A Supplementary File for "Reviewing and Benchmarking Parameter Control Methods in Differential Evolution"

Ryoji Tanabe, *Member, IEEE,* and Alex Fukunaga

***Abstract*—This is a supplementary file for "Reviewing and Benchmarking Parameter Control Methods in Differential Evolution".**

S.1. Details of PCM-YADE

This section describes PCM-YADE [1]. At the beginning of each iteration t, an optimization state indicator I_t of the population $\boldsymbol{P}^t$ is calculated. Then, the current search state S_t is determined with the probability $\overline{I}_t \in [0,1]$ (a normalized value of I_t), and meta-parameters F^{pop}_{t+1} and C^{pop}_{t+1} for adaptation of values of F and C are updated based on S_t. Finally, for $i \in \{1, ..., N\}$, $F_{i,t}$ and $C_{i,t}$ for each individual $\boldsymbol{x}_{i,t}$ are assigned based on F^{pop}_{t+1} and C^{pop}_{t+1}, respectively.

For each iteration t, individuals are sorted based on their objective function values in descending order. The distance between each individual and the best individual $\boldsymbol{x}^{\mathrm{best},t}$ having the lowest objective function value in $\boldsymbol{P}^t$ is calculated. Individuals in $\boldsymbol{P}^t$ are also sorted based on their distance values in ascending order. The ranks of $\boldsymbol{x}_{i,t}$ according to the two sorting procedures are denoted as $f_{i,t}$ and $d_{i,t}$, respectively. The optimization state indicator I_t of $\boldsymbol{P}^t$ is given as follows:

$$I_t = \sum_{i=1}^{N} |f_{i,t} - d_{i,t}|. \tag{S.1}$$

I_t is further normalized into the range $[0,1]$ as follows:

$$\overline{I}_t = \frac{I_t - I^{\min}}{I^{\max} - I^{\min}}, \tag{S.2}$$

where $I^{\min} = 0$. When the population size N is the even number, $I^{\max} = N^2/2$. Otherwise, $I^{\max} = (N+1)(N-1)/2$.

The current search state S_t is determined with the probability $\overline{I}_t$:

$$S_t = \begin{cases} S^{\mathrm{exploration}} & \text{if } \mathrm{randu}[0,1] < \overline{I}_t \\ S^{\mathrm{exploitation}} & \text{otherwise} \end{cases}. \tag{S.3}$$

Then, the meta-parameters F^{pop}_{t+1} and C^{pop}_{t+1} are updated based on the current search situation S_t as follows:

$$F^{\mathrm{pop}}_{t+1} = \begin{cases} F^{\mathrm{pop}}_t + c_F \Delta F_{p,t} & \text{if } S_t = S^{\mathrm{exploration}} \\ F^{\mathrm{pop}}_t - c_F \Delta F_{p,t} & \text{otherwise} \end{cases}, \tag{S.4}$$

$$C^{\mathrm{pop}}_{t+1} = \begin{cases} C^{\mathrm{pop}}_t - c_C \Delta C_{p,t} & \text{if } S_t = S^{\mathrm{exploration}} \\ C^{\mathrm{pop}}_t + c_C \Delta C_{p,t} & \text{otherwise} \end{cases}, \tag{S.5}$$

where the two parameters c_F and c_C control the step-size for parameter adaptation and are usually set to 0.1 and 0.05, respectively. $\Delta F^{\mathrm{pop}}_t$ and $\Delta C^{\mathrm{pop}}_t$ are given as follows:

$$\Delta F^{\mathrm{pop}}_t, \Delta C^{\mathrm{pop}}_t = \begin{cases} \frac{I - I_{\min}}{I_{\max} - I_{\min}} & \text{if } S_t = S^{\mathrm{exploration}} \\ \frac{I_{\max} - I}{I_{\max} - I_{\min}} & \text{otherwise} \end{cases}. \tag{S.6}$$

Finally, $F_{i,t}$ and $C_{i,t}$ for each individual $\boldsymbol{x}^{i,t}$ are generated as follows:

$$F_{i,t+1} = \begin{cases} F_{p,t} + \Delta F_{i,t} & \text{if } f_{i,t} > \frac{N}{2} \;\&\; d_{i,t} > \frac{N}{2} \\ F_{p,t} - \Delta F_{i,t} & \text{if } f_{i,t} < \frac{N}{2} \;\&\; d_{i,t} < \frac{N}{2} \\ F_{p,t} & \text{otherwise} \end{cases}, \tag{S.7}$$

$$C_{i,t+1} = \begin{cases} C_{p,t} - \Delta C_{i,t} & \text{if } f_{i,t} > \frac{N}{2} \;\&\; d_{i,t} > \frac{N}{2} \\ C_{p,t} + \Delta C_{i,t} & \text{if } f_{i,t} < \frac{N}{2} \;\&\; d_{i,t} < \frac{N}{2} \\ C_{p,t} & \text{otherwise} \end{cases}, \tag{S.8}$$

where $\Delta F_{i,t}$ and $\Delta C_{i,t}$ are calculated as follows:

$$\Delta F_{i,t}, \Delta C_{i,t} = \begin{cases} \frac{(f_{i,t} + d_{i,t}) - N}{2N} & \text{if } f_{i,t} > \frac{N}{2} \;\&\; d_{i,t} > \frac{N}{2} \\ \frac{N - (f_{i,t} + d_{i,t})}{2N} & \text{if } f_{i,t} < \frac{N}{2} \;\&\; d_{i,t} < \frac{N}{2} \end{cases}. \tag{S.9}$$

S.2. The original description of PCM-SLADE

According to the original description in [2], PCM-SLADE does not use the arithmetic mean for the update of μ_F and μ_C. It is described in [2] that μ_F and μ_C in PCM-SLADE are updated as follows:

$$\mu_F = (1-c)\,\mu_F + c\,\frac{\sum_{s \in \boldsymbol{S}^F} s}{N}, \tag{S.10}$$

$$\mu_C = (1-c)\,\mu_C + c\,\frac{\sum_{s \in \boldsymbol{S}^C} s}{N}, \tag{S.11}$$

where $\boldsymbol{S}^F$ and $\boldsymbol{S}^C$ are sets of successful F and C parameters for each iteration, respectively. Also, c is a learning rate, and N denotes the population size. In (S.10) and (S.11), the sum of elements in $\boldsymbol{S}^F$ and $\boldsymbol{S}^C$ is divided by N.

R. Tanabe is with the Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems, Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. e-mail: (rt.ryoji.tanabe@gmail.com).

A. Fukunaga is with the Graduate School of Arts and Sciences, The University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo, Japan. e-mail: (fukunaga@idea.c.u-tokyo.ac.jp).

However, our preliminary experiments show that the performance of DE algorithms with PCM-SLADE using (S.10) and (S.11) is poor on most nonseparable functions. Since the size of $\boldsymbol{S}^F$ and $\boldsymbol{S}^C$ is usually significantly lower than N, the second terms $c(\sum_{s\in\boldsymbol{S}^F} s)/N$ and $c(\sum_{s\in\boldsymbol{S}^C} s)/N$ in (S.10) and (S.11) are very close to zero. For this reason, μ_F and μ_C converge to zero by repeatedly performing the update with (S.10) and (S.11). This undesirable mechanism causes the poor performance of DE algorithms with PCM-SLADE. Based on these results, we believe that the original description of PCM-SLADE is incorrect, and that the authors of [2] intended to describe the arithmetic mean as the following (S.12) and (S.13), instead of the inscrutable operations in (S.10) and (S.11).

$$\mu_F = (1-c)\,\mu_F + c\,\frac{\sum_{s\in\boldsymbol{S}^F} s}{|\boldsymbol{S}^F|}, \tag{S.12}$$

$$\mu_C = (1-c)\,\mu_C + c\,\frac{\sum_{s\in\boldsymbol{S}^C} s}{|\boldsymbol{S}^C|}. \tag{S.13}$$

S.3. The restart strategy of [3]

As described in the main paper, the restart strategy of [3] was incorporated into all methods, except for PCM-SinDE and PCM-DETVSF. The search process of DE is restarted when any of the following three criteria are met:

(1) Solution vector criterion: Restart if $\Delta x^t_j < \varepsilon_x(\max_{i\in\{1,...,N\}}\{|x^{i,t}_j|\})$ at least one index $j \in \{1,...,D\}$, where $\Delta x^t_j = \max_{i\in\{1,...,N\}}\{x^{i,t}_j\} - \min_{i\in\{1,...,N\}}\{x^{i,t}_j\}$ and $\boldsymbol{x}^{i,t} \in \boldsymbol{P}^t$. In this paper, $\varepsilon_x = 10^{-12}$.

(2) Objective value criterion: Restart if $\Delta f^t < \varepsilon_f \max_{i\in\{1,...,N\}}\{|f(\boldsymbol{x}^{i,t})|\}$, where $\Delta f^t = \max_{i\in\{1,...,N\}}\{f(\boldsymbol{x}^{i,t})\} - \min_{i\in\{1,...,N\}}\{f(\boldsymbol{x}^{i,t})\}$. In this paper, $\varepsilon_f = 10^{-12}$.

(3) Best-so-far solution criterion: Restart if the best solution found after the last restart has not been updated for $\mathrm{FEvals}^{\mathrm{stop}}$ function evaluations. In this paper, $\mathrm{FEvals}^{\mathrm{stop}} = 500 \times D$.

S.4. Empirical Cumulative Distribution Function (ECDF)

Below, we explain how to interpret the ECDF figures. Each BBOB function includes 15 function instances with different problem parameters (e.g., the position of the optimal solution and its objective value). Thus, the BBOB functions consist of 360 ($= 24 \times 15$) distinct function instances for each dimensionality. Let $f^{\mathrm{target}} = f^{\mathrm{opt}} - \Delta f$ be the target objective value, where f^{opt} denotes the objective value of the optimal solution of a given instance, and Δf is a target precision. In the COCO software, 51 Δf values are uniformly selected from the range $[10^{-8}, 10^{2}]$ on a log-scale as follows: $\{10^2, 10^{1.8}, ..., 10^{-8}\}$. Thus, 51 f_{target} values are given for each function instance, and a total of $18\,360$ ($= 360 \times 51$) f_{target} values are defined for all the 360 instances.

In ECDF figures, the vertical axis "proportion of function + target pairs" indicates the proportion of target objective values which a given algorithm can reach within specified evaluations. For example, in Fig.1(b) in the main paper, PCM-CoDE reaches about 20 percent of the $18\,360$ f_{target} values within $1\,000 \times D$ evaluations. If a given algorithm finds the optimal solutions on all 360 instances, the vertical value becomes 1.

S.5. Average Performance Score (APS)

We used the average performance score (APS) [4] in order to aggregate experimental results on the 24 BBOB functions. Below, we describe the APS. Suppose that the performance of n algorithms $A_1, ..., A_n$ is compared on a set of K test functions based on the error value $|f(\boldsymbol{x}^{\mathrm{bsf}} - f(\boldsymbol{x}^*)|$ achieved by multiple independent runs[1], where $\boldsymbol{x}^{\mathrm{bsf}}$ is the best-so-far solution found during the search process, and $\boldsymbol{x}^*$ is the optimal solution of the target function.

For each $i \in \{1,...,n\}$ and $j \in \{1,...,n\}\backslash\{i\}$, if A_j outperforms A_i on a k-th function f_k ($k \in \{1,...,K\}$) with a statistical significance, $\delta_{i,j} = 1$. Otherwise, $\delta_{i,j} = 0$. The Wilcoxon rank-sum test with $p < 0.05$ was used for the statistical test in our study. Then, a performance score of A_i on f_k is given as follows:

$$\mathrm{P}_k(A_i) = \sum_{j\in\{1,...,n\}\backslash\{i\}} \delta_{i,j}, \tag{S.14}$$

where $\mathrm{P}_k(A_i)$ represents the number of algorithms outperforming A_i on f_k according to the error value. If $\mathrm{P}_k(A_i) = 0$, all of the other $n-1$ algorithms do not perform significantly better than A_i on f_k.

The APS value of A_i is the average of the performance score values of A_i for all the K test functions as follows:

$$\mathrm{APS}(A_i) = \frac{1}{K}\sum_{k\in\{1,...,K\}} \mathrm{P}_k(A_i). \tag{S.15}$$

The APS value of A_i represents how good relatively the performance of A_i is among the n algorithms on average over all the K test functions. A small APS value indicates that the performance of a corresponding algorithm is better than other $n-1$ algorithms.

S.6. Results for each DE operator with the default parameter settings

In this section, we describe results for each DE operator in detail as follows:

- **rand/1/bin** (Fig. S.1): While PCM-SHADE performs well for $D \leq 20$, PCM-jDE and PCM-FDSADE show good performance for higher-dimensional problems.
- **rand/2/bin** (Fig. S.2): For all D, PCM-IDE and PCM-SDE outperform the other PCMs at around $10^3 \times D$ function evaluations. Beyond $10^3 \times D$ evaluations, PCM-SHADE is the best PCM for $D \in \{3, 5, 10, 20\}$, and PCM-SLADE and PCM-ISADE perform well for $D = 40$.
- **best/1/bin** (Fig. S.3): PCM-DERSF, PCM-CoBiDE, and PCM-cDE have high performance for $D = 3$, $D \in \{5, 10\}$, and $D \in \{20, 40\}$, respectively.

[1] Note that $n = 25$ and $K = 24$ in our study.

• **best/2/bin** (Fig. S.4): PCM-CoBiDE and PCM-SWDE show good performance for $D \in \{5, 10\}$. PCM-SHADE also performs best for $D = 40$.

• **current-to-rand/1/bin** (Fig. S.5): Although PCM-SHADE and PCM-CoBiDE outperform the other methods at the end of the search for $D = 10$ and 20 respectively, PCM-cDE is the best PCM for almost all D.

• **current-to-best/1/bin** (Fig. S.6) and **current-to-$p$best/1/bin** (Fig. S.7): Results of the 25 methods using the current-to-best/1/bin and current-to-$p$best/1/bin operators are similar to each other. For $D \geq 5$, PCM-cDE, PCM-SHADE, and PCM-CoBiDE outperform other methods. PCM-IDE has high performance for $D = 5$. The performance of PCM-JADE is relatively better than other methods for $D \geq 20$.

• **rand-to-$p$best/1/bin** (Fig. S.8): Three PCMs (PCM-cDE, PCM-SHADE, and PCM-CoBiDE) perform well for $D \geq 5$.

• **rand/1/sec** (Fig. S.9): PCM-CoBiDE, PCM-SHADE, PCM-RDE, and PCM-cDE show good performance for $D \in \{5, 10\}$. PCM-cDE and PCM-SWDE perform well for $D = 20$. The performance of PCM-IDE improves gradually with increasing D. For $D = 40$, PCM-IDE has the best anytime performance, followed by PCM-SinDE and the three PCM-jDE variants.

• **rand/2/sec** (Fig. S.10): PCM-SHADE has the best performance for $D \in \{5, 10\}$. For $D \geq 20$, PCM-IDE is the best PCM, followed by PCM-SinDE and PCM-ISADE.

• **best/1/sec** (Fig. S.11): In addition to PCM-CoBiDE and PCM-SHADE, PCM-DERSF, the simplest among the 24 PCMs, performs relatively well for $D \in \{3, 5, 10, 20\}$. For $D \geq 20$, PCM-CoBiDE outperforms other methods.

• **best/2/sec** (Fig. S.12): The baseline static DE with $F = 0.5$ and $C = 0.9$ is the best method for $D \leq 5$ and still competitive with the 24 PCMs for $D \in \{10, 20\}$. For $D = 10$, PCM-CoBiDE outperforms other methods at around the end of the search. PCM-cDE and PCM-CoBiDE perform well for $D = 20$. PCM-FDSADE and PCM-CoBiDE show high performance for $D = 40$.

• **current-to-rand/1/sec** (Fig. S.13): PCM-cDE outperforms other methods for $D \in \{2, 3, 20, 40\}$. PCM-SHADE and PCM-CoBiDE show good performance for $D = 10$ and $D = 20$, respectively.

• **current-to-best/1/sec** (Fig. S.14): PCM-cDE, PCM-SHADE, and PCM-CoBiDE outperform other methods for most values of D. In addition to the three PCMs, PCM-DEDPS and PCM-IDE perform relatively well for $D \in \{2, 3\}$ and $D \in \{5, 10\}$, respectively.

• **current-to-$p$best/1/sec** (Fig. S.15): PCM-CoBiDE, PCM-cDE, and PCM-DERSF have high performance for $D = 5$. For $D \geq 10$, PCM-cDE is the best PCM.

• **rand-to-$p$best/1/sec** (Fig. S.16): For $D \in \{5, 10\}$, although PCM-CoBiDE performs best at around the end of the search, the baseline DE with the fixed parameter setting and PCM-cDE have good performance before that. For $D \geq 20$, PCM-cDE outperforms other methods.

Algorithm S.1: The basic DE algorithm

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using a mutation strategy with F (see Table S.1);
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using a crossover method with C (see Algorithms S.2, S.3, and S.4);
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

TABLE S.1: Eight representative mutation strategies for DE. The indices r_1, r_2, r_3, r_4, r_5 in Table S.1 are randomly selected from $\{1, ..., N\} \backslash \{i\}$ such that they differ from each other. The individual $\boldsymbol{x}^{\mathrm{best},t}$ is the best individual in the population $\boldsymbol{P}^t$. For each individual, the individual $\boldsymbol{x}^{p\mathrm{best},i,t}$ is randomly selected from the top $\max(\lfloor N\,p \rfloor, 2)$ members in $\boldsymbol{P}^t$, where $p \in [0, 1]$ controls the greediness of the current-to-$p$best/1 and rand-to-$p$best/1 strategies. A small p value encourages exploitation of the search. The individuals $\tilde{\boldsymbol{x}}^{r_2,t}$ and $\tilde{\boldsymbol{x}}^{r_3,t}$ in the current-to-$p$best/1 and rand-to-$p$best/1 strategies are randomly selected from a union of $\boldsymbol{P}^t$ and an external archive $\boldsymbol{A}^t$, where inferior parent individuals are preserved in $\boldsymbol{A}^t$.

Strategies	Definitions
rand/1	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{r_1,t} + F_{i,t}\,(\boldsymbol{x}^{r_2,t} - \boldsymbol{x}^{r_3,t})$
rand/2	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{r_1,t} + F_{i,t}\,(\boldsymbol{x}^{r_2,t} - \boldsymbol{x}^{r_3,t}) + F_{i,t}\,(\boldsymbol{x}^{r_4,t} - \boldsymbol{x}^{r_5,t})$
best/1	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{\mathrm{best},t} + F_{i,t}\,(\boldsymbol{x}^{r_1,t} - \boldsymbol{x}^{r_2,t})$
best/2	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{\mathrm{best},t} + F_{i,t}\,(\boldsymbol{x}^{r_1,t} - \boldsymbol{x}^{r_2,t}) + F_{i,t}\,(\boldsymbol{x}^{r_3,t} - \boldsymbol{x}^{r_4,t})$
current-to-rand/1	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{i,t} + F_{i,t}\,(\boldsymbol{x}^{r_1,t} - \boldsymbol{x}^{i,t}) + F_{i,t}\,(\boldsymbol{x}^{r_2,t} - \boldsymbol{x}^{r_3,t})$
current-to-best/1	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{i,t} + F_{i,t}\,(\boldsymbol{x}^{\mathrm{best},t} - \boldsymbol{x}^{i,t}) + F_{i,t}\,(\boldsymbol{x}^{r_1,t} - \boldsymbol{x}^{r_2,t})$
current-to-$p$best/1	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{i,t} + F_{i,t}\,(\boldsymbol{x}^{p\mathrm{best},t} - \boldsymbol{x}^{i,t}) + F_{i,t}\,(\boldsymbol{x}^{r_1,t} - \tilde{\boldsymbol{x}}^{r_2,t})$
rand-to-$p$best/1	$\boldsymbol{v}^{i,t} = \boldsymbol{x}^{r_1,t} + F_{i,t}\,(\boldsymbol{x}^{p\mathrm{best},t} - \boldsymbol{x}^{r_1,t}) + F_{i,t}\,(\boldsymbol{x}^{r_2,t} - \tilde{\boldsymbol{x}}^{r_3,t})$

Algorithm S.2: Binomial crossover (bin)

Randomly select j_{rand} from $\{1, ..., D\}$;
for $j \in \{1, ..., D\}$ **do**
    if $\mathrm{randu}[0, 1] < C_{i,t}$ *or* $j == j_{\mathrm{rand}}$ **then** $u_j^{i,t} \leftarrow v_j^{i,t}$;
    else $u_j^{i,t} \leftarrow x_j^{i,t}$;

Algorithm S.3: Exponential crossover (exp)

Randomly select j from $\{1, ..., D\}$;
$\boldsymbol{u}^{i,t} \leftarrow \boldsymbol{x}^{i,t}$, $k \leftarrow 1$;
repeat
    $u_j^{i,t} \leftarrow v_j^{i,t}$, $j \leftarrow 1 + j$ modulo D, $k \leftarrow k + 1$;
until $\mathrm{randu}[0, 1] < C_{i,t}$ *and* $k < D$;

Algorithm S.4: Shuffled exponential crossover (sec)

Generate $\boldsymbol{s}^{i,t} = (s_{1,i,t}, ..., s_{D,i,t})^{\mathrm{T}}$ whose elements are randomly shuffled variable indices $\{1, ..., D\}$;
$\boldsymbol{u}^{i,t} \leftarrow \boldsymbol{x}^{i,t}$, $k \leftarrow 1$;
repeat
    $j \leftarrow s_k^{i,t}$, $u_j^{i,t} \leftarrow v_j^{i,t}$, $k \leftarrow k + 1$;
until $\mathrm{randu}[0, 1] < C_{i,t}$ *and* $k < D$;

Algorithm S.5: A DE algorithm with PCM-DERSF

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
    for $i \in \{1, ..., N\}$ **do**
        $F_{i,t} \leftarrow \mathrm{randu}[0.5, 1]$;
        Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
        Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with C;
    for $i \in \{1, ..., N\}$ **do**
        if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
        else
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
    $t \leftarrow t + 1$;

Algorithm S.6: A DE algorithm with PCM-DETVSF

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
    $F_t \leftarrow (F^{\max} - F^{\min})\left(\frac{t^{\max} - t}{t^{\max}}\right) + F^{\min}$;
    for $i \in \{1, ..., N\}$ **do**
        Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with F_t;
        Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with C;
    for $i \in \{1, ..., N\}$ **do**
        if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
        else
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
    $t \leftarrow t + 1$;

Algorithm S.7: A DE algorithm with PCM-SinDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
    Generate F_t according to equation (1) in the main paper;
    Generate C_t according to equation (2) in the main paper;
    for $i \in \{1, ..., N\}$ **do**
        Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with F_t;
        Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with C_t;
    for $i \in \{1, ..., N\}$ **do**
        if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
        else
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
    $t \leftarrow t + 1$;

Algorithm S.8: A DE algorithm with PCM-ZMDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
    for $i \in \{1, ..., N\}$ **do**
        $F_{i,t} \leftarrow \mathrm{randn}(0.75, 0.1)$;
        $C_{i,t} \leftarrow \mathrm{randu}[0.8, 1]$;
        Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
        Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
    for $i \in \{1, ..., N\}$ **do**
        if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
        else
            $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
    $t \leftarrow t + 1$;

Algorithm S.9: A DE algorithm with PCM-CoDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\boldsymbol{q}^1 = (1, 0.1)$, $\boldsymbol{q}^2 = (1, 0.9)$, and $\boldsymbol{q}^3 = (0.8, 0.2)$, where $\boldsymbol{q} = (F, C)$;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    Randomly select an index r from $\{1, 2, 3\}$;
    $(F_{i,t}, C_{i,t}) \leftarrow q^r$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.10: A DE algorithm with PCM-SWDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    if $\mathrm{randu}[0, 1] \leq 0.5$ **then** $F_{i,t} \leftarrow 0.5$;
    else $F_{i,t} \leftarrow 2$;
    if $\mathrm{randu}[0, 1] \leq 0.5$ **then** $C_{i,t} \leftarrow 0$;
    else $C_{i,t} \leftarrow 1$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.11: A DE algorithm with PCM-DEPD

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
  Calculate F_t according to equation (3) in the main paper;
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with F_t;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with C;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.12: A DE algorithm with PCM-jDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
for $i \in \{1, ..., N\}$ **do** $F_{i,t} \leftarrow 0.5$, $C_{i,t} \leftarrow 0.9$;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    if $\mathrm{randu}[0, 1] \leq \tau_F$ **then** $F^{\mathrm{trial}}_{i,t} \leftarrow \mathrm{randu}[0.1, 1]$;
    else $F^{\mathrm{trial}}_{i,t} \leftarrow F_{i,t}$;
    if $\mathrm{randu}[0, 1] \leq \tau_C$ **then** $C^{\mathrm{trial}}_{i,t} \leftarrow \mathrm{randu}[0, 1]$;
    else $C^{\mathrm{trial}}_{i,t} \leftarrow C_{i,t}$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F^{\mathrm{trial}}_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C^{\mathrm{trial}}_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $F_{i,t+1} \leftarrow F^{\mathrm{trial}}_{i,t}$, $C_{i,t+1} \leftarrow C^{\mathrm{trial}}_{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$, $F_{i,t+1} \leftarrow F_{i,t}$, $C_{i,t+1} \leftarrow C_{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.13: A DE algorithm with PCM-FDSADE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
for $i \in \{1, ..., N\}$ **do** $F_{i,t} \leftarrow 0.5$, $C_{i,t} \leftarrow 0.9$;
while *The termination criteria are not met* **do**
  $\phi_t = f^{\mathrm{std}}_t / (f^{\mathrm{max}}_t - f^{\mathrm{min}}_t)$;
  for $i \in \{1, ..., N\}$ **do**
    if $\mathrm{randu}[0, 1] \leq K(1 - \phi_t)$ **then** $F^{\mathrm{trial}}_{i,t} \leftarrow \mathrm{randu}[0.1, 1]$;
    else $F^{\mathrm{trial}}_{i,t} \leftarrow F_{i,t}$;
    if $\mathrm{randu}[0, 1] \leq K(1 - \phi_t)$ **then** $C^{\mathrm{trial}}_{i,t} \leftarrow \mathrm{randu}[0, 1]$;
    else $C^{\mathrm{trial}}_{i,t} \leftarrow C_{i,t}$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F^{\mathrm{trial}}_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C^{\mathrm{trial}}_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $F_{i,t+1} \leftarrow F^{\mathrm{trial}}_{i,t}$, $C_{i,t+1} \leftarrow C^{\mathrm{trial}}_{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$, $F_{i,t+1} \leftarrow F_{i,t}$, $C_{i,t+1} \leftarrow C_{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.14: A DE algorithm with PCM-ISADE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
for $i \in \{1, ..., N\}$ **do** $F_{i,t} \leftarrow \mathrm{randu}[0,1]$, $C_{i,t} \leftarrow \mathrm{randu}[0,1]$;
while *The termination criteria are not met* **do**
  $\alpha = (f(\boldsymbol{x}^{i,t}) - f_t^{\min})/(f_t^{\mathrm{avg}} - f_t^{\min})$;
  for $i \in \{1, ..., N\}$ **do**
    if $\mathrm{randu}[0,1] < \tau_F$ & $f(\boldsymbol{x}^{i,t}) < f_t^{\mathrm{avg}}$ **then**
      $F_{i,t}^{\mathrm{trial}} \leftarrow \alpha(F_{i,t} - 0.1) + 0.1$;
    else if $\mathrm{randu}[0,1] < \tau_F$ & $f(\boldsymbol{x}^{i,t}) \geq f_t^{\mathrm{avg}}$ **then**
      $F_{i,t}^{\mathrm{trial}} \leftarrow \mathrm{randu}[0.1,1]$;
    else
      $F_{i,t}^{\mathrm{trial}} \leftarrow F_{i,t}$;
    if $\mathrm{randu}[0,1] < \tau_C$ & $f(\boldsymbol{x}^{i,t}) < f_t^{\mathrm{avg}}$ **then**
      $C_{i,t}^{\mathrm{trial}} \leftarrow \alpha\, C_{i,t}$;
    else if $\mathrm{randu}[0,1] < \tau_C$ & $f(\boldsymbol{x}^{i,t}) \geq f_t^{\mathrm{avg}}$ **then**
      $F_{i,t}^{\mathrm{trial}} \leftarrow \mathrm{randu}[0,1]$;
    else
      $C_{i,t}^{\mathrm{trial}} \leftarrow C_{i,t}$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}^{\mathrm{trial}}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}^{\mathrm{trial}}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $F_{i,t+1} \leftarrow F_{i,t}^{\mathrm{trial}}$, $C_{i,t+1} \leftarrow C_{i,t}^{\mathrm{trial}}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$, $F_{i,t+1} \leftarrow F_{i,t}$, $C_{i,t+1} \leftarrow C_{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.15: A DE algorithm with PCM-cDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\boldsymbol{q}^1 = (0.5, 0), \boldsymbol{q}^2 = (0.5, 0.5), \boldsymbol{q}^3 = (0.5, 1), \boldsymbol{q}^4 = (0.8, 0), \boldsymbol{q}^5 = (0.8, 0.5), \boldsymbol{q}^6 = (0.8, 1), \boldsymbol{q}^7 = (1, 0), \boldsymbol{q}^8 = (1, 0.5), \boldsymbol{q}^9 = (1, 1)$;
for $k \in \{1, ..., 9\}$ **do** $n_k^{\mathrm{succ}} \leftarrow 0$;
while *The termination criteria are not met* **do**
  Calculate $s_{k,t}$ for each $k \in \{1, ..., 9\}$ by equation (10) in the main paper;
  if $\exists s_{k,t} \leq \delta$ **then**
    For each k, $n_k^{\mathrm{succ}} \leftarrow 0$;
    Recalculate $s_{k,t}$ for each k by equation (10) in the main paper;
  for $i \in \{1, ..., N\}$ **do**
    Select a pair of parameters $\boldsymbol{q}$ from $\boldsymbol{q}^1, ..., \boldsymbol{q}^9$ with the probability $s_{k,t}$ and $(F_{i,t}, C_{i,t}) \leftarrow \boldsymbol{q}$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
      if $\boldsymbol{q}^k$ *is used for generating* $\boldsymbol{u}^{i,t}$ **then** $n_k^{\mathrm{succ}} \leftarrow n_k^{\mathrm{succ}} + 1$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.16: A DE algorithm with PCM-SaDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\mu^C \leftarrow 0.5$, $\boldsymbol{H}^C \leftarrow \emptyset$, $k \leftarrow 1$;
while *The termination criteria are not met* **do**
  if $t \geq t^{\text{learn}}$ **then**
    Set μ^C to the median value of elements in $\boldsymbol{H}^C$;
  for $i \in \{1, ..., N\}$ **do**
    $F_{i,t} \leftarrow \text{randn}(0.5, 0.3)$;
    $C_{i,t} \leftarrow \text{randn}(\mu^C, 0.1)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  $\boldsymbol{H}^{C,k} \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{H}^{C,k} \leftarrow \boldsymbol{H}^{C,k} \cup \{C_{i,t}\}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $k \leftarrow (k \bmod t^{\text{learn}}) + 1$;
  $t \leftarrow t + 1$;

