Department of Industrial Engineering and Management,
Yuan-Ze University, Taiwan, R.O.C.

The Production Scheduling and Soft-Computing Lab

A Particle Swarm Optimizer to solve the Continuous Problem

2005/5/12

1. Introduction

The work introduces the particle swarm optimizer to deal with continuous problem. Because the velocity in PSO is the most important parameter, we compare different velocity update methods. Then, the work is the same with OpenSA that uses the concept of Object-Oriented to design the component.

The section 2 describes how to continuous problem by PSO. Besides, we describe various velocity update methods. The section 3 is Object-Oriented analysis that is partially done by the class diagram of unified model language (UML). The experimental result and statistics analysis are presented in section 4. Finally, the discussion and conclusions are drawn in section 5.

2. Solving the Continuous Problem

There are four main procedures of PSO, including generating an initial population, evaluating of the new solutions, updating velocity by global best solution and local best positions, and repeating iteration. The following sub-sections describe them in detail.

2.1 Generate an Initial Population

The method to generate the initial solution is very easy because we only consider the solution should locate in the boundary. The study applies the Himmelblau function as an example. The problem has two dimensions which is named x_1 and x_2, and its corresponding boundary are lie between ± 6. Thus, both the solution of x_1 and x_2 lie between the upper bound and lower bound is valid.
The author uses the following equation to generate each initial solution x_i.

$$x_i = \text{lwBounds}_i + U(0,1)*(\text{upBound}_i - \text{lwBounds}_i) \qquad i = 1, \ldots n \qquad (1)$$

where
n: it's the number of dimension

Therefore, if there is a random value U(0,1) = 0.3, then the value x_i is –6 + 0.3*(6 – (-6)) = -2.4. By doing that, the initial stage is done.

2.2 Evaluating the New Solutions

The Himmelblau function is a two dimension problem which shows as following:

$$Z = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7) + 0.1 \times \left\lfloor (x_1 - 3)^2 + (x_2 - 2)^2 \right\rfloor \qquad (2)$$

Suppose the x_1 is –1.56 and x_2 is 2.6, then the objective value Z = 35.598 + 3.24 + 2.115 = 40.95.

2.3 Updating Velocity

The velocity update strategy is the most important technique of PSO. The study explains the inertia weight in the sub section 2.3.1. Then, some researchers proposed their approach to improve the inertia weight, its revision approach are discussed after the 2.3.2.

2.3.1 Inertia weight

The inertia weight updates current velocity according to global best position among the individuals and its local best position of an individual found so far. Therefore, the equation of inertia weight is shown below:

$$v_{id}^{new} = w_i \cdot v_{id}^{old} + C_1 \cdot rand_1 \cdot (p_{id} - x_{id}) + C_2 \cdot rand_2 \cdot (p_{gd} - x_{id}) \qquad (2)$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \qquad (3)$$

where

d : The dimension of the solution

v_{id}^{new} : The updated velocity

v_{id}^{old} : The old velocity

w_i : Inertia weight of individual i.

C_1 : The parameter represents how much the particle trusts its own past experience.

C_2 : The parameter represents how much the particle can trust the swarm.

$rand_1$ and $rand_2$: The random number.

p_{id} : The current best solution found so far of an individual.

p_{gd} : The current best solution found so far among all individuals.

x_{id} : The dimension d of solution x_i

2.3.2 The V_{max} Approach

The inertia weight might produce higher velocity, so that it causes the solution may out of boundary. Therefore, the purpose of V_{max} restricts the velocity that doesn't move too far away. The equation 2 can be applied here directly, then, the different places are as following:

$$\text{If } v_{id} > V_{max},\ v_{id} = V_{max} \qquad (3)$$

$$\text{Else if } v_{id} < -V_{max},\ v_{id} = -V_{max} \qquad (4)$$

Fan modified the method a little bit, it is shown at equation 5 and 6:

$$\text{If } v_{id} > (1-(\frac{t}{T})^h) \cdot V_{max},\ v_{id} = (1-(\frac{t}{T})^h) \cdot V_{max} \qquad (5)$$

$$\text{Else if } v_{id} < -(1-(\frac{t}{T})^h) \cdot V_{max},\ v_{id} = -(1-(\frac{t}{T})^h) \cdot V_{max} \qquad (6)$$

where

t: the number of current iteration

T: Total iteration
h: a positive constant and we may set it to 1.

By the revision, we can understand it allows swarm can move far away in the beginning. Then, it gradually decreases so that it won't move too far away.

2.3.3 Constriction Factor Method

The method is related to inertia weight and *Vmax* approach, which multiplies a constant K outsides the velocity update function. It's shown at equation 7, 8, and 9.

$$v_{id}^{new} = K \cdot [v_{id}^{old} + C_1 \cdot rand_1 \cdot (p_{id} - x_{id}) + C_2 \cdot rand_2 \cdot (p_{gd} - x_{id})] \tag{7}$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \tag{8}$$

$$K = \frac{2}{\left| 2 - j - \sqrt{j^2 - 4j} \right|} \tag{9}$$

where
$j = C_1 + C_2,\ j > 4$
(j was set to 4.1, so $K = 0.729$)

2.3.4 Guarantee Convergence PSO

The original version of PSO may encounter the problem of stagnating at a solution. It is because the current global best doesn't move anymore. So the approach is to revise the algorithm to solve the problem and shows at the equation 10 and 11.

$$v_{id}^{new} = w_i \cdot v_{id}^{old} - x_{id} + p_{gd} + r^{old} \cdot r \tag{10}$$

where
r: random number from U (-1, 1)
r : the scaling factor

The r in the beginning is set to $r_0 = 1.0$. Then, the r^{new} is calculated by:

$$r^{new} = \begin{cases} 2r^{old} & if\ \#\,success > s_c \\ 0.5r^{old} & if\ \#\,failures > f_c \\ r^{old} & otherwise \end{cases} \tag{11}$$

The GCPSO only affects the update equation of the global best, it needs to conjunction with other velocity update strategy for non-global best particle Therefore, the study uses the GCPSO conjunctions with *Vmax*.

2.3.5 Change the Global Version to Neighborhood Version

The approach is simply changing the p_{gd} to p_{ld} where p_{gd} is current global optimal and p_{ld} is the neighborhood best position after a move. Therefore, the inertia weight and *Vmax* can be re-written as equation 12, constriction factor method is at equation 13, and GCPSO is at equation 14.

$$v_{id}^{new} = w_i \cdot v_{id}^{old} + C_1 \cdot rand_1 \cdot (p_{id} - x_{id}) + C_2 \cdot rand_2 \cdot (p_{ld} - x_{id}) \tag{12}$$

$$v_{id}^{new} = K \cdot [v_{id}^{old} + C_1 \cdot rand_1 \cdot (p_{id} - x_{id}) + C_2 \cdot rand_2 \cdot (p_{ld} - x_{id})] \tag{13}$$

$$v_{id}^{new} = w_i \cdot v_{id}^{old} - x_{id} + p_{ld} + r^{old} \cdot r \tag{14}$$

2.4 Termination Criterion and Penalty Cost of Infeasible Solution

The termination criterion of the PSO is by number of iterations. Thus, after attaining the number of iterations, the algorithm stops.

We observe a phenomenon caused by PSO is that its solution may out of the boundary. Thus, the solution becomes infeasible. The reason is the larger velocity pulls the solution out of the bound. Therefore, the study would like to add a penalty cost when the solution is infeasible so that PSO may not accept the solution. We show that by the following pseudo code.

```
if $x_{id} < lwBound_d$ do
    obj = obj + $K \cdot (x_{id} - lwBound_d)$
else if $x_{id} > upBound_d$
    obj = obj + $K \cdot (upBound_d - x_{id})$
Endif
```

where
x_{id}: The solution x_i at dimension d.
$lwBound_d$: lower bound at dimension d.
$upBound_d$: upper bound at dimension d.
K: constant value of penalty factor. We set it to 2.

2.5 Conclusion of the Section 2

The section 2 discusses numerous velocity update strategies, such as inertia weight, *Vmax*, constriction factor, GCPSO, and changing the global particle to local best particle, the later on section 4 will do experiments to validate them. The section 3 describes the design phase of the Java component.

3. An Object-Oriented Component Design

3.1 Introduction of the Component and Its Interfaces

The study does a well-designed callable component, which is named OpenPSO. The OpenPSO defines some general interfaces for each procedure of PSO. Basically, the interface looks like a blue print. When the class program implements it, the behaviors defined by the interface show on the class program that is expected. The following table shows these procedures’ corresponding interfaces.

Table 3.1 The purposes and its corresponding interfaces

Purposes	Interface
Control the main procedures of PSO	MainI.java
Velocity update strategy	MoveI
Evaluation of objective function	ObjectiveFunctionI

Take the mainI for example, it is an interface which defines the behavior of main procedures, such as starting PSO, initial stage of PSO, velocity update strategy, calculating objective values, and so on. The figure 4.1 shows the structure of MainI which is presented by UML diagram. It also describes the there are two classes, SingleThreadSA and MainContinuous, implement the Main. Moreover, there is an applications will call the interface, Himmelblau.

