Information Sciences

Adjustable driving force based particle swarm optimization algorithm

--Manuscript Draft--

Manuscript Number:	INS-D-22-1158R2
Article Type:	VSI:Theo and Appl of KIEOGNN
Keywords:	Particle swarm optimization; Novelty driving force; Hybrid driving force; Adjustable parameters
Corresponding Author:	Xuewen Xia, Ph.D. Minnan Normal University Zhangzhou, CHINA
First Author:	Fei Yu
Order of Authors:	Fei Yu
	Lei Tong
	Xuewen Xia, Ph.D.
Manuscript Region of Origin:	CHINA
Abstract:	Particle swarm optimization algorithm (PSO) is a popular optimizer, in which each particle selects its learning exemplars relying on their fitness. Thus, the search process of each particle can be seen as driven by a fitness-based force. Intuitively, the driving force is conducive to the optimizing process. However, it may bring a premature convergence of a population. In this work, a novelty-based driving force is put forward to overcome deficiencies of the fitness-based driving force. In the new proposed adjustable driving force based PSO, named as ADFPSO, two types of exemplars respectively with high fitness and high novelty are saved in two archives. In each generation, a particle respectively chooses two exemplars from the two archives to update its velocity. In addition, three time-varying parameters are introduced to adjust the particle's learning weights for the two exemplars aiming to satisfy distinct requirements of different evolution stages. Comprehensive properties of ADFPSO are extensively testified by a set of experiments, in which nine PSO variants are adopted as peer algorithms and two CEC test suites are selected as optimization problems. Moreover, distinct characteristics of the proposed novelty-based driving force are also analyzed based on a few experiments.

Dear Editors:

We would like to submit the enclosed manuscript entitled “Adjustable driving force based particle swarm optimization algorithm”, which we wish to be considered for publication in “Information Sciences”. No conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

In this work, a novelty-based driving force is introduced to overcoming deficiencies of the fitness-based driving force in canonical PSO. In the new proposed adjustable driving force based PSO, named as ADFPSO, two types of exemplars respectively with high fitness and high novelty are saved in two archives. In each generation, a particle respectively chooses two exemplars from the two archives to update its velocity. In addition, three time-varying parameters are introduced to adjust a particle’s learning weights for the two driving forces aiming to satisfy distinct requirements of different evolution stages.

I hope this paper is suitable for “Information Sciences”.

We deeply appreciate your consideration of our manuscript, and we look forward to receiving comments from the reviewers. If you have any queries, please don’t hesitate to contact me at the address below.

Thank you and best regards.

Yours sincerely,

Xuewen Xia

Corresponding author:
Name: Xuewen Xia
E-mail: xwxia@whu.edu.cn

Reviewer #1: The authors have overcome all the questions. My suggestion is acceptance.

Response: Thanks for the reviewer's enthusiastic work for our study.

Reviewer #2: Authors have addressed all the issues according to my previous comments. The related work has been enriched and the indistinct description as well as deficient analysis has been further refined. More discussions have also been added. This paper has been revised thoroughly to reach the standard for publication. Consequently, I advise to accept this paper.

Response: Thanks for the reviewer's enthusiastic help for our study.

Reviewer #3: In the revised version, the authors have addressed the comments properly. I regard that the manuscript can be accept after some typos being corrected.
1. "performance" in the manuscript is an uncountable noun. However, the authors misused it in some sentences.

Response: Thanks for the reviewer's careful review. According to the comment, we correct this type of errors in our manuscript. Moreover, the manuscript is further polished by us. Note that modifications in the revised manuscript are highlighted with red color.

2. In Section Conclusion, the sentence "During the search process of the population, the former one ..., while the latter one ..." has two unclear referents, i.e., "the former one" and "the latter one".

Response: According to the suggestion, the sentence is rewritten. Moreover, the manuscript is further polished by us.

Highlights:

- In this paper, a novelty-based driving force is introduced to overcome deficiencies of the fitness-based driving force.
- During the evolution, two types of exemplars respectively with high fitness and high novelty are saved in two archives.
- In each generation, a particle respectively chooses two exemplars from the two archives to update its velocity.
- Three time-varying parameters are introduced to adjust a particle's learning weights for the two driving forces aiming to satisfy distinct requirements of different evolution stages.

Abstract:

Particle swarm optimization algorithm (PSO) is a popular optimizer, in which each particle selects its learning exemplars relying on their fitness. Thus, the search process of each particle can be seen as driven by a fitness-based force. Intuitively, the driving force is conducive to the optimizing process. However, it may bring a premature convergence of a population. In this work, a novelty-based driving force is put forward to overcome deficiencies of the fitness-based driving force. In the new proposed adjustable driving force based PSO, named as ADFPSO, two types of exemplars respectively with high fitness and high novelty are saved in two archives. In each generation, a particle respectively chooses two exemplars from the two archives to update its velocity. In addition, three time-varying parameters are introduced to adjust the particle's learning weights for the two exemplars aiming to satisfy distinct requirements of different evolution stages. Comprehensive properties of ADFPSO are extensively testified by a set of experiments, in which nine PSO variants are adopted as peer algorithms and two CEC test suites are selected as optimization problems. Moreover, distinct characteristics of the proposed novelty-based driving force are also analyzed based on a few experiments.

Source File - cls

Click here to access/download

LaTeX/ MS Word Source Files

elsarticle.cls

Adjustable driving force based particle swarm optimization algorithm

Fei Yu[a], Lei Tong[b], Xuewen Xia*[a,c]

[a] *College of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China*
[b] *School of Tourism Management, Wuhan Business University, Wuhan, China*
[c] *Key Lab of Intelligent Optimization and Information Processing, Minnan Normal University, Zhangzhou 363000, China*

Abstract

Particle swarm optimization algorithm (PSO) is a popular optimizer, in which each particle selects its learning exemplars relying on their fitness. Thus, the search process of each particle can be seen as driven by a fitness-based force. Intuitively, the driving force is conducive to the optimizing process. However, it may bring a premature convergence of a population. In this work, a novelty-based driving force is put forward to overcome deficiencies of the fitness-based driving force. In the new proposed adjustable driving force based PSO, named as ADFPSO, two types of exemplars respectively with high fitness and high novelty are saved in two archives. In each generation, a particle respectively chooses two exemplars from the two archives to update its velocity. In addition, three time-varying parameters are introduced to adjust the particle's learning weights for the two exemplars aiming to satisfy distinct requirements of different evolution stages. Comprehensive properties of ADFPSO are extensively testified by a set of experiments, in which nine PSO variants are adopted as peer algorithms and two CEC test suites are selected as optimization problems. Moreover, distinct characteristics of the proposed novelty-based driving force are also analyzed based on a few experiments.

Keywords: Particle swarm optimization, Novelty driving force, Hybrid driving force, Adjustable parameters

1. Introduction

During the last few years, a lot of optimization techniques play a very important role in various research fields. As a popular intelligence algorithm, particle swarm optimization (PSO) is proposed in 1995 [14], and favorable performance of it has been verified by many applications, including supply chain, DNA sequences collections, cryptography, feature selection, and so on [25, 27, 31, 35, 45].

In the canonical PSO, a particle adjusts its flight trajectory to search for a candidate solution under a "driving force" of two exemplars, i.e., the historical best positions of the particle and the population [42]. Generally, it is a commonly used method that a particle selects other particles with higher fitness as exemplars due to that the exemplars can drive the particle to fly towards more promising regions. Although the driving force exclusively based on fitness values can speed up the convergence of a population, it may cause the population very vulnerable to premature convergence when optimizing a complicated fitness landscape. In other words, the fitness-based driving force is harmful for the exploration though it is conducive to the exploitation. To overcome the weaknesses and enhance the exploration capability of the population, some study incorporate various disturbances in the search process, which can be deemed a randomness-based driving force, intending to bring a high population diversity, and then improve the exploration capability [2, 16].

Nowadays, unlike focusing on an objective-based optimization in the PSO community, some researches in the artificial life field pour much attention on systems without an explicit objective [3, 36]. A popular strategy disregarding an explicit objective is generating an artificial life system with a high novelty instead of a good fitness value. Because the study completely neglects the explicit target, it can overcome some shortcomings inherent in those fitness-based searches, such as deception and local optima. Moreover, some researchers draw a counterintuitive conclusion that neglecting (or partial neglecting) an explicit objective is beneficial for pursuing the objective [7, 17].

In this study, based on discussions mentioned above, an adjustable driving force based PSO (ADFPSO) is proposed, in which a novelty-based driving force as well as a fitness-based force is used to guide a population. To efficiently utilize distinct advantages of the two driving forces, weights of them are adjusted in the entire evolution process aiming to satisfy the different requirements of different evolution stages. The main characteristics of this study can be summarized as follows.

(1) The behavior novelty is introduced as a supplementary evaluation indicator for the fitness-based evaluation indicator, which can overcome shortcomings of the traditional indicator.

(2) In each generation, particles with higher fitness and higher novelty respectively saved in two archives can be regarded as two types of candidate exemplars. Based on the exemplars, a particle can extract much helpful and distinct knowledge from them.

(3) During the evolution process, a particle's learning weights of the two types of exemplars are adjusted, aiming to satisfy various requirements of different evolution stages.

The remainder of the paper is organized as follows. Related work of PSO is introduced in Section 2, while motivation and details of ADFPSO are explained in Section 3. Comparison experiments between ADFPSO and other state-of-art PSO algorithms are presented in Section 4. In addition, to figure out properties of the adjustable driving force, sensitivity analyses of them are provided in this section based on results of a set of experiments. Finally, conclusions of the study are briefly summarized in Section 6.

2. Related work

2.1. Canonical PSO

When applying PSO to optimize a problem, a population's flight trajectory can be considered as a continuous optimization process, and the position of each particle can be seen as a candidate solution. In the generation t, the particle i is characterized by two vectors, i.e., $\mathbf{X}_i^t = [x_{i,1}^t, x_{i,2}^t, ..., x_{i,D}^t]$ denotes its position while $\mathbf{V}_i^t = [v_{i,1}^t, v_{i,2}^t, ..., v_{i,D}^t]$ $(1 \leq i \leq N)$ represents its flight velocity, where D represents the dimensionality of the problem to be optimized, and N is the number of particles in the population. When performing the optimization process, the particle i adjusts its trajectory according to its own personal historical best position $\mathbf{PB}_i^t = [pb_{i,1}^t, pb_{i,2}^t, ..., pb_{i,D}^t]$ and the population's historical best position $\mathbf{GB}^t = [gb_1^t, gb_2^t, ..., gb_D^t]$. Update rules of $\mathbf{V}_i^t$ and $\mathbf{X}_i^t$ are defined as Eq. (1) and Eq. (2), respectively.

$$v_{i,j}^{t+1} = w \times v_{i,j}^t + c_1 \times r_{1,j} \times (pb_{i,j}^t - x_{i,j}^t) + c_2 \times r_{2,j} \times (gb_j^t - x_{i,j}^t) \tag{1}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{2}$$

where w represents an inertia weight; c_1 and c_2 are two acceleration coefficients; $r_{1,j}$ and $r_{2,j}$ are two random numbers uniformly distributed in the interval [0, 1]; $x_{i,j}^t$ and $v_{i,j}^t$ represent the position and velocity in the j^{th} dimension of the particle i at generation t, respectively.

2.2. Review of PSO

On the basis of research objects, the majority of PSO improvements can be generally classified into three categories, i.e., parameters adjustment, learning exemplars selection, and hybridization strategy.

2.2.1. Parameters adjustment

It can be observed from Eq.(1) that a particle's velocity is determined by three crucial parameters, i.e., w, c_1, and c_2. Thus, various researches from different perspectives are proposed to study parameters' adjustments, such as linearly adjustments [29, 32] and nonlinearly adjustments [4, 21, 33]. A fundamental motivation of this type of improvements is that a population in PSO should have favorable performance on the exploration and the exploitation in early and the later evolution stages, respectively. Some experiments have confirmed the adjustments' effectiveness. The main advantage of the adjustments is easily to be implemented, however, current searching states of a population are ignored. Thus, these improvements cannot exhibit very reliable and comprehensive performance, especially in complicated problems. To enhance a population's adaptability, various adaptive mechanisms are proposed during the last decades [37], and the favorable performance of them also have been verified by extensive studies [9, 21, 46].

2.2.2. Learning exemplars selection

In the traditional PSO, a particle selects its learning exemplars relying on two popular topologies, i.e., GPSO and LPSO. Some experiments verify that the LPSO and GPSO topologies are conducive to keeping population diversity and speeding up convergence, respectively [15]. Although basic properties of selected exemplars based on the two topologies are understood to some extend, it is quite difficult, if not impossible, to select proper exemplars for particles due to that different problems have their own distinct characteristics. Moreover, even in a specific problem, fixing exemplars for a particle may be an inappropriate choice since the optimization process is dynamic and different fitness landscapes in the problem may have their own distinct properties. Thus, to fulfill variable requirements in different evolution stages, various dynamic selections of exemplars are presented in recent decades [13, 18, 23, 26, 28]. Through the dynamic selections, not only different particles can choose their own exemplars, but also a particle can adaptively adjust its exemplars in different evolution stages.

2.2.3. Hybridization strategy

In the last decades, lots of outstanding PSO variants have been proposed. Thus, it is really quite a natural idea that integrating some of them into a framework through a reasonable approach. A typical hybrid PSO algorithm, for example, is called as an ensemble PSO (EPSO) [24], in which 5 popular PSO variants are integrated into a simple framework through a self-adaptive scheme. Furthermore, considering different optimization algorithms or search strategies have distinct advantages, many researchers pour many attentions on integrating two or more of them into a single PSO framework. For instance, some local searching strategies [23, 40, 41] and genetic operators [6, 8, 11, 45] are popular auxiliaries to increase the convergence speed and improve the population diversity. In addition, many other outstanding swarm intelligence algorithms with distinct search behaviors, including differential evolution (DE) algorithm, ant colony (ACO), and salp swarm algorithm (SSA), are also selected as efficient supplements for the canonical PSO [12, 30, 34, 37]. Experiments of these PSO variants exhibit that many shortcomings of the basic algorithms (or strategies) are partly conquered through organic integration methods [25, 30, 39].

3. ADFPSO

3.1. Motivation

When optimizing a problem with a specific objective, it is a widely accepted strategy in PSO community that evaluating a particle is with respect to the specific objective function. According to the measuring results, named as particles' fitness values, each particle i in the canonical PSO selects its exemplars, i.e., $\mathbf{PB}_i^t$ and $\mathbf{GB}^t$, to perform the corresponding search process. Intuitively, it is a very rational choice for a particle to perform the search process under a fitness-based driving force created by the elite exemplars. However, it is harmful for the population diversity. As a result, the population may be apt to fall into local optima when optimizing a complicated problem.

In contrast to the fitness-based driving force applied in PSO, a few studies of artificial life often ignore predetermined explicit objectives. A common strategy, for instance, is creating an open-ended system through pursuing behavioral novelties rather than seeking high fitness values [7]. A few studies on neural networks [10] and robotics systems [7, 17] verify promising properties of the novelty-based driving force on various applications. To simplicity, the fitness-based driving force and the novelty-based driving force are abbreviated as $\digamma_{fit}$ and $\digamma_{nov}$ in this study, respectively.

Based on these researches, $\digamma_{nov}$ is applied to make up deficiencies of $\digamma_{fit}$ in the PSO framework. During the search process, weights of the two distinct forces can be adjusted, and then particles in different evolution stages can be propelled by different driving forces. Hence, distinct requirements in different evolution stages can be satisfied to some extent.

3.2. Two driving forces in ADFPSO

From Eq. (1) we can observe that the particle i is driven by forces generated by $\mathbf{GB}^t$ and $\mathbf{PB}_i^t$, and learning weights of the two forces are determined by c_1 and c_2, respectively. Since both $\mathbf{GB}^t$ and $\mathbf{PB}_i^t$ have high fitness values, both the driving forces can be classified into a same category, i.e., $\digamma_{fit}$. In this study, two types of driving force, i.e., $\digamma_{fit}$ and $\digamma_{nov}$, are proposed to guide a population's search process intending to overcome unfavorable properties of a single driving force.

In ADFPSO, those personal historical best solutions $\mathbf{PB}_i$ $(1 \leq i \leq N)$ with high fitness and high novelty values are saved in two archives (denoted by $\mathcal{A}_{\mathcal{F}}$ and $\mathcal{A}_{\mathcal{N}}$), respectively. When the particle i updating its velocity vector $\mathbf{V}_i$, two exemplars regarded as $\mathbf{EF}_{i1}$ and $\mathbf{EN}_{i2}$ $(1 \leq i1, i2 \leq M)$ are randomly selected from $\mathcal{A}_{\mathcal{F}}$ and $\mathcal{A}_{\mathcal{N}}$, respectively, where M denotes the maximum size of the two archives. As a result, the particle can be regarded as driven by a hybrid driving force generated by the two different types of exemplars. Thus, the update of velocity in ADFPSO can be defined as Eq. (3).

$$v_{i,j}^{t+1} = w \times v_{i,j}^{t} + c_1 \times r_{1,j} \times (ef_{i1,j} - x_{i,j}^{t}) + c_2 \times r_{2,j} \times (en_{i2,j} - x_{i,j}^{t}) \tag{3}$$

where $ef_{i1,j}$ and $en_{i2,j}$ denote the jth dimension of $\mathbf{EF}_{i1}$ and $\mathbf{EN}_{i2}$, respectively; two subscripts $i1$ and $i2$ are two random integers in the interval [1, M].

3.3. Adjustable of driving forces

From the motivation of ADFPSO discussed in Section 3.1 we can see that $\digamma_{nov}$ is favorable for the exploration, while $\digamma_{fit}$ is beneficial for the exploitation. Since a population in PSO should pour different attentions on the exploration or the exploitation in different evolution stages, weights of the two driving forces must be adjusted accordingly.

From Eq. (3) we can observe that there are three parameters can determine the two driving forces' influences. The one is the maximum size M of $\mathcal{A}_{\mathcal{F}}$ and $\mathcal{A}_{\mathcal{N}}$. When M is very small, only those $\mathbf{PB}_i$ with very higher fitness can be saved in $\mathcal{A}_{\mathcal{F}}$, and other $\mathbf{PB}_i$ with relatively lower fitness are abandoned. For instance, when M=1, only one $\mathbf{PB}_i$ with the best fitness is saved in $\mathcal{A}_{\mathcal{F}}$. In this case, the best exemplar in $\mathcal{A}_{\mathcal{F}}$ can generate the greatest F_{fit} for other particles. Meanwhile, we also notice that $\mathcal{A}_{\mathcal{N}}$ with a smaller M only saves those $\mathbf{PB}_i$ with a very greater novelty, which can generate a powerful F_{nov}. However, while the population has been converged, differences among of those $\mathbf{PB}_i$ in $\mathcal{A}_{\mathcal{N}}$, measured by the novelty values, are very small. Thus, a smaller M cannot bring a greater F_{nov} in the later evolution stage. In summary, a smaller M can bring a greater F_{fit}, especially in the later evolution stage, and then strengthen the population's exploitation.

On the contrary, when M is greater, many $\mathbf{PB}_i$ with relatively higher fitness but not the highest fitness also have a change to be saved in $\mathcal{A}_{\mathcal{F}}$. Thus, these exemplars, compared with the best exemplar, in the archive can bring relative lower F_{fit} for other particles. Similarly, we can observe that a greater M enables many $\mathbf{PB}_i$ with greater novelty values have a chance to create a larger F_{nov} for particles. Thus, it can be observed that a larger M can offer a greater F_{nov}, and then enable the particle to pour much more attention on the exploration.

From the discussions aforementioned, the value of M should gradually decreases in the entire evolution process. In ADFPSO, the value of M can be described as Eq. (4).

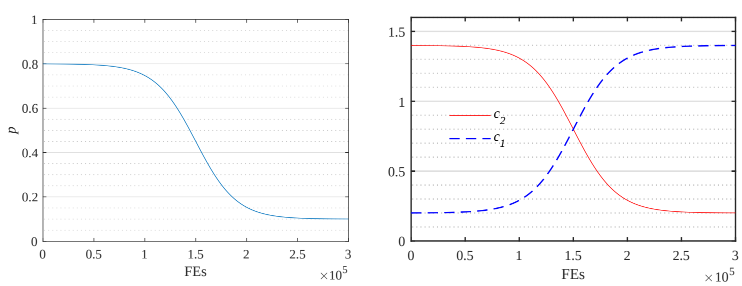
$$M = \lceil p \times N \rceil \tag{4}$$

where N is the population size, $p = 0.7 \times \frac{1}{1+e^{0.001\times\frac{FEs-MaxFEs/2}{D}}} + 0.1$, where FEs is consumed function evaluations while $MaxFEs$ is a predefined maximum function evaluations. Note that the value of p is an empirical value obtained through a set of experiments. The change process of p during an entire search process is demonstrated in Figure 1(a).

In Eq. (3), we can treat "$ef_{i1,j} - x_{i,j}^t$" and "$en_{i1,j} - x_{i,j}^t$" as F_{fit} and F_{nov}, respectively. Thus, c_1 and c_1 can be seen as a particle's two learning weights for the two driving forces, respectively. It is widely accepted strategy that enhancing the exploration and the exploitation capabilities during the early and the final evolution stages, respectively. Thus, a larger c_1 and a smaller c_2 should be set during the early evolution period aiming to enhance the global search ability of the population. On the contrary, a smaller c_1 and a greater c_2 should be applied during the final evolution period intending to speed up the convergence. Thus, in ADFPSO, definitions of c_1 and c_1 are respectively described by Eq.(5) and Eq.(6), while the change processes of them are demonstrated in Figure 1(b). Note that the value of c_1 and c_2, similar as that of p, are determined by trial-and-error.

$$c_1 = 1.2 \times \frac{1}{1 + e^{0.001 \times \frac{FEs - MaxFEs/2}{N}}} + 0.2 \tag{5}$$

$$c_2 = 1.4 - c_1 \tag{6}$$



(a) The change process of p. (b) The change processes of c_1 and c_2.

Figure 1: The change processes of p, c_1, and c_2.

3.4. Definition of novelty

In ADFPSO, the "novelty" is a new proposed concept. Hence, how to measure a particle's novelty is the first issue that needs to be addressed. In general, the term "novelty" denotes a particle's uniqueness. In this study, for simplicity, the novelty of the particle i is evaluated based on the average Euclidean distance between the particle's position $\mathbf{X}_i$ and its the K-nearest neighbors' position. Thus, the novelty value of the particle i is defined as Eq. (7).

$$\mathcal{N}_i = \frac{1}{K}\sum_{j=1}^{K} dist(\mathbf{X}_i, \mu_j) \tag{7}$$

where μ_j is the jth-nearest neighbor's position of the particle i, and $dist(\mathbf{X}_i, \mu_j)$ means an Euclidean distance between $\mathbf{X}_i$ and μ_j.

From Eq. (7) we can see that the parameter K is a crucial factor for measuring a particle's novelty. In fact, when calculating the novelty value, a particle with different K values may offer different novelties even in a same population distribution. In this work, K=2 is achieved through trial-and-error.

Although the novelty of a particle is characterized by the average distance between the particle' position and its neighbors' position, it is not to say that the measurement is the most favorable or suitable for every problem. In fact, each application may have its own distinct characteristics. Hence, it is more realistic that selecting a proper method to define the index "novelty" for a specific problem.

3.5. Pseudocode of ADFPSO

According to the introduction above mentioned, the pseudo-code of ADFPSO can be detailed as **Algorithm 1**.

4. Experimental verification and comparisons

4.1. Setup of Experiments

To figure out properties of ADFPSO and examine the performance of it, a set of experiments is conducted in this section. Furthermore, characteristics of proposed strategies applied in ADFPSO are also verified by a few experiments.

Algorithm 1. ADFPSO

Input: FEs=0, t=1, $MaxFEs$, N, K=2, $\mathcal{A}_{\mathcal{F}}$=$\mathcal{A}_{\mathcal{N}}$=$\emptyset$, M, c_1, and c_2;

Randomly initialize position vectors $\mathbf{X}_i^t$ and velocity vectors $\mathbf{V}_i^t$ $(1 \leq i \leq N)$;
While $FEs < MaxFEs$
  Evaluate $\mathbf{X}_i^t$ $(1 \leq i \leq N)$; FEs=FEs+N;
  Update $\mathbf{PB}_i^t$ and $\mathbf{GB}^t$;
  Sort $\mathbf{PB}_i^t$ according to the fitness values;
  Saved those M $\mathbf{PB}_i^t$ with higher fitness into $\mathcal{A}_{\mathcal{F}}$;
  Sort $\mathbf{PB}_i^t$ according to the novelty values measured by Eq.(7);
  Saved those M $\mathbf{PB}_i^t$ with higher novelty into $\mathcal{A}_{\mathcal{N}}$;
  Randomly choose $\mathbf{EF}_{i1}$ and $\mathbf{EN}_{i2}$ from $\mathcal{A}_{\mathcal{F}}$ and $\mathcal{A}_{\mathcal{N}}$, respectively;
  Update $\mathbf{V}_i^t$ and $\mathbf{X}_i^t$ $(1 \leq i \leq N)$ based on Eqs. (3) and (2), respectively;
  Update M, c_1, and c_2 according to Eqs. (4), (3), and (2), respectively;
  $t = t + 1$;
End While

Output: $\mathbf{GB}^t$.

4.1.1. Peer algorithms and benchmark functions

In this study, 9 other PSO algorithms are selected as competitors, and parameter settings of them are summarized in Table 1, which are the same as that in corresponding literatures.

Table 1: Parameter settings of all the peer algorithms

Algorithm	Year	Parameters Settings
SLPSO [18]	2012	$w = 0.9 \sim 0.5$,η=1.496,γ=0.01
PSODDS [13]	2013	χ=0.7298, c_1=c_2=2.05
SL-PSO [5]	2015	M=100, α=0.5, β=0.01
CCPSO-ISM [19]	2015	P=0.05, G=5, c=2.0, w=0.6
SRPSO [38]	2015	w_I=1.05, w_F=0.5, c_1=c_2=1.49445
EPSO [24]	2017	ensemble wPSO, CLPSO, FDR-PSO, HPSO-TVAC, and LIPS
XPSO [43]	2020	μ_1=μ_2=μ_3=1.35, σ=0.1, $Stag_{GB}$=10, $Stag_{max}$=5, η=0.2,p=0.5
TAPSO [44]	2020	w=0.7298, p_c=0.5, p_m = 0.02, M=$N/4$
AWPSO [22]	2021	$w = 0.9 \sim 0.4$, a=0.000035, b=0.5, c=0, d=1.5
ADFPSO	-	w=0.9$\sim$0.4, p, c_1, and c_2 are adjustable, K=2

To take a fair competition, two popular benchmark suites, i.e., CEC2013 [20] and CEC2017 [1] test suites, are chosen in this work. Note that the dimensionality of each function is D=30. To achieve statistical results, each peer algorithm is carried out 51 independent runs on each test function. The maximum number of function evaluations ($MaxFEs$) in each run is set to 300 000.

4.2. Comparison on CEC2013 test suite

In CEC2013 test suite, 28 functions can be divided into three types relying on their properties, i.e., unimodal functions (f_1-f_5), basic multimodal functions (f_6-f_{20}), and composition functions (f_{21}-f_{28}). Experimental results of the 3 different types of functions, depicted by mean value (*Mean*) of the 51 independent runs, are respectively included in Table 2 - Table 4, in which the best *Mean* results are highlighted with rectangles. Analyses of the results are presented as follows according to the functions' properties. In addition, t-test results between ADFPSO and other peer algorithms on each test functions are also presented in these tables, in which symbols "(+)", "(-)", and "(=)" denote that ADFPSO is significantly better than, significantly worse than, and almost the same as the corresponding competitors, respectively.

4.2.1. Unimodal functions (f_1-f_5)

Experiments results on f_1-f_5 demonstrated in Table 2 show that ADFPSO yields the most favorable performance on the 5 simple unimodal functions measured by the number of achieved best results, followed by SL_PSO, EPSO, and TAPSO. Moreover, the experimental results indicate that ADFPSO is the only one that can obtain the global optimal solution of f_1 in each run. In other words, ADFPSO has more stable and reliable performance on f_1.

Table 2: Solution accuracy and t-test results on unimodal functions (f_1-f_5) in CEC2013 test suite.

		SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO	ADFPSO
f_1	*Mean*	5.66e-13(+)	5.89e+01(+)	1.03e-13(+)	2.27e-13(+)	1.07e+02(+)	2.54e-13(+)	1.69e-13(+)	3.79e-13(+)	3.12e+03(+)	$\boxed{0.00e+00}$
f_2	*Mean*	2.61e+06(+)	9.49e+05(-)	6.27e+05(-)	1.00e+07(+)	3.13e+06(+)	1.65e+05(-)	$\boxed{1.25e+05(-)}$	4.59e+06(+)	1.63e+07(+)	1.60e+06
f_3	*Mean*	2.86e+09(+)	6.50e+09(+)	2.18e+07(+)	8.21e+08(+)	4.28e+09(+)	5.31e+07(+)	2.55e+08(+)	1.87e+06(+)	3.58e+10(+)	$\boxed{2.32e+05}$
f_4	*Mean*	5.05e+04(+)	5.47e+03(+)	5.78e+03(+)	3.28e+04(+)	2.16e+02(-)	$\boxed{1.69e+02(-)}$	6.52e+03(+)	1.80e+02(-)	3.12e+03(=)	3.11e+03
f_5	*Mean*	1.08e-11(=)	6.78e+01(+)	$\boxed{1.25e-13(=)}$	2.54e-13(=)	1.24e+02(+)	2.76e-13(=)	1.74e-13(=)	4.24e-13(=)	1.88e+03(+)	4.79e-13
(#)	+	4	4	3	4	4	2	3	3	4	/
(#)	-	0	1	1	0	1	2	1	1	0	/
(#)	*Best*	0	0	1	0	0	1	1	0	0	2

4.2.2. Basic multimodal functions (f_6-f_{20})

The experimental results presented in Table 3 indicate that ADFPSO exhibits more outstanding performance than other peer algorithms because it offers the highest accuracy on 7 out of the 15 simple multimodal functions, while SLPSO offers the best performance on 3 functions. Although CCPSO-ISM cannot exhibit excellent performance on the unimodal functions, it attains favorable performance on the 15 multimodal functions. In addition, TAPSO also displays outstanding performance on this type of functions.

