Estimation of the Convergence Points of a Population Using an Individual Pool

Jun Yu
Graduate School of Design
Kyushu University
Fukuoka, Japan
yujun@kyudai.jp

Hideyuki Takagi
Faculty of Design
Kyushu University
Fukuoka, Japan
http://www.design.kyushu-u.ac.jp/~takagi/

***Abstract*—We employ an individual pool to increase the precision of the estimated convergence points of a population by using individual information from past generations. Better individuals from past generations are kept in the pool, and poorer individuals are replaced with new better individuals when the pool becomes full; convergence points for the population are thus estimated using those individuals from the pool that keeps excellent individuals in past generations. The estimated convergence points are used as elite individuals, and replace the worse individuals in current population to accelerate evolutionary computation. Besides the proposed basic pool storage mechanism, we further develop an extended version which enhances the interaction between an individual pool and the population in the latest generation. We evaluate these proposed methods using differential evolution and 14 benchmark functions. The experimental results show that introducing an individual pool can improve the convergence speed and accuracy with the same computational cost, and the extended version could further enhance the accelerated effect in almost all cases.**

***Index Terms*—multi-modal optimization, estimation of convergence points, acceleration, pool**

I. Introduction

Evolutionary Computation (EC) is a population-based optimization, which mainly simulates biological evolution and survival of the fittest to find the optima. Accelerating the convergence speed and improving the convergence accuracy have always been hot topics. There are several approaches to achieve the above objective, such as by developing new EC algorithms [1], [2], [6], improving EC operations [3], [4], and approximating the fitness landscape for a rough but quick search [5]. However, we have found that the vectors between two subsequent generations, from parent to offspring, also contain a lot of information and can be used to guide evolution.

The estimation method [7], [8] takes a novel perspective of using rigorous mathematical reasoning to estimate the convergence point of the population. Although the original proposal is suitable for unimodal optimization, the situation for multimodal optimization is also being studied and some study results have also been presented. For example, [9] develops a clustering method for bipolar tasks and proposes four improvements to increase the accuracy of the estimated convergence points. [10] firstly attempts to combine EC with the estimation method and analyzes the effect of the proposed four improvements of [9] and their combinations.

The main objective of this paper is to introduce an individual pool storage mechanism to preserve and use outstanding individuals from past generations instead of the current generation for estimating the convergence points of a population. Furthermore, we also develop an extended version which improves the interaction between an individual pool and the population to more fully use historical information. Then, we compare the basic version and extended version with a baseline algorithm using 14 benchmark functions.

The remaining paper is organized as following. We briefly summarize the estimation method [7], [8] in Section II-A and an extension of the estimation method for multimodal tasks [9] is described in Section II-B. The proposed individual pool storage mechanism and the extended version is presented in Section III. Then, we compare them with the baseline algorithm using 14 benchmark functions of 3 different dimensions in Section IV. Finally, the discussions and conclusion are given in Section V and Section VI, respectively.

II. Previous Research

A. Estimation Method

The convergence point for the vectors between parent individuals and their offspring can be calculated mathematically [7], [8]. Let us begin by defining symbols. $\boldsymbol{a}_i$ and $\boldsymbol{c}_i$ in the Figure 1 are the i-th parent individual and its offspring, respectively ($\boldsymbol{a}_i, \boldsymbol{c}_i \in \mathbb{R}^d$). Then, the i-th vector is defined as the direction vector, $\boldsymbol{b}_i = \boldsymbol{c}_i - \boldsymbol{a}_i$. The unit direction vector of the $\boldsymbol{b}_i$ is given as $\boldsymbol{b}_{0i} = \boldsymbol{b}_i / ||\boldsymbol{b}_i||$, i.e. $\boldsymbol{b}_{0i}^T \boldsymbol{b}_{0i} = 1$.

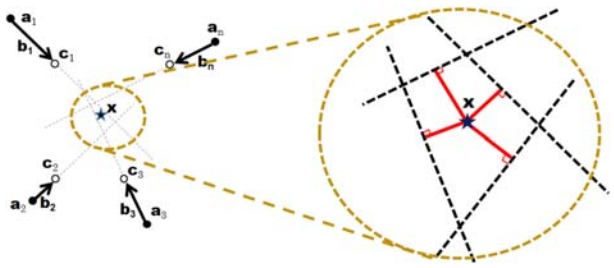


Fig. 1: Vector $\boldsymbol{b}_i$ $(= \boldsymbol{c}_i - \boldsymbol{a}_i)$ is calculated from a parent individual $\boldsymbol{a}_i$ and its offspring $\boldsymbol{c}_i$ in the d-dimensional searching space. The $\star$ mark is the convergence point for these vectors.