Algorithm S.17: A DE algorithm with PCM-SaNSDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\mu_C \leftarrow 0.5$, $\boldsymbol{H}^C \leftarrow \emptyset$, $k \leftarrow 1$, $p \leftarrow 0.5$;
n^{total1}, n^{total2}, n^{succ1}, $n^{\text{succ2}} \leftarrow 0$;
while *The termination criteria are not met* **do**
  if $t \geq t^{\text{learn}}$ **then**
    Update μ_C based on equation (13) in the main paper;
  if $t \bmod t^{\text{learn}} = 0$ **then**
    Update p according to equation (12) in the main paper;
    n^{total1}, n^{total2}, n^{succ1}, $n^{\text{succ2}} \leftarrow 0$;
  for $i \in \{1, ..., N\}$ **do**
    Generate $F_{i,t}$ with p according to equation (11) in the main paper;
    $C_{i,t} \leftarrow \text{randn}(\mu^C, 0.1)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  $\boldsymbol{H}^{C,k} \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    If $F_{i,t}$ was generated according to the Normal distribution, $n^{\text{total1}} \leftarrow n^{\text{total1}} + 1$. Otherwise, $n^{\text{total2}} \leftarrow n^{\text{total2}} + 1$;
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
      $\boldsymbol{H}^{C,k} \leftarrow \boldsymbol{H}^{C,k} \cup \{C_{i,t}\}$;
      If $F_{i,t}$ was generated according to Normal distribution, $n^{\text{succ1}} \leftarrow n^{\text{succ1}} + 1$. Otherwise, $n^{\text{succ2}} \leftarrow n^{\text{succ2}} + 1$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $k \leftarrow (k \bmod t^{\text{learn}}) + 1$;
  $t \leftarrow t + 1$;

Algorithm S.18: A DE algorithm with PCM-JADE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\mu_F, \mu_C \leftarrow 0.5$;
while *The termination criteria are not met* **do**
  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    $F_{i,t} \leftarrow \mathrm{randc}(\mu_F, 0.1)$;
    $C_{i,t} \leftarrow \mathrm{randn}(\mu_C, 0.1)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{S}^F \leftarrow \boldsymbol{S}^F \cup \{F_{i,t}\}$, $\boldsymbol{S}^C \leftarrow \boldsymbol{S}^C \cup \{C_{i,t}\}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  if $\boldsymbol{S}^F, \boldsymbol{S}^C \neq \emptyset$ **then**
    $\mu_F \leftarrow (1-c)\,\mu_F + c\,\mathrm{mean}_L(\boldsymbol{S}^F)$;
    $\mu_C \leftarrow (1-c)\,\mu_C + c\,\mathrm{mean}_A(\boldsymbol{S}^C)$;
  $t \leftarrow t+1$;

Algorithm S.19: A DE algorithm with PCM-IMDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\mu_F, \mu_C \leftarrow 0.5$;
while *The termination criteria are not met* **do**
  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    $F_{i,t} \leftarrow \mathrm{randc}(\mu_F, 0.1)$;
    $C_{i,t} \leftarrow \mathrm{randn}(\mu_C, 0.1)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{S}^F \leftarrow \boldsymbol{S}^F \cup \{F_{i,t}\}$, $\boldsymbol{S}^C \leftarrow \boldsymbol{S}^C \cup \{C_{i,t}\}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  if $\boldsymbol{S}^F, \boldsymbol{S}^C \neq \emptyset$ **then**
    $c_F \leftarrow \mathrm{randu}[0, 0.2]$, $c_C \leftarrow \mathrm{randu}[0.0, 0.1]$;
    $\mu_F \leftarrow (1-c_F)\,\mu_F + c_F\,\mathrm{mean}_P(\boldsymbol{S}^F)$;
    $\mu_C \leftarrow (1-c_C)\,\mu_C + c_C\,\mathrm{mean}_P(\boldsymbol{S}^C)$;
  $t \leftarrow t+1$;

Algorithm S.20: A DE algorithm with PCM-SHADE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
Set all values in $\boldsymbol{M}^F = (M_1^F, ..., M_H^F)$, $\boldsymbol{M}^C = (M_1^C, ..., M_H^C)$ to 0.5;
$k \leftarrow 1$;
while *The termination criteria are not met* **do**
  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    Select the memory index $r_{i,t}$ from $\{1, ..., H\}$ randomly;
    $F_{i,t} \leftarrow \mathrm{randc}(M_{r_{i,t}}^F, 0.1)$;
    $C_{i,t} \leftarrow \mathrm{randn}(M_{r_{i,t}}^C, 0.1)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{S}^F \leftarrow \boldsymbol{S}^F \cup \{F_{i,t}\}$, $\boldsymbol{S}^C \leftarrow \boldsymbol{S}^C \cup \{C_{i,t}\}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  if $\boldsymbol{S}^F, \boldsymbol{S}^C \neq \emptyset$ **then**
    $M_k^F \leftarrow \mathrm{mean}_L(\boldsymbol{S}^F)$;
    $M_k^C \leftarrow \mathrm{mean}_L(\boldsymbol{S}^C)$;
    $k \leftarrow (k \,\mathrm{modulo}\, H) + 1$;
  $t \leftarrow t + 1$;

Algorithm S.21: A DE algorithm with PCM-SLADE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\mu_F, \mu_C \leftarrow 0.5$;
while *The termination criteria are not met* **do**
  $\boldsymbol{S}^F \leftarrow \emptyset$, $\boldsymbol{S}^C \leftarrow \emptyset$;
  for $i \in \{1, ..., N\}$ **do**
    $F_{i,t} \leftarrow \mathrm{randn}(\mu_F, 0.1)$;
    $C_{i,t} \leftarrow \mathrm{randc}(\mu_C, 0.1)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $\boldsymbol{S}^F \leftarrow \boldsymbol{S}^F \cup \{F_{i,t}\}$, $\boldsymbol{S}^C \leftarrow \boldsymbol{S}^C \cup \{C_{i,t}\}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  if $\boldsymbol{S}^F, \boldsymbol{S}^C \neq \emptyset$ **then**
    $\mu_F \leftarrow (1-c)\,\mu_F + c\,\mathrm{mean}_A(\boldsymbol{S}^F)$;
    $\mu_C \leftarrow (1-c)\,\mu_C + c\,\mathrm{mean}_A(\boldsymbol{S}^C)$;
  $t \leftarrow t + 1$;

Algorithm S.22: A DE algorithm with PCM-EPSDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\boldsymbol{F}^{\mathrm{pool}} = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$;
$\boldsymbol{C}^{\mathrm{pool}} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$;
For each individual $\boldsymbol{x}^{i,t}$, assign the $F_{i,t}$ and $C_{i,t}$ values randomly from each pool;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $F_{i,t+1} \leftarrow F_{i,t}, C_{i,t+1} \leftarrow C_{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
      Randomly reassign the $F_{i,t+1}$ and $C_{i,t+1}$ values from each pool;
  $t \leftarrow t + 1$;

Algorithm S.23: A DE algorithm with PCM-CoBiDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
for $i \in \{1, ..., N\}$ **do** Generate $F_{i,t}$ and $C_{i,t}$ by equations (14) and (15) in the main paper;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $F_{i,t+1} \leftarrow F_{i,t}, C_{i,t+1} \leftarrow C_{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
      Generate $F_{i,t+1}$ and $C_{i,t+1}$ by equations (14) and (15) in the main paper;
  $t \leftarrow t + 1$;

Algorithm S.24: A DE algorithm with PCM-DEDPS

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
$\boldsymbol{F}^{\mathrm{pool}} = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$, $\boldsymbol{C}^{\mathrm{pool}} = \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$;
$k \leftarrow 1$;
for $F \in \boldsymbol{F}^{\mathrm{pool}}$ **do**
  for $C \in \boldsymbol{C}^{\mathrm{pool}}$ **do**
    $\boldsymbol{q}^k \leftarrow (F, C)$, $n_k^{\mathrm{total}} \leftarrow 0$, $n_k^{\mathrm{succ}} \leftarrow 0$, $k \leftarrow k + 1$;
$m \leftarrow |\boldsymbol{F}^{\mathrm{pool}}| \times |\boldsymbol{C}^{\mathrm{pool}}|$;
while *The termination criteria are not met* **do**
  Randomly assign all possible pairs $\boldsymbol{q}^1, ..., \boldsymbol{q}^m$ to all individuals without replacement;
  if $N > m$ **then** the randomly selected pairs from $\boldsymbol{q}^1, ..., \boldsymbol{q}^m$ are assigned to $N - m$ remaining individuals;
  for $i \in \{1, ..., N\}$ **do**
    if $\boldsymbol{q}^k$ *is assigned to* $\boldsymbol{x}^{i,t}$ **then** $n_k^{\mathrm{total}} \leftarrow n_k^{\mathrm{total}} + 1$;
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
      if $\boldsymbol{q}^k$ *is used for generating* $\boldsymbol{u}^{i,t}$ **then** $n_k^{\mathrm{succ}} \leftarrow n_k^{\mathrm{succ}} + 1$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  if $t \in \{50, 100, 150, 200\}$ **then**
    for $k \in \{1, ..., m\}$ **do**
      $n_k^{\mathrm{score}} \leftarrow n_k^{\mathrm{succ}} / n_k^{\mathrm{total}}$;
    Sort the m parameter combinations based on their score values and remove their lower half;
    $m \leftarrow m/2$;
    for $k \in \{1, ..., m\}$ **do**
      $n_k^{\mathrm{total}} \leftarrow 0$, $n_k^{\mathrm{succ}} \leftarrow 0$;
  $t \leftarrow t + 1$;

Algorithm S.25: A DE algorithm with PCM-RDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
  Sort individuals in $\boldsymbol{P}^t$ based on their objective function values so that $f(\boldsymbol{x}^{1,t}) \leq ... \leq f(\boldsymbol{x}^{N,t})$;
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$ which is generated according to equation (16) based on the rank of a selected base vector;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$ which is generated according to equation (17) based on the rank of a selected base vector;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.26: A DE algorithm with PCM-IDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
  Sort individuals in $\boldsymbol{P}^t$ based on their objective function values so that $f(\boldsymbol{x}^{1,t}) \leq ... \leq f(\boldsymbol{x}^{N,t})$;
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$ which is generated based on the rank of a selected base vector;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$ which is generated based on the rank of $\boldsymbol{x}^{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.27: A DE algorithm with PCM-YADE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
while *The termination criteria are not met* **do**
  Assign $F_{i,t}$ and $C_{i,t}$ to each $\boldsymbol{x}^{i,t}$ according to the method described in Section S.1;
  for $i \in \{1, ..., N\}$ **do**
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

Algorithm S.28: A DE algorithm with PCM-SDE

$t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$ randomly;
for $i \in \{1, ..., N\}$ **do** $F_{i,t} \leftarrow \mathrm{randn}(0.5, 0.15)$;
while *The termination criteria are not met* **do**
  for $i \in \{1, ..., N\}$ **do**
    Randomly select r_1, r_2, r_3 from $\{1, ..., N\}$ such that they differ from each other;
    $F'_{i,t} \leftarrow F_{r_1,t} + \mathrm{randn}(0, 0.5)(F_{r_2,t} - F_{r_3,t})$;
    $C_{i,t} \leftarrow \mathrm{randn}(0.5, 0.15)$;
    Generate the mutant vector $\boldsymbol{v}^{i,t}$ using an arbitrary mutation strategy with $F'_{i,t}$;
    Generate the trial vector $\boldsymbol{u}^{i,t}$ by crossing $\boldsymbol{x}^{i,t}$ and $\boldsymbol{v}^{i,t}$ using an arbitrary crossover method with $C_{i,t}$;
  for $i \in \{1, ..., N\}$ **do**
    if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$ **then**
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{u}^{i,t}$, $F_{i,t+1} \leftarrow F'_{i,t}$;
    else
      $\boldsymbol{x}^{i,t+1} \leftarrow \boldsymbol{x}^{i,t}$;
  $t \leftarrow t + 1$;

TABLE S.2: Parameter settings for the DE without a PCM: Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F	[0, 1]	0.5
C	[0, 1]	0.9

(b) rand/1/bin

	#1	#2	#3	Best
r^N	1.09	2.25	0.70	5
F	0.43	0.31	0.48	0.5
C	0.31	0.61	0.48	0.9

(c) rand/1/sec

	#1	#2	#3	Best
r^N	4.13	3.91	1.75	5
F	0.28	0.26	0.51	0.5
C	0.84	0.77	0.75	0.9

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	2.30	9.78	0.98	5
F	0.56	0.43	0.66	0.5
C	0.21	0.66	0.25	0.9

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	5.88	8.63	2.64	5
F	0.47	0.61	0.47	0.5
C	0.88	0.15	0.83	0.9

TABLE S.3: Parameter settings for PCM-DERSF. Table (a) shows the range of each parameter value and their default settings, where $F^{\min} < F^{\max}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
C	[0, 1]	0.9
$F^{\min}$	[0, 1]	0.5
$F^{\max}$	[0, 1]	1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	5.35	2.42	4.75	5
C	0.87	0.55	0.86	0.9
$F^{\min}$	0.44	0.01	0.07	0.5
$F^{\max}$	0.55	0.14	0.76	1

(c) rand/1/sec

	#1	#2	#3	Best
r^N	3.24	0.64	3.65	0.64
C	0.83	0.88	0.83	0.88
$F^{\min}$	0.09	0.41	0.50	0.41
$F^{\max}$	0.96	0.80	0.51	0.80

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	1.15	4.23	1.88	5
C	0.42	0.62	0.17	0.9
$F^{\min}$	0.57	0.05	0.65	0.5
$F^{\max}$	0.58	0.49	0.68	1

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	2.44	2.06	4.93	5
C	0.78	0.77	0.76	0.9
$F^{\min}$	0.70	0.37	0.63	0.5
$F^{\max}$	0.91	0.96	0.70	1

TABLE S.4: Parameter settings for PCM-DETVSF: Table (a) shows the range of each parameter value and their default settings, where $F^{\min} < F^{\max}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
C	[0, 1]	0.9
$F^{\min}$	[0, 1]	0.4
$F^{\max}$	[0, 2]	1.2

(b) rand/1/bin

	#1	#2	#3	Best
r^N	7.07	3.43	8.52	7.07
C	0.86	0.75	0.71	0.86
$F^{\min}$	0.11	0.05	0.27	0.11
$F^{\max}$	1.10	1.99	0.68	1.10

(c) rand/1/sec

	#1	#2	#3	Best
r^N	1.39	1.17	0.55	0.55
C	0.71	0.28	0.74	0.74
$F^{\min}$	0.25	0.06	0.22	0.22
$F^{\max}$	0.68	1.44	0.96	0.96

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	2.47	6.47	3.30	6.47
C	0.18	0.97	0.92	0.97
$F^{\min}$	0.14	0.51	0.57	0.51
$F^{\max}$	0.82	1.24	1.37	1.24

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	0.85	5.90	4.58	5
C	0.84	0.72	0.66	0.9
$F^{\min}$	0.08	0.23	0.19	0.4
$F^{\max}$	1.29	1.65	1.22	1.2

TABLE S.5: Parameter settings for PCM-SinDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
ω	[0.1, 0.4]	0.25

(b) rand/1/bin

	#1	#2	#3	Best
r^N	1.91	2.88	6.61	2.88
ω	0.16	0.21	0.24	0.21

(c) rand/1/sec

	#1	#2	#3	Best
r^N	0.90	6.02	4.01	4.01
ω	0.27	0.36	0.29	0.29

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	3.15	1.51	1.79	5
ω	0.14	0.13	0.15	0.25

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	0.97	5.20	1.51	5
ω	0.33	0.30	0.39	0.25

TABLE S.6: Parameter settings for PCM-ZMDE. Table (a) shows the range of each parameter value and their default settings, where $C^{\min} < C^{\max}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
μ_F	[0.5, 1]	0.75
$C^{\min}$	[0, 1]	0.8
$C^{\max}$	[0, 1]	1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	0.53	2.63	1.21	1.21
μ_F	0.58	0.72	0.69	0.69
$C^{\min}$	0.10	0.20	0.27	0.27
$C^{\max}$	0.79	0.84	0.89	0.89

(c) rand/1/sec

	#1	#2	#3	Best
r^N	4.43	1.86	2.89	1.86
μ_F	0.54	0.51	0.57	0.51
$C^{\min}$	0.18	0.63	0.02	0.63
$C^{\max}$	0.90	0.94	0.97	0.94

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	6.82	6.56	0.67	6.56
μ_F	0.72	0.57	0.59	0.57
$C^{\min}$	0.22	0.83	0.03	0.83
$C^{\max}$	0.84	0.97	0.60	0.97

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	1.61	5.74	0.66	0.66
μ_F	0.89	0.57	0.88	0.88
$C^{\min}$	0.68	0.22	0.85	0.85
$C^{\max}$	1.00	0.42	0.99	0.99

TABLE S.7: Parameter settings for PCM-CoDE: Table (a) shows the range of each parameter value and their default settings, where $F^{\min} < F^{\max}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F_1	[0, 1]	1
C_1	[0, 1]	0.1
F_2	[0, 1]	1
C_2	[0, 1]	0.9
F_3	[0, 1]	0.8
C_3	[0, 1]	0.2

(b) rand/1/bin

	#1	#2	#3	Best
r^N	5.31	9.86	0.51	0.51
F_1	0.04	0.96	0.92	0.92
C_1	0.82	0.92	0.60	0.60
F_2	0.54	0.39	0.51	0.51
C_2	0.13	0.01	0.51	0.51
F_3	0.70	0.46	0.07	0.07
C_3	0.93	0.94	0.31	0.31

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.55	1.70	3.43	2.55
F_1	0.52	0.68	0.55	0.52
C_1	0.34	0.90	0.24	0.34
F_2	0.28	0.10	0.60	0.28
C_2	0.33	0.55	0.99	0.33
F_3	0.75	0.49	0.57	0.75
C_3	0.97	0.30	0.71	0.97

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	4.35	7.51	1.11	1.11
F_1	0.93	0.92	0.91	0.91
C_1	0.28	0.26	0.92	0.92
F_2	0.23	0.77	0.38	0.38
C_2	0.57	0.62	0.21	0.21
F_3	1.00	0.42	0.99	0.99
C_3	0.98	0.36	0.00	0.00

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	1.43	7.00	4.30	4.30
F_1	0.72	0.25	0.87	0.87
C_1	0.73	0.68	0.89	0.89
F_2	0.55	0.97	0.92	0.92
C_2	0.82	0.82	0.38	0.38
F_3	0.37	0.58	0.36	0.36
C_3	0.59	0.02	0.48	0.48

TABLE S.8: Parameter settings for PCM-SWDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F_1	[0, 2]	0.5
F_2	[0, 2]	2
C_1	[0, 1]	0
C_2	[0, 1]	1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	4.98	0.66	1.48	1.48
F_1	0.72	1.21	1.06	1.06
F_2	0.05	1.63	0.98	0.98
C_1	0.16	0.62	1.00	1.00
C_2	0.67	0.05	0.05	0.05

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.01	3.14	1.67	3.14
F_1	0.89	0.57	0.17	0.57
F_2	0.36	1.43	1.19	1.43
C_1	0.82	0.49	0.80	0.49
C_2	0.92	0.97	0.90	0.97

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	2.62	0.73	2.64	5
F_1	0.97	1.17	1.22	0.5
F_2	0.49	0.12	0.07	2
C_1	0.53	0.15	0.28	0
C_2	0.34	0.45	0.82	1

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	8.06	7.85	0.59	5
F_1	0.08	1.65	0.41	0.5
F_2	1.02	0.77	1.87	2
C_1	0.89	0.95	0.87	0
C_2	0.76	0.43	0.13	1

TABLE S.9: Parameter settings for PCM-DEPD. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
$F^{\min}$	[0, 1]	0.4
C	[0, 1]	0.5

(b) rand/1/bin

	#1	#2	#3	Best
r^N	0.78	6.61	4.22	4.22
$F^{\min}$	0.25	0.26	0.93	0.93
C	0.39	0.10	0.81	0.81

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.04	3.08	1.48	1.48
$F^{\min}$	0.22	0.34	0.62	0.62
C	0.80	0.86	0.89	0.89

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	1.02	0.60	0.95	0.60
$F^{\min}$	0.87	0.65	0.67	0.65
C	0.29	0.50	0.34	0.50

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	4.04	3.87	7.53	7.53
$F^{\min}$	0.36	0.56	0.60	0.60
C	0.05	0.23	0.77	0.77

TABLE S.10: Parameter settings for PCM-jDE: Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F^{init}	[0, 1]	0.5
C^{init}	[0, 1]	0.9
τ_F	[0.05, 0.3]	0.1
τ_C	[0.05, 0.3]	0.1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	1.36	1.07	7.43	1.07
F^{init}	0.29	0.71	0.02	0.71
C^{init}	0.81	0.48	0.47	0.48
τ_F	0.12	0.06	0.14	0.06
τ_C	0.10	0.14	0.21	0.14

(c) rand/1/sec

	#1	#2	#3	Best
r^N	1.10	2.19	1.86	1.10
F^{init}	0.07	0.13	0.03	0.07
C^{init}	0.61	0.13	0.05	0.61
τ_F	0.26	0.06	0.24	0.26
τ_C	0.08	0.29	0.23	0.08

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	3.29	3.21	0.76	3.29
F^{init}	0.60	0.87	0.16	0.60
C^{init}	0.03	0.72	0.34	0.03
τ_F	0.18	0.15	0.13	0.18
τ_C	0.25	0.08	0.12	0.25

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	5.24	3.87	1.47	1.47
F^{init}	0.77	0.50	0.96	0.96
C^{init}	0.40	0.93	0.19	0.19
τ_F	0.16	0.14	0.26	0.26
τ_C	0.10	0.19	0.16	0.16

TABLE S.11: Parameter settings for PCM-FDSADE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F^{init}	[0, 1]	0.5
C^{init}	[0, 1]	0.9
K	[0, 1]	0.3

(b) rand/1/bin

	#1	#2	#3	Best
r^N	1.58	0.90	5.09	1.58
F^{init}	0.33	0.66	0.25	0.33
C^{init}	0.92	0.92	0.36	0.92
K	0.23	0.03	0.12	0.23

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.22	0.51	1.70	0.51
F^{init}	0.51	0.50	0.24	0.50
C^{init}	0.58	0.70	0.23	0.70
K	0.26	0.42	0.64	0.42

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	1.56	6.37	1.36	5
F^{init}	0.49	0.58	0.48	0.5
C^{init}	0.11	0.10	0.56	0.9
K	0.40	0.48	0.86	0.3

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	3.97	0.73	2.41	2.41
F^{init}	0.60	0.86	0.01	0.01
C^{init}	0.18	0.31	0.12	0.12
K	0.05	0.95	0.61	0.61

TABLE S.12: Parameter settings for PCM-ISADE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
τ_F	[0.05, 0.3]	0.1
τ_C	[0.05, 0.3]	0.1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	4.30	2.44	1.44	2.44
τ_F	0.21	0.08	0.24	0.08
τ_C	0.18	0.27	0.26	0.27

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.51	2.32	1.76	2.51
τ_F	0.09	0.06	0.15	0.09
τ_C	0.26	0.25	0.19	0.26

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	9.76	3.43	3.49	3.49
τ_F	0.16	0.07	0.21	0.21
τ_C	0.25	0.23	0.28	0.28

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	4.44	5.68	2.22	2.22
τ_F	0.07	0.23	0.07	0.07
τ_C	0.25	0.25	0.28	0.28

TABLE S.13: Parameter settings for PCM-cDE: Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F_1	[0, 1]	0.5
F_2	[0, 1]	0.8
F_3	[0, 1]	1
C_1	[0, 1]	0
C_2	[0, 1]	0.5
C_3	[0, 1]	1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	3.50	7.38	4.79	5
F_1	0.85	0.89	0.75	0.5
F_2	0.45	0.55	0.72	0.8
F_3	0.02	0.10	0.63	1
C_1	0.51	0.92	0.81	0
C_2	0.06	0.70	0.28	0.5
C_3	0.51	0.43	0.84	1

(c) rand/1/sec

	#1	#2	#3	Best
r^N	4.68	1.20	5.00	5
F_1	0.85	0.62	0.50	0.5
F_2	0.31	0.57	0.80	0.8
F_3	0.90	0.90	1.00	1
C_1	0.18	0.08	0.00	0
C_2	0.85	0.92	0.50	0.5
C_3	0.96	0.20	1.00	1

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	0.75	2.54	6.45	5
F_1	0.97	0.93	0.10	0.5
F_2	0.75	0.45	0.84	0.8
F_3	0.44	0.87	0.20	1
C_1	0.70	0.38	0.59	0
C_2	0.19	0.04	0.96	0.5
C_3	0.75	0.81	0.00	1

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	4.68	0.95	5.43	5
F_1	0.85	0.87	0.80	0.5
F_2	0.31	0.04	0.17	0.8
F_3	0.90	0.39	0.70	1
C_1	0.10	0.49	0.19	0
C_2	0.85	0.93	0.93	0.5
C_3	0.96	0.26	0.47	1

TABLE S.14: Parameter settings for PCM-SaDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
C^{init}	[0, 1]	0.5
μ_F	[0, 1]	0.5
t^{learn}	$\{1, \ldots, 100\}$	50

(b) rand/1/bin

	#1	#2	#3	Best
r^N	5.26	2.89	6.58	6.58
C^{init}	0.76	0.59	0.76	0.76
μ_F	0.05	0.90	0.62	0.62
t^{learn}	79	71	17	17

(c) rand/1/sec

	#1	#2	#3	Best
r^N	5.33	5.29	5.47	5.33
C^{init}	0.88	0.97	1.00	0.88
μ_F	0.03	0.41	0.38	0.03
t^{learn}	58	45	90	58

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	4.52	0.70	0.97	4.52
C^{init}	0.68	0.55	0.52	0.68
μ_F	0.75	0.15	0.46	0.75
t^{learn}	84	35	10	84