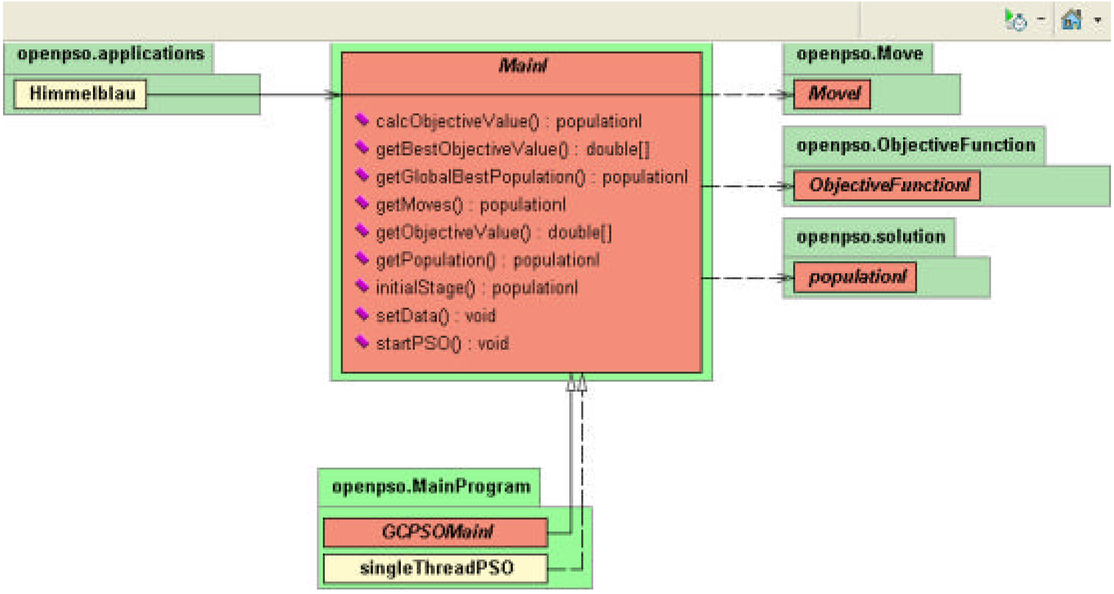

Figure 3.1 The main interface

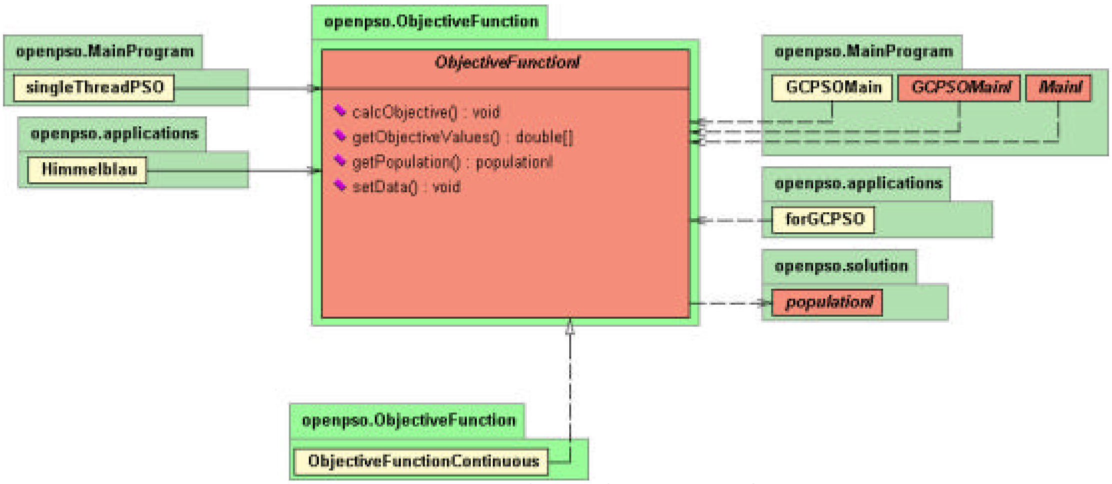

Figure 3.2 The interface of objective function

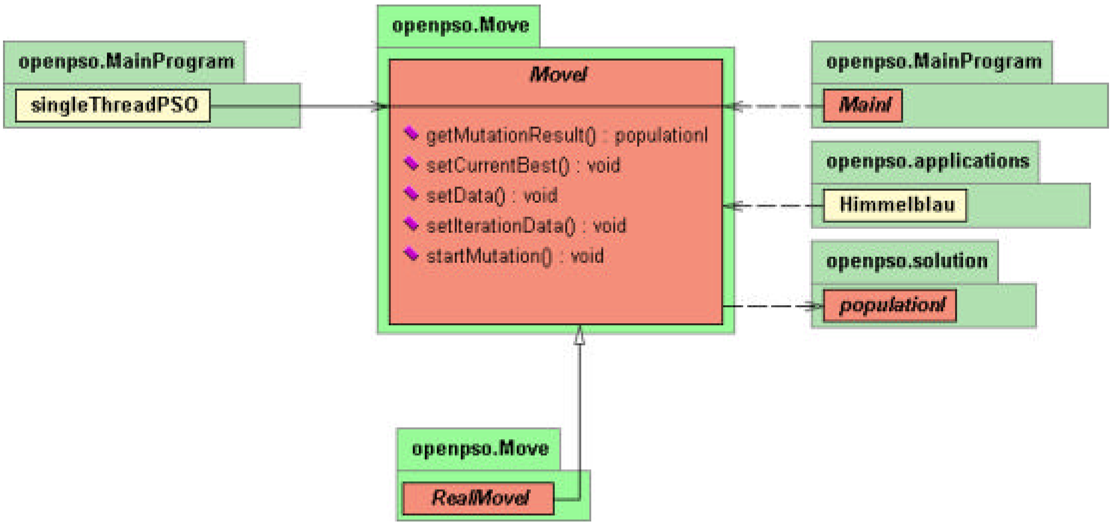

Figure 3.3 The interface of velocity update strategy

3.2 Velocity Update Strategy

The study set the inertial weight as a based class which provides frequently used methods. Therefore, when others class like Vmax, constriction factor, or GCPSO extends this class, they won't declare again. So the programs become simplicity. Of course, we just override the methods which are different so that it performs different behavior from its parent class, inertia weight. The following figures illustrate the structure of these velocity update strategies.

