Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Optimal algorithm to multimodal solution space formed by CFD simulator

Yoshifumi Kuriyama * **Ken'ichi Yano** **

* *Faculty of Engineering, Gifu University, Gifu, Japan*
(Tel: +81-58-293-2508; e-mail: n3812202@edu.gifu-u.ac.jp)
** *Faculty of Engineering, Mie University, Mie, Japan*
(e-mail: yano@mach.mie-u.ac.jp)

Abstract: For die-casting, various approaches to finding the optimal control settings and optimal mold by using a computational fluid dynamics (CFD) simulator have recently been studied and applied. However, the optimal value obtained by using a conventional CFD simulator does not consistently agree with experiment, owing to computational error. Accordingly, we analyze the problem of optimization with a CFD simulator in order to develop an optimization method that can consistently produce suitable results. To assess the effectiveness of the proposed method, it was applied to the optimization of plunger speed in die-casting.

1. INTRODUCTION

Die-casting is used to manufacture a wide range of products. However, one problem with Die-casting for high-speed injection molding is the formation of shrinkage cavities. To avoid this, velocity control of the plunger has proved highly effective; however, analyzing the behavior of molten metal remains extremely difficult, and skilled operators of die-casting equipment typically must decide the velocity control input based on their experience and instinct. Setting of the plunger speed is also complicated by the fact that the behavior of molten metal cannot be seen as it is injected at the sleeve and into the mold.

In recent years, numerical simulators for fluid analysis based on computational fluid dynamics (CFD:P.Stefano et al. [1954]) have been used to study the behavior of a fluid around an object and its thermal hydraulics. CFD is a technique based on the Navier-Stokes equations, as well as on energy and mass conservation laws. Recently, for Die-casting, various approaches to finding the optimal control settings or optimal mold shape by using a CFD simulator have been studied and applied. To obtain the optimal velocity for injection molding to avoid the formation of shrinkage cavities, optimization with CFD simulator is effective.

However, in cases where the problem has many local solutions, the most widely used general genetic algorithm (GA:K.Ikeda et al. [2000]) is highly likely to find a local solution, whereas finding the global optimal solution is difficult. Specifically, the solution space is unstable due to the effect of the loss of significant digits or underflow error, leading to discrepancies between simulation and experiment. Consequently, a simulation with low computational cost does not always correspond to actual production of a high-quality product.

The aim of this study is to design a solution search algorithm that requires a smaller population to find a solution that can be applied to the production of high-quality products. Thus, we propose a multi-subcenters solution search algorithm to compute the optimal plunger velocity for die-casting. The effectiveness of the proposed system is shown through simulation and experimental results.

2. MULTI-SUBCENTERS SOLUTION SEARCH ALGORITHM

In the case of optimization with few search points, the distribution of search points is important for the derivation of the global optimal solution. Conventionally, search points are selected at random. However, the distribution of such search points is uneven, as shown in Fig. 1, where the shaded areas in the figure contain no search points. Such areas without search points make it difficult to find the global optimal solution with GA, leading to greater computational cost.

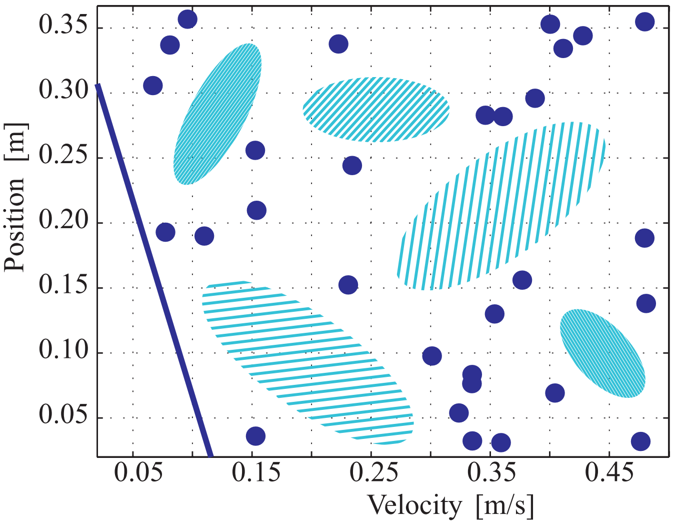

Fig. 1. Setting the distribution of search points by using random generation.