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	2.30	2.42	6.56	6.56
C^{init}	0.76	0.79	0.82	0.82
μ_F	0.02	0.12	0.13	0.13
t^{learn}	40	55	14	14

TABLE S.15: Parameter settings for PCM-SaNSDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
C^{init}	[0, 1]	0.5
p^{init}	[0, 1]	0.5
μ_F	[0, 1]	0.5
t^{learn}	$\{1, \ldots, 100\}$	50

(b) rand/1/bin

	#1	#2	#3	Best
r^N	2.57	3.10	8.25	2.57
C^{init}	0.74	0.88	0.44	0.74
p^{init}	0.77	0.72	0.15	0.77
μ_F	0.37	0.34	0.17	0.37
t^{learn}	18	42	20	18

(c) rand/1/sec

	#1	#2	#3	Best
r^N	7.34	2.96	8.48	2.96
C^{init}	0.98	0.83	0.96	0.83
p^{init}	0.45	0.97	0.38	0.97
μ_F	0.14	0.05	0.39	0.05
t^{learn}	60	38	29	38

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	5.31	2.54	7.63	5.31
C^{init}	0.45	0.78	0.03	0.45
p^{init}	0.63	0.55	0.42	0.63
μ_F	0.28	0.35	0.34	0.28
t^{learn}	1	76	69	1

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	7.35	2.16	1.50	1.50
C^{init}	0.65	0.56	0.91	0.91
p^{init}	0.17	0.13	0.99	0.99
μ_F	0.61	0.33	0.13	0.13
t^{learn}	35	88	92	92

TABLE S.16: Parameter settings for PCM-JADE: Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
μ_F^{init}	[0, 1]	0.5
μ_C^{init}	[0, 1]	0.5
c	[0.05, 0.2]	0.1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	1.42	3.39	3.75	1.42
μ_F^{init}	0.07	0.11	0.87	0.07
μ_C^{init}	0.98	0.24	0.09	0.98
c	0.08	0.16	0.12	0.08

(c) rand/1/sec

	#1	#2	#3	Best
r^N	8.18	0.75	0.51	8.18
μ_F^{init}	0.15	0.64	0.63	0.15
μ_C^{init}	1.00	0.88	0.61	1.00
c	0.08	0.19	0.19	0.08

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	4.11	1.29	1.22	4.11
μ_F^{init}	0.07	0.51	0.63	0.07
μ_C^{init}	0.60	0.80	0.86	0.60
c	0.11	0.15	0.18	0.11

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	3.46	7.00	5.63	5.63
μ_F^{init}	0.66	0.46	0.43	0.43
μ_C^{init}	0.16	0.32	0.83	0.83
c	0.18	0.15	0.09	0.09

TABLE S.17: Parameter settings for PCM-IMDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
μ_F^{init}	[0, 1]	0.5
μ_C^{init}	[0, 1]	0.5

(b) rand/1/bin

	#1	#2	#3	Best
r^N	2.86	6.61	6.09	6.61
μ_F^{init}	0.14	0.51	0.12	0.51
μ_C^{init}	0.66	0.98	0.77	0.98

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.05	8.05	2.49	8.05
μ_F^{init}	0.34	0.39	0.58	0.39
μ_C^{init}	0.91	0.94	0.89	0.94

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	4.02	2.45	7.76	7.76
μ_F^{init}	0.22	0.54	0.91	0.91
μ_C^{init}	0.66	0.50	0.76	0.76

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	7.17	4.83	2.36	7.17
μ_F^{init}	0.68	0.33	0.53	0.68
μ_C^{init}	0.92	0.77	0.83	0.92

TABLE S.18: Parameter settings for PCM-SHADE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F^{init}	[0, 1]	0.5
C^{init}	[0, 1]	0.5
H	$\{5, \ldots, 20\}$	10

(b) rand/1/bin

	#1	#2	#3	Best
r^N	3.90	4.03	3.52	5
F^{init}	0.31	0.37	0.80	0.5
C^{init}	0.88	0.89	0.91	0.5
H	1	11	19	10

(c) rand/1/sec

	#1	#2	#3	Best
r^N	2.35	4.76	4.79	2.35
F^{init}	0.18	0.44	0.40	0.18
C^{init}	0.68	0.33	0.64	0.68
H	7	4	3	7

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	3.96	3.90	0.94	5
F^{init}	0.49	0.14	0.50	0.5
C^{init}	0.75	0.40	0.15	0.5
H	19	8	13	10

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	5.05	5.95	4.51	5.05
F^{init}	0.09	0.29	0.08	0.09
C^{init}	0.57	0.59	0.57	0.57
H	2	5	10	2

TABLE S.19: Parameter settings for PCM-SLADE: Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
μ_F^{init}	[0, 1]	0.5
μ_C^{init}	[0, 1]	0.5
c	[0.05, 0.2]	0.1

(b) rand/1/bin

	#1	#2	#3	Best
r^N	0.89	2.81	3.24	2.81
μ_F^{init}	0.89	0.92	0.45	0.92
μ_C^{init}	0.62	0.66	0.63	0.66
c	0.09	0.09	0.08	0.09

(c) rand/1/sec

	#1	#2	#3	Best
r^N	0.63	1.02	2.45	2.45
μ_F^{init}	0.41	0.55	0.61	0.61
μ_C^{init}	0.48	0.98	0.14	0.14
c	0.06	0.14	0.09	0.09

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	6.32	5.07	9.63	6.32
μ_F^{init}	0.87	0.67	0.57	0.87
μ_C^{init}	0.21	0.74	0.78	0.21
c	0.15	0.17	0.12	0.15

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	3.64	4.05	3.38	3.64
μ_F^{init}	0.81	0.78	0.54	0.81
μ_C^{init}	0.60	0.30	0.24	0.60
c	0.05	0.16	0.16	0.05

TABLE S.20: Parameter settings for PCM-EPSDE. Table (a) shows the range of each parameter value and their default settings, where $F^{\mathrm{start}} < F^{\mathrm{end}}$ and $C^{\mathrm{start}} < C^{\mathrm{end}}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F^{start}	$\{0.1, ..., 1\}$	0.4
F^{end}	$\{0.2, ..., 1\}$	0.9
C^{start}	$\{0, ..., 1\}$	0.1
C^{end}	$\{0.1, ..., 1\}$	0.9

(b) rand/1/bin

	#1	#2	#3	Best
r^N	5.68	0.74	2.04	2.04
F^{start}	0.3	0.4	0.9	0.9
F^{end}	0.4	0.9	1	1
C^{start}	0	0.3	0	0
C^{end}	0.3	1	1	1

(c) rand/1/sec

	#1	#2	#3	Best
r^N	7.83	8.00	4.24	4.24
F^{start}	0.4	0.7	0.3	0.3
F^{end}	0.7	0.9	0.8	0.8
C^{start}	0.2	0.5	0	0
C^{end}	0.8	1	1	1

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	1.00	5.26	3.04	5
F^{start}	0.3	0.2	0.2	0.4
F^{end}	1	0.7	0.9	0.9
C^{start}	0.5	0.1	0	0.1
C^{end}	1	0.2	0.3	0.9

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	9.21	1.74	3.20	1.74
F^{start}	0.3	0.2	1	0.2
F^{end}	1	1	1	1
C^{start}	0.4	0.5	0.9	0.5
C^{end}	1	0.9	1	0.9

TABLE S.21: Parameter settings for PCM-CoBiDE. Table (a) shows the range of each parameter value and their default settings, where $\mu_F^1 < \mu_F^2$ and $\mu_C^1 < \mu_C^2$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
μ_F^1	$[0, 1]$	0.65
μ_F^2	$[0, 1]$	1
μ_C^1	$[0, 1]$	0.1
μ_C^2	$[0, 1]$	0.95

(b) rand/1/bin

	#1	#2	#3	Best
r^N	1.44	0.71	3.74	0.71
μ_F^1	0.57	0.27	0.23	0.27
μ_F^2	0.66	0.37	0.64	0.37
μ_C^1	0.61	0.20	0.22	0.20
μ_C^2	0.94	0.22	0.55	0.22

(c) rand/1/sec

	#1	#2	#3	Best
r^N	1.02	1.68	1.06	1.06
μ_F^1	0.08	0.36	0.62	0.62
μ_F^2	0.47	0.78	0.89	0.89
μ_C^1	0.37	0.40	0.53	0.53
μ_C^2	0.37	0.53	0.97	0.97

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	3.20	0.98	5.38	5.38
mu_F^1	0.17	0.27	0.18	0.18
mu_F^2	0.79	0.53	0.49	0.49
μ_C^1	0.37	0.49	0.27	0.27
μ_C^2	0.73	0.98	0.62	0.62

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	4.32	4.72	5.72	5
mu_F^1	0.13	0.20	0.64	0.65
mu_F^2	0.17	0.42	0.71	1
μ_C^1	0.11	0.57	0.12	0.1
μ_C^2	0.33	0.62	0.13	0.95

TABLE S.22: Parameter settings for PCM-DEDPS: Table (a) shows the range of each parameter value and their default settings, where $F^{\mathrm{start}} \leq F^{\mathrm{end}}$ and $C^{\mathrm{start}} \leq C^{\mathrm{end}}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
F^{start}	$\{0.1, \ldots, 0.9, 0.99, 1\}$	0.4
F^{end}	$\{0.2, \ldots, 0.9, 0.99, 1\}$	0.99
C^{start}	$\{0, \ldots, 0.9, 0.99, 1\}$	0.2
C^{end}	$\{0.1, \ldots, 0.9, 0.99, 1\}$	0.99

(b) rand/1/bin

	#1	#2	#3	Best
r^N	9.00	5.11	4.78	5
F^{start}	0.8	0.3	0.4	0.4
F^{end}	0.9	0.7	1	0.99
C^{start}	0.6	0.2	0	0.2
C^{end}	0.99	0.9	0.9	0.99

(c) rand/1/sec

	#1	#2	#3	Best
r^N	5.97	9.10	7.12	5.97
F^{start}	0.5	0.3	0.4	0.5
F^{end}	0.8	0.9	1	0.8
C^{start}	0.9	0	0.7	0.9
C^{end}	0.99	0.99	1	0.99

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	6.03	0.60	8.52	5
F^{start}	0.4	0.6	0.4	0.4
F^{end}	0.5	0.6	0.5	0.99
C^{start}	0	0.1	0	0.2
C^{end}	0.99	0.99	1	0.99

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	1.97	1.82	5.00	1.82
F^{start}	0.6	0.6	0.2	0.6
F^{end}	0.99	0.99	0.4	0.99
C^{start}	0.8	0.8	0	0.8
C^{end}	0.8	0.99	0.6	0.99

TABLE S.23: Parameter settings for PCM-RDE. Table (a) shows the range of each parameter value and their default settings, where $F^{\min} < F^{\max}$ and $C^{\min} < C^{\max}$. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
$F^{\min}$	[0, 1]	0.6
$F^{\max}$	[0, 1]	0.95
$C^{\min}$	[0, 1]	0.85
$C^{\max}$	[0, 1]	0.95

(b) rand/1/bin

	#1	#2	#3	Best
r^N	4.18	0.86	2.61	2.61
$F^{\min}$	0.12	0.35	0.39	0.39
$F^{\max}$	0.27	0.74	0.50	0.50
$C^{\min}$	0.03	0.23	0.10	0.10
$C^{\max}$	0.78	0.32	0.70	0.70

(c) rand/1/sec

	#1	#2	#3	Best
r^N	4.58	4.05	4.11	4.05
$F^{\min}$	0.32	0.51	0.32	0.51
$F^{\max}$	0.62	0.57	0.93	0.57
$C^{\min}$	0.06	0.31	0.07	0.31
$C^{\max}$	0.96	0.92	0.86	0.92

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	4.65	3.08	4.61	5
$F^{\min}$	0.18	0.22	0.28	0.6
$F^{\max}$	0.76	0.79	0.47	0.95
$C^{\min}$	0.02	0.59	0.01	0.85
$C^{\max}$	0.85	0.73	0.09	0.95

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	5.23	7.05	4.98	5
$F^{\min}$	0.40	0.25	0.50	0.6
$F^{\max}$	0.52	0.53	0.95	0.95
$C^{\min}$	0.44	0.86	0.54	0.85
$C^{\max}$	0.54	0.90	0.84	0.95

TABLE S.24: Parameter settings for PCM-IDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5

(b) rand/1/bin

	#1	#2	#3	Best
r^N	7.37	3.62	5.87	7.37

(c) rand/1/sec

	#1	#2	#3	Best
r^N	9.69	9.62	7.49	9.69

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	6.46	2.58	4.51	2.58

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	3.62	4.19	8.13	3.62

TABLE S.25: Parameter settings for PCM-YADE: Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5

(b) rand/1/bin

	#1	#2	#3	Best
r^N	3.62	4.10	0.80	0.80

(c) rand/1/sec

	#1	#2	#3	Best
r^N	0.64	0.93	0.76	0.93

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	2.00	3.69	3.85	2.00

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	3.67	3.62	3.35	3.35

TABLE S.26: Parameter settings for PCM-SDE. Table (a) shows the range of each parameter value and their default settings. Table (b), (c), (d), and (e) describe tuned parameter settings by SMAC (three runs #1, #2, and #3 were performed) and the best parameter values among them and the default settings for DE algorithms with the rand/1/bin, rand/1/sec, current-to-$p$best/1/bin, and current-to-$p$best/1/sec operators, respectively.

(a) Range and Default setting

	Range	Default
r^N	[0.5, 10]	5
μ_C	[0, 1]	0.5

(b) rand/1/bin

	#1	#2	#3	Best
r^N	2.39	2.37	8.88	8.88
μ_C	0.39	0.01	0.68	0.68

(c) rand/1/sec

	#1	#2	#3	Best
r^N	7.50	9.38	8.08	7.50
μ_C	0.70	0.80	0.69	0.70

(d) current-to-$p$best/1/bin

	#1	#2	#3	Best
r^N	5.64	5.46	2.54	5.64
μ_C	0.85	0.52	0.78	0.85

(e) current-to-$p$best/1/sec

	#1	#2	#3	Best
r^N	5.67	9.06	4.49	9.06
μ_C	0.81	0.85	0.80	0.85

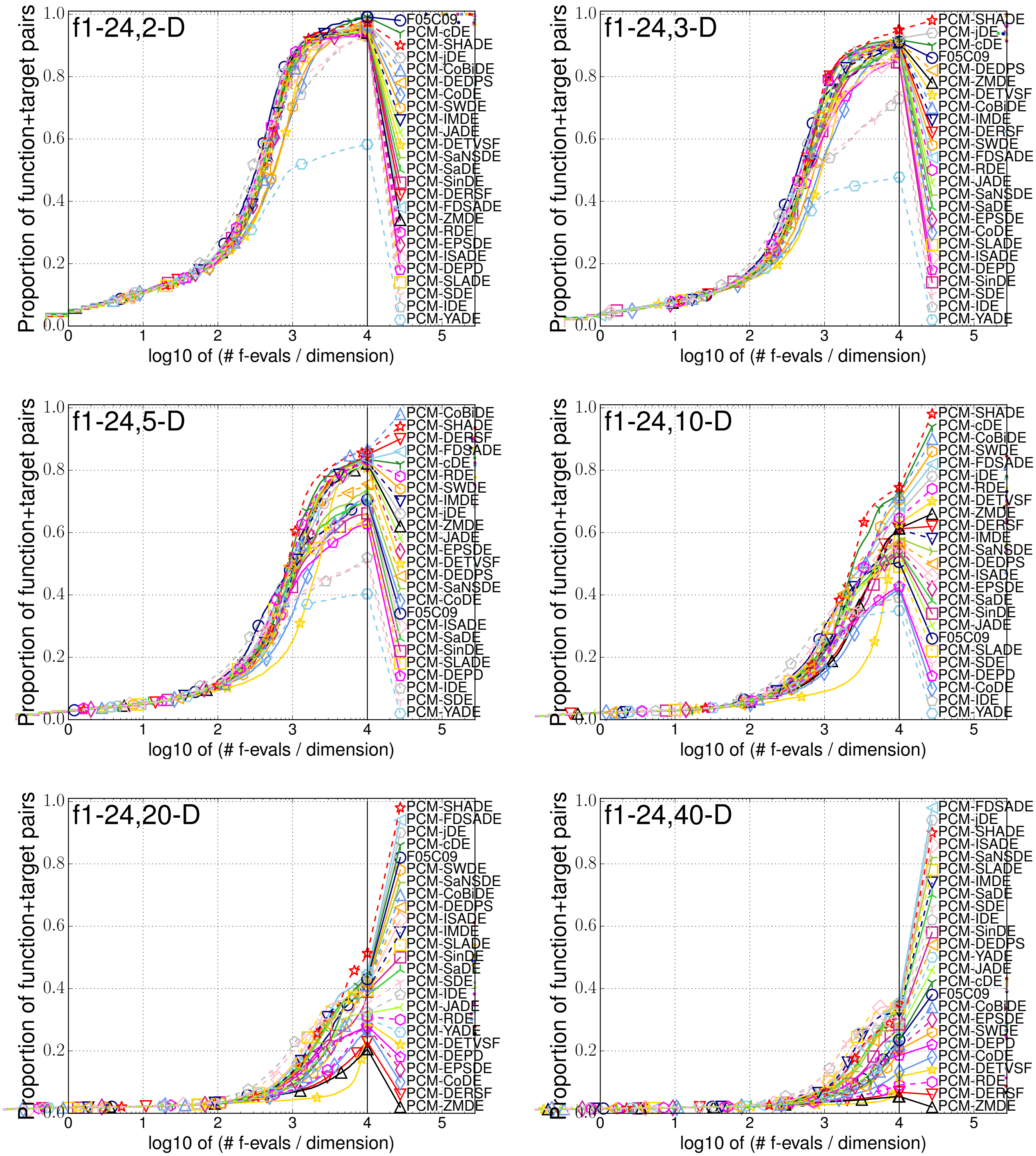

Fig. S.1: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

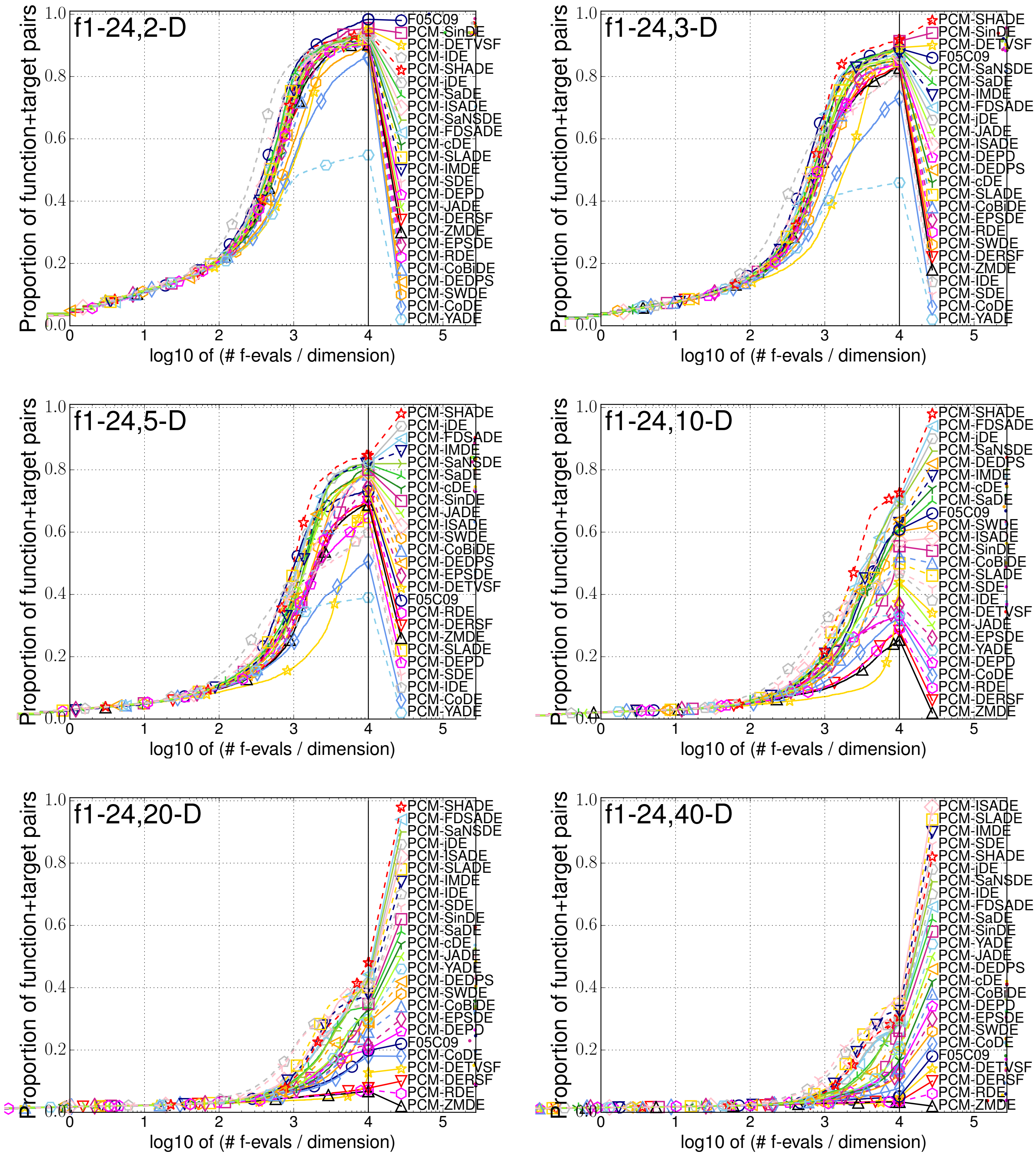

Fig. S.2: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/2/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

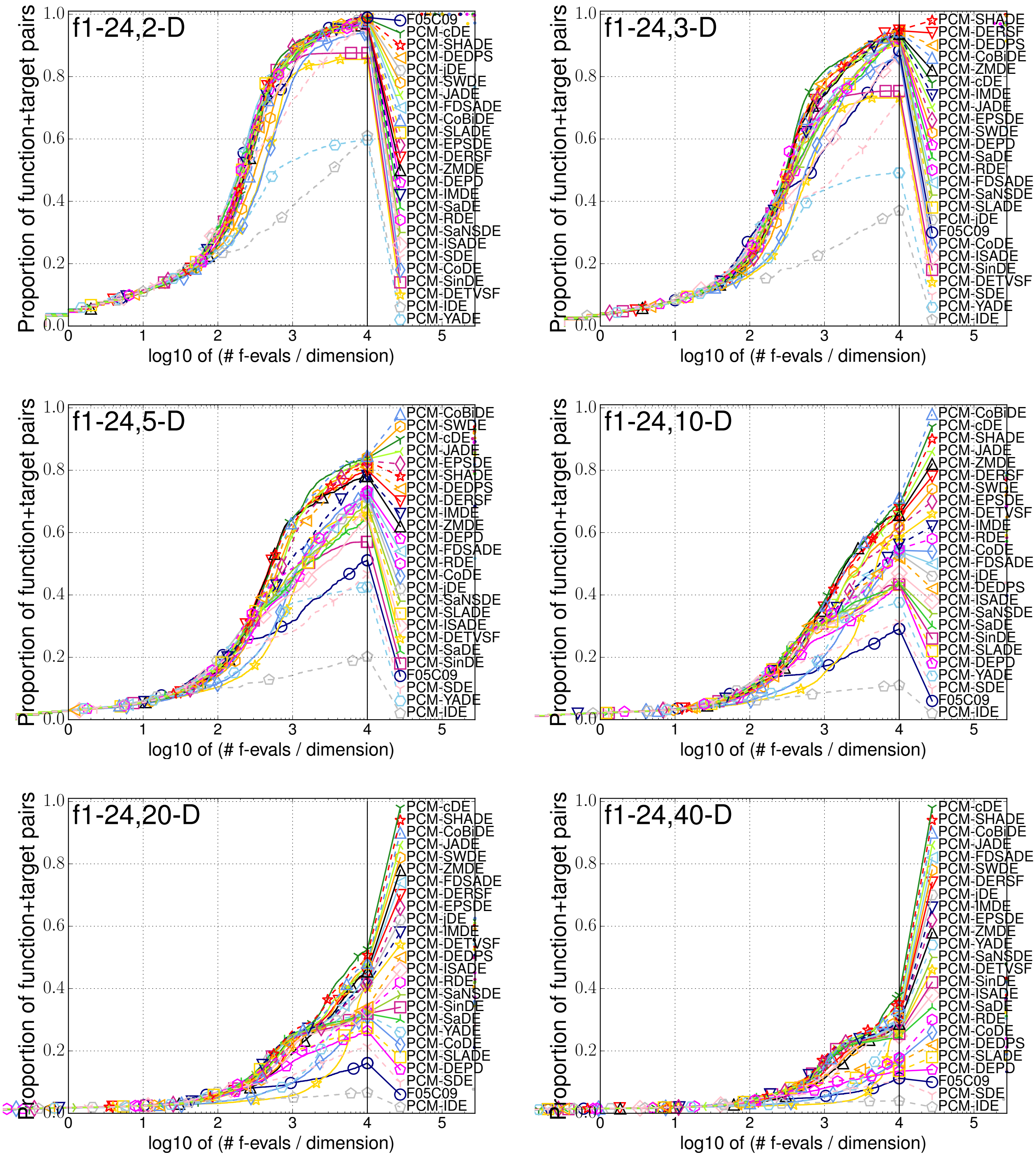

Fig. S.3: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

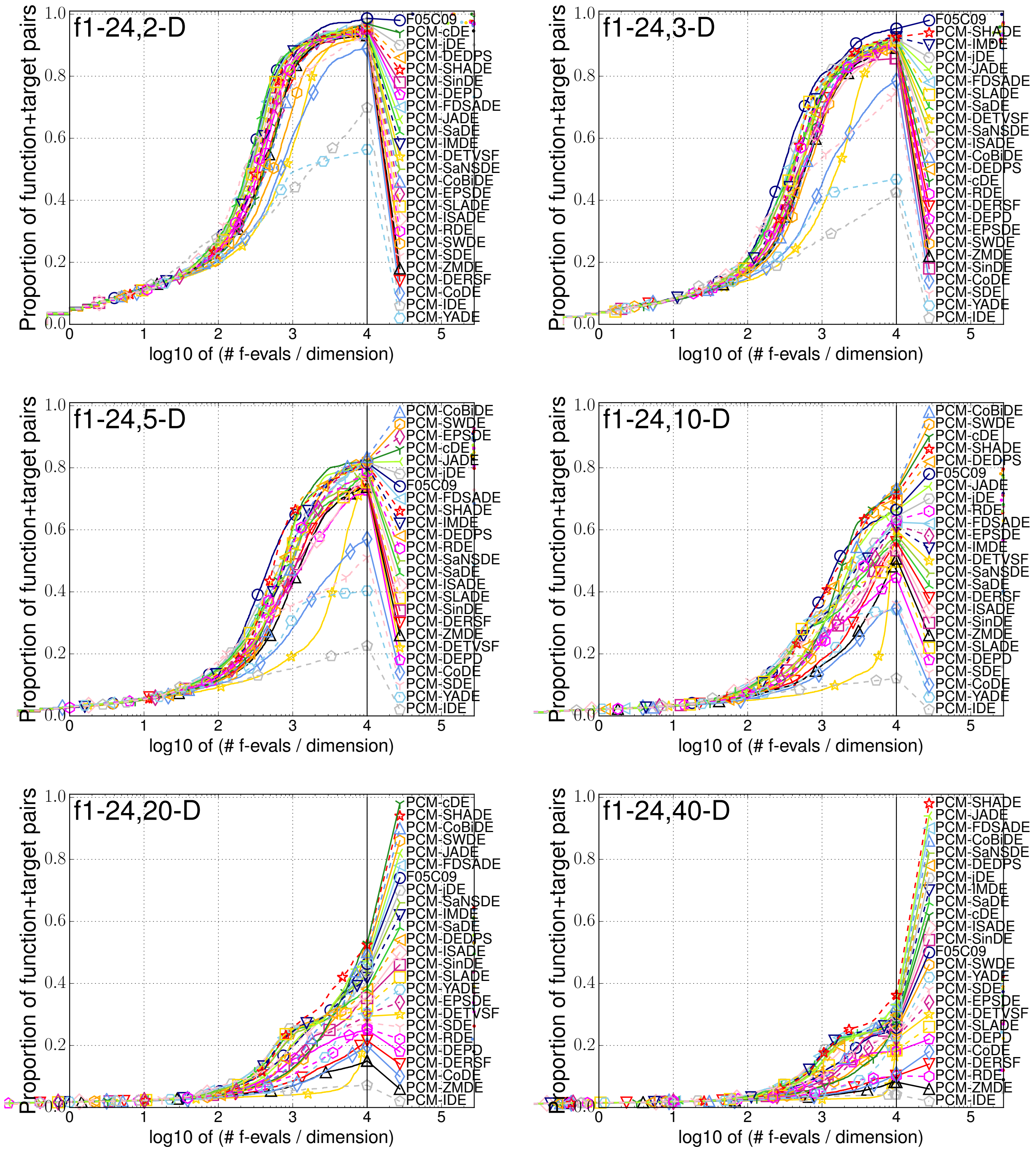

Fig. S.4: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/2/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