Table 3: Solution accuracy and t-test results on basic multimodal functions (f_6-f_{20}) in CEC2013 test suite.

		SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO	ADFPSO
f_6	*Mean*	2.32e+01(-)	5.39e+01(=)	1.92e+01(-)	2.00e+01(-)	7.02e+01(+)	1.09e+01(-)	$\boxed{4.97e+01(-)}$	5.81e+01(+)	2.49e+02(+)	4.41e+01
f_7	*Mean*	1.04e+02(+)	1.02e+02(+)	5.37e+00(+)	1.03e+02(+)	4.26e+01(+)	3.31e+01(+)	7.94e+01(+)	3.95e+00(+)	1.06e+02(+)	$\boxed{3.38e-01}$
f_8	*Mean*	2.10e+01(=)	2.09e+01(=)	2.09e+01(=)	2.09e+01(=)	2.10e+01(=)	2.09e+01(=)	$\boxed{2.09e+01(=)}$	2.09e+01(=)	2.09e+01(=)	2.09e+01
f_9	*Mean*	3.09e+01(+)	2.59e+01(+)	9.84e+00(+)	3.02e+01(+)	1.78e+01(+)	2.23e+01(+)	2.13e+00(+)	9.90e+00(+)	2.09e+01(-)	$\boxed{5.32e+01}$
f_{10}	*Mean*	3.49e-01(+)	7.38e+01(+)	2.49e-01(+)	1.97e+00(+)	8.98e+01(+)	1.21e-01(+)	7.79e-02(+)	9.83e-02(+)	6.42e+02(+)	$\boxed{4.15e-02}$
f_{11}	*Mean*	1.38e-13(-)	8.07e+01(+)	1.46e+01(+)	$\boxed{8.15e-14(-)}$	3.78e+01(+)	1.13e-13(-)	1.25e-13(-)	1.69e+01(+)	1.05e+02(+)	9.98e+00
f_{12}	*Mean*	1.17e+02(+)	1.49e+02(+)	1.59e+02(+)	2.59e+02(+)	5.39e+01(+)	5.99e+01(+)	7.59e+01(+)	3.87e+01(=)	1.18e+02(+)	$\boxed{3.06e+01}$
f_{13}	*Mean*	1.65e+02(+)	2.49e+02(+)	1.58e+02(+)	2.78e+02(+)	1.18e+02(+)	1.07e+02(+)	1.65e+02(+)	1.07e+02(+)	1.87e+02(+)	$\boxed{4.31e+01}$
f_{14}	*Mean*	$\boxed{6.86e-02(-)}$	1.99e+03(=)	1.02e+03(+)	6.90e+02(-)	1.37e+03(-)	3.28e+01(-)	3.49e-01(-)	1.50e+03(-)	1.65e+03(-)	2.06e+03
f_{15}	*Mean*	4.29e+03(+)	4.05e+03(+)	4.76e+03(+)	3.85e+03(+)	3.03e+03(=)	3.69e+03(+)	3.77e+03(+)	5.39e+03(+)	4.09e+03(+)	$\boxed{3.00e+03}$
f_{16}	*Mean*	1.00e+00(-)	9.30e-01(-)	2.38e+00(=)	$\boxed{8.60e-01(-)}$	2.51e+00(=)	1.89e+00(-)	1.18e+00(-)	2.41e+00(=)	2.01e+00(-)	2.42e+00
f_{17}	*Mean*	$\boxed{3.04e+01(-)}$	1.07e+02(-)	1.67e+02(+)	3.59e+01(-)	6.23e+01(-)	3.65e+01(-)	3.09e+01(-)	7.70e+01(-)	9.33e+01(-)	1.51e+02
f_{18}	*Mean*	1.42e+02(-)	1.60e+02(-)	1.91e+02(+)	1.74e+02(-)	1.98e+02(+)	$\boxed{8.29e+01(-)}$	9.86e+01(-)	1.94e+02(+)	1.72e+02(-)	1.84e+02
f_{19}	*Mean*	$\boxed{9.82e-01(-)}$	9.81e+00(+)	3.40e+00(-)	1.28e+00(-)	3.90e+00(=)	1.84e+00(-)	1.44e+00(-)	2.49e+00(-)	6.71e+03(=)	4.79e+00
f_{20}	*Mean*	1.17e+01(+)	1.39e+01(+)	1.33e+01(+)	1.47e+01(+)	1.38e+01(+)	1.11e+01(+)	1.24e+01(+)	1.19e+01(+)	1.49e+01(+)	$\boxed{1.07e+01}$
(#)	+	7	9	11	7	9	7	7	9	8	/
(#)	-	7	3	2	7	2	7	7	3	5	/
(#)	*Best*	3	0	0	2	0	1	2	0	0	7

4.2.3. Composition functions (f_{21}-f_{28})

Generally, composition functions are very complex optimization problems. In this part, an experiment is conducted to explore characteristics of all the peer algorithms on this type of functions. The comparison results presented in Table 4 manifest that ADFPSO offers the most outstanding performance on 4 out of the 8 composition functions, measured by values of $Mean$, followed by CCPSO-ISM and XPSO. Although SLPSO yields the second best performance on the 15 simple multimodal functions (f_6-f_{20}), it cannot achieve the most favorable performance on any composition functions. In contrast, CCPSO-ISM exhibits the second best performance though it yields unsatisfied result on f_1-f_5. The reliable performance of ADFPSO and CCPSO-ISM on multimodal problems and composition problems verifies that the adjustable driving force and exchange helpful information with multiple particles is beneficial for optimizing multimodal functions.

Table 4: Solution accuracy and t-test results on composition functions (f_{21}-f_{28}) in CEC2013 test suite.

		SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO	ADFPSO
f_{21}	*Mean*	3.09e+02(=)	3.29e+02(=)	2.82e+02(-)	$\boxed{1.75e+02(-)}$	3.47e+02(=)	2.84e+02(-)	3.13e+02(=)	3.09e+02(=)	4.95e+02(+)	3.33e+02
f_{22}	*Mean*	1.16e+02(-)	2.39e+03(+)	5.81e+02(-)	6.51e+02(-)	1.66e+03(+)	1.50e+02(-)	1.10e+02(-)	$\boxed{1.03e+02(-)}$	2.06e+03(+)	7.92e+02
f_{23}	*Mean*	4.85e+03(+)	5.01e+03(+)	4.66e+03(+)	4.94e+03(+)	3.29e+03(+)	3.99e+03(+)	4.57e+03(+)	4.73e+03(+)	4.50e+03(+)	$\boxed{1.97e+03}$
f_{24}	*Mean*	2.74e+02(+)	2.77e+02(+)	2.40e+02(+)	2.87e+02(+)	2.60e+02(+)	2.47e+02(+)	2.63e+02(+)	2.44e+02(+)	2.76e+02(+)	$\boxed{2.00e+02}$
f_{25}	*Mean*	2.90e+02(+)	2.98e+02(+)	2.54e+02(+)	3.15e+02(+)	2.88e+02(+)	2.86e+02(+)	2.85e+02(+)	2.80e+02(+)	2.95e+02(+)	$\boxed{2.23e+02}$
f_{26}	*Mean*	2.32e+02(=)	2.45e+02(=)	2.73e+02(+)	$\boxed{2.01e+02(-)}$	2.74e+02(+)	2.12e+02(=)	3.07e+02(+)	2.93e+02(+)	3.36e+02(+)	2.24e+02
f_{27}	*Mean*	1.03e+03(+)	9.69e+02(+)	4.98e+02(+)	6.54e+02(+)	7.95e+02(+)	7.38e+02(+)	8.49e+02(+)	4.80e+02(+)	9.35e+02(+)	$\boxed{3.07e+02}$
f_{28}	*Mean*	4.28e+02(+)	1.19e+03(+)	3.19e+02(+)	$\boxed{2.44e+02(=)}$	6.45e+02(+)	2.84e+02(=)	2.71e+02(=)	1.82e+03(=)	1.89e+03(+)	2.73e+02
(#)	+	5	6	6	4	7	4	5	5	8	/
(#)	-	1	0	2	3	0	2	1	1	0	/
(#)	*Best*	0	0	0	3	0	0	0	1	0	4

4.2.4. Statistical results on CEC2013 test suite

In this part, two popular statistical tests are applied to examine performance of all the peer algorithms. Concretely, a two-tailed t-test is used to find out whether there is significant difference between ADFPSO's performance and other competitors' performance on each test function; and a Friedman-test is adopted to examine overall performance of all the peer algorithms. Note that, the significance level of the two statistical tests is set as α=0.05 in this study.

1) t-test results. The t-test results on the 3 different types of functions between ADFPSO and other 9 competitors are demonstrated in Table 2, Table 3, and Table 4, respectively. Moreover, based on the results presented in the 3 tables, the overall results of the t-test on the test suite are shown in Table 5, in which symbols "Sum(+)","Sum(=)", "Sum(-)" denote the number of "(+)", "(=)", and "(-)" in the tables, respectively. The comprehensive performance (CP) (the lower the better) is equal to "Sum(+)" minus "Sum(-)".

From the results presented in Table 2 - Table 4 we can see that ADFPSO dominates all the other 9 peer algorithms on the majority of the 5 unimodal functions except EPSO. Although EPSO cannot yield a favorable property on this kind of functions in terms of the values of *Mean*, it exhibits a same performance as ADFPSO because it is significantly better than ADFPSO on 2 test functions. On the 15 basic multimodal functions, both EPSO and SLPSO display a comparable result with ADFPSO since the numbers of functions that they dominating ADFPSO and they dominated by ADFPSO are the same. Thus, we can observe that organic integrating various PSO variants in EPSO can offer stable and reliable performance. On the complicated functions, i.e., f_{21} - f_{28}, ADFPSO achieves more outstanding performance than all the competitors. The overall results of t-test demonstrated in Table 5 show that ADFPSO is significantly better than all the peer algorithms on more than half of the functions, except EPSO, which is significantly worse than ADFPSO on 13 out of the 28 functions. The favorable performance of ADFPSO on CEC2013 test suite, especial on the multimodal functions and the composition functions, testifies that the new introduced hybrid driving force plays positive performance on enhancing the exploration ability. In addition, the outstanding property of EPSO manifests that organically integrating different excellent PSOs can bring very reliable and comprehensive performance on different types of functions.

Table 5: Overall results of t-test on CEC2013 test suite

ADFPSO v.s.	SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO
Sum(+)	16	19	20	15	20	13	15	17	20
Sum(=)	4	5	3	3	5	4	4	6	3
Sum(-)	8	4	5	10	3	11	9	5	5
CP	8	15	15	5	17	2	6	12	15

2) Friedman-test results. In the section, a Friedman-test is conducted to further illustrate comprehensive performance of the 10 algorithms. The results are presented in Table 6, in which the algorithms are sorted in ascending order of values of rankings (the lower the better). In addition, Friedman-tests on the 3 different types of functions are also separately carried out, and the results are also presented in Table 6.

Table 6: Friedman-test on CEC2013 test suite

Average	Overall		Unimodal Functions		Basic Multimodal Functions		Composition Functions	
Rank	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	EPSO	3.52	SL_PSO	3.20	EPSO	3.50	ADFPSO	3.38
2	ADFPSO	3.95	EPSO	3.20	TAPSO	3.73	XPSO	3.75
3	TAPSO	4.29	ADFPSO	3.60	ADFPSO	4.37	EPSO	3.75
4	XPSO	4.77	TAPSO	3.80	SLPSO	5.07	SL_PSO	3.94
5	SL_PSO	4.86	XPSO	4.60	XPSO	5.37	CCPSO-ISM	5.25
6	CCPSO-ISM	5.57	CCPSO-ISM	6.20	CCPSO-ISM	5.53	TAPSO	5.63
7	SLPSO	5.79	PSODDS	7.00	SL_PSO	5.90	SLPSO	6.25
8	SRPSO	6.61	SLPSO	7.20	SRPSO	6.57	SRPSO	6.31
9	PSODDS	7.45	SRPSO	7.20	PSODDS	7.13	PSODDS	8.31
10	AWPSO	8.21	AWPSO	9.00	AWPSO	7.83	AWPSO	8.44

From Table 6 we see that EPSO yields the most promising result measured by the overall performance, followed by ADFPSO, TAPSO, and XPSO. Although ADFPSO has fared marginally worsen than EPSO, the implementation of ADFPSO is significantly simpler than EPSO, because there are 5 PSO variants involved in EPSO while only 3 time-varying parameters are applied in ADFPSO. Moreover, although ADFPSO does not offer the most favorable results on the unimodal and basic multimodal functions, it attains the most outstanding property on the complicated composition functions. Although ADFPSO and AWPSO share a common strategy, i.e. adjusting of c_1 and c_2, ADFPSO overwhelming dominates AWPSO on all the three types of functions. Thus, we can draw a preliminary conclusion that the adjustable hybrid driving force can bring many positive properties for ADFPSO, including enhancing speeding up convergence and keeping population diversity. In addition, the proposed adjustable driving force can demonstrate very competitive and outstanding properties on those complicated composition functions.

4.3. Comparison on CEC2017 test suite

In CEC2017 test suite, 30 functions can be divided into 4 distinct classes, i.e., unimodal functions (F_1-F_3), simple multimodal functions (F_4-F_{10}), hybrid functions (F_{11}-F_{20}), and composition functions (F_{21}-F_{30}) based on their properties. The comparison results, demonstrated in Table 7 - Table 9, and corresponding analyses are presented as follows according to the functions' properties.

4.3.1. Unimodal functions (F_1-F_3)

From Table 7 we notice that EPSO achieves the best result on 2 out of the 3 unimodal functions measured by *Mean* values, while SRPSO achieves the best result on one function. Although ADFPSO displays very favorable performance on unimodal functions in CEC2013 test suite, it does not exhibit reliable performance on F_1-F_3 in CEC2017 test suite.

Table 7: Solution accuracy and t-test results on unimodal functions in CEC2017 test suite.

		SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO	ADFPSO
F_1	*Mean*	7.13e+03(+)	4.69e+06(=)	4.19e+03(+)	5.17e+03(+)	8.74e+07(+)	$\boxed{9.95e+01(-)}$	2.85e+03(=)	4.69e+03(+)	2.02e+09(+)	1.86e+03
F_2	*Mean*	4.58e+12(-)	6.09e+20(=)	1.37e+11(-)	9.24e+11(-)	5.00e+29(=)	$\boxed{2.09e+10(-)}$	3.61e+12(-)	2.06e+12(-)	1.05e+34(=)	3.60e+14
F_3	*Mean*	3.76e+04(+)	2.79e+02(+)	8.30e+03(+)	4.91e+04(+)	$\boxed{3.33e-04(-)}$	3.82e+00(-)	9.03e+01(=)	4.81e-03(-)	2.33e+03(+)	4.82e+01
(#)	+	2	1	2	2	1	0	0	1	2	/
(#)	-	1	0	1	1	1	3	1	2	0	/
(#)	*Best*	0	0	0	0	1	2	0	0	0	0

4.3.2. Simple multimodal functions (F_4-F_{10})

From Table 8 we can see that ADFPSO displays the most outstanding results on the 7 simple multimodal functions. Concretely, it attains the best result on 3 out of the 7 functions, followed by TAPSO, EPSO, SRPSO, and SL_PSO. Although EPSO yields the most promising characteristics on the 3 simple functions, it only achieves the best result on one simple multimodal function.

Table 8: Solution accuracy and t-test results on simple multimodal functions in CEC2017 test suite.

		SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO	ADFPSO
F_4	*Mean*	6.96e+01(-)	9.10e+01(=)	7.94e+01(-)	7.56e+01(-)	1.15e+02(+)	5.40e+01(-)	$\boxed{2.78e+01(-)}$	1.15e+02(+)	2.10e+02(+)	9.09e+01
F_5	*Mean*	6.58e+01(+)	1.08e+02(+)	2.32e+01(+)	1.33e+02(+)	5.03e+01(+)	4.55e+01(+)	5.71e+01(+)	4.95e+01(+)	7.99e+01(+)	$\boxed{1.53e+01}$
F_6	*Mean*	3.34e-05(=)	1.16e+01(+)	3.21e-06(=)	1.20e-01(+)	4.71e-01(+)	$\boxed{5.13e-13(=)}$	1.30e-02(+)	5.20e-02(+)	2.06e+00(+)	1.75e-06
F_7	*Mean*	1.13e+02(+)	1.33e+02(+)	1.88e+02(+)	1.04e+02(+)	$\boxed{7.55e+01(=)}$	8.76e+01(=)	8.58e+01(=)	8.58e+01(=)	1.08e+02(+)	8.42e+01
F_8	*Mean*	6.20e+01(+)	9.09e+01(+)	2.03e+01(+)	1.35e+02(+)	5.30e+01(+)	4.85e+01(+)	6.23e+01(+)	4.37e+01(+)	7.07e+01(+)	$\boxed{1.40e+01}$
F_9	*Mean*	2.01e+02(+)	1.32e+03(+)	1.03e-01(+)	5.82e+03(+)	4.37e+00(+)	6.20e+01(+)	1.24e+02(+)	7.47e+00(+)	2.37e+02(+)	$\boxed{0.00e+00}$
F_{10}	*Mean*	2.81e+03(+)	3.22e+03(+)	$\boxed{1.02e+03(-)}$	2.79e+03(+)	2.39e+03(+)	1.94e+03(+)	2.53e+03(+)	2.65e+03(+)	3.03e+03(=)	1.57e+03
(#)	+	5	6	4	6	6	4	5	6	6	/
(#)	-	1	0	2	1	0	1	1	0	0	/
(#)	*Best*	0	0	1	0	1	1	1	0	0	3

4.3.3. *Hybrid functions and Composition functions (F_{11}-F_{30})*

The comparison results on the hybrid and composition functions, which can be regarded as two types of complicated functions, are demonstrated in Table 9. On these complicated functions, ADFPSO yields the most outstanding performance because it attains the best result on 7 out of the 20 complicated functions. Although EPSO cannot yield favorable characteristics on simple multimodal functions in the test suite, it displays very promising properties on the complicated mulitmodal problems. Concretely, EPSO offers the most outstanding results on 5 out of the 20 complicated functions. Furthermore, SL_PSO and TAPSO also exhibit reliable performance on these functions.

Table 9: Solution accuracy and t-test results on hybrid and composition functions in CEC2017 test suite.

		SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO	ADFPSO
F_{11}	*Mean*	7.90e+01(+)	1.34e+02(+)	3.71e+01(=)	1.51e+02(+)	9.69e+01(+)	$\boxed{2.72e+01(-)}$	8.28e+01(+)	9.56e+01(+)	1.86e+02(+)	3.55e+01
F_{12}	*Mean*	7.81e+05(+)	1.57e+06(+)	5.08e+04(-)	6.75e+05(+)	5.80e+05(=)	3.16e+05(=)	$\boxed{2.52e+05(-)}$	2.52e+05(=)	4.89e+07(+)	2.71e+05
F_{13}	*Mean*	1.62e+04(=)	1.11e+05(=)	1.31e+04(=)	2.01e+03(-)	1.59e+06(=)	$\boxed{5.05e+02(-)}$	1.52e+04(=)	1.19e+04(-)	9.12e+06(+)	2.90e+04
F_{14}	*Mean*	2.15e+04(+)	9.52e+03(+)	1.72e+04(+)	2.96e+04(+)	9.97e+03(+)	1.61e+04(+)	2.17e+03(+)	4.92e+03(+)	2.94e+04(+)	$\boxed{1.58e+03}$
F_{15}	*Mean*	8.97e+03(=)	7.38e+03(=)	2.60e+03(-)	2.56e+02(-)	6.78e+03(=)	$\boxed{2.28e+02(-)}$	5.29e+03(-)	6.35e+03(=)	2.67e+04(=)	2.22e+04
F_{16}	*Mean*	6.26e+02(+)	1.08e+03(+)	$\boxed{2.16e+02(-)}$	6.59e+02(+)	5.36e+02(+)	5.84e+02(+)	8.69e+02(+)	5.83e+02(+)	8.02e+02(+)	3.61e+02
F_{17}	*Mean*	1.77e+02(+)	5.26e+02(+)	$\boxed{6.74e+01(-)}$	2.39e+02(+)	2.43e+02(+)	1.74e+02(+)	2.98e+02(+)	1.59e+02(+)	2.83e+02(+)	1.08e+02
F_{18}	*Mean*	3.33e+05(+)	9.69e+04(=)	1.30e+05(+)	1.40e+05(+)	1.75e+05(+)	1.44e+05(+)	$\boxed{3.15e+04(-)}$	1.53e+05(+)	2.93e+05(+)	7.74e+04
F_{19}	*Mean*	1.29e+04(+)	1.59e+04(=)	3.81e+03(=)	$\boxed{7.92e+01(-)}$	3.60e+04(=)	2.36e+02(-)	7.86e+03(+)	5.79e+03(=)	4.08e+05(+)	5.77e+03
F_{20}	*Mean*	1.96e+02(=)	5.69e+02(+)	$\boxed{1.33e+02(-)}$	3.42e+02(+)	2.03e+02(+)	2.21e+02(+)	3.35e+02(+)	1.84e+02(=)	2.50e+02(+)	1.84e+02
F_{21}	*Mean*	2.59e+02(+)	3.15e+02(+)	$\boxed{2.20e+02(=)}$	2.67e+02(+)	2.53e+02(+)	2.45e+02(+)	2.61e+02(+)	2.43e+02(+)	2.79e+02(+)	2.22e+02
F_{22}	*Mean*	1.36e+03(+)	1.44e+03(+)	2.16e+02(=)	9.89e+02(+)	4.26e+02(+)	1.00e+02(=)	8.22e+02(+)	3.45e+02(+)	1.86e+03(+)	$\boxed{1.00e+02}$
F_{23}	*Mean*	4.24e+02(+)	5.17e+02(+)	3.69e+02(+)	4.53e+02(+)	4.57e+02(+)	3.98e+02(+)	4.34e+02(+)	3.92e+02(+)	5.44e+02(+)	$\boxed{3.63e+02}$
F_{24}	*Mean*	5.07e+02(+)	5.78e+02(+)	4.48e+02(+)	5.49e+02(+)	5.65e+02(+)	5.00e+02(+)	5.21e+02(+)	4.76e+02(+)	6.26e+02(+)	$\boxed{4.31e+02}$
F_{25}	*Mean*	3.91e+02(=)	4.01e+02(+)	3.88e+02(=)	3.90e+02(+)	3.94e+02(+)	$\boxed{3.86e+02(-)}$	3.92e+02(+)	3.98e+02(+)	4.34e+02(+)	3.87e+02
F_{26}	*Mean*	1.85e+03(+)	2.20e+03(+)	1.19e+03(+)	5.98e+02(+)	1.34e+03(+)	5.34e+02(=)	1.43e+03(+)	1.05e+03(+)	2.19e+03(+)	$\boxed{4.15e+02}$
F_{27}	*Mean*	5.33e+02(+)	5.58e+02(+)	5.19e+02(+)	5.19e+02(+)	5.49e+02(+)	5.13e+02(+)	5.19e+02(+)	5.34e+02(+)	5.74e+02(+)	$\boxed{5.07e+02}$
F_{28}	*Mean*	3.67e+02(-)	4.47e+02(+)	3.76e+02(-)	4.51e+02(+)	4.34e+02(+)	3.45e+02(-)	$\boxed{3.28e+02(-)}$	3.77e+02(-)	5.99e+02(+)	4.17e+02
F_{29}	*Mean*	6.15e+02(+)	9.58e+02(+)	5.18e+02(+)	7.35e+02(+)	6.31e+02(+)	5.08e+02(+)	6.97e+02(+)	5.93e+02(+)	7.70e+02(+)	$\boxed{4.69e+02}$
F_{30}	*Mean*	1.21e+04(+)	6.27e+04(=)	4.66e+03(-)	1.18e+04(+)	7.16e+03(=)	$\boxed{4.10e+03(-)}$	4.73e+03(-)	8.88e+03(=)	5.73e+05(+)	7.73e+03
(#)	+	15	15	7	17	15	10	14	13	19	/
(#)	-	1	0	7	3	0	7	5	2	0	/
(#)	*Best*	0	0	4	1	0	5	3	0	0	7

4.3.4. *Statistical results on CEC2017 test suite*

1) t-test results. The results of t-test on CEC test suite are detailed in Table 10. From the results we can see that ADFPSO significantly dominates the 6 competitors on at least 20 out of the 30 test functions, except SL_PSO, EPSO, and TAPSO. Furthermore, EPSO also verifies very promising performance on the test suite. Concretely, it outperforms ADFPSO on 11 functions, while it is dominated by ADFPSO on 14 functions. Although AWPSO and ADFPSO share a same mechanism, i.e., adjustment of c_1 and c_2, AWPSO cannot yield significant better performance than ADFPSO on any function. The distinct performance of the two algorithms manifests that the adjusted acceleration coefficients together with the hybrid driving force can exhibit more distinct advantages on different functions.

Table 10: Overall results of t-test on CEC2017 test suite

ADFPSO v.s.	SLPSO	PSODDS	SL_PSO	CCPSO-ISM	SRPSO	EPSO	TAPSO	XPSO	AWPSO
Sum(+)	23	22	13	25	21	14	17	20	27
Sum(=)	4	8	7	0	8	5	6	6	3
Sum(-)	3	0	10	5	1	11	7	4	0
CP	18	22	3	20	20	3	10	16	27

2) Friedman-test results. Similar as the Friedman-test applied in CEC2013 test suite, the test in this part is also performed to verify comprehensive performance of the 10 peer PSO variants on CEC2017 test suite. From the statistic test results listed in Table 11 we notice that EPSO offers the best performance on the unimodal functions, followed by XPSO and ADFPSO. In contrast, ADFPSO yields the most promising property on both the basic multimodal funcitons and the hybrid and composition functions, on which EPSO and SL_PSO also display promising performance. The Friedman-test results on the test suite further validate the competitive performance of ADFPSO on different test functions, especially on the hybrid and composition functions.

5. Effectiveness of involved strategies

From the experiments detailed in the last section we know that the favorable performance of ADFPSO benefits from the 2 newly introduced strategies, i.e., novelty-based driving force and 3 adjustable parameters. In this section, thus, a few experiments are performed to measure the strategies' performance. Because of limitations of space, only 9 functions in CEC2013 test suite are tested in this

Table 11: Friedman-test on CEC2017 test suite

Average	Overall		Unimodal Functions		Basic Multimodal Functions		Hybrid and Composition Functions	
Rank	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	ADFPSO	2.83	EPSO	1.67	ADFPSO	2.14	ADFPSO	2.85
2	EPSO	2.95	XPSO	3.67	EPSO	3.29	SL_PSO	2.93
3	SL_PSO	3.25	ADFPSO	4.33	SL_PSO	3.57	EPSO	3.03
4	XPSO	4.45	TAPSO	4.33	TAPSO	4.86	XPSO	4.43
5	TAPSO	5.02	SL_PSO	4.67	XPSO	4.86	TAPSO	5.18
6	SRPSO	6.10	CCPSO-ISM	6.33	SRPSO	5.00	CCPSO-ISM	6.05
7	SLPSO	6.35	SRPSO	6.33	SLPSO	6.07	SLPSO	6.30
8	CCPSO-ISM	6.48	SLPSO	7.33	CCPSO-ISM	7.79	SRPSO	6.45
9	PSODDS	8.43	PSODDS	7.33	AWPSO	8.43	PSODDS	8.40
10	AWPSO	9.13	AWPSO	9.00	PSODDS	9.00	AWPSO	9.40

work, i.e., 3 unimodal functions (f_1-f_3), 3 basic multimodal functions (f_6-f_8), and 3 composition functions (f_{21}-f_{23}).

To give an overall examination of the introduced strategies, population diversity and fitness are selected as two metrics. The definition of population diversity can be described as Eq. (8).

$$div = \frac{1}{N \cdot |L|} \cdot \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{ij} - \overline{x}_j)^2} \tag{8}$$

where N denotes the population size, D is the dimensionality of the problem, $|L|$ means the length of the longest diagonal of the solution space, x_{ij} is the j^{th} value of the i^{th} particle's position vector $\mathbf{X}_i$ and $\overline{x}_j = \frac{1}{N} \cdot \sum_{i=1}^{N} x_{ij}$ denotes the average value of the j^{th} dimension of all x_{ij}.

5.1. Overall characteristics of ADFPSO

In this part, the overall characteristics of ADFPSO is analyzed by the 3 types of functions, the results of which are illustrated by Figure 2, Figure 3, and Figure 4, respectively. Note that, GPSO and LPSO in the figures denote the canonical PSO with global and local topological versions, respectively.