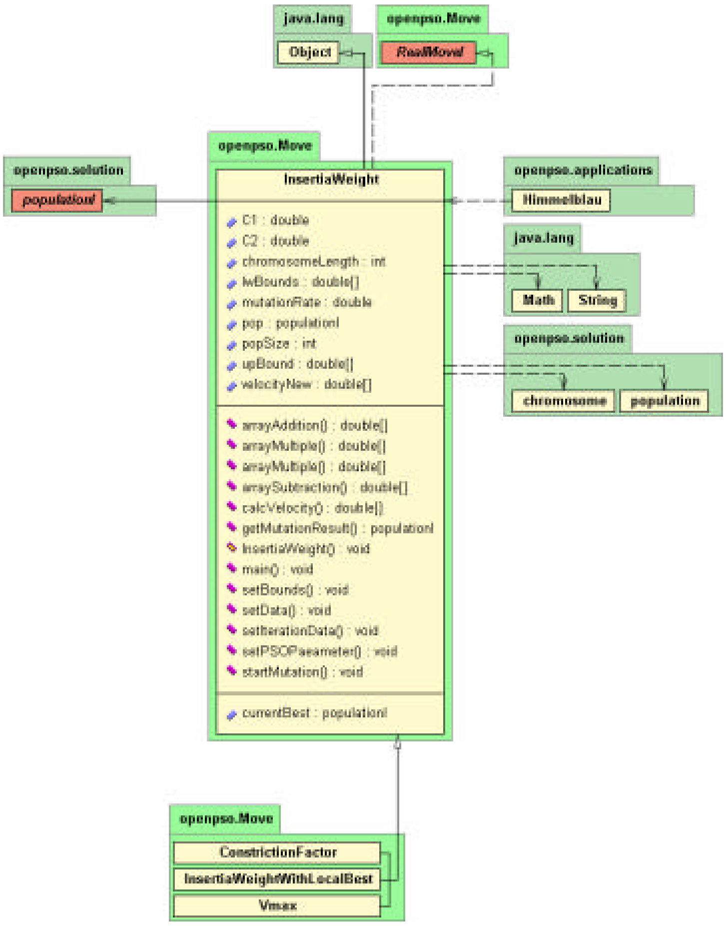

Figure 3.4 The inertia weight

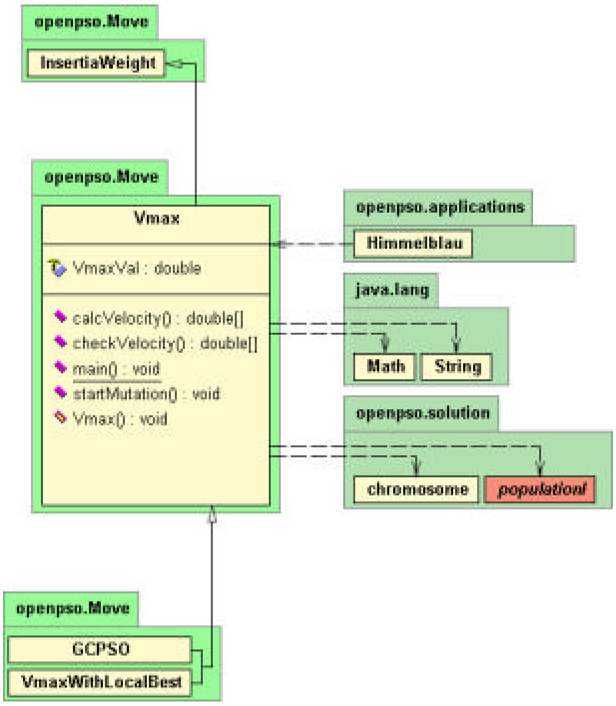

Figure 3.5 The Vmax

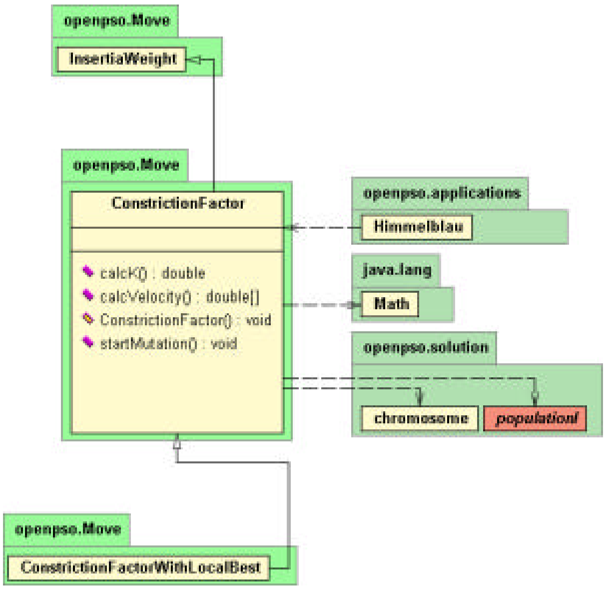

Figure 3.6 The constriction factor

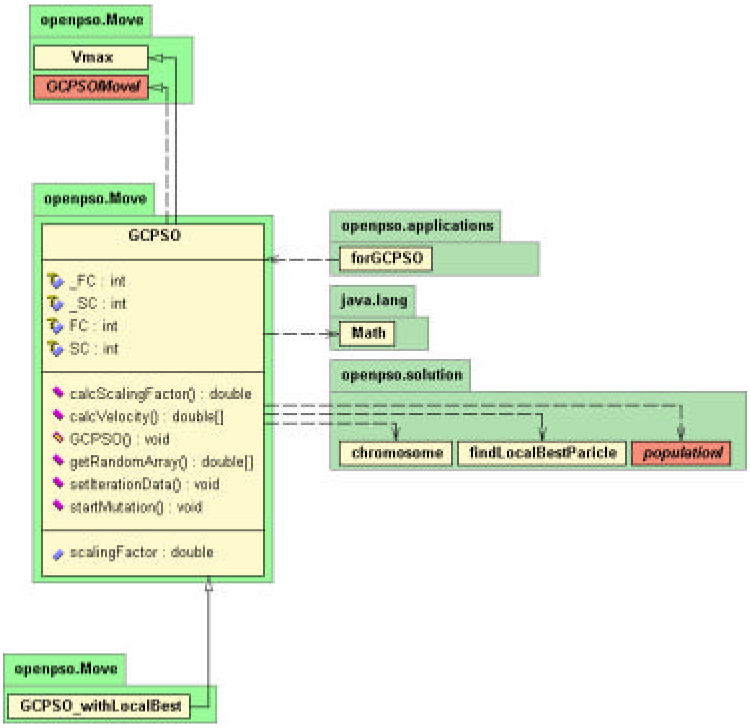

Figure 3.7 The GCPSO

4. Experimental Result

In the experiment, because the study wants to understand the different effect between different velocity update strategies, we verify inertia weight, V_{max}, constriction factor, and GCPSO in the beginning. We present the statistic model in section 4.1.

Then, compare the result of changing the global version to local best position appeared in section 4.2. Finally, because the particles fly freely, it may cause higher velocity so that the infeasible solutions occur. Therefore, we are interested in the effect of reducing the effect of infeasible solution, the experiment is conducted in section 4.3. Above experiment results are analyzed by ANOVA.

There are some parameters used by PSO and are discussed by some researches, we adopt their suggestion of their setting. These parameters shows as following:

Population size: 40
Inertia weight: 0.9
C_1: 2
C_2: 2
number of generations: 200
Sc: 10
Fc: 15

4.1 Compare the Velocity Update Strategy

From above descriptions, it is the one-factor and four treatments experiment. The statistics model can be written as following:

$$T_i = X_i + e_i \tag{15}$$

Where

T_i: The objective value for or Himmelblau function

X_i: The treatments of velocity update strategy i

e_i: The experimental error

The corresponding treatments in the experiment are inertia weight, V_{max}, constriction factor, and GCPSO. The table 4.1 is the data of the experiment and table 4.2 shows the ANOVA result for objective value, and table 4.3 is the ANOVA result of implementation time.

Table 4.1 The result of the first experiment

Num	Velocity update strategy	Time*	x_1	x_2	Obj value
1	ConstrictionFactor	32	2.971552	2.05336	0.049226
2	ConstrictionFactor	32	3.10028	1.814403	0.555297
3	ConstrictionFactor	31	2.812258	2.247712	1.468446
4	ConstrictionFactor	31	3.028019	1.952791	0.040259
5	ConstrictionFactor	31	3.020464	1.947353	0.040398
6	ConstrictionFactor	46	2.92462	2.067994	0.184862
7	ConstrictionFactor	16	2.908456	2.298193	1.485083
8	ConstrictionFactor	16	2.91124	2.136578	0.380049
9	ConstrictionFactor	16	2.795933	2.208239	1.410109
10	ConstrictionFactor	32	3.037932	1.928461	0.084559
11	ConstrictionFactor	15	2.970059	2.032868	0.032008
12	ConstrictionFactor	15	3.003431	1.978314	0.006912
13	ConstrictionFactor	16	3.008422	2.069438	0.099577
14	ConstrictionFactor	32	3.051168	1.929528	0.108918
15	ConstrictionFactor	31	2.973489	2.00429	0.023899
16	ConstrictionFactor	15	2.990891	1.985471	0.009295
17	ConstrictionFactor	16	3.071911	1.828187	0.416788
18	ConstrictionFactor	16	3.008781	2.056715	0.069368
19	ConstrictionFactor	15	2.970406	1.945576	0.113493
20	ConstrictionFactor	32	2.909084	2.187541	0.608844
21	ConstrictionFactor	16	3.127746	1.795704	0.759981
22	ConstrictionFactor	31	3.010128	1.917624	0.098864
23	ConstrictionFactor	16	3.017643	1.936149	0.056838
24	ConstrictionFactor	16	3.561392	-1.66596	1.815548
25	ConstrictionFactor	15	3.026914	1.937714	0.058122

26	ConstrictionFactor	31	3.019399	1.948714	0.03812
27	ConstrictionFactor	16	2.9479	2.027516	0.083527
28	ConstrictionFactor	31	2.983506	1.991049	0.014349
29	ConstrictionFactor	16	3.015625	1.955396	0.02852
30	ConstrictionFactor	32	2.968129	2.036679	0.037308
31	ConstrictionFactor	15	3.087406	1.795557	0.586414
32	ConstrictionFactor	15	3.008258	2.037406	0.033091
33	ConstrictionFactor	16	2.915176	2.139875	0.377901
34	ConstrictionFactor	31	3.020467	1.956664	0.029412
35	ConstrictionFactor	15	2.970838	2.03437	0.031725
36	ConstrictionFactor	32	2.923073	2.090573	0.220871
37	ConstrictionFactor	16	2.993087	2.025593	0.009559
38	ConstrictionFactor	15	2.945548	2.106216	0.194434
39	ConstrictionFactor	31	2.971789	2.039077	0.033777
40	ConstrictionFactor	16	2.9983	1.976699	0.010081
41	ConstrictionFactor	31	3.003841	1.9887	0.001853
42	ConstrictionFactor	15	3.013617	2.006558	0.009437
43	ConstrictionFactor	32	3.056957	1.90191	0.168379
44	ConstrictionFactor	16	2.928175	2.141551	0.347934
45	ConstrictionFactor	31	3.023762	1.87193	0.224817
46	ConstrictionFactor	16	2.896894	2.179872	0.608453
47	ConstrictionFactor	31	3.067626	1.827357	0.411749
48	ConstrictionFactor	15	3.001192	1.926691	0.087093
49	ConstrictionFactor	31	2.917411	2.143934	0.385681
50	ConstrictionFactor	16	2.967972	2.034049	0.035991
51	InertiaWeight	46	2.950817	2.071914	0.10886
52	InertiaWeight	15	2.906279	2.105648	0.318225
53	InertiaWeight	31	3.560691	-1.83047	1.529639
54	InertiaWeight	31	2.948572	2.016111	0.084456
55	InertiaWeight	31	2.812837	2.214706	1.285841
56	InertiaWeight	31	3.010407	1.999546	0.003941
57	InertiaWeight	32	3.047952	1.97005	0.072986
58	InertiaWeight	31	3.008442	1.984625	0.00407
59	InertiaWeight	31	2.968556	2.165972	0.438912
60	InertiaWeight	16	3.582301	-1.76746	1.543878
61	InertiaWeight	31	3.015857	2.070575	0.11996
62	InertiaWeight	32	3.022031	1.939318	0.052695
63	InertiaWeight	32	3.043331	1.954994	0.065553
64	InertiaWeight	31	3.048816	1.876069	0.21737
65	InertiaWeight	31	3.125301	2.000535	0.607701
66	InertiaWeight	15	2.930866	2.153206	0.390453

67	InertiaWeight	32	2.905765	2.085394	0.288422
68	InertiaWeight	16	3.104804	1.70686	1.093851
69	InertiaWeight	16	3.060581	1.831055	0.386674
70	InertiaWeight	32	3.041519	1.89562	0.156014
71	InertiaWeight	15	2.996823	2.029283	0.013373
72	InertiaWeight	31	3.027559	1.965859	0.029238
73	InertiaWeight	16	3.015506	2.058519	0.087421
74	InertiaWeight	16	2.993699	2.017301	0.004447
75	InertiaWeight	15	3.141207	1.814365	0.791288
76	InertiaWeight	16	3.03562	1.879867	0.19588
77	InertiaWeight	16	3.564966	-1.87063	1.560353
78	InertiaWeight	15	2.975614	2.013675	0.018445
79	InertiaWeight	16	2.985631	2.010933	0.006538
80	InertiaWeight	16	3.041049	2.018491	0.084514
81	InertiaWeight	15	3.05544	1.850802	0.306778
82	InertiaWeight	15	3.048169	1.859778	0.26814
83	InertiaWeight	16	2.94251	1.867246	0.553043
84	InertiaWeight	16	2.965693	2.078887	0.099196
85	InertiaWeight	16	2.96318	2.121356	0.225927
86	InertiaWeight	32	2.963006	1.998089	0.05164
87	InertiaWeight	15	3.055989	1.881789	0.212801
88	InertiaWeight	15	3.056514	1.907697	0.156183
89	InertiaWeight	31	3.560841	-2.03644	2.289227
90	InertiaWeight	16	2.990355	2.070615	0.077848
91	InertiaWeight	31	3.029504	1.920886	0.089258
92	InertiaWeight	16	3.036315	1.9749	0.0419
93	InertiaWeight	32	2.988944	2.010472	0.004088
94	InertiaWeight	15	3.54831	-1.74991	1.613369
95	InertiaWeight	31	2.98953	2.065185	0.065222
96	InertiaWeight	16	3.082501	2.033544	0.334839
97	InertiaWeight	31	3.053845	1.965674	0.092232
98	InertiaWeight	15	2.982441	1.924006	0.133089
99	InertiaWeight	32	2.922426	2.152707	0.406787
100	InertiaWeight	16	3.002115	1.988078	0.002079
101	Vmax	16	3.07832	1.949275	0.196663
102	Vmax	15	2.984931	2.091761	0.130752
103	Vmax	16	2.972175	2.037291	0.03189
104	Vmax	16	3.110602	1.991291	0.45205
105	Vmax	16	3.111465	1.857801	0.484868
106	Vmax	32	2.991725	1.908077	0.155945
107	Vmax	31	3.044695	1.975741	0.063408