Let $\boldsymbol{x} \in \mathbb{R}^d$ be the point that is the nearest to the n extended directional line segments, $\boldsymbol{a}_i + t_i\boldsymbol{b}_i$ ($t_i \in R$). By *nearest*, we mean that the total distance from $\boldsymbol{x}$ to the n extended directional line segments, $\mathrm{J}(\boldsymbol{x}, \{t_i\})$ in Eq.(1), becomes the minimum.

As the minimum line segment from the convergence point $\boldsymbol{x}$ to the extended directional line segments is the orthogonal projection from $\boldsymbol{x}$, we may insert an orthogonal condition, Eq. (2), into Eq. (1) and thus remove t_i.

$$\mathrm{J}(\boldsymbol{x}, \{t_i\}) = \sum_{i=1}^{n} \|\boldsymbol{a}_i + t_i\boldsymbol{b}_i - \boldsymbol{x}\|^2 \tag{1}$$

$$\boldsymbol{b}_i^{\mathrm{T}}(\boldsymbol{a}_i + t_i\boldsymbol{b}_i - \boldsymbol{x}) = 0 \quad \text{(orthogonal condition)} \tag{2}$$

The $\hat{\boldsymbol{x}}$ that minimizes the total distance in Eq. (1) is obtained by partially differentiating each element of $\boldsymbol{x}$ and setting them equal 0. Finally, the convergence point $\hat{\boldsymbol{x}}$ is given by Eq. (3), where $\boldsymbol{I}_d$ is the unit matrix.

$$\hat{\boldsymbol{x}} = \left\{ \sum_{i=1}^{n} \left(\boldsymbol{I}_d - \boldsymbol{b}_{0i}\boldsymbol{b}_{0i}^{\mathrm{T}} \right) \right\}^{-1} \left\{ \sum_{i=1}^{n} \left(\boldsymbol{I}_d - \boldsymbol{b}_{0i}\boldsymbol{b}_{0i}^{\mathrm{T}} \right) \boldsymbol{a}_i \right\} \tag{3}$$

B. Estimation for Multimodal Task

The estimation method discussed in the previous section is clearly effective for unimodal tasks, but it is not always valid for multi-modal tasks because vectors go towards different local optima. To make this method applicable to general optimization tasks, we must extend the basic estimation method to multi-modal tasks. As the first step in tackling this challenge, we started with the bipolar task and have developed a method to cluster vectors using their direction and location information. In this paper, we are not concerned about the clustering and ignore the interpretation of it, but a detailed description of the clustering steps can be found in [9].

Four improvements are proposed to improve clustering correctness and the precision of the estimated local minima. Each improvement targets a different problem; they are briefly introduced below and a more detailed analysis can be found in [9].

Improvement 1: Creating a vector in a tiny area

To move the vector towards the local minimum, create a vector in a tiny area which can be approximated by a hyperplane and let the individual with higher fitness value and another be offspring and parent, respectively.

Improvement 2: Clustering only top vectors

This improvement is based on the fact that the fitness of individuals in areas of poorer fitness is lower than those near local minima; we may assume therefore that the vector directions of individuals located closer to local minima go towards their nearest local minima more correctly.

Improvement 3: Correcting directions of vectors

The direction of a vector can be corrected to go towards a local minimum by using orthogonal vectors. We calculate a synthesized vector by weighting vectors according to increasing fitness value and let the endpoint of the synthesized vector be a new offspring to replace the original ones.

Improvement 4: Correction of Mis-clustering.

Improvement 4 is a method for correcting for mis-clustering using estimated convergence points after applying *Improvements 1, 2,* and *3*. After these corrections, we calculate the convergence points once again.

III. Proposed Methods

In our previous study, we just use the current generation to estimate the convergence points and ignore the fact that the previous generations can also offer useful information. In fact, as the population gradually converges to the optimal area, the diversity of the population is also gradually lost and the similarity between individuals increases. This tendency is not conducive to the accurate estimation of the convergence points of the population, so we use historical individuals to alleviate the loss of diversity and improve the accuracy of the estimated points. To achieve this, we introduce an individual pool storage mechanism to preserve relatively outstanding individuals from previous generations. We furthermore develop an extension to this method which increases the communication between this individual pool and the population.