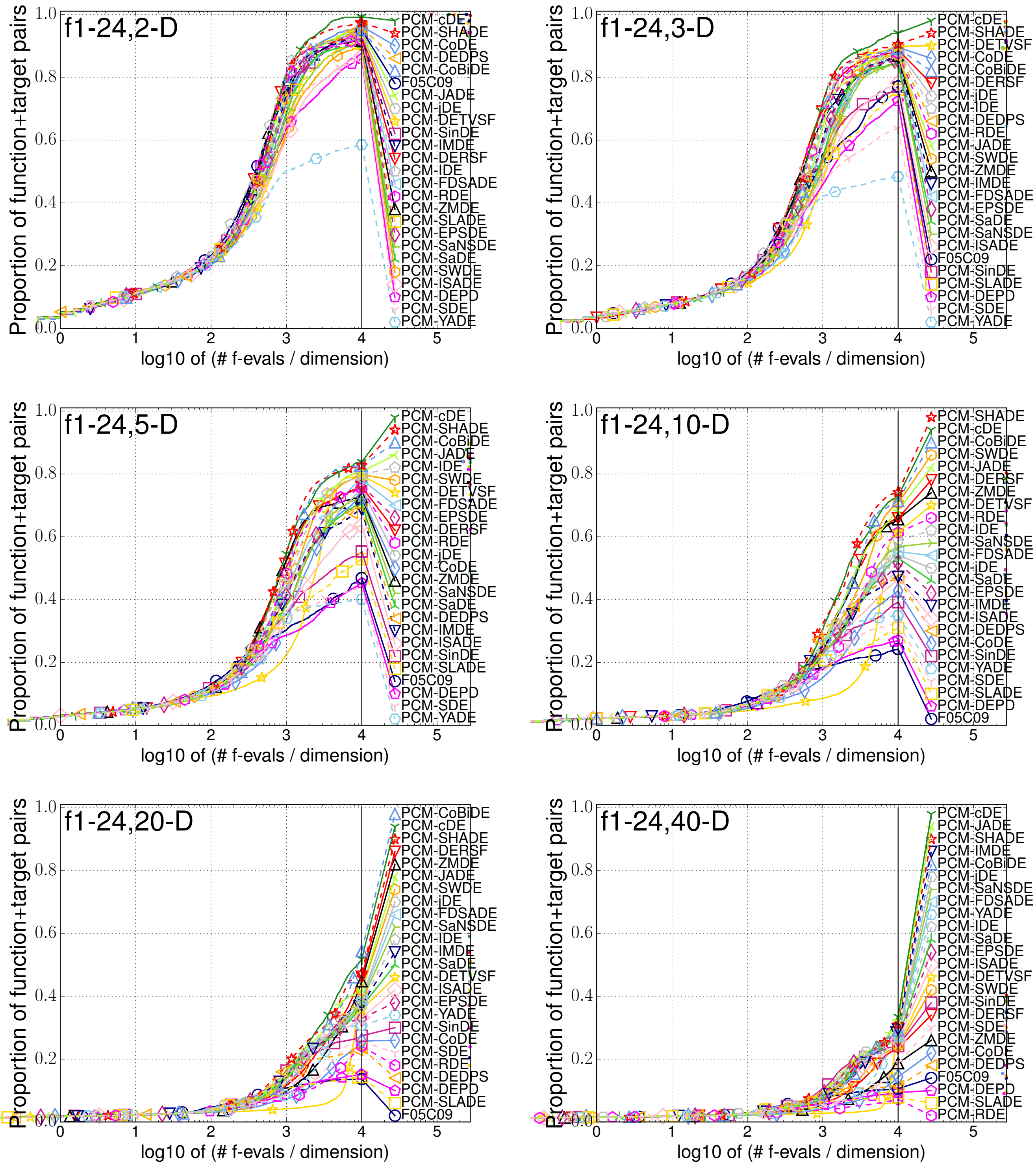

Fig. S.5: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-rand/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

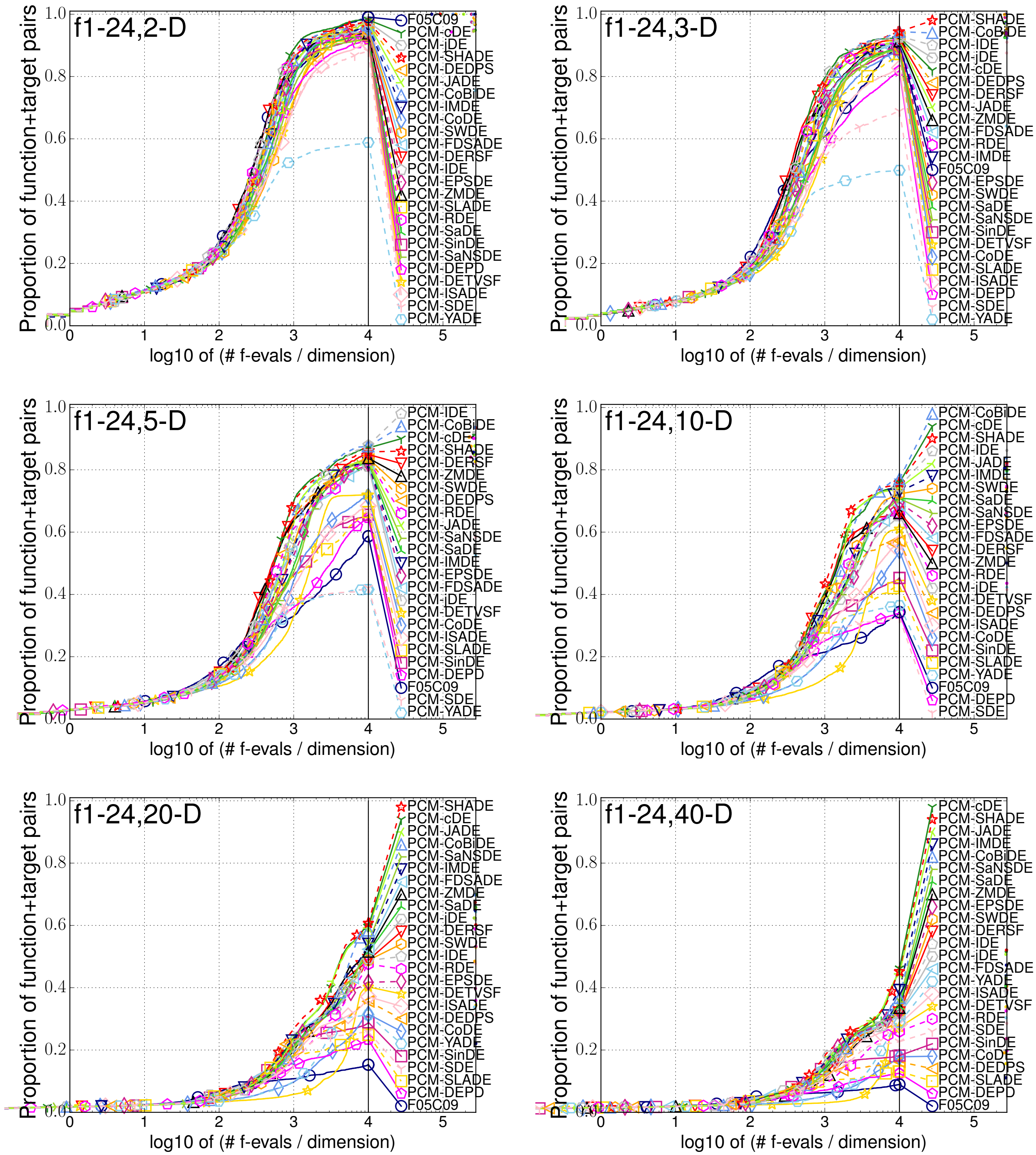

Fig. S.6: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

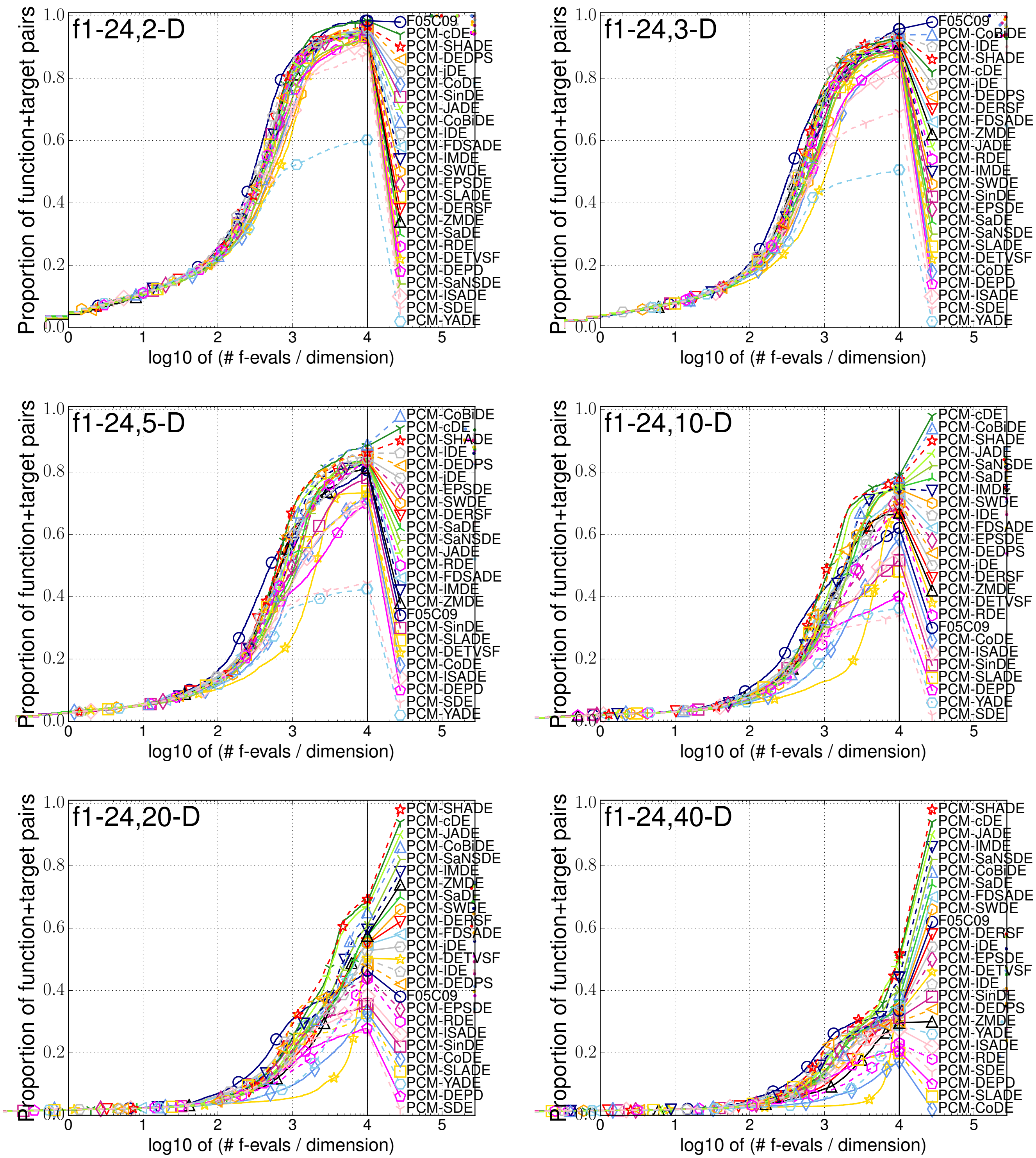

Fig. S.7: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

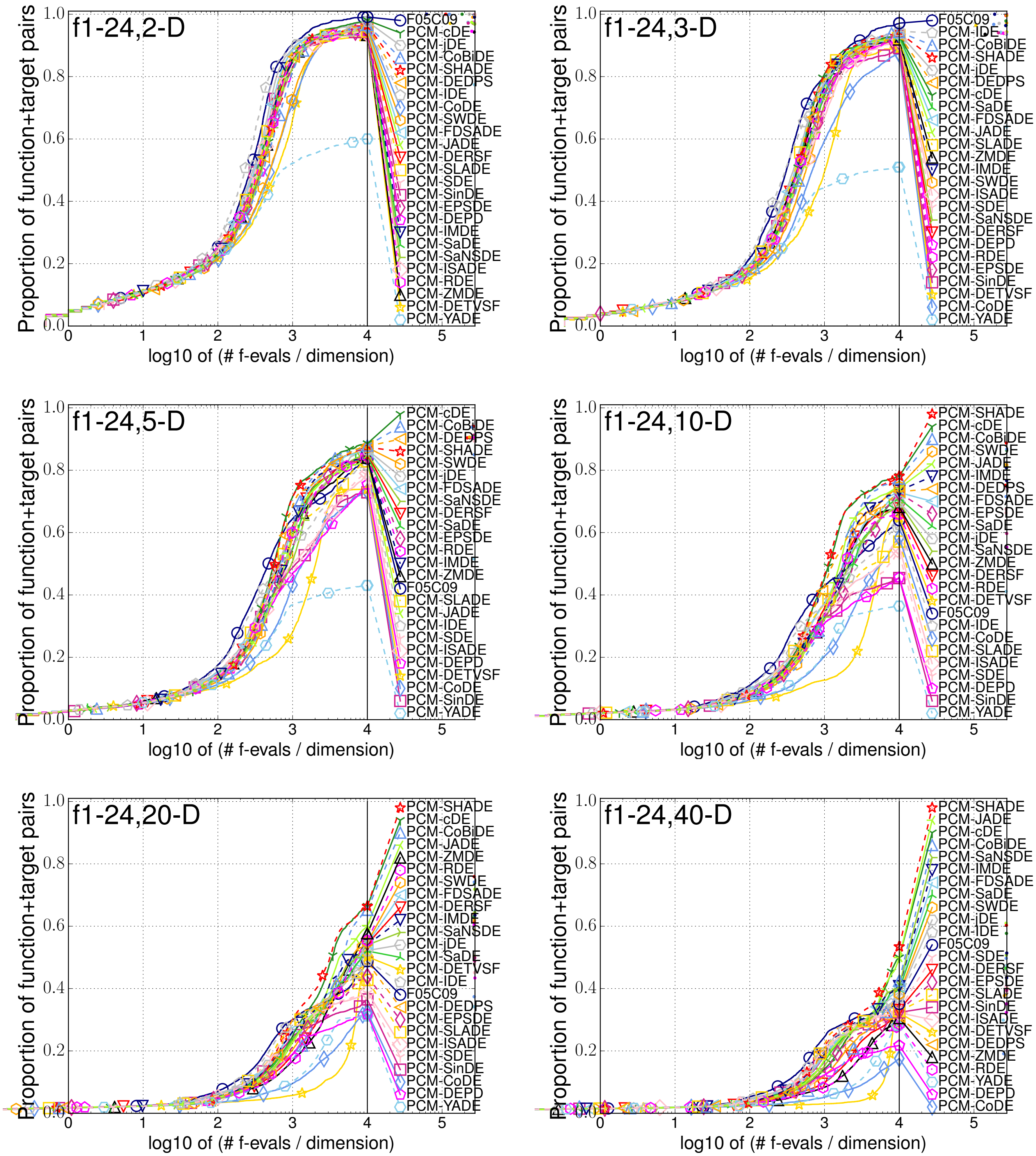

Fig. S.8: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand-to-$p$best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

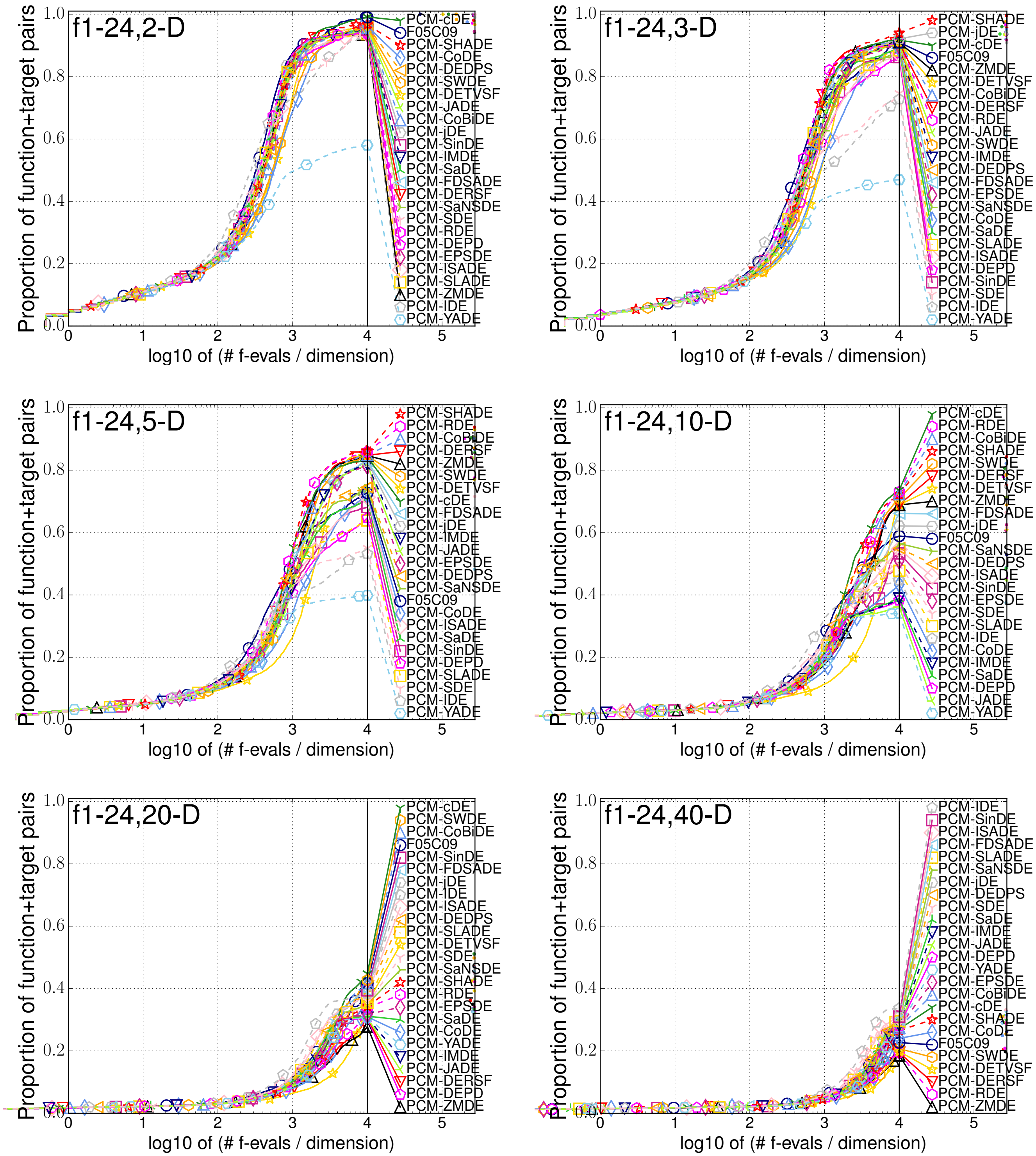

Fig. S.9: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

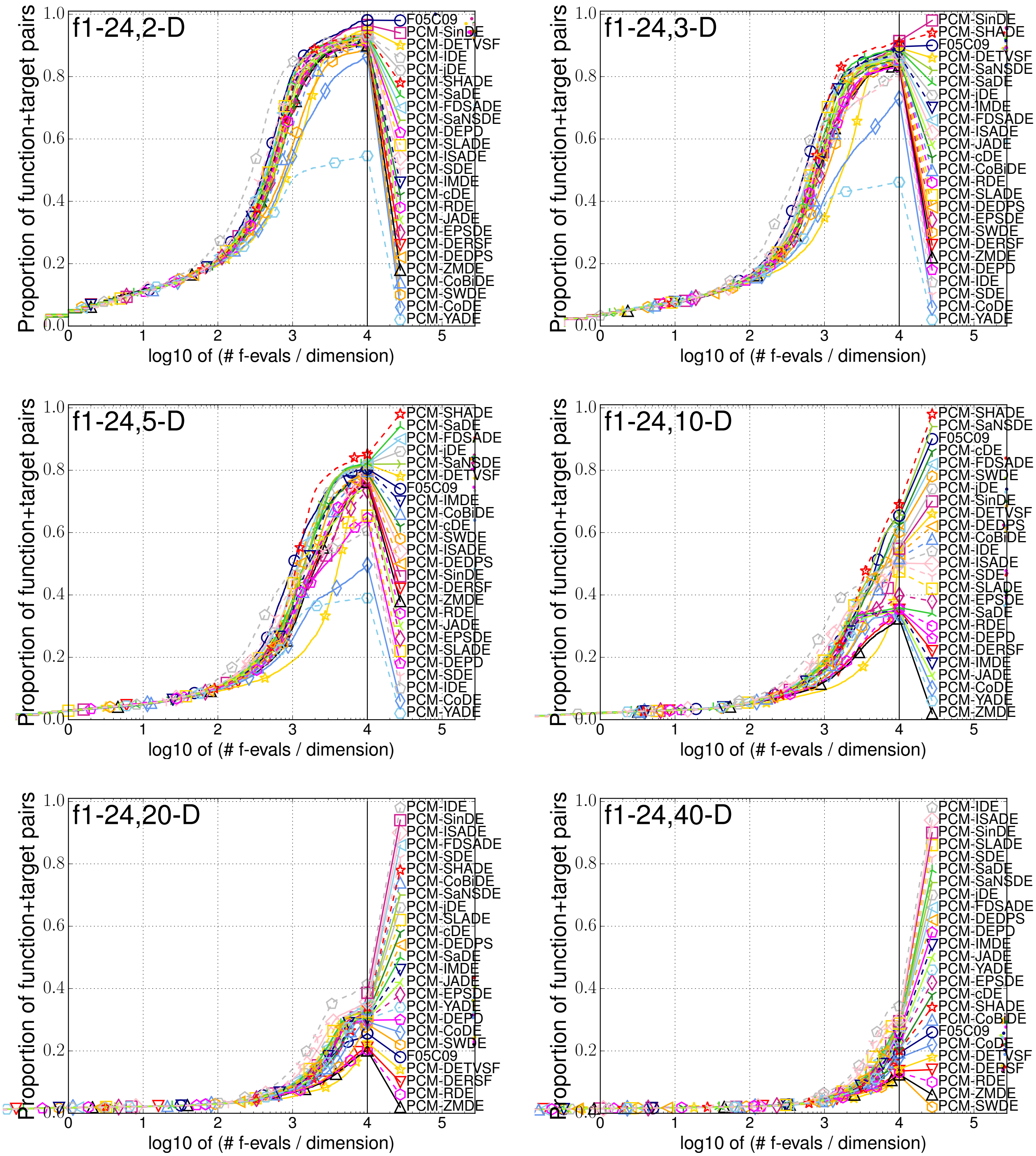

Fig. S.10: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/2/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

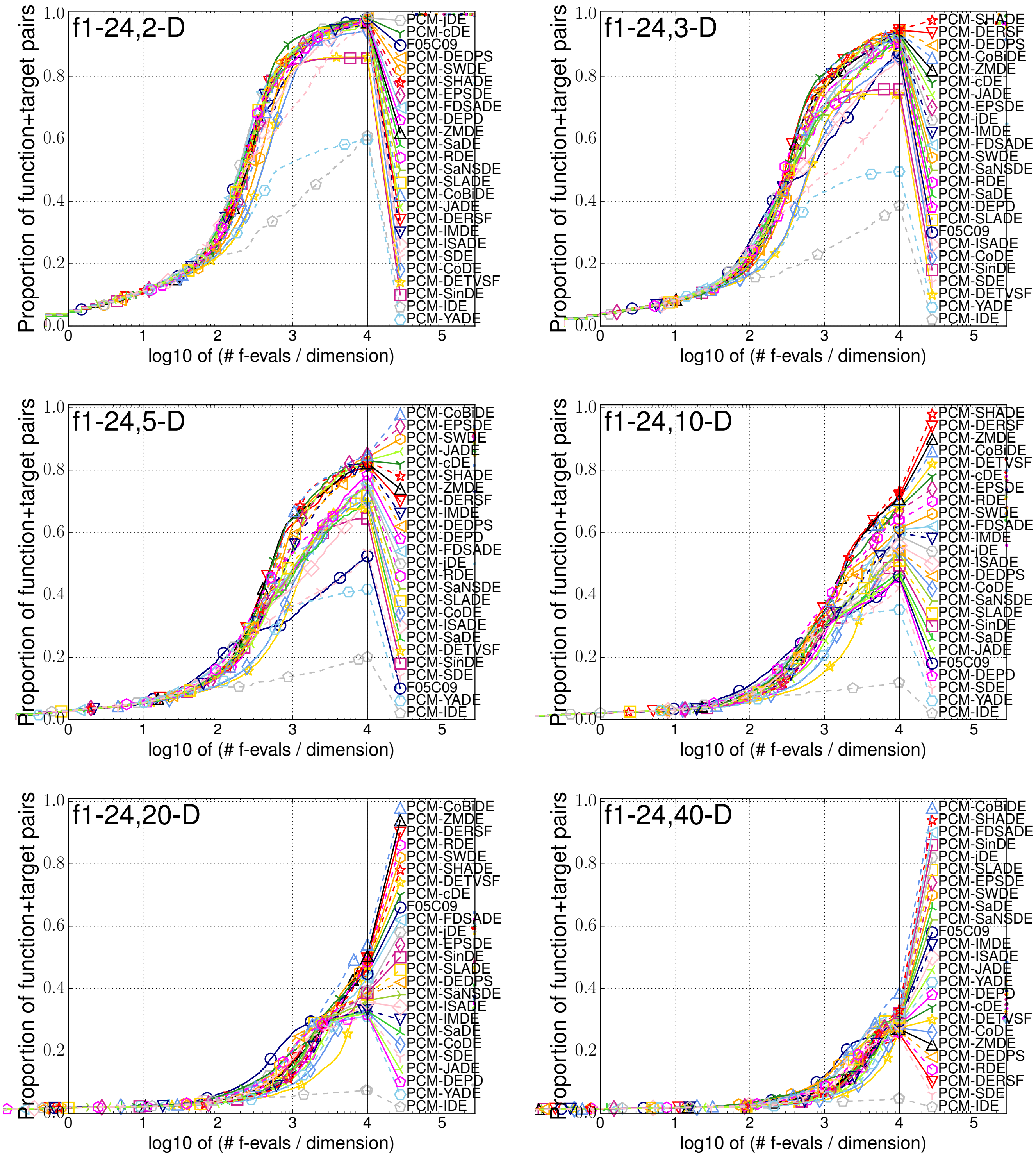

Fig. S.11: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

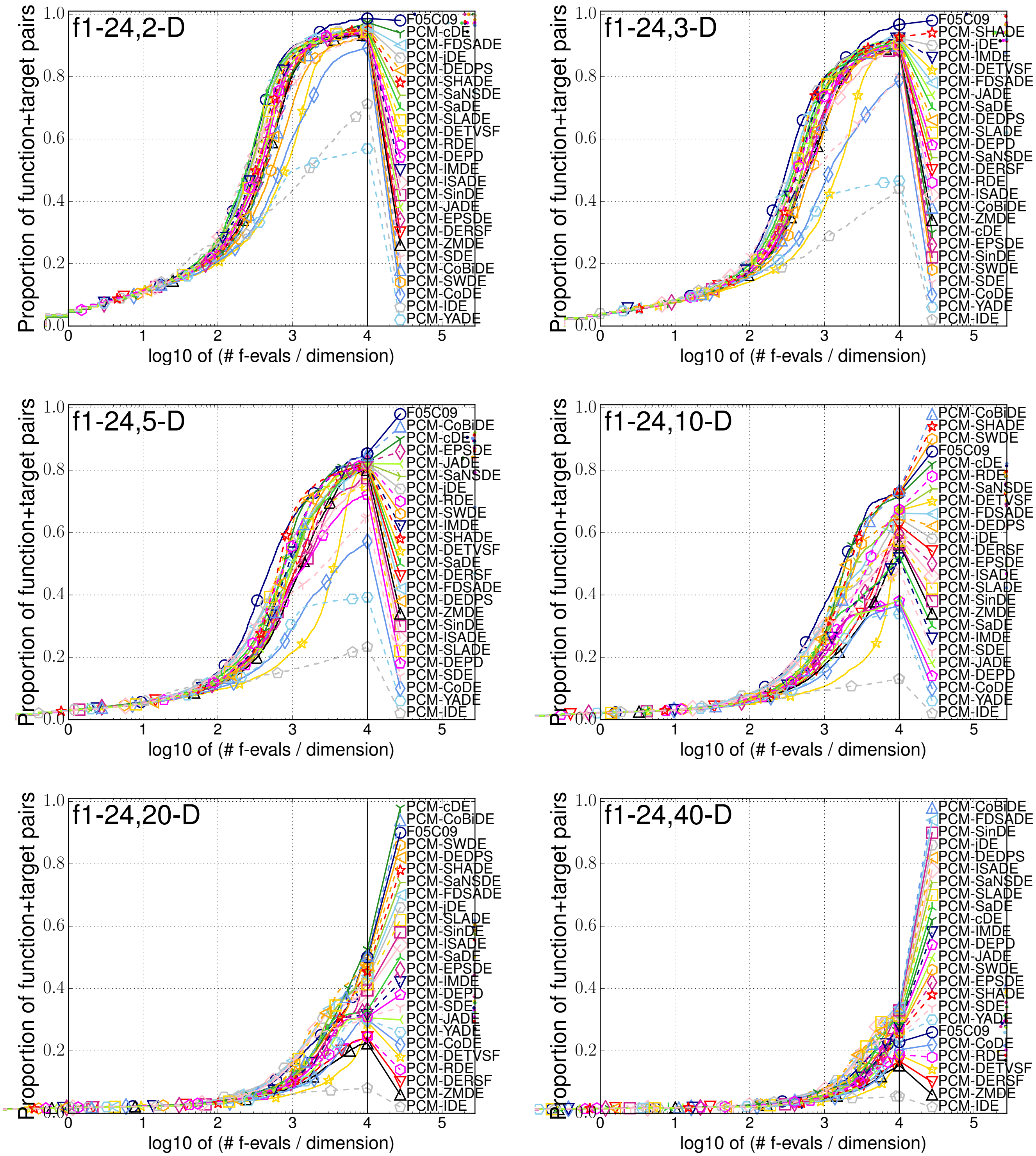

Fig. S.12: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/2/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

