Use of GA based Approach for Engineering Design Through WWW

Shu-Tan Hsieh
Department of Accounting
National Kaohsiung University of Applied Sciences
415 Chien Kung Road, Kaohsiung 807
Taiwan
shutan@cc.kuas.edu.tw

Abstract: - Today's industrial environment requires engineering design to be achieved by geographically distributed engineering teams who may work on different computer platforms, so the analogy can be presented as the distributed constraint optimization problems. This paper presents an agile approach that carries out a concurrent optimization of a product design and its associate constraint satisfaction in manufacturing perspective. Also, the approach has been implemented through the World Wide Web (WWW) regardless of the geographical constraints and different platforms used. In this paper, the hybrid evolution computation (EC) approaches combing genetic algorithm and stochastic annealing algorithms are applied to find optimal or near optimal solutions for two engineering design cases. The main contribution of this paper is to provide an agile approach for solving the engineering design problem which is modeled by the nonlinear programming model, and the approach is implemented through the WWW regardless of the geographical constraints and different platforms used. Experimental results are presented to exhibit the superior performance of the proposed methodology.

Key-Words: - Evolutionary computation, genetic algorithm, stochastic annealing, nonlinear programming, world wide web

1 Introduction

To remain competitive, industrial organizations are continually faced with many challenges to reduce product development time, improvement product quality, and reduce production costs and lead times, etc. Because of the reasons, it is necessary for the agile corporations to rapidly respond to the changes of market so that it is crucial to develop the solution approaches for the purpose of supporting decision making quickly. Also, today's industrial environment requires design to be achieved by geographically distributed engineering teams who may work on different computer platforms, so the analogy presented as the distributed constrained optimization problems should be solved. Firstly, critical to the success of overcoming the above difficulties is the development of model-based system that provides the computational functions or decision technologies and they can be used in analytical decision making of the constrained optimization problems. Secondly, remote execution of computational software is equivalent to a remote procedure call [3], and the HyperText Transmission Protocol (HTTP) servers used on the WWW provide such a facility through the Common Gateway Interface (CGI) [15]. When the system of server is developed, the system mediates between users and servers. The proposed system is able to address the transport of input and output data, providing interoperation via a standardized communication language between clients and servers. It allows engineering teams to contact the system through WWW browsers regardless of the geographical constraints and computer platforms for obtaining the optimal solution(s) remotely. The agile approach of product design should have the capability of quick model analyzing for decision-making, and support the solutions through the WWW to the decision-makers who may be geographically separated and operate on differing computer platforms.

Many engineering design problems can be formulated as the mathematical models in the form of nonlinear programming. Recently, Genetic Algorithms (GAs), originally developed by [9], and the related methods have been widely studied and applied to solve a variety of optimization problems, usually of a combinatorial nature. Owing to numerous reports of successful applications of these innovative algorithms, GA based approaches have attracted more recent attention than most other heuristic methods. These approaches have been widely applied to the nonlinear and mixed integer programming problems by [6] [5] [10] and [17], and their effective and efficient solutions were reported.

In this research a hybrid evolution computation (EC) is used as a solution searching approach in the system to solve nonlinearly constrained engineering design problems including the reliability design problem of life support system in a space capsule [18] [19] [24]and the pressure vessel design problem [7] [11] [12] [13] [21] [22] [23]. Numerical examples indicate that hybrid evolutionary computation approaches perform well for engineering design problems considered in this paper. In particular, as reported, solutions obtained by hybrid EC approaches are as well as or better than previously best known solutions. In general, treating decision technologies as information services that can be accessed over an information network is the fundamental idea explored in this paper. Use of Common Gateway Interface (CGI), which can be used to allow design teams to use decision technologies (the hybrid EC approaches) made available by providers. Remote computation can easily be launched by a user equipped with a web browser by supplying parameters that control the computation via HTML forms. For example, in the cases of this paper, engineering teams can remote execution of the proposed approach which helps teams find the optimal design parameters based on the requirement of their design.

Remote execution completely solves the difficulties of the collaboration between different teams because of the geographic distance and the compatibility of computer platform problems. The engineering teams never need to actually own or install a copy of the software. From the enterprise's point of view, it solves the heterogeneity and version management problems. Also, it helps the quick response to the changes of market for reducing the time to market.

The rest of the paper is structured as follows: In the next section, the Remote execution through the WWW is presented. Section 3 presents the EC approaches including GA, SA-GA, and MCA-GA. Two applications are introduced and the corresponding computational results using the EC approaches are compared with those using the methods in the previous literatures. Section 5 provides our conclusions.