Thus, the search points should be distributed uniformly. On the other hand, in order to improve the convergence performance, the search points should be densely located in areas that are expected to contain the global optimal solution. Therefore, to satisfy these criteria, we developed the multi-subcenters solution search algorithm, which features distribution control S.Martinez et al. [2007], convergence control and cluster analysis capabilities, where the distribution control algorithm is responsible for the distribution of search points and the convergence control algorithm regulates their density.

2.1 Distribution algorithm

Fig. 2 shows the basic concept of the distribution algorithm. The search points are moved by repulsive forces, which are represented as extending circles in the figure.

In this distribution algorithm, each search point is first placed in a unique circle, which expands as the calculation progresses, as described by Eq.(1), where $R(\lambda)$ is the radius of the circle, R_{add} is the expansion factor added to the radius, λ is the number of cycles and $\boldsymbol{m_q}$ is the motion vector of the search point.

$$R(\lambda+1) = \begin{cases} \forall R(\lambda) + R_{\text{add}}, & \text{if } \forall \boldsymbol{m_q} = 0 \\ \forall R(\lambda), & \text{otherwise} \end{cases} \tag{1}$$

When a circle touches another circle or the boundary, repulsive forces arise and spread the circles apart. The search points are moved by the repulsive forces in accordance with Eq.(2), Eq.(3) and Eq.(4), where F_{rep} is the repulsive force, $\boldsymbol{q}$ is a search point, $\boldsymbol{p}$ is another search point, F_{lim} is the repulsive forces reacting from the boundary, B_{min} and B_{max} are respectively the upper and lower boundaries, $\boldsymbol{q_k}$ is the k-th column in vector $\boldsymbol{q}$. Also, n is the number of variables, R_p is the radius of vector $\boldsymbol{q}$ and R_q is the radius of vector $\boldsymbol{q}$.

$$\boldsymbol{m}_q = \begin{cases} (\boldsymbol{q}-\boldsymbol{p})/F_{\text{rep}}^2, & \text{if } \|\boldsymbol{q}-\boldsymbol{p}\| \le R_q + R_p \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

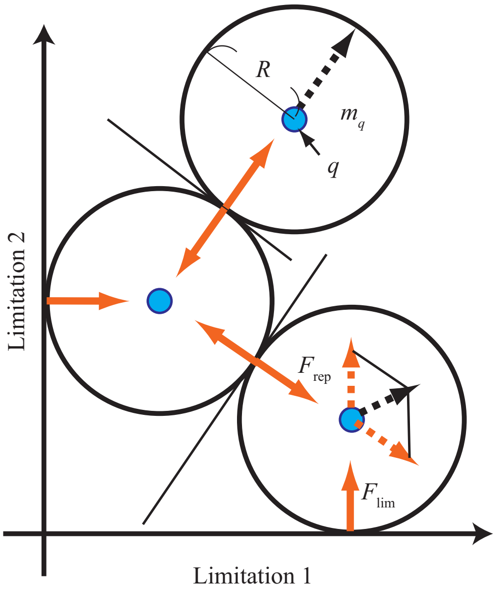

Fig. 2. Basic concept of distribution algorithm.

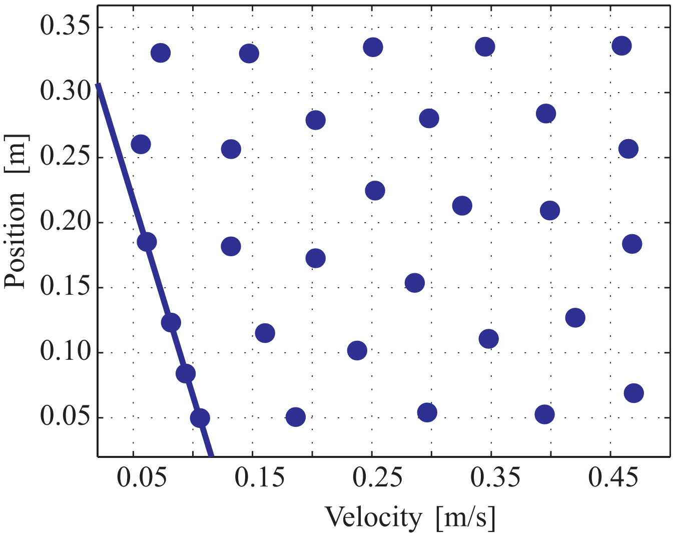

Fig. 3. Setting the distribution of search points by using the developed distribution algorithm.