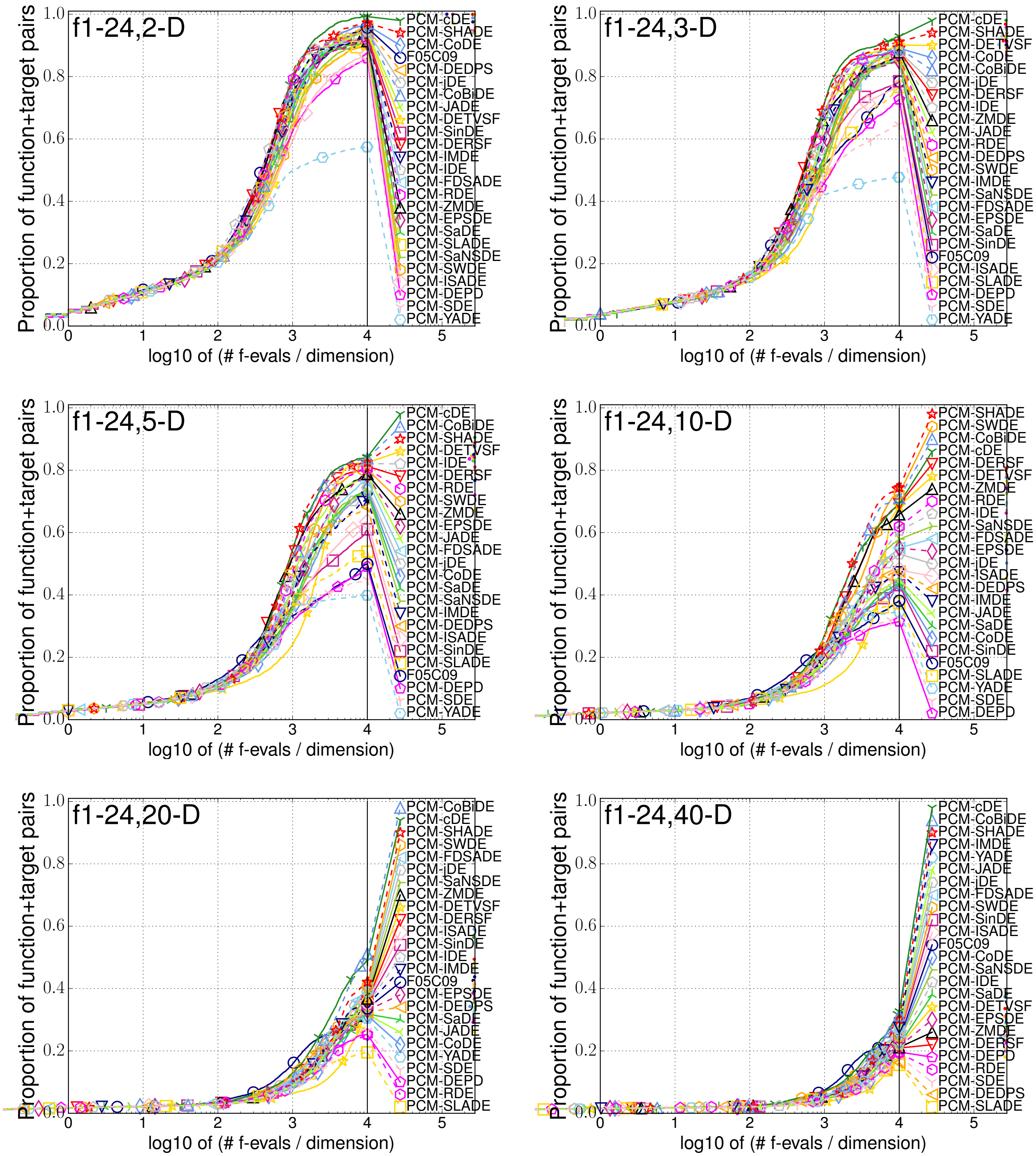

Fig. S.13: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-rand/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

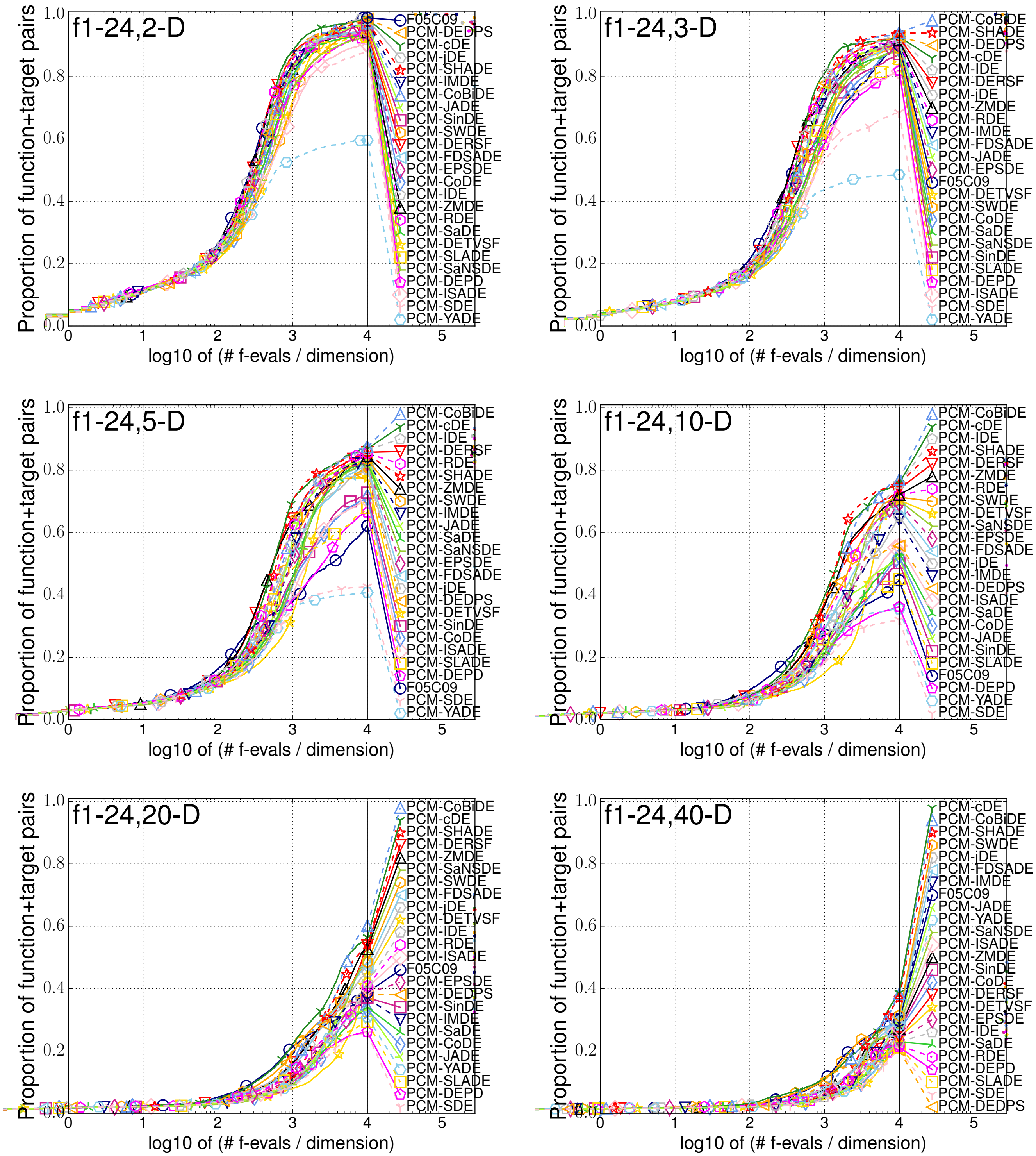

Fig. S.14: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

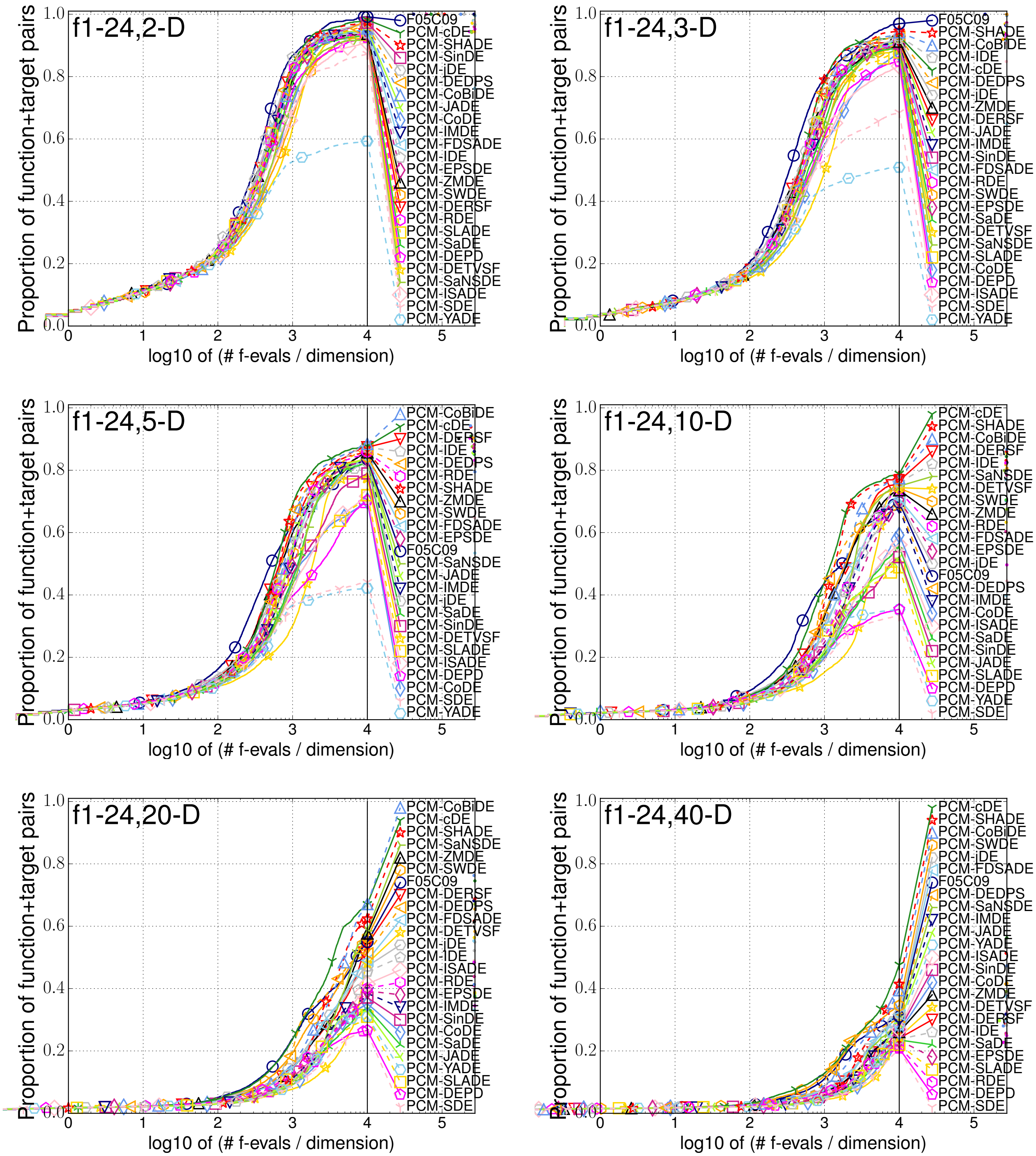

Fig. S.15: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

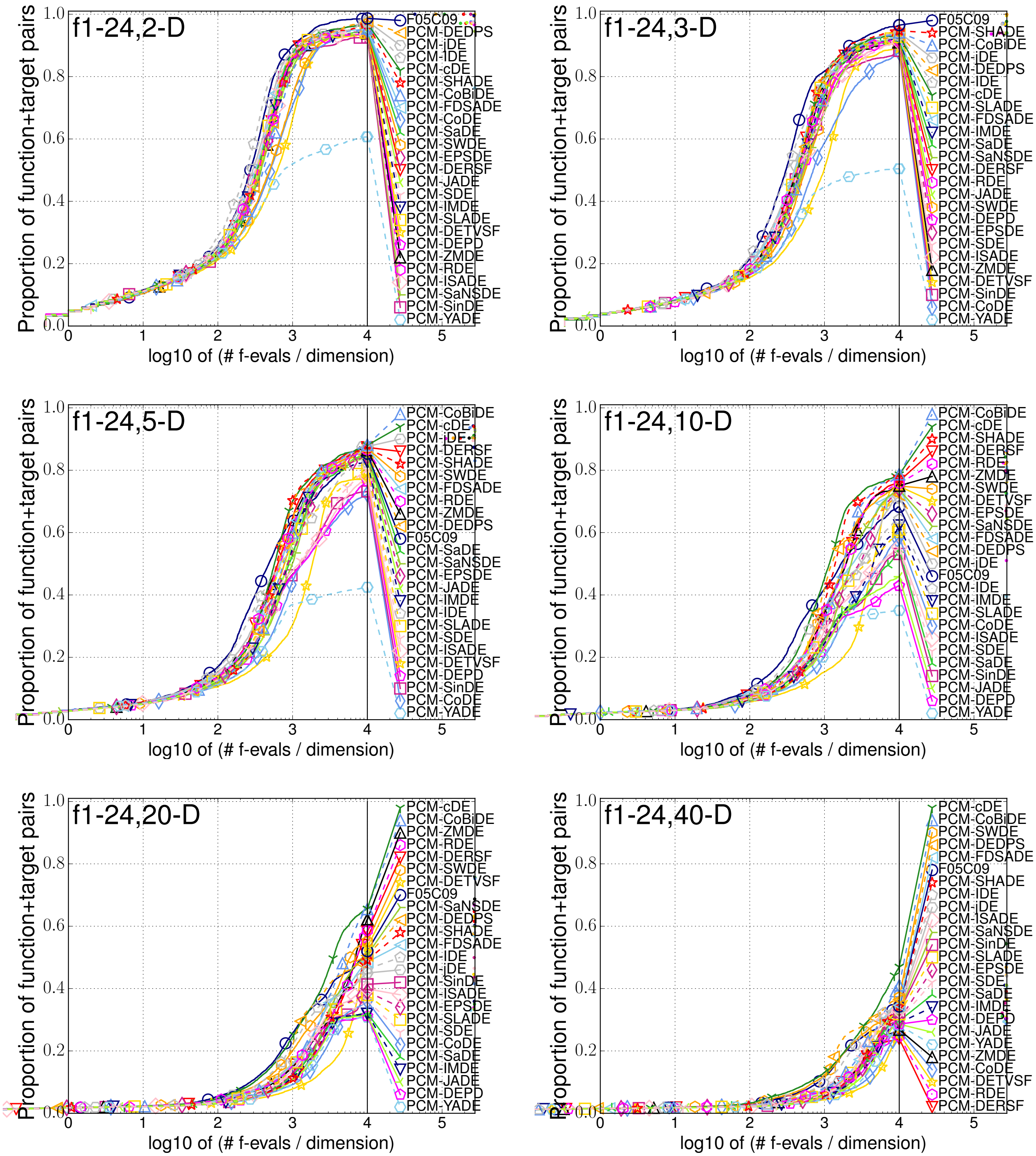

Fig. S.16: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand-to-$p$best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

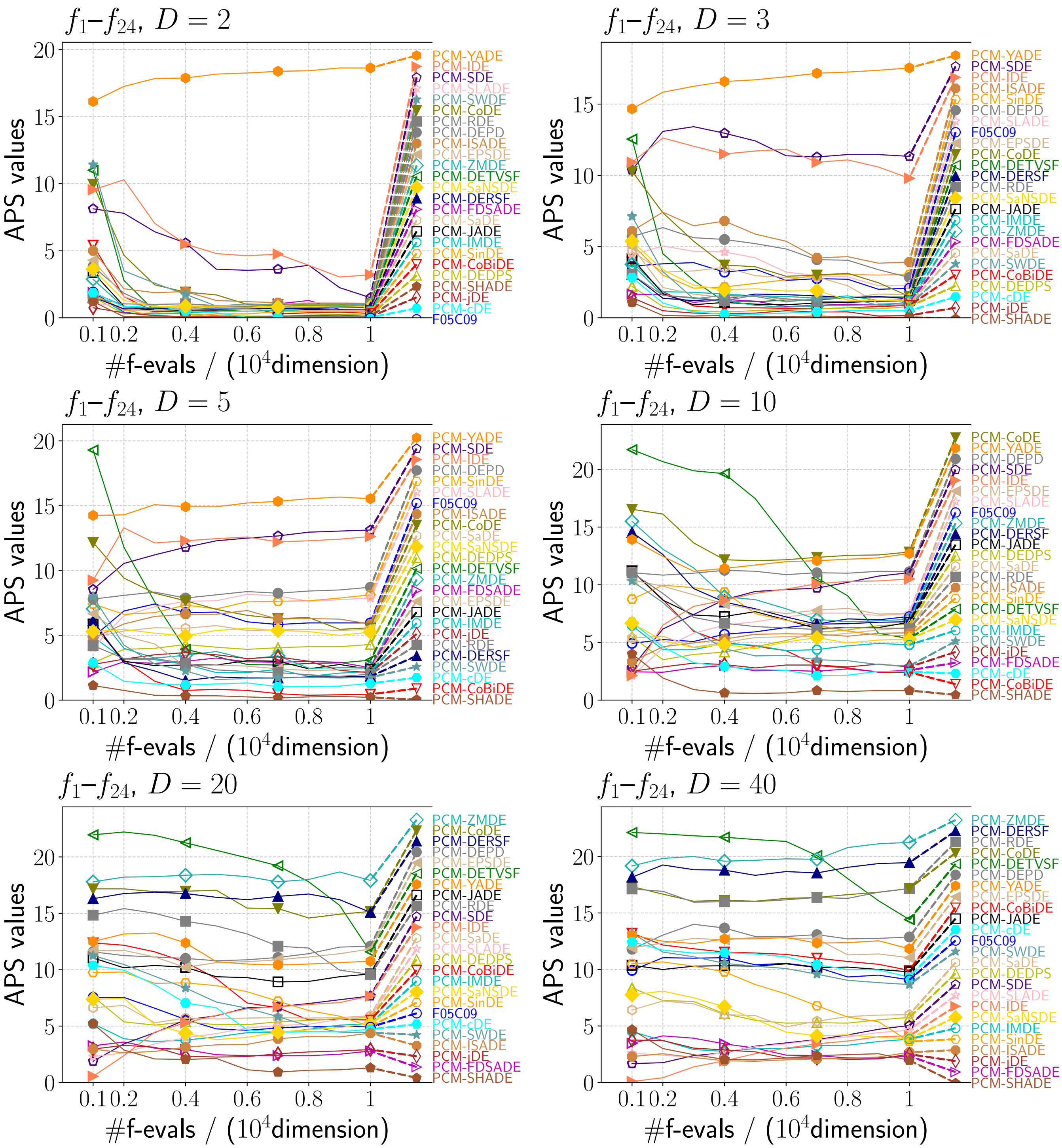

Fig. S.17: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

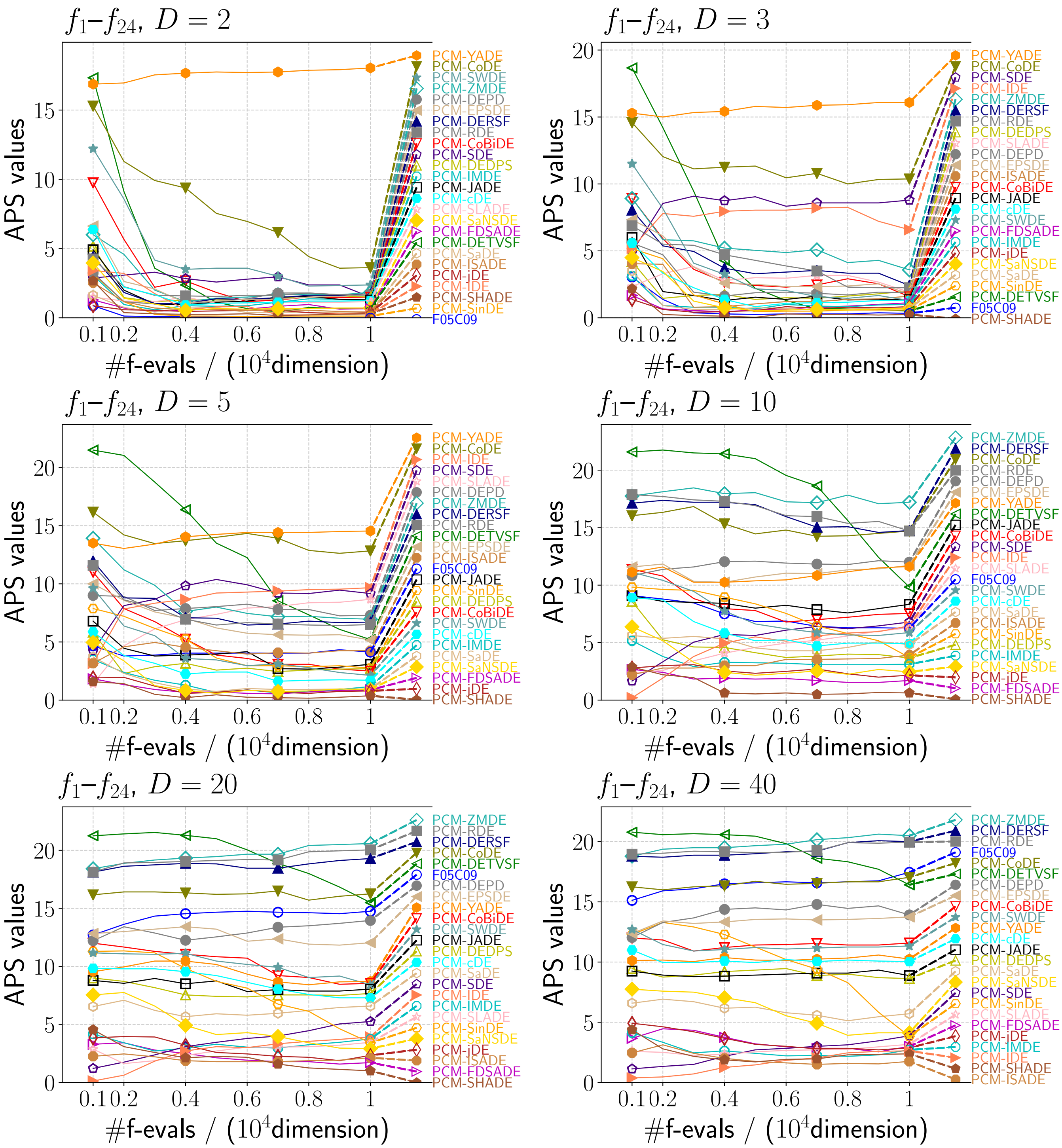

Fig. S.18: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/2/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