2 Optimization Approach Over the Internet

The interaction between WWW servers and clients includes the uses of HTTP/CGI, Java or anonymous telnet servers. If the anonymous telnet server is applied, only the static files/documents are transmitted from server to client. However, this way is not able to achieve the purpose of agile collaborative engineering design. Currently, only the uses of HTTP/CGI or Java are the feasible ways to do so. It is makes the system be able to solve the design tasks in the form of sophistical mathematical programming and the execution is over the Internet. The WWW server processes data via HTTP/CGI in response to input given by users from the client browser. Java involves the download of an executable program to the user's machine. Since compiled Java programs can be executed on any platform as long as it has an installed Java environment, it has quickly become the standard language to provide downloadable software for client-side computing in the WWW. This is made possible using a Web browser that integrates a Java interpreter required to execute the Java programs.

In this study, the interaction between WWW servers and clients is the uses of HTTP/CGI to activate the GA based system for deciding the design parameters optimally. The basic idea is illustrated in Figure 1. The engineering designers can send their design parameters as the input which can be sent to the Web Server and GA based system can find out the optimal design solutions based on the design parameters given from engineering teams or users. Another focus in this research is the hybrid EC approach has been applied as the problem solver. The methodology of the evolutionary algorithms is introduced in next section. We believe that the hybrid EC and the integration with the use of web is a concept that could be developed with useful procedures, metrics, and supporting software in a way that is to achieve the goal of agile approach in design for manufacturing.

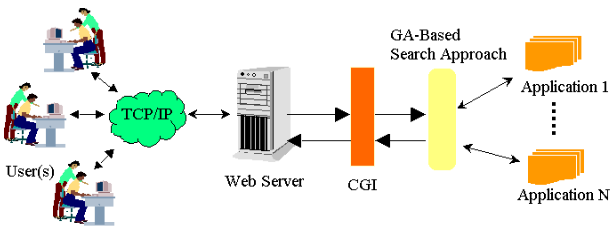

Fig.1 Basic idea of the client/server to activate the system

3 GA-based Optimization Approach

GAs are efficient search methods based on the principles of natural selection and population genetics in which random operators on a population of candidate solutions are employed to generate new points in the search space [8]. For any GA, a so-called chromosome representation is needed to describe each individual in the population of interest. Each individual or chromosome contains a sequence of genes from a certain alphabet. Although the alphabet was limited to binary digits in Holland's original design [9], other very useful problem-specific representations of an individual or chromosome for function optimization have also been proposed.

GAs can search the solution space efficiently to find an optimal or near optimal solution by the use of evaluation and genetic operator functions to maintain the useful schemata in the population, in which a schema with a higher fitness will have higher probability of survival in each generation and thereby a higher probability of generating offspring. Improved solutions are often found among the new offspring, since they have an inherently good schema, i.e., one that remains in the population over generations. This characteristic has been discussed in detail by [16].

Next, a brief introduction of the basic GAs is presented; further described by [8] and [16]. In GAs, the most interesting aspect of evolution, which includes reproduction, crossover and mutation etc. is that of natural selection, which can be accomplished by the following steps:

Step 1. Randomly generate an initial population of chromosomes.
Step 2. Evaluate the fitness function for each individual in the population.
Step 3. If the stopping criteria have been achieved, then stop; else, proceed to step 4.
Step 4. Perform reproduction, crossover and mutation within the population.
Step 5. Form the new generation from the individuals resulting from step 4. Go to step 2.

In our implementation, all solutions will be represented by strings of binary digits. Each string consisting of substring includes all the decision variables. The details have been described in Section 3.3.

The effectiveness of GA depends on complementary crossover and mutation operators. The crossover operator determines the rate of convergence, while the mutation operator makes the GA search jump from local optimum, thus avoiding the premature convergence to a local optimum. It is not very adequate to directly apply GA through WWW. Because the solution searching speed of GA is more than others although the superior performance is better than others. For this reason, more efficient methodology is needed. The hybrid evolutionary optimization algorithms combining GA and stochastic annealing algorithms have been applied to enhance effectiveness of solution searching.

3.1 The SA-GA Hybrid Algorithm

The hybrid SA-GA algorithm that combines the GA and SA proposed by [14]. It is a general purpose procedure which carries forward several strengths and leaves behind several weaknesses of genetic algorithms and simulated annealing [14]. The aforementioned geometric annealing schedule doesn't strictly ensure asymptotic convergence to a global optimum as the logarithmic annealing schedule does, but it is much faster and has been found to yield very good solutions in practice [4] [20] [2]. Instead of using the roulette wheel method in the selection process, the replacement between each parent and its offspring is decide by the possibility of Boltzmann trial. The convergence criterion used is the same as that for the GA. In the algorithm below, *T* is temperature, and *n* is population size.