108	Vmax	15	3.076924	2.024574	0.273668
109	Vmax	31	2.989169	2.142193	0.342284
110	Vmax	31	2.977316	2.077901	0.091004
111	Vmax	16	2.930875	2.075479	0.169809
112	Vmax	15	2.991594	2.051694	0.040694
113	Vmax	15	3.046911	1.843867	0.324999
114	Vmax	15	3.024034	1.875865	0.210964
115	Vmax	16	2.928316	2.044078	0.157139
116	Vmax	16	3.027469	1.952932	0.039489
117	Vmax	15	2.993719	1.980428	0.010404
118	Vmax	16	2.992885	1.964575	0.027998
119	Vmax	16	3.134307	2.042851	0.847802
120	Vmax	15	2.995988	1.974358	0.013757
121	Vmax	31	3.044939	1.961089	0.066445
122	Vmax	16	2.963243	2.030296	0.043191
123	Vmax	16	3.019303	1.966252	0.02007
124	Vmax	16	3.007984	2.0221	0.014348
125	Vmax	16	2.969912	2.112996	0.194738
126	Vmax	15	3.055562	2.042684	0.19628
127	Vmax	31	3.078005	1.957962	0.195287
128	Vmax	16	3.04372	1.978835	0.060958
129	Vmax	31	2.951829	1.976153	0.117176
130	Vmax	15	3.079782	1.828889	0.433192
131	Vmax	16	2.969545	1.983012	0.049266
132	Vmax	16	2.9945	1.959465	0.033126
133	Vmax	15	2.98472	1.907111	0.177908
134	Vmax	16	2.958039	2.122919	0.233901
135	Vmax	31	2.958029	2.004125	0.061312
136	Vmax	15	2.990778	1.989159	0.007141
137	Vmax	15	2.936085	2.067193	0.142174
138	Vmax	16	3.061177	1.966682	0.119422
139	Vmax	16	3.088952	1.931864	0.257471
140	Vmax	31	2.903036	2.165234	0.519046
141	Vmax	16	3.014038	1.863768	0.267044
142	Vmax	31	2.970587	1.995344	0.034891
143	Vmax	15	2.954114	1.834842	0.65647
144	Vmax	31	2.9384	2.057994	0.12565
145	Vmax	16	2.976077	2.059052	0.054006
146	Vmax	31	2.963577	1.945392	0.137757
147	Vmax	16	2.913976	1.963611	0.351058
148	Vmax	32	3.024034	2.077929	0.167106

149	Vmax	15	2.991464	2.098178	0.158273
150	Vmax	31	3.08097	1.993586	0.239881
151	GCPSO	63	3.147024	1.866459	0.733569
152	GCPSO	31	2.958051	2.246321	1.01429
153	GCPSO	31	2.967719	2.096558	0.142239
154	GCPSO	31	3.036474	1.750142	0.818209
155	GCPSO	47	3.557698	-1.84295	1.544524
156	GCPSO	47	2.970419	1.641689	2.079791
157	GCPSO	31	3.210419	1.689076	1.88307
158	GCPSO	31	2.821991	2.407376	3.031541
159	GCPSO	32	3.045145	2.085828	0.286309
160	GCPSO	15	2.965973	2.27908	1.359706
161	GCPSO	31	2.895877	2.191574	0.671535
162	GCPSO	32	3.088912	1.930712	0.25779
163	GCPSO	32	-2.72075	3.099759	3.654019
164	GCPSO	31	2.85381	2.008398	0.732848
165	GCPSO	15	2.710361	2.26194	2.636411
166	GCPSO	16	3.17346	1.852627	1.013466
167	GCPSO	31	3.787573	-1.74503	4.054723
168	GCPSO	16	2.927025	2.137571	0.335672
169	GCPSO	31	2.984323	2.025743	0.012461
170	GCPSO	15	-2.76438	2.964298	4.526099
171	GCPSO	32	2.942227	1.778479	1.124906
172	GCPSO	31	3.087937	1.670217	1.313777
173	GCPSO	16	2.851935	1.967827	0.885871
174	GCPSO	31	2.877316	2.232809	0.989644
175	GCPSO	15	3.019388	2.111051	0.279649
176	GCPSO	16	3.08777	1.678435	1.255592
177	GCPSO	15	2.877063	1.783799	1.772836
178	GCPSO	16	3.128964	2.139735	1.369247
179	GCPSO	15	2.956944	2.017451	0.058085
180	GCPSO	15	3.657969	-2.08192	2.783573
181	GCPSO	16	2.854142	1.966508	0.867337
182	GCPSO	31	2.875133	2.023733	0.506408
183	GCPSO	16	3.045594	1.908946	0.131417
184	GCPSO	31	2.907516	2.216066	0.784493
185	GCPSO	15	2.920338	2.209521	0.717441
186	GCPSO	16	3.035271	1.944898	0.058478
187	GCPSO	31	3.537856	-2.06024	2.585433
188	GCPSO	16	2.800474	1.792304	2.84615
189	GCPSO	31	2.9311	2.151944	0.384487

190	GCPSO	15	3.562458	-1.1371	5.921739
191	GCPSO	32	3.103249	2.296768	2.769024
192	GCPSO	15	2.95538	2.224969	0.827513
193	GCPSO	16	3.067706	2.181641	1.039126
194	GCPSO	15	3.651596	-1.7991	1.784524
195	GCPSO	15	3.067786	1.930045	0.160379
196	GCPSO	16	3.243527	1.900212	2.04648
197	GCPSO	15	3.016794	1.914394	0.102316
198	GCPSO	16	3.010714	2.061672	0.084515
199	GCPSO	15	-2.78104	3.026188	3.893181
200	GCPSO	31	2.946121	2.044039	0.092273

Time*: The 1000 is equal to 1 second.

Table 4.2 The ANOVA of the first experiment for objective value

Source	**DF**	**Seq SS**	**Adj SS**	**Adj MS**	**F**	**P**
Velocity update strategy	3	48.426	48.426	16.142	28.27	0
Error	196	111.924	111.924	0.571		
Total	199	160.35				

Table 4.3 The ANOVA of the first experiment for implementation time

Source	**DF**	**Seq SS**	**Adj SS**	**Adj MS**	**F**	**P**
Velocity update strategy	3	510.12	510.12	170.04	2.2	0.089
Error	196	15127.16	15127.16	77.18		
Total	199	15637.28				

From the table 4.2, it shows there is significant difference among these strategies. By the figure 4.1, we can understand there is no difference that the Constriction factor, inertia weight, and Vmax. If we only compare the average, the Vmax is slightly better than constriction factor, and the constriction factor is slightly better than inertia weight. However, the GCPSO is significantly different from others.

From table 4.3, if we set the confidence level is 95%, it won’t present significant different. However, we can see the P-value is really closed to 0.05. Moreover, by the figure 4.2, the Vmax may use less time to do calculation and the GCPSO uses higher time to finish the job.

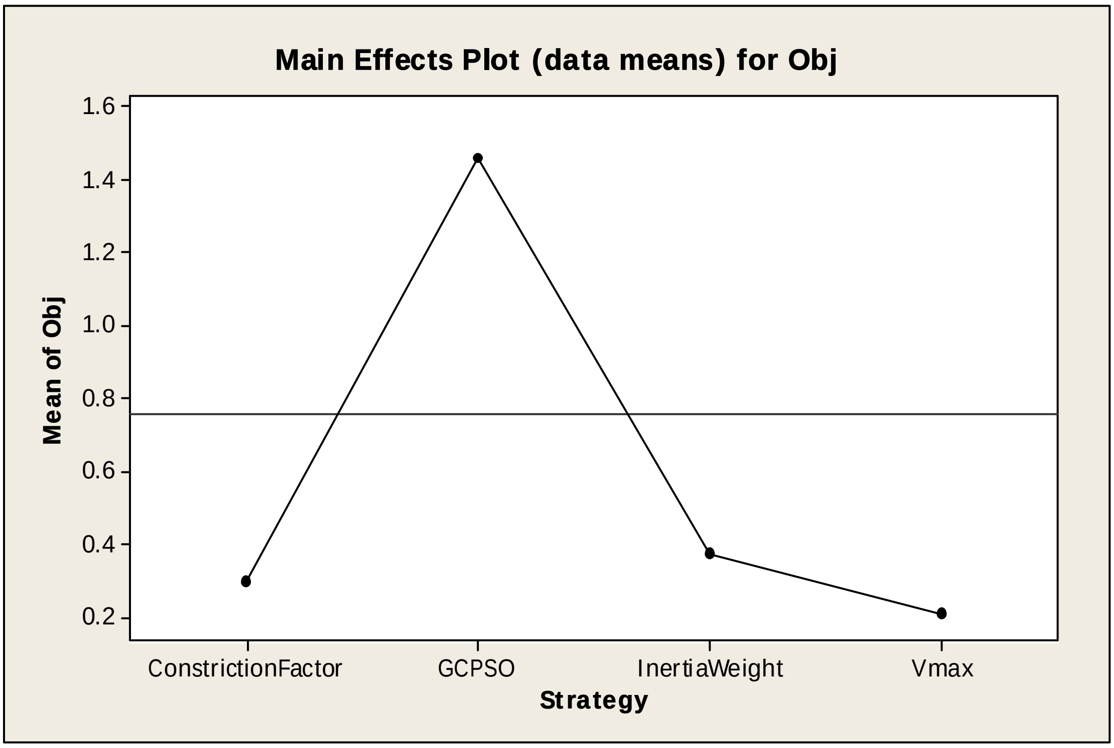

Figure 4.1 The main effect of different velocity update strategy on objective value