$$F_{\text{rep}} = (\|\boldsymbol{q}-\boldsymbol{p}\|)^{-1}, \ (\boldsymbol{q} \ne \boldsymbol{p}) \tag{3}$$

$$\boldsymbol{m}_q = \begin{cases} -F_{\text{lim}}, & \text{if } q_k - R \le B_{\text{min},k} \\ F_{\text{lim}}, & \text{if } q_k + R \ge B_{\text{max},k} \\ 0, & \text{otherwise} \end{cases} \quad (1 \le k \le n) \tag{4}$$

The circles continue to expand until the movement of the search points stops. When this occurs, the search points should be distributed uniformly. Eq.(5) is applied as an additional constraint on the movement of the search points.

$$\boldsymbol{q}(\lambda+1) = \begin{cases} \boldsymbol{q}(\lambda), & \forall f(\boldsymbol{q}(\lambda) + \sum \boldsymbol{m_q}) \notin L \\ \boldsymbol{q}(\lambda) + \sum \boldsymbol{m_q}, & \text{otherwise} \end{cases} \tag{5}$$

Here, f is a constraint function defined by the user, and L is the constraint. The distribution of search points which results form using the distribution algorithm is shown in Fig. 3, where the edges of the figure indicate the upper and lower boundaries, and the inner solid diagonal line indicates the additional constraint condition. In contrast to Fig. 1, the search points are distributed uniformly. Thus, the distribution algorithm can ensure sufficient separation between the search points.

2.2 Convergence Algorithm

After the distribution algorithm, the convergence algorithm is executed, in which the search points converge in accordance with neighboring evaluated points. Fig. 4 shows the basic concept of the convergence algorithm, where the crosses in the figure indicate evaluated points. If there are evaluated points in the vicinity of a search point, the latter is moved as shown in the figure. Here, R_{max} is the effective range of a neighboring evaluated point, which is determined by the maximum value of R used in the distribution algorithm.

Fig. 5 presents the basic concept of the convergence algorithm.

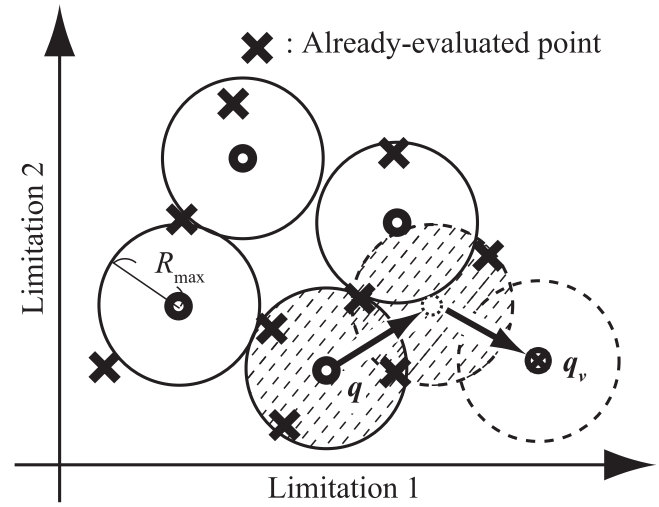

Fig. 4. Basic concept of the convergence algorithm.

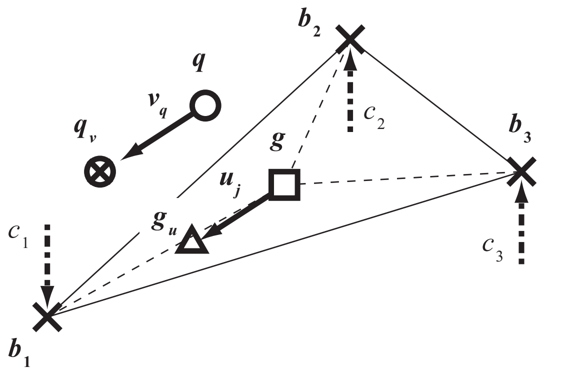

Fig. 5. Basic concept of the movement of search points.

