Problem Data Based Optimization (PDBO) 
Algorithm for Continuous Optimization Problems

Abdulelah G. Saif, Safia Abbas, and Zaki Fayed, Member, IAENG

Abstract—Problem Data-Based Optimization (PDBO) algorithm is appeared in 2015 by Abdulelah Saif, Safia Abbas and Zaki Fayed for combinatorial optimization problems and is applied to Discrete Time, Cost and Quality Trade -off problem (DTCQTP). In this paper, Problem Data-Based Optimization (PDBO) algorithm is adapted to solve continuous optimization problems. The proposed algorithm called the PDBO-CO (PDBO for continuous optimization) is tested on few benchmark functions and on COCOMO II model coefficients by using NASA 93 Dataset. The obtained results for benchmark functions are compared with the ones obtained using IWD-CO(Intelligent Water Drops for continuous optimization ) and the obtained results from the optimized COCOMO II PA model coefficients by PDBO-CO are compared with ones optimized by IWD and Genetic algorithm (GA) and with the current COCOMO II PA model coefficients. The obtained results are satisfactory, which encourage other researches in this regard.

Index Terms—COCOMO II, Meta-heuristic, Numerical functions, Optimization, PDBO algorithm

I. INTRODUCTION

PDBO algorithm is a single agent meta-heuristic algorithm that is invented for combinatorial optimization problems by applying it to DTCQTP which depends on possibility calculated from problem's data. PDBO assumes the problem is represented in the form of a graph $G = (V, E)$, in which the set of nodes $V$ represents the activities and modes, and the set of $E$ represents edges that connects between activities and modes.

For optimization problems, at each iteration, PDBO selects the first node $n_i$ then depending on the best possibility values, it moves to the next adjacent node $n_i$. After that, in order to increase the chance of selecting other nodes rather than node $n_i$, in the next iteration, PDBO technique updates the $N_{possibility}(n_i, n_k) = N_{possibility}(n_i, n_k) + (\text{cost}/\alpha)$ where $\alpha > 0$. Finally, after the best iteration solution found, in order to evaporate the $N_{possibilities}$, PDBO considers the parameter $\beta \in [0,1]$, such that $N_{possibility}(n_i, n_k) = N_{possibility}(n_i, n_k) - \beta$, where $\beta$ is the evaporation rate (reduction rate) of $N_{possibility}(n_i, n_k)$ for virtual edge between $n_i$ and $n_k$.[1]

The PDBO is single agent meta-heuristic. Meta-heuristics especially nature-inspired swarm-based optimization algorithms which are being increasingly used for solving optimization problems. Several meta-heuristics are basically suitable for continuous optimization whereas the rest of them are initially defined for combinatorial optimization. Particle swarm optimization [2] and ant colony optimization [3] are among the popular meta-heuristics, which are used for optimization problems.

So far, the PDBO algorithm has been used for the Discrete Time, Cost and Quality Trade -off problem (DTCQTP). Naturally, the PDBO algorithm is appropriate for combinatorial optimization problems. In this research, the PDBO is used for continuous optimization. In a continuous optimization problem, a number of continuous variables (parameters) are needed to be obtained such that a function is minimized or maximized. Here, the proposed PDBO algorithm called the “PDBO-CO” (the PDBO algorithm for continuous optimization) encodes the real continuous variables into integer numbers. Then, the PDBO tries to optimize the given function in the integer representation. Finally, the best solution is considered as the final solution. Next section talks about PDBO-CO. For this purpose, a few benchmark functions and COCOMO II model (for software cost estimation by using NASA 93 Dataset) are utilized for testing the proposed PDBO-CO for the continuous optimization, which are given in section III. At the end, conclusion is given in section IV.

II. THE PROPOSED PDBO-CO ALGORITHM

In this section, the steps to optimize a given function by the PDBO-CO are explained. In fact, solutions are constructed with the help of a graph. The proposed PDBO-CO is shown in figure 2. The following subsections explain the components of the PDBO-CO.

A. Problem Representation

Given a function $f : S \rightarrow R$, find $X^* \in S : \forall X \in S \ f(X^*) \leq f(X)$ (minimization) or $f(X^*) \geq f(X)$ (maximization). Function $f$ is called the objective function, its domain $S$ is called the search space, and the elements of $S$ are called feasible solutions. A feasible solution $X$ is a vector of optimization variables $X = \{X_1, X_2, \ldots, X_n\}$. A feasible solution $X^*$ that minimizes/maximizes the objective function is called an optimal solution. The maximization over an objective function $f$ is equivalent to minimization over the function $-f$.[4]

To minimize this function by PDBO_Co, a graph of $n$ nodes ($n$ is number of variables) and 10 other virtual nodes

---

Manuscript submitted July 23, 2015; revised July 29, 2015. The authors gratefully acknowledge the support of Ain Shams University and Yemen government in supporting them.