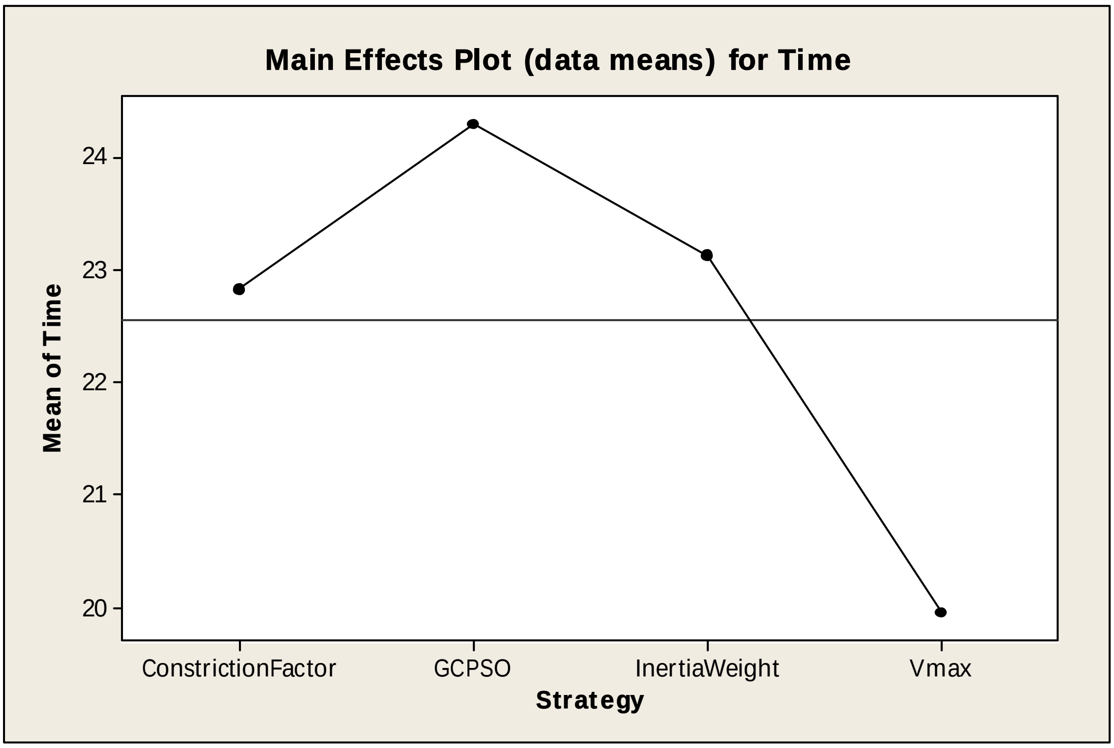

Figure 4.2 The main effect of different velocity update strategy on implementation time

4.2 Compare the Changing the Global Best Position to Local Best Position

We compare the solution quality that uses global best position and it is substituted with local best position. Hence, if we still use 50 runs for the experiment, there will be 400 records (50*4*2 = 400). The statistics model are as following:

$$T_{ij} = X_i + Y_j + (XY)_{ij} + e_{ij} \tag{16}$$

Where

T_{ij}: The objective value for or Himmelblau function under the i and j.

X_i: The treatments of velocity update strategy i

Y_j: The treatments of using the global best or local best solution j

$(XY)_{ij}$: The interaction between two factors X and Y

e_i: The experimental error

The table 4.4 is the experiment of with local best particle and there are 200 records. Then, we adopt the experiment result from the first experiment to do the ANOVA. Therefore, there will be 400 records.

Table 4.4 The result of the second experiment

Num	Velocity update strategy	Time*	x_1	x_2	Obj value
1	ConstrictionFactor	31	3.090513	1.755534	0.785829
2	ConstrictionFactor	16	2.89531	2.130937	0.429517
3	ConstrictionFactor	32	3.036254	1.890728	0.164568
4	ConstrictionFactor	31	3.570795	-1.93084	1.699584
5	ConstrictionFactor	32	2.907192	2.183649	0.593475
6	ConstrictionFactor	15	2.984891	2.015796	0.007953
7	ConstrictionFactor	15	3.049294	1.778635	0.630672
8	ConstrictionFactor	16	2.866854	2.260836	1.236241
9	ConstrictionFactor	31	3.028786	2.086242	0.213624
10	ConstrictionFactor	16	2.877292	2.201294	0.79856
11	ConstrictionFactor	31	3.019623	1.945555	0.04249
12	ConstrictionFactor	31	2.96035	2.005137	0.053973
13	ConstrictionFactor	31	2.987388	2.032054	0.015611
14	ConstrictionFactor	16	3.098157	1.726565	0.962062
15	ConstrictionFactor	31	2.988174	2.027754	0.011938
16	ConstrictionFactor	16	2.958056	2.032923	0.055615
17	ConstrictionFactor	32	3.044978	1.857371	0.274169
18	ConstrictionFactor	15	3.032324	2.019713	0.058686
19	ConstrictionFactor	16	3.040743	1.960176	0.056563
20	ConstrictionFactor	16	2.91968	2.111614	0.277613
21	ConstrictionFactor	15	2.938651	2.039729	0.115729
22	ConstrictionFactor	31	2.971989	2.01295	0.024487

23	ConstrictionFactor	16	2.90295	2.131676	0.39714
24	ConstrictionFactor	16	3.019104	1.92958	0.068886
25	ConstrictionFactor	31	3.06079	1.740193	0.84873
26	ConstrictionFactor	16	2.91531	2.102747	0.273824
27	ConstrictionFactor	31	2.981459	1.986097	0.021101
28	ConstrictionFactor	15	3.066532	1.901702	0.195117
29	ConstrictionFactor	31	2.874846	2.18434	0.726034
30	ConstrictionFactor	16	2.99393	2.013035	0.002704
31	ConstrictionFactor	31	3.066562	1.916295	0.171859
32	ConstrictionFactor	16	2.988324	2.011751	0.004669
33	ConstrictionFactor	32	2.953788	1.885785	0.393235
34	ConstrictionFactor	15	3.023692	1.945622	0.044584
35	ConstrictionFactor	31	2.838681	2.257382	1.350981
36	ConstrictionFactor	16	2.97357	2.001471	0.024956
37	ConstrictionFactor	31	2.997301	1.990605	0.002279
38	ConstrictionFactor	31	3.002953	1.957834	0.02765
39	ConstrictionFactor	16	3.045222	1.906744	0.135346
40	ConstrictionFactor	32	2.973732	1.975503	0.04833
41	ConstrictionFactor	15	2.926795	2.147982	0.37673
42	ConstrictionFactor	31	3.006712	1.989409	0.002162
43	ConstrictionFactor	16	3.028918	1.887186	0.172923
44	ConstrictionFactor	31	3.120506	1.816283	0.648673
45	ConstrictionFactor	15	2.9901	2.12738	0.272363
46	ConstrictionFactor	32	3.053378	1.863991	0.260081
47	ConstrictionFactor	16	2.985208	1.999544	0.008216
48	ConstrictionFactor	31	2.980477	1.973481	0.036235
49	ConstrictionFactor	16	2.994751	2.023316	0.007966
50	ConstrictionFactor	31	3.024533	1.839042	0.354938
51	InertiaWeight	62	3.582628	-1.81834	1.504526
52	InertiaWeight	31	3.555183	-1.85265	1.560774
53	InertiaWeight	15	2.782825	1.943773	1.919315
54	InertiaWeight	15	2.732384	2.283718	2.476844
55	InertiaWeight	15	3.484276	-1.5829	2.523585
56	InertiaWeight	46	3.001502	2.059034	0.06312
57	InertiaWeight	16	3.13461	1.617942	1.769937
58	InertiaWeight	16	2.780643	1.928557	2.049512
59	InertiaWeight	16	3.002893	2.116416	0.251684
60	InertiaWeight	16	2.869068	2.083148	0.515548
61	InertiaWeight	16	3.589557	-1.81638	1.508299
62	InertiaWeight	16	2.989382	2.064605	0.063921
63	InertiaWeight	16	3.014157	2.005807	0.009695

64	InertiaWeight	16	3.595173	-1.65756	1.876937
65	InertiaWeight	31	-3.06801	3.123626	6.265373
66	InertiaWeight	16	3.045649	1.918409	0.113768
67	InertiaWeight	31	3.274292	1.763777	2.595617
68	InertiaWeight	15	2.842652	2.22259	1.105277
69	InertiaWeight	16	2.970035	2.137168	0.292644
70	InertiaWeight	16	3.015018	1.69751	1.27177
71	InertiaWeight	15	2.908593	2.229365	0.87469
72	InertiaWeight	16	2.987196	1.786372	0.764002
73	InertiaWeight	16	2.97881	2.079064	0.093731
74	InertiaWeight	15	3.54873	-1.93256	1.771966
75	InertiaWeight	31	3.008492	2.073033	0.109541
76	InertiaWeight	16	2.869663	2.158384	0.651312
77	InertiaWeight	31	3.589987	-1.56751	2.318283
78	InertiaWeight	15	2.8311	1.892144	1.544635
79	InertiaWeight	32	2.890361	2.04556	0.367287
80	InertiaWeight	16	3.10753	1.885378	0.410484
81	InertiaWeight	31	3.033292	2.220483	1.111479
82	InertiaWeight	16	2.973011	2.149043	0.35219
83	InertiaWeight	31	2.889118	2.050681	0.373184
84	InertiaWeight	15	3.577945	-1.44397	3.135023
85	InertiaWeight	31	3.099596	1.563132	2.172108
86	InertiaWeight	16	2.843816	2.160148	0.831509
87	InertiaWeight	31	-2.91304	3.083372	4.102427
88	InertiaWeight	31	3.560494	-1.88571	1.598356
89	InertiaWeight	16	2.879145	1.992146	0.540698
90	InertiaWeight	31	3.026761	1.959324	0.032817
91	InertiaWeight	15	3.117805	1.873432	0.494712
92	InertiaWeight	31	3.594629	-1.84172	1.517763
93	InertiaWeight	16	3.593408	-1.78452	1.533194
94	InertiaWeight	31	2.964498	2.055866	0.061252
95	InertiaWeight	16	2.884472	2.080186	0.406333
96	InertiaWeight	16	3.535565	-2.1089	3.053867
97	InertiaWeight	15	2.999367	2.061314	0.065378
98	InertiaWeight	31	3.574178	-1.81283	1.507638
99	InertiaWeight	16	3.047348	2.108809	0.401999
100	InertiaWeight	31	3.073314	1.967573	0.174134
101	Vmax	32	3.0344	1.973603	0.037986
102	Vmax	31	3.050323	1.902845	0.152324
103	Vmax	31	2.934356	2.09178	0.185982
104	Vmax	32	2.984776	1.999752	0.008633