Let *T* to be a sufficiently high value
Step 1. Generate an initial population randomly.
Step 2. Repeatedly generate each new population from the current population by using a roulette-wheel selection as follows;
for *i = 0* to *(n-1)*, where the *n* is the population size.
{
(1) From parents *P(2i)*and *P(2i+1)* in the mating pool generate offspring *C(2i)* and *C(2i+1)* using a recombination operator (such as crossover), followed by a neighborhood operator (such mutation).
(2) Replace parent P(i+1) in G with offspring C(i+1) with probability using Boltzmann decision function represented by B(T).
(3) Where B(T)=$1/(1+e^{(E_{parent}-E_{child})/T})$
(4) Periodically lower T, i.e., reduce the Temperature (T) using the annealing function T. *Where T =* αT
}
Step 3. Stop while the convergence criterion is met. Otherwise go to Step 2.

3.2 Microcanonical Annealing (MCA)

MCA has been shown to converge to a global minimum with unit probability given a logarithmic annealing schedule[1]. It models a physical system whose total energy is always conserved [2].

The basic algorithm of MCA has been illustrated by [2]. If C(*i*)<P(*i*) then C(*i*) is accepted as the new solution. If C(*i*)$\geq$P(i) then C(*i*) is accepted as the new solution only if $E_k \geq$ C(*i*)-P(*i*). If C(*i*) $\geq$ P(*i*) and $E_k <$ C(*i*)-P(*i*) then the current solution P(i) is retained. In the event that C(*i*) is accepted as the new solution, the kinetic energy demon is updated $E_k^{n+1} = E_k^n + (P(i) - C(i))$ to ensure the conservation of the total energy. The energy, E_k, is annealed in a manner similar to the temperature parameter T in simulated annealing. The energy provides an extra degree of freedom which helps MCA jump out from the local optima to the global optima. The implementation of the algorithm in this research is described as follows:

Step 1. Generate an initial population *(G)* randomly and let E_k be a high value.

Step 2. Generate each new population in a mating pool from the current population by using a roulette-wheel selection as follows;

for *i = 1* to *n*, where the *n* is the population size. Do:

{

(1) From a pair of parents in the mating pool generate a pair of offspring using the GA recombination operator (such as crossover), followed by a neighborhood operator (such mutation).

(2) ***if*** *C(i)<P(i)*,
add *C(i)* to the next generation *G_new*.
else if *C(i)-P(i)<* E_k while *C(i)>P(i)*, add *C(i)* to the next generation *G_new* and let $E_k = E_k + P(i) - C(i)$.
otherwise the current solution *P(i)* is retained in the next generation *G_new*.

}

Step 3. Let *G* = *G_new*; reduce the energy using annealing function $E_k = A(E_k)$.

Step 4. Stop while the convergence criterion is met. Otherwise go to Step 2.

In our implementation, $E_k = E_k + P(i) - C(i)$ is true, only if *C(i)* is greater than *P(i)* and the difference between *C(i)* and *P(i)* is less than E_k.

3.3 Solution representation

In our implementation, solutions of the optimization problem will be represented by strings of binary digits, each string consisting of four substrings for each variable x_i (where *i* = n, i.e., the number of the decision variables). Each substring in turn consists of a binary substring representing either the continuous or discrete value of the engineering design variable. The decision variables may include real numbers and/or integer. The representation of real-numbers in our GA approach is the same as that described by [16]. Also, the integer can be obtained by rounding up the real numbers which are represented from the binary string as above described. This structure of chromosome is illustrated in Figure 2.

To provide an efficient search through the infeasible region, and to assure that the final best solution is feasible, a penalty function technique (based on the constraint violation) is needed to transforms the constrained problem into unconstrained problem by penalizing those solutions that are infeasible. One approach uses distance metric of the infeasible solution from the feasible region effective is more effective [8]. In the area of combinatorial optimization, Lagrangian relaxation is one such penalty method. It has been shown that penalty functions based on the distance from feasibility outperform those based up the number of violated constraints. When a high degree of penalty is imposed, more emphasis is placed on obtaining feasibility and the GA-based approaches will move very quickly towards a feasible solution, so that the system will tend to converge to feasible points even if it begins far from optimal. For the above reasons, the penalty guide approach has been applied in this research for handling the constraints of the problems.