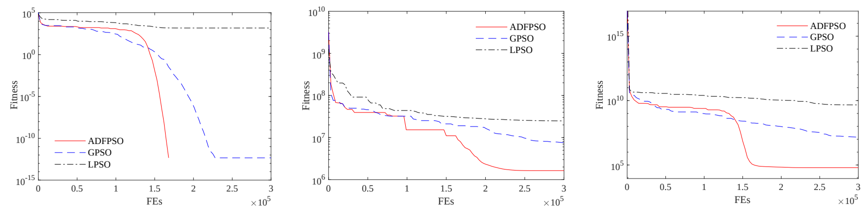
The results demonstrated in Figure 2 show that LPSO offers the lowest convergence speed in the evolution process, while GPSO is slightly better than LPSO, in terms of the convergence speed. On the contrary, ADFPSO offers the same performance as GPSO in the initial evolution stage measured by the population diversity. During the later evolution stage, ADFPSO exhibits the highest convergence speed, and yields the best solutions on all the 3 unimodal functions. In addition, we can find a common phenomenon that curves of the population diversity of the 3 unimodal functions are steeply declined in the middle of evolution process. Thus, an elementary conclusion can be drawn that ADFPSO has a positive effect on enhancing the population diversity and improving the convergence speed at the initial and the later evolution processes, respectively.

The comparison results illustrated by Figure 3 and Figure 4 also show similar phenomena as that in Figure 2. Concretely, ADFPSO exhibits the highest convergence speed in the later evolution process, while LPSO offers the lowest convergence speed in the whole evolution process on the majority of the test functions. Furthermore, ADFPSO also attains the best solution on 5 out of the 6 multimodal functions.

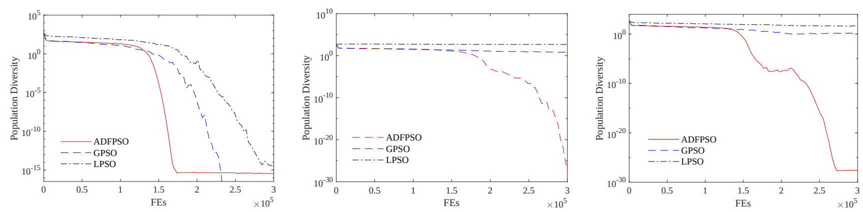
The comparison results on the three different types of functions verify that ADFPSO has very reliable comprehensive performance. Concretely, the adjustable driving force strengthens the "novelty" pursuit in the early evolution stage, which plays a positive effect on keeping population diversity. On the contrary, the adjustable driving force pay more attention on "fitness" pursuit in the later evolution stage, and then speeds up the convergence in the stage.

5.2. Effectiveness of p

In ADFPSO, the motivation of applying the hybrid driving force is to trade off the diversity and the speed of convergence. From the discussions in Section 3.3 we know that the parameter p plays a significant role in adjusting the driving force. Specifically, the change process of the p (see Figure 1(a)) shows that population should be mainly driven by $\digamma_{nov}$ in the initial evolution. Thus, the population can keep a higher diversity. On the contrary, the population should be mainly driven by $\digamma_{fit}$ in the later evolution. Thus, the convergence process can be speeded up. In this part, experiments are executed to verify the properties of the hybrid driving force on the 3 types of functions, and the convergence processes of the population are respectively illustrated by Figure 5 - Figure 7. Note that, in the figures, ADFPSO/GPSO denotes an ADFPSO



(a) f_1-fitness (b) f_2-fitness (c) f_3-fitness

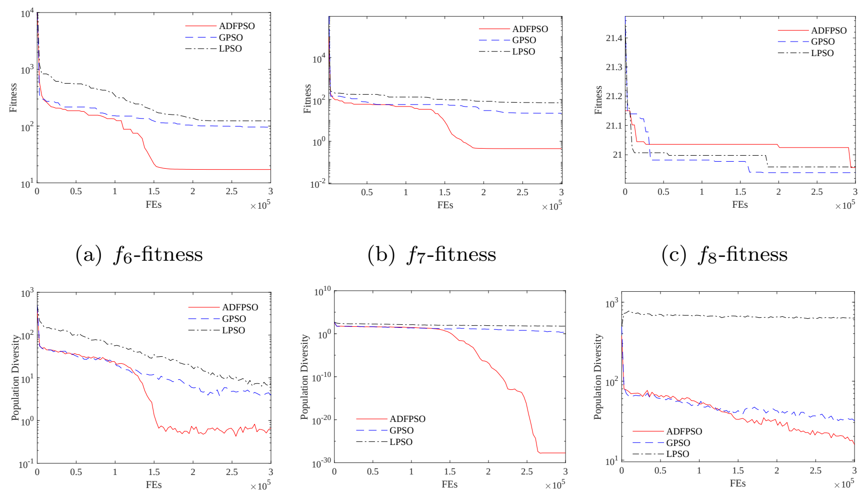


(d) f_1-population diversity (e) f_2-population diversity (f) f_3-population diversity

Figure 2: Change processes of fitness and population diversity when utilizing ADFPSO, GPSO, and LPSO to optimize the 3 unimodal functions.

variant in which $\mathbf{PB}_i$ and $\mathbf{GB}$ are two popular fitness-based driving forces applied in GPSO.

It can be observed from Figure 5 that ADFPSO significantly dominates ADFPSO/GPSO on all the 3 unimodal problems, in terms of fitness values. From the change processes of the fitness and the population diversity we notice that ADFPSO/GPSO achieves a high convergence speed on f_1, which is a simple unimodal function. Moreover, it also displays higher convergence speeds than ADFPSO during the early evolution stage on all the unimodal problems. On the contrary, ADFPSO displays the lower convergence speed than ADFPSO/GPSO on the 3 unimodal functions in the initial evolution, while ADFPSO exhibits a higher convergence speed in the later evolution. From the distinct performance we can observe that it is $\mathcal{F}_{fit}$ implied in the GPSO topology enables ADFPSO/GPSO to have the fast convergence, especially in the former evolution stage. Within the later evolution process, on the contrary, the main driving force of ADFPSO switches from $\mathcal{F}_{nov}$ to $\mathcal{F}_{fit}$. Thus, ADFPSO exhibits



(a) f_6-fitness (b) f_7-fitness (c) f_8-fitness

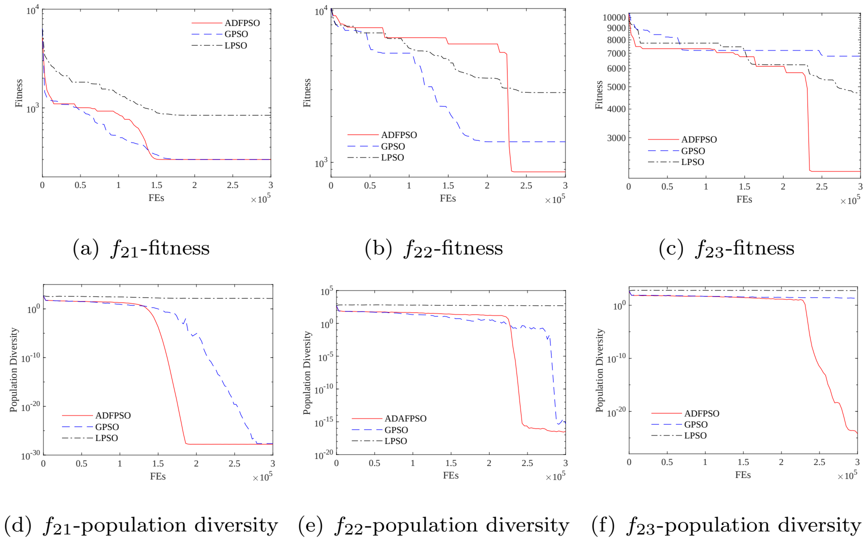
(d) f_6-population diversity (e) f_7-population diversity (f) f_8-population diversity

Figure 3: Change processes of fitness and population diversity when utilizing ADFPSO, GPSO, and LPSO to optimize the 3 basic multimodal functions.

more favorable performance on the exploitation.

Furthermore, the experiments on the basic multimoal functions and the composition functions respectively demonstrated in Figure 6 and Figure 7 also show similar phenomenon. Concretely, ADFPSO outperforms ADFPSO/GPSO, measured by the fitness value, on the majority of the test functions. Moreover, ADFPSO exhibits a lower convergence speed in the early evolution stage, while it displays a higher convergence speed during the later evolution stage. On the contrary, although ADFPSO/GPSO has higher convergence speeds and relatively accurate results in the initial evolution process, it is outperformed by ADFPSO in the later evolution process, measured by both convergence speed and solutions' accuracy.

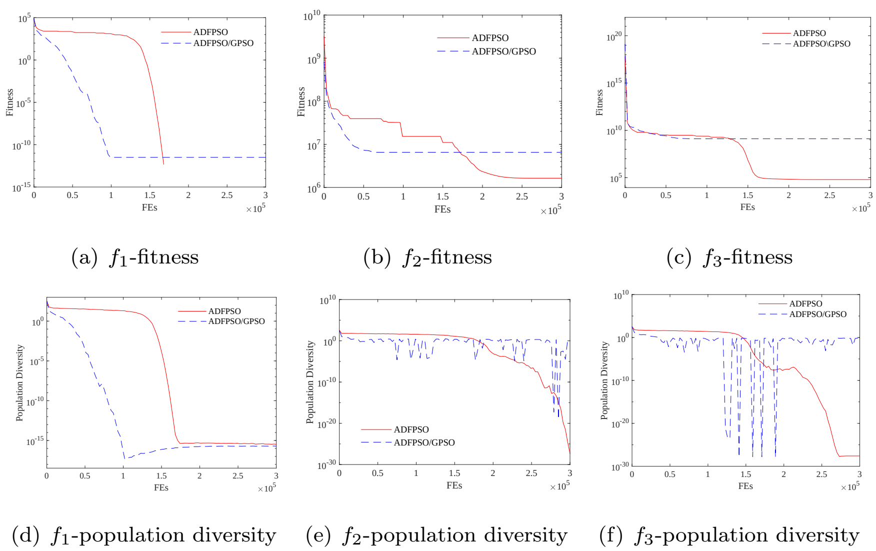
The comparison results among different functions manifest that the proposed hybrid driving force can provide a good trade off between the exploration and the exploitation in the search process, and then enables ADFPSO to yield favorable comprehensive performance.



(a) f_{21}-fitness (b) f_{22}-fitness (c) f_{23}-fitness

(d) f_{21}-population diversity (e) f_{22}-population diversity (f) f_{23}-population diversity

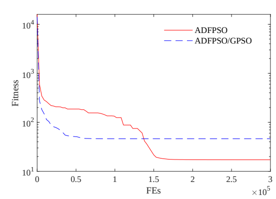
Figure 4: Change processes of fitness and population diversity when utilizing ADFPSO, GPSO, and LPSO to optimize the 3 composition functions.



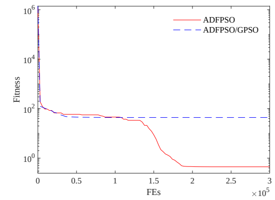
(a) f_1-fitness (b) f_2-fitness (c) f_3-fitness

(d) f_1-population diversity (e) f_2-population diversity (f) f_3-population diversity

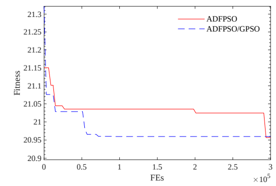
Figure 5: Performance of p on the 3 simple unimodal functions.



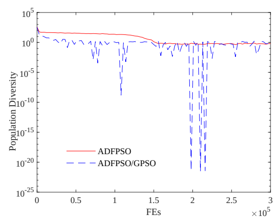
(a) f_6-fitness



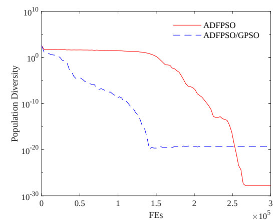
(b) f_7-fitness



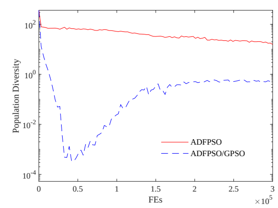
(c) f_8-fitness



(d) f_6-population diversity

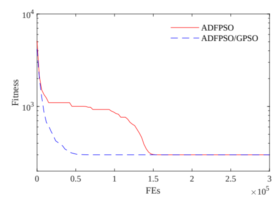


(e) f_7-population diversity

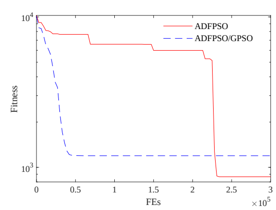


(f) f_8-population diversity

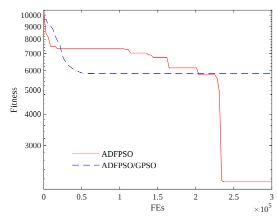
Figure 6: Performance of p on the 3 basic multimodal functions.



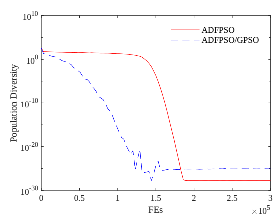
(a) f_{21}-fitness



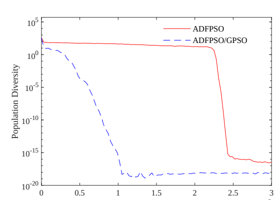
(b) f_{22}-fitness



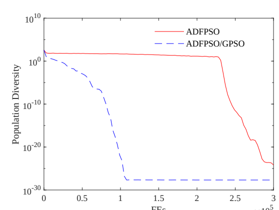
(c) f_{23}-fitness



(d) f_{21}-population diversity



(e) f_{22}-population diversity

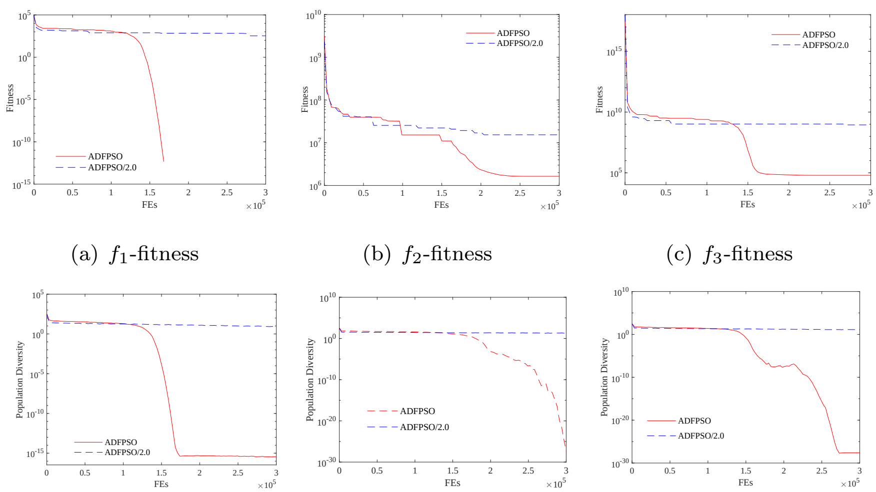


(f) f_{23}-population diversity

Figure 7: Performance of p on the 3 composition functions.

5.3. *Effectiveness of c_1 and c_2*

In ADFPSO, c_1 and c_2 deemed as two adjustable acceleration coefficients in this study are the other two important parameters introduced to tune the weights of $\digamma_{fit}$ and $\digamma_{nov}$. In this part, experiments are executed to verify characteristics of c_1 and c_2. Comparison results between ADFPSO and ADFP-SO/2.0 on the 3 different types of functions are illustrated by Figure 8, Figure 9, and Figure 10, respectively. Note that ADFPSO/2.0 denotes an ADFPSO variant, in which time-varying c_1 and c_2 are replaced by 2.0.

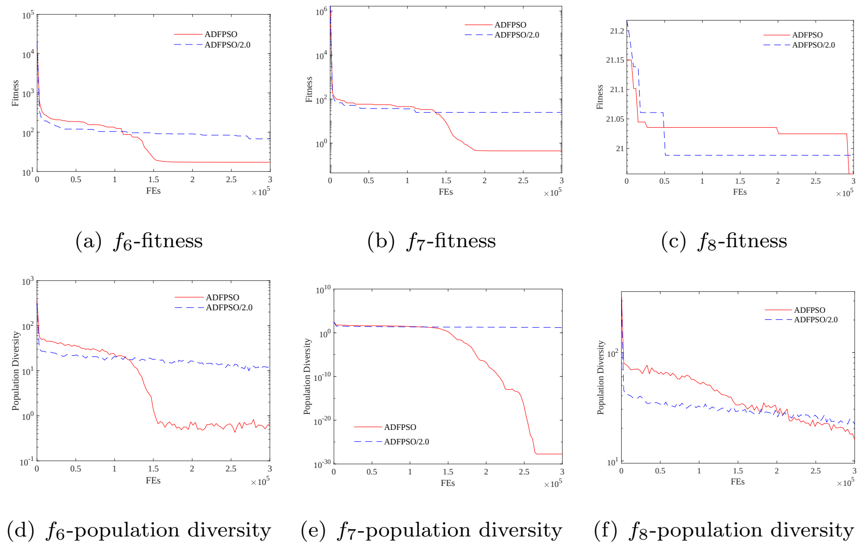


(a) f_1-fitness (b) f_2-fitness (c) f_3-fitness

(d) f_1-population diversity (e) f_2-population diversity (f) f_3-population diversity

Figure 8: Performance of c_1 and c_2 on the 3 simple unimodal functions.

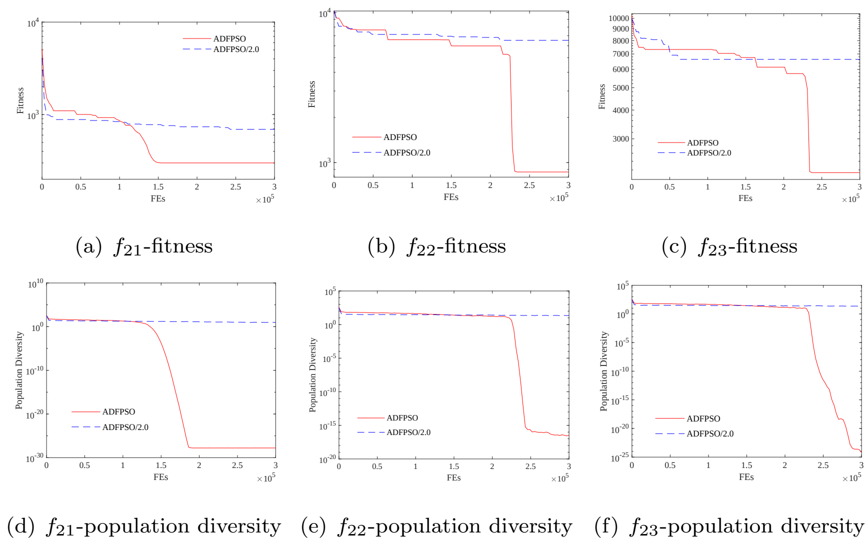
It can be observed from Figure 8 that ADFPSO/2.0 displays unfavorable performance measured by solution accuracy due to that the population of ADF-PSO/2.0 does not converge during the entire evolution process, even for the simple unimodal function f_1, though c_1=c_1=2.0 is popular applied in many PSO algorithms. The results show that the two driving forces, i.e., $\digamma_{fit}$ and $\digamma_{nov}$, with a same weight cannot meet the requirement of population convergence. In addition, the comparison results demonstrated in Figure 9 and Figure 10 also offer a common phenomena. Concretely, the population in ADFPSO/2.0 cannot



(a) f_6-fitness (b) f_7-fitness (c) f_8-fitness

(d) f_6-population diversity (e) f_7-population diversity (f) f_8-population diversity

Figure 9: Performance of c_1 and c_2 on the 3 basic multimodal functions.



(a) f_{21}-fitness (b) f_{22}-fitness (c) f_{23}-fitness

(d) f_{21}-population diversity (e) f_{22}-population diversity (f) f_{23}-population diversity

Figure 10: Performance of c_1 and c_2 on the 3 composition functions.

convergent or has a very slow convergence speed during the evolution process. However, the adjustable c_1 and c_1 enable ADFPSO to have a favorable convergence process. From the experimental results we obtain a conclusion that, with the time-varying c_1 and c_1, the weights of the two driving forces are properly adjusted in the whole evolution process. Thus, distinct requirements of various evolution stages can be satisfied.

6. Conclusion

In the PSO community, it is a widely accepted choice that applying some elites, in terms of fitness values, to drive a population. However, there is an obvious flaw that the population may easily be premature convergence when optimizing a complicated multimodal problem, in which deception is one of the common phenomena. Thus, in this study, a novelty-based driving force, which entirely ignoring the objective of a problem, was introduced to overcome the deficiency of the fitness-based driving force.

In this study, an adjustable driving force based PSO (ADFPSO) was proposed, in which a single fitness-based driving force applied in the canonical PSO is replaced by a hybrid driving force in which the fitness-based driving force and the novelty-based driving force are two basic driving forces. During the search process of the population, the fitness-based driving force is used to improve the exploitation capability, while the novelty-based driving force is applied to enhance the exploration capability. Considering distinct requirements of the different evolution stages, weights of the two driving forces are adjusted during the search process through three time-varying parameters, i.e., p, c_1, and c_2. Based on the parameters, the novelty-based driving force plays a more significant role in the initial evolution stage aiming to increase the population's exploration ability. On the contrary, the fitness-based driving force plays a more important role intending to improve the population's exploitation capability.

To testify comprehensive performance of the proposed ADFPSO, two popular test suites, i.e., CEC2013 and CEC2017 test suites, were chosen as test functions, while other 9 popular PSO variants were adopted as competitors. The comparison results verified that ADFPSO can offer outstanding and comprehensive performance on the test suites, especially on complicated multimodal functions. Moreover, properties of the newly proposed strategies involved in ADFPSO were also examined by a few experiments. Based on these experimental results, a few preliminary conclusions can be obtained. First, the novelty-based driving force has positive performance on keeping the population diversity. Second, the adjustable driving force can offer more reliable and promising characteristics than only one driving force. Last, the adjustable driving force has very favorable performance on complicated multimodal functions.

Although the favorable property of the novelty-based driving force has been verified by extensive experiments, it is dangerous to regard that it is superior to the traditional fitness-based driving force in general. In fact, the two driving forces have their own merits. Thus, it is a promising research subject that organic hybrid the two driving forces, and then adjusting their weights during the entire search process to fulfill distinct requirements of different evolution stages. To achieve the objectives, how to design an efficient "novelty" index for a specific problem is the first issue should be dealt with. Although a particle's novelty is defined as an average distance between the particle and its neighbors in this study, it does not mean that the definition is the best choice. In fact, it is more suitable and feasible that applying appropriate metrics to measure a particle's novelty according to a problem's specific property. Furthermore, another one issue should be further studied is designing a proper adaptive adjustment for the weights of the two distinct driving forces, rather than the simple time-varying adjustments in ADFPSO. Only giving enough consideration to the current stage of a population and properties of a problem's fitness landscape, can distinct virtues of various driving forces be sufficiently utilized.

References

[1] N.H. Awad, M.Z. Ali, J.J. Liang et al., Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Nanyang Technological Univ., Singapore, Tech. Rep., 2016.

[2] Y. Bi, A. Lam, H. Quan et al. A comprehensively improved particle swarm optimization algorithm to guarantee particle activity, Russ. Phys. J. 64(2021) 866-875.

[3] A. Channon, Passing the alife test: activity statistics classify evolution in geb as unbounded, in: Proceedings of the European conference on artificial life, ECAL'01, Springer. pp. 417-426.

[4] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Comput. Oper. Res. 33(3)(2004) 859-871.

[5] R. Cheng and Y. Jin, A social learning particle swarm optimiation algorithm for scalable optimization, Inf. Sci., 291(6)(2015) 43-60.

[6] P. Dziwiński, Ł. Bartczuk, A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic, IEEE Trans. Fuzzy Syst. 28(6)(2020) 1140-1154.

[7] J. Gomes, P. Urbano, A.L. Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intel. 7(2013) 115-144.

[8] Y.J. Gong, J.J. Li, Y.C. Zhou et al., Genetic learning particle swarm optimization, IEEE Trans. Cybern. 46(10)(2016) 2277-2290.

[9] K.R. Harrison, A.P. Engelbrecht, B.M. Omubuki-Berman, Self-adaptive particle swarm optimization: a review and analysis of convergence, Swarm Intel. 12(2018) 187-226.

[10] B. Inden, Y.C. Jin, R. Haschke et al., An examination of different fitness and novelty based selection methods for the evolution of neural networks, Soft Comput. 17(2013) 753-767.

[11] T. Jamrus, C. Chien, M. Gen et al., Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing, IEEE Trans. Semicond. Manuf. 31(1)(2018) 32-41.

[12] C. Jiang, J. Fu, W. Liu, Research on vehicle routing planning based on adaptive ant colony and particle swarm optimization algorithm, Int. J. ITS Res. 19(2021) 83-91.

[13] X. Jin, Y.Q. Liang, D.P. Tian et al., Particle swarm optimization using dimension selection methods, Appl. Math. Comput. 219(10)(2013) 5185-5197.

[14] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. of IEEE International Conference on Neural Network, CNN'95, Perth, Australia, 1995, pp. 1942-1948.

[15] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proc. of IEEE Congress on Evolutionary Computation, CEC'02, Honolulu, HI, May, 2002, pp. 1671-1676.

[16] N. Kumar, A.A. Shaikh, S.K. Mahato et al., Applications of new hybrid algorithm based on advanced cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations, Expert Syst. Appl. 172(2021) 114646.

[17] J. Lehman, K.O. Stanley, Abandoning objectives: evolution through the search for novelty alone, Evol. Comput. 19(2)(2011) 189-223.

[18] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man, Cybern. B, Cybern. 42(3)(2012) 627-646.

[19] Y. Li, Z.H. Zhan, S. Lin et al., Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, Inf. Sci. 293(3)(2015) 370-382.

[20] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, Nanyang Technological Univ., Singapore, Tech. Rep., 2013.

[21] H. Liu, X.W. Zhang, H. Liang et al., Stability analysis of the human behavior-based particle swarm optimization without stagnation assumption, Expert Syst. Appl. 159(2020) 113638.

[22] W.B. Liu, Z.D. Wang, Y. Yuan et al., A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE Trans. Cybern. 51(2)(2021)1085-1093.

[23] W.J. Luo, Y.Y. Qiao, X. Lin et al., Preuss, Hybridizing niching, particle swarm optimization, and evolution strategy for multimodal optimization, IEEE Trans. Cybern. early access, doi: 10.1109/TCYB.2020.3032995.

[24] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55(2017), 533-548.

[25] A. Mullai, K. Mani, Enhancing the security in RSA and elliptic curve cryptography based on addition chain using simplified swarm optimization and particle swarm optimization for mobile devices, Int. J. Inf. Tech. 13(2021) 551-564.

[26] Y. Ning, Z. Peng, Y. Dai et al., Enhanced particle swarm optimization with multi-swarm and multi-velocity for optimizing high-dimensional problems, Appl. Intell. 49(2019) 335-351.

[27] S. Phoemphon, C. So-In, N. Leelathakul, Improved distance estimation with node selection localization and particle swarm optimization for obstacle-aware wireless sensor networks, Expert Syst. Appl. 175(2021) 114773.

[28] C. Qiu, A novel multi-swarm particle swarm optimization for feature selection, Genet. Program. Evolvable Mach. 20(2019) 503-529.

[29] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8(3)(2004) 240-255.

[30] F. Redoloza, L. Li, A comparison of extremal optimization, differential evolution and particle swarm optimization methods for well placement design in groundwater management, Math. Geosci. 53(2021) 711-735.

[31] M. Shafipour, A. Rashno, S. Fadaei, Particle distance rank feature selection by particle swarm optimization, Expert Syst. Appl. 185(2)(2021) 115620.

[32] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proc. of IEEE World Congress on Computational Intelligence, Anchorage, AK, May, 1998, pp. 68-73.

[33] Y. Shi, R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: Proc. of IEEE Congress on Evolutionary Computation, CEC'01, Soul, Korea, 2001, pp. 101-106.

[34] N. Singh, S.B. Singh, E.H. Houssein, Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions, Evol. Intel. early access, doi: 10.1007/s12065-020-00486-6.

[35] S. Som-In, W. Kimpan, Enhancing of particle swarm optimization based method for multiple motifs detection in DNA sequences collections, IEEE/ACM Trans. Comput. Biol. Bioinform. 17(3)(2020) 990-998.

[36] R. Standish, Open-ended artificial evolution, Int. J. Comput. Intell. Appl. 3(2)(2003) 167-175.

[37] B. Tang, K. Xiang, M. Pang, An integrated particle swarm optimization approach hybridizing a new self-adaptive particle swarm optimization with

a modified differential evolution, Neural Comput. Applic. 32(2020) 4849-4883.

[38] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. 294(C)(2015) 182-202.

[39] G. Tian, Y. Ren, M. Zhou, Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution and particle swarm optimization combined algorithm, IEEE Trans. Intell. Transp. Syst. 17(11)(2016) 3009-3021.

[40] Y.J. Wang, Improving particle swarm optimization performance with local search for high-dimensional function optimization, Optim. Methods Softw. 25(5)(2010) 781-795.

[41] X.W. Xia, J.N. Liu, Z.B. Hu, An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space, Appl. Soft Comput. 23(5)(2014) 76-90.

