49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Computational Fluid Dynamics Optimization of Shape of Sprue for Die Casting Considering Product Quality

Ken'ichi Kanazawa and Ken'ichi Yano

Abstract— **Computational fluid dynamics (CFD) simulators have recently been used for optimization in die casting and various other fields. However, solving an optimization problem with a CFD simulator (CFD optimization problem) has the issue of uncertainty in the evaluated values from the CFD simulations. Such problems are, of course, difficult to optimize compared to general problems. In this paper, we propose an optimization algorithm that can search for good solutions to CFD optimization problems exactly. We have applied the algorithm to the optimization of the shape of a sprue for die casting at an actual plant.**

I. INTRODUCTION

Recently, the development of computational fluid dynamics (CFD) simulators has made it possible to analyze fluid behavior using personal computers. CFD simulation is a technique whereby fluid is analyzed by solving the Navier-Stokes equations numerically. With advances in computing power and the sharp decrease in the price of personal computers, CFD simulators have become prevalent in a variety of fields[1]. Moreover, for die casting and other casting methods, CFD simulators have allowed visualization of molten metal and have been used to optimize casting design or casting conditions for improving product quality[2].

Many studies have been carried out on optimization using a CFD simulator. Obayashi et al. optimized the shape of the wing planform of a supersonic aircraft by using the Multiple Objective Genetic Algorithm[3]. In addition, our laboratory has used the Genetic Algorithm (GA) to optimize die casting plunger speed in order to decrease air entrapment[4].

However, a CFD simulator produces uncertain analysis results, owing to error in the numerical calculation. For optimization problems using a CFD simulator (CFD optimization problems), the optimal solution and its neighboring solutions are difficult to find with a simple GA.

In the present study, we aim to devise a CFD optimization algorithm that can search exactly for good solutions to an optimization problem that has uncertain analytical values, such as the CFD optimization problem. Specifically, the algorithm treats the uncertainty in the evaluated values from a CFD simulator as noise, and optimizes the problem using the noise-corrected evaluated values. In this way, the algorithm can search for solutions in an environment where the uncertainty of the CFD simulator is reduced. The proposed algorithm is used to optimize the shape of a sprue for die casting, and the effectiveness of the algorithm is demonstrated through experiments at an actual die-casting plant.

K. Kanazawa is with Department of Human and Information Systems Engineering, Gifu University, 1-1 Yanagido, Gifu-city, 501-1193, Japan
K. Yano is with Department of Mechanical Engineering, Mie University, 1577 Kurimamachiya-cho, Tsu-city, 514-8507, Japan yanolab-mail@gifu-u.ac.jp

II. CFD OPTIMIZATION PROBLEM

A CFD simulator enables fluid analyses by solving a set of partial differential equations called the Navier-Stokes equations. The conventional methods for solving these equations are the difference method, the finite element method and the boundary element method. All these methods subdivide an analytical region into very small elements, such as grid points, to solve the equations discretely and approximately. Therefore, the analysis results vary depending on the method of division. The CFD simulator used in this study employs the difference method and subdivides an analytical region into a mesh of fixed rectangular cells. The intended analysis precision can be set by adjusting the mesh spacing or cell size. In general, a fine mesh setting improves the analysis precision, but increases the analysis time. In contrast, a coarse mesh setting decreases the analysis time, but reduces the analysis precision.

We examined the amount of error in the analysis results from a CFD simulator. The simulation used for the sprue shape optimization described herein was carried out by varying the mesh position finely but maintaining the mesh spacing. The results are shown in Fig. 1.

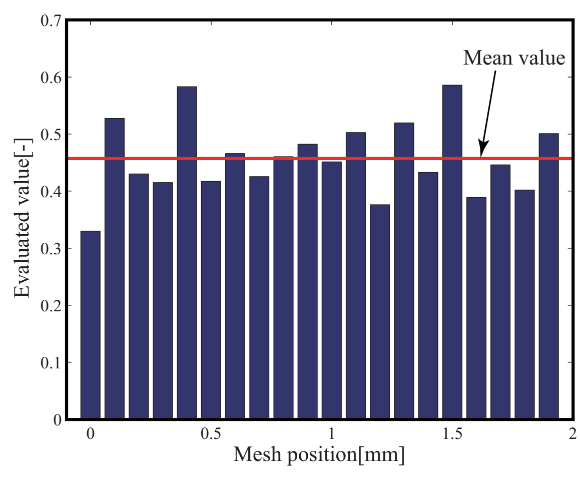

Fig. 1. Variation in evaluated value depending on mesh position

The horizontal axis of this graph shows fine displacement of the mesh position, and the vertical axis shows the evaluated value. Ideally, as the mesh position is changed, the evaluated values would remain fixed. However, the evaluated values actually vary considerably. This variability is reduced if the mesh is set finer, but such an approach for improving the analysis precision is difficult because the analysis time is increased.