Abdulhafeez Ghalib Farhan Saif is Ph.D student at Ain Shams University, Egypt (phone: 0020154415035; abdulhafeez.ghalib@gmail.com).

Safia Abbas Mahmood Abbas is lecturer at Ain Shams University, Egypt (Safia_abbas@yahoo.com).

Zaki Taha Ahmed Fayed is Emeritus Professor at Ain Shams University, Egypt (ZFayed@hotmail.com).
(digits i.e. domains) numbered from 0 to 9 which are connected to each node (variable) as in figure 1. Each of above variables is expressed by 4 digits which are chosen among 10 digits by PDBO_CO algorithm according to minimum possibilities (Po). First digit is integral part of a variable and the remaining 3 are fractions part. The possibilities are placed on the edges between variables and digits as in figure 1.

\[ \sum_{k=0}^{9} \text{Cost(Edge}(X_i,D_k)) = \text{Total Cost}(X_i) \]

E.g. \( \text{Total Cost}(X_1) = \sum_{k=0}^{9} \text{Cost(Edge}(X_1,D_k)) \)

Then, calculate the possibility of choosing the digit node \( D_k \) connected to variable node \( X_i \) among others as follow:

\[ \text{Possibility}(X_i,D_k) = \frac{\text{Cost(Edge}(X_i,D_k))}{\text{Total Cost}(X_i)} \]

E.g. \( \text{Possibility}(X_1,D_0) = 0/45 = 0.0, \quad \text{Possibility}(X_1,D_1) = 1/45 = 0.02, \quad \text{Possibility}(X_1,D_2) = 2/45 = 0.044, \quad \text{Possibility}(X_1,D_3) = 3/45 = 0.066, \quad \text{Possibility}(X_1,D_4) = 4/45 = 0.088, \quad \text{Possibility}(X_1,D_5) = 9/45 = 0.2. \]

C. Digit Selection Mechanism
PDBO_CO starts its journey from node 1 (\( X_1 \)) from which selects 4 digits among 10 digits which are connected to it according to minimum Possibility(\( X_i,D_k \)) in order and finishes it by visiting the last node (\( X_n \)) from which selects 4 digits among 10 digits which are connected to it. This step applies for all variables, if there is an improvement in the objective function, otherwise PDBO_CO selects four digits randomly from [0,9].

The 4 digits for each variable \( X_i \) are selected as follow:

\[ X_i = 4 \text{ digits whose Possibility}(X_i,D_k) \text{ are the smallest }, i=1,...,n, k=0,...,9 \]

E.g. \( X_1 = 0123(4 \text{ digits}); \) these digits are selected by PDBO_CO because 0.0, 0.022, 0.044 and 0.066 are the four smallest Possibility(\( X_1,D_k \)) among all in order.

Note: You can make PDBO_CO selects more than 4 digits, if you need. To obtain negative value to variable, its selected digits are multiplied by -1.

D. Updates Possibilities
PDBO_CO updates the Possibility(\( X_i,D_k \)) of the four selected digits for each variable \( X_i \) to be:

\[ \text{Possibility}(X_i,D_k) = \frac{\text{Cost(Edge}(X_i,D_k))}{\alpha} \]

where \( \alpha > 0 \) (\( \alpha \) user selected, here \( \alpha = 00001 \)).

E. Evaporate Possibilities
In this step, PDBO_CO has two choices:

1. Evaporates the Possibility(\( X_i,D_k \)) of the four selected digits for each variable \( X_i \) in the current iteration, if there is an improvement in the objective function in the current iteration.

2. Evaporates the Possibility(\( X_i,D_k \)) of the best selected digits for each variable \( X_i \) obtained from all iterations, if there is no improvement in the objective function in the current iteration.

The equation used is:

\[ \text{Possibility}(X_i,D_k) = \text{Possibility}(X_i,D_k) - \beta \]

where \( \beta > 0 \) (\( \beta \) user selected, here \( \beta = 0.00001 \)).