A. The Individual Pool Mechanism

The individual pool introduces a separate storage mechanism, different from the population, to save the better individuals from previous generations. Suppose the size of the individual pool is N; past individuals are copied to the individual pool in turn until the maximum limit N is reached. Then, better individuals from each subsequent generation are selected and replace the worse individuals in the pool. There are many ways to implement this replacement strategy. In this paper, we adopt a greedy replacement strategy, which means that selected individuals will replace the worst ones in the individual pool in turn if they are better than the worst ones. Finally, individuals coming from the individual pool - rather than those from the current generation - are used to estimate the convergence points of the population. Fig. 2 demonstrates how to use the pool to estimate the convergence points of the population. This framework for the proposed individual pool can be summarised as below in Algorithm 1.

B. Extended Version

The above outlined proposal emphasizes the use of past information to estimate the convergence points of the population but does not re-use the outstanding individuals in the past to guide the search of individuals in current generation. There should be many methods to further use historical information and enhance performance.

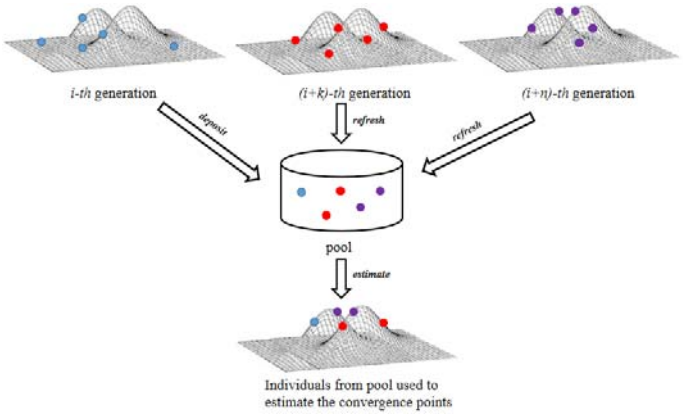


Fig. 2: Individuals from a pool are used to estimate the convergence points.

Algorithm 1 The general framework for using a pool to estimate the convergence points for accelerating EC.

1: Generate an initial population.
2: **for** $G = 1$ to $MaxGeneration$ **do**
3: **if** the maximum limit N is not reached **then**
4: copy individual to the pool in turn.
5: **end if**
6: **if** the maximum limit N is reached **then**
7: Obtain the estimated convergence point(s) using individuals from the pool.
8: Replace worse individuals in the pool with the better ones from the following generations.
9: **end if**
10: Generate the offspring using EC.
11: Replace the worse individuals in the population with the estimated point(s).
12: **end for**
13: return the optimum

In this part, we develop an extended version to obtain a better acceleration effect by strengthening the interaction between the individual pool and the population. There are two new extra modifications based on the basic proposed pool version. The first approach is to make full use of the individuals generated from the first improvement proposed in [9]. In our previous research, the generated offspring around the parent is just used to construct the moving vector, and is then discarded. So, in the first approach, we replace the parent with its offspring if the offspring is better than its parent. Nevertheless, the population and the individual pool is independent, and there remains no interaction between them. The second approach therefore focuses on improving the interaction of the two storage mechanisms. This time, we first rank the population and remove the $a\% * M$ ranked lower individuals, where M is population size. Then, randomly select $a\% * M$ individuals from the pool, and copy them to population to keep the population size unchanged. In the following experiments, a is set to 10.

IV. Experimental Evaluations

14 benchmark functions from the CEC2005 test suite [12] are used in our evaluations. Table I shows their types, characteristics, variable ranges, and optimum fitness values. These landscape characteristics include shifted, rotated, global on bounds, unimodal and multi-modal. We test them with three dimensional settings, D =2-D, 10-D, and 30-D.