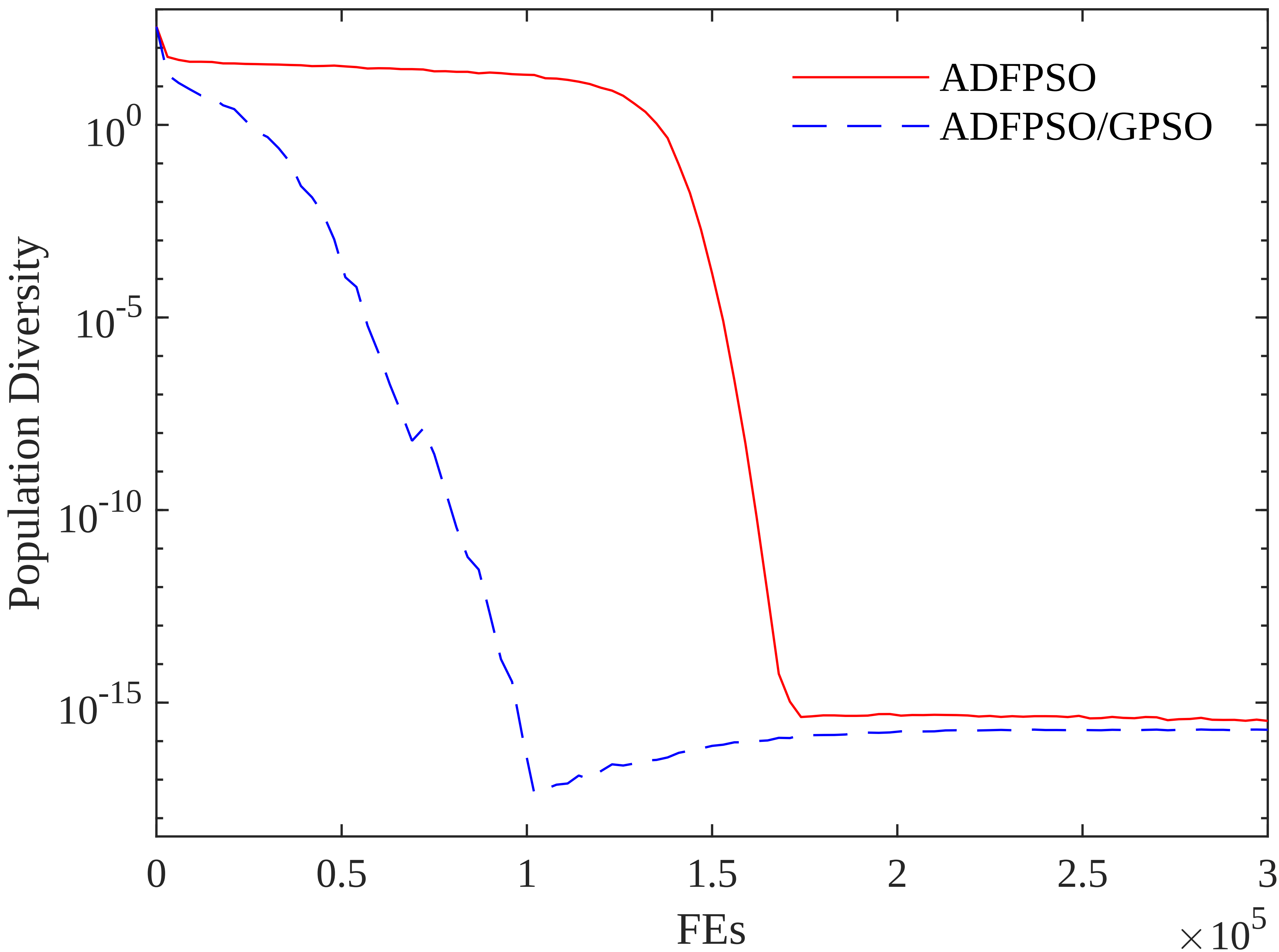
[42] X.W. Xia, Y. Xing, B. Wei et al., A fitness-based multi-rule particle swarm optimization, Swarm Evol. Compt. 44(2019) 349-364.

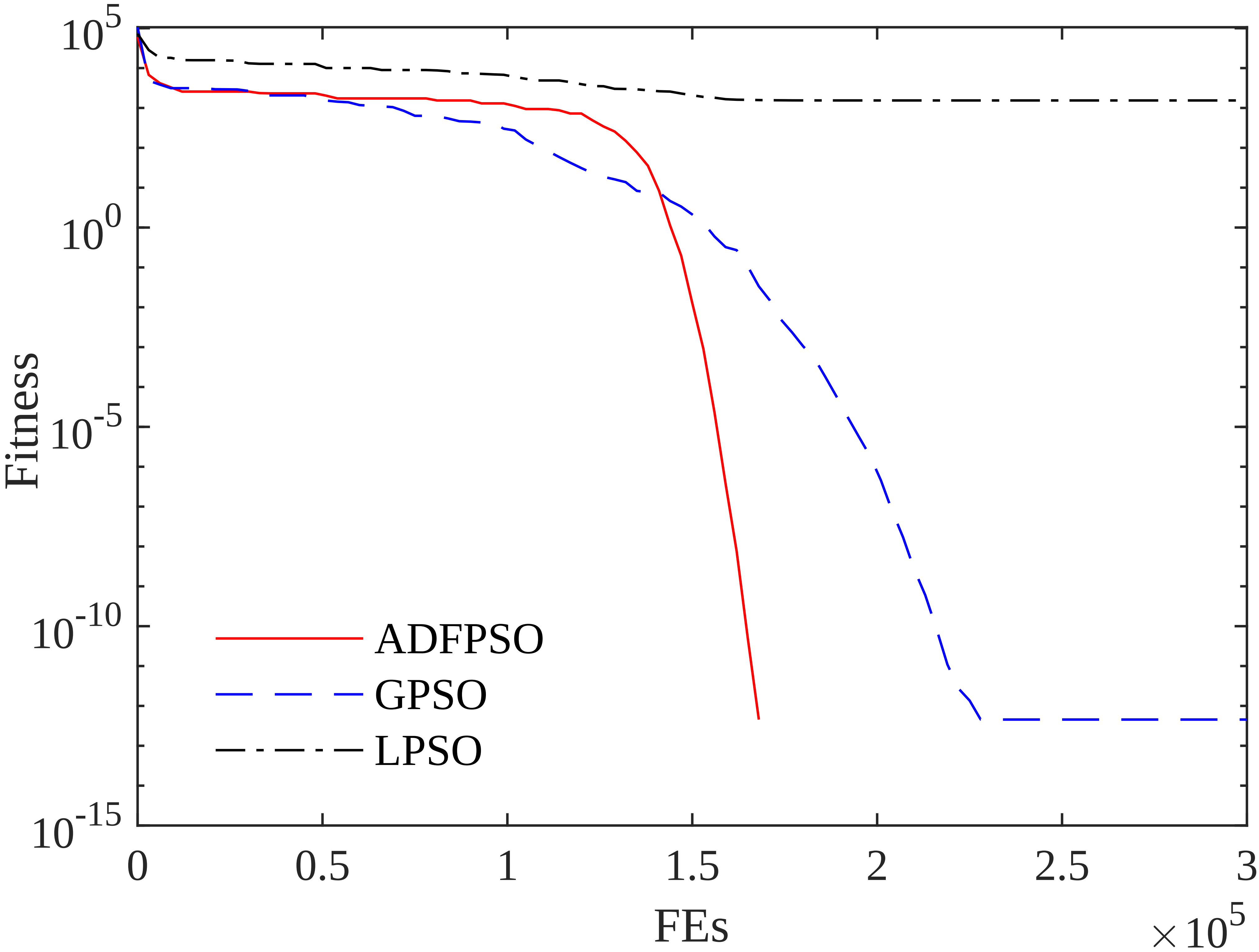
[43] X.W. Xia, L. Gui, G.L. He et al., An expanded particle swarm optimization based on multi-exemplar and forgetting ability, Inf. Sci. 508(2020) 105-120.

[44] X.W. Xia, L. Gui, F. Yu et al., Triple archives particle swarm optimization, IEEE Trans. Cybern. 50(12)(2020) 4862-4875.

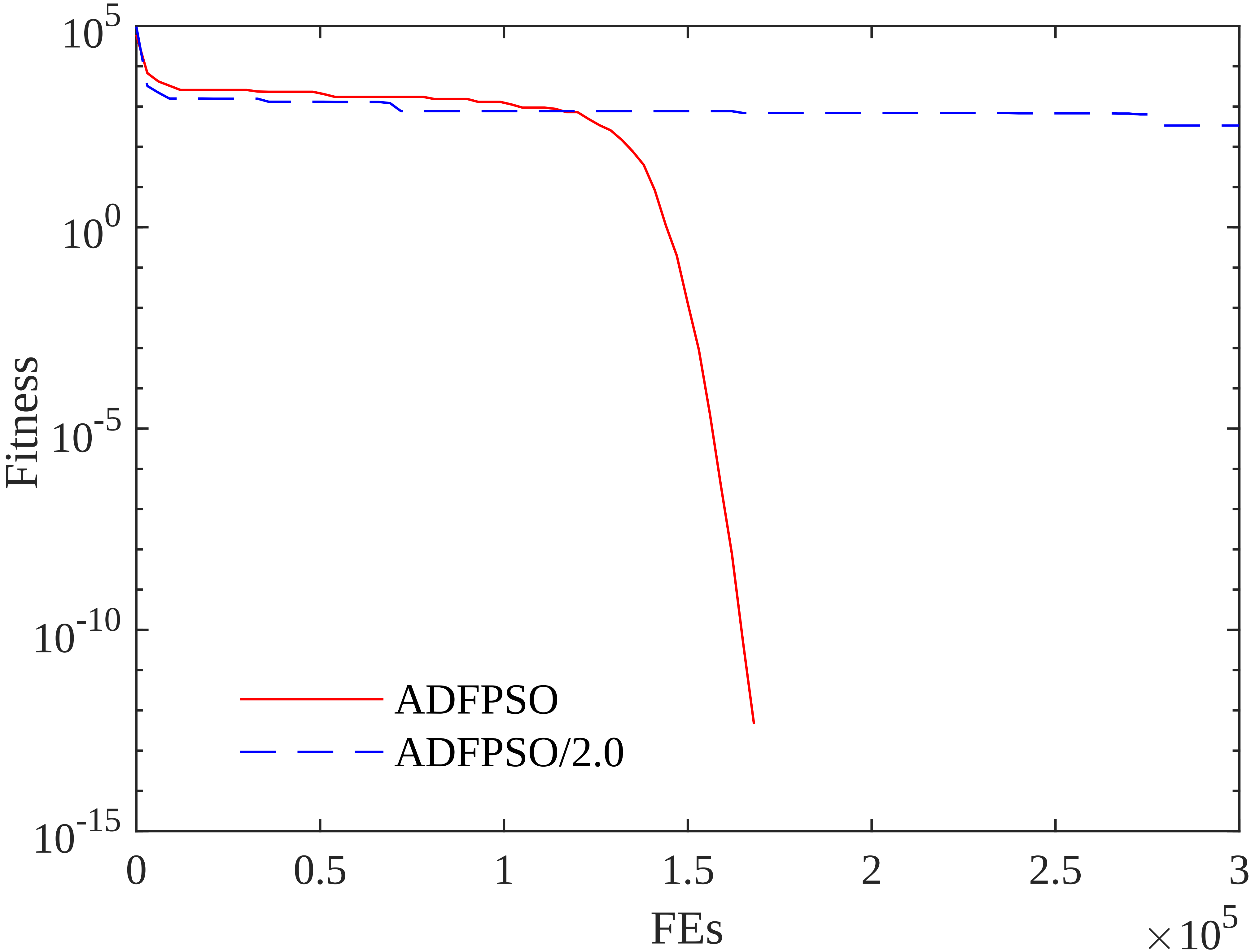
[45] F. Xiong, P. Gong, P. Jin et al., Supply chain scheduling optimization based on genetic particle swarm optimization algorithm, Cluster Comput. 22(2019) 14767-14775.

[46] N.Y. Zeng, Z.D. Wang, W.B. Liu et al., A dynamic neighborhood-based switching particle swarm optimization algorithm, IEEE Trans. Cybern. early access, doi: 10.1109/TCYB.2020.3029748.

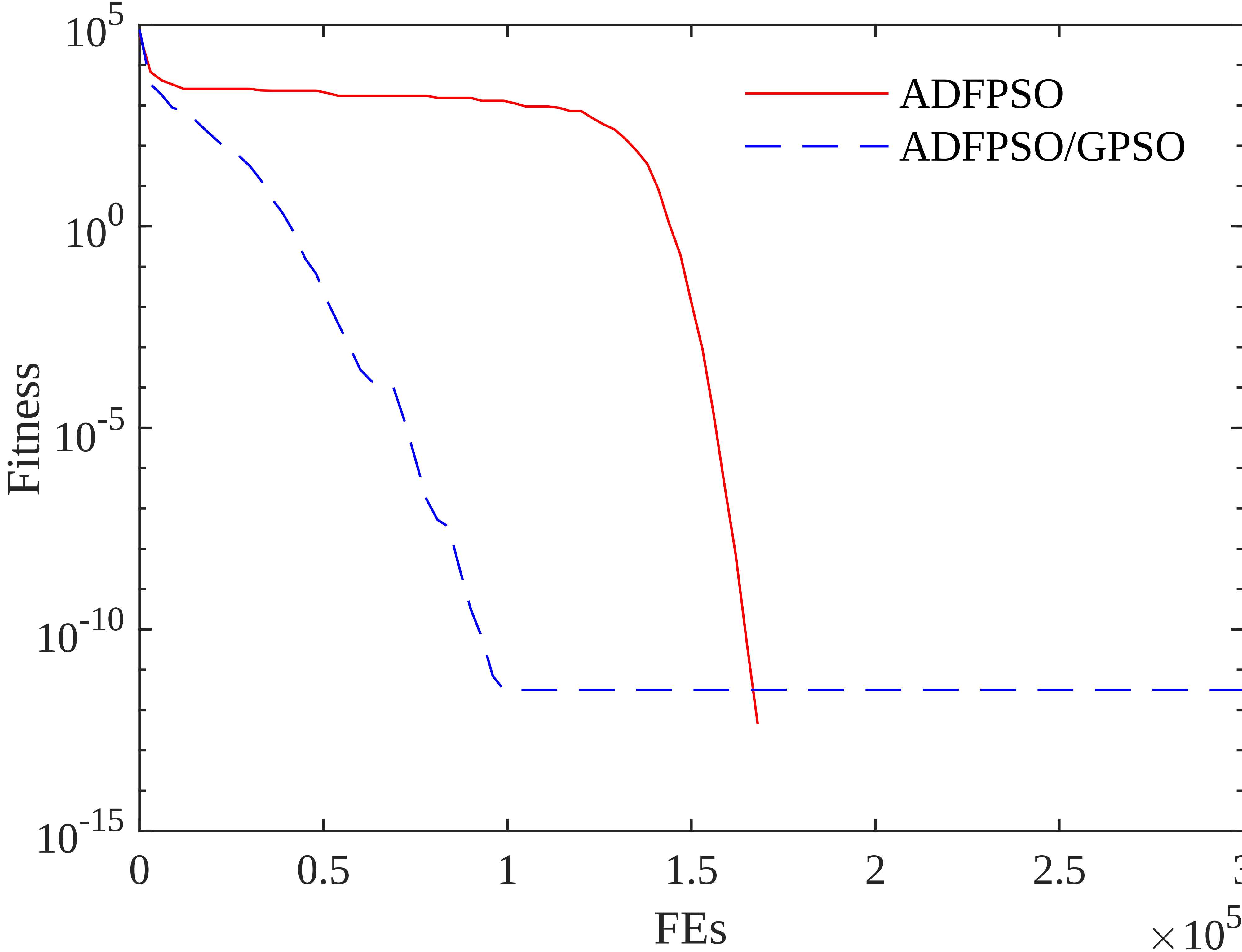

ADFPSO
ADFPSO/GPSO
Population Diversity
10^{0}
10^{-5}
10^{-10}
10^{-15}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$



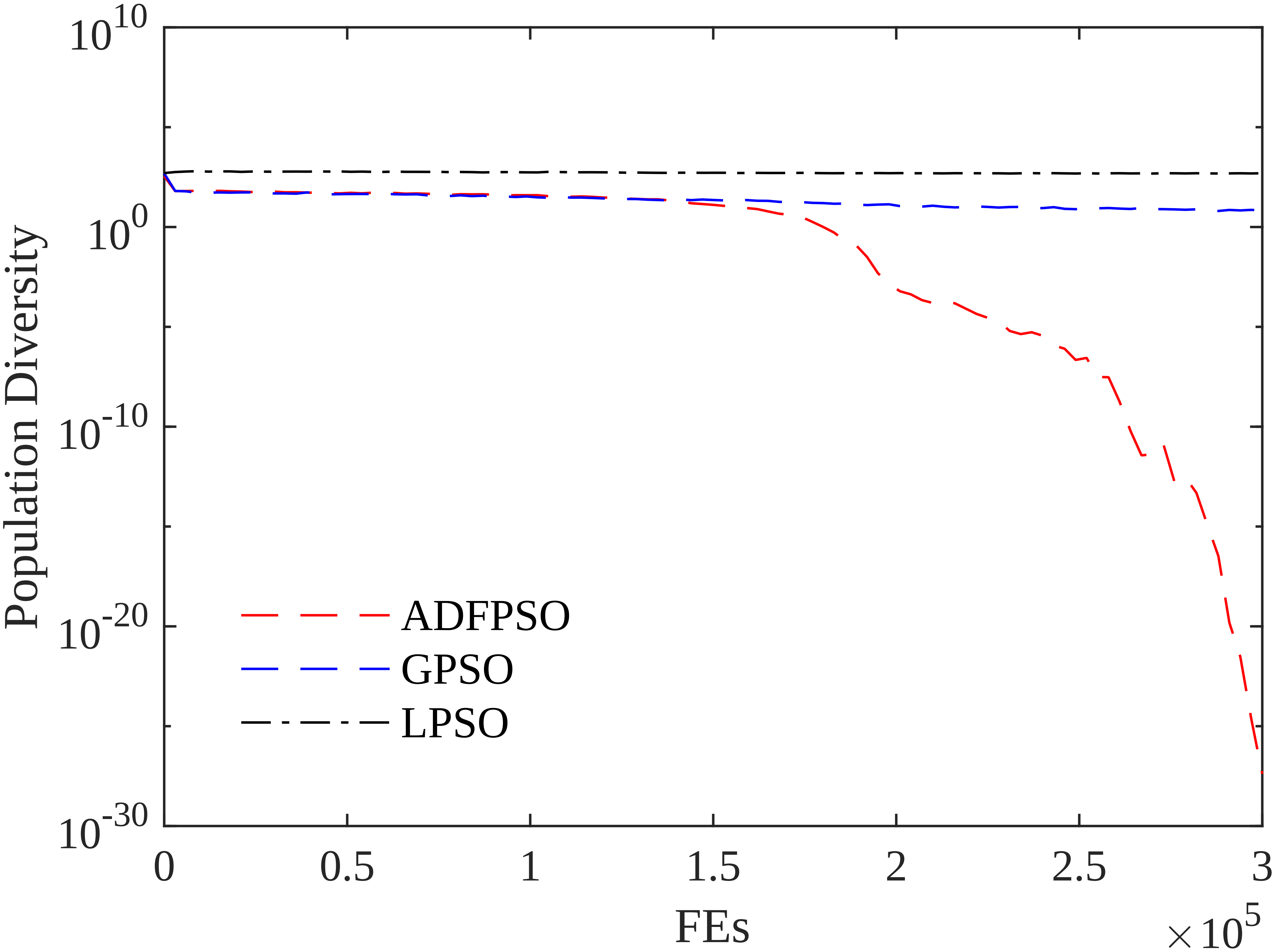
ADFPSO
GPSO
LPSO
Fitness
FEs
$\times 10^5$
10^5
10^0
10^{-5}
10^{-10}
10^{-15}
0
0.5
1
1.5
2
2.5
3



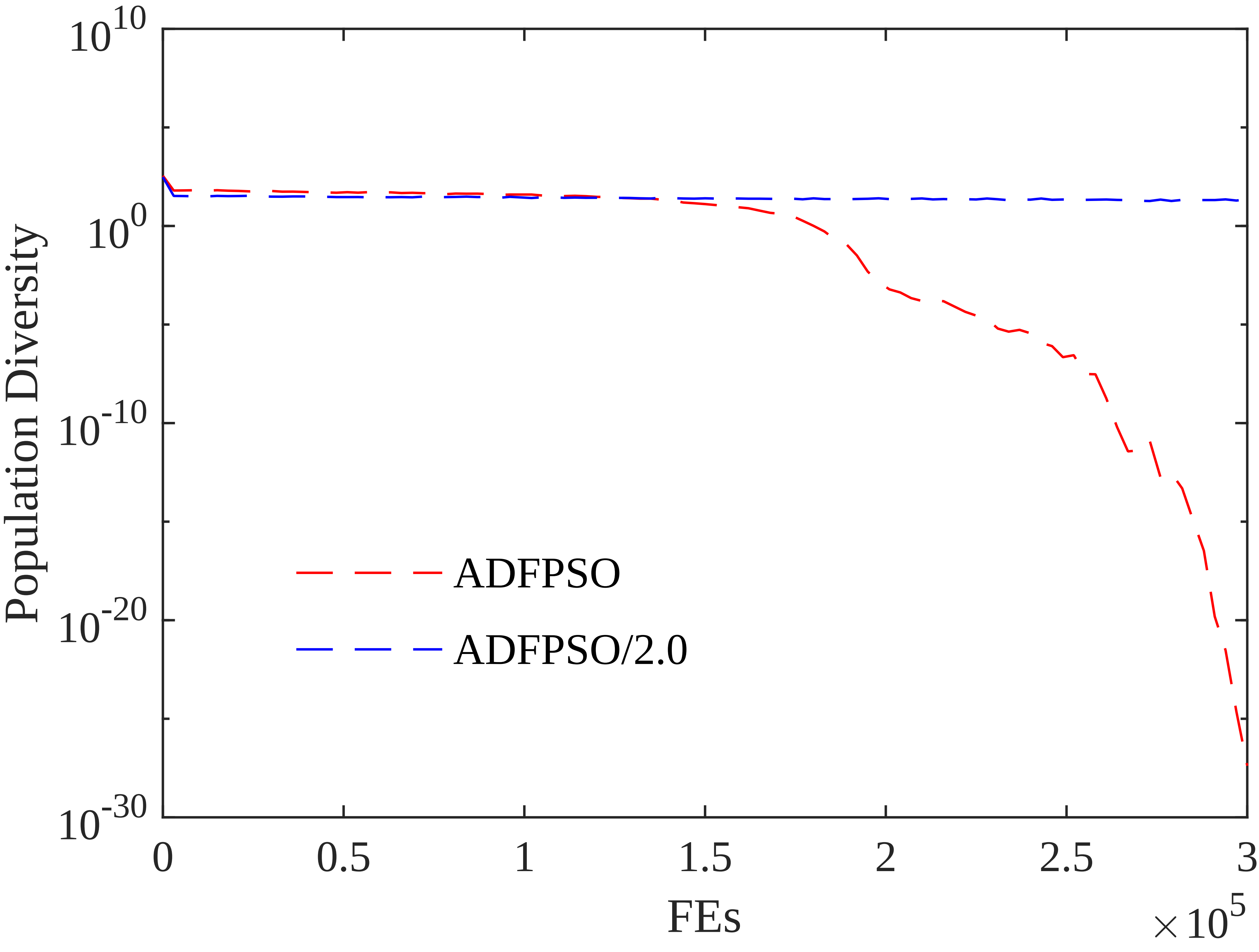
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
ADFPSO/2.0



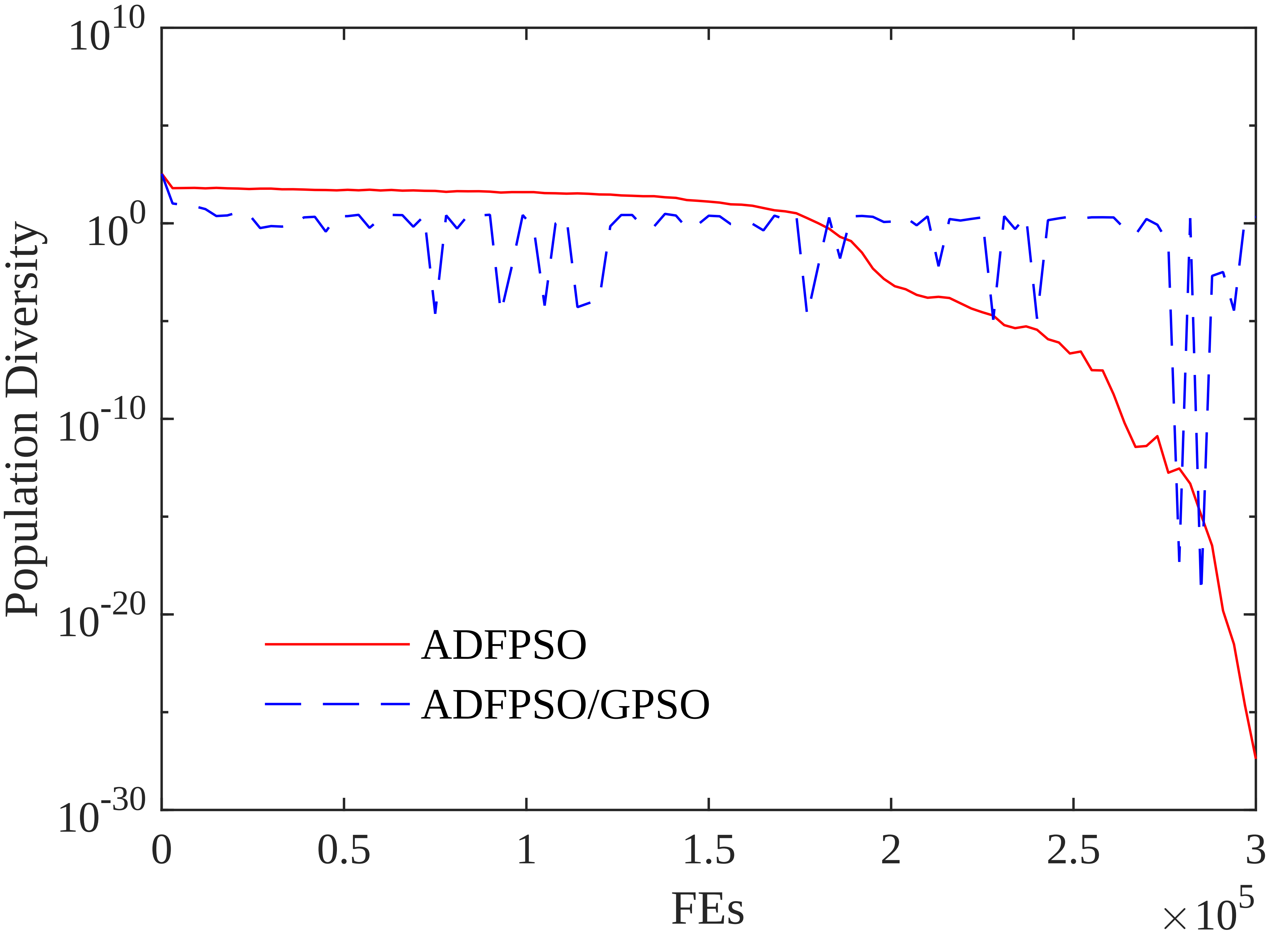
Fitness
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
ADFPSO/GPSO



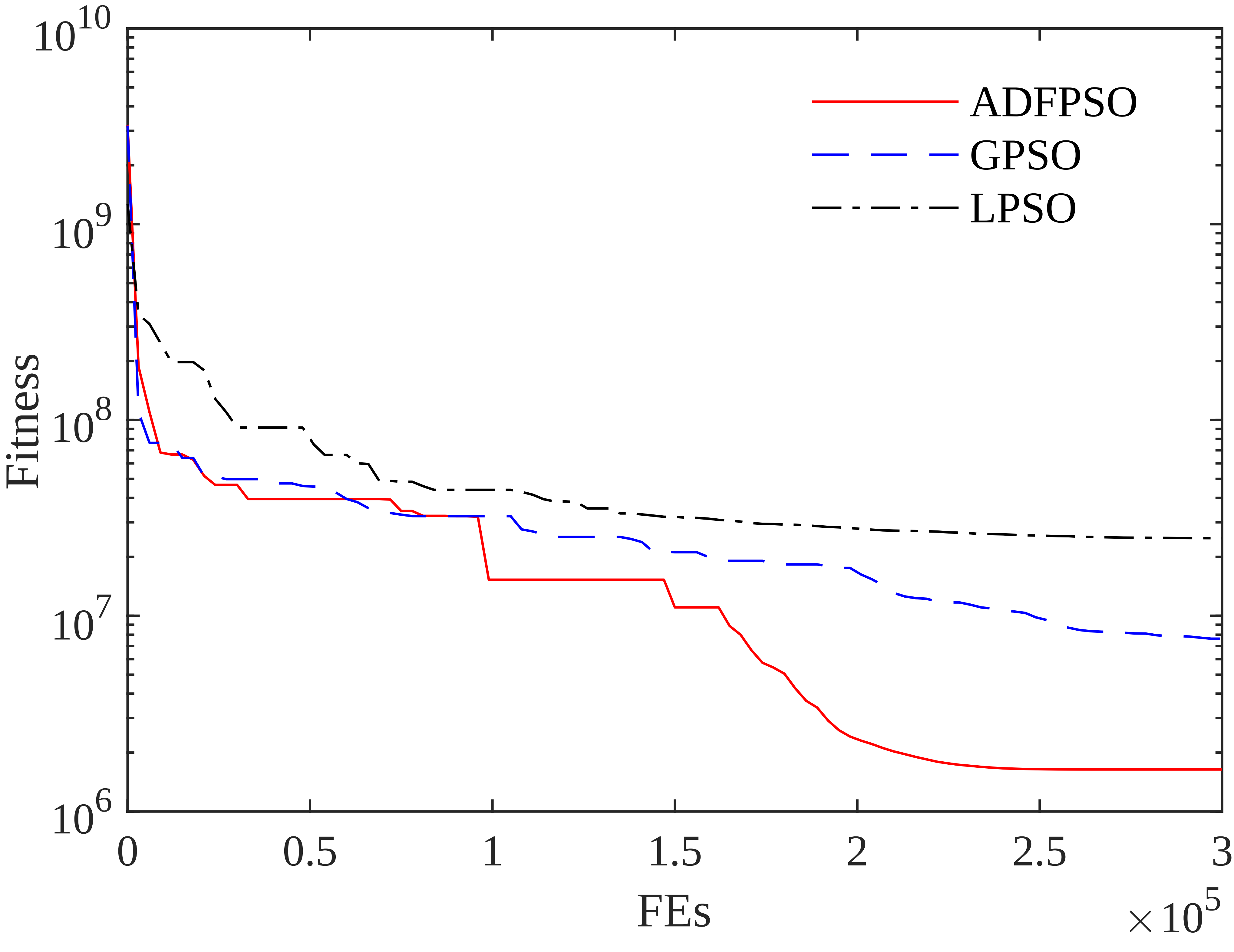
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
Population Diversity
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
GPSO
LPSO

