4. RESULT AND ANALYSIS

In this chapter, the IHDE-BPSO3 algorithm is employed to address the order acceptance and parallel machines scheduling problems. An experimental design is conducted to evaluate the performance of the IHDE-BPSO3 across varying problem sizes and parameter settings. The experiments in this paper are performed on a computer equipped with an Intel i9-9900 CPU operating at 3.10GHz and 16GB of memory. The execution program is coded in C/C++ language and compiled using Visual Studio 2022. For the analysis, Microsoft Excel 2021 is used to interpret calculations, assess solution time, and evaluate the solution quality for each group of experimental factors.

4.1 Proposed IHDE-BPSO3 Structures

There are two proposed structures for the IHDE-BPSO3 algorithm, known as IHDE-BPSO3 type 1 and type 2. In the type 1 strategy, after each mutation and crossover operation, the solution undergoes the local search process. As for the type 2 strategy, the mutant vector does not undergo the local search process before the crossover operation, instead, it directly undergoes the crossover. The reason for developing two different structures is that type 2 has a similar structure to the original IHDE-BPSO3, and the aim is to observe the performance between the two.

4.1.1 IHDE-BPSO3 Type 1

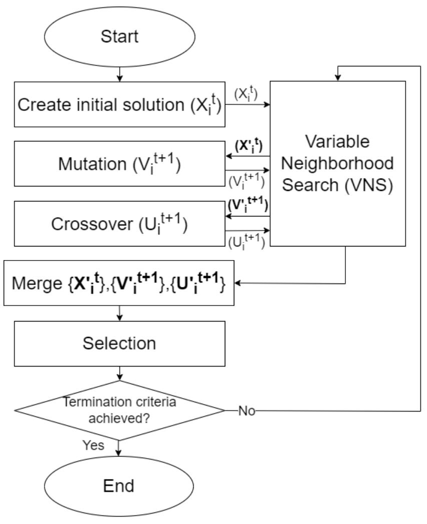
Start
Create initial solution (X_i^t)
(X_i^t)
Variable Neighborhood Search (VNS)
($X'_i{}^t$)
Mutation (V_i^{t+1})
(V_i^{t+1})
($V'_i{}^{t+1}$)
Crossover (U_i^{t+1})
(U_i^{t+1})
Merge {$X'_i{}^t$},{$V'_i{}^{t+1}$},{$U'_i{}^{t+1}$}
Selection
Termination criteria achieved?
No
Yes
End

Figure 4.1 IHDE-BPSO3 Algorithm Type 1 Flowchart

The detailed steps of IHDE-BPSO3 algorithm strategy 1 as follows:

1. Setting IHDE-BPSO3 algorithm parameters including inertia weight ω, weight of personal best c_1, weight of global best c_2, random numbers r_1 and r_2, and crossover probability P_{CR}.
2. Generate number of particles N_k and encode each particle k using permutation coding in Section 3.4.2 to produce:
   - Initial position $X^t_{kij} \in \{x_{k11}, x_{k12},...., x_{k1n}, x_{k22},...., x_{k2n}\}$
   - Initial mutant vector $V^t_{kij} \in \{v_{k11}, v_{k12},...., v_{k1n}, v_{k22},...., v_{k2n}\}$
3. Perform Variable Neighborhood Search (VNS) method as described in Section 3.4.3 to improve the current solution.
   a. Use the order insertion method as described in Section 3.4.3 to improve the current solution, x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (N\_Order + 1))$ times. Then, proceed to step b. If the current solution x does not improve, continue improving it until $(N\_Order \times (N\_Order + 1))$ times, and then go to step b.
   b. Use the machine revised method as described in Section 3.4.3 to improve the current solution x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (M\_Machine + 1))$ times. Then, proceed to step c. If the current solution x does not improve, continue improving it until $(N\_Order \times (M\_Machine + 1))$ times, and then go to step c.
   c. Use the order swap method as described in Section 3.4.3 to improve the current solution x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (N\_Order - 1))$ times. Then, proceed to step d. If the current solution x does not improve, continue improving it until $(N\_Order \times (N\_Order - 1))$ times, and then go to step d.
   d. Use the machine swap method as described in section 3.4.3 to improve the current solution x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (N\_Order - 1))$ times. Then, proceed to step e. If the current

solution x does not improve, continue improving it until $(N\_Order \times (N\_Order - 1))$ times, and then go to step e.

e. If improve $\neq 0$, go back to step a. If improve = 0, proceed to step 4.

4. Store the current solution $\{X'^{t}_{kij}\}$.
5. Record p-best and g-best.
6. Perform mutation to create mutant vector V^{t+1}_{kij} using the mutation formula as explained in Section 3.4.4.
7. Perform Variable Neighborhood Search (VNS) method as described in Section 3.4.3 to improve the current solution.
8. Store the current solution $\{V'^{t+1}_{kij}\}$.
9. Perform crossover to create trial vector U^{t+1}_{kij} using the crossover formula as explained in section 3.4.4.
10. Perform Variable Neighborhood Search (VNS) method as described in Section 3.4.3 to improve the current solution.
11. Store the current solution $\{U'^{t+1}_{kij}\}$.
12. Merge $\{X'^{t}_{kij}\}$,$\{V'^{t+1}_{kij}\}$, and $\{U'^{t+1}_{kij}\}$, sort based on the objective value.
13. Retain the best N_k solutions for the next generation.