TABLE I: Benchmark functions (Uni=Uni-modal, Multi=Multi-modal, Sh=Shifted, Rt=Rotated, GB=Global on Bounds, NM=Number Matrix)

No.	Types	Characteristics	Ranges	Optimum fitness
f_1	Uni	Sh Sphere	$[-100, 100]$	-450
f_2		Sh Schwefel 1.2		-450
f_3		Sh Rt Elliptic		-450
f_4		f_2 with Noise		-450
f_5		Schwefel 2.6 GB		-310
f_6	Multi	Sh Rosenbrock	$[-100, 100]$	390
f_7		Sh Rt Griewank	$[0, 600]$	-180
f_8		Sh Rt Ackley GB	$[-32, 32]$	-140
f_9		Sh Rastrigin	$[-5, 5]$	-330
f_{10}		Sh Rt Rastrigin	$[-5, 5]$	-330
f_{11}		Sh Rt Weierstrass	$[-0.5, 0.5]$	90
f_{12}		Schwefel 2.13	$[-\pi, \pi]$	-460
f_{13}		Sh Expanded F8F2	$[-5, 5]$	-130
f_{14}		Sh Rt Scaffer F6	$[-100, 100]$	-300

To test the effectiveness of the individual pool, we use exactly the same experimental settings as [10], where we investigated various combinations of proposed improvements in [9]. The experimental results showed that the combination of *Improvements 1, 2, and 4* with differential evolution (DE) can obtain the best performance in most cases. In this evaluation, we use the best combination as the baseline algorithm. Two experiments were designed where the individual pool is combined with the baseline algorithm. In the first experiment, we combine the baseline algorithm with basic individual pool, while in the second experiment, we combine the baseline algorithm with extended version. The DE experimental parameters are set as in Table II. Additionally, the size of the pool is set to the same size as the that of population in each of the searching dimensions, 2-D, 10-D, and 30-D.

TABLE II: DE algorithm parameter settings.

population size for 2-D, 10-D, and 30-D search	20, 50, and 100
scale factor F	1
crossover rate	0.9
DE operations	DE/rand/1/bin
stop condition; max. # of fitness evaluations, MAX_{NFC}, for for 2-D, 10-D, and 30-D search	800, 10,000, 60,000
dimensions of benchmark functions, D	2, 10, and 30
# of trial runs	30

For fair evaluations, we evaluate convergence along the number of fitness calls instead of generations. We test each benchmark function with 30 trial runs in 3 different dimensional spaces and apply the Wilcoxon signed-rank test on the fitness values at the maximal number of fitness calculations, MAX_{NFC}, to check the significance of the proposal and

TABLE III: Wilcoxon signed-rank test results on the fitness values at the maximal number of fitness calculations of two experiments for 30 trial runs for 14 functions. $\gg$ and $>$ mean that the left is better than the right with significance levels of 1%, 5% respectively, and $\approx$ means there is no significance. B represents the basic algorithm described in the previous section, P1 and P2 are basic version and extended version used in experiment 1 and experiment 2, respectively.

Func.	Experiment 1			Experiment 2		
	2-D	10-D	30-D	2-D	10-D	30-D
f_1	P1 $\gg$ B	B $\approx$ P1	B $\gg$ P1	P2 $\gg$ B	P2 $\gg$ B	P2 $\gg$ B
f_2	P1 $\gg$ B	P1 $\approx$ B	B $\approx$ P1	P2 $\gg$ B	P2 $\gg$ B	P2 $\gg$ B
f_3	P1 $\approx$ B	B $\approx$ P1	B $\approx$ P1	P2 $>$ B	B $\approx$ P2	P2 $\gg$ B
f_4	P1 $\gg$ B	P1 $\approx$ B	B $\approx$ P1	P2 $\gg$ B	B $>$ P2	B $\gg$ P2
f_5	P1 $>$ B	P1 $\gg$ B	P1 $\approx$ B	P2 $\gg$ B	B $\approx$ P2	B $\approx$ P2
f_6	B $\approx$ P1	B $\approx$ P1	B $\approx$ P1	B $\approx$ P2	B $\approx$ P2	P2 $\approx$ B
f_7	P1 $\approx$ B	P1 $\gg$ B	P1 $\gg$ B	P2 $\gg$ B	P2 $\gg$ B	P2 $\gg$ B
f_8	B $\approx$ P1	P1 $\gg$ B	P1 $\gg$ B	P2 $\approx$ B	P2 $\gg$ B	P2 $\gg$ B
f_9	B $\approx$ P1	P1 $\gg$ B	P1 $\gg$ B	B $\approx$ P2	P2 $\gg$ B	P2 $\gg$ B
f_{10}	P1 $\approx$ B	P1 $\gg$ B	P1 $\approx$ B	P2 $\approx$ B	P2 $\gg$ B	B $\approx$ P2
f_{11}	P1 $\gg$ B	P1 $\gg$ B	P1 $\gg$ B	P2 $\gg$ B	P2 $\gg$ B	P2 $\gg$ B
f_{12}	P1 $>$ B	B $\approx$ P1	B $\approx$ P1	P2 $\gg$ B	B $\approx$ P2	B $\approx$ P2
f_{13}	B $\approx$ P1	P1 $\gg$ B	P1 $\gg$ B	B $\approx$ P2	P2 $\gg$ B	P2 $\gg$ B
f_{14}	B $\gg$ P1	B $\approx$ P1	P1 $\gg$ B	B $>$ P2	P2 $\gg$ B	P2 $\gg$ B