Set \( \alpha \) and \( \beta \) parameters.
2. Represent the problem in the form of graph as figure 1.
3. Determine problem dataset if exists.
4. Initialize the Possibility(\( X_i,D_k \)), \( i=1,...,n, k=0,...,9 \).
5. While (termination condition not met) do

   For each variable \( X_i \):
   - if there is an improvement in the objective function then
     Select 4 digits for variable \( X_i \) in the graph with Minimum Possibility in order.
   - Else
     Selects 4 digits for \( X_i \) randomly from [0,9].
   End if

   Update Possibility of virtual edge between \( X_i \) and selected digit \( D_k \) by
   \[ \text{Possibility}(X_i,D_k) = \text{Possibility}(X_i,D_k) + \frac{\text{Cost(Edge}(X_i,D_k))}{\alpha}. \]
   End i for

6. Find iteration solution i.e. evaluate the objective function.
7. If there is an improvement in the objective function then

   evaporate the possibilities of virtual edges edge(\( X_i,D_k \)) between all variables and their selected digits at this iteration by \( \text{Possibility}(X_i,D_k) = \text{Possibility}(X_i,D_k) - \beta \).

Else

   evaporate the possibilities of virtual edges edge(\( X_i,D_k \)) between all variables and their best selected digits from all iterations by \( \text{Possibility}(X_i,D_k) = \text{Possibility}(X_i,D_k) - \beta \).

End if
End while

10. Return the best solution

Fig. 2. The proposed PDBO_CO algorithm.
III. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed PDBO-CO, a few benchmark functions taken from [5] and COCOMO II Post Architecture model [6] are selected. The algorithm is implemented in c# and is tested and evaluated on CPU (Core( i5) 3210 M, 2.50 GHz) and 4GB RAM using Windows 7 as the operating system.

A. Benchmark Functions

The selected functions are shown in table I. For the functions f1 , f2, f3 , and f4 , the dimension of the input vectors are here selected to be ten. In contrast, the dimension of the last function f18 is originally fixed to the value of two.

<table>
<thead>
<tr>
<th>Benchmark Function</th>
<th>Ranges</th>
<th>Dim.</th>
<th>Minimum value (f_{min})</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1(X) = \sum_{i=0}^{n-1} x_i^2</td>
<td>-5.12 \leq x_i \leq 5.12</td>
<td>n \geq 1</td>
<td>0</td>
</tr>
<tr>
<td>f2(X) = \sum_{i=0}^{n-1}</td>
<td>x_i</td>
<td>+ \prod_{i=0}^{n-1}</td>
<td>x_i</td>
</tr>
<tr>
<td>f3(X) = \sum_{i=0}^{n-1} \left( \sum_{j=0}^{n-1} x_j \right)^2</td>
<td>-100 \leq x_i \leq 100</td>
<td>n \geq 1</td>
<td>0</td>
</tr>
<tr>
<td>f4(X) = \max</td>
<td>x_i</td>
<td>, 0 \leq i &lt; n</td>
<td>-100 \leq x_i \leq 100</td>
</tr>
<tr>
<td>f18(X) = \left[ [1 - (x_1 + x_2 + 1) \right] + (19 - 14x_1 + 3x_1^2) + (14x_2 + 6x_1x_2 + 3x_2^2)] \times \left[ [30 + (2x_1 - 3x_2)] + (18 - 32x_1 + 12x_2 + 84x_2^2 - 36x_1x_2 + 27x_2^2) \right]</td>
<td>2 \leq x_i \leq 2</td>
<td>n = 2</td>
<td>3</td>
</tr>
</tbody>
</table>

For each function, the PDBO-CO is run five times and the results are compared with that of IWD-CO found in [7]. The results of PDBO-CO and IWD-CO are shown in table II. The PDBO-CO converges to optimal values of the five functions.

### Table II: The results of the PDBO-CO and IWD-CO

<table>
<thead>
<tr>
<th>Benchmark function</th>
<th>PDBO-CO</th>
<th>IWD-CO</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>1.28E+17</td>
<td>6.44E-16</td>
</tr>
<tr>
<td>f2</td>
<td>2.33E+08</td>
<td>3.92E-08</td>
</tr>
<tr>
<td>f3</td>
<td>2.09E+13</td>
<td>7.51E-11</td>
</tr>
<tr>
<td>f4</td>
<td>1.00E+06</td>
<td>2.25E-03</td>
</tr>
<tr>
<td>f18</td>
<td>3.00E+05</td>
<td>3.00E00</td>
</tr>
</tbody>
</table>

B. COCOMO II Post Architecture model

COCOMO II PA model is one of software cost estimation methods which calculates the software development effort (in person months) by using the following equation:

\[ E = B \times (SIZE)^a \times \prod_{i=1}^{17} EMI_i \]  