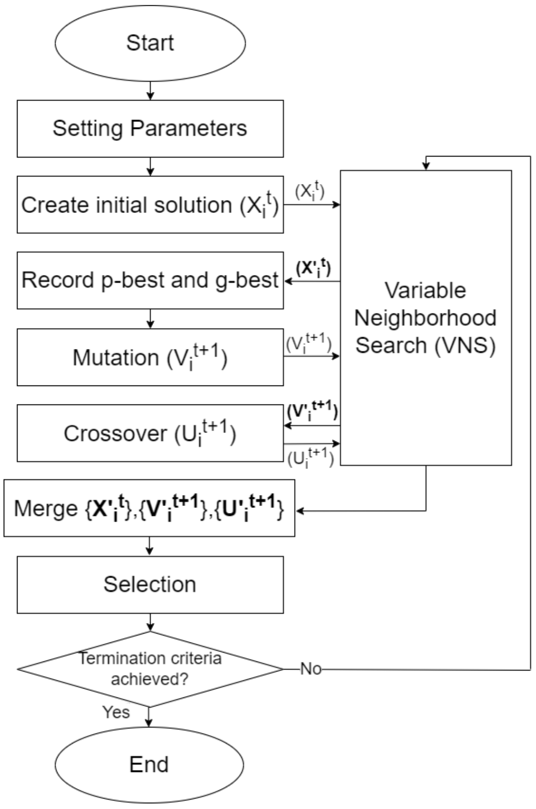

Figure 4.2 IHDE-BPSO3 Algorithm Type 1 Detailed Flowchart

4.1.2 IHDE-BPSO3 Type 2

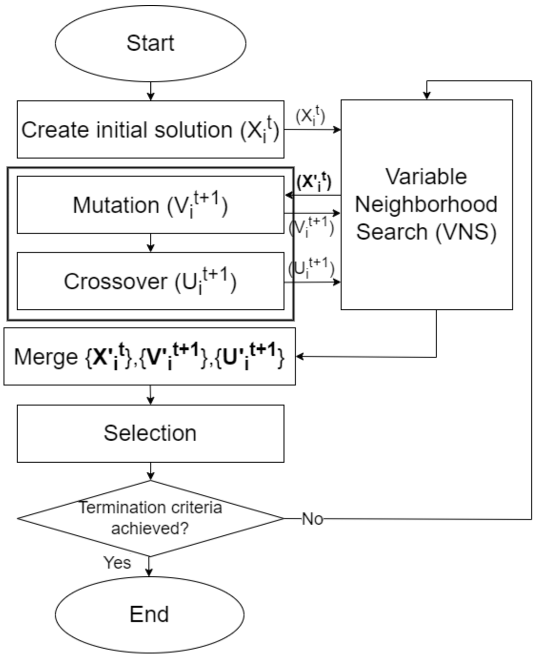
Start
Create initial solution (X_i^t)
(X_i^t)
Mutation (V_i^{t+1})
(X'_i^t)
(V_i^{t+1})
Crossover (U_i^{t+1})
(U_i^{t+1})
Variable Neighborhood Search (VNS)
Merge {X'_i^t},{V'_i^{t+1}},{U'_i^{t+1}}
Selection
Termination criteria achieved?
No
Yes
End

Figure 4.3 IHDE-BPSO3 Algorithm Type 2 Flowchart

The detailed steps of IHDE-BPSO3 algorithm strategy 2 as follows:

1. Setting IHDE-BPSO3 algorithm parameters including inertia weight ω, weight of personal best c_1, weight of global best c_2, random numbers r_1 and r_2, and crossover probability P_{CR}.
2. Generate number of particles N_k and encode each particle k using permutation coding in Section 3.4.2 to produce:
   - Initial position $X^t_{kij} \in \{x_{k11}, x_{k12}, \dots, x_{k1n}, x_{k22}, \dots, x_{k2n}\}$
   - Initial mutant vector $V^t_{kij} \in \{v_{k11}, v_{k12}, \dots, v_{k1n}, v_{k22}, \dots, v_{k2n}\}$
3. Perform Variable Neighborhood Search (VNS) method as described in Section 3.4.3 to improve the current solution.
   a. Use the order insertion method as described in Section 3.4.3 to improve the current solution, x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (N\_Order + 1))$ times. Then, proceed to step b. If the current solution x does not improve, continue improving it until $(N\_Order \times (N\_Order + 1))$ times, and then go to step b.
   b. Use the machine revised method as described in Section 3.4.3 to improve the current solution x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (M\_Machine + 1))$ times. Then, proceed to step c. If the current solution x does not improve, continue improving it until $(N\_Order \times (M\_Machine + 1))$ times, and then go to step c.
   c. Use the order swap method as described in Section 3.4.3 to improve the current solution x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (N\_Order - 1))$ times. Then, proceed to step d. If the current solution x does not improve, continue improving it until $(N\_Order \times (N\_Order - 1))$ times, and then go to step d.
   d. Use the machine swap method as described in Section 3.4.3 to improve the current solution x. If the solution improves, update the current solution x to the improved solution x' and continue improving it until $(N\_Order \times (N\_Order - 1))$ times. Then, proceed to step e. If the current

solution x does not improve, continue improving it until $(N\_Order \times (N\_Order - 1))$ times, and then go to step e.

   e. If improve ≠ 0, go back to step a. If improve = 0, proceed to step 4.

4. Store the current solution $\{X'^{t}_{kij}\}$.
5. Record p-best and g-best.
6. Perform mutation to create mutant vector V^{t+1}_{kij} using the mutation formula as explained in Section 3.4.4. Proceed to step 7 directly to do the crossover operation.
   a. Perform Variable Neighborhood Search (VNS) method as described in Section 3.4.3 to improve the current solution.
   b. Store the current solution $\{V'^{t+1}_{kij}\}$.
7. Perform crossover to create trial vector U^{t+1}_{kij} using the crossover formula as explained in Section 3.4.4.
8. Perform Variable Neighborhood Search (VNS) method as described in Section 3.4.3 to improve the current solution.
9. Store the current solution $\{U'^{t+1}_{kij}\}$.
10. Merge $\{X'^{t}_{kij}\}$,$\{V'^{t+1}_{kij}\}$, and $\{U'^{t+1}_{kij}\}$, sort based on the objective value.
11. Retain the best N_k solutions for the next generation.