105	Vmax	32	2.988	2.031842	0.015262
106	Vmax	32	2.899077	2.102606	0.347288
107	Vmax	31	2.969491	2.064572	0.068129
108	Vmax	31	3.030416	1.97296	0.030552
109	Vmax	31	2.99035	2.096784	0.15212
110	Vmax	31	2.963192	2.026078	0.042261
111	Vmax	31	2.992251	1.891068	0.211618
112	Vmax	16	3.015224	2.019321	0.020985
113	Vmax	15	2.989671	2.110263	0.199712
114	Vmax	31	3.012001	2.084547	0.152975
115	Vmax	16	3.070537	1.976596	0.164919
116	Vmax	15	2.901472	2.084598	0.309575
117	Vmax	15	3.010699	1.962112	0.020291
118	Vmax	16	2.845547	2.095706	0.71122
119	Vmax	16	3.046513	2.012727	0.096173
120	Vmax	16	2.985708	2.041743	0.02596
121	Vmax	16	2.981396	2.094922	0.138173
122	Vmax	15	3.095944	1.960535	0.302185
123	Vmax	15	3.076656	1.999447	0.222598
124	Vmax	31	3.016863	1.892648	0.162034
125	Vmax	16	2.800616	2.121503	1.167614
126	Vmax	31	2.982314	2.070523	0.074339
127	Vmax	16	2.940903	2.047237	0.110346
128	Vmax	32	3.045685	1.818314	0.432411
129	Vmax	15	3.036234	2.04005	0.10647
130	Vmax	31	3.112108	1.986582	0.456024
131	Vmax	16	3.10144	1.791297	0.649516
132	Vmax	31	3.084637	1.918132	0.244735
133	Vmax	15	3.040976	2.012239	0.075781
134	Vmax	32	2.918934	2.248897	1.013674
135	Vmax	16	3.022233	1.937569	0.055541
136	Vmax	31	2.994236	2.031187	0.014501
137	Vmax	16	3.039476	1.974979	0.049352
138	Vmax	16	3.022055	1.955971	0.031267
139	Vmax	31	2.999359	2.034537	0.020299
140	Vmax	15	3.04422	1.989948	0.066386
141	Vmax	32	2.925699	2.118708	0.277284
142	Vmax	16	2.957505	2.096726	0.150739
143	Vmax	31	2.994839	2.096478	0.157376
144	Vmax	16	2.967417	1.96949	0.07442
145	Vmax	16	2.996411	2.052041	0.044169

146	Vmax	15	2.875884	1.89837	0.963728
147	Vmax	31	2.865704	2.008329	0.619479
148	Vmax	16	2.992788	2.018855	0.005335
149	Vmax	31	3.096596	1.962135	0.30758
150	Vmax	16	2.963145	1.976755	0.075964
151	GCPSO	47	3.455612	-1.58153	2.8016
152	GCPSO	31	3.137639	1.479686	2.932809
153	GCPSO	32	3.048916	1.764436	0.711418
154	GCPSO	31	3.74033	-2.04218	3.416863
155	GCPSO	15	3.757184	-1.87892	3.176033
156	GCPSO	16	3.056984	1.81347	0.457131
157	GCPSO	31	3.098664	1.706265	1.084549
158	GCPSO	31	3.692084	-1.56995	3.160165
159	GCPSO	31	3.075434	1.952804	0.182141
160	GCPSO	16	-2.68346	3.171558	3.902026
161	GCPSO	15	3.191568	1.731418	1.509847
162	GCPSO	32	-2.92094	3.196254	4.266176
163	GCPSO	15	3.179627	1.572715	2.301871
164	GCPSO	31	2.956892	2.200466	0.645688
165	GCPSO	16	2.956687	1.936454	0.190103
166	GCPSO	31	2.604399	2.24998	4.336122
167	GCPSO	47	3.46417	-1.87967	2.299739
168	GCPSO	31	2.961587	2.14684	0.334534
169	GCPSO	32	3.566438	-1.6905	1.726419
170	GCPSO	31	2.861547	1.915928	1.023477
171	GCPSO	16	3.689233	-1.66573	2.571162
172	GCPSO	16	2.939433	2.096462	0.182555
173	GCPSO	15	2.854108	2.289182	1.527431
174	GCPSO	16	3.024013	1.868635	0.2364
175	GCPSO	31	3.494462	-1.85575	1.930308
176	GCPSO	16	3.211681	1.905517	1.519523
177	GCPSO	31	3.122895	2.039695	0.709429
178	GCPSO	15	3.492115	-1.96815	2.330585
179	GCPSO	16	3.052359	2.010994	0.1171
180	GCPSO	15	2.862412	2.147478	0.663252
181	GCPSO	16	2.912236	2.220989	0.808468
182	GCPSO	16	2.972958	1.824982	0.601485
183	GCPSO	31	2.762307	2.154495	1.648008
184	GCPSO	16	3.528155	-1.48974	2.816256
185	GCPSO	16	3.126761	1.66665	1.409375
186	GCPSO	31	2.946529	1.798729	0.943146

187	GCPSO	16	2.924063	2.252113	1.038202
188	GCPSO	31	3.124128	1.756265	0.900513
189	GCPSO	16	2.844186	1.978017	0.931379
190	GCPSO	31	3.638011	-1.94406	1.855186
191	GCPSO	16	3.662001	-1.94222	2.005516
192	GCPSO	31	3.036515	1.70216	1.151647
193	GCPSO	15	2.978166	1.94614	0.089267
194	GCPSO	32	2.79681	1.816665	2.67559
195	GCPSO	15	3.458594	-1.93122	2.541553
196	GCPSO	16	2.921439	2.169114	0.483637
197	GCPSO	15	3.047714	1.928399	0.102362
198	GCPSO	31	3.078176	1.759285	0.744871
199	GCPSO	16	3.193436	1.807153	1.310284
200	GCPSO	16	2.849657	2.062492	0.680799

Time*: The 1000 is equal to 1 second.

Table 4.5 The ANOVA of the Himmelblau function

Source	**DF**	**Seq SS**	**Adj SS**	**Adj MS**	**F**	**P**
Velocity update strategy	3	102.021	102.021	34.007	51.11	0*
Use Local best	1	7.158	7.158	7.158	10.76	0.001*
Velocity update strategy* Use Local best	3	12.297	12.297	4.099	6.16	0*
Error	392	260.801	260.801	0.665		
Total	399	382.277				

By the table 4.5, the “Velocity update strategy, “ “Use Local best,” and the interaction between the “Velocity update strategy “ and “Use Local best” cause significant difference. Hence, we examine the interaction between the “Velocity update strategy “ and “Use Local best” first.

The figure 4.3 indicates except the inertia weight, there is no significant difference when different velocity update strategies that are guided by local best or global best. Hence, the inertia weight with local best particle may cause worsen solution. Finally, the figure 4.4 illustrates the Vmax and constriction factor may better than inertia weight and GCPSO when using the local best particle.

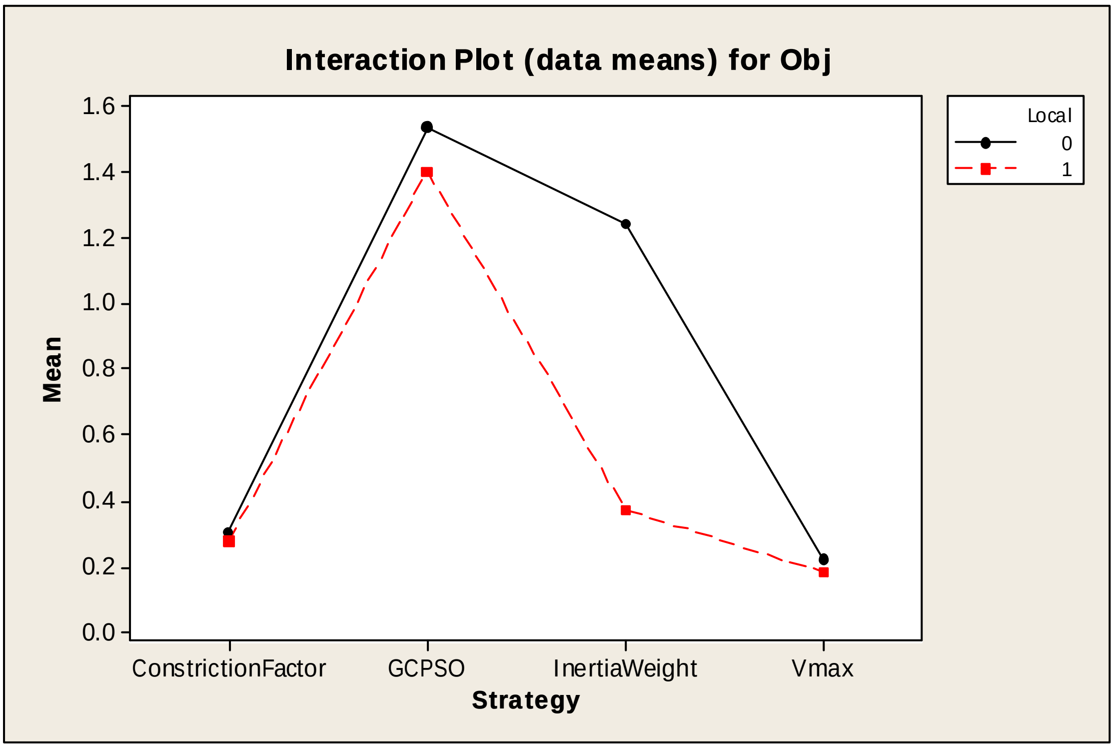

Figure 4.3 the interaction between the "Velocity update strategy " and "Use Local best" where local 0 is guided by local best and 1 is guided by global best