baseline algorithm. Tables III show the statistical test result of experiment 1 and experiment 2. The Fig. 3 shows the average convergence curves of three methods with 30 trial runs on 30D.

V. Discussions

We begin our discussion with an explanation of the superiority of our proposed strategies. In our previous research, only the current generation was used to estimate the convergence points; meanwhile the distribution of population became more and more concentrated with the evolution of population. It may be detrimental to estimate the convergence points from a population which is thus gradually converging. However, an individual pool can overcome the above limitation and preserve outstanding individuals from previous generations to ensure the diversity and breadth of the distribution. Although the proposed individual pool can flexibly use historical information, its size needs to be further studied; too large or too small and it is not helpful for estimating. In the next step, we will focus on dynamically adjusting the size according to the optimization problem.

The extended version emphasizes the more efficient use of information and interaction between the population and the individual pool. In our previous studies, the improvement 1 proposed in [9] can be further utilized. In that improvement, the offspring will replace its parent if it is better. Here we have also introduced an interactive mechanism to strengthen the communication between the population and the pool. Using better individuals from the pool to replace worse ones in the population can accelerate the convergence of the population, which in turn will improve the quality of the pool. This virtuous circle, thus formed, can obtain better accelerated convergence.

To analyze the performance of the two proposed versions, the Wilcoxon signed-rank test were applied at the stop condition in three different dimensions. The statistical results reproduced in Table III show that either of the two proposed version can improve the performance of the original baseline algorithm, and the extended version can further improve performance in almost all evaluation cases.

From the results of statistical tests, we can see that the basic individual pool can accelerate multimodal optimization, while it does not achieve much acceleration effect in unimodal optimization. The extended version can achieve favorable results in both cases. This could be caused by the interaction between the population and the pool, while detailed reasons need to be depth studied in our future work. Otherwise, f_2 and f_4 are exactly the same benchmark function, with f_4 just adding noise interference based on f_2. The experimental results reveal that the noise may reduce the accuracy of the estimation such that no acceleration effect can be produced. It may be a big challenge to remove the noise and smooth the landscape to obtain better performance.

VI. Conclusion

Based on our previous studies, we introduced an individual pool to increase the availability of historical information, and an extended version was also proposed to further improve the interaction between the pool and the population. The experimental results show that these improvements can enhance the convergence speed and accuracy.

In future work, we will further develop the adaptive archive strategy to adjust the storage size in real time according to the progress of the optimization problem.

Acknowledgment

This work was supported in part by Grant-in-Aid for Scientific Research (JP15K00340).

References

[1] Back, T., Hammel, U., and Schwefel, H.-P., "Evolutionary computation: Comments on the history and current state," IEEE Trans. on Evolutionary Computation, vol.1, no.1, pp.3–17 (1997).

[2] Coello Coello, C.A. "Evolutionary multi-objective optimization: A historical view of the field," IEEE Computational Intelligence Magazine (2006).

[3] Das, S. and Suganthan, P.N. "Differential evolution: A survey of the state-of-the-art," IEEE Trans. on Evolutionary Computation, vol.15, no.1, pp.4–31 (2011).

[4] Eiben, A.E., Hinterding, R., and Michalewicz, Z., "Parameter control in evolutionary algorithms," IEEE Trans. on Evolutionary Computation, vol.3, no.2, pp.124–141 (1999).