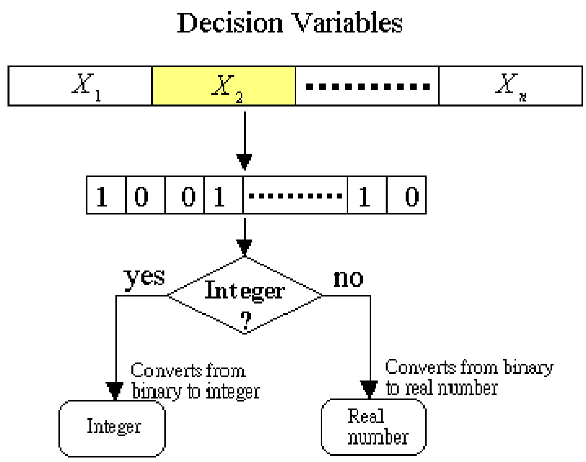

Fig.2 Chromosome representing the solution of an optimization problem.

When including a sub-subsection you must use, for its heading, small letters, 11pt, left justified, bold, Times New Roman as here.

4 Applications and the corresponding computational results

In order to illustrate the proposed agile approach for engineering design and performance of the GA-based search algorithms, two engineering design examples are presented in this section. In particular, as reported, solutions obtained by using GA-based approaches are as well as or better than previously best known solutions.

4.1 The reliability design of the life support system in a space capsule

The system reliability is an important problem in system design. The first example for illustrating the validity of the proposed algorithm is the Life support system in a space capsule [22][18] [19] [24]. The schematic is described in Figure 3. Also, the system reliable problem is formulated as a nonlinear programming model. The details of the model is described as follows:

Notation:

R_i : reliability of the system component i

$\overline{R}_i$: unreliability of the system component i

C_i : cost of the system component i

C_S : total cost of the system

R_S : reliability of the system

$R_{S,\min}$: minimum required system reliability

K_i , α_i : constants associated with cost function of component i

The formulation of the system reliability problem is described as follows:

$$Min.\, C_S = 2 \cdot \sum_{i=1}^{4} K_i \cdot R_i^{\alpha_i}$$

Subject to

$0.5 = R_{i,\min} \leq R_i \leq 1, \quad i = 1,2,3,4.$

$R_{S,\min} \leq R_S \leq 1$

where,

$$R_S = 1 - R_3 \cdot \left(\overline{R}_1 \cdot \overline{R}_4\right)^2 - \overline{R}_3 \cdot \left[1 - R_2 \cdot \left(1 - \overline{R}_1 \cdot \overline{R}_4\right)\right]^2$$

In this case, the reported solution obtained from the proposed model has compared with others which failed to produce the optimum solution obtained by GA and hybrid EC approaches. The same parameters have been used in the application for the comparison, where k_1 =100, k_2 =100, k_3 =150, k_4 =100, $R_{S,\min}$ =0.9, and α_i =0.6 for all i (i = 1,2,3,4). Best two solutions obtained by using EC based search algorithm are compared with the results from previous literature in Table 1. It shows that previous methods failed to produce the optimum solution obtained by the proposed methodologies, i.e., those techniques provided the inferior solution compared with the solutions obtained from the class of evolutionary algorithms.

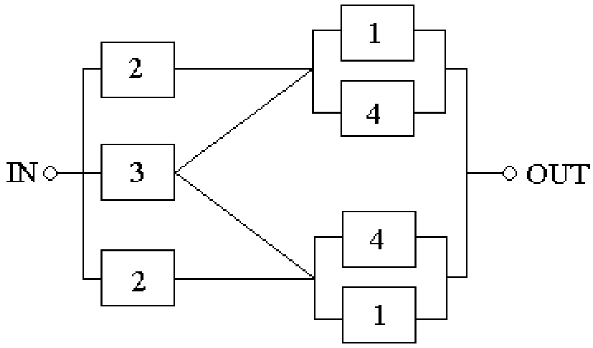

Fig.3 Life support system in a space capsule.

Table 1. Solutions comparison.

	[18]	[18]	[22]	[24]	SA-GA	MCA-GA
R_1	0.50095	0.50006	0.50001	0.5059	0.50000	0.50001
R_2	0.83775	0.83887	0.84062	0.7380	0.83892	0.83891
R_3	0.50025	0.50001	0.50000	0.5034	0.50000	0.5000
R_4	0.50015	0.50002	0.50000	0.6042	0.5000	0.5000
C_1	132.100	131.960	131.952	133.371	131.951	131.953
C_2	179.845	179.989	180.214	166.672	179.996	179.994
C_3	131.990	131.953	131.951	198.732	131.951	131.951
C_4	197.968	197.932	197.926	147.821	197.926	197.926
R_S	0.90001	0.90001	0.90050	0.9845	0.90000	0.90000
C_S	641.903	641.833	642.04	647.489	641.824	641.824

4.2 Pressure vessel design problem

The problem of designing a pressure vessel is illustrated in Figure 4. The objective is to minimize the total manufacturing cost for the pressure vessel [7] [12] [11] [13] [21] [23].