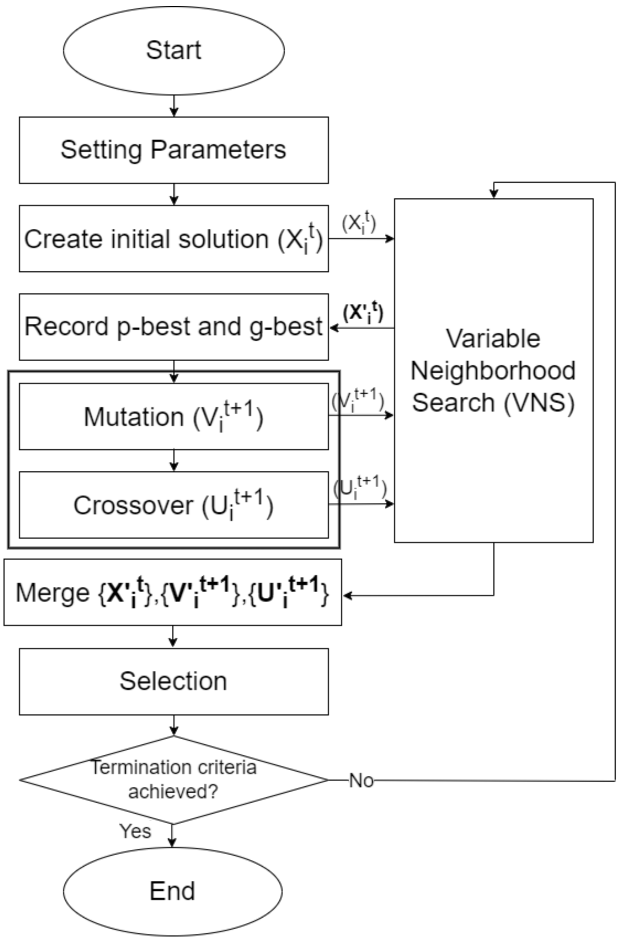

Figure 4.4 IHDE-BPSO3 Algorithm Type 2 Detailed Flowchart

4.2 Problem Parameters

The order acceptance and scheduling problem in this paper is set as follows: when the order quantity is $I = \{10, 11, 12\}$, the number of machines is $M = \{1, 2, 3\}$. Otherwise, when the order quantity is $I = \{30, 50, 100, 150, 200\}$, the number of machines is $M = \{3, 6, 9\}$. The number of particles were set to $N_k = 10$. The problem parameters are generated using a uniform distribution with the upper bound and lower bound as follows:

1. Revenue of order i (r_i) = $U(100, 200)$.
2. Penalty weight of delayed order i (w_i) = $U(2, 10)$.
3. Processing time of order i (p_i) = $U(5, 20)$.
4. Set up time of order i at the beginning (s_{0i}) = $U(2, 8)$.
5. Set up time of order j after order i (s_{ij}) = $U(2, 8)$.
6. Due date of order i (d_i) = $U(15, 60)$.

4.3 IHDE-BPSO3 Parameter Tuning

The IHDE-BPSO3 parameters that needed to be set include: inertia weight ω, weight of personal best c_1, weight of global best c_2, random numbers r_1 and r_2, and crossover probability P_{CR}. The inertia weight ω controls the balance between exploration and exploitation, while $c_1$and c_2 indicate the influence of p-best and g-best, respectively. The crossover probability P_{CR} determines the selection between the mutant vector and the parent vector to create the trial vector. Two different strategies are employed and the results are then compared.

4.3.1 This Study's Strategy

This study tested the original parameter settings for IHDE-BPSO3, as per Wu (2023), based on Equations (2.18), (2.19), and (2.20), where the values of, c1, and c2 change with each iteration. According to Wu (2023), the crossover probability P_{CR} is set to 0.5. Three replications are carried out, and the mean error of the objective values is averaged. To apply these equations, the maximum iterations for each problem are set as follows:

Table 4.1
Max Iterations Setting for This Study's Strategy

No.	No. of Orders	Max Iterations
1	10	25000
2	11	25000
3	12	25000
4	30	1500
5	50	400
6	100	100
7	150	35
8	200	15

4.3.2 Chen's Strategy

Chen (2013) developed a parameter setting strategy for his proposed PSO-VNS algorithm. According to the strategy, if the personal best (p-best) and global best (g-best) values remain unchanged for a specific number of iterations, the inertia weight ω is reduced by 0.01. Additionally, the weight of the personal best component c_1 is increased by 0.01, and the weight of the global best component c_2 is also increased by 0.01. Initially, he set the parameters as shown in Table 4.2 below.

Table 4.2

Initial Parameters Setting for Chen's Strategy

Parameters	Value
Inertia Weight (ω)	0.2
Weight of P-best (c_1)	0.8
Weight of G-best (c_2)	0.8
Random Numbers $(r_1 \& r_2)$	$U(0, 1)$

To optimize the algorithm's performance, parameter testing is conducted using specific problem instances, where the number of orders is represented as $I = \{30, 50, 100\}$, and the number of machines is denoted as $M = \{3, 6,\}$. The objective values from three replications are averaged. After that, the highest objective value is obtained and the differences between this highest value and the lower ones are calculated using Equation 3.16 to obtain the percentage error. Subsequently, the percentage errors are totaled, the parameter that yields the smallest total percentage error is considered the best.

For testing, the initial parameters are set identically to Chen's, as outlined in Table 4.2 and the crossover probability P_{CR} is set to 0.5. The testing begins by varying the number of iterations to determine the best value at which the conditions mentioned in the previous section are satisfied. Following that, the inertia weight value is also adjusted to find the best initial inertia weight for the algorithm. Finally, the crossover probability is adjusted to identify the optimal selection probability to create the trial vector.

4.3.2.1 IHDE-BPSO3 Type 1 Parameter Testing

a. Change Number of Iterations

Testing begins by varying number of iterations, i, from $i = \{10, 20, 30, 40, 50\}$ iterations. The result can be seen in Table 4.3 below.

Table 4.3

Percentage Error when Varying Number of Iterations for IHDE-BPSO3 Type 1

Orders	**Machines**	**% Error**				
		> 50 iter	**> 40 iter**	**> 30 iter**	**> 20 iter**	**> 10 iter**
30	**3**	0.026%	**0.000%**	**0.000%**	0.026%	**0.000%**
30	**6**	0.240%	0.383%	**0.000%**	0.176%	0.054%
50	**3**	0.066%	0.194%	0.258%	**0.000%**	0.107%
50	**6**	0.117%	0.090%	0.005%	0.104%	**0.000%**
100	**3**	0.065%	0.065%	0.065%	0.065%	**0.000%**
100	**6**	0.065%	0.065%	0.065%	0.065%	**0.000%**
Total % Error		0.580%	0.798%	0.394%	0.437%	**0.161%**

From Table 4.3 above, it can be seen that the number of iterations 10 yields the smallest total percentage error. This result indicates that the condition described in Section 4.3.2 is most suitable when applied on 10 iterations. Therefore, the number of iterations is fixed to 10.

b. Change Initial Inertia Weight (ω) Value

For the initial inertia weight, the value is varied from $\omega = \{0.2;\ 0.5;\ 0.8;\ 1.1\}$. The result can be seen in Table 4.4 below.