Let us consider this variability in the evaluated values as noise. Then, an evaluated value from the CFD simulator can be modeled by the following equation:

$$\begin{aligned} Y &= F(x) \\ &= \tilde{F}(x) + D(x, \gamma, \sigma) \end{aligned} \tag{1}$$

where $\tilde{F}(x)$ is the true ideal objective function without noise, and its function value is thought to be close to the mean value in Fig. 1 but cannot be observed. $F(x)$ and Y are actually obtained as the observed objective function and the observed value, respectively, by adding noise $D(x, \gamma, \sigma)$ to $\tilde{F}(x)$. In $D(x, \gamma, \sigma)$, γ is the random element of the noise that corresponds to mesh position and σ is the intensity of the noise that corresponds to mesh spacing. The observed value Y is always constant if γ, σ and x are constant. This shows that a CFD simulator has reproducibility under identical conditions.

A general optimization problem entails finding the solution x to minimize the objective function $F(x)$ in (1). Meanwhile, a CFD optimization problem entails finding x to minimize the true objective function $\tilde{F}(x)$, which cannot actually be obtained, with the help of the observed value Y including noise. Therefore, the CFD optimization problem can be considered an optimization problem including noise, and finding the optimal solution is more difficult for a CFD optimization problem than a general optimization problem. For such optimization problems including noise, GA and other similar algorithms can realistically be used to find the optimal solution for $F(x)$ but not $\tilde{F}(x)$.

III. Construction of Optimization Algorithm with Noise Correction

We constructed a new optimization algorithm that can search for good solutions to an optimization problem including noise, such as a CFD optimization problem. This optimization algorithm is based on the Real-coded Genetic Algorithm (RealGA) and compensates for noise by using a response surface. A response surface is an approximate function of an objective function obtained by regression analysis. The optimization algorithm treats function values of the approximate function as evaluated values with noise correction (corrected values), and then optimizes the problem.

In addition, to compare the performance of the proposed algorithm to that of RealGA, we optimized a simplified model of a CFD optimization problem using these algorithms.

A. Noise Compensation Using Response Surface

The algorithm uses the least-squares method to generate a response surface, and the approximate function as the response surface is a multivariate polynomial function.

In the N-ary objective function $F(x)$, the variable x is described by

$$x = (x_1, x_2, \ldots, x_n, \ldots, x_N) \tag{2}$$

Let an individual (x, Y) consist of this variable x and its observed value Y, which is the function value of $F(x)$. Let us derive the approximate function $f(x)$ by applying the least-squares method to N-individuals.

Assuming that the variable of a d-th individual is

$$x_{[d]} = (x_{1[d]}, x_{2[d]}, \ldots, x_{n[d]}, \ldots, x_{N[d]}) \tag{3}$$

the individual is $(x_{[d]}, Y_{[d]})$. Meanwhile, assuming that the approximate function $f(x)$ is a more general function, not only a polynomial function, $f(x)$ is then given by

$$\begin{aligned} f(x) = a_1 f_1(x) + a_2 f_2(x) + \cdots \\ + a_p f_p(x) + \cdots + a_P f_P(x) \end{aligned} \tag{4}$$

where P is the term number of the function $f(x)$. Each $f_p(x)$ can be taken as an arbitrary function regardless of linearity or nonlinearity. Since these $f_p(x)$ are linearly combined by the coefficient a_p, however, $f(x)$ is a linear function in this sense. The least-squares method can approximate function $F(x)$ as the linear function $f(x)$.

Therefore, to evaluate the approximate function $f(x)$, we evaluate coefficient $a_p (p = 1, \ldots, P)$, and each a_p is calculated using (5). In this equation, the summation ranges of all Σ are $d = 1, \ldots, D$.