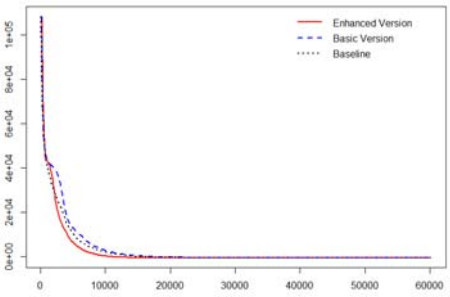
[5] Michael D. S., Hod L., "Coevolution of Fitness Predictors," IEEE Trans. on Evolutionary Computation, vol.12, no.6, pp.736–749 (2008).

[6] Mullen, R.J., Monekosso, D., Barman, S., and Remagnino, P., "A review of ant algorithms," Expert Systems with Applications, vol.36, no.6, pp.9608–9617 (2009).

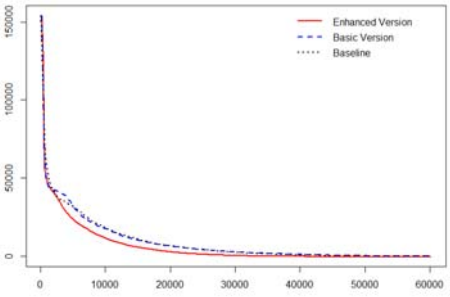
[7] Murata, N., Nishii, R., Takagi, H., and Pei Y., "Estimation Methods of the Convergence Point of Moving Vectors Between Generations," Japanese Society for Evolutionary Computation Symposium 2014, Hatsukaichi, Japan, pp.210–215 (Dec., 2014). (in Japanese).

[8] Murata, N., Nishii, R., Takagi, H., and Pei Y., "Analytical Estimation of the Convergence Point of Populations," 2015 IEEE Congress on Evolutionary Computation (CEC2015), Sendai, Japan, pp. 2619–2624 (May 25-28, 2015).

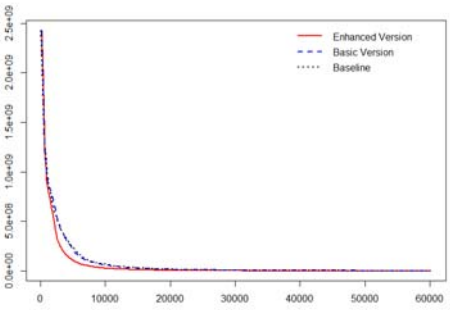
[9] Yu J. and Takagi H., "Clustering of Moving Vectors for Evolutionary Computation," 7th Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR2015), pp.169-174, Fukuoka, Japan (13-15, Oct. 2015).
[10] Yu J., Pei Y. and Takagi H., "Accelerating Evolutionary Computation Using Estimated Convergence Points," 2016 IEEE Congress on Evolutionary Computation (CEC2016), Vanvouver, Canada, pp.1438-1444 (July 24-29, 2016).
[11] Arthur, D. and Vassilvitskii, S., "k-means++: the advantages of careful seeding," 18th ACM-SIAM symposium on discrete algorithms (SODA2007), PA, USA, pp.1027–1035 (2007).
[12] Suganthan P. N., Hansen N. , Liang J. J., Deb K. , Chen Y.-P., Auger A. , and Tiwari S. . "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," https://www.lri.fr/ hansen/Tech-Report-May-30-05.pdf.