Table 4.4

Percentage Error when Varying Initial Inertia Weight for IHDE-BPSO3 Type 1

Orders	**Machines**	**% Error**			
		0.2	**0.5**	**0.8**	**1.1**
30	**3**	**0.000%**	**0.000%**	**0.000%**	**0.000%**
30	**6**	0.014%	0.185%	0.027%	**0.000%**
50	**3**	0.226%	0.249%	**0.000%**	0.101%
50	**6**	0.092%	0.366%	0.033%	**0.000%**
100	**3**	0.115%	0.154%	0.000%	0.165%
100	**6**	0.478%	0.227%	0.513%	**0.000%**
Total % Error		0.925%	1.181%	0.573%	**0.267%**

Based on the Table 4.4, it can be seen that the initial inertia weight of 1.1 results in the smallest total percentage error. Consequently, the initial inertia weight is set to 1.1.

c. Change Crossover Probability (P_{CR}) Value

The crossover probability defines the selection proportion for the parent vector and the mutant vector to generate the trial vector. Therefore, the value is varied within $P_{CR} = \{0.2;\ 0.5;\ 0.8\}$. The results can be seen in Table 4.5 below.

Table 4.5

Percentage Error when Varying Crossover Probability for IHDE-BPSO3 Type 1

Orders	**Machines**	**% Error**		
		0.2	**0.5**	**0.8**
30	**3**	0.026%	**0.000%**	0.026%
30	**6**	**0.000%**	0.016%	0.054%
50	**3**	0.226%	**0.000%**	0.116%
50	**6**	0.444%	**0.000%**	0.274%
100	**3**	0.145%	**0.000%**	0.086%
100	**6**	0.631%	**0.000%**	0.228%
Total % Error		1.472%	**0.016%**	0.785%

From Table 4.5, it can be seen that the best value for the crossover probability is 0.5. Therefore, the crossover probability is set to 0.5.

4.3.2.2 IHDE-BPSO3 Type 2 Parameter Testing

Same as type 1, parameter testing is also conducted for the IHDE-BPSO3 type 2.

a. Change Number of Iterations

In this part, The number of iterations is adjusted within $i = \{10, 20, 30, 40, 50\}$ iterations. The results are presented in Table 4.5 below.

Table 4.6

Percentage Error when Varying Number of Iterations for IHDE-BPSO3 Type 2

Orders	**Machines**	**% Error**				
		> 50 iter	**> 40 iter**	**> 30 iter**	**> 20 iter**	**> 10 iter**
30	**3**	0.026%	**0.000%**	0.026%	0.026%	**0.000%**
30	**6**	**0.000%**	0.123%	0.051%	0.155%	0.073%
50	**3**	0.052%	0.262%	0.315%	0.149%	0.000%
50	**6**	0.132%	0.115%	0.114%	**0.000%**	0.332%
100	**3**	**0.000%**	**0.000%**	**0.000%**	**0.000%**	0.138%
100	**6**	**0.000%**	**0.000%**	**0.000%**	**0.000%**	0.008%
Total % Error		**0.211%**	0.500%	0.507%	0.330%	0.551%

From Table 4.6, it can be seen that employing 50 iterations is the most suitable when applying the special conditions outlined in Section 4.3.1 for the IHDE-BPSO3 type 2. Thus, the number of iterations is fixed to 50.

b. Change Initial Inertia Weight (ω) Value

The inertia weight value is varied within $\omega = \{0.2;\ 0.5;\ 0.8;\ 1.1\}$. The result can be seen in Table 4.6 below.

Table 4.7

Percentage Error when Varying Initial Inertia Weight for IHDE-BPSO3 Type 2

Orders	**Machines**	**% Error**			
		0.2	**0.5**	**0.8**	**1.1**
30	**3**	0.026%	**0.000%**	**0.000%**	**0.000%**
30	**6**	0.165%	0.299%	0.138%	**0.000%**
50	**3**	**0.000%**	0.210%	0.094%	0.132%
50	**6**	0.262%	0.127%	0.357%	**0.000%**
100	**3**	0.342%	0.179%	**0.000%**	0.400%
100	**6**	0.190%	**0.000%**	0.219%	0.049%
Total % Error		0.986%	0.814%	0.807%	**0.581%**

Based on Table 4.7, it is shown that an initial inertia weight of 1.1 yields the smallest total percentage error. As a result, the initial inertia weight is set at 1.1.

c. Change Crossover Probability (P_{CR}) Value

The crossover probability value is varied within $P_{CR} = \{0.2;\ 0.5;\ 0.8\}$. The results can be seen in Table 4.8 below.

Table 4.8

Percentage Error when Varying Crossover Probability for IHDE-BPSO3 Type 2

Orders	**Machines**	**% Error**		
		0.2	**0.5**	**0.8**
30	**3**	**0.000%**	**0.000%**	**0.000%**
30	**6**	0.035%	**0.000%**	0.124%
50	**3**	**0.000%**	0.079%	0.149%
50	**6**	0.158%	0.096%	**0.000%**
100	**3**	**0.000%**	0.269%	0.041%
100	**6**	**0.000%**	0.176%	0.189%
Total % Error		**0.193%**	0.620%	0.503%

Table 4.8 indicates that the best crossover probability value is 0.2. This is reasonable since the mutant vector selected in the crossover operation for the IHDE-BPSO3 type 2 did not undergo the local search process beforehand. Hence, the result might not be good. Thus, the crossover probability for IHDE-BPSO3 type 2 is set to 0.2.

4.3.3 Parameters Comparison

Two parameter strategies are compared to find out the suitable parameter setting for IHDE-BPSO3 in this study. For comparison, two parameter settings are tested in mid to large problems with three replications. To compare it fairly, there are two termination conditions that are tried. The first termination condition is when the time reached 60 seconds, the program will stop. The second condition is using maximum iterations. The mean errors are averaged and the one with the smallest total mean error is chosen.

4.3.3.1 Comparison Based on Termination Time

In this section, two parameter settings for both types of IHDE-BPSO3 are compared with a termination time set to 60 seconds as the stopping criteria. The results can be seen in the table below.

Table 4.9

Parameter Settings Comparison for IHDE-BPSO3 Type 1 Based on Termination Time

Case	**IHDE-BPSO Type 1**		**Mean Error**	
	Orders	**Machines**	**Chen (2013)**	**This Study (2023)**
1	**30**	**3**	**0.000%**	**0.000%**
2	**30**	**6**	**0.000%**	0.047%
3	**30**	**9**	0.021%	**0.000%**
4	**50**	**3**	**0.000%**	0.206%
5	**50**	**6**	0.117%	0.182%
6	**50**	**9**	0.143%	0.125%
7	**100**	**3**	0.415%	**0.000%**
8	**100**	**6**	**0.000%**	0.296%
9	**100**	**9**	0.030%	0.211%
10	**150**	**3**	**0.000%**	0.633%
11	**150**	**6**	0.149%	0.022%
12	**150**	**9**	**0.000%**	0.260%
13	**200**	**3**	0.204%	0.355%
14	**200**	**6**	0.101%	0.037%
15	**200**	**9**	0.026%	0.126%
Total Mean Error			**1.206%**	2.501%

Based on Table 4.9, it is shown that Chen's parameter strategy is better than the current strategy, with a total mean error of 1.206% for IHDE-BPSO3 type 1.