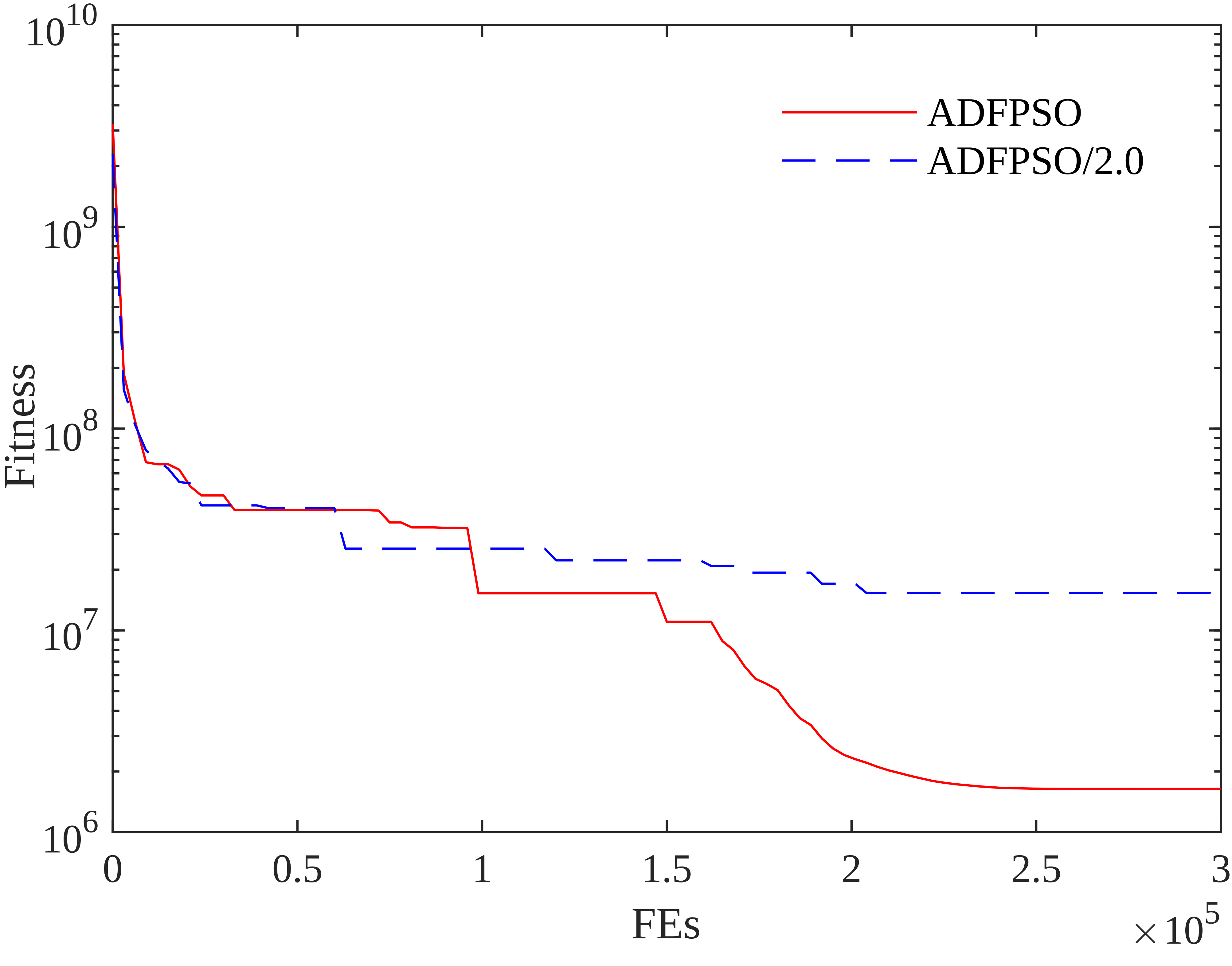


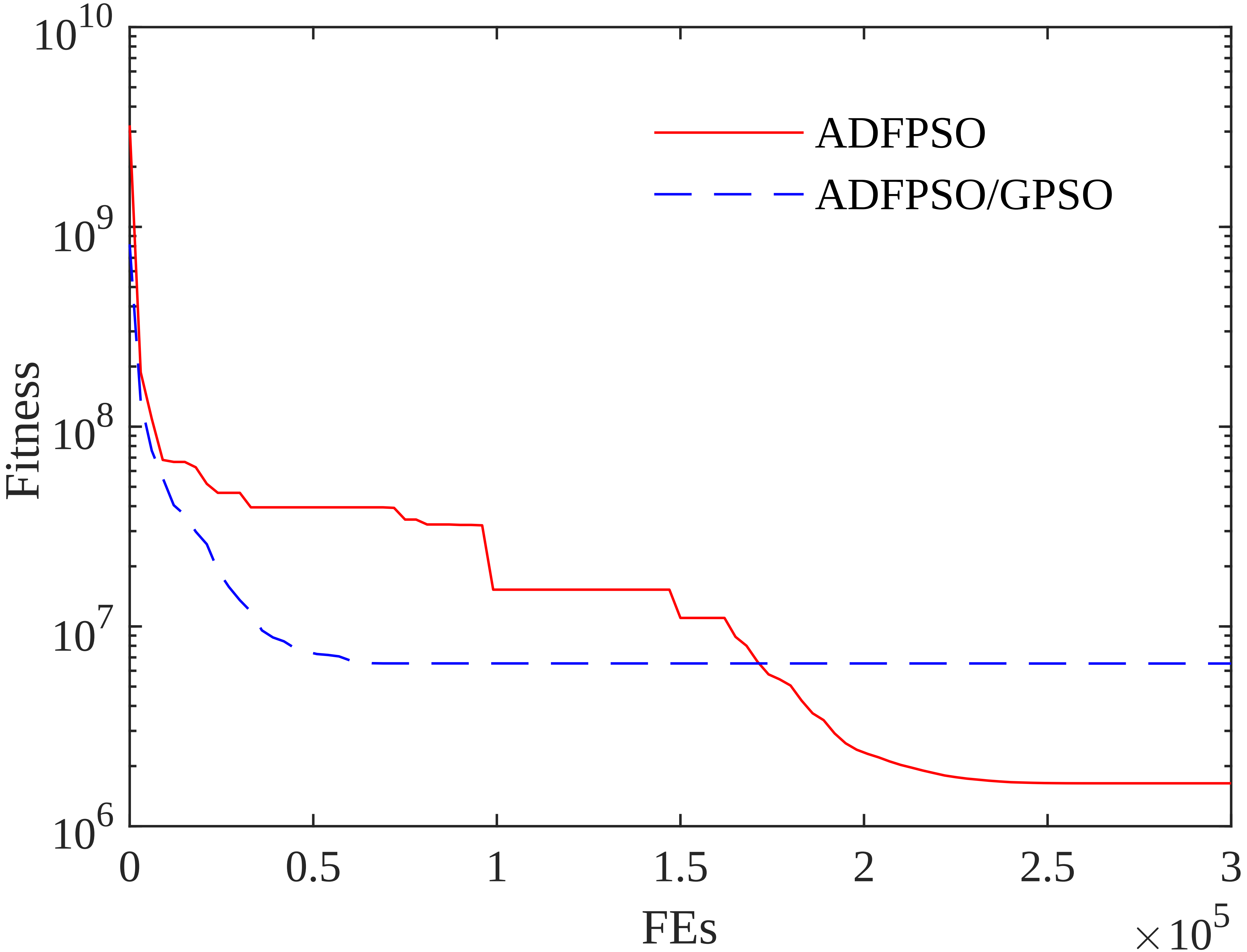
Population Diversity
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
ADFPSO/2.0

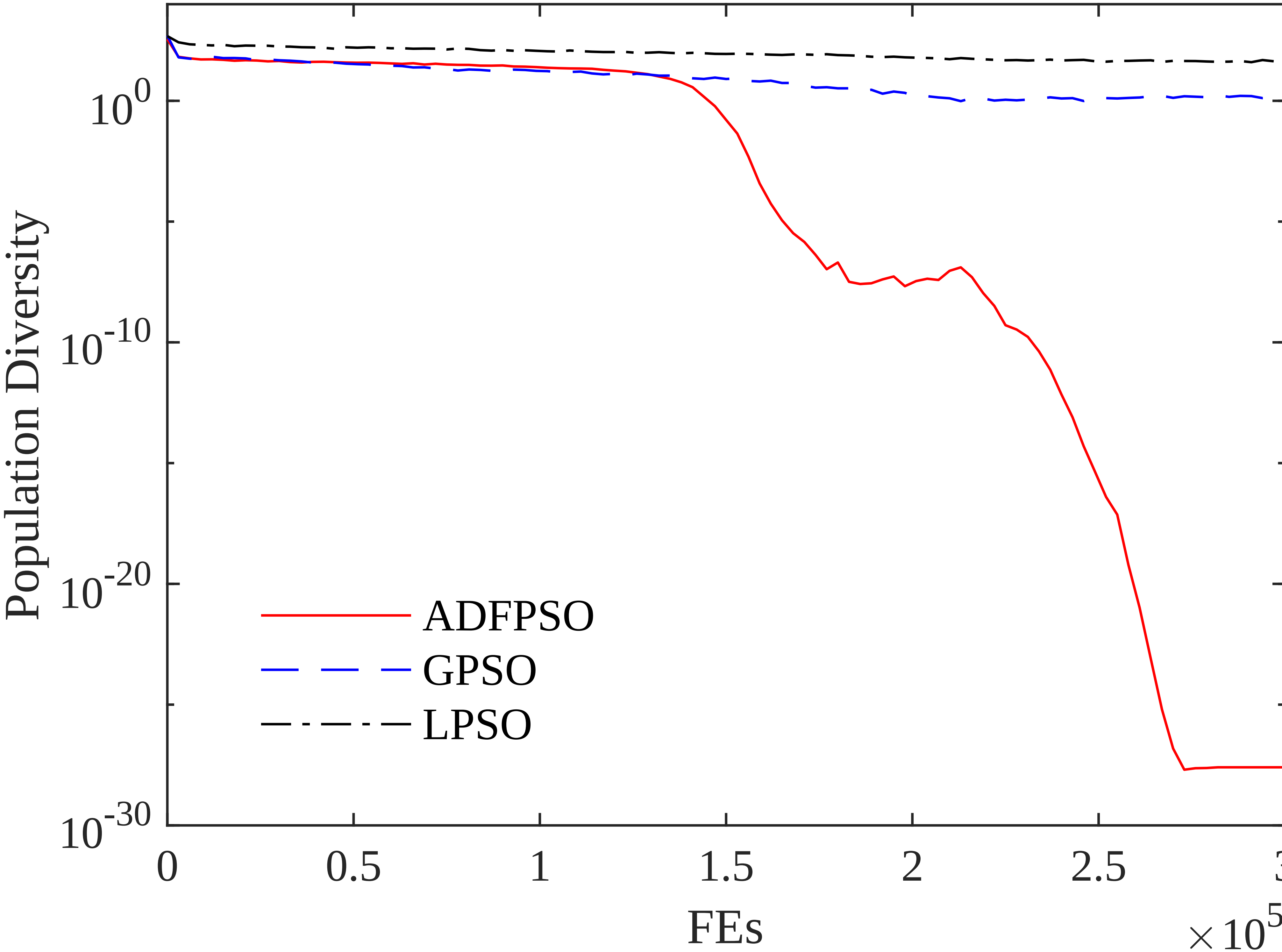


10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
Population Diversity
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
ADFPSO/GPSO

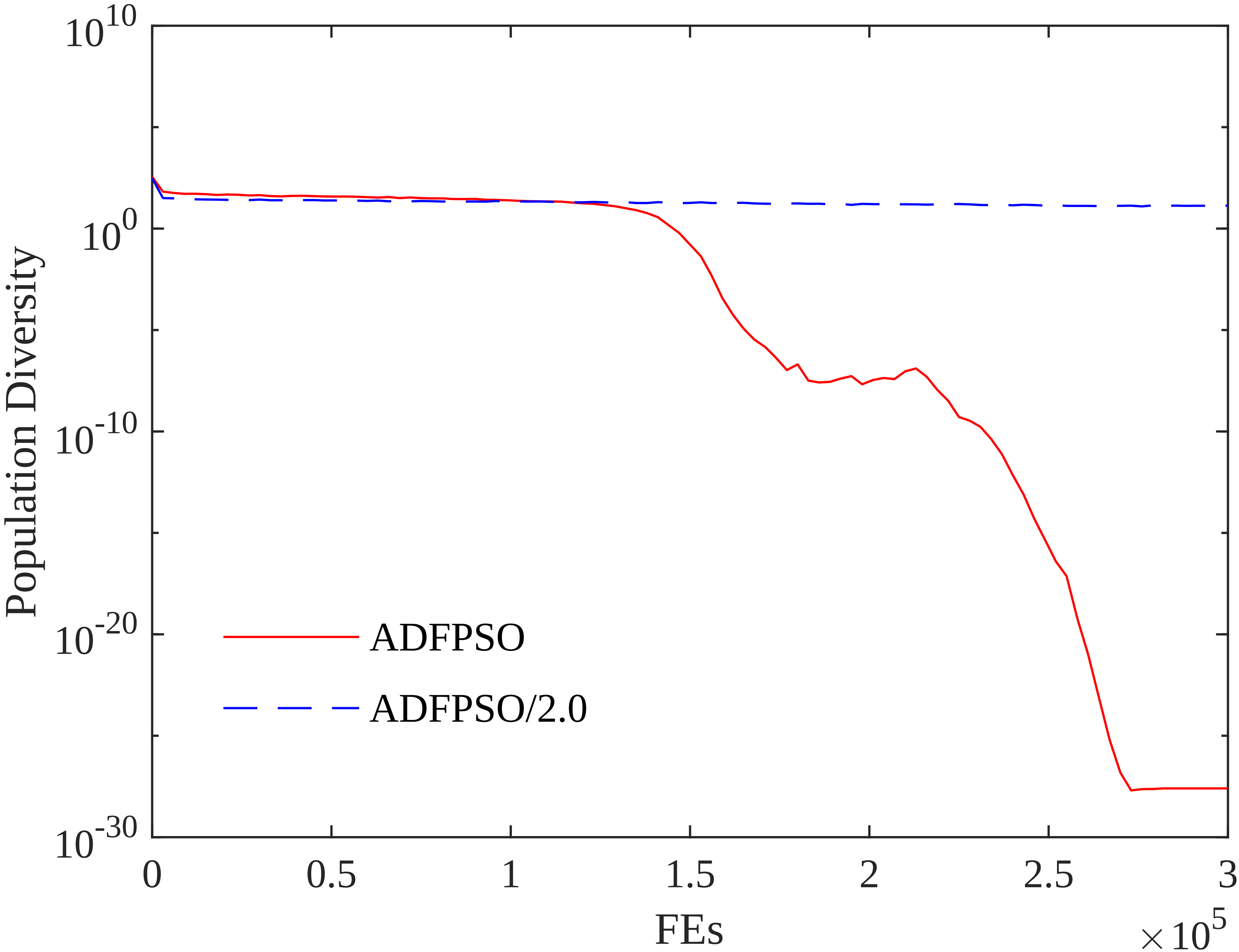

ADFPSO
GPSO
LPSO
Fitness
FEs
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
0
0.5
1
1.5
2
2.5
3
$\times 10^{5}$


ADFPSO
ADFPSO/2.0
Fitness
FEs
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
0
0.5
1
1.5
2
2.5
3
$\times 10^{5}$

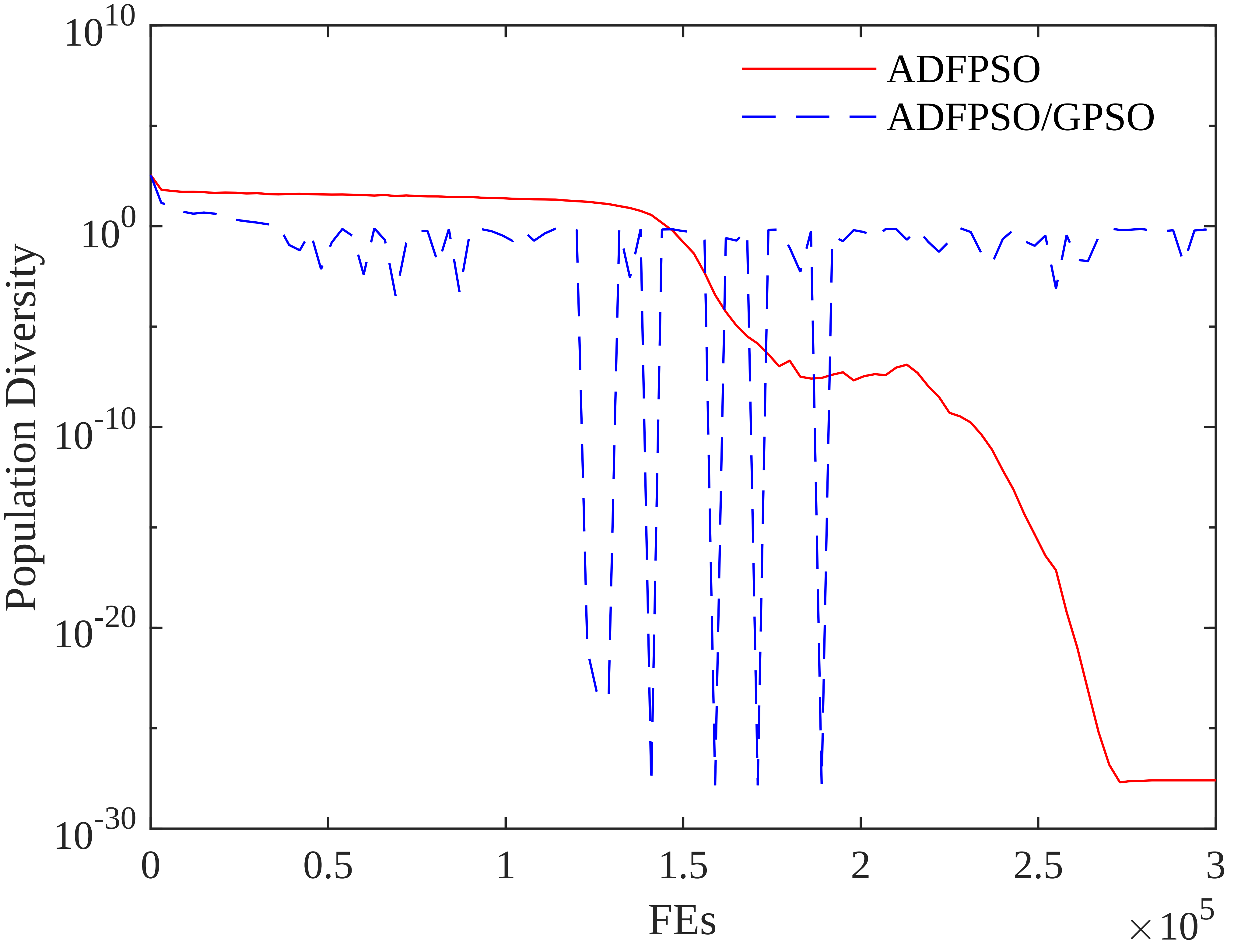

ADFPSO
ADFPSO/GPSO
Fitness
FEs
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
0
0.5
1
1.5
2
2.5
3
$\times 10^5$



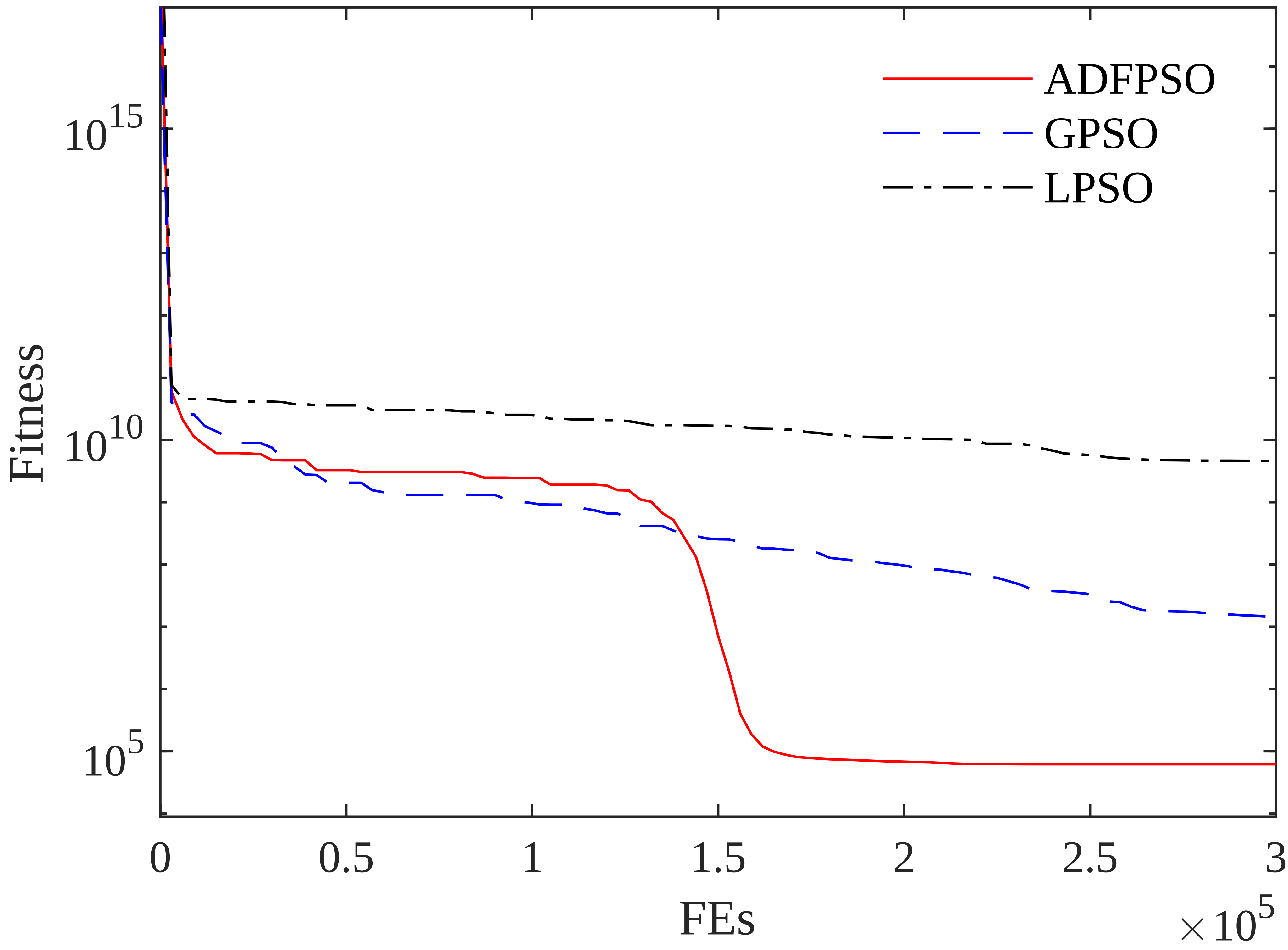
Population Diversity
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
GPSO
LPSO



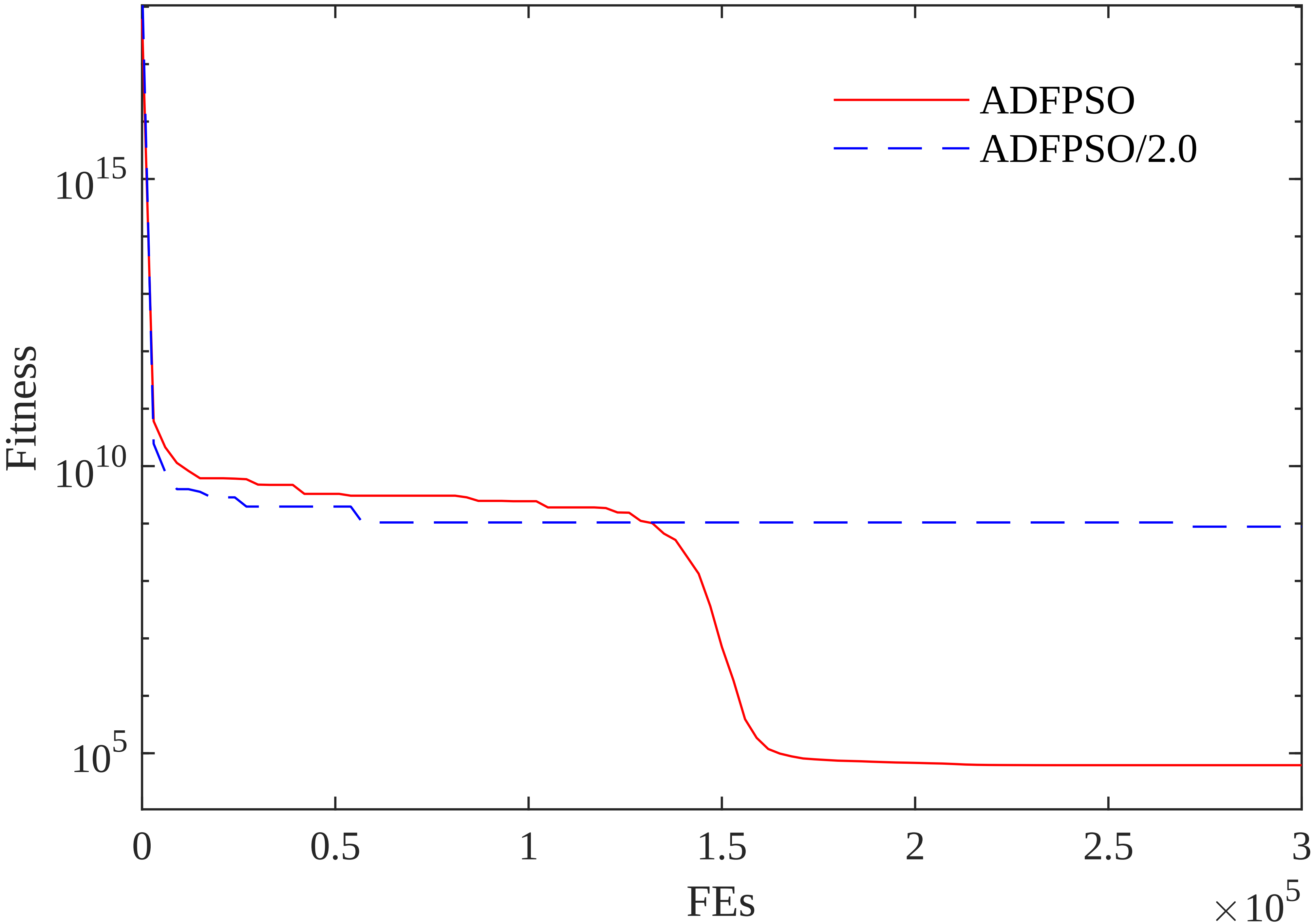
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
Population Diversity
ADFPSO
ADFPSO/2.0
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$