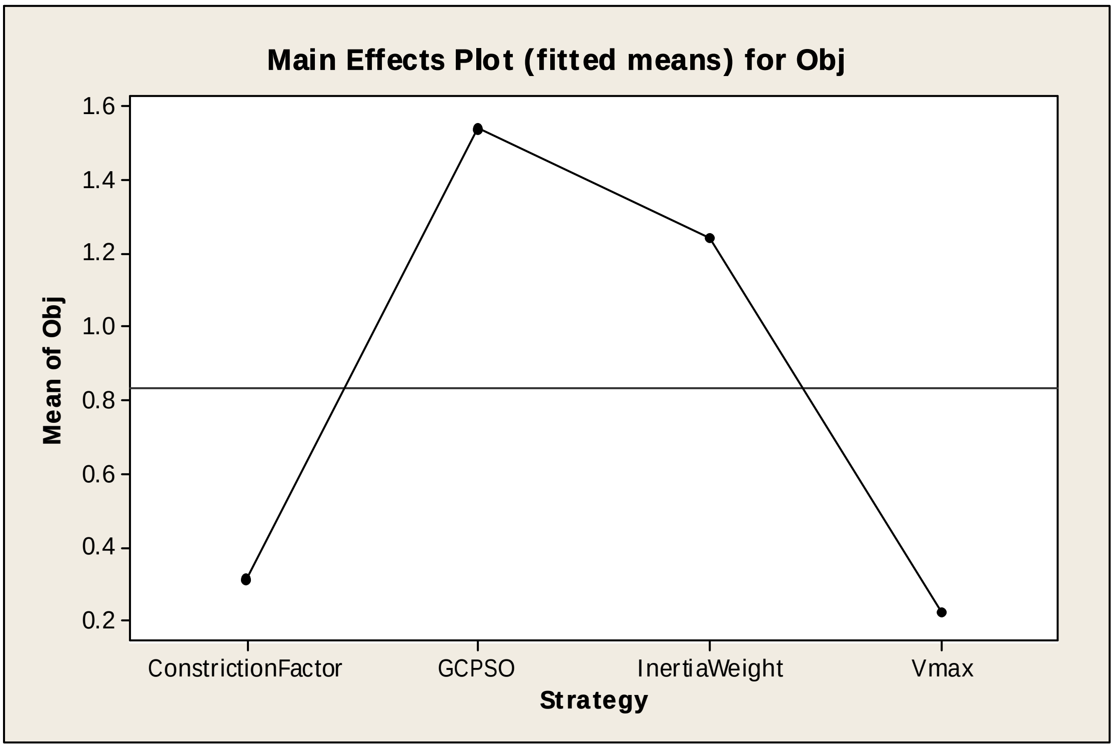

Figure 4.4 The main effect of different velocity update strategy on implementation time

4.3 Compare the Velocity Update Strategy

From above descriptions, it is the one-factor and four treatments experiment. The statistics model can be written as following:

$$T_{ij} = X_i + Y_j + (XY)_{ij} + e_{ij} \tag{17}$$

Where

T_{ij}: The objective value for or Himmelblau function under the i and j.

X_i: The treatments of velocity update strategy i

Y_j: The treatments of using the penalty function. The 0 mean it applies the penalty function. 1 is without using the penalty function.

$(XY)_{ij}$: The interaction between two factors X and Y

e_i : The experimental error

The same with section 4.2, there will be 400 records because we also accept the experiment result of section 4.1 and compare the performance. The result shows at the table 4.6. The ANOVA shows at table 4.7.

Table 4.6 The result of the third experiment

Num	Velocity update strategy	Time*	x_1	x_2	Obj value
1	ConstrictionFactor	31	2.946254	2.14234	0.320894
2	ConstrictionFactor	32	2.986764	1.964463	0.037071
3	ConstrictionFactor	16	2.948815	2.01921	0.082361
4	ConstrictionFactor	16	3.046361	1.847855	0.309556
5	ConstrictionFactor	47	2.990278	2.013086	0.003897
6	ConstrictionFactor	31	2.905499	2.155694	0.470502
7	ConstrictionFactor	31	2.964878	2.051312	0.055255
8	ConstrictionFactor	15	2.942591	2.114581	0.224413
9	ConstrictionFactor	31	2.994772	2.004161	8.74E-04
10	ConstrictionFactor	31	3.093915	1.917645	0.293931
11	ConstrictionFactor	15	2.965673	2.067375	0.076916
12	ConstrictionFactor	15	3.080059	1.798067	0.557674
13	ConstrictionFactor	31	2.90901	2.147442	0.426211
14	ConstrictionFactor	16	3.067815	1.867451	0.277881
15	ConstrictionFactor	32	3.513978	-2.13343	3.487073
16	ConstrictionFactor	15	2.985959	2.013891	0.006701
17	ConstrictionFactor	31	2.982299	2.073849	0.081777
18	ConstrictionFactor	16	3.057776	1.875059	0.234342
19	ConstrictionFactor	31	3.002726	1.948039	0.042511
20	ConstrictionFactor	15	3.031131	1.812854	0.469308
21	ConstrictionFactor	32	3.029088	2.025353	0.05764
22	ConstrictionFactor	16	2.989187	2.021158	0.007473
23	ConstrictionFactor	31	3.015286	1.976898	0.010683
24	ConstrictionFactor	16	2.986945	2.027922	0.012502
25	ConstrictionFactor	16	2.980423	2.029168	0.017446
26	ConstrictionFactor	15	2.963672	2.071246	0.086161

27	ConstrictionFactor	15	3.005098	2.104447	0.207514
28	ConstrictionFactor	32	2.917868	2.089914	0.239975
29	ConstrictionFactor	16	2.927823	2.023157	0.164794
30	ConstrictionFactor	31	2.990048	2.000503	0.003567
31	ConstrictionFactor	16	2.913719	2.06152	0.229236
32	ConstrictionFactor	31	3.614603	-1.8228	1.56193
33	ConstrictionFactor	15	2.925438	2.129147	0.310393
34	ConstrictionFactor	31	3.058021	1.901886	0.171002
35	ConstrictionFactor	32	2.833853	2.258639	1.389249
36	ConstrictionFactor	16	3.012341	2.030983	0.030006
37	ConstrictionFactor	31	3.024724	1.99268	0.020147
38	ConstrictionFactor	16	3.023833	2.059675	0.112524
39	ConstrictionFactor	31	3.107558	1.769409	0.768312
40	ConstrictionFactor	15	2.993506	2.015603	0.003727
41	ConstrictionFactor	31	3.01274	1.875078	0.226076
42	ConstrictionFactor	16	2.993935	2.013565	0.002882
43	ConstrictionFactor	31	3.014329	1.986124	0.006947
44	ConstrictionFactor	16	2.954821	2.124516	0.242027
45	ConstrictionFactor	32	3.052414	1.900814	0.160687
46	ConstrictionFactor	15	3.598833	-1.68187	1.791958
47	ConstrictionFactor	31	3.053768	1.94893	0.097738
48	ConstrictionFactor	16	3.042616	1.904165	0.137157
49	ConstrictionFactor	31	2.984842	2.040457	0.024707
50	ConstrictionFactor	15	3.595812	-1.79476	1.527272
51	InertiaWeight	47	2.979785	2.036965	0.023863
52	InertiaWeight	32	3.079295	1.828223	0.434293
53	InertiaWeight	31	3.040007	1.945665	0.065947
54	InertiaWeight	31	3.070014	1.961244	0.156785
55	InertiaWeight	31	2.760294	2.381061	3.063871
56	InertiaWeight	31	2.906968	2.24711	1.013269
57	InertiaWeight	16	3.059768	1.879798	0.225671
58	InertiaWeight	31	3.118964	1.882358	0.489343
59	InertiaWeight	15	3.108745	1.798082	0.65222
60	InertiaWeight	16	2.844848	2.167738	0.847358
61	InertiaWeight	16	3.028055	1.979587	0.025062
62	InertiaWeight	31	2.979688	1.950494	0.076121
63	InertiaWeight	16	3.046886	1.965798	0.070369
64	InertiaWeight	16	3.0231	2.013611	0.029444
65	InertiaWeight	31	3.078806	1.749947	0.796985
66	InertiaWeight	15	2.923569	2.168735	0.477199
67	InertiaWeight	31	2.939701	2.129609	0.279842

68	InertiaWeight	16	3.027073	1.908856	0.114462
69	InertiaWeight	31	2.973221	2.026305	0.024266
70	InertiaWeight	16	3.016161	2.007404	0.013079
71	InertiaWeight	32	2.935403	2.085349	0.170687
72	InertiaWeight	15	3.057452	1.9455	0.111608
73	InertiaWeight	31	3.588333	-1.70543	1.689961
74	InertiaWeight	16	2.884868	2.120906	0.459401
75	InertiaWeight	16	2.98244	2.05111	0.039109
76	InertiaWeight	15	2.997809	2.006954	7.03E-04
77	InertiaWeight	16	3.013582	1.893651	0.162163
78	InertiaWeight	32	2.989753	2.171564	0.512746
79	InertiaWeight	15	3.594552	-1.78925	1.530125
80	InertiaWeight	31	3.016073	1.984801	0.00867
81	InertiaWeight	16	3.573383	-1.96069	1.812785
82	InertiaWeight	31	2.939801	2.106269	0.206148
83	InertiaWeight	15	3.075999	1.848943	0.354844
84	InertiaWeight	32	3.011972	2.013277	0.011559
85	InertiaWeight	32	2.871383	2.045204	0.508841
86	InertiaWeight	15	2.91813	1.923401	0.462334
87	InertiaWeight	31	3.054692	1.865996	0.255667
88	InertiaWeight	16	3.576204	-1.92642	1.676235
89	InertiaWeight	31	2.844846	2.204253	0.995246
90	InertiaWeight	15	2.936479	2.168234	0.453541
91	InertiaWeight	32	2.918201	2.055793	0.205304
92	InertiaWeight	16	2.946075	2.064358	0.109505
93	InertiaWeight	31	3.074973	1.971084	0.184174
94	InertiaWeight	16	2.980461	2.071601	0.076579
95	InertiaWeight	31	3.041919	1.830075	0.380775
96	InertiaWeight	15	3.050961	1.873397	0.22799
97	InertiaWeight	15	2.923042	2.153125	0.407034
98	InertiaWeight	16	3.071681	1.981711	0.174382
99	InertiaWeight	16	3.00092	2.048086	0.041355
100	InertiaWeight	16	3.006884	2.143694	0.399089
101	Vmax	31	2.996619	2.011017	0.001764
102	Vmax	15	2.958025	2.065586	0.085124
103	Vmax	15	2.957816	2.028508	0.055185
104	Vmax	16	2.913592	2.110544	0.297735
105	Vmax	16	2.990446	1.987641	0.00833
106	Vmax	32	2.917416	2.028708	0.213438
107	Vmax	31	3.026027	2.04472	0.083704
108	Vmax	32	3.014166	1.990312	0.006331