Next, we define each $f_p(x)$, which is the component of the approximate function $f(x)$. Since we let $f(x)$ be a polynomial function, $f_p(x)$ can be expressed as follows:

$$f_p(x) = \prod_{n=1}^{N} x_n^{m_{pn}} \tag{6}$$

where m_{pn} is a nonnegative integer. Since $f_p(x)$ is expressed as a product of the variables x_n, $a_p f_p(x)$ is one term. Then, the order of the term $a_p f_p(x)$ is

$$q_p = \sum_{n=1}^{N} m_{pn} \tag{7}$$

Consequently, the order of the approximate function $f(x)$, that is, q, is

$$q = \max_p q_p \tag{8}$$

$f(x)$ can be made up of these terms in various combinations. For simplicity, let the order q be the only parameter deciding the combination of the terms, and let $f(x)$ be composed of all terms the order of which is equal to or less than q, including the constant term. Then, the term number of $f(x)$, P, is given by

$$P = {}_{N+q}\mathrm{C}_N \tag{9}$$

Eventually, the corrected value $y_{[d]}$ of an individual $(x_{[d]}, Y_{[d]})$ is obtained as follows, by substituting $x_{[d]}$ into (4).

$$y_{[d]} = f(x_{[d]}) \tag{10}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \sum f_1(x_{[d]})^2 & \sum f_2(x_{[d]})f_1(x_{[d]}) & \cdots & \sum f_P(x_{[d]})f_1(x_{[d]}) \\ \sum f_1(x_{[d]})f_2(x_{[d]}) & \sum f_2(x_{[d]})^2 & \cdots & \sum f_P(x_{[d]})f_2(x_{[d]}) \\ \vdots & \vdots & \ddots & \vdots \\ \sum f_1(x_{[d]})f_P(x_{[d]}) & \sum f_2(x_{[d]})f_P(x_{[d]}) & \cdots & \sum f_P(x_{[d]})^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum Y_{[d]} f_1(x_{[d]}) \\ \sum Y_{[d]} f_2(x_{[d]}) \\ \vdots \\ \sum Y_{[d]} f_P(x_{[d]}) \end{bmatrix} \tag{5}$$

B. Optimization Algorithm

The proposed optimization algorithm is based on RealGA and able to find a good solution to an optimization problem including noise such as a CFD optimization problem by compensating for noise.

A flowchart of the proposed algorithm is shown in Fig. 2.

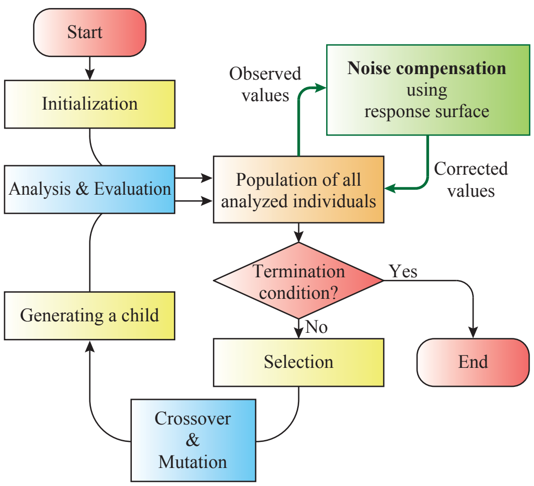

Fig. 2. Outline of proposed algorithm

At first, an initial population of individuals is generated randomly using uniform random numbers. Next, these individuals are analyzed and the observed values of the individuals are obtained. The algorithm evaluates the approximate function as a response surface from the observed values, and the corrected values are obtained. Then, candidates for parent individuals are selected from all the analyzed individuals based on their corrected values. Furthermore, two actual parent individuals are selected from the candidates by roulette selection, and one child individual is generated by genetic operations such as crossover and mutation. This child individual is analyzed like the initial population, and this flow is repeated until the number of analyzed individuals reaches the maximum number of individuals. Eventually, the individual that has the best corrected value becomes the optimized solution obtained from the algorithm.

The specific parameters of the algorithm are initial population size, the number of candidate parent individuals, and maximum number of individuals.

C. Validity Verification of Proposed Algorithm

We verified the validity of the proposed algorithm by using a simplified model. Since a CFD optimization problem requires an immense amount of time to be optimized and true evaluated values are not observed, we constructed a model to replace this CFD optimization problem. The model was optimized by using the proposed algorithm and RealGA to compare the performance of both algorithms at searching for solutions.