Table 4.10

Parameter Settings Comparison for IHDE-BPSO3 Type 2 Based on Termination Time

Case	IHDE-BPSO3 Type 2		Mean Error	
	Orders	**Machines**	**Chen (2013)**	**This Study (2023)**
1	**30**	**3**	0.000%	0.000%
2	**30**	**6**	0.049%	0.034%
3	**30**	**9**	0.000%	0.021%
4	**50**	**3**	0.126%	0.133%
5	**50**	**6**	0.181%	0.000%
6	**50**	**9**	0.097%	0.100%
7	**100**	**3**	0.175%	0.000%
8	**100**	**6**	0.000%	0.283%
9	**100**	**9**	0.171%	0.199%
10	**150**	**3**	0.498%	0.428%
11	**150**	**6**	0.401%	0.041%
12	**150**	**9**	0.022%	0.087%
13	**200**	**3**	0.156%	0.167%
14	**200**	**6**	0.055%	0.099%
15	**200**	**9**	0.000%	0.376%
Total Mean Error			**1.929%**	1.968%

For IHDE-BPSO3 type 2, Chen's parameter strategy is also better than the current strategy, with a total mean error of 1.929%.

4.3.3.2 Comparison Based on Maximum Iterations

In this section, the comparison of two parameter settings for both types of IHDE-BPSO3 uses the maximum iterations as the termination criteria. The results are displayed in the table below.

Table 4.11

Parameter Settings Comparison for IHDE-BPSO3 Type 1 Based on Maximum Iterations

Case	**IHDE BPSO TYPE 1**		**Mean Error**	
	Orders	**Machines**	**Chen (2013)**	**This Study (2023)**
1	**30**	**3**	0.000%	0.000%
2	**30**	**6**	0.000%	0.030%
3	**30**	**9**	0.000%	0.000%
4	**50**	**3**	0.000%	0.206%
5	**50**	**6**	0.117%	0.182%
6	**50**	**9**	0.151%	0.055%
7	**100**	**3**	0.274%	0.000%
8	**100**	**6**	0.000%	0.418%
9	**100**	**9**	0.064%	0.077%
10	**150**	**3**	0.041%	0.355%
11	**150**	**6**	0.101%	0.000%
12	**150**	**9**	0.016%	0.044%
13	**200**	**3**	0.284%	0.000%
14	**200**	**6**	0.028%	0.188%
15	**200**	**9**	0.000%	0.173%
Total Mean Error			**1.076%**	1.728%

As shown in Table 4.11, Chen's parameter is better than current strategy for the IHDE-BPSO type 1 with a total mean error of 1.076%.

Table 4.12

Parameter Settings Comparison for IHDE-BPSO3 Type 2 Based on Maximum Iterations

Case	**IHDE-BPSO3 Type 2**		**Mean Error**	
	Orders	**Machines**	**Chen (2013)**	**This Study (2023)**
1	**30**	**3**	0.000%	0.000%
2	**30**	**6**	0.049%	0.069%
3	**30**	**9**	0.000%	0.000%
4	**50**	**3**	0.126%	0.133%
5	**50**	**6**	0.166%	0.000%
6	**50**	**9**	0.170%	0.087%
7	**100**	**3**	0.175%	0.000%
8	**100**	**6**	0.000%	0.364%
9	**100**	**9**	0.000%	0.328%
10	**150**	**3**	0.224%	0.428%
11	**150**	**6**	0.232%	0.052%
12	**150**	**9**	0.037%	0.132%
13	**200**	**3**	0.190%	0.300%
14	**200**	**6**	0.026%	0.064%
15	**200**	**9**	0.000%	0.356%
	Total Mean Error		**1.396%**	2.312%

As seen in Table 4.12, Chen's parameter is also better than the current strategy for the IHDE-BPSO3 type 2 with a total mean error of 1.076%. After comparing using two different termination criteria, Chen's parameters outperform current strategy in both types of IHDE-BPSO3. Therefore, Chen's parameter strategy is used for the final testing.

4.4 Computational Experiments

To evaluate the algorithm's performance, experiments were conducted. Cases 1-9 represent the small-sized problems, while cases 10-24 represent the mid to large problems. The experimental factors can be seen in Table 4.13 below.

Table 4.13

Experimental Factors

Case	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Orders	10	10	10	11	11	11	12	12	12	30	30	30
Machines	1	2	3	1	2	3	1	2	3	3	6	9
Case	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
Orders	50	50	50	100	100	100	150	150	150	200	200	200
Machines	3	6	9	3	6	9	3	6	9	3	6	9

4.5 Comparison of Two Proposed IHDE-BPSO3s

For the comparison part, the parameter settings for IHDEBPSO3 types 1 and 2 can be seen in Table 4.14 below. These settings are based on the parameter testing results from the previous sections.

Table 4.14

IHDE-BPSO3 Parameters Setting

Parameter	IHDE-BPSO3 Type 1	IHDE-BPSO3 Type 2
Number of Iterations	> 10 iterations	> 50 iterations
Initial Inertia Weight (ω)	1.1	1.1
c_1, c_2	0.8, 0.8	0.8, 0.8
r_1, r_2	$U(0, 1), U(0, 1),$	$U(0, 1), U(0, 1),$
Crossover Probability (P_{CR})	0.5	0.2

The algorithms are then compared using the experimental factors from Section 4.4. Ten replications are conducted, with the termination time is set to 60 seconds. The differences in the objective values for each replication are calculated using Equation 3.16. Subsequently, the percentage errors are averaged, and the result can be represented as the mean error.

Table 4.15

Mean Error of Two Proposed IHDE-BPSO3s for the Small Problems

Case	**Orders**	**Machines**	**Mean Error**	
			IHDE-BPSO3 Type 1	**IHDE-BPSO3 Type 2**
1	**10**	**1**	0.000%	0.000%
2	**10**	**2**	0.000%	0.000%
3	**10**	**3**	0.000%	0.000%
4	**11**	**1**	0.000%	0.000%
5	**11**	**2**	0.000%	0.000%
6	**11**	**3**	0.000%	0.000%
7	**12**	**1**	0.000%	0.000%
8	**12**	**2**	0.000%	0.000%
9	**12**	**3**	0.000%	0.000%

For the small-sized problems, it can be observed that the mean error of both algorithms is 0%. This indicates that both algorithms perform well in small-sized problems and reach the optimal solutions.