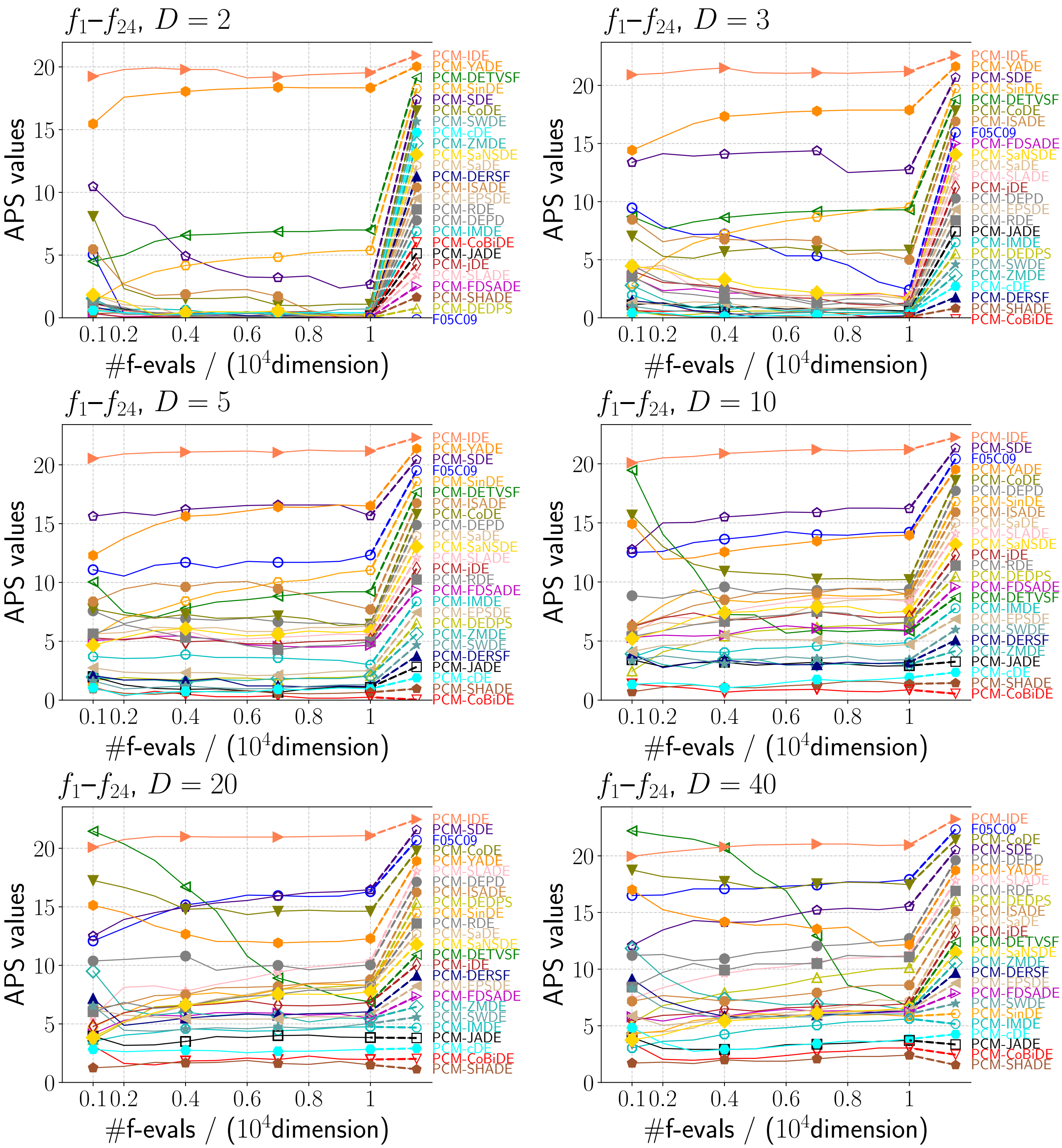

Fig. S.19: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

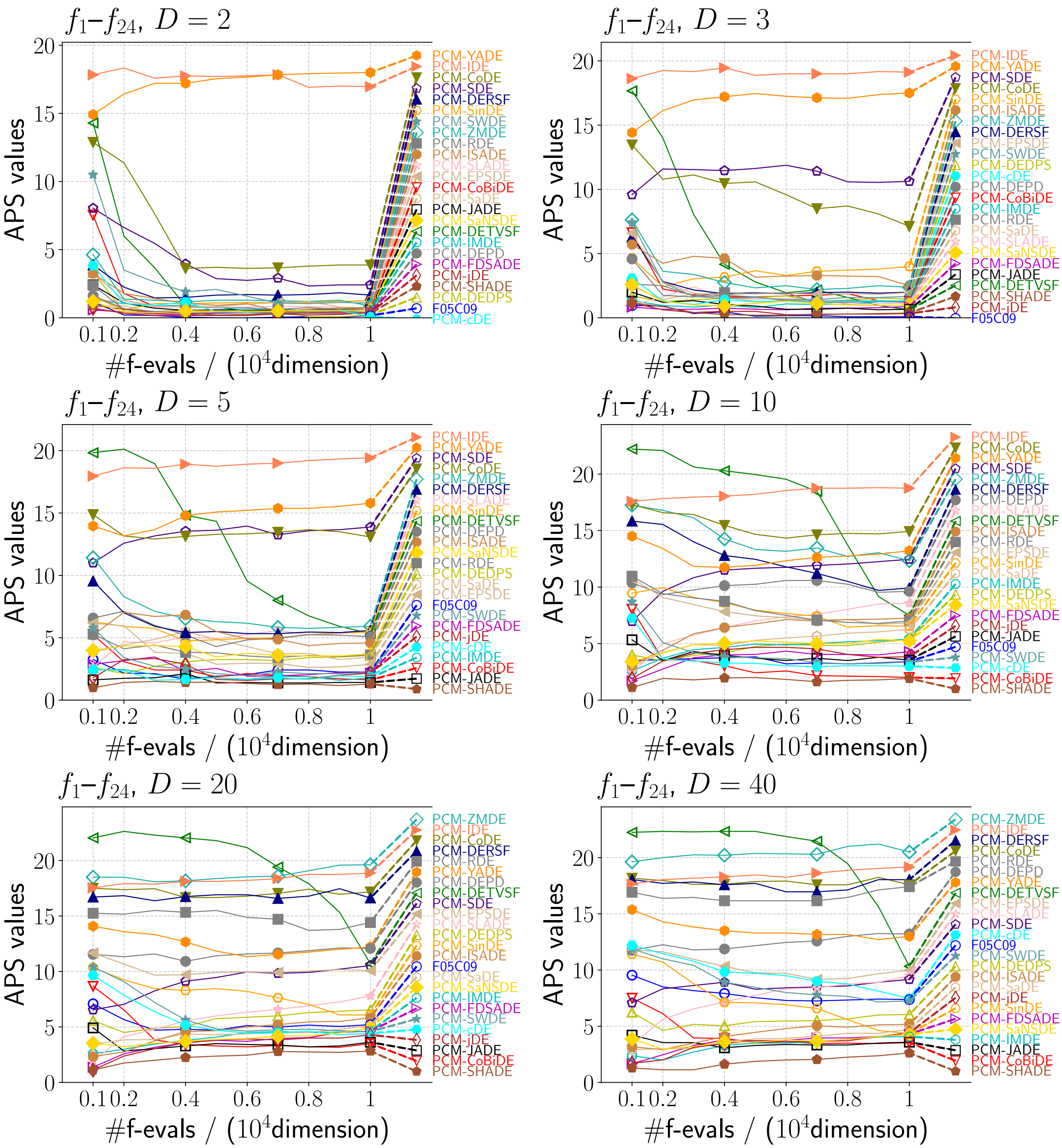

Fig. S.20: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/2/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

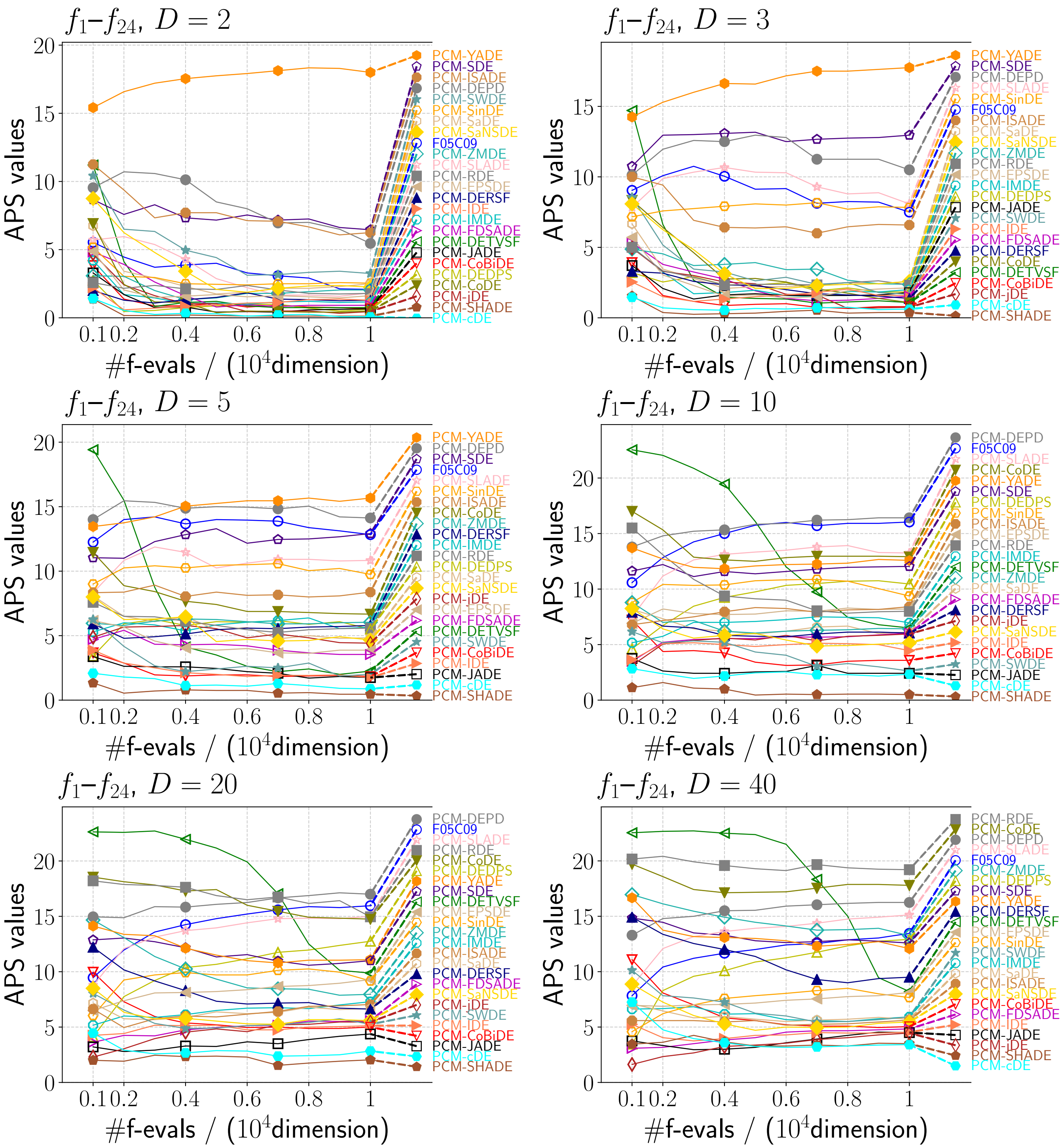

Fig. S.21: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-rand/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

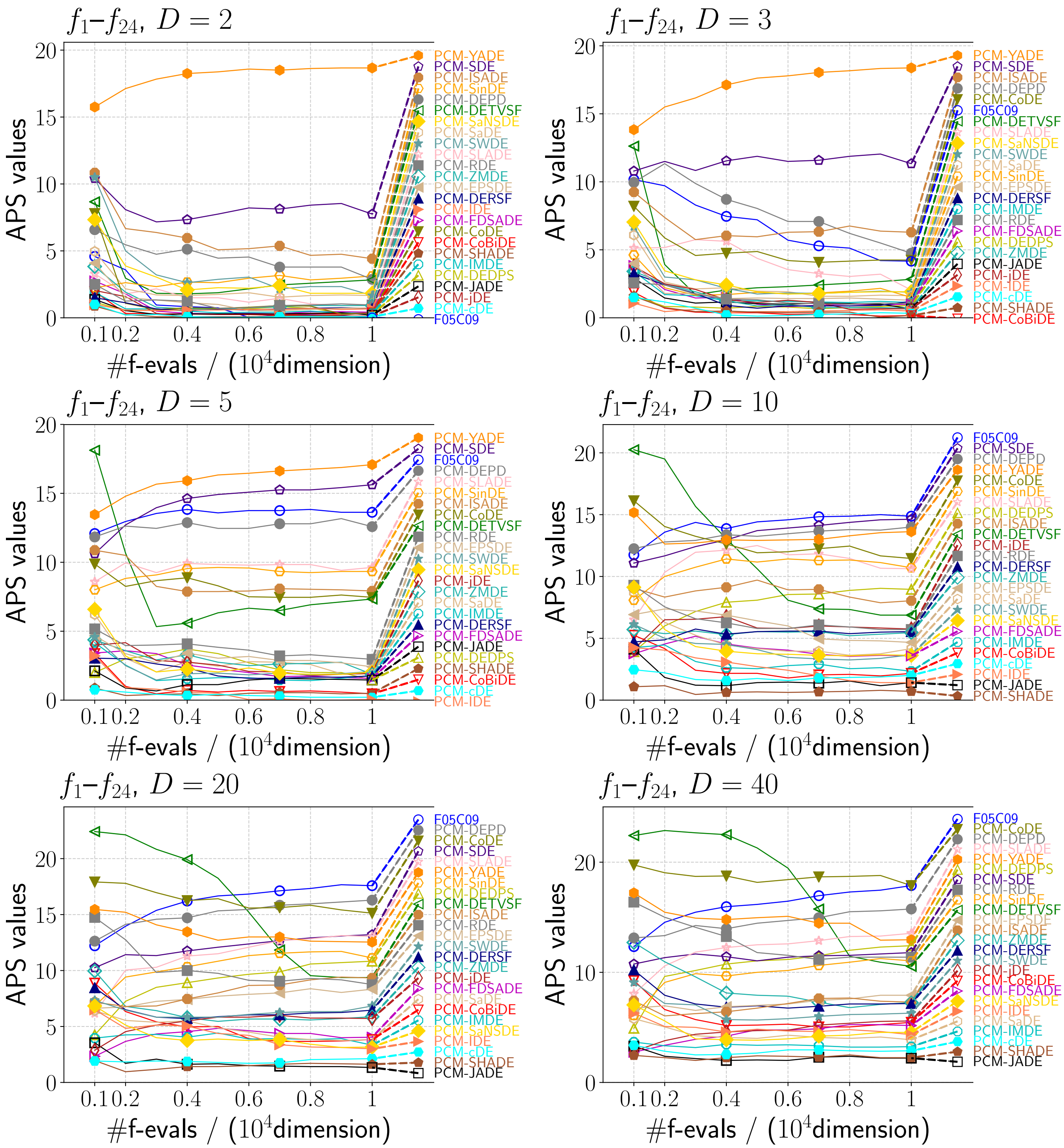

Fig. S.22: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

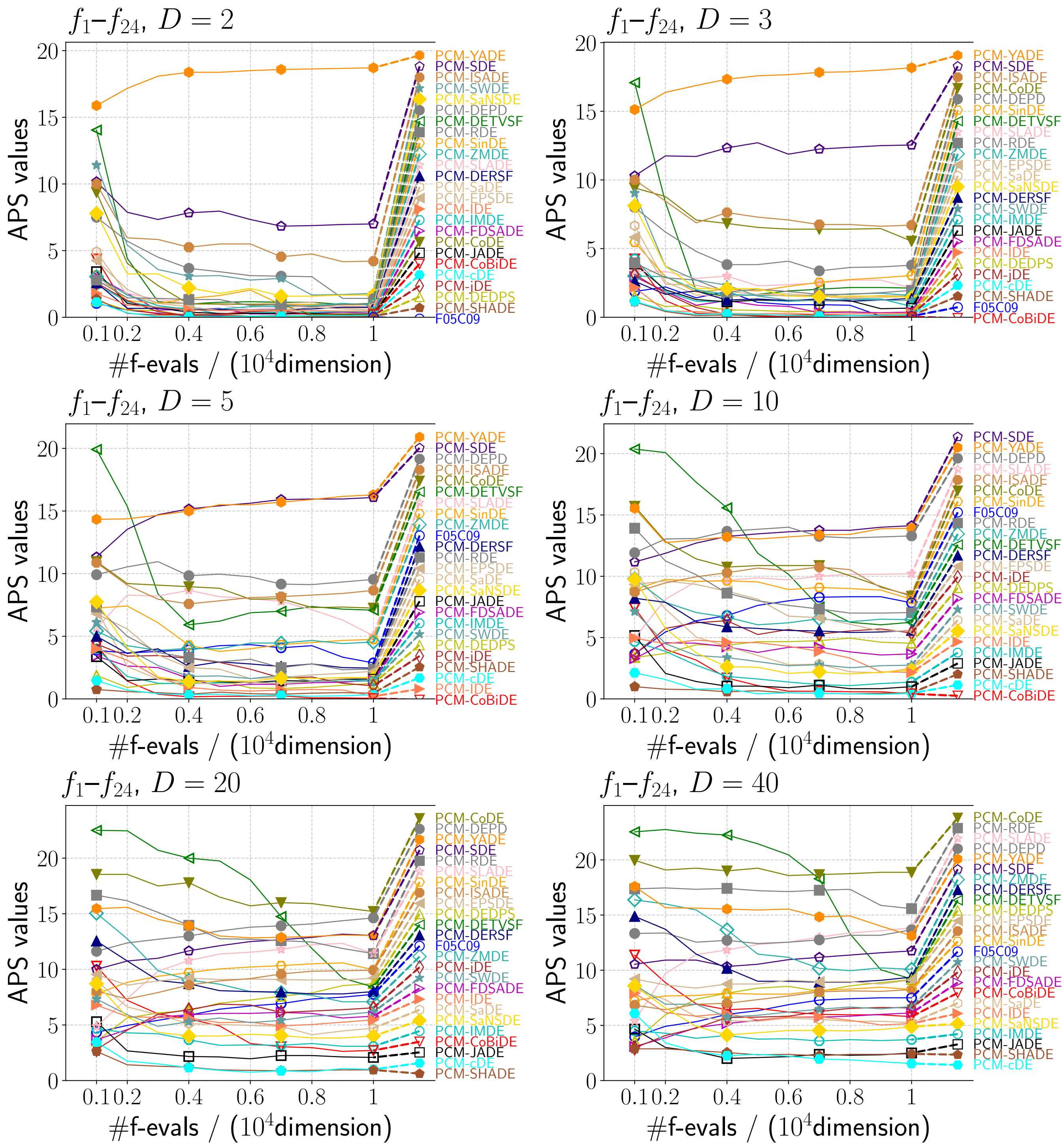

Fig. S.23: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

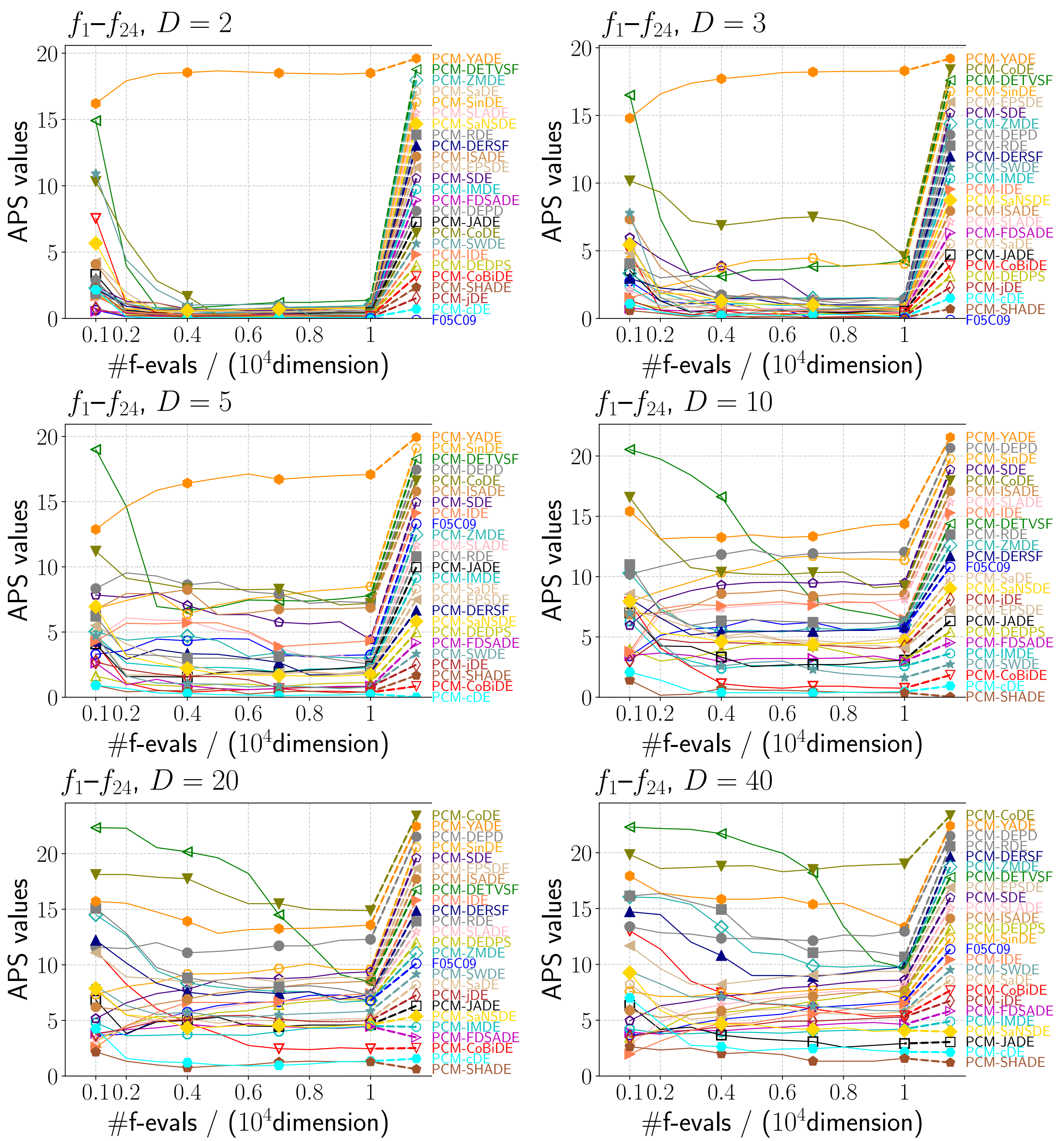

Fig. S.24: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand-to-$p$best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

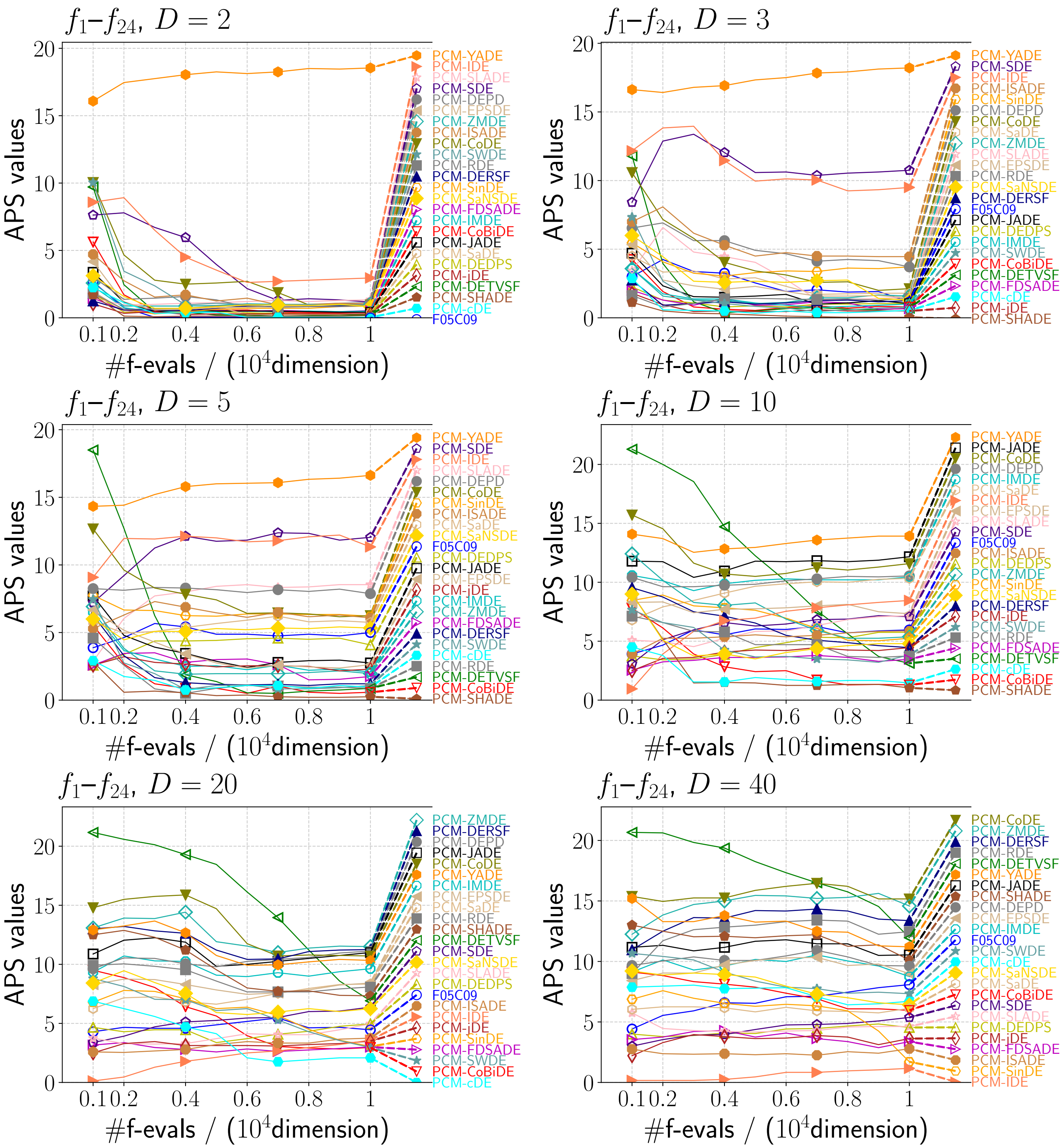

Fig. S.25: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

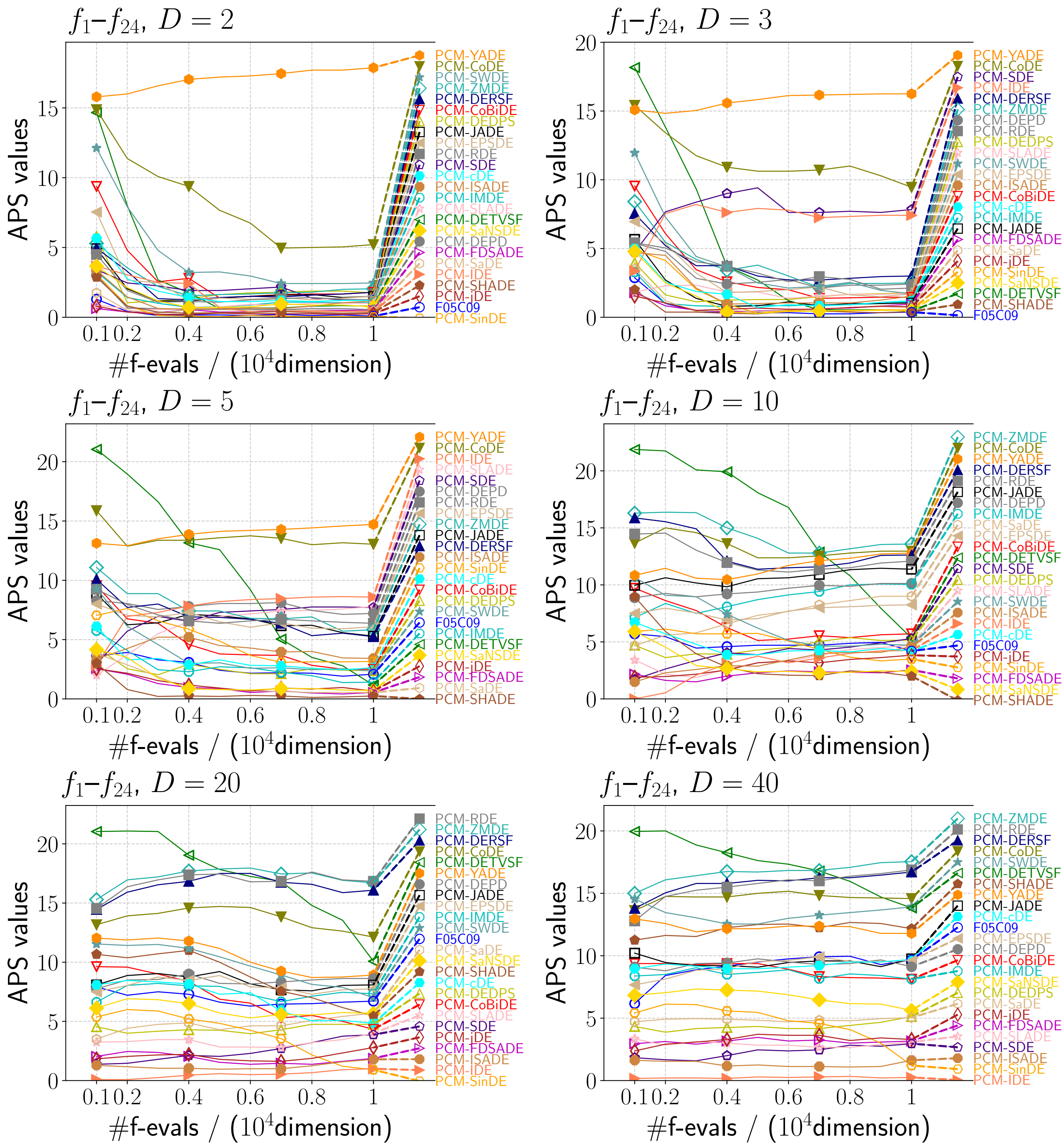

Fig. S.26: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/2/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