The movement of search points is governed by Eq.(6)∼Eq.(11). The search points of $\boldsymbol{q}$ search for nearby evaluated points present inside the circle. The center of gravity coordinates $\boldsymbol{g}$ of the selected evaluated points are calculated by using Eq.(7). Furthermore, the center of gravity coordinates $\boldsymbol{g_u}$, where the evaluated value of c_i is considered as the load, are calculated by using Eq.(8). Then, the motion vector $\boldsymbol{u_j}$, which is given by Eq.(9) is calculated from Eqs.(7) and (8), where $\boldsymbol{b_i}$ is the position vector of the evaluated point.

$$\boldsymbol{g} = \left(\sum_{i=1}^{n+1} \boldsymbol{b_i}\right)(n+1)^{-1} \tag{6}$$

$$\boldsymbol{g_u} = \left(\sum_{i=1}^{n+1} \boldsymbol{b_i} c_i^{-1}\right)\left(\sum_{i=1}^{n+1} c_i^{-1}\right)^{-1} \tag{7}$$

$$\boldsymbol{u_j} = \boldsymbol{g_u} - \boldsymbol{g} \tag{8}$$

The movement of a search point is decided by the motion vector $\boldsymbol{u_j}$, as described by Eqs.(9), (10) and (11).

$$\boldsymbol{v_q} = \begin{cases} 0 & (d = 0 \text{ or } \forall f(\boldsymbol{q} + \boldsymbol{v_q}) \notin L) \\ \left(\sum_{j=1}^{d} \boldsymbol{u_j}\right) d^{-1} & \text{otherwize} \end{cases} \tag{9}$$

$$\boldsymbol{q_v} = \boldsymbol{q} + \boldsymbol{v_q} \tag{10}$$

Here, $\boldsymbol{q_v}$ is the moved search point, $\boldsymbol{v_q}$ is the motion vector of the search point and d is the number of combined evaluated points as given by Eq.(6). If the number of evaluated points a is less than $n+1$, then d=0 is applied, as shown in Eq.(6).

$$d = \begin{cases} {}_aC_{n+1} & (a \geq n+1) \\ 0 & (a < n+1) \end{cases} \tag{11}$$

These calculations are continued until the movement stops. The shifted points become the analyzed points of the next generation.

3. APPLICATION TO DIE-CASTING PROCESS

The effectiveness of the proposed method is tested through application to die-casting. Then, the proposed method is compared with GA. The effectiveness of the optimization method of die-casting was proved by past study(K. Yano et al. [2008]).

At actual casting plants, the multistep velocity can be controlled and the velocity pattern, which has five phases, is derived from past studies(K. Yano et al. [2008], Y.Kuriyama et al. [2009]). The plunger velocity is expressed as shown in Fig. 6. In this study, the velocity is set to v_1,v_2,v_3, and the acceleration distance is set to x_1,x_2, where x_3 is filling position, which takes a constant value of 0.367 m, because of the constraint of the plant.

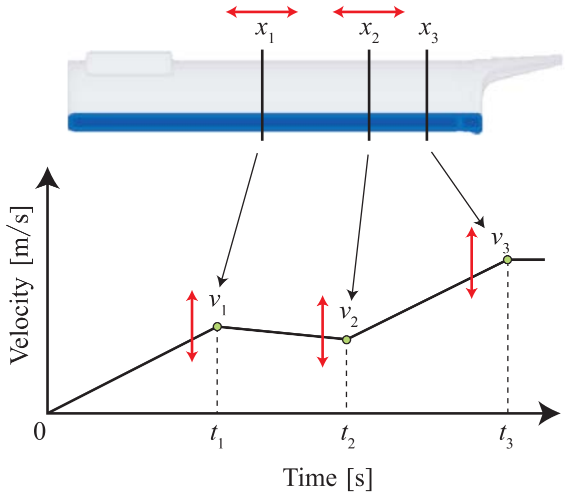

Fig. 6. Die-casting simulation model using 5 variables.

In this study, the plunger tip is flat, and hot-work die steel (SKD61) is used for the die, sleeve and plunger. An aluminum alloy (ADC12) is assumed to be the molten metal. Table 1 shows the fluid properties of ADC12. We set the die temperature during pouring to the range between 110 and 150°C (steady state) and the molten metal temperature in the melting furnace is set to between 660 and 680°C. We used Yushiro AZ7150W as a parting agent.