(7)

A- multiplicative constant with value 2.94 that scales the effort according to specific project conditions. Size - Estimated size of a project in Kilo Source Lines of Code or Unadjusted Function Points. E - An exponential factor that accounts for the relative economies or diseconomies of scale encountered as a software project increases its size. EMI - Effort Multipliers. The coefficient E is determined by weighing the predefined scale factors (SFi) and summing them using following equation:

\[ E = B + 0.01 \times \sum_{i=1}^{17} SFI_i \]  

(8)

The development time (TDEV) is derived from the effort according to the following equation:

\[ TDEV = C \times (Effort)^d \]  

(9)

\[ F = D + 0.002 \times \sum_{i=1}^{17} SFI_i \]  

(10)

B=0.91,D=0.28. The values of effort multipliers and scale factors used in the implementation are taken from [6].

C. Dataset Description Used To Evaluate COCOMO II PA Model

Experiments have been conducted on NASA 93 data set found in [8]. The dataset consist of 93 completed projects with its size in kilo line of code (KLOC), actual effort in person-month, development time in months .

IV. RESULT ANALYSIS

The best results of PDBO-CO, IWD[9] and GA [6] are achieved using many iterations and a solution set is received from which the best solution is chosen i.e. a solution with the best fitness function values (Mean Magnitude of Relative Error (MMRE) for effort and time).

The final best solution obtained for coefficients(variables) by PDBO-CO is: \( A= 3.734, B=1.006, C=0.0402 \) and \( D=0.2827 \).

The final best solution obtained for coefficients(variables) by IWD is: \( A=3.762, B=1.005, C=4.484 \) and \( D=0.288 \).

The final best solution obtained for coefficients(variables) by GA is: \( A=3.673, B=1.005, C=0.244 \) and \( D=0.342 \).

Current COCOMO II PA model coefficients are the following: \( A=2.94, B=0.91, C=3.67 \) and \( D=0.28 \).
The following tables, table III and table IV, show the comparison among the actual, effort and time, values and estimated, effort (person month) and time (months), values for the first ten project dataset using PDBO, IWD and GA algorithm optimized and current COCOMO II PA model coefficients with their estimated project size.

### Table III: Estimated Development Effort Values

<table>
<thead>
<tr>
<th>Pr. No</th>
<th>Project Size (KLOC)</th>
<th>Actual Effort (PM)</th>
<th>Calculated Effort (PM) using coefficients optimized by PDBO</th>
<th>Calculated Effort (PM) using coefficients optimized by IWD</th>
<th>Calculated Effort (PM) using COCOMO II PA model current coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.9</td>
<td>117.6</td>
<td>103.095</td>
<td>103.53</td>
<td>102.10</td>
</tr>
<tr>
<td>2</td>
<td>24.6</td>
<td>117.6</td>
<td>97.8907</td>
<td>98.9090</td>
<td>96.8997</td>
</tr>
<tr>
<td>3</td>
<td>7.7</td>
<td>31.2</td>
<td>30.4278</td>
<td>30.593</td>
<td>30.555</td>
</tr>
<tr>
<td>4</td>
<td>8.2</td>
<td>36</td>
<td>32.4158</td>
<td>32.590</td>
<td>32.326</td>
</tr>
<tr>
<td>5</td>
<td>9.7</td>
<td>25.2</td>
<td>38.3842</td>
<td>38.3842</td>
<td>38.239</td>
</tr>
<tr>
<td>6</td>
<td>2.2</td>
<td>8.4</td>
<td>8.62855</td>
<td>8.6264</td>
<td>8.6279</td>
</tr>
<tr>
<td>7</td>
<td>3.5</td>
<td>10.8</td>
<td>13.7655</td>
<td>13.851</td>
<td>13.797</td>
</tr>
<tr>
<td>8</td>
<td>66.6</td>
<td>352.8</td>
<td>266.609</td>
<td>267.48</td>
<td>262.35</td>
</tr>
<tr>
<td>9</td>
<td>7.7</td>
<td>37.6</td>
<td>36.6467</td>
<td>36.847</td>
<td>36.565</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>72</td>
<td>33.1678</td>
<td>33.316</td>
<td>32.899</td>
</tr>
</tbody>
</table>