We take the brachistochrone problem, which is a dynamic model similar to CFD, as the simplified model. A brachistochrone is the curve between two points that can be covered in the least time by a body that begins at the start point with zero speed and moves along the curve to the end point under the action of gravity. Here, the body is a mass point and friction is not taken into account. The conventional brachistochrone problem entails evaluating this curve analytically, but we defined the descent curve optimization problem as finding the descent curve that is defined by certain design variables to minimize the descent time calculated numerically.

Let the start point P_s and the end point P_e of the descent curve be

$$P_s = (0, 0) \tag{11}$$
$$P_e = (x_e, -z_e) \quad (x_e > 0,\ z_e > 0) \tag{12}$$

and set N-relay points between P_s and P_e:

$$P_n = \left(\frac{n}{N+1}x_e, -z_n\right) \qquad (z_n > 0) \\ , n = 1, \ldots, N \tag{13}$$

where z_n $(n = 1, \ldots, N)$ are the design variables of the descent curve. Then, this descent curve is defined as a spline curve joining these points and is described by the following equation:

$$z = f(x, z_1, \ldots, z_N) \tag{14}$$

An example of the descent curve is shown in Fig. 3.

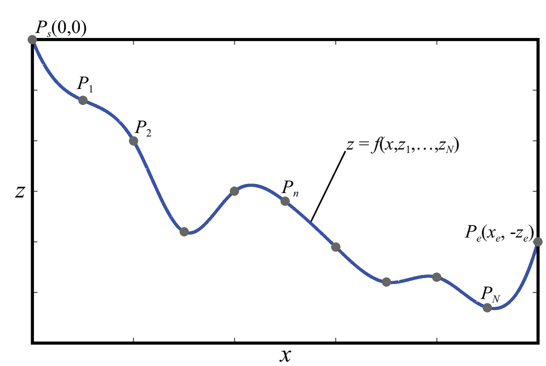

Fig. 3. Example of a descent curve

The time a body takes to reach the end point of f(x) from the start point, that is, the descent time, is calculated by

$$\tilde{T}(z_1, \ldots, z_N) = \int_0^{x_e} \sqrt{\frac{1 + \left(\frac{d}{dx} f\right)^2}{2gf}} dx \tag{15}$$

where g is the gravitational acceleration.

Assuming that noise is added to this descent time as in (1), we obtain

$$T = \tilde{T} + D(z, \gamma, \sigma) \tag{16}$$

Here, $\tilde{T}$ is the true descent time or a real evaluated value, T is the observed descent time or observed value, D is the noise generated based on a normal random number, and σ is the standard deviation of the normal random number as the intensity noise. In addition, let the random element of the noise γ be constant in one optimization, and the value of $D(z, \gamma, \sigma)$ is unique if z, γ and σ are fixed.

Therefore, the descent curve optimization problem is defined by the following equation:

$$\begin{array}{ll} \text{Minimize} & \tilde{T}(z_1, \ldots, z_N) \\ \text{Subject to} & 0 < z_n, \quad n = 1, \ldots, N \end{array} \tag{17}$$

Let the real evaluated values $\tilde{T}$ be not observed during the optimization, and we must search for the solution to minimize $\tilde{T}$ based on the observed values T.

We optimized the descent curve using both the proposed algorithm and RealGA, and compared the performance of these algorithms at searching for solutions. The parameters of the proposed algorithm and RealGA are listed in Table I and Table II, respectively. The optimization problem has six variables ($N = 6$). The maximum of the each variable is 10, the minimum is 0.1, and the step size is 0.1. In addition, the gravitational acceleration is $g = 9.81[\text{m/s}^2]$ and the end point is $P_e = (10, -10)[\text{m}]$. By setting the standard deviation σ at four levels, $\sigma = 0.0$, 0.1, 0.2 and 0.3, we obtain four optimization problems with different noise intensities. When $\sigma = 0.0$, the optimization problem has no noise. These optimization problems with different noise intensity were optimized by each of the algorithms 50 times, and the performance of these algorithms is evaluated by comparing the mean value for each set of 50 optimized values.

TABLE I

PARAMETERS FOR PROPOSED ALGORITHM

Maximum number of individuals	300
Initial population size	20
Number of candidate parent individuals	20
Selection method	Roulette selection
Crossover method	BLX-α ($\alpha = 0.2$)
Approximation order : q	2

The comparison of the optimized results is shown in Fig. 4.

