When the mesh points were simple, the result showed a good surface quality during small computation time (see Fig. 8a).

At the first iteration, the total errors are bad in general. Therefore, the location of vertex points were changed for trial, and then converged at 400th iteration as shown in Fig. 8b. In Table 1, the result shows the best fitting normalized error value after 10,000 iterations is close to zero.



(a) The given data points and surface data points after fitting in GA



(b) Total errors as a function of the generations

Fig. 8 Simple surface fitting test by using GA

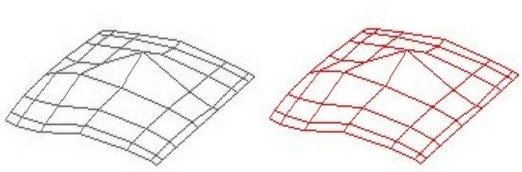
Table 1 Normalized errors between the given data points and the simple surface points

No. of vertex points	GA time	Normalized errors
7×8	3min	0.00012822

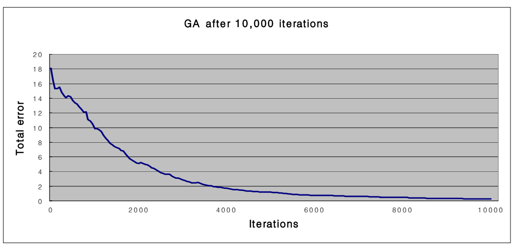
4.2 Complicated shape

The complicated mesh consist of 7×8 given data points.

Fig. 9a shows the best fitting result at 10,000th iteration and Fig. 9b shows the total error curve is oscillated which can be explained as follows. In the complicated shape, there are some discontinuous points. This was due to task difficulty of searching for a model function. At the first 2,000 iterations, the locations of the vertex points were changed during GA process, the error value decreased and increased slightly. Finally, our normalized error value converged at the 10,000th iteration (see Table 2).



(a) The given data points and surface data points after fitting in GA



(b) Total errors as a function of the generations

Fig. 9 Complicated surface fitting test by using GA

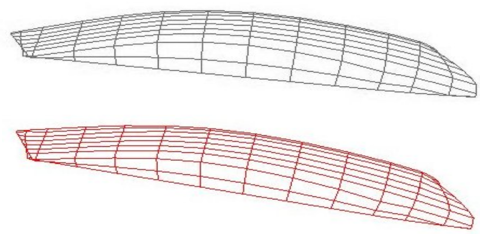
Table 2 Normalized errors between the given data points and the complicated surface points

No. of vertex points	GA time	Normalized errors
7×8	3min	0.001461274

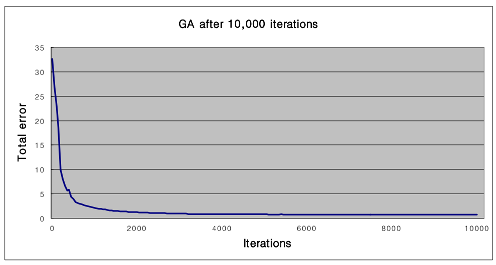
4.3 Yacht hull surface

To further evaluate our method, yacht surface were obtained.

L_{OA}:12m, Beam:2.685m, Draft:0.542m.



(a) The given data points and surface data points after fitting in GA



(b) Total errors as a function of the generations

Fig. 10 Yacht surface fitting test by using GA

Fig. 10 shows the result of surface fitting. In this case, the yacht surface shows the best fitting after 10000 iterations in Fig. 10a, and the total error value converged at 4000th iteration in Fig. 10b. The normalized error value after 10,000 iterations is computed in Table 3

Table 3 Normalized errors between the given data points and the yacht surface points

No. of vertex points	GA time	Max error (mm)	Normalized errors
10×14	15min	18.135	0.00448344

Generally speaking, our GA method is good for global searching. The result depends on the iteration time in which the convergence is guaranteed by the GA process.

5. Conclusions

The surface fitting by using GA technique has been applied to many cases of parametric hull form with high accuracy. In most major CAD systems, such as CATIA, AutoCAD and MaxSurf packages, the B-spline surface can be generated by specifying the knot values and parametric values or skinning through the data points. In our GA, only one rectangular B-Spline surface was generated in a fully automatic approach to the location of vertex point and parametric value until satisfied precision is reached. Different from other traditional methods, matrices inversion and continuity problems between surfaces are not necessarily required in this method. In conclusion, it is more effective on surface accuracy rather than other methods.

In ship design process, the hull form is designed by the irregular offset points, especially in bulbous bow part and stern part. In the future trends, we believe that the B-spline surface fitting using GA technique can be applied for irregular data points.

Reference

- Birmingham, R. W. and Smith, T.A.G, 1998, “Automatic hull form generation: a practical tool for design and research,” Proceedings of the Seventh International Symposium on Practical Design of Ships and Mobile Units, pp. 281-287.
- Choi, B.K., 1991, Surface modeling for CAD /CAM. Elsevier Science.
- Goldberg, D.E., 1989, Genetic algorithm in search, optimization, and machine learning, Addison-Wesley.
- Lee, J.H. and Kim, D.J., 2004, “A Study on the Surface Modeling of Hull Form for General purpose CAD program”, Journal of the Society of Naval Architectures of Korea, Vol. 41, No. 1, pp. 75-81.
- Mathews, J.H. and Fink, K.D., 1992, Numerical Methods Using Matlab, Pearson Prentice Hall.
- Rogers, D.F., 1977, "B-spline curves and surfaces for ship hull definition," SCAHD77, pp. 1-25.
- Srinivas, M. and Patnaik, L.M., 1998, “Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms,” IEEE Trans Syst, Vol. 24, No. 4, pp. 656-667.
- Westgaard, G. and Nowacki, H., 2001, "Construction of Fair Surfaces over Irregular Meshes," Journal of Computing and Information Science in Engineering, Vol. 1, No. 4, pp. 376-384.
- Whitley, D. and Kauth, J., 1988, “GENITOR: A different Genetic Algorithm,” Proc. of the Rocky Mountain Conf. on Artificial Intelligence, pp. 118-130.
- Yoon, B.H., Sur, S.W., Kim, W.D. and Kim, K.W., 1985, “Representation of Ship Hull Surface Using B-Spline,” Journal of the Society of Naval Architectures of Korea, Vol. 22, No. 3, pp 19-26.


<Tat-Hien Le>


< 김 동 준 >


< 민 경 철 >


< 표 상 우 >