### Table IV: Estimated Development Time Values

<table>
<thead>
<tr>
<th>Pr. No</th>
<th>Project Size (KLOC)</th>
<th>Actual Time (Month s)</th>
<th>Calculated Time (Month s) using coefficients optimized by PDBO</th>
<th>Calculated Time (Month s) using coefficients optimized by IWD</th>
<th>Calculated Time (Month s) using COCOMO II PA model current coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.9</td>
<td>15.3</td>
<td>18.3044</td>
<td>18.061</td>
<td>17.961</td>
</tr>
<tr>
<td>2</td>
<td>24.6</td>
<td>15</td>
<td>17.9969</td>
<td>17.808</td>
<td>17.887</td>
</tr>
<tr>
<td>3</td>
<td>7.7</td>
<td>10.1</td>
<td>12.2816</td>
<td>12.069</td>
<td>12.429</td>
</tr>
<tr>
<td>4</td>
<td>8.2</td>
<td>10.4</td>
<td>12.5384</td>
<td>12.230</td>
<td>12.807</td>
</tr>
<tr>
<td>5</td>
<td>9.7</td>
<td>11</td>
<td>13.2508</td>
<td>13.239</td>
<td>13.857</td>
</tr>
<tr>
<td>6</td>
<td>2.2</td>
<td>6.6</td>
<td>8.13550</td>
<td>8.3570</td>
<td>8.369</td>
</tr>
<tr>
<td>7</td>
<td>3.5</td>
<td>7.8</td>
<td>9.47570</td>
<td>9.5590</td>
<td>9.320</td>
</tr>
<tr>
<td>8</td>
<td>66.6</td>
<td>21</td>
<td>24.9739</td>
<td>22.424</td>
<td>22.374</td>
</tr>
<tr>
<td>9</td>
<td>7.5</td>
<td>13.6</td>
<td>13.0516</td>
<td>12.670</td>
<td>12.571</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>14.4</td>
<td>12.6328</td>
<td>12.308</td>
<td>12.914</td>
</tr>
</tbody>
</table>

The graphical comparison among effort values and among time values described in table III and table IV respectively is shown in figure 3 and figure 4 respectively.

### Table V: Performance Measure Comparison

<table>
<thead>
<tr>
<th>Results</th>
<th>PDBO</th>
<th>TWD</th>
<th>GA</th>
<th>COCOMO II</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMRE for Effort</td>
<td>0.474929</td>
<td>0.474806</td>
<td>0.4752</td>
<td>0.6</td>
</tr>
<tr>
<td>MMRE for Time</td>
<td>0.093497</td>
<td>0.095901</td>
<td>0.092301</td>
<td>0.43</td>
</tr>
<tr>
<td>PRED (.25) for Effort</td>
<td>0.419355</td>
<td>0.419355</td>
<td>0.387097</td>
<td>0.09</td>
</tr>
<tr>
<td>PRED (.25) for Time</td>
<td>0.989247</td>
<td>0.95</td>
<td>1</td>
<td>0.06</td>
</tr>
</tbody>
</table>

The MMRE (Mean Magnitude of Relative Error) and PRED (.25) which show the performance of PDBO, IWD, GA and COCOMO II PA model in estimating the effort and time for the whole dataset.

\[
MMRE = \frac{1}{n} \sum_{j=1}^{n} | \frac{Actual - Estimated}{Actual} | (11)
\]

\[
MRE_{\text{p}} \left( \text{p} \right) = k / n 
\]

where MRE_{\text{p}} is the number of projects where MRE is less than or equal to p, and n is the total number of projects.

---

**Fig. 3.** Comparison among Effort values vs. Size.

**Fig. 4.** Comparison among Time values vs. Size.
From table V and figure 5, the MMRE of PDBO, IWD and GA for effort and time is equal and lower than that of COCOMO II PA model, whereas PRED(0.25) of PDBO and of IWD for effort is equal and for time PDBO is larger than that of IWD. PRED(0.25) of GA for effort is smaller than that of PDBO and IWD, but is the largest for time. COCOMO II is the worst among all.

It shows clearly that optimized coefficients by PDBO, IWD and GA algorithm produces more accurate results than the old coefficients.

V. CONCLUSION

This paper adapts PDBO algorithm to solve continuous optimization problems. The proposed PDBO algorithm, PDBO-CO, is tested on few well known benchmark functions and on COCOMO II PA model coefficients by using NASA 93 dataset. The obtained results for benchmark functions are compared with the ones obtained using IWD-CO and the obtained results from the optimized COCOMO II PA model coefficients by PDBO-CO are compared with ones optimized by IWD and GA and with the current COCOMO II PA model coefficients. The obtained results are satisfactory. In the future, other coding methods may be used instead of integer numbers.

REFERENCES