Fig. 7 shows an overview of the mesh setting, and Table 2 lists the parameters for the mesh setting.

Table 1. Fluid properties of ADC12

Density of fluid [kg/ m^3]	2700
Viscosity of fluid [Pa·s]	0.0030
Specific heat [J/(kg· K)]	1100
Thermal conductivity [W/(m· K)]	100.5
Initial temperature [K]	653.15

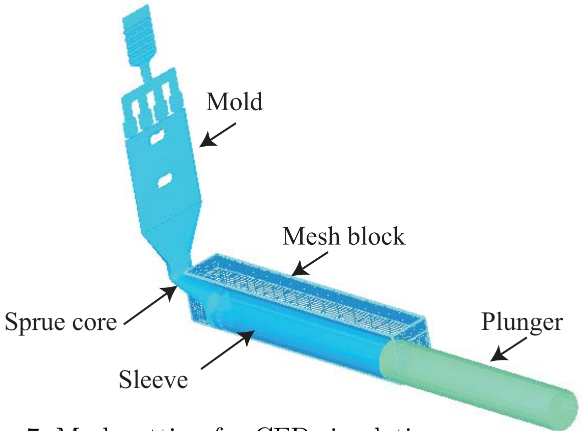

Fig. 7. Mesh setting for CFD simulation.

Table 2. Mesh parameter.

	Cell size	Number of cell
X-direction	0.004	20
Y-direction	0.002∼0.006	132
Z-direction	0.0022∼0.0035	29
Total number of cell		76500

As seen in Fig. 7, the sleeve is symmetrical about the X axis. Thus, the analysis area is set as only a one-sided model to reduce the analysis time to only about ten minutes. Table 2 shows the minimum settings to perform calculations quickly and accurately, and the mesh parameter is set so that a rough mesh is used around the start point of the sleeve because the velocity is low and the fluid is stable in the section. On the other hand, a fine mesh is used around the end point of the sleeve because the turbulence at the early stage of filling caused by collision with the sprue core in the section.

The optimization problem was defined with a cost function equivalent to the sum of the weighted quantity of air entrainment and the weighted filling time, as shown in Eq.(12),

$$\text{minimize}: J = w_a A(v_i, x_i) + w_t t_3 + K_p + A_{\text{shut}} \tag{12}$$

$$\text{subject to}: \begin{array}{c} 0.02 \leq v_i \leq 0.60 \\ 0.02 \leq x_i \leq 0.36 \\ 0 \leq t_i \leq 2.0 \\ A_{\text{shut}} \leq 3.0 \end{array} \tag{13}$$

Here, A is the function of air entrainment, t_3 is the filling time, x_i is the acceleration distance, and $w_a = 1.5$ and $w_t = 1.0$ are weighting factors. K_p is a penalty applied each time the conditions shown in Eq.(13) are satisfied; the penalty $K_p = 10^8$, which is large enough to avoid the penalty conditions, will be added to satisfy the criteria. Also, A_{shut} is the volume of trapped air, not including the air surrounding the sleeve wall, plunger and molten metal, when the plunger injection is switched from low speed to high speed. A_{shut} is defined as shown in Fig. 8.

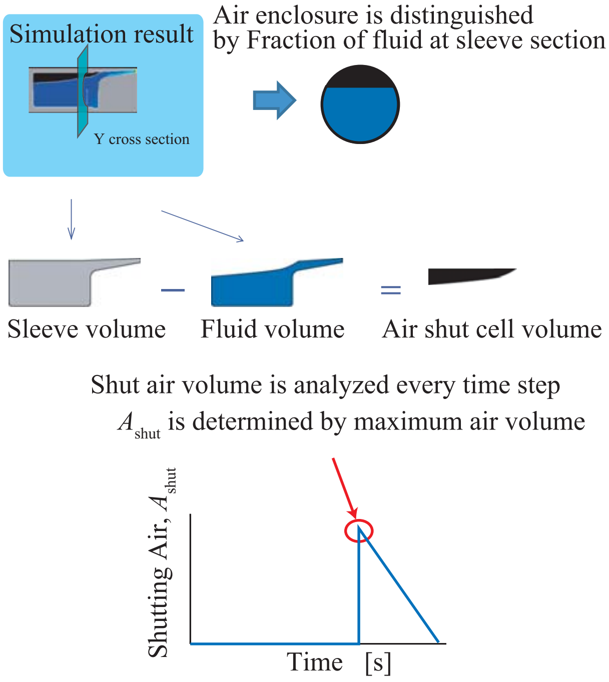
Simulation result
Y cross section
Air enclosure is distinguished by Fraction of fluid at sleeve section
Sleeve volume
−
Fluid volume
=
Air shut cell volume
Shut air volume is analyzed every time step
A_{shut} is determined by maximum air volume
Shutting Air, A_{shut}
Time [s]

Fig. 8. Distinction of the shutting the air in the sleeve.