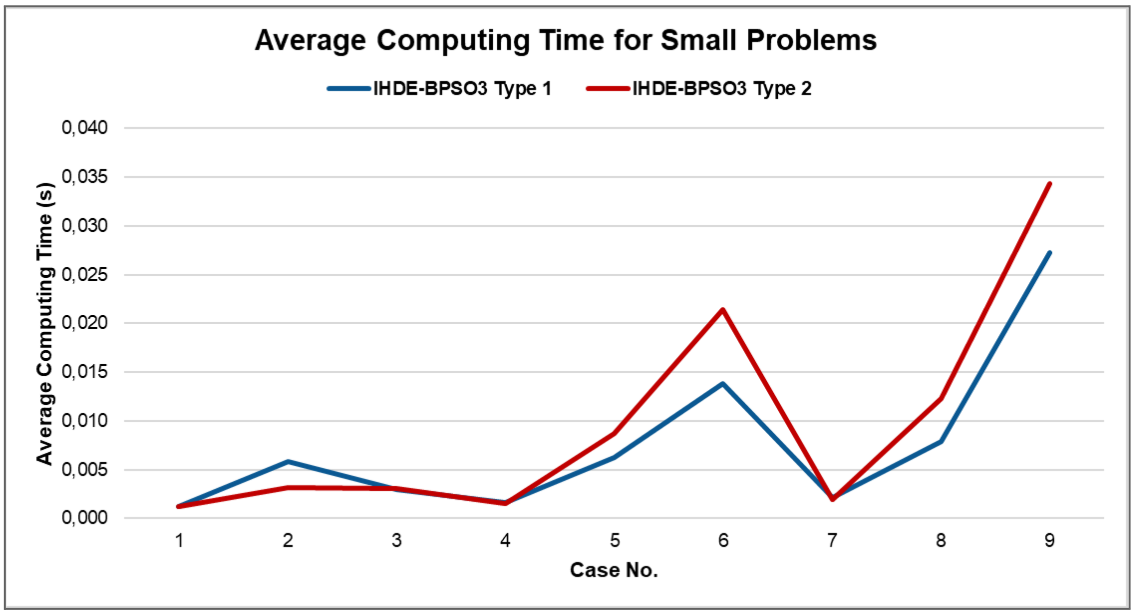

Figure 4.5 Line Chart of Average Computing Time for Two IHDEBPSO3s

Based on the average computing time required to reach the optimal solution for the small problems with ten replications, Figure 4.5 shows that overall, type 1 requires shorter time than type 2 to reach the optimal solution. This means that type 1 converges faster towards the optimal solution than type 2.

Table 4.16

Mean Error of Two Proposed IHDE-BPSO3s for Mid to Large Problems

Case	Orders	Machines	Mean Error	
			IHDE-BPSO3 Type 1	IHDE-BPSO3 Type 2
10	**30**	**3**	**0.000%**	**0.000%**
11	**30**	**6**	0.037%	0.075%
12	**30**	**9**	0.013%	0.001%
13	**50**	**3**	0.061%	0.072%
14	**50**	**6**	0.067%	0.166%
15	**50**	**9**	0.052%	0.086%
16	**100**	**3**	0.174%	0.145%
17	**100**	**6**	0.110%	0.051%
18	**100**	**9**	0.072%	0.143%
19	**150**	**3**	0.300%	0.216%
20	**150**	**6**	0.041%	0.219%
21	**150**	**9**	0.115%	0.047%
22	**200**	**3**	0.134%	0.230%
23	**200**	**6**	0.237%	0.061%
24	**200**	**9**	0.036%	0.145%
Total Mean Error			**1.449%**	1.657%

Table 4.16 presents the mean errors for IHDE-BPSO3 type 1 and type 2 across various problem sizes and machine configurations. In the case of 30 orders, both types have a 0.000% mean error with a 3-machine configuration. However, as the number of machines increases, both types experience an increment in mean errors when the number of machines is 6, followed by a decrease when the number of machines is 9. For the 50 orders scenario, type 1 consistently outperforms type 2 across all machine configurations, indicating that type 1 is more suitable for this size of problems.

Scaling up to the larger problem size, in the case of 100 orders, both algorithms display higher error percentages. Type 2 outperforms type 1 in 3 and 6 machine configurations, indicating its better adaptability to larger problem sizes. The 150 orders problem introduces more variability, with type 1 shows a notable increase, particularly for the 3-machine configuration where the mean error reaches 0.3%, while type 2 outperforms type 1 in 3 and 9 machine configurations. As for the 200 orders scenario, type 1 demonstrates competitive performance, especially for the 3-machine and 9-machine configurations, while type 2 shows an increase in the mean errors.

In summary, IHDE-BPSO3 type 1 outperforms type 2 in mid-sized problems, while type 2 outperforms type 1 several times in large-sized problems. Based on the results, it can be concluded that type 1 is more suitable for mid-sized problems, while type 2 is more suitable for large-sized problems.

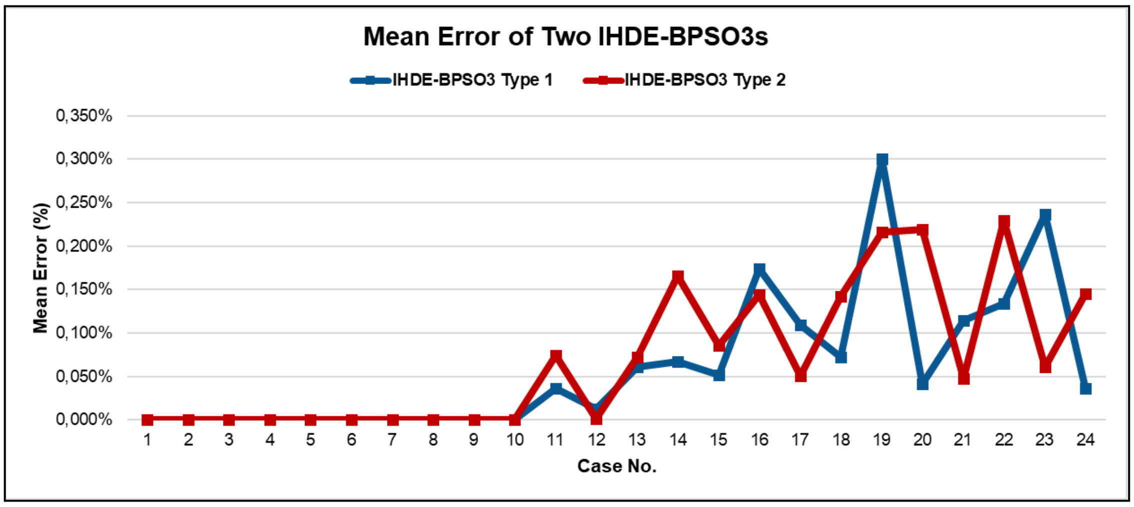

Figure 4.6 Line Chart of Testing Mean Error for Two IHDEBPSO3s

Figure 4.6 illustrates the performance of each type of IHDE-BPSO3 algorithms. For the small problem where the number of orders is $I = \{10,\ 11,\ 12\}$ and the number of machines is $M = \{1,\ 2,\ 3\}$, both algorithms can achieve the optimal solution with the mean error value of 0.000%. However, when the number of orders is increased to $I = \{30,\ 50,\ 100, 150, 200\}$, and the number of machines is $M = \{3,\ 6,\ 9\}$, the algorithms are unable to find the optimal solution, and the mean error starts to increase irregularly.

Hence, when the mean errors are totaled, the IHDE=BPSO3 type 1 yields a smaller total mean error with a value of 1.449% than the type 2 (1.657%). Thus, it can be concluded that the IHDE-BPSO3 type 1 slightly better than the type 2. For some cases, this might happen because the effect of the crossover solutions generated by IHDE-BPSO3 type 2 may not be good due to the absence of the local search process for the mutant vector. Even though the crossover probability for type 2 has been set to 0.2, there is still a chance that the mutant vector will be selected to be the trial vector. As a result, the algorithm might encounter difficulty in generating good populations, resulting in difficulties converging towards the optimal result.

4.6 Comparison with PSO-VNS Algorithm

The IHDE-BPSO3 type 1 are then compared with the well developed classic algorithm, Particle Swarm Optimization (PSO) embedded with the Variable Neighborhood Search (VNS). Ten replications were tested to obtain the mean error of two algorithms. The termination time is set to 60 seconds.

Table 4.17

Mean Error of two Algorithms for Small Problems

Case	**Orders**	**Machines**	**Mean Error**	
			IHDE-BPSO3 Type 1	**PSO-VNS**
1	**10**	**1**	0.000%	0.000%
2	**10**	**2**	0.000%	0.000%
3	**10**	**3**	0.000%	0.000%
4	**11**	**1**	0.000%	0.000%
5	**11**	**2**	0.000%	0.000%
6	**11**	**3**	0.000%	0.000%
7	**12**	**1**	0.000%	0.000%
8	**12**	**2**	0.000%	0.000%
9	**12**	**3**	0.000%	0.000%

For the small problems, both algorithms yield a 0.000% of mean error. This means both of the algorithms can reach the optimal solutions for the small problems.

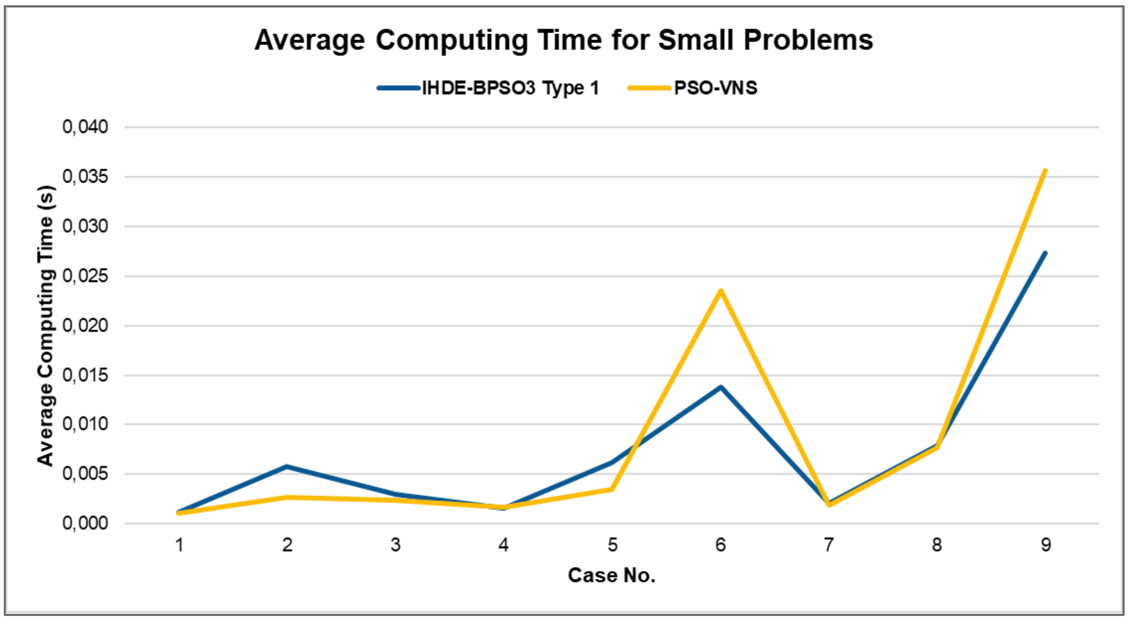

Figure 4.7 Line Chart of Average Computing Time for Two Algorithms

Based on the average computing time required to reach the optimal solution for small-sized problems with ten replications, Figure 4.7 shows that overall, PSO-VNS requires less time than IHDE-BPSO3 Type 1 to reach the optimal solution. However in some cases, as the number of orders increases, PSO-VNS could take more time to reach the optimal solutions.

Table 4.18

Mean Error of Two Algorithms for Mid to Large Problems

Case	**Orders**	**Machines**	**Mean Error**	
			IHDE-BPSO3 Type 1	**PSO-VNS**
10	**30**	**3**	**0.000%**	**0.000%**
11	**30**	**6**	0.048%	0.031%
12	**30**	**9**	0.015%	**0.000%**
13	**50**	**3**	0.056%	0.087%
14	**50**	**6**	0.190%	0.062%
15	**50**	**9**	0.184%	0.016%
16	**100**	**3**	0.283%	0.209%
17	**100**	**6**	0.229%	0.175%
18	**100**	**9**	0.167%	0.079%
19	**150**	**3**	0.214%	0.243%
20	**150**	**6**	0.072%	0.159%
21	**150**	**9**	0.129%	0.113%
22	**200**	**3**	0.270%	0.162%
23	**200**	**6**	0.398%	0.045%
24	**200**	**9**	0.159%	0.113%
Total Mean Error			2.413%	**1.495%**

When tested on mid to large problems, it is shown that the mean error for both algorithms increases with the growing number of orders and machines. PSO-VNS outperforms IHDE-BPSO3 type 1 as the number of orders and machines getting larger.

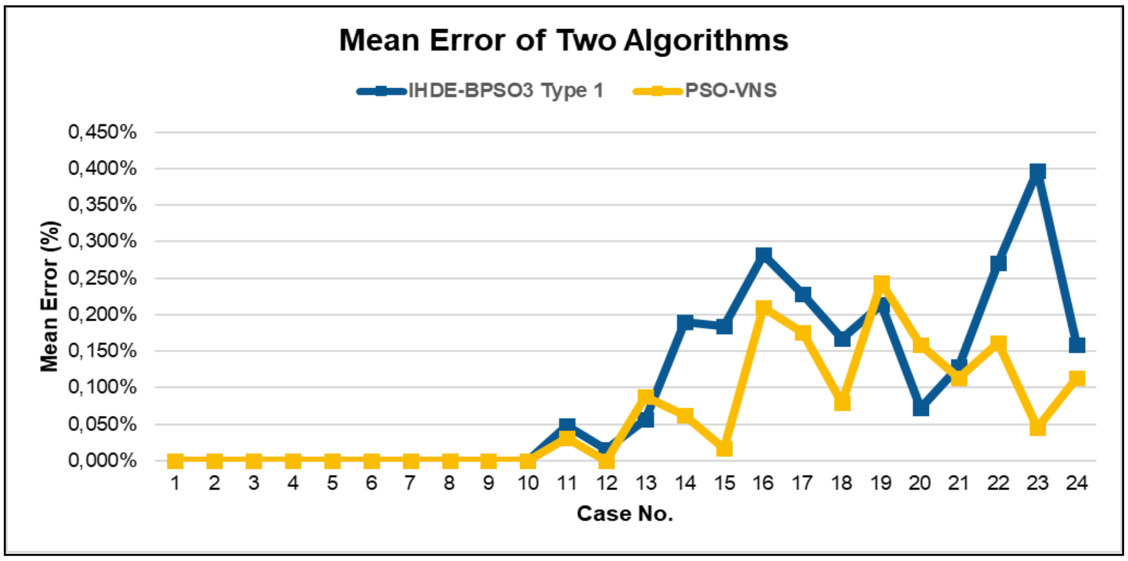

Figure 4.8 Line Chart of Testing Mean Error for Two Algorithms

Figure 4.8 shows the performance of two algorithms. For small problems, both IHDE-BPSO3 type 1 and PSO-VNS can achieve the optimal solution with a mean error value of 0.000%. However, as the number of orders and machines increases, the mean error starts to rise. This is because the algorithms are unable to find the optimal solution due to the computing time for each iteration increases as the problem size grows.

When the mean errors are totaled, IHDE-BPSO3 type 1 yields a larger total mean error with a value of 2.413% compared to PSO-VNS (1.495%). Thus, it can be concluded that PSO-VNS slightly outperforms IHDE-BPSO3 type 1. This is make sense since the IHDE-BPSO3 takes more time to compute. The inclusion of mutation, crossover, three times of local search and selection from three different populations in one iteration increases the time of each iteration.

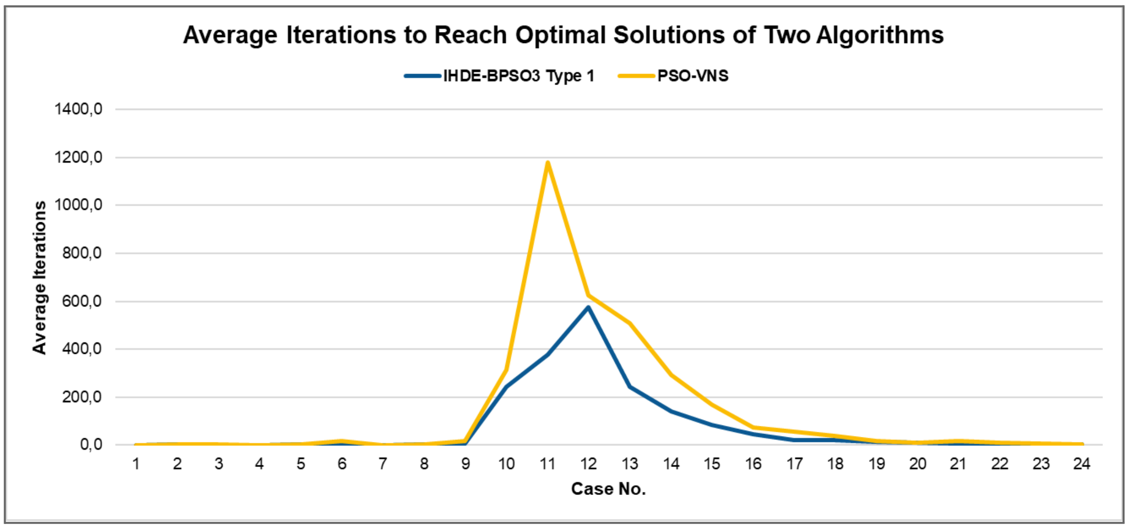

Figure 4.9 Line Chart of Average Iterations to Reach Optimal Solutions for Two Algorithms

In Figure 4.9, it is shown that IHDE-BPSO3 requires fewer iterations to reach its optimal solution compared to PSO-VNS. However, when testing IHDE-BPSO3 with a problem-solving time of 60 seconds for mid to large-sized problems, the algorithm resulted in fewer iterations. Hence, the algorithm is unable to reach more iterations to converge towards a better solution. Nevertheless, the largest gap between IHDE-BPSO3 solutions in large-sized problems was only 0.398%. For a newly introduced algorithm, this difference wasn't substantial. By integrating mutation and crossover operations for generating new populations to increase the diversity, employing selection to obtain the best popoulation for next generations, and incorporating the memory concept from particle swarm optimization, IHDE-BPSO3 demonstrates improved effectiveness in global exploration as it undergoes continuous development.