Notation:

x_1: the thickness of the spherical head.

x_2 : the thickness of the shell

x_3 : the inner radius

x_4: the length of the shell

Based on the mathematical modeling of the vessel design problem, the formulation is described as follows:

$$Min.\ f(X) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(X) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(X) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 750 \cdot 1728 \le 0$$

$$g_4(X) = -240 + x_4 \le 0$$

$$1.00 \le x_1 \le 1.375$$

$$0.625 \le x_2 \le 1.000$$

$$47.5 \le x_3 \le 52.5$$

$$90.0 \le x_4 \le 112.0$$

where x_1 and x_2 are discrete variables with discreteness 0.0625. And x_3 and x_4 are continuous variables of the side constraints which specify the ranges for variables x_3 and x_4.

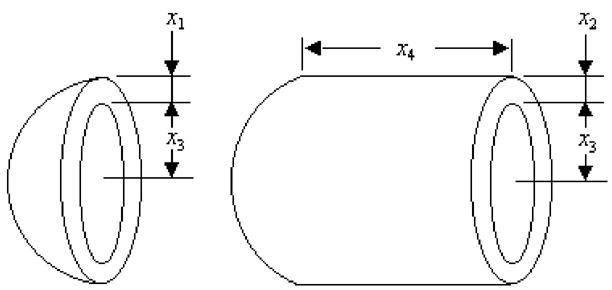

Fig. 4 Pressure Vessel [21]

In our implementation, solutions of the mixed-discrete optimization problem is represented by strings of binary digits, each string consisting of four substrings for each variable x_i (where i = 1, 2, 3, 4). Each substring in turn consists of a binary substring representing either the continuous or discrete value of the engineering design variable. The representation of real numbers (for example, the values of continuous variables x_3 within [47.5, 52.5] and x_4 within [90.00, 112.00]) in our solution representation approach is the same as that described by [16]. But the discrete number is not easy to be represented from the binary string directly without any transformation. Since x_1 and x_2 are discrete variables with deterministic discreteness 0.0625, they can be completely expressed as: x_1=1+ $0.0625y_{1i}$, x_2=0.625 + $0.0625y_{2i}$, where y_{1i}, y_{2i} = 0, 1, 2,...,6. So, the discrete variables x_1 and x_2 are represented by integer variables y_{1i} and y_{2i} respectively which range from 0 to 6.

The numerical results using different parameters are shown in Table 2 and Table 3, in which the best two solutions of each problem are reported and compared with solutions reported previously in the literature. Firstly, according to the range of the side constraints, three best solutions of the proposed method in this study has been compared with those obtained by using other methods in Table 2. Where the design variables x_3 and x_4 are continuous and the side constraints are $47.5 \le x_3 \le 52.5$ and $90.0 \le x_4 \le 112.0$. When the side constraints are $40.0 \le x_3 \le 80.0$ and $20.0 \le x_4 \le 60.0$, the coefficient of the third item in the objective function is 3.1611 instead of 3.1661, the solution comparisons are shown in Table 3.

Table 2 The comparison of solutions ($47.5 \le x_3 \le 52.5$ and $90.0 \le x_4 \le 112.0$)

	[7]	[21]	[11] [12]	SA-GA	MCA-GA
f(X)	8048.6	7982.5	7127.3	7059.4	7059.4
x_1	1.125	1.125	1.000	1.0000	1.0000
x_2	0.625	0.625	0.625	0.6250	0.6250
x_3	48.380	48.970	51.250	51.0799	51.0799
x_4	111.745	106.720	90.991	90.0054	90.0054
g_1	-0.191	-0.179	-1.011	-0.0142	-0.0142
g_2	-0.163	-0.145	-0.136	-0.1377	-0.1377
g_3	-75.875	-0.000	-18759.754	-33.074	-33.074
g_4	-128.255	-122.299	-149.009	-149.994	-149.994

Table 2 shows that all three best solutions found by the GA and hybrid EC approaches are much better than those found by [7], [21], [12], and [11]. Table 3 indicates that solutions obtained by our GA and hybrid EC approaches are no better than the solution found by [13] but better than those found by [23].

The objective of the above two problems in Section 4.1 and 4.2 is to minimize the design and/or manufacturing cost subject to the nonlinear constraint(s). As demonstrated in the previous section, the best solutions obtained by the proposed methods are all better than or as well as the well-known best solutions by other methods for the both design problems. Comparative results show that the proposed GA based search algorithm is very comparable with other current methods. The solutions found by the proposed approaches sometimes vary in the design variables. Also, this may offer the design engineer a variety of options from which to choose with negligible difference in the design cost. It is concluded that the proposed methods provide solutions which are exhibit quality comparable to other heuristic algorithms while at the same time offering several advantages, including the generation of multiple design alternatives, the lack of requirement for the computation of derivatives, and relatively efficient computation.