Three parameters are introduced to calculate the amount of air entrapment.

- D_1: Volume/opening column of fluid in the Y cross section.
- D_2: Threshold of air entrapment amount.
- D_3: Calculation time step.

We used fluid analysis software to perform calculations at each time interval specified by D_3 to output the cell column fraction and the fraction of fluid in order to determine the filling per sleeve cross section. We calculated the space volume at the back where the fraction of fluid is less than $D_1 \times 100\%$ for the cross section, and we defined the maximum space volume as the amount of air entrapment A_{shut} m^3. If plunger velocity input is designed to minimize air entrapment by using this simulator, good results can be expected for actual production experiments.

4. VALIDATION OF THE PROPOSED OPTIMIZATION

The performance of the proposed optimization is validated for the case of plunger velocity controlled by two variables. The velocity is set as shown in Fig. 9

Fig. 10 shows the relationship of the cost function calculated using the CFD simulator for velocity and switching position. In this figure, brown indicates the penalty area and a lower value of the cost function signifies better quality. In this case, the global optimal solution is 0.58 m/s, 0.22 m, However, are is many local solutions around the global solution as shown in the solid circle in the figure.

The proposed optimization algorithm is validated by conducting an experiment by using this solution space, and

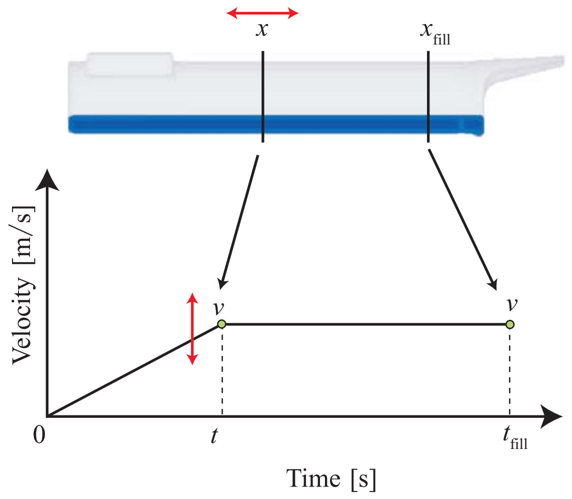

Fig. 9. Die-casting simulation model using 2 variables.

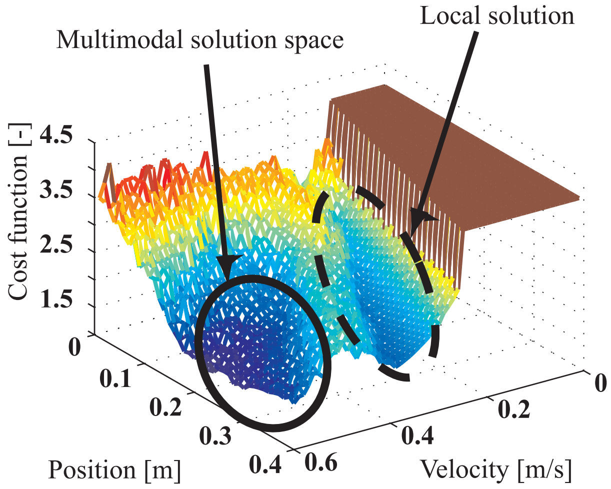

Fig. 10. Relationship of air entrapment with plunger velocity and switching position as calculated with CFD simulator.

GA is used as a target for comparison. We seek to determine which algorithm converges more rapidly to the optimal global solution at 0.58 m/s, 0.22 m, where the maximum number of generations is 1000 and the number of trial is 10, The population size for one generation is 30. Table 3 lists the parameters for the proposed algorithm, and Table 4 lists the parameters for GA.

Table 3. Parameters for proposed method.