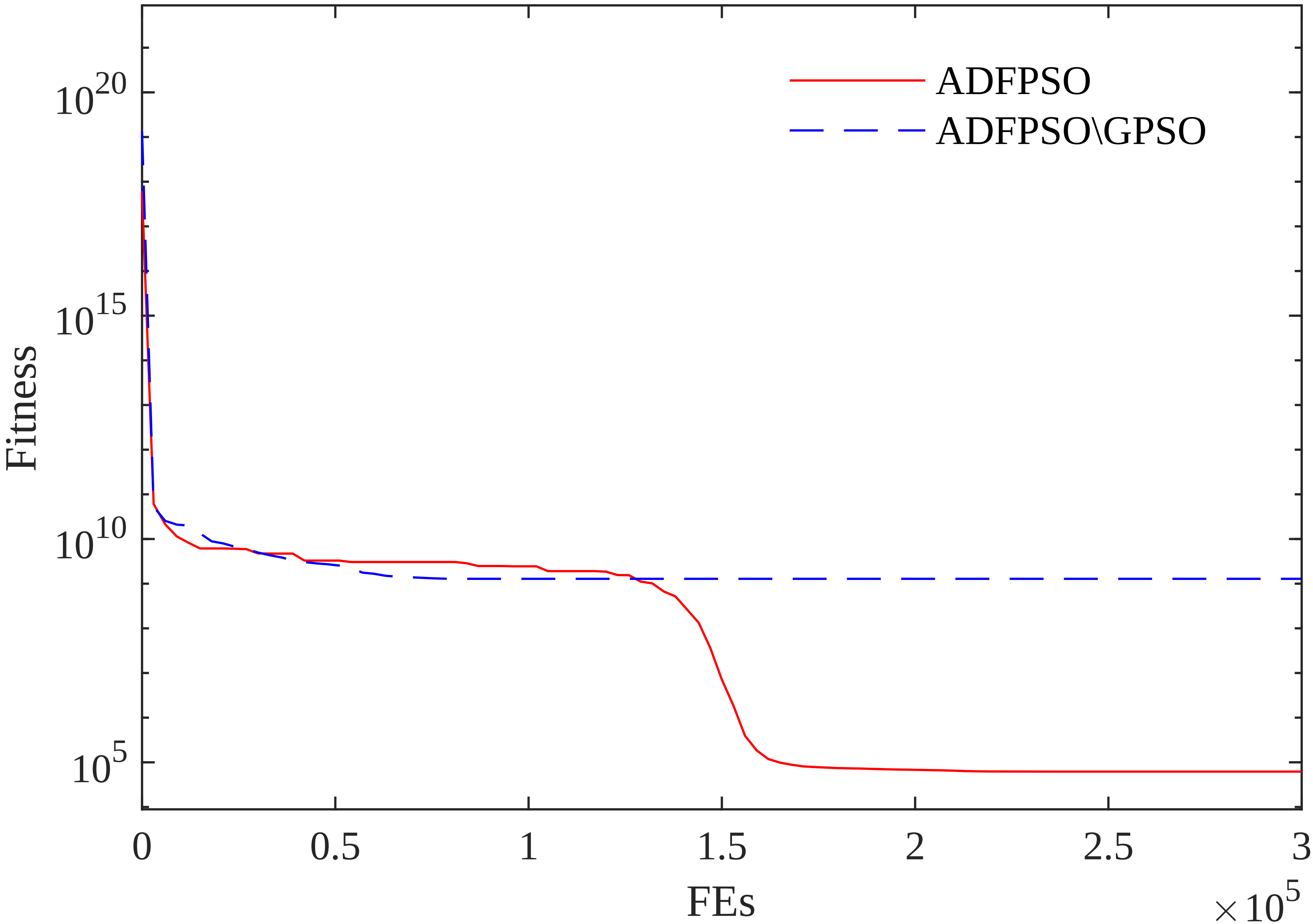


ADFPSO
ADFPSO/GPSO
Population Diversity
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$

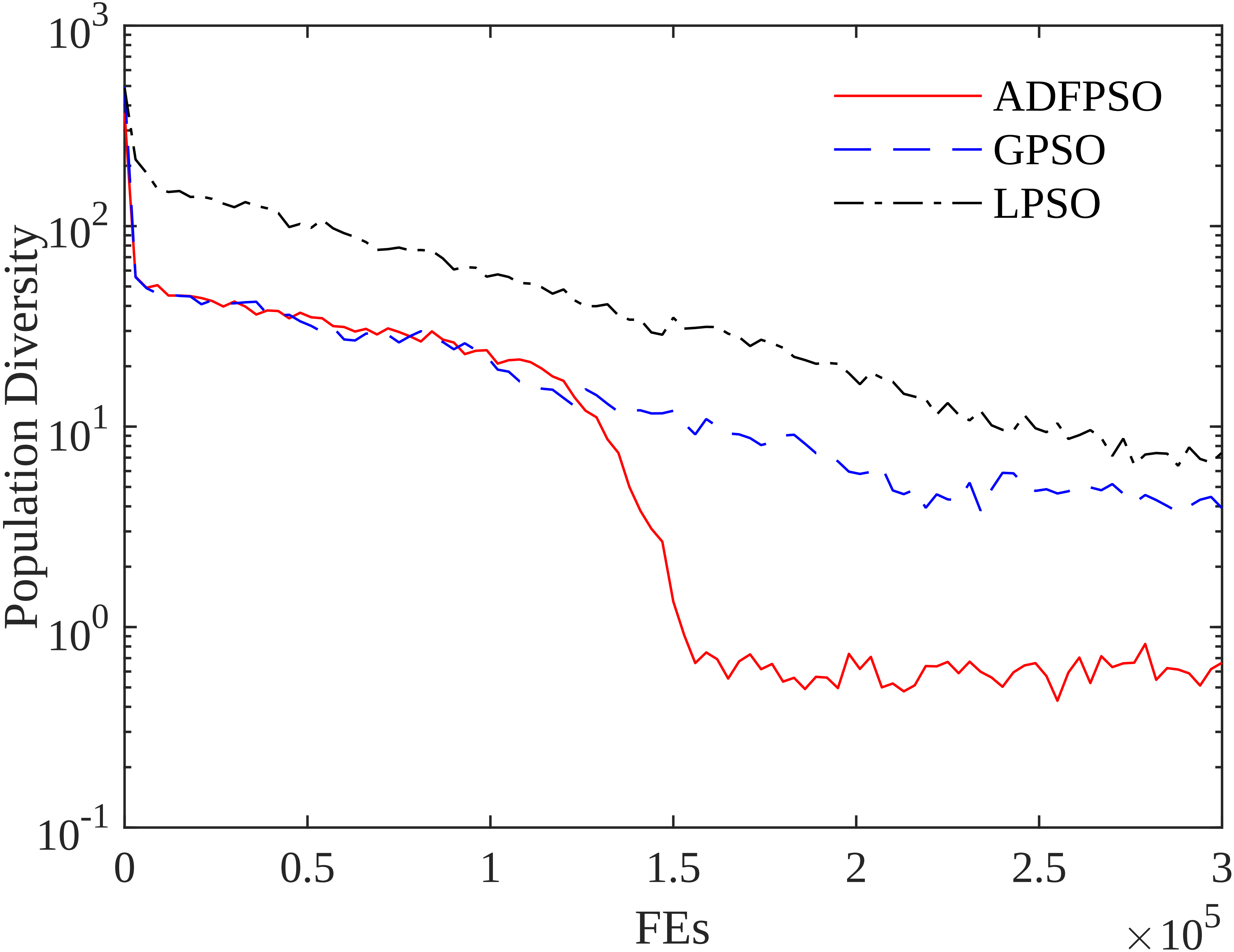

ADFPSO
GPSO
LPSO
Fitness
FEs
10^{15}
10^{10}
10^{5}
0
0.5
1
1.5
2
2.5
3
$\times 10^{5}$

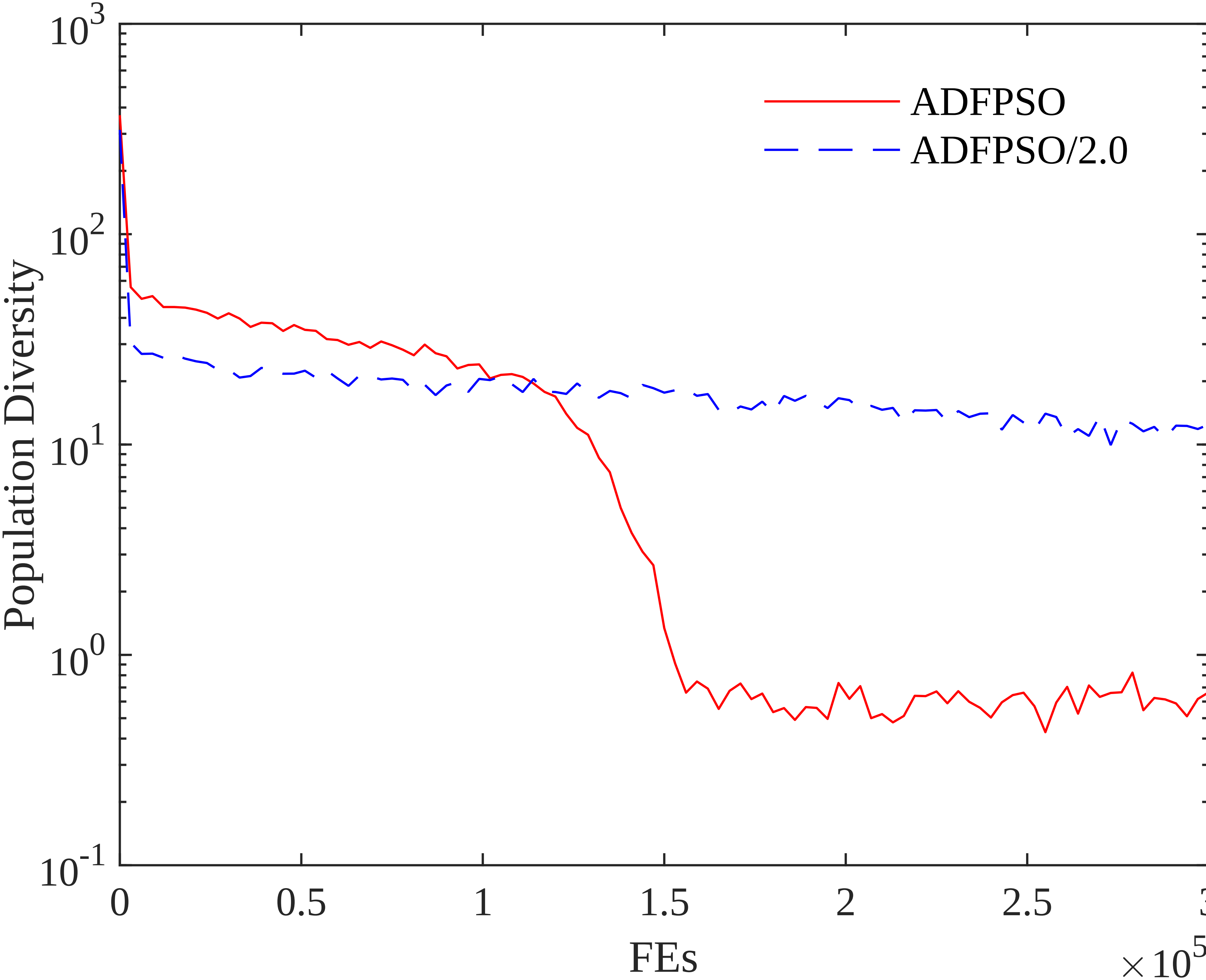
Click here to access/download;Figure;F03-fit-c.eps

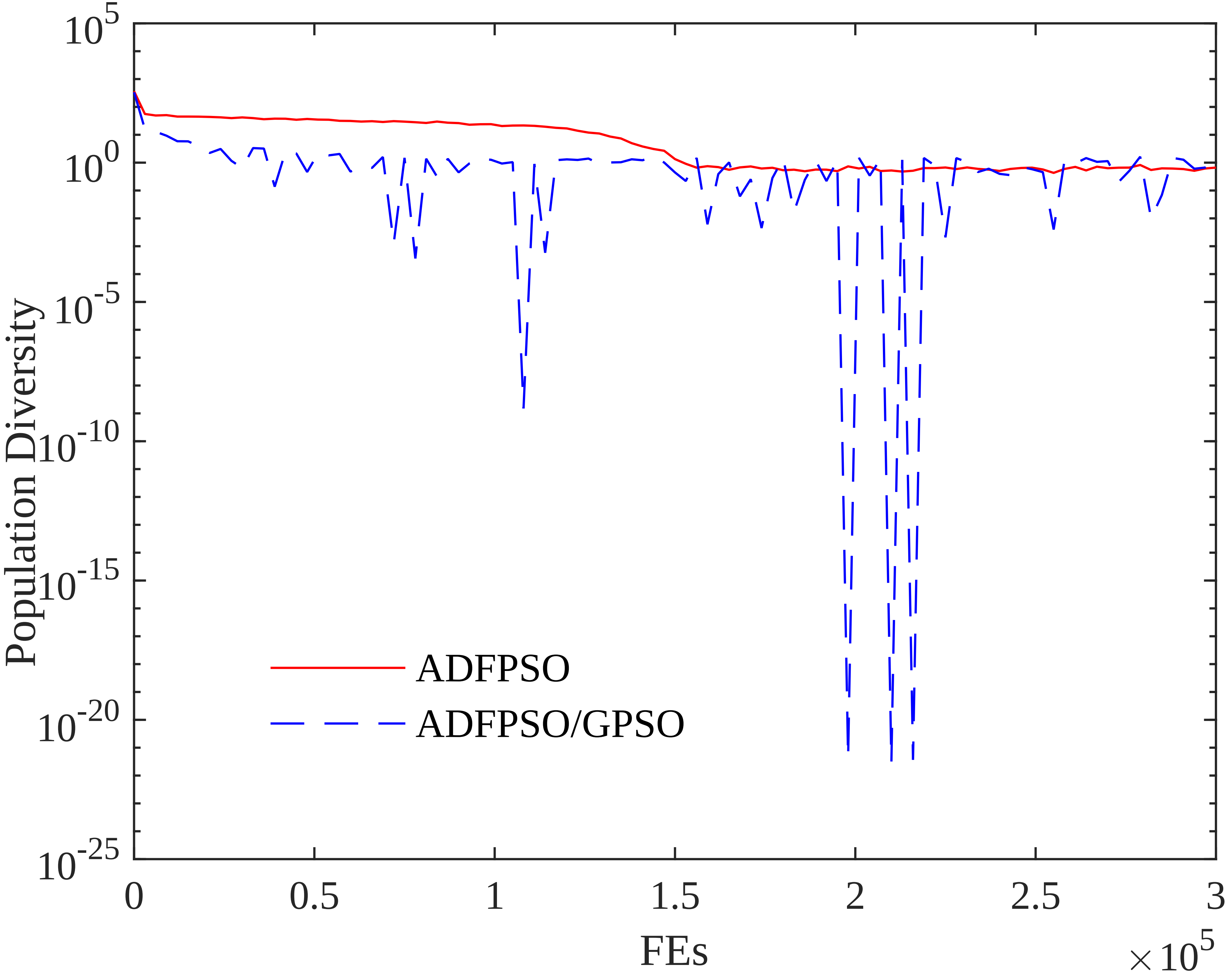




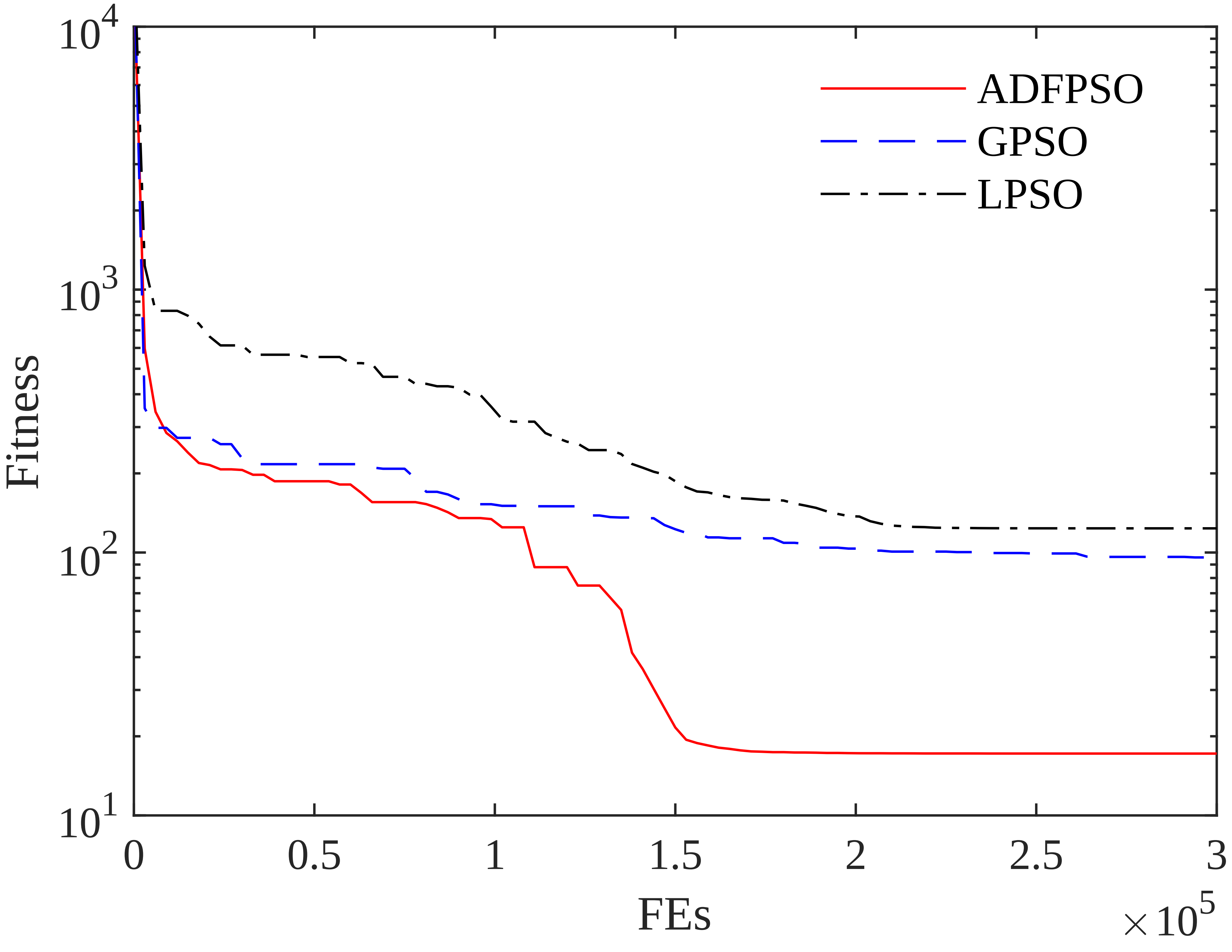
ADFPSO
ADFPSO\GPSO
Fitness
FEs
10^{20}
10^{15}
10^{10}
10^{5}
0
0.5
1
1.5
2
2.5
3
$\times 10^{5}$

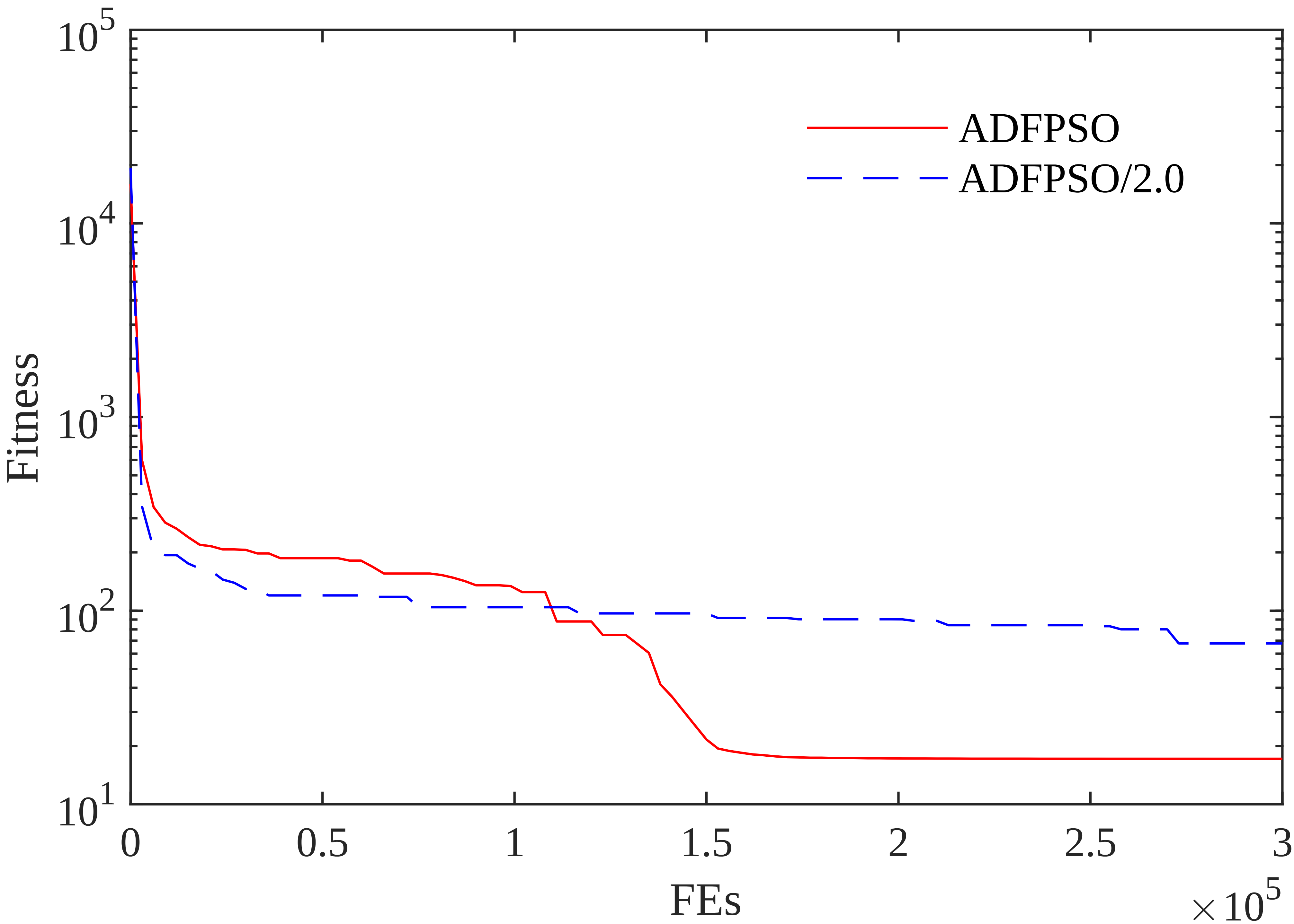

ADFPSO
GPSO
LPSO
Population Diversity
FEs
10^3
10^2
10^1
10^0
10^{-1}
0
0.5
1
1.5
2
2.5
3
$\times 10^5$


ADFPSO
ADFPSO/2.0
Population Diversity
FEs
$\times 10^5$
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
0
0.5
1
1.5
2
2.5
3

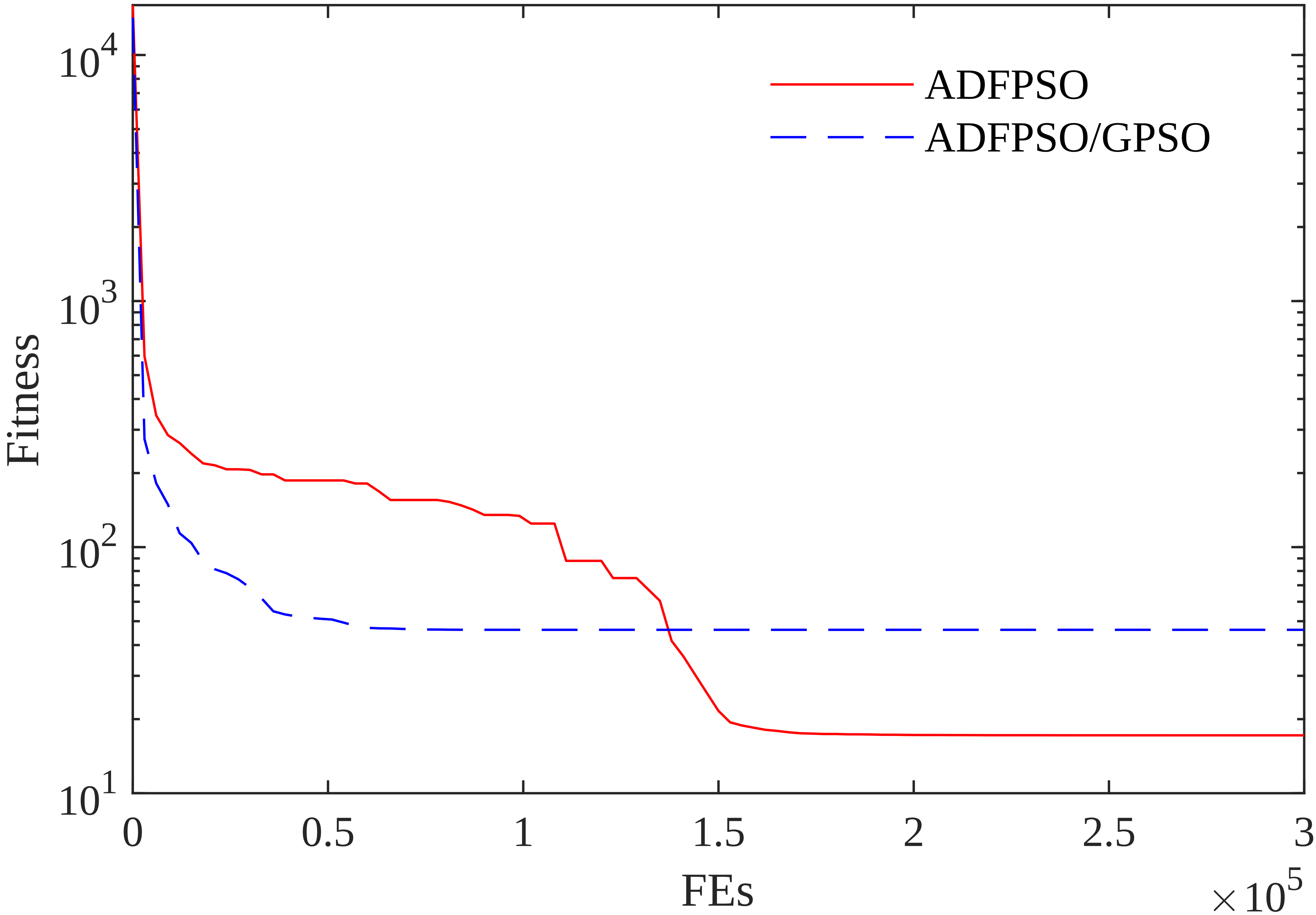


10^5
10^0
10^-5
10^-10
10^-15
10^-20
10^-25
Population Diversity
ADFPSO
ADFPSO/GPSO
0
0.5
1
1.5
2
2.5
3
FEs
×10^5

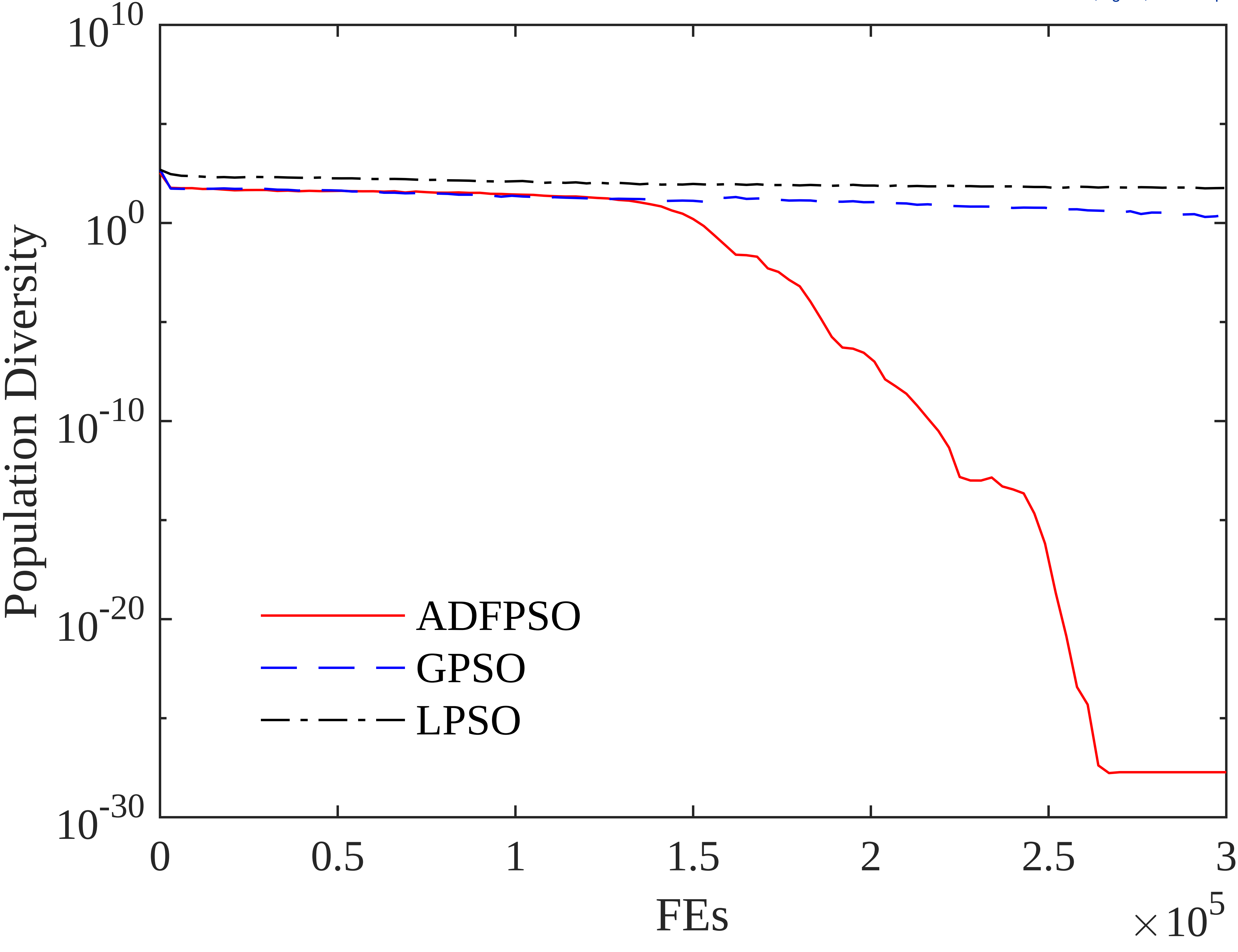

ADFPSO
GPSO
LPSO
Fitness
FEs
$\times 10^5$
10^4
10^3
10^2
10^1
0
0.5
1
1.5
2
2.5
3