Table 3. A comparison of solutions ($40.0 \le x_3 \le 80.0$ and $20.0 \le x_4 \le 60.0$)

	[13]	[23]	SA -GA	MCA-GA
$f(X)$	7197.7	7207.497	7198.801	7198.801
x_1	1.1250	1.1250	1.1250	1.1250
x_2	0.6250	0.6250	0.6250	0.6250
x_3	58.29012	58.1978	58.275946	58.275946
x_4	44.6928	44.2930	43.775336	43.775336
g_1	-6.84e-007	-0.001782	-0.0002742	-0.0002742
g_2	-0.0689	-0.06973	-0.0690475	-0.0690475
g_3	-10674	-974.5829	-48.22068	-48.22068
g_4	-195.3071	-195.7070	-196.22466	-196.22466

Figures 5 and Figure 6 show the convergence results of each evolutionary algorithm on the two engineering design applications. In general, the two hybrid evolutionary algorithms yield very good results as well as traditional GA does, but they were found to have much faster convergence compared with GA. With our limited experiment, the convergence of the hybrid EC approaches are more than 6 times faster than that of the traditional GA in both engineering design applications. Above all, the MCA-GA in this research has the best performance including the fast convergence and best solution quality shown in Table 2 and Table 3.

Due to the concluded advantages of the hybrid EC approach, MCA-GA is applied in the proposed agile engineering design approach for finding the optimal or near optimal solution(s) of the optimal design problems. Through the Web based interface, the users can choose the specified design problem. The input of specified values for each design parameter can be sent to the server over the Internet. Figure 7 and Figure 9 show the Web based interface (browser) from which users can entry the input and submit them to server. Based on the specified parameters, the optimal solution can be obtained by using the MCA-GA activated by CGI. When the solution has been obtained, the solution can be sent back to the designers through WWW and displayed on the web browser. The results of the Figure 7 and Figure 9 are shown in Figure 8 and Figure 10 respectively.

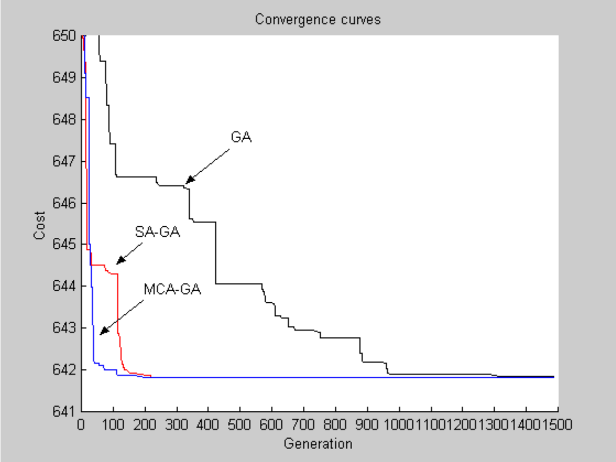

Fig.5 The convergence curves for three algorithms in example 1.

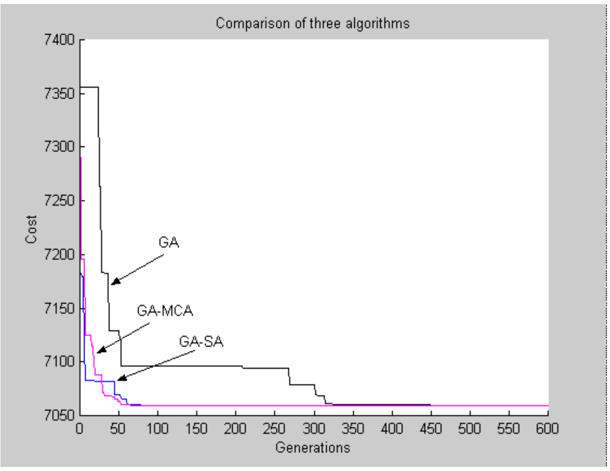

Fig.6 The convergence curves for three algorithms in example 2

5 Conclusion

In this paper, we have discussed the idea behind the agile remote engineering design approach and have described the hybrid evolution computation based approaches for solving the engineering design problems formulated by nonlinear programming models. Two applications were tested to illustrate the performance of the proposed the hybrid EC algorithms compared with other methodologies in previous literatures. The main contributions of the paper are as follows: (1). The hybrid EC approaches are applied in this research to improve the solution quality and the convergence speed greatly (2). An agile approach is proposed for finding optimal solutions of the engineering design problems. And it's implemented through WWW without limited by the constraints of geographic distance and computer platform used. However, the following issue needs further studies: the extension of the current work can be developed as a collaborative approach for the different engineering design teams geographically separated. We hope that this paper will interest other researchers to extend the proposed idea as well as to use WWW technology in other ways to improve the use of computer-based information systems in decision making for the agile engineering approach design in industry.