109	Vmax	16	3.053361	1.98637	0.096021
110	Vmax	31	2.974263	2.086089	0.111681
111	Vmax	32	2.957503	2.132055	0.269768
112	Vmax	32	3.041543	1.974482	0.054659
113	Vmax	31	2.954828	1.908104	0.294757
114	Vmax	31	3.04127	2.052132	0.155082
115	Vmax	15	2.943413	1.879534	0.484936
116	Vmax	32	3.089116	2.012974	0.32945
117	Vmax	16	2.899199	2.093339	0.332304
118	Vmax	31	2.974969	2.071804	0.078095
119	Vmax	16	3.114036	1.969756	0.446289
120	Vmax	31	3.023678	2.043439	0.074595
121	Vmax	15	2.957423	2.053681	0.071092
122	Vmax	31	2.973927	1.980344	0.041757
123	Vmax	16	2.983739	2.069501	0.072343
124	Vmax	16	2.923689	1.940236	0.360104
125	Vmax	16	3.084098	1.864099	0.338262
126	Vmax	16	3.008345	1.967153	0.015289
127	Vmax	15	2.98329	1.97945	0.024298
128	Vmax	15	3.010041	1.954615	0.02915
129	Vmax	31	2.96794	2.093128	0.132501
130	Vmax	16	3.023532	1.885027	0.181143
131	Vmax	31	2.977057	2.061351	0.057349
132	Vmax	16	2.963192	1.949988	0.127939
133	Vmax	32	2.945662	2.10781	0.198747
134	Vmax	15	3.014442	1.900666	0.140313
135	Vmax	15	2.934381	2.009945	0.145094
136	Vmax	31	2.887034	2.035915	0.398218
137	Vmax	16	2.98417	1.997605	0.010104
138	Vmax	31	2.950446	2.01555	0.078454
139	Vmax	16	3.036256	1.941823	0.063562
140	Vmax	32	3.080155	1.973068	0.213423
141	Vmax	15	2.980766	1.983195	0.024872
142	Vmax	31	2.995363	2.031376	0.014961
143	Vmax	16	2.975256	1.982733	0.036101
144	Vmax	31	3.071374	1.966579	0.164298
145	Vmax	15	2.840573	2.118543	0.772377
146	Vmax	32	2.847226	2.109141	0.70588
147	Vmax	16	3.013537	1.953551	0.030391
148	Vmax	15	3.035998	1.887836	0.172498
149	Vmax	16	3.032983	1.923346	0.087355

150	Vmax	16	2.862218	1.944595	0.874141
151	GCPSO	62	3.038508	1.925534	0.090108
152	GCPSO	31	2.959755	2.142817	0.31568
153	GCPSO	16	3.475091	-1.8212	2.080982
154	GCPSO	15	2.970043	1.74628	1.154949
155	GCPSO	31	3.637777	-1.81953	1.673522
156	GCPSO	32	3.010556	2.009056	0.007472
157	GCPSO	31	2.99544	2.333162	2.175598
158	GCPSO	31	3.072625	1.827789	0.419811
159	GCPSO	31	3.398396	-1.93605	3.510277
160	GCPSO	16	2.962224	1.700743	1.59578
161	GCPSO	31	3.200931	1.645676	2.001496
162	GCPSO	16	3.582418	-1.83052	1.505678
163	GCPSO	15	2.768309	2.095041	1.572218
164	GCPSO	31	3.036961	1.686049	1.273264
165	GCPSO	32	2.966591	2.009328	0.03624
166	GCPSO	32	3.025372	1.877573	0.204403
167	GCPSO	31	3.059115	1.93873	0.122085
168	GCPSO	15	2.722393	2.41467	3.814836
169	GCPSO	16	3.024723	2.130013	0.395052
170	GCPSO	31	3.589043	-1.72954	1.624768
171	GCPSO	16	2.651184	2.4944	5.726543
172	GCPSO	16	3.131737	2.146986	1.46498
173	GCPSO	31	3.092613	1.97073	0.287713
174	GCPSO	16	2.918768	2.06594	0.208121
175	GCPSO	31	2.949906	2.05676	0.091257
176	GCPSO	15	2.878958	2.391156	2.668537
177	GCPSO	16	3.199643	1.464587	3.266036
178	GCPSO	16	3.536467	-1.72937	1.692777
179	GCPSO	31	2.730291	2.180276	2.10855
180	GCPSO	16	2.901093	1.78146	1.505474
181	GCPSO	31	3.568747	-1.74123	1.591559
182	GCPSO	15	3.054445	1.73413	0.891927
183	GCPSO	32	3.196288	1.668067	1.839875
184	GCPSO	15	3.598534	-1.25657	4.897908
185	GCPSO	16	3.080151	1.856517	0.344843
186	GCPSO	31	3.241461	2.155342	3.558561
187	GCPSO	15	3.096374	1.861717	0.396192
188	GCPSO	32	2.90805	1.986667	0.3317
189	GCPSO	15	3.00101	1.862347	0.300805
190	GCPSO	16	2.948236	1.796731	0.943883

191	GCPSO	15	2.933845	2.166145	0.446374
192	GCPSO	15	3.123326	1.964008	0.51924
193	GCPSO	32	-2.89996	3.027375	4.317782
194	GCPSO	15	2.849767	1.883249	1.359147
195	GCPSO	32	3.226521	1.848144	1.719513
196	GCPSO	15	2.980159	2.224519	0.878668
197	GCPSO	31	3.141421	1.945208	0.670047
198	GCPSO	16	3.022439	2.082575	0.177413
199	GCPSO	16	3.084	2.03378	0.346181
200	GCPSO	16	3.455126	-1.99935	2.951695

Time*: The 1000 is equal to 1 second.

Table 4.7 The ANOVA of the Himmelblau function

Source	**DF**	**Seq SS**	**Adj SS**	**Adj MS**	**F**	**P**
Velocity update strategy	3	98.66	98.66	32.887	53.63	0
Penalty	1	0.237	0.237	0.237	0.39	0.535
Velocity update strategy* Penalty	3	0.094	0.094	0.031	0.05	0.985
Error	392	240.388	240.388	0.613		
Total	399	339.379				

By the table 4.7, only the "Velocity update strategy" cause significant difference and the penalty function doesn't cause significant difference.

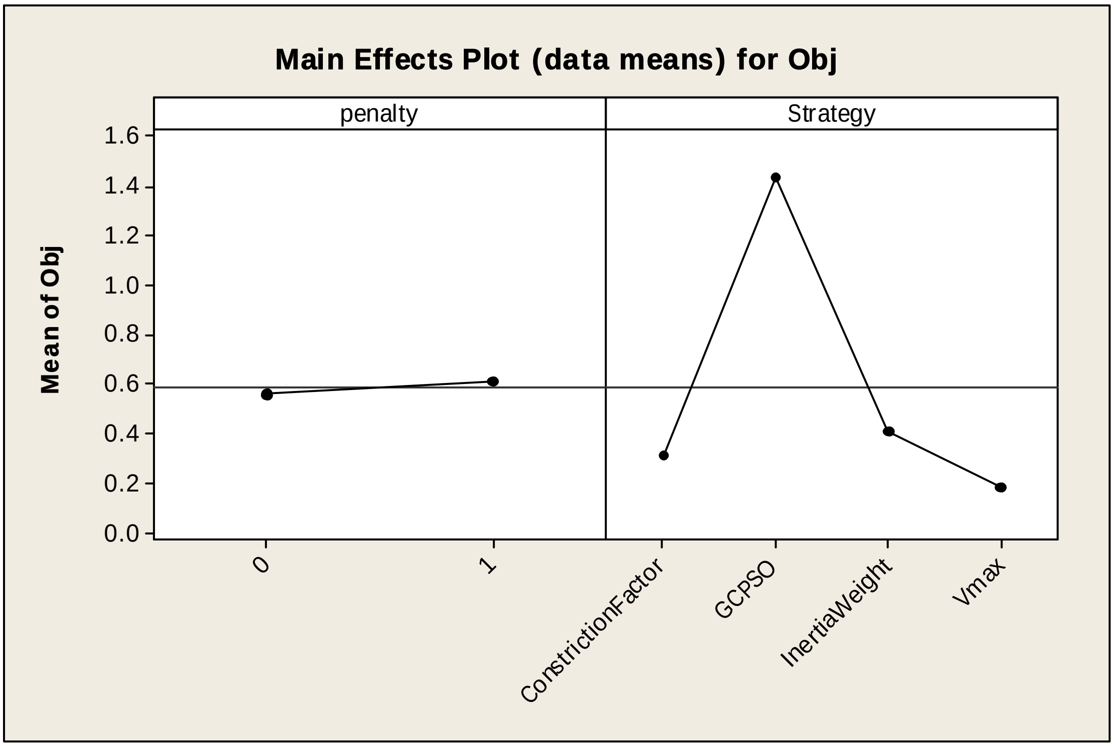

Figure 4.5 The main effect plot of the factors "Velocity update strategy" and "penalty function"

5. Discussion and Conclusions

We compare various velocity update strategy in the study. From the section 4.1, the result shows inertia weight may as good as V_{max}, constriction factor and better than GCPSO. Therefore, it doesn't accordance with the declaration by the previous research. Besides, since it has to find out the current global particle so that GCPSO can update the velocity, it need more O(n) time to accomplish it in every generation. Therefore, it is the reason why it may cause higher computational time.

The second experiment compares the effect of using the global particle or the local best particle. Generally, although we can get smaller objective value in average when using the global best particle, the ANOVA result presents there might not have significant difference between the two strategies. However, when inertia weight uses the local best particle to guide the search, it may cause worse solution. So inertia weight might have less ability to escape current position.

In the last experiment, the study conducts an additional experiment, which compares the result of applying the penalty function or not. The result shows it may not have any significant improvement by the technique. In other words, to prevent the infeasible solution by setting the penalty function may not improve the solution quality. But it may let each particle won't fly too far away.

Finally, we observe the particles sometimes fly far from the boundary so that these solutions become infeasible. Its reason might be the higher velocity pulls the particle strongly out of the boundary. In conventional approach, the search usually is limited within the feasible region. Consequently, because the Vmax deals with the problem to prevent particles fly too far away owing to the higher velocity, the solution quality becomes more ideal. Thus, the future research of PSO should determine a good velocity update strategy that won't cause the particles infeasible.