Parameter	Numerics
Number of generation	60
Number of population	30
Additional radius, R_{add}	0.05
Force from limitation. F_{lim}	0.1

Table 4. Parameters for GA.

Parameter	Numeric
Number of generation	60
Number of population	30
Number of elite preservation	1
Crossover fraction	0.80
Mutation fraction	0.01

Fig. 11 shows the results of the convergence performance, where the horizontal axis is the number of trials and the vertical axis is the generation when convergence occurred. The left side bar shows the result obtained by using GA, and the right side bar is the result obtained by using the proposed algorithm. In addition, the initial locations of the search points are same. We can see from the figure that in comparison with GA, the proposed algorithm performs better, because the generation of convergence was lower. However, the generation of convergence of the 2nd and 6th trials is 1000, this means that GA became trapped in a local solution.

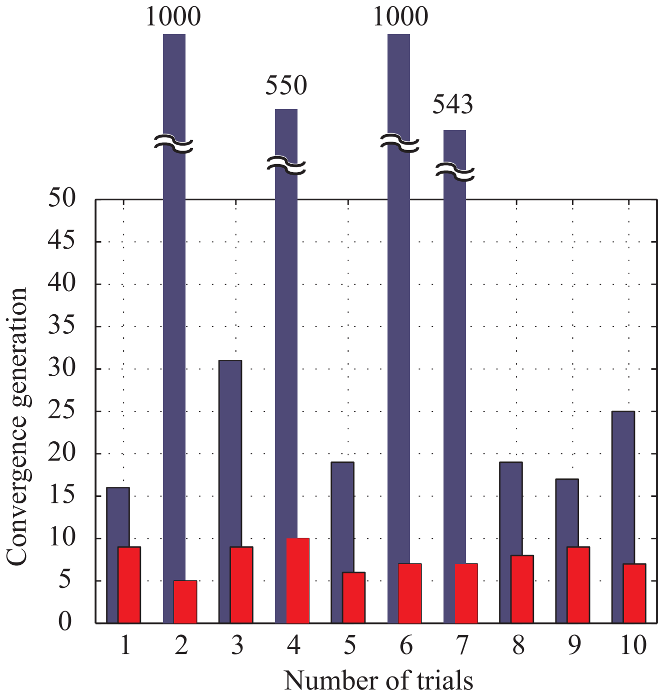

Fig. 11. Convergence generation by using GA and proposed method.

5. OPTIMIZATION OF THE PLUNGER VELOCITY FOR 5 VARIABLE

The parameters for the proposed method are listed in Table 3, and the parameters for GA, which we compare with the proposed method, are listed in Table 4. The initial population is the same for each algorithm, allowing us to perform the calculations under the same conditions.

The results of calculations using the proposed method and GA are shown in Table 5.

Table 5. Performance comparison of first optimization results.

Parameter	Proposed method	GA
Cost function J	1.789	1.794
Air entrainment A	0.193	0.203
Finish time [s]	1.50	1.489
Convergence generation	39	39

6. EXPERIMENTAL RESULTS

We next performed experiments at an actual die-casting plant, using the optimal velocity input calculated by the proposed method and GA. The optimized parameters calculated by using the proposed method and GA are listed in Table 6. A blister test was carried out to investigate the quantity of entrapped air. This was done by heating the specimen in a furnace, which increased the pressure of the entrapped air and formed blisters on the surface. Fig. 12 shows the total area on the test piece surface by using the each calculated optimal velocity where air bubbles formed as a result of the blister test. Here, a solid line indicates an air bubble area over $1{\times}10^{-6}$ m^2, and a broken line indicates air bubble area less than $1{\times}10^{-6}$ m^2, and the conventional input which is used in K. Yano et al. [2008] is indicated as a comparison. From the figure, we can see that the optimal velocity calculated by proposed method resulted in less air bubble formation than that calculated using GA. Moreover, the results in Table 6 show that the proposed method performed better than the GA even if the cost function was similar. The experimental results demonstrate that optimization with the proposed method corresponded accurately to good results in experiment.

Table 6. Experimental first result of optimization.