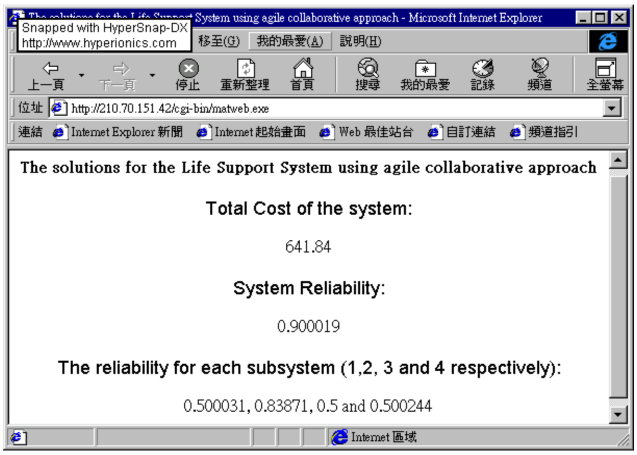

Fig.8 Outputs of the design of Life support system.

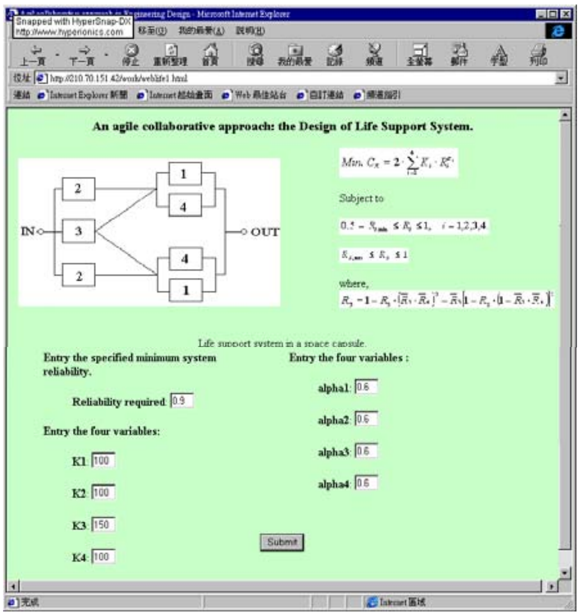

Fig.7 Obtaining inputs for the design of Life support system.

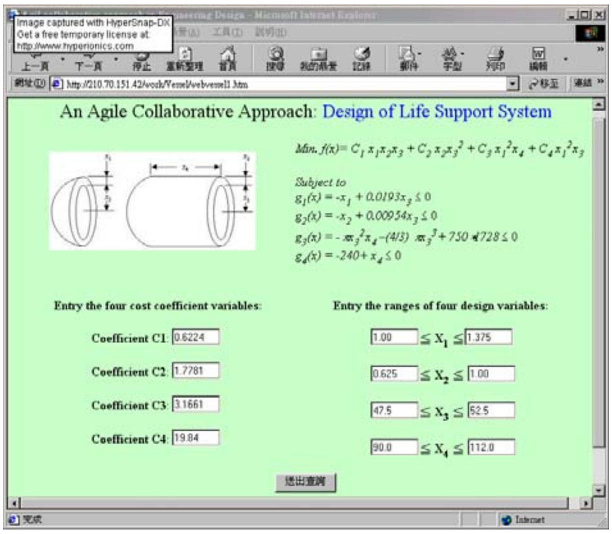

Fig.9. Obtaining inputs for the vessel design

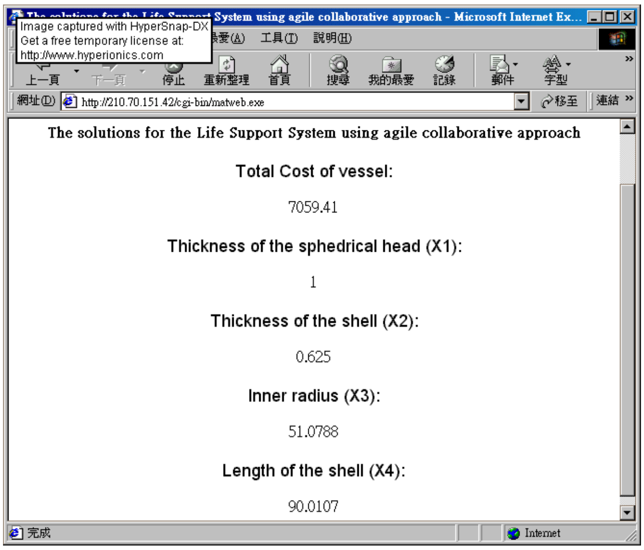

Fig. 10. Receiving outputs of the vessel design.