This figure shows that the proposed algorithm can obtain considerably better solutions than RealGA when the problem has noise: $\sigma = 0.1$, 0.2 and 0.3. This is because the proposed algorithm can estimate the initially evaluated values well by means of noise compensation. Unexpectedly, the proposed algorithm yields slight better solutions than RealGA when the problem has no noise: $\sigma = 0.0$. Therefore, the proposed algorithm exhibits better performance than normal RealGA for problems with and without noise.

TABLE II

PARAMETERS FOR REALGA

Maximum number of individuals	300
Population size	20
Number of elite individuals	15
Selection method	Roulette selection
Crossover method	BLX-α ($\alpha = 0.2$)

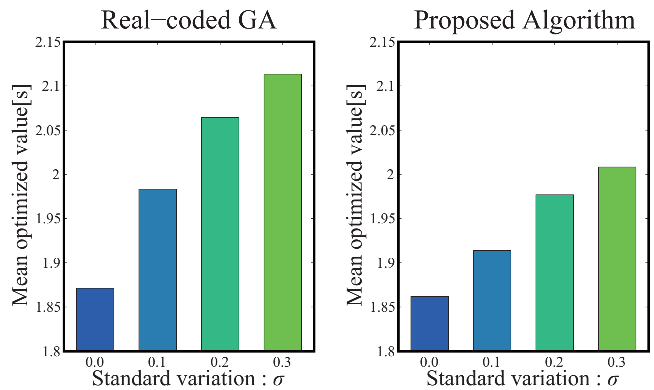

Fig. 4. Mean values of 50 optimized values versus σ for each algorithm

IV. OPTIMIZATION OF SPRUE SHAPE FOR DIE CASTING

A. Formulation of Optimization Problem

We applied the proposed algorithm constructed in the preceding section to the optimization of sprue shape for die casting. The parameters for the proposed algorithm in this optimization are taken from Table I.

A basic sprue shape and the layout of the design variables are shown in Fig. 5, and the parameters for the design variables are listed in Table III.

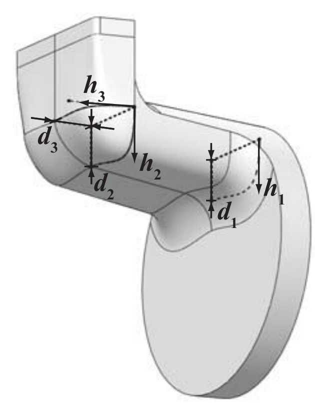

Fig. 5. Basic sprue shape and layout of design variables

TABLE III

PARAMETERS FOR DESIGN VARIABLES

Design variable	Max	Min	Step size
d_1, d_2, d_3[mm]	20	1	0.1
h_1, h_2, h_3[mm]	100	5	0.5

In Fig. 5, d_1, d_2, d_3 and h_1, h_2, h_3 are the design variables that define the thickness of the center parts and the side parts of the sprue, respectively.

An overview of the mesh setting is shown in Fig. 6, and the parameters for the mesh setting are listed in Table IV.

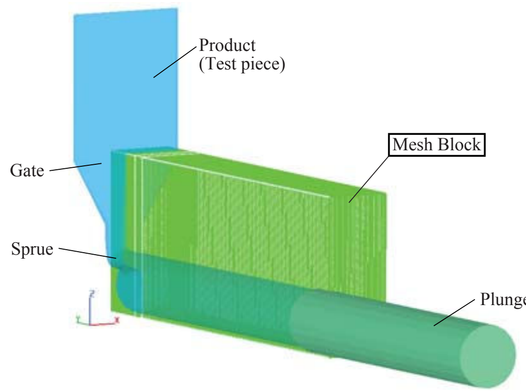

Fig. 6. Mesh setting for CFD simulation

TABLE IV

MESH PARAMETERS

Block	Cell size [m]	Number of cells
X-direction	0.002	40
Y-direction	0.0005 $\sim$ 0.002	329
Z-direction	0.002	100
Total number of cells		1,316,000

As seen in Fig. 6, the die casting device is symmetrical about the Y-Z plane. Thus, the analytical region is set as only a one-sided model. Furthermore, the mesh of thin-walled parts such as the gate is set finely to increase the analysis precision.

In the optimization, the shape of the sprue is evaluated on the basis of the amount of entrapped air [4]. Since air entrapment is responsible for inner defects of die-cast products, a sprue can be evaluated by determining the amount of air in the product by using a CFD simulator.