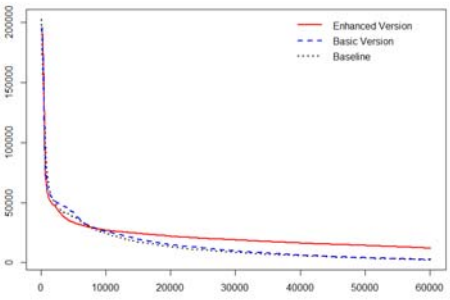
(a) f_1



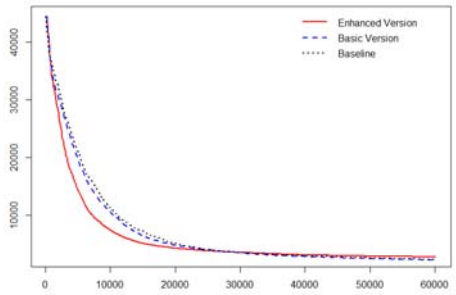
(b) f_2



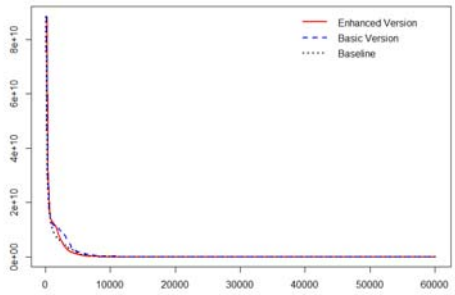
(c) f_3



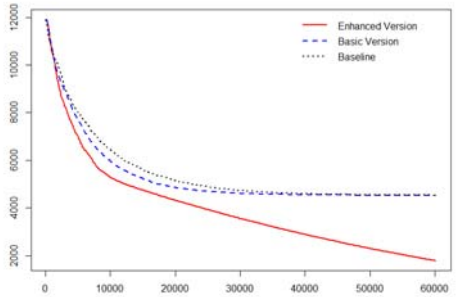
(d) f_4



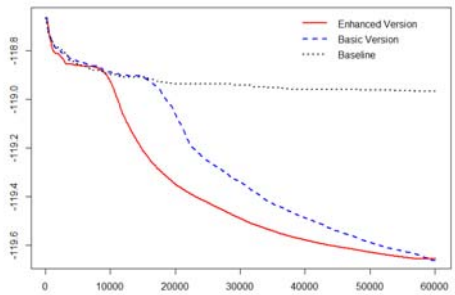
(e) f_5



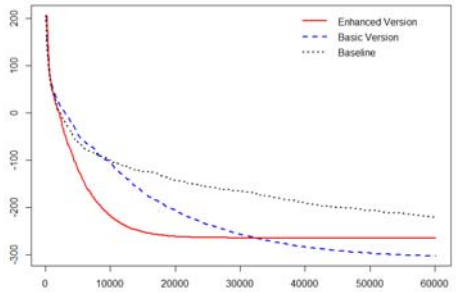
(f) f_6



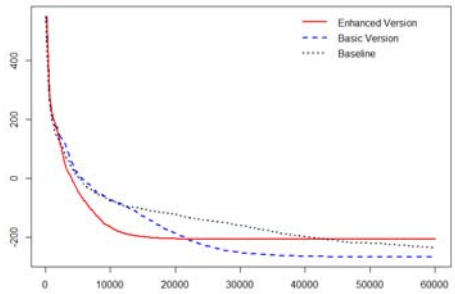
(g) f_7



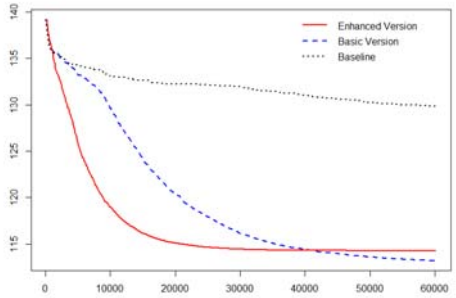
(h) f_8



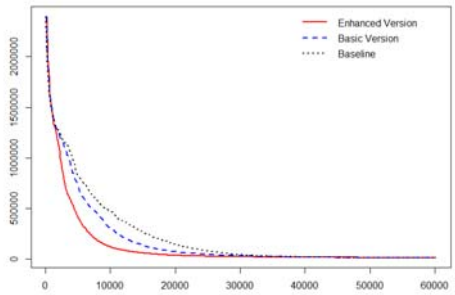
(i) f_9



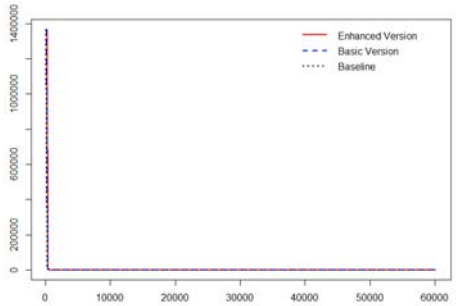
(j) f_{10}



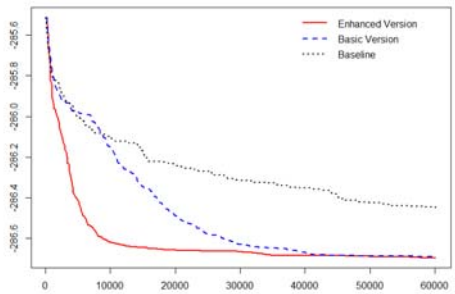
(k) f_{11}



(l) f_{12}



(m) f_{13}



(n) f_{14}

Fig. 3: Convergence curves of 30-D f_1–f_{14}.