References:

[1] Bhanot, G., Creutz, M., and Neuberger, H., Microcanonical simulation of Ising systems, *Nuclear Phys.*, Vol. B235, No. FS 11, 1984, pp. 417-434.

[2] Bhandarkar, Suchendra M. and Zhang, Hui, Image segmentation using evolutionary computation, *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 1, 1999, pp. 1-21.

[3] Bloomer, J., Power Programming with RPC, O'Reilly & Associates, Inc., Sebastopol, CA, 1995.

[4] Casotto, A., Romeo, F., and Sangiovanni-Vincentelli, A., A parallel simulated annealing algorithm for the placement of macro cells, *IEEE Trans. Computer-Aided Design*, Vol. 9, 1987, pp. 838-847.

[5] Chen, T.-C., "Genetic algorithms for an Engineering Design Optimization Problem," *Proceedings of the 1998 CIIE National Conference*, Vol. 1, 1998, pp. 429-434.

[6] Coit, D.W., and Smith, AE., Penalty guided genetic search for reliability design optimization, *Computers & Industrial Engineering*, Vol. 30, No. 4, 1996, pp. 895-904.

[7] Fu, J., Fenton, R.G. and Cleghorn, W.L., A mixed integer discrete continuous programming method and its application to engineering design optimization, *Engineering Optimization*, Vol. 17, 1991, pp. 263-280.

[8] Goldberg, D.D., Genetic algorithms in search, optimization & machine learning, New York: Addison Wesley, 1989.

[9] Holland, J.H., Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press, 1975.

[10]Hsieh, Yi-Hsieh, Chen, Ta-Cheng, and Bricker, D.L., Genetic algorithms for reliability design problems, *Microelectronics Reliability*, Vol. 38, 1998, pp. 1599-1605.

[11]Li, Han-Lin and Chang, Ching-Ter, An approximate approach of global optimization for polynomial programming problems, *European Journal of Operational Research*, Vol. 107, 1998, pp. 625-632.

[12]Li, Han-Lin and Chou, Chin-Tan, A global approach for nonlinear mixed discrete programming in design optimization, *Engineering Optimization*, Vol. 22, 1994, pp. 109-122.

[13]Litinetski, Victor V. and Abramzon, Boris M., MARS – A multistart adaptive random search method for global constrained optimization in engineering applications, *Engineering Optimization*, Vol. 30, 1998, pp. 125-154.

[14]Mahfoud, Samir and Goldberg, David E., Parallel recombinative simulated annealing: A genetic algorithm, *Parallel Computing*, Vol. 21, 1995, pp. 1-28.

[15]McCool, Rob (1994), The Common Gateway Interface, URL: http://hoohoo.ncsa.uiuc.edu/cgi/overview.html.

[16]Michalewicz, Z., Genetic algorithms + data structures = evolution programs, Berlin-Heidelberg: Springer-Verlag, 1994.

[17]Painton, L. and Campbell, J., Genetic algorithms in optimization of system reliability, *IEEE Transactions on Reliability*, Vol.44, 1995, pp. 172-178

[18]Ravi, V., Murty, B.S.N. and Reddy, P.J., Nonequilibrium Simulated Annealing Algorithm Applied to Reliability Optimization of Complex Systems, *IEEE Transactions On Reliability*, Vol. 46, No. 2, 1997, pp. 233-239.

[19]Richardson, J.T., Palmer, M.R., Liepins, G, and Hilliard, M.,Some guidelines for genetic algorithms with penalty functions," *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989, pp. 191-197.

[20]Romeo, F. and Sangiovanni-Vincentelli, A., A theoretical framework for simulate annealing, *Algorithmica*, Vol. 6, 1991, pp. 302-345.

[21]Sandgren, E., Nonlinear integer and discrete programming in mechanical design optimization, *Journal of Mechanical Design*, Vol. 112, 1990, pp. 223-229.

[22]Tillman, F.A., Hwang, C.L., and Kuo, W., Optimization of System Reliability, Marcel Dekker, 1980.

[23]Wu, Shyue-Jian and Chow Pei-Tse, Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization, *Engineering Optimization*, Vol. 24, 1995, pp. 137-159.

[24]Zhang, J.H. and Xu, X.H., An efficient evolutionary programming algorithm, *Computer & Operations Research*, Vol. 26, No. 7, 1999, pp. 645-663.