The amount of entrapped air in each time step, $a(i)$, which outflows from the gate, is

$$a(i) = \frac{\sum_{k=1}^{n}(V_{ak}F_{fk}V_{fk}V_{ck})}{\sum_{k=1}^{n}(F_{fk}V_{fk}V_{ck})}V_{out}(i) \tag{18}$$

where V_a is the volume fraction of entrapped air, F_f is the volume fraction of fluid, V_f is the volume fraction of a cell, V_c is the volume of a mesh cell, n is the total number of mesh cells in the measurement region and $V_{out}(i)$ is the volume of fluid outflow in each time step. Thus, the amount of entrapped air in a product, A, is obtained by the following summation of $a(i)$,

$$A = \sum_{i=1}^{N_f} a(i) \tag{19}$$

where N_f is the number of data until the total volume of fluid outflow reaches the volume of the product.

Therefore, the optimization problem is defined by the following equation:

$$\begin{array}{ll} \text{Minimize} & \tilde{A}(d_1, d_2, d_3, h_1, h_2, h_3) \\ \text{Subject to} & 1 \leq d_2 \leq d_1 \leq 20 \\ & 1 \leq d_3 \leq 20 \\ & 5 \leq h_2 \leq h_1 \leq 100 \\ & 5 \leq h_3 \leq 100 \end{array} \tag{20}$$

It should be noted here that $\tilde{A}$ is the true objective function without noise, but the objective function observed is actually A.

B. Optimization Result

The optimized shape of the sprue is shown in Fig. 7. This figure also shows the initial shape of the sprue for comparison.

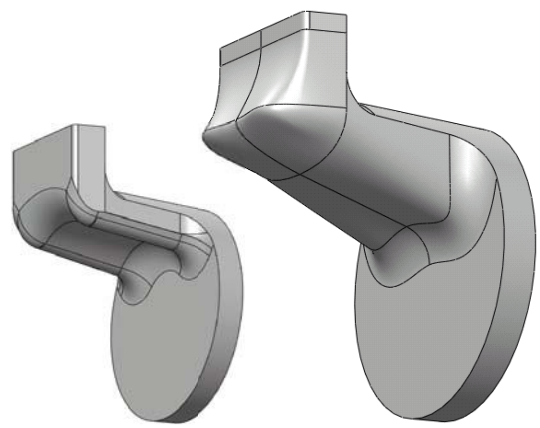

Fig. 7. Optimized and initial shape of sprue

The computation time for the optimization was about 200 hours using a PC with an Intel Core 2 Quad processor (2.83 GHz). The values of the design variables are $d_1 = 20.0$, $d_2 = 12.6$, $d_3 = 19.5$, $h_1 = 100.0$, $h_2 = 28.0$ and $h_3 = 87.5$, and the observed value and the corrected value are 0.5366 and 0.4570, respectively.

The optimized shape is inflated compared to the initial shape. It is thought that the shape moderates the flow of molten metal, and thus reduces air entrapment in the molten metal. In addition, a comparison of the simulation results for these sprue shapes is shown in Fig. 8.

In this figure, the parts enclosed by an ellipse denote the parts of the fluid with high air entrapment. For the initial shape, the fluid with trapped air is inside the fluid flow,

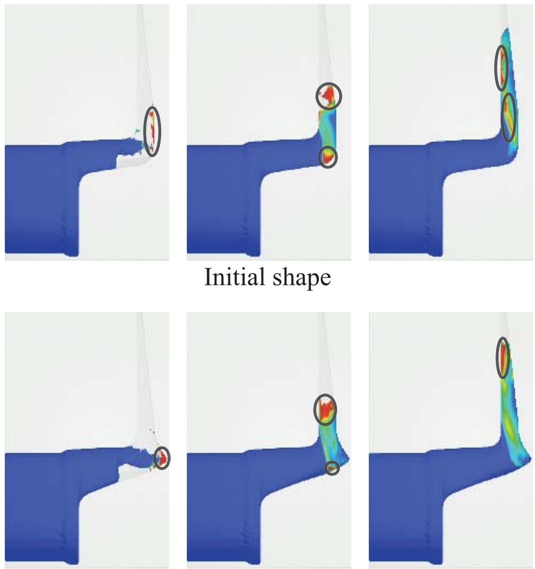

Fig. 8. Simulation results

but for the optimized shape, the fluid with trapped air is at the head of the flow. Therefore, it is expected that the actual amount of entrapped air would be reduced by using the optimized sprue shape.