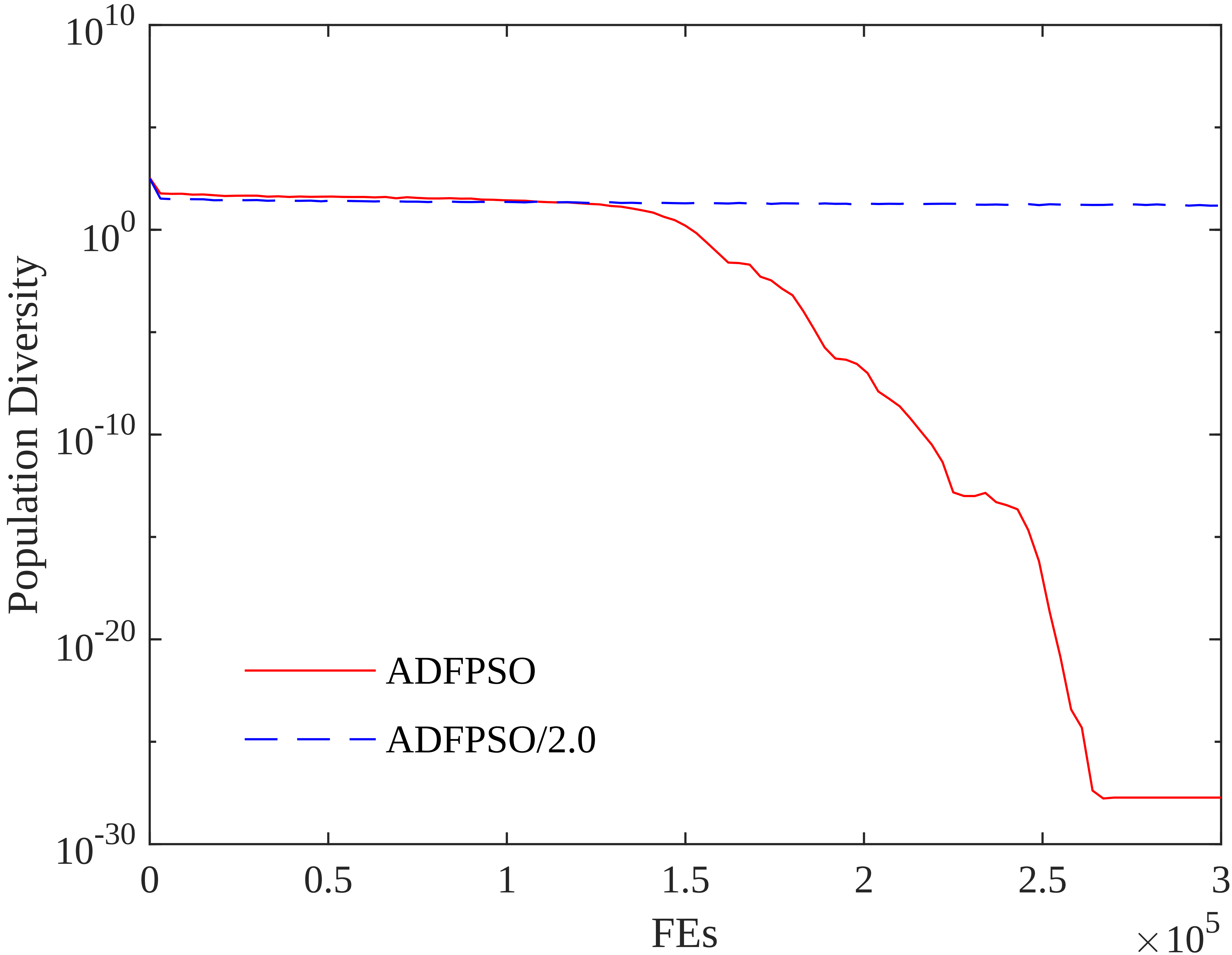
ADFPSO
ADFPSO/2.0
Fitness
FEs
10^5
10^4
10^3
10^2
10^1
0
0.5
1
1.5
2
2.5
3
$\times 10^5$



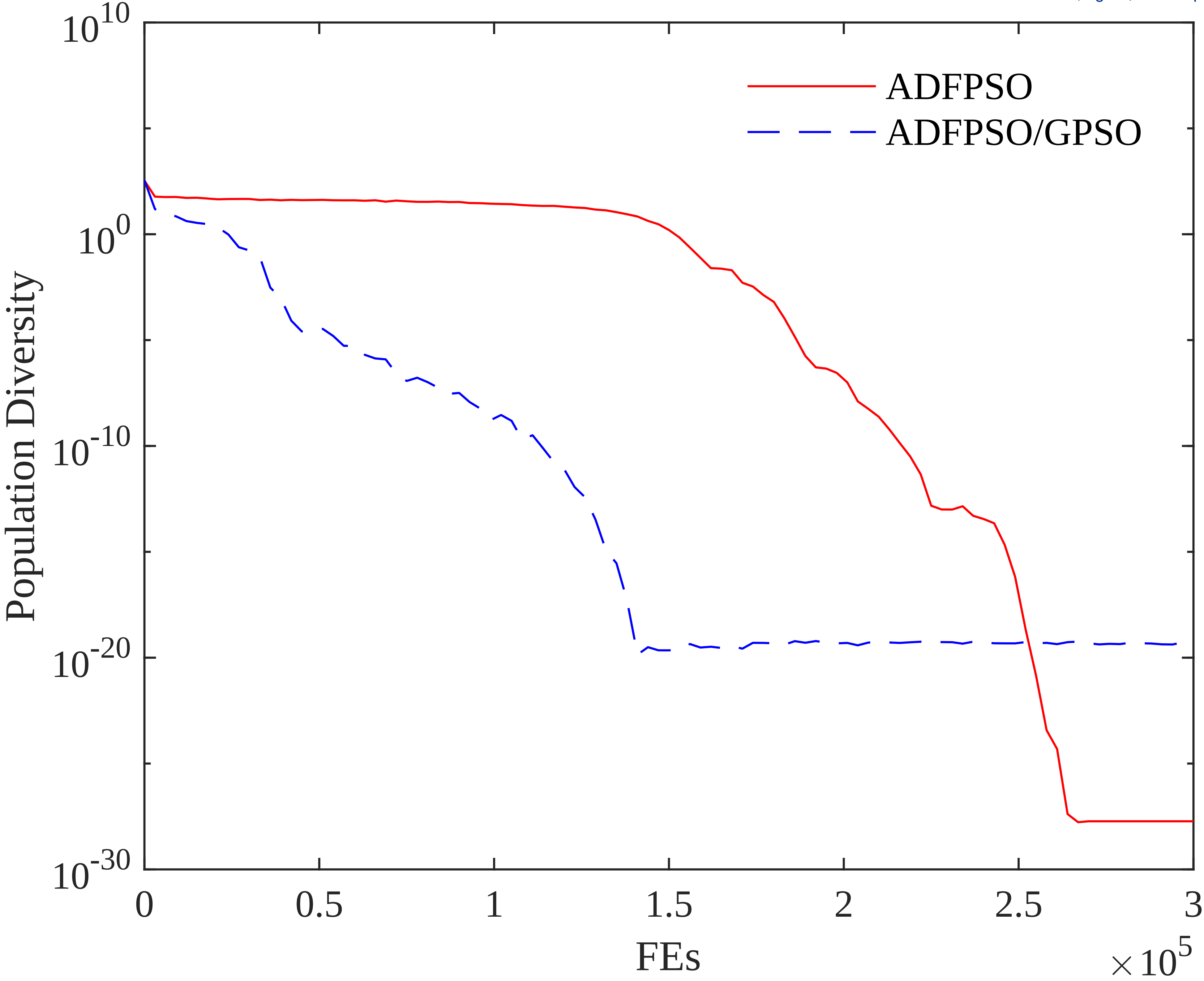
ADFPSO
ADFPSO/GPSO
Fitness
FEs
10^4
10^3
10^2
10^1
0
0.5
1
1.5
2
2.5
3
$\times 10^5$



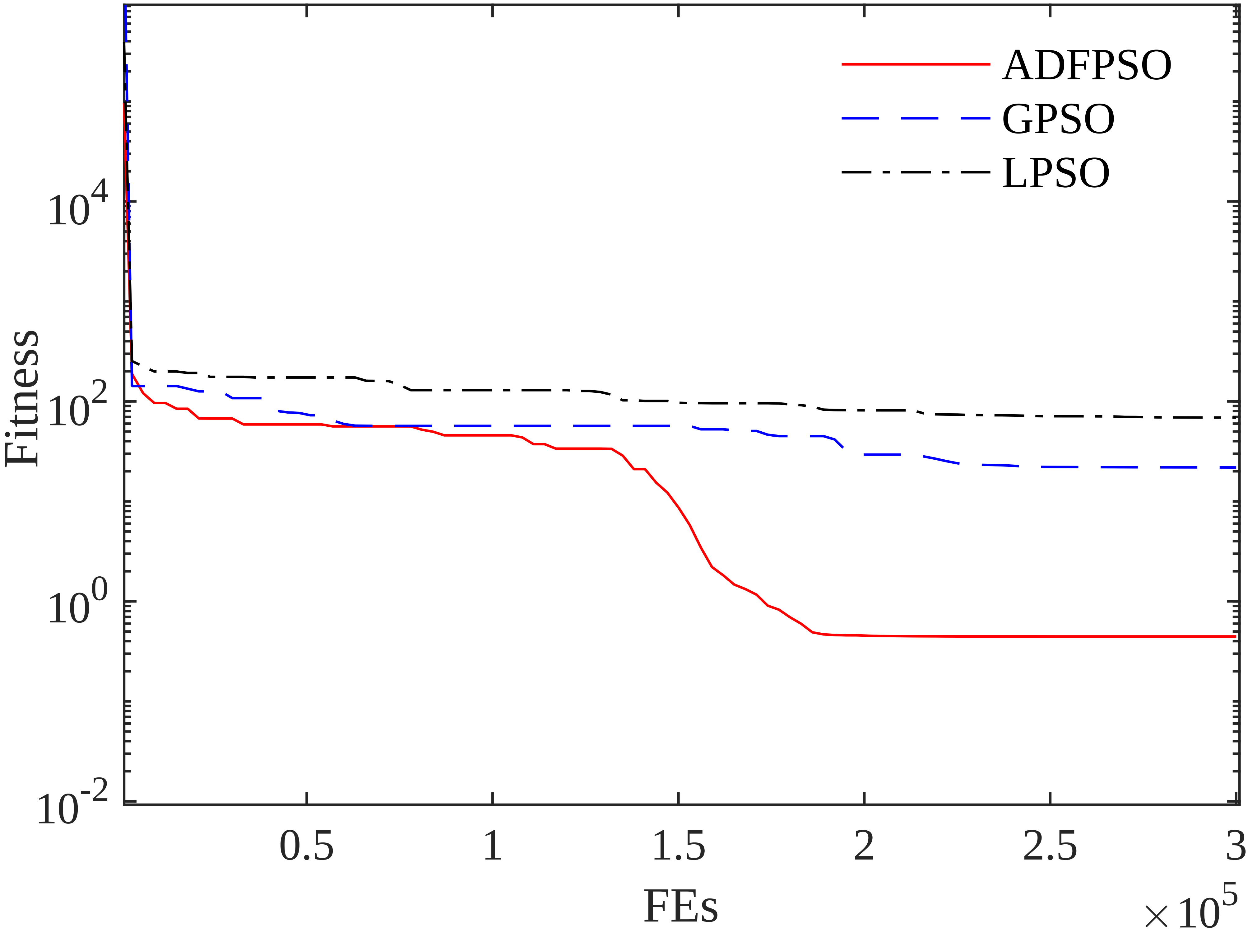
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
Population Diversity
ADFPSO
GPSO
LPSO
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$

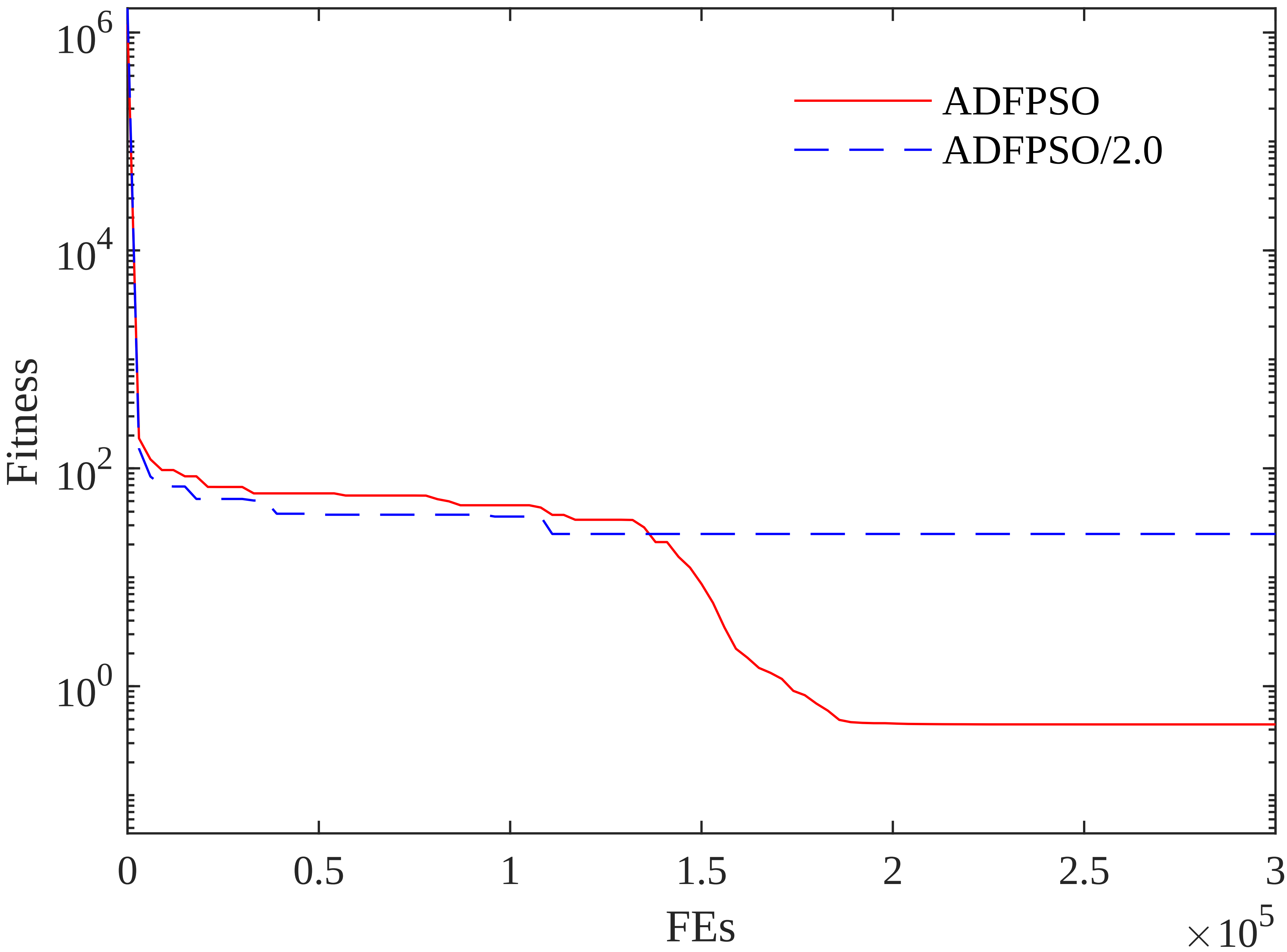


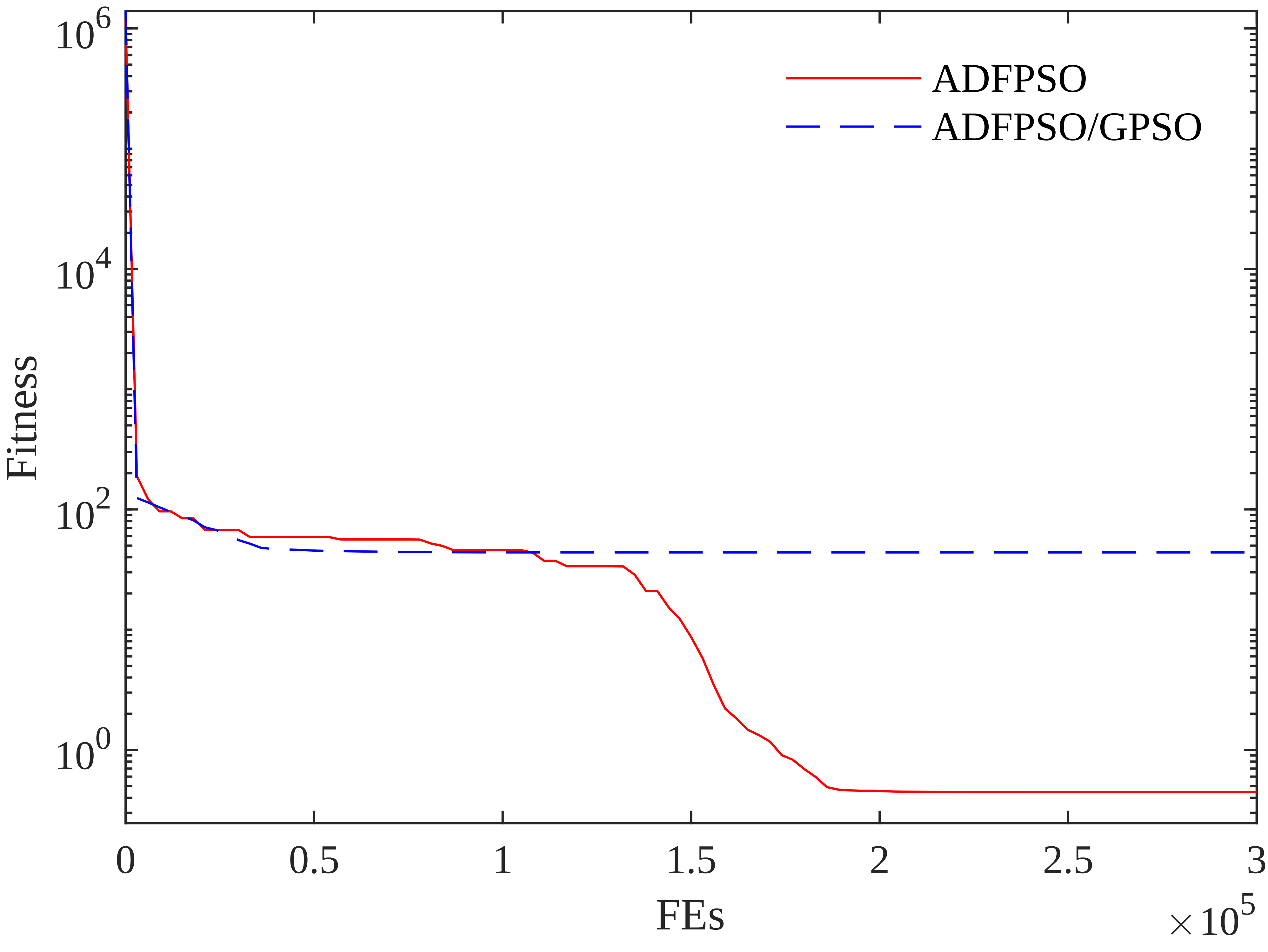
10^10
10^0
10^-10
10^-20
10^-30
Population Diversity
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
ADFPSO/2.0



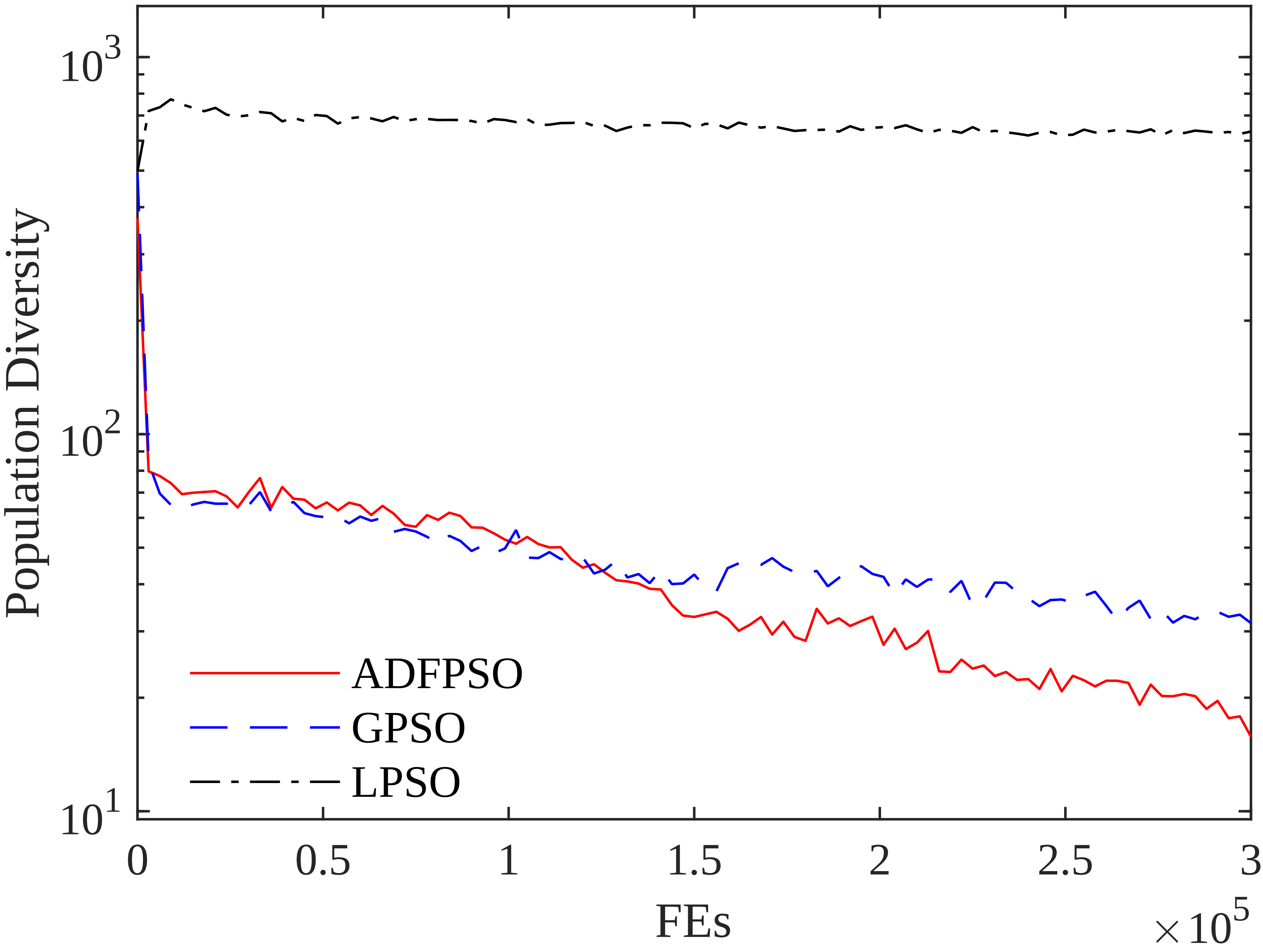
ADFPSO
ADFPSO/GPSO
Population Diversity
FEs
$\times 10^5$
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5
3


ADFPSO
GPSO
LPSO
Fitness
10^{4}
10^{2}
10^{0}
10^{-2}
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$


ADFPSO
ADFPSO/2.0
Fitness
10^6
10^4
10^2
10^0
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$

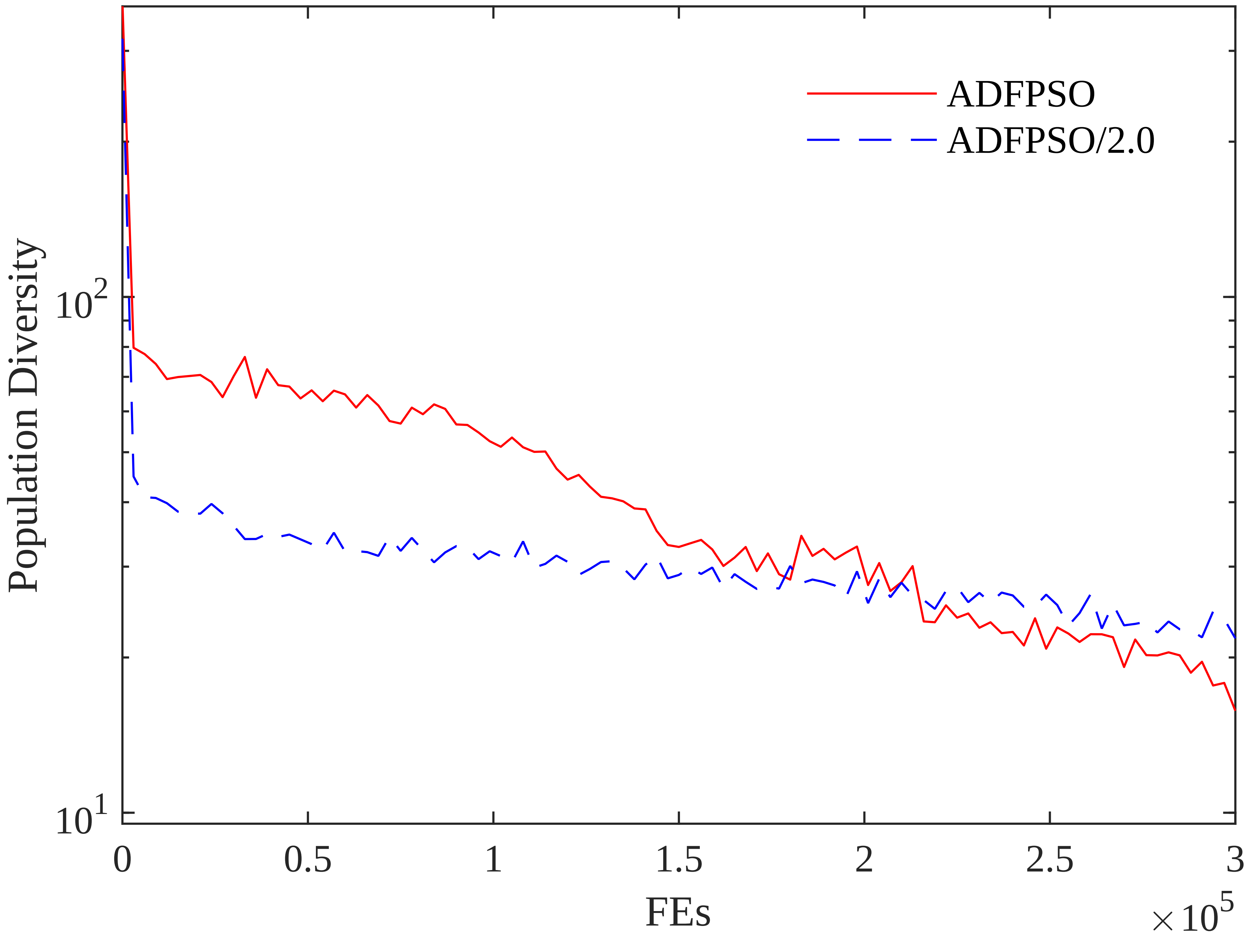


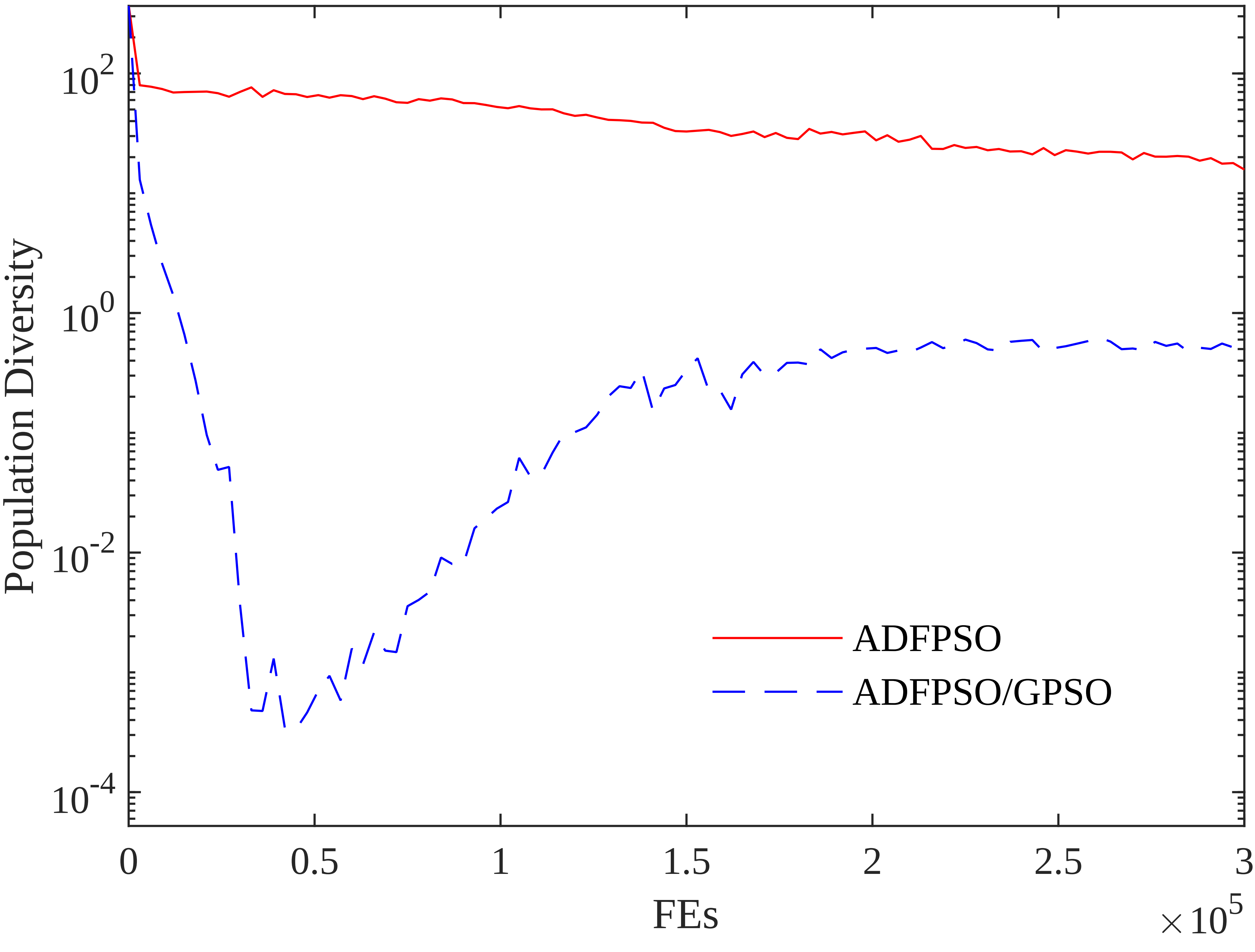
ADFPSO
ADFPSO/GPSO
Fitness
FEs
10^6
10^4
10^2
10^0
0
0.5
1
1.5
2
2.5
3
$\times 10^5$