Proposed method	Velocity [m/s]	Position [m]
i=1	0.41	0.242
i=2	0.34	0.340
i=3	0.59	0.367
Cost function 1.789		
Total area of the air bubble : $0.779{\times}10^{-6}$ [m^2]		
GA	Velocity [m/s]	Position [m]
i=1	0.44	0.279
i=2	0.22	0.290
i=3	0.60	0.367
Cost function : 1.794		
Total area of the air bubble : $0.913\ {\times}10^{-6}$ [m^2]		

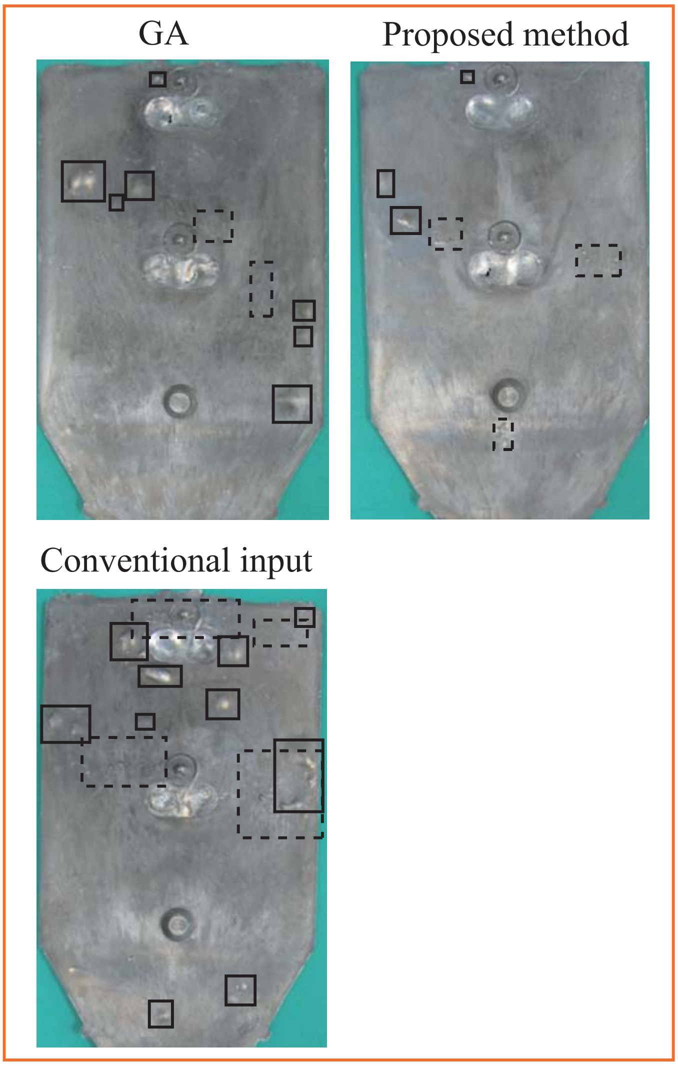

Fig. 12. Experimental result of optimization by using proposed method.

7. CONCLUSION

The purpose of this study was to design a solution search algorithm that requires smaller populations to find the optimal solution corresponding to the production of a high-quality product. We proposed the multi-subcenters solution search algorithm for computing the optimal plunger velocity for die-casting. Experimental results showed that in comparison with GA, the proposed method could better optimize the plunger velocity in die-casting resulting in less air entrapment. Moreover, the results obtained by using the proposed method were better than those obtained using GA even if the cost function was similar. The experimental results demonstrated that optimization with the proposed method corresponded accurately to superior production by die-casting.

REFERENCES

P.Stefano, P.Carlo and M.Martin. Integrating Multibody Simulation and CFD : toward Complex Multidisciplinary Design Optimization. *JSME international journal Series B, Fluids and thermal engineering*, volume 48, pages 224–228, 2005.

K.Ikeda and S.Kobayashi. GA based on the UV-structure Hypothesis and Its Application to JSP *6th Parallel Problem Solving from Nature*, pages 273–282, September 2000.

S.Martinez, J.Cortes and F.Bullo. Motion Coordination with Distributed Information. *IEEE Control System Magazine*, volume 27, pages 75–87, 27, 2007.

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*, volume 2, pages 259–265, 2008.

Y.Kuriyama, S.Hayashi, K.Yano and M.Watanabe Solution search algorithm for a CFD optimization problem with multimodal solution space. *Proc. of IEEE Conference on Decision and Control*, pages 5556–5561, December 2009.