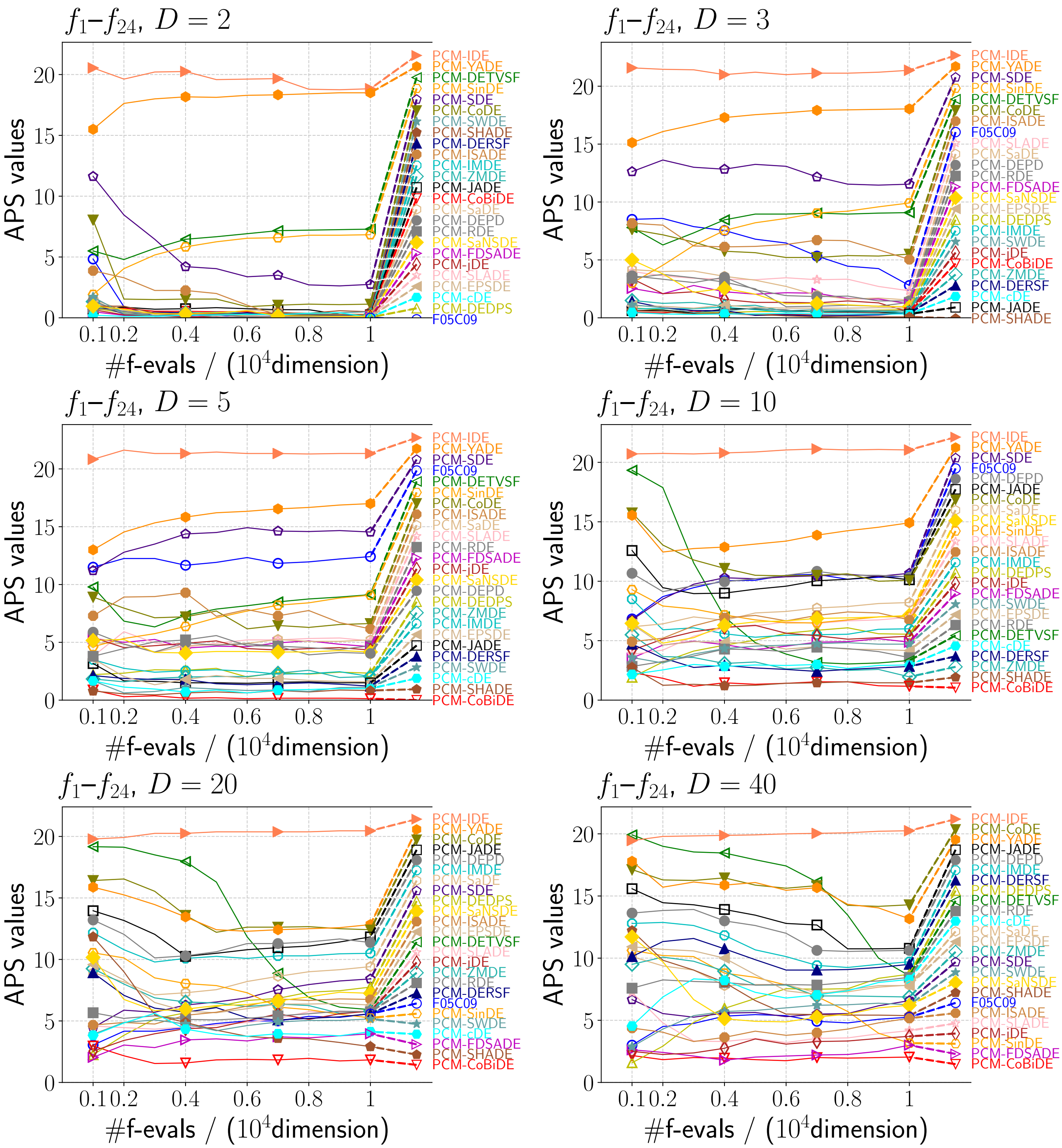

Fig. S.27: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

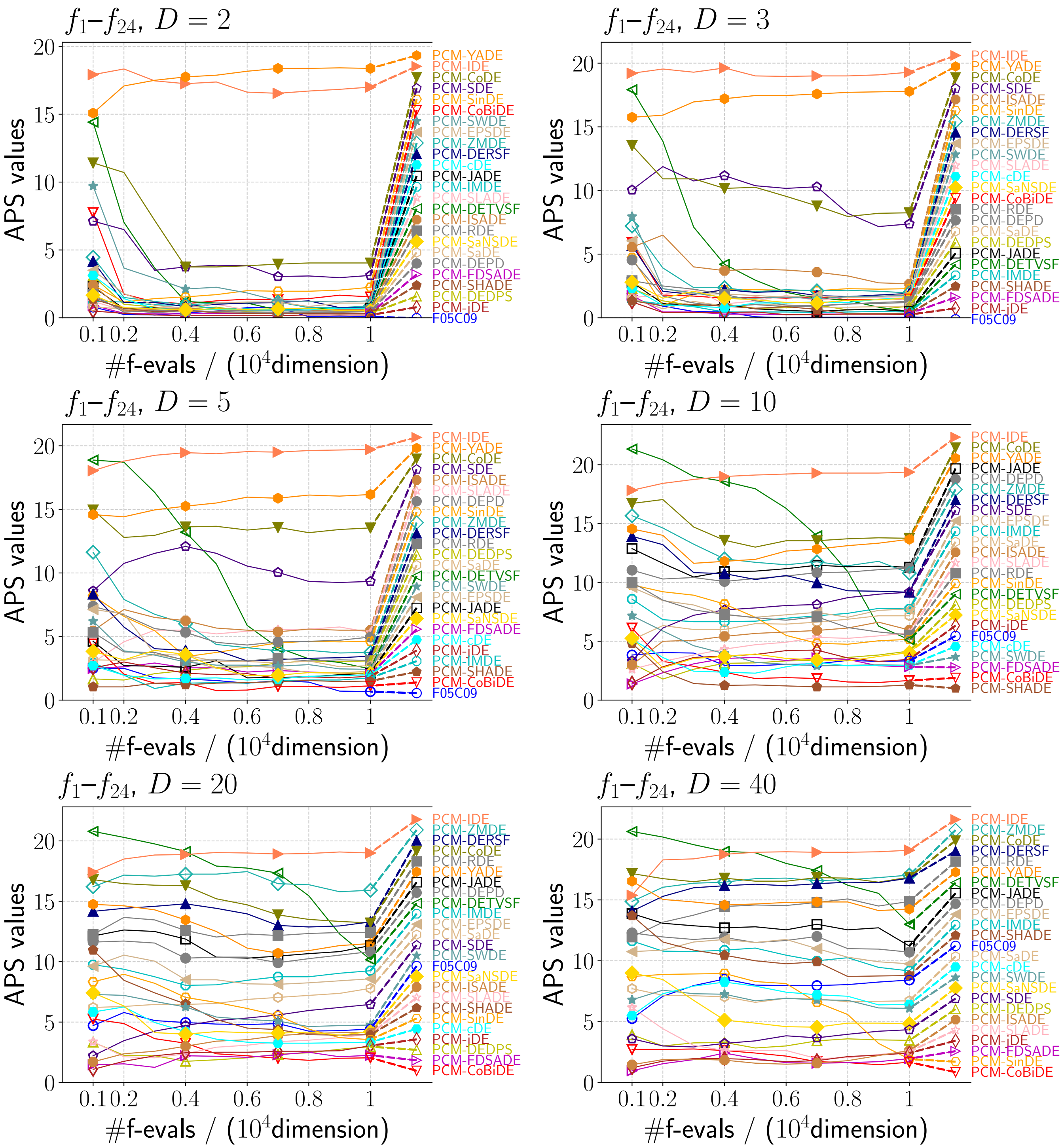

Fig. S.28: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the best/2/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

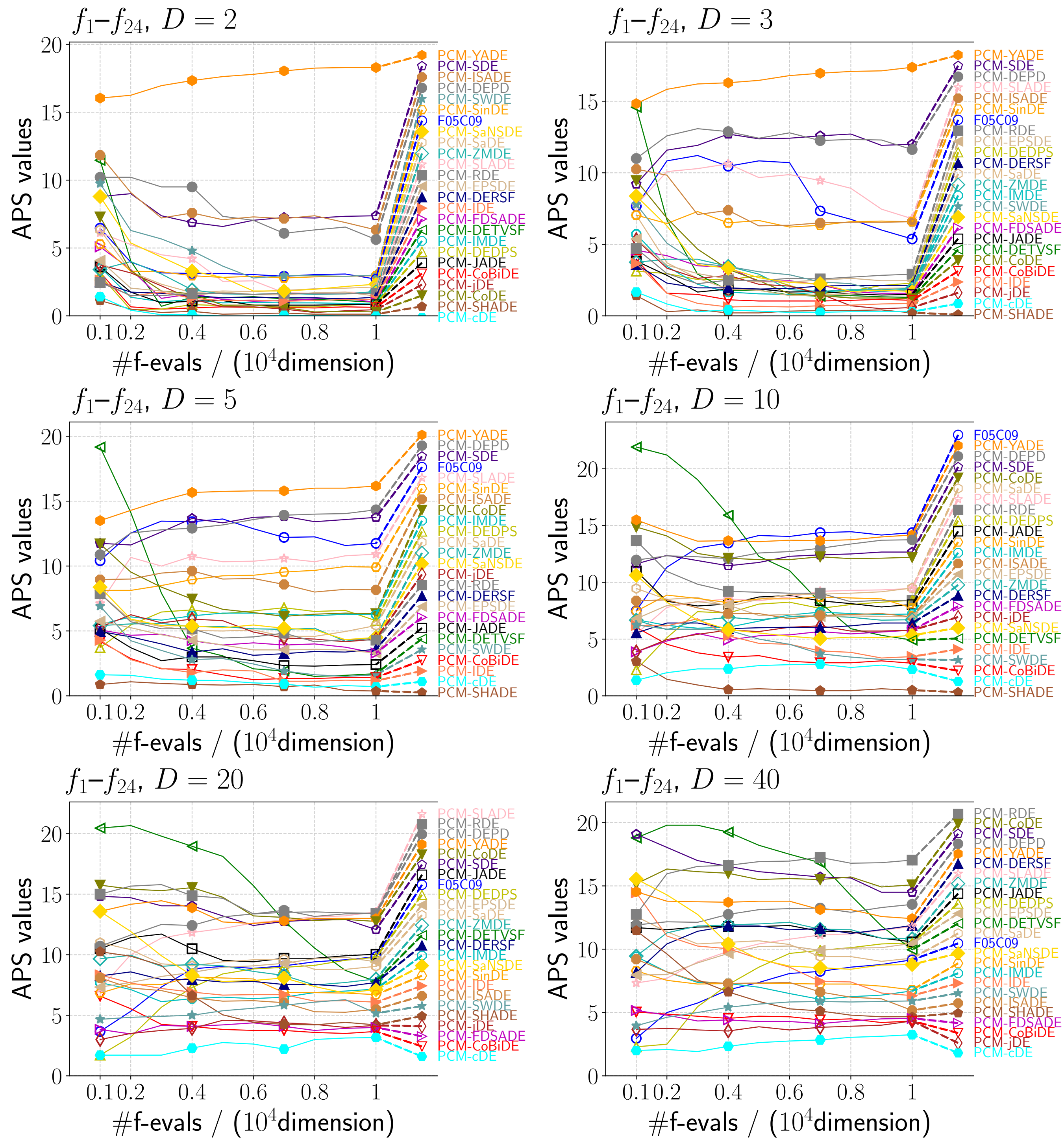

Fig. S.29: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-rand/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

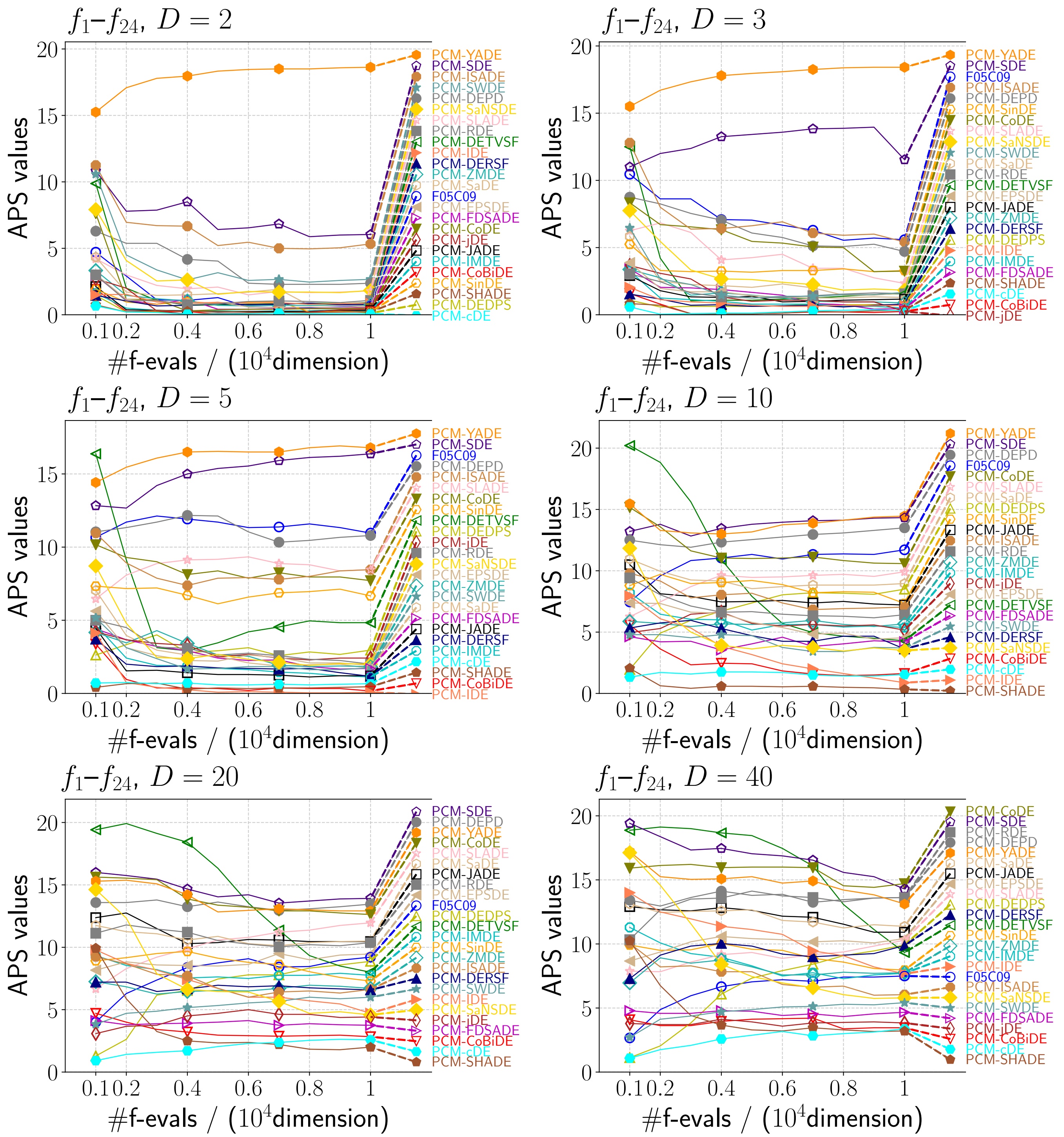

Fig. S.30: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

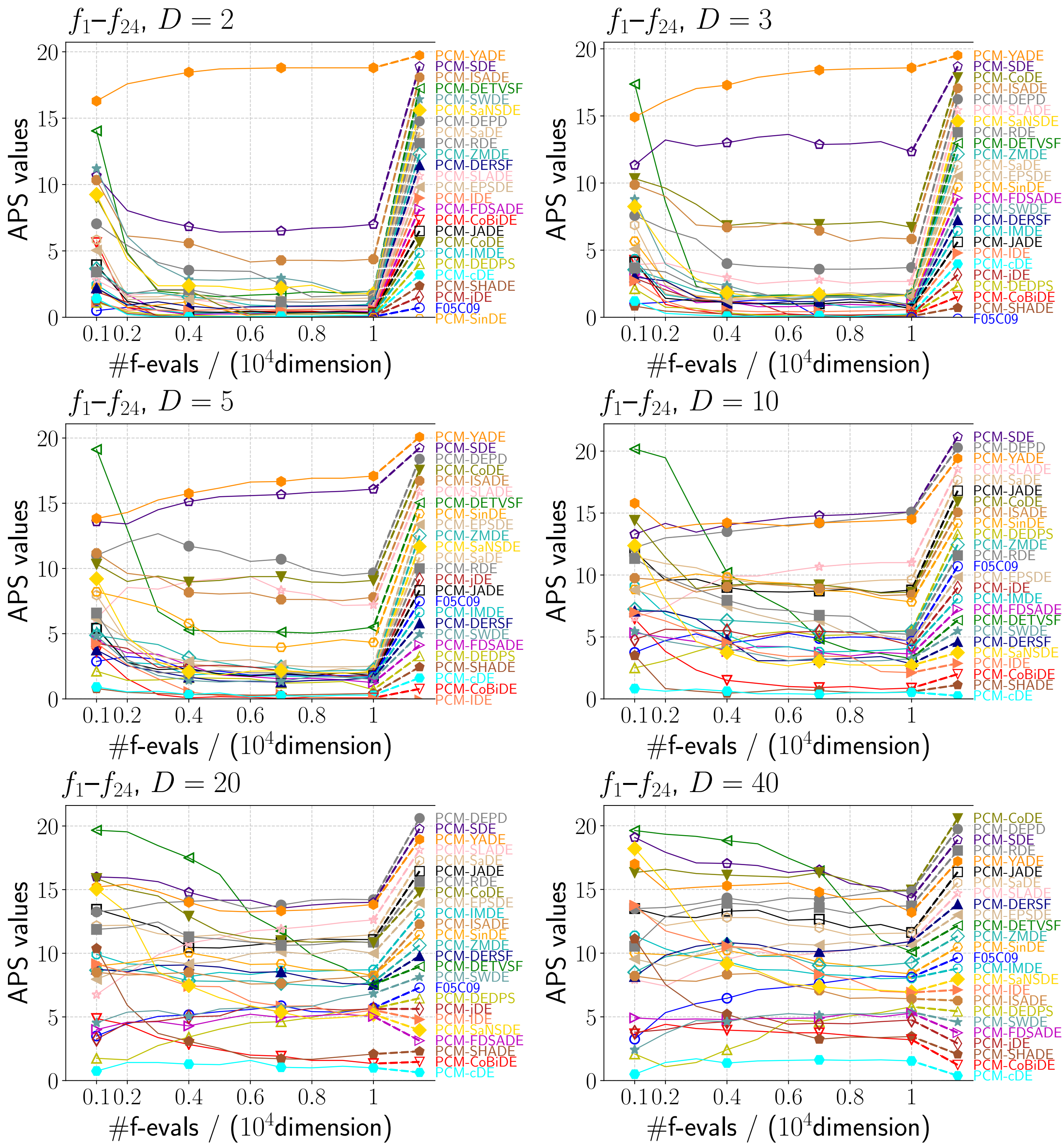

Fig. S.31: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

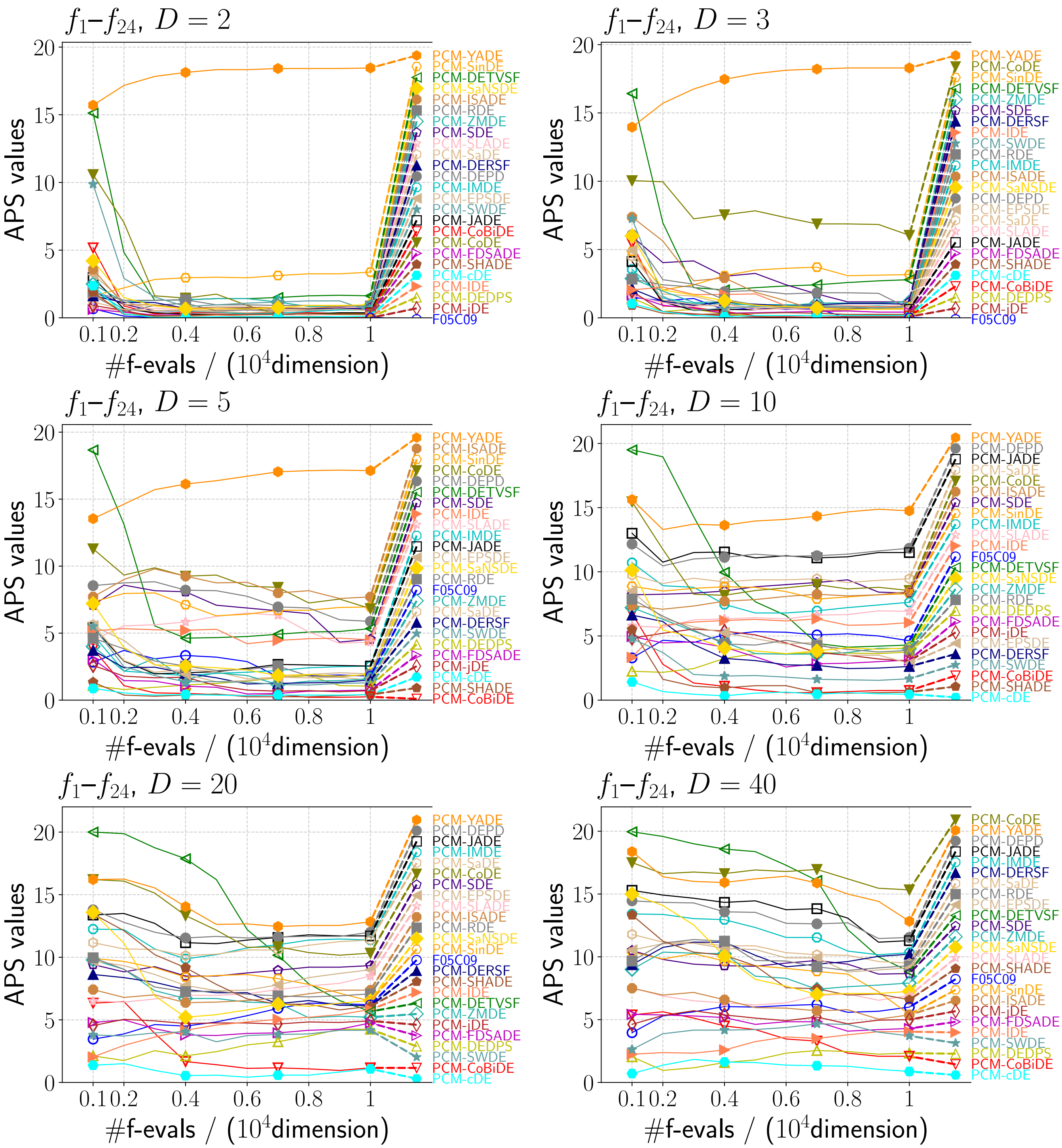

Fig. S.32: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand-to-$p$best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