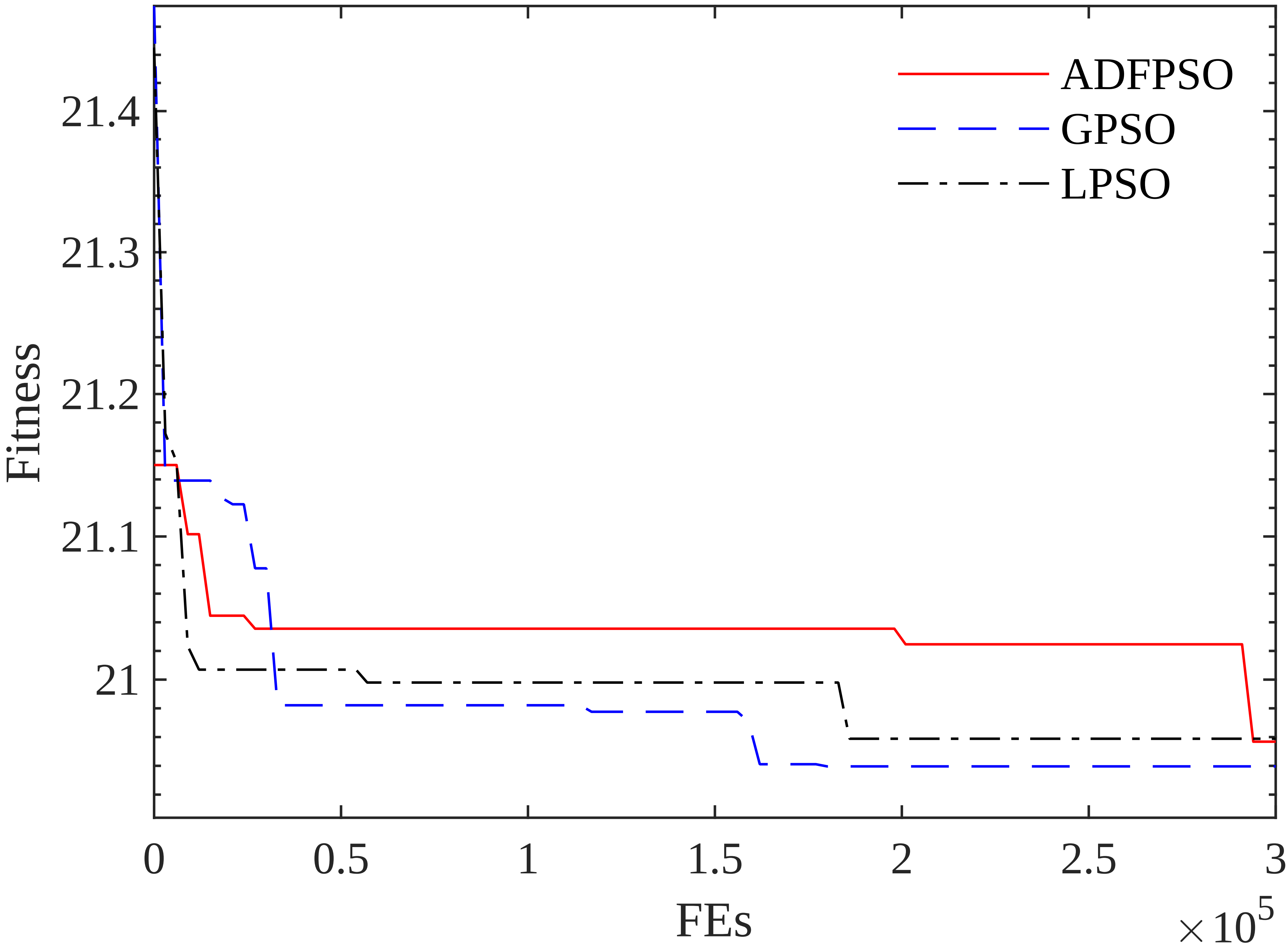
Population Diversity
10^3
10^2
10^1
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
GPSO
LPSO

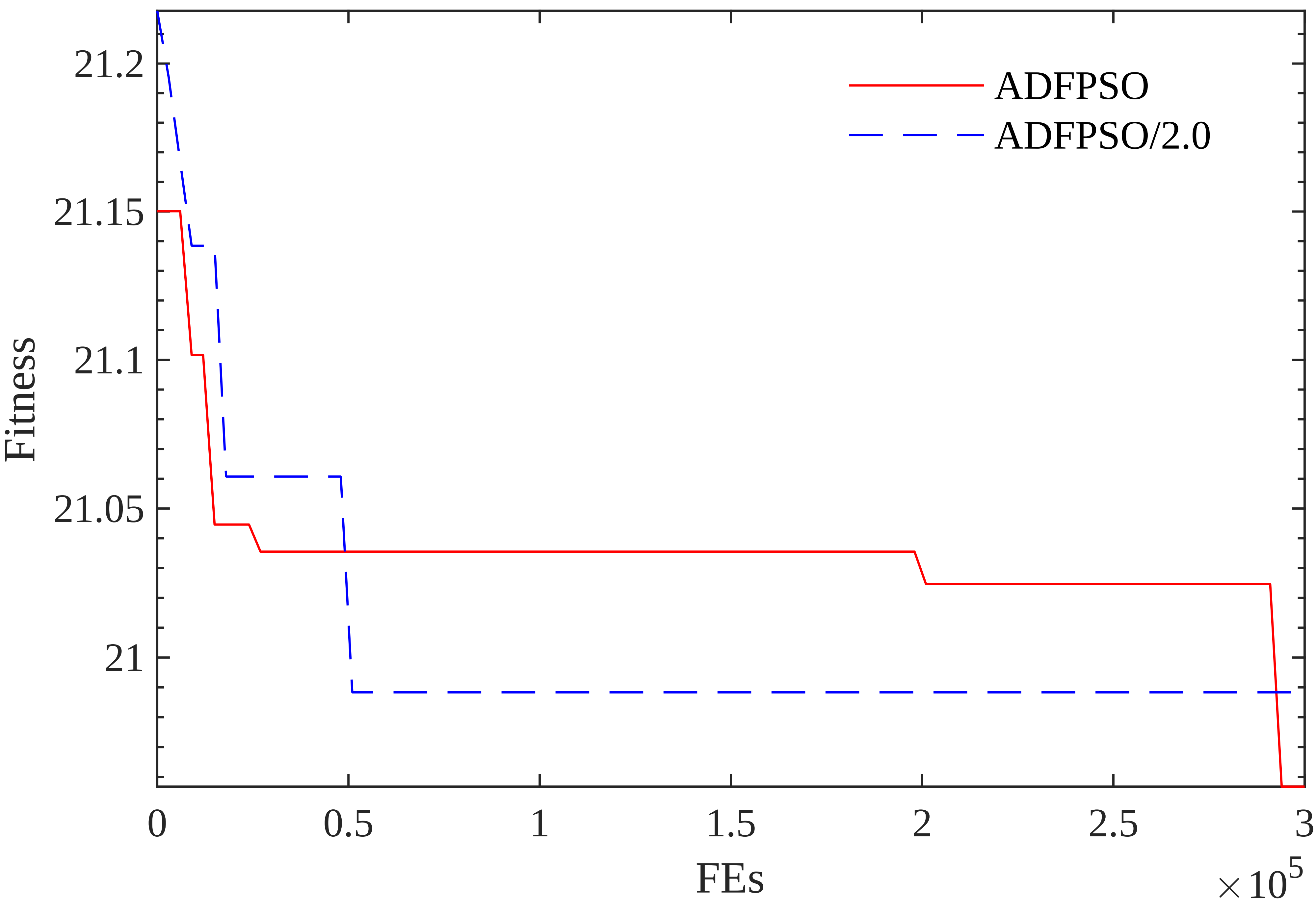
Click here to access/download;Figure;F08-div-c.eps



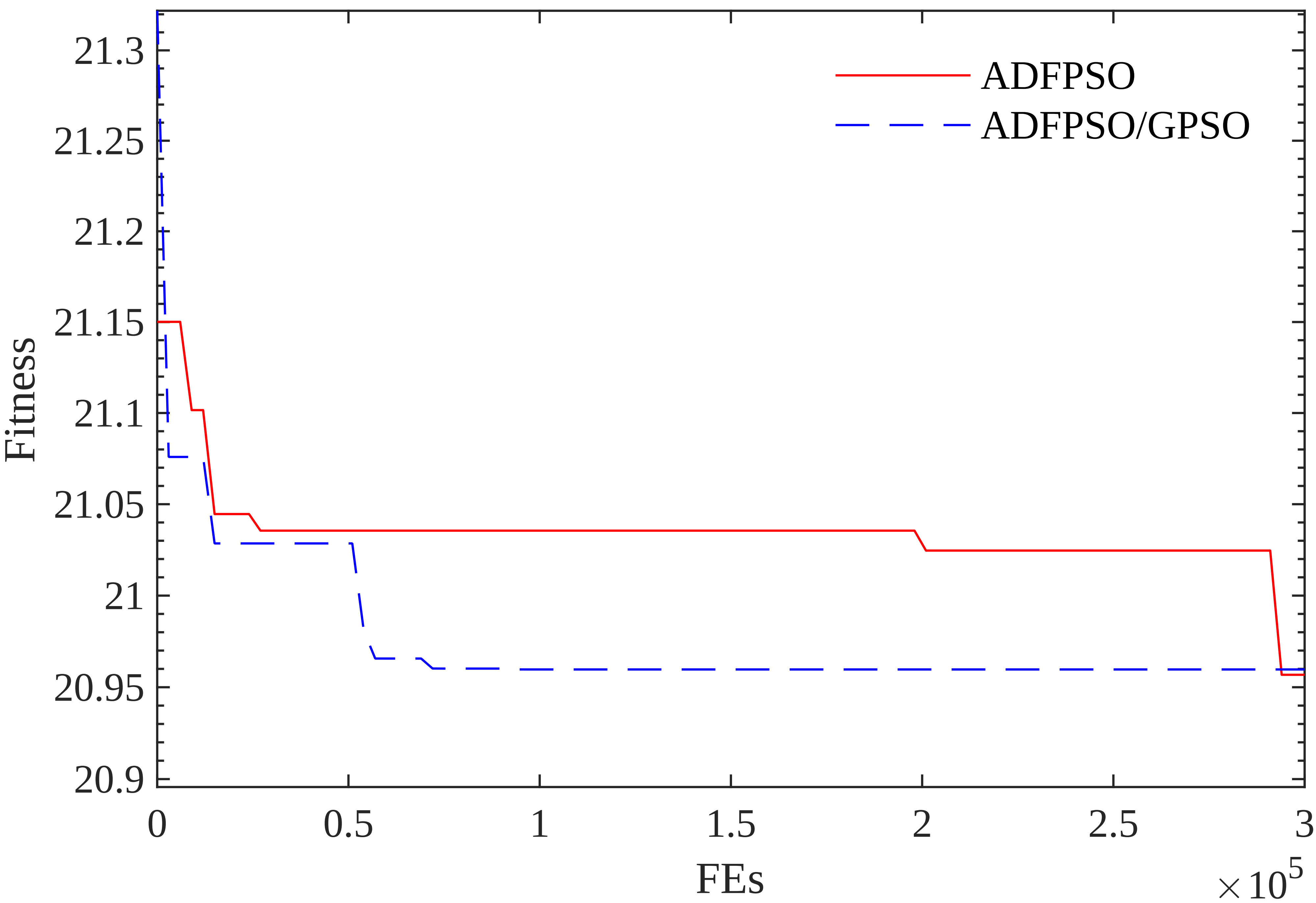


Population Diversity
10^{2}
10^{0}
10^{-2}
10^{-4}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADFPSO
ADFPSO/GPSO

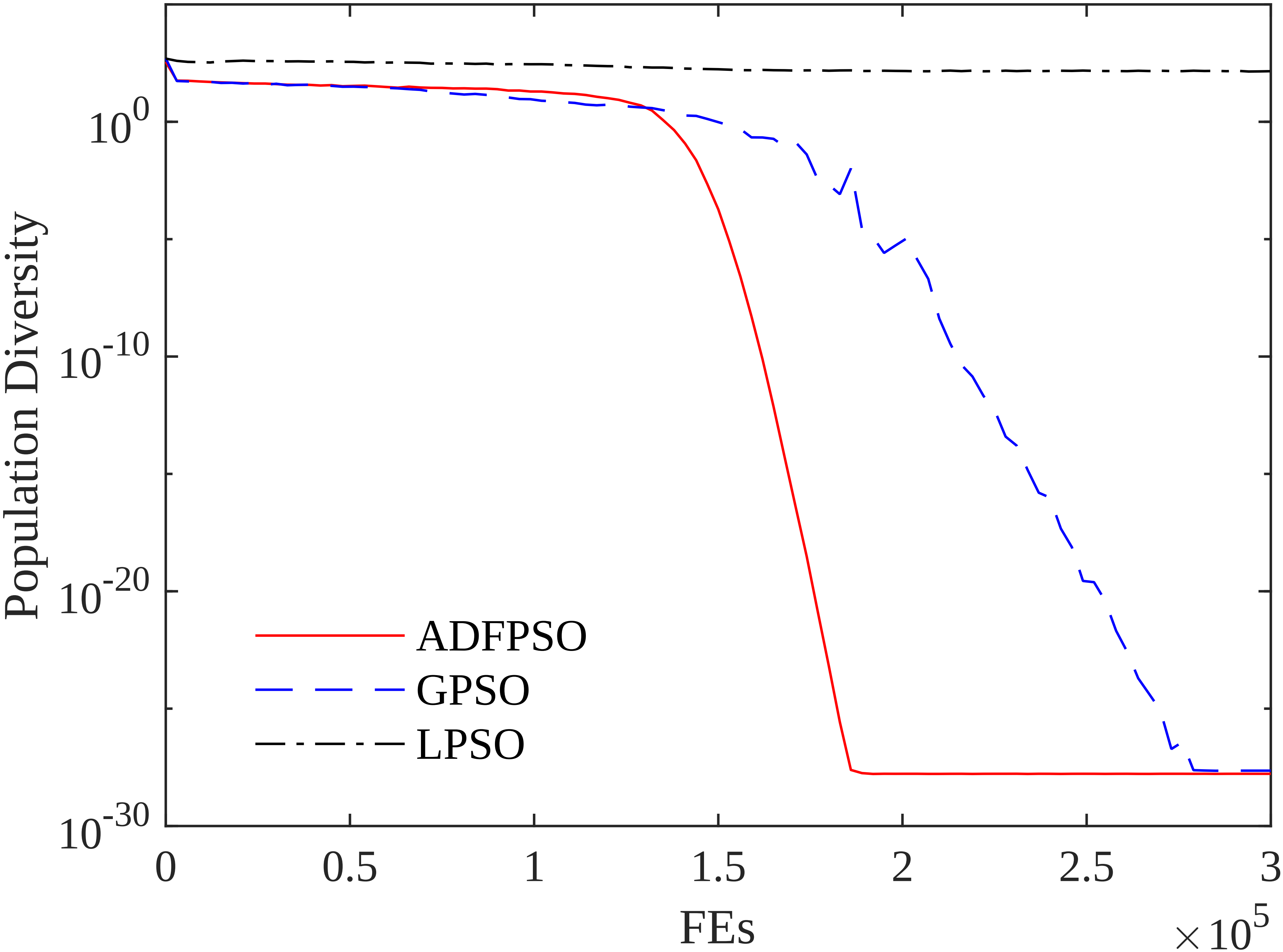

ADFPSO
GPSO
LPSO
21.4
21.3
21.2
21.1
21
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$



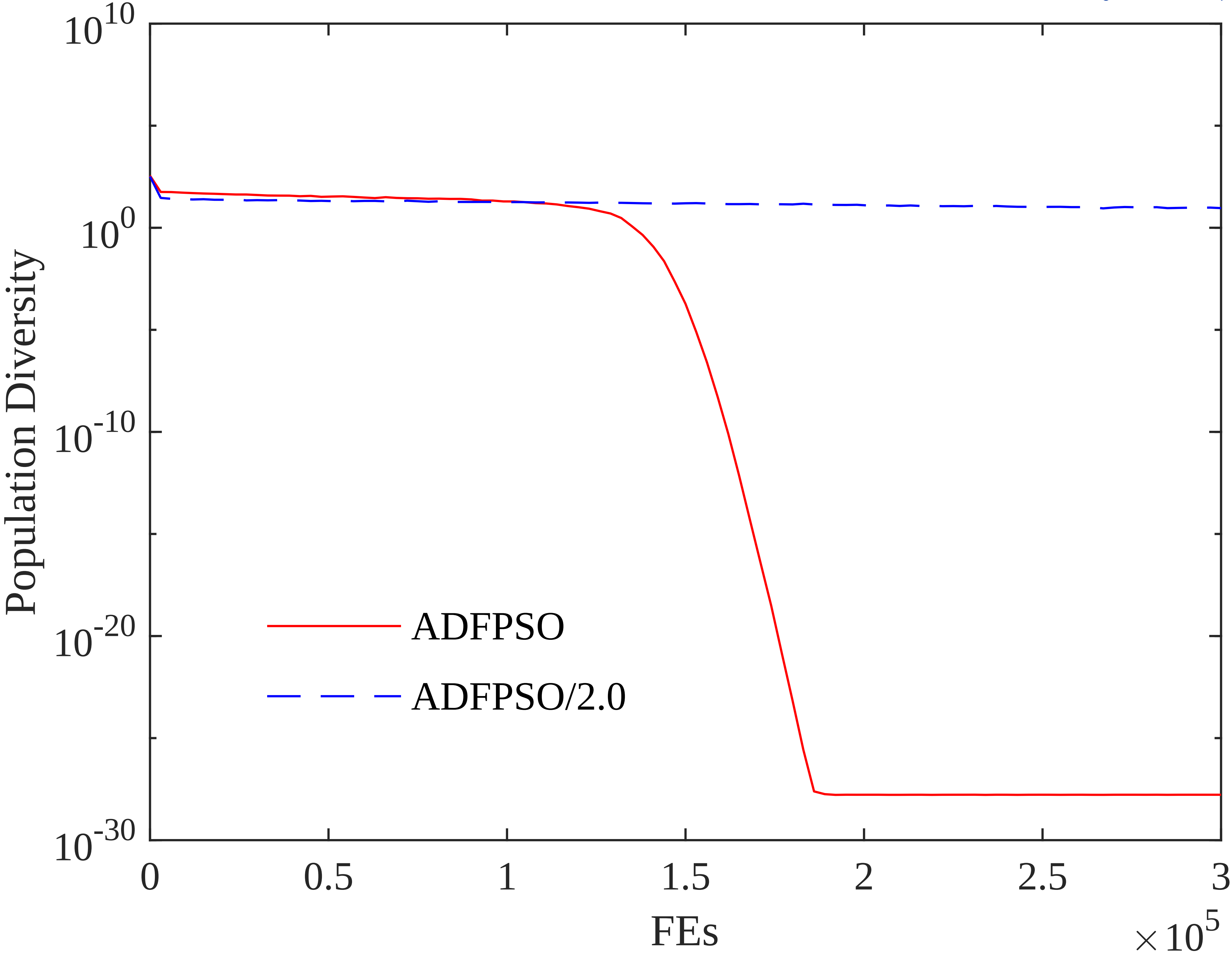
ADFPSO
ADFPSO/2.0
21.2
21.15
21.1
21.05
21
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$



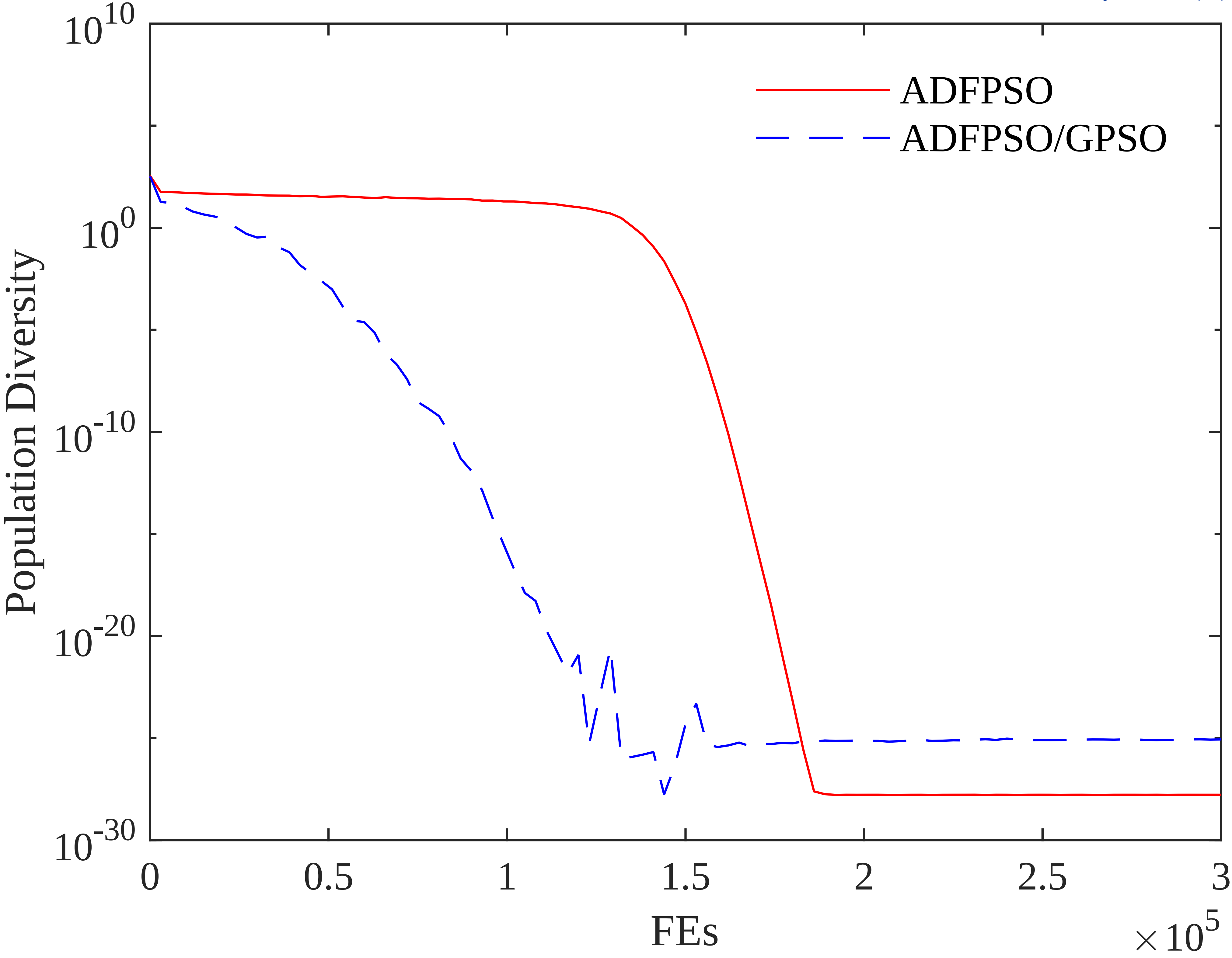
ADFPSO
ADFPSO/GPSO
Fitness
FEs
21.3
21.25
21.2
21.15
21.1
21.05
21
20.95
20.9
0
0.5
1
1.5
2
2.5
3
$\times 10^5$



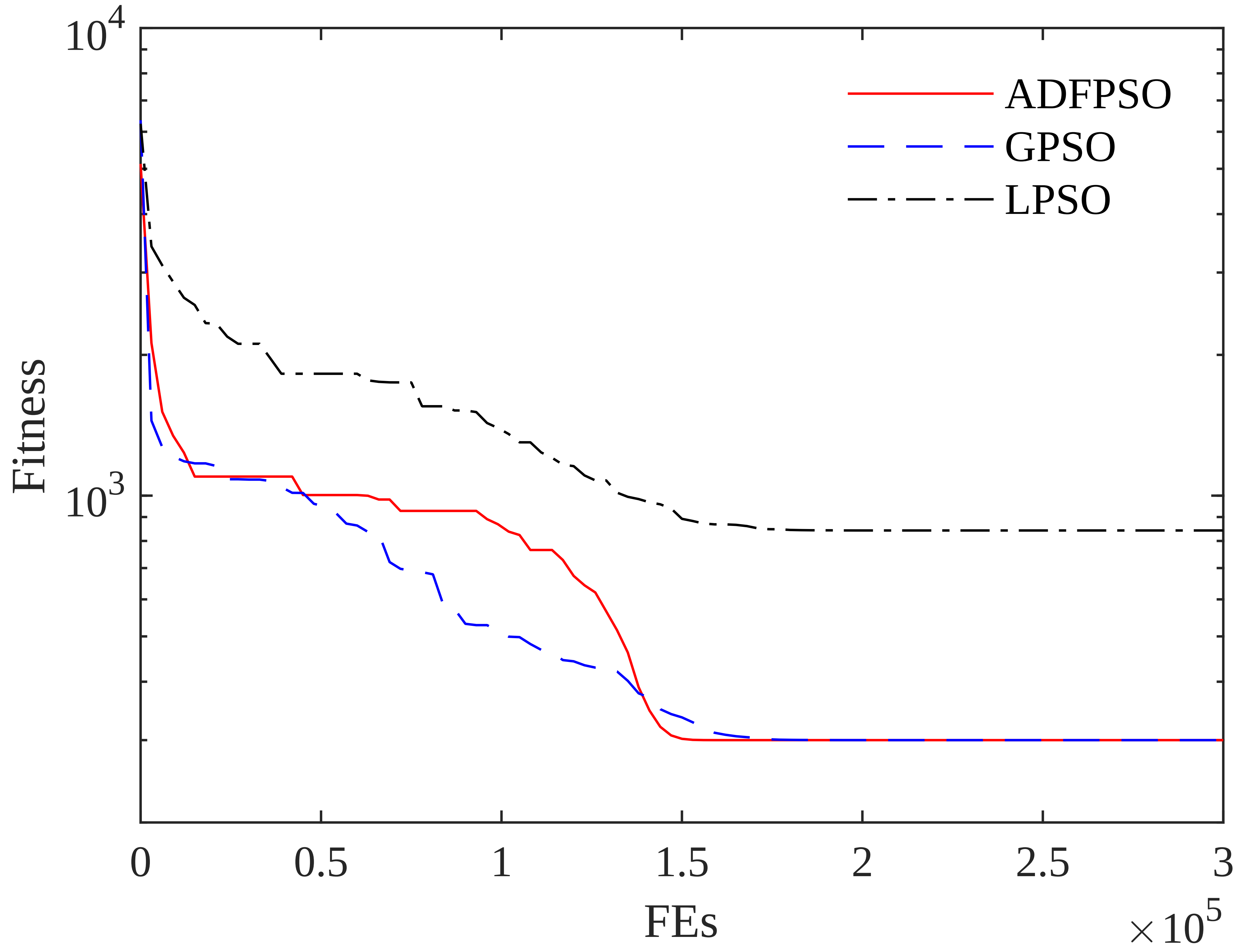
Population Diversity
10^0
10^-10
10^-20
10^-30
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
GPSO
LPSO