C. Experimental Result

Experiments at an actual die-casting plant were performed with the optimized sprue and the initial sprue. We used the blister test to evaluate test pieces in these experiments. The blister test is a method for analyzing the air capacity of a product by heating the product and expanding the air. The test pieces after blister tests are shown in Fig. 9; the areas with air bubbles in the test pieces are shown in Fig. 10.

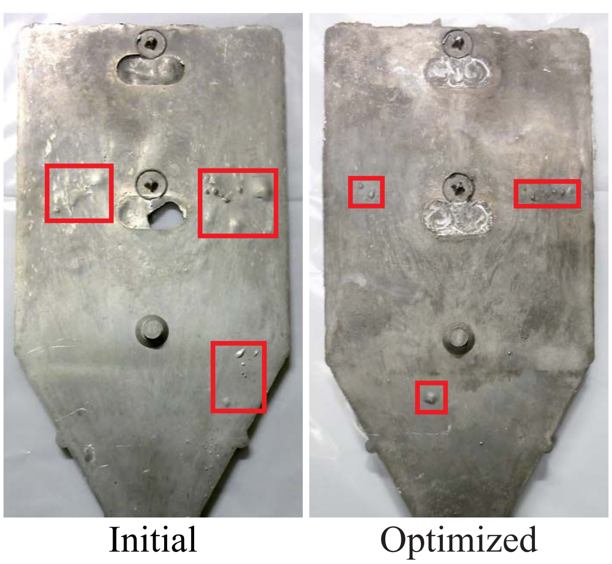

Fig. 9. Test pieces after blister tests

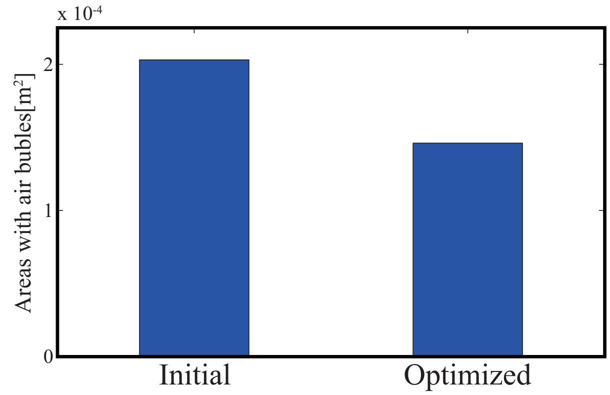

Fig. 10. Areas with air bubbles

The areas with air bubbles were measured using graph paper by counting the squares in the air bubble over $0.001[\mathrm{m}]$ in diameter. These figures indicate that the test piece had fewer air bubbles using the optimized sprue than the initial sprue. Thus, by using the proposed algorithm, we can optimize the shape of the sprue to reduce these product defects.

V. Conclusion

An optimization algorithm that can search for good solution to the CFD optimization problem exactly by compensating for the variation in uncertain evaluated values using a response surface has been constructed. Moreover, the shape of a sprue for die casting was optimized and then evaluated experimentally at an actual die-casting plant. As a result, it has been found the proposed algorithm reduces product defects and leads to improved quality.

References

[1] S. Pieri, C. Poloni and M. Muhlmeier, "Integrating Multibody Simulation and CFD : toward Complex Multidisciplinary Design Optimization," JSME international journal. Ser. B, Fluids and thermal engineering, vol.48, no.2, pp.224-228, 2005.

[2] P. Cleary, M. Prakash, M. Sinnott, J. Grandfield, V. Alguine and K. Oswald, "3D SPH Simulations Of The Aluminium Ingot Casting Process," Third Int. Conf. on CFD in the Minerals and Process Industries, pp.409-414, 2003.

[3] S. Obayashi and Y. Takeguchi, "Multipoint Aerodynamic Design of Supersonic Wing Planform Using MOGA," Proceedings of the Seventh Annual Conference of the Computational Fluid Dynamics Society of Canada, Halifax, June, 1999.

[4] K. Yano, K. Hiramitsu, Y. Kuriyama and S. Nishido, "Optimum Velocity Control of Die Casting Plunger Accounting for Air Entrapment and Shutting," International Journal of Automation Technology, Vol.2, No.4, pp.259-265, 2008.