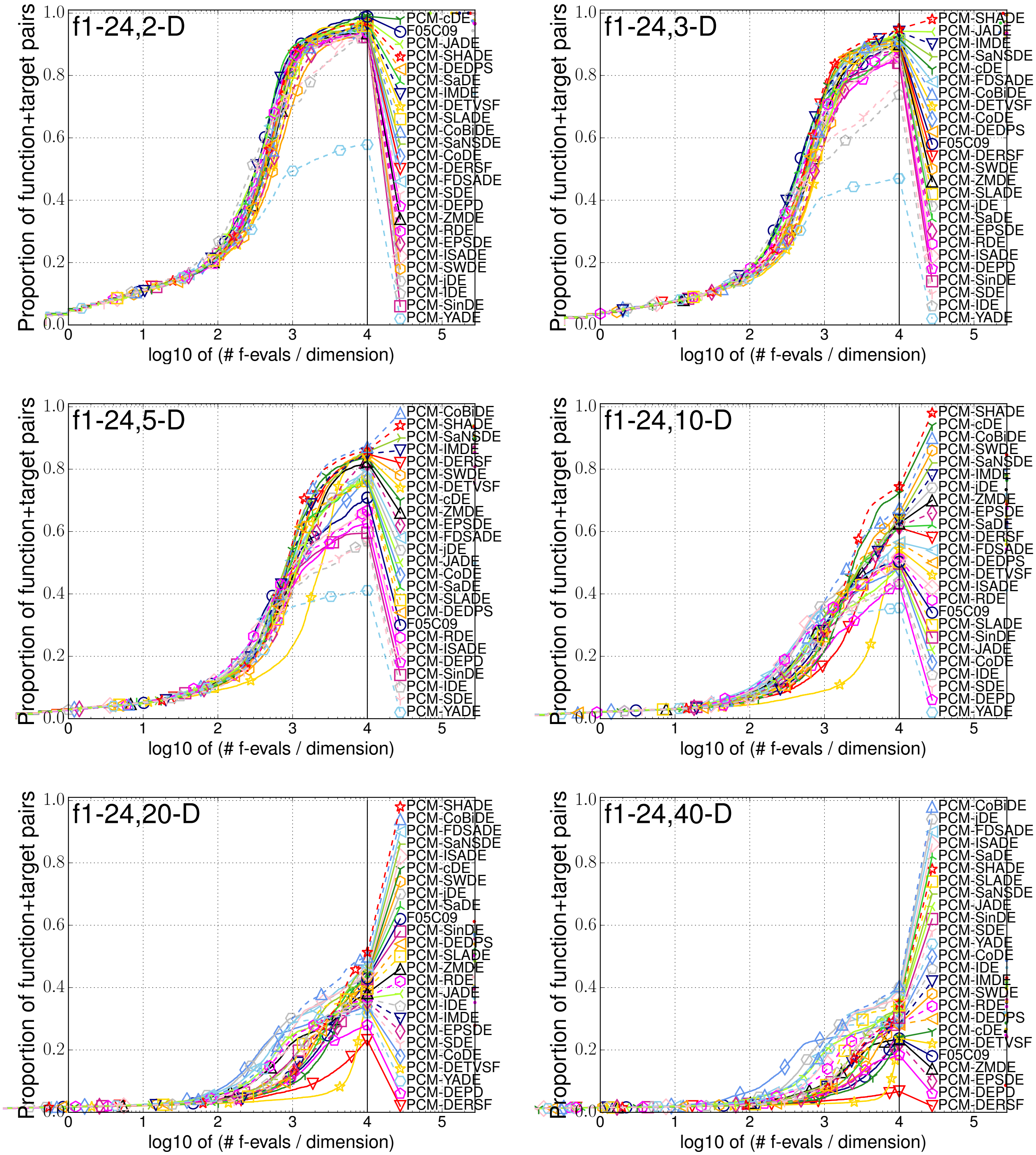

Fig. S.33: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

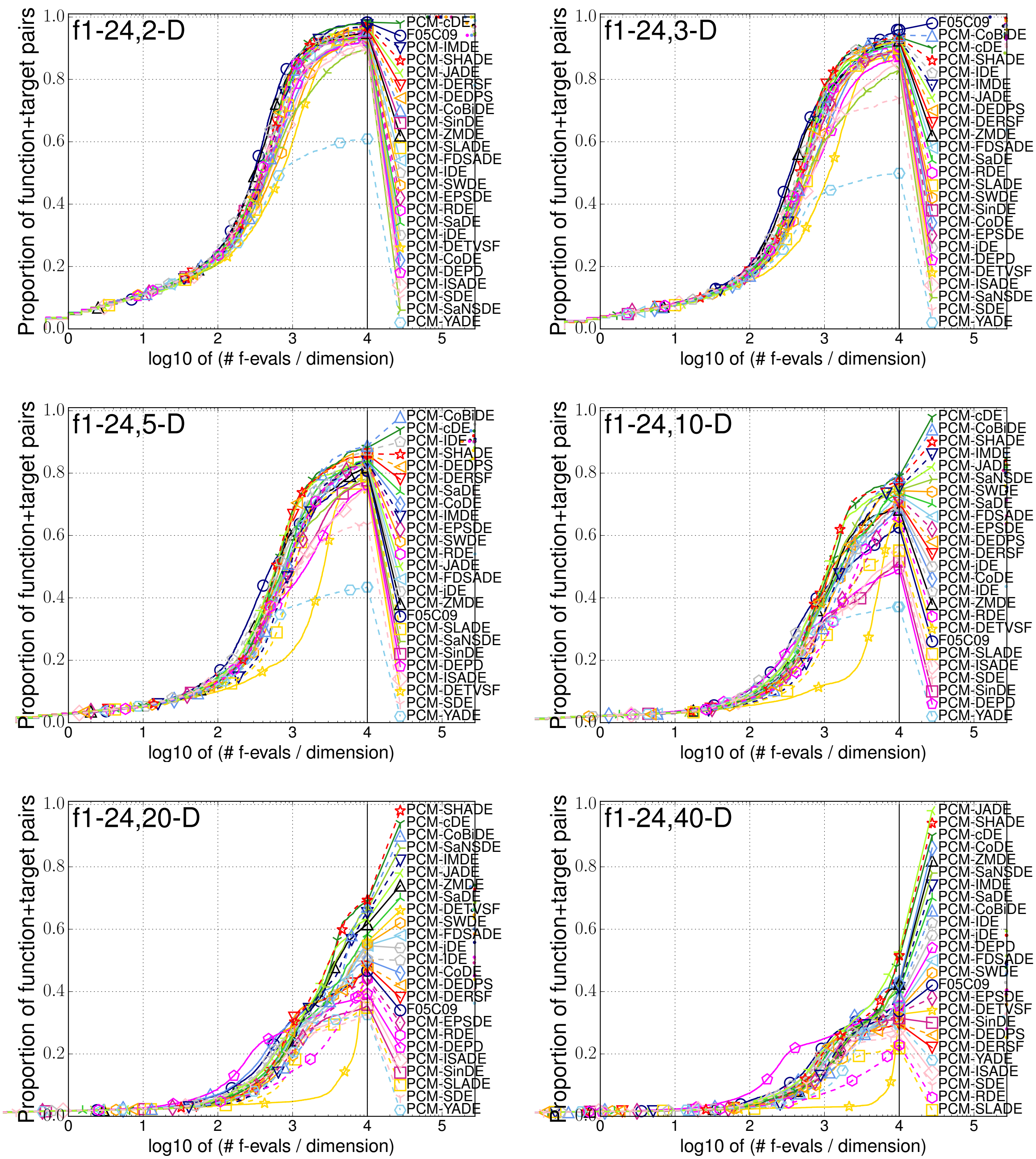

Fig. S.34: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

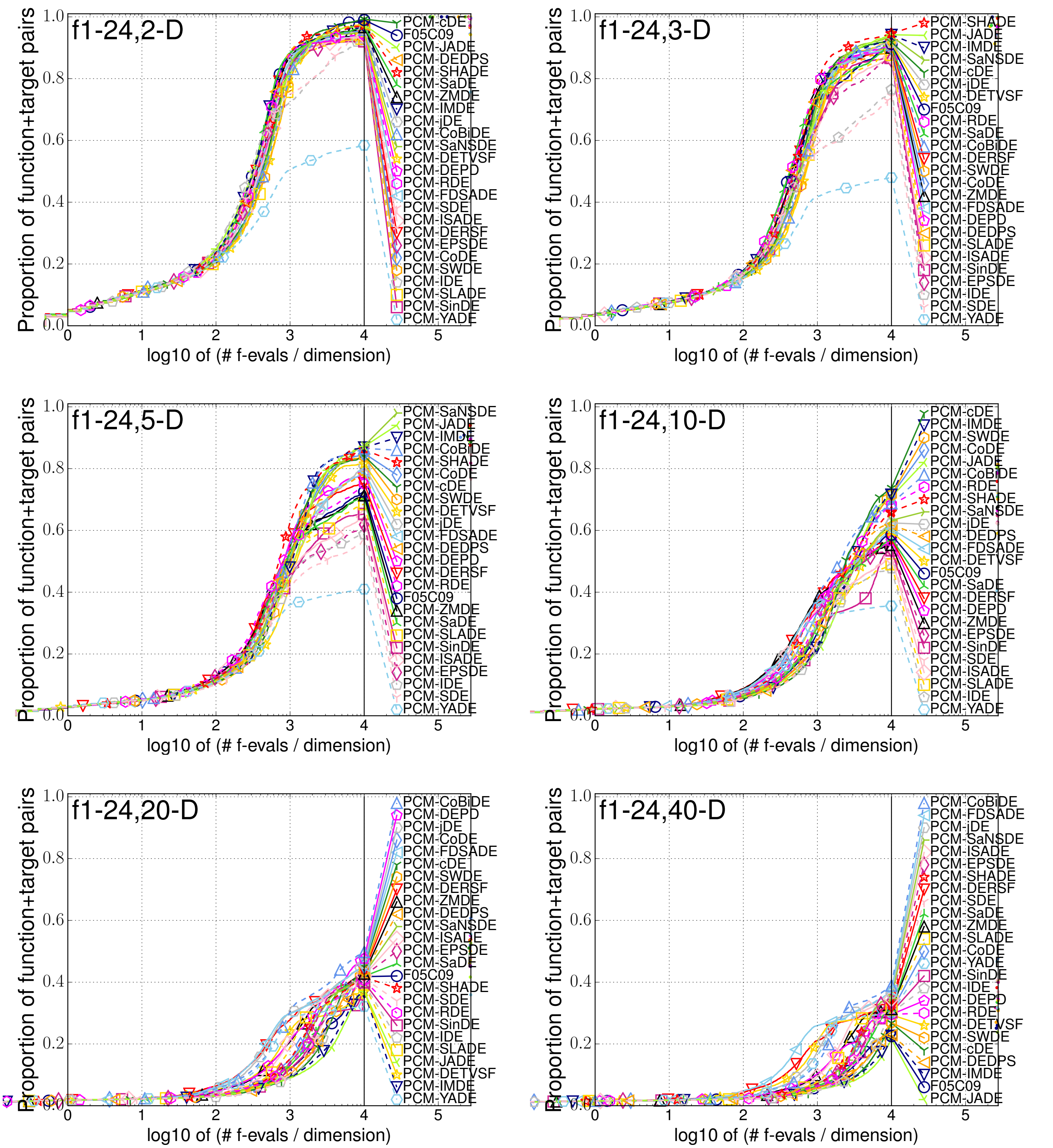

Fig. S.35: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

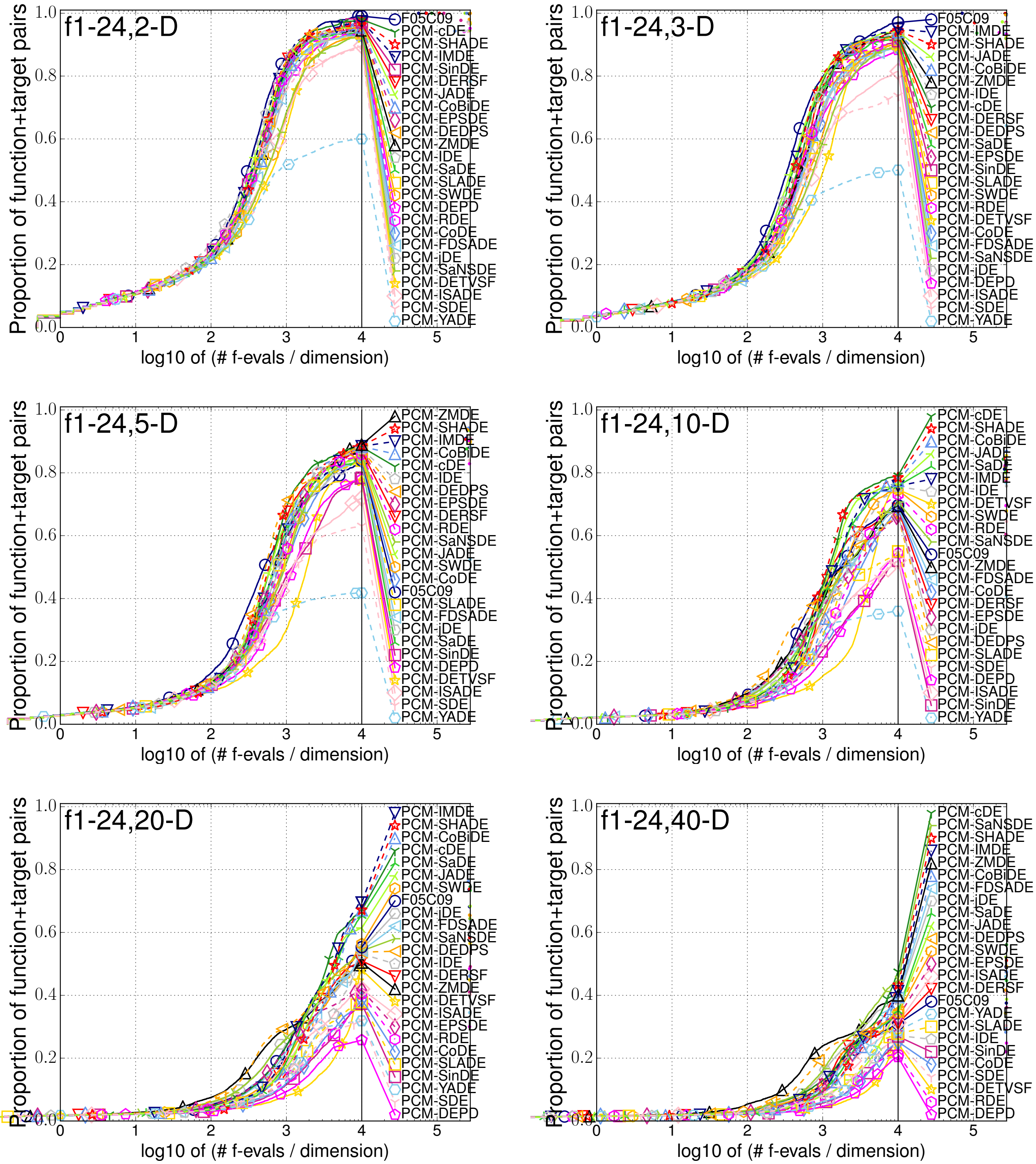

Fig. S.36: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For how to read these ECDF figures, see the caption of Fig. 1 in the main paper.

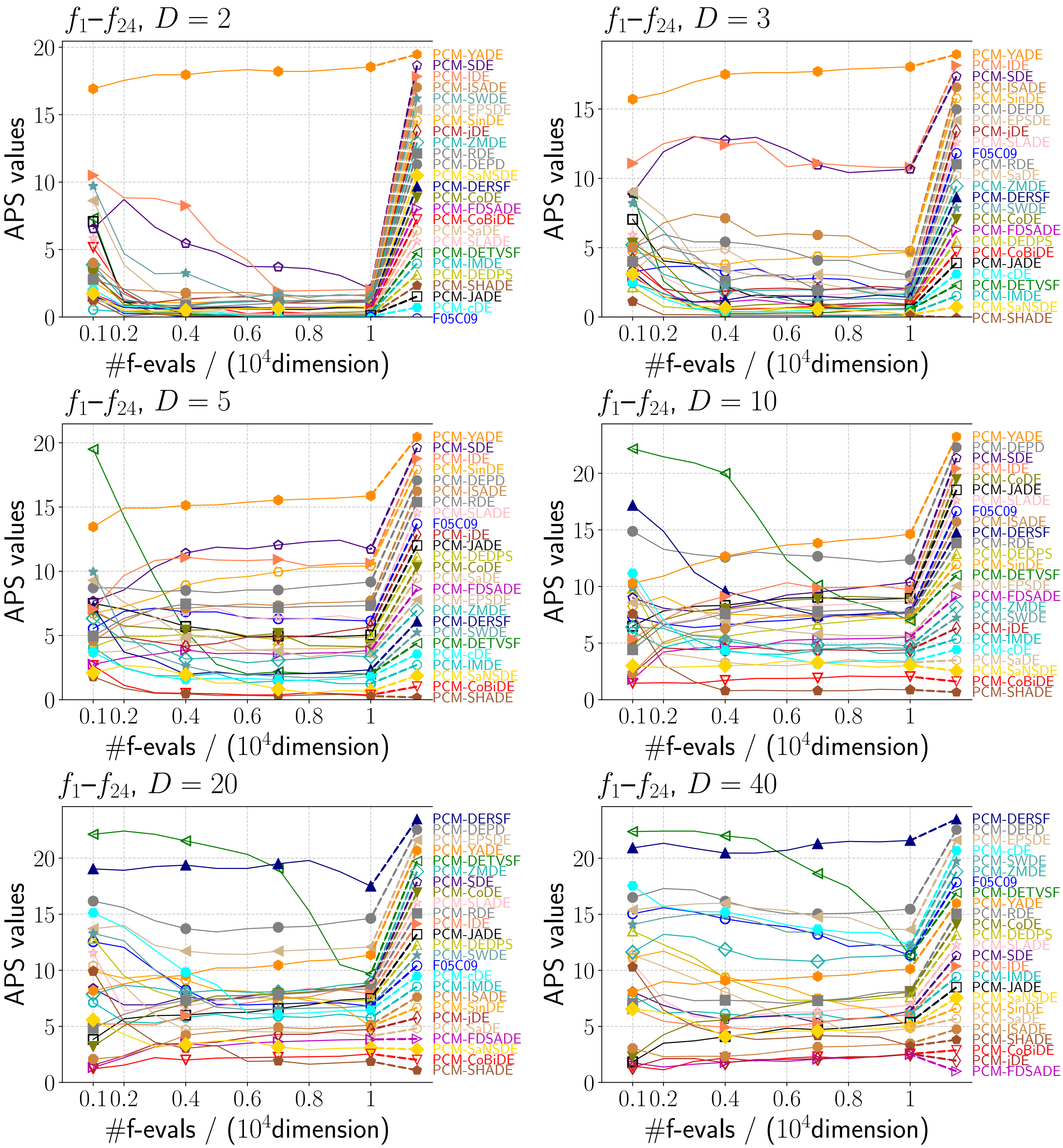

Fig. S.37: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

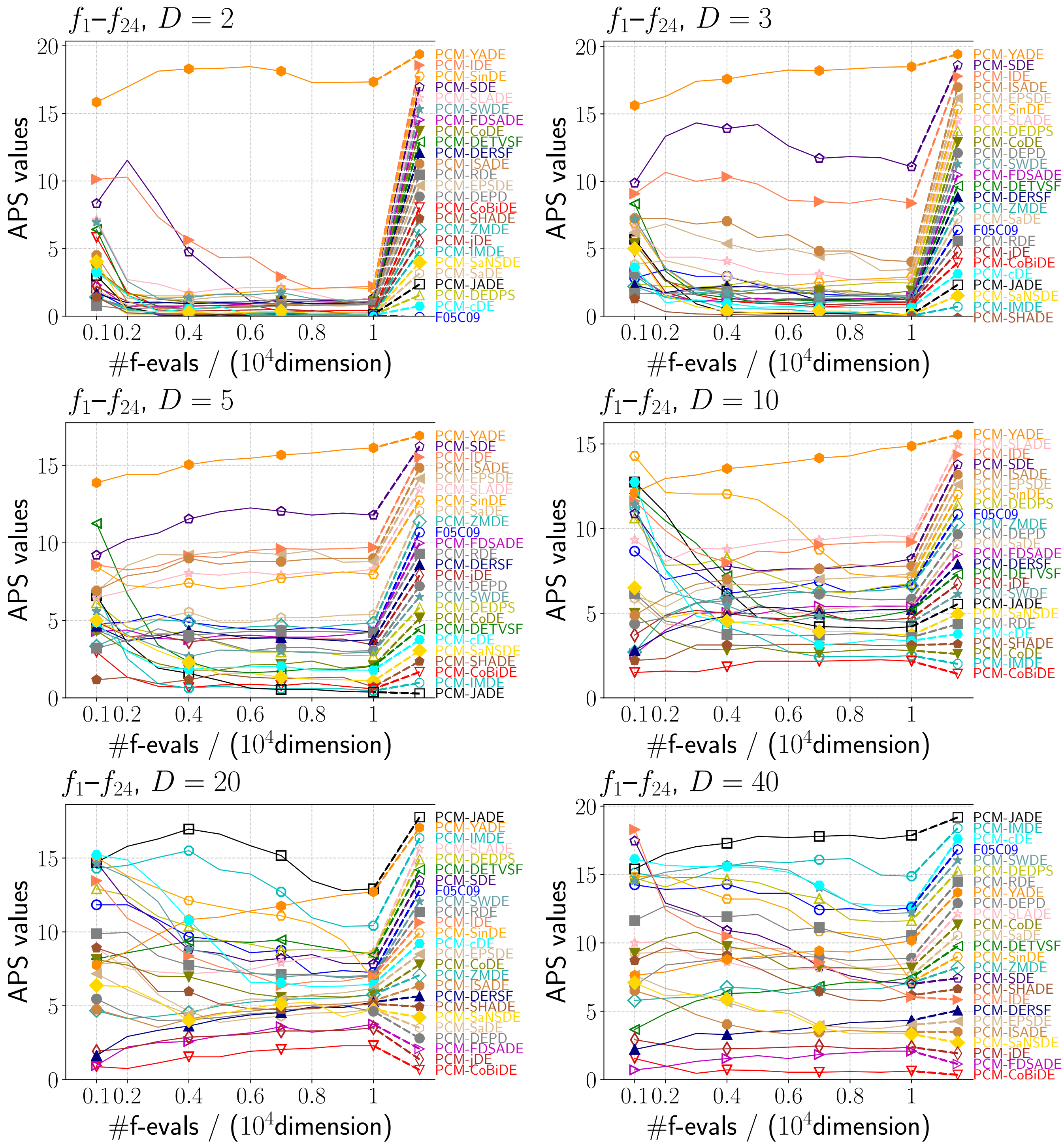

Fig. S.38: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the rand/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

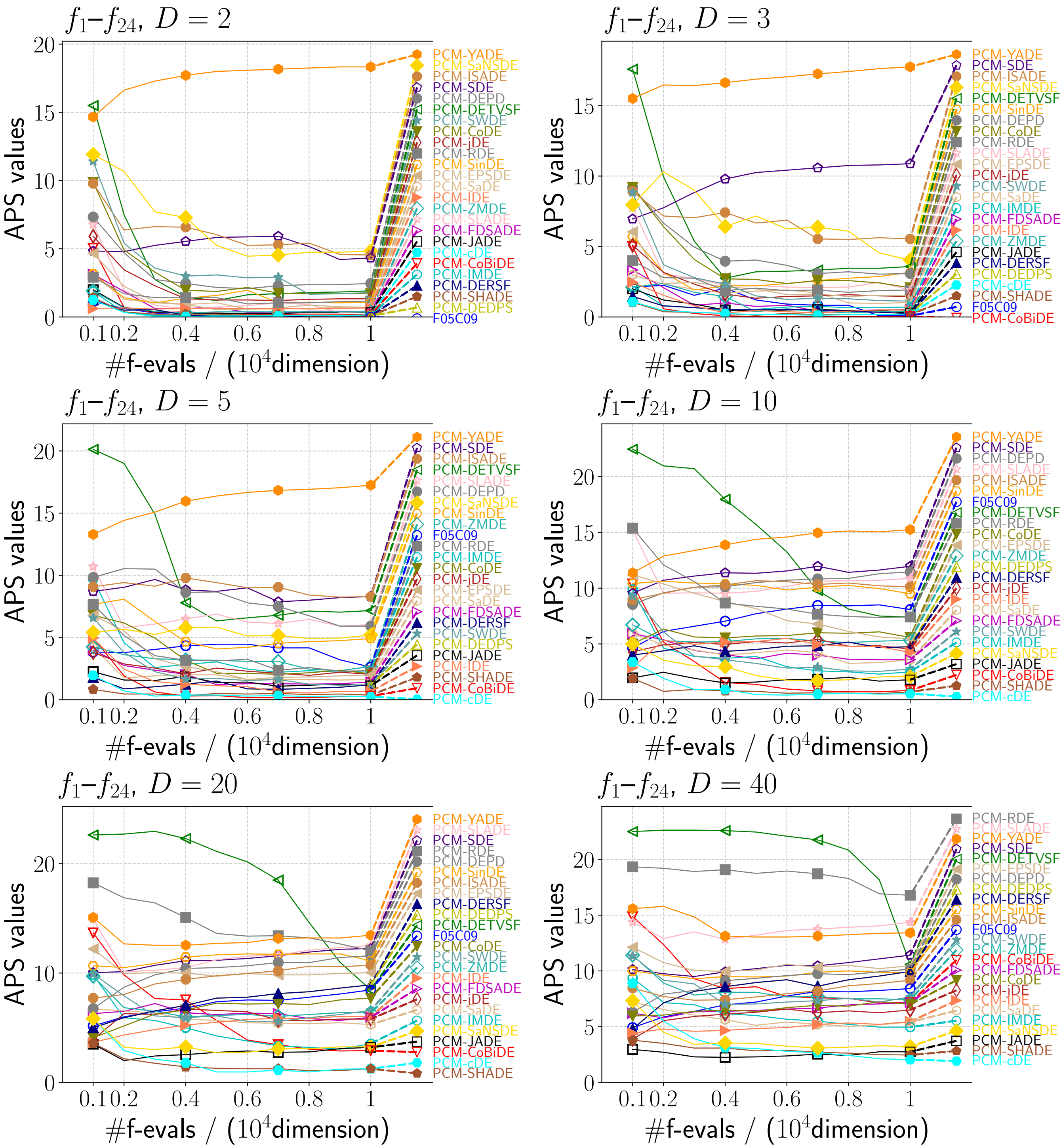

Fig. S.39: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/bin operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

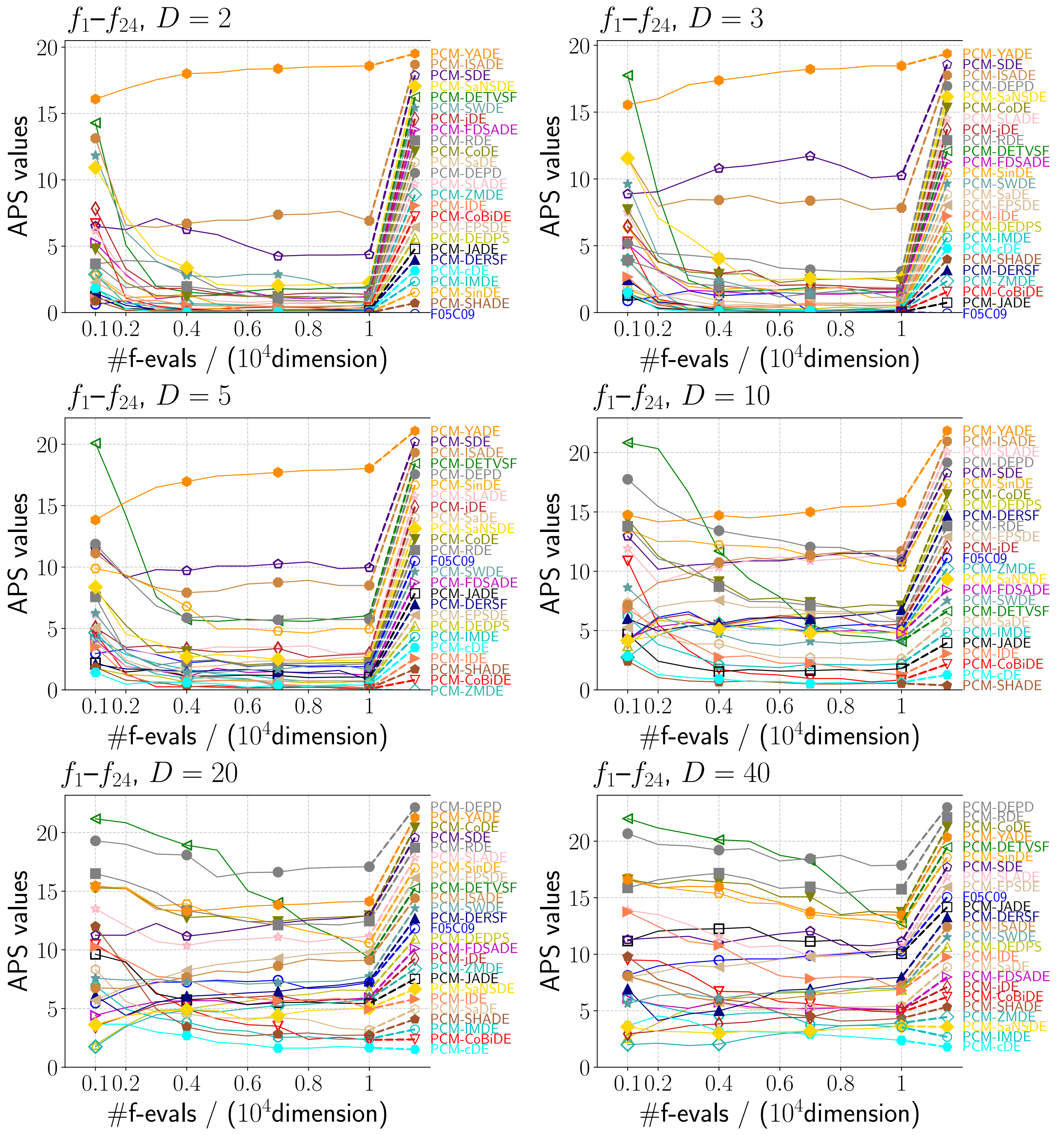

Fig. S.40: Comparisons of the 25 methods (the 24 PCMs and $F05C09$) using the current-to-$p$best/1/sec operator on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20, 40\}$). Tuned hyper parameter settings are used for each method. These figures show the Average Performance Score (APS) [4] values for all functions (lower is better). The Wilcoxon rank-sum test with $p < 0.05$ was used for the APS calculation.

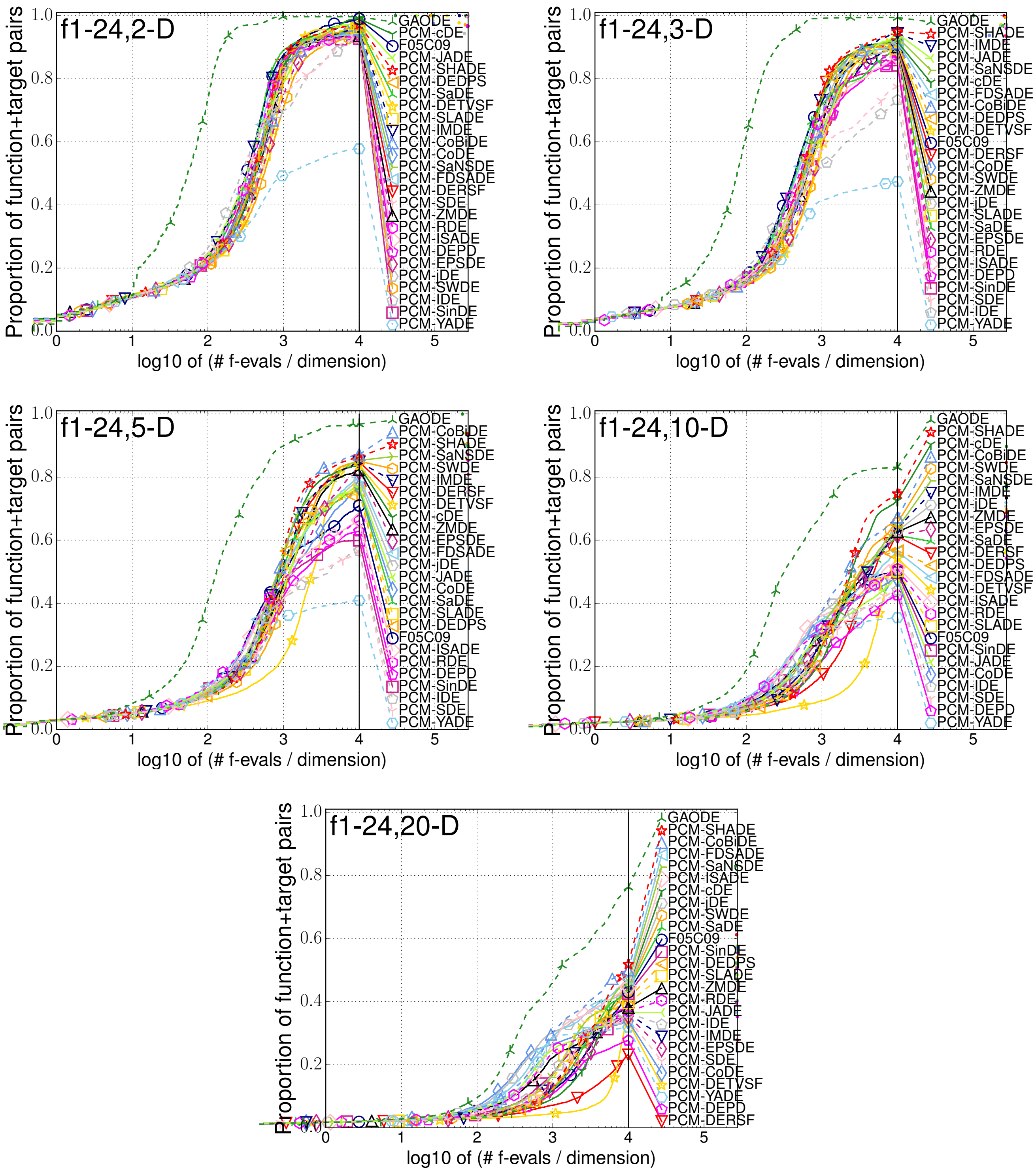

Fig. S.41: Comparisons of the 25 methods with GAODE on the BBOB benchmarks ($D \in \{2, 3, 5, 10, 20\}$). Tuned hyper parameter settings were used for each PCM. The rand/1/bin operator was used for all methods. These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the number of function evaluations (FEvals) divided by dimension for 50 targets in $10^{[-8..2]}$ for all functions (higher is better). For details of the ECDF, see the `COCO software` manual (http://coco.gforge.inria.fr/).

REFERENCES

[1] W. Yu, M. Shen, W. Chen, Z. Zhan, Y. Gong, Y. Lin, O. Liu, and J. Zhang, "Differential Evolution With Two-Level Parameter Adaptation," *IEEE Trans. Cybernetics*, vol. 44, no. 7, pp. 1080–1099, 2014.

[2] Z. Zhao, J. Yang, Z. Hu, and H. Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems," *EJOR*, vol. 250, no. 1, pp. 30–45, 2016.

[3] R. Tanabe and A. Fukunaga, "Tuning differential evolution for cheap, medium, and expensive computational budgets," in *IEEE CEC*, 2015, pp. 2018–2025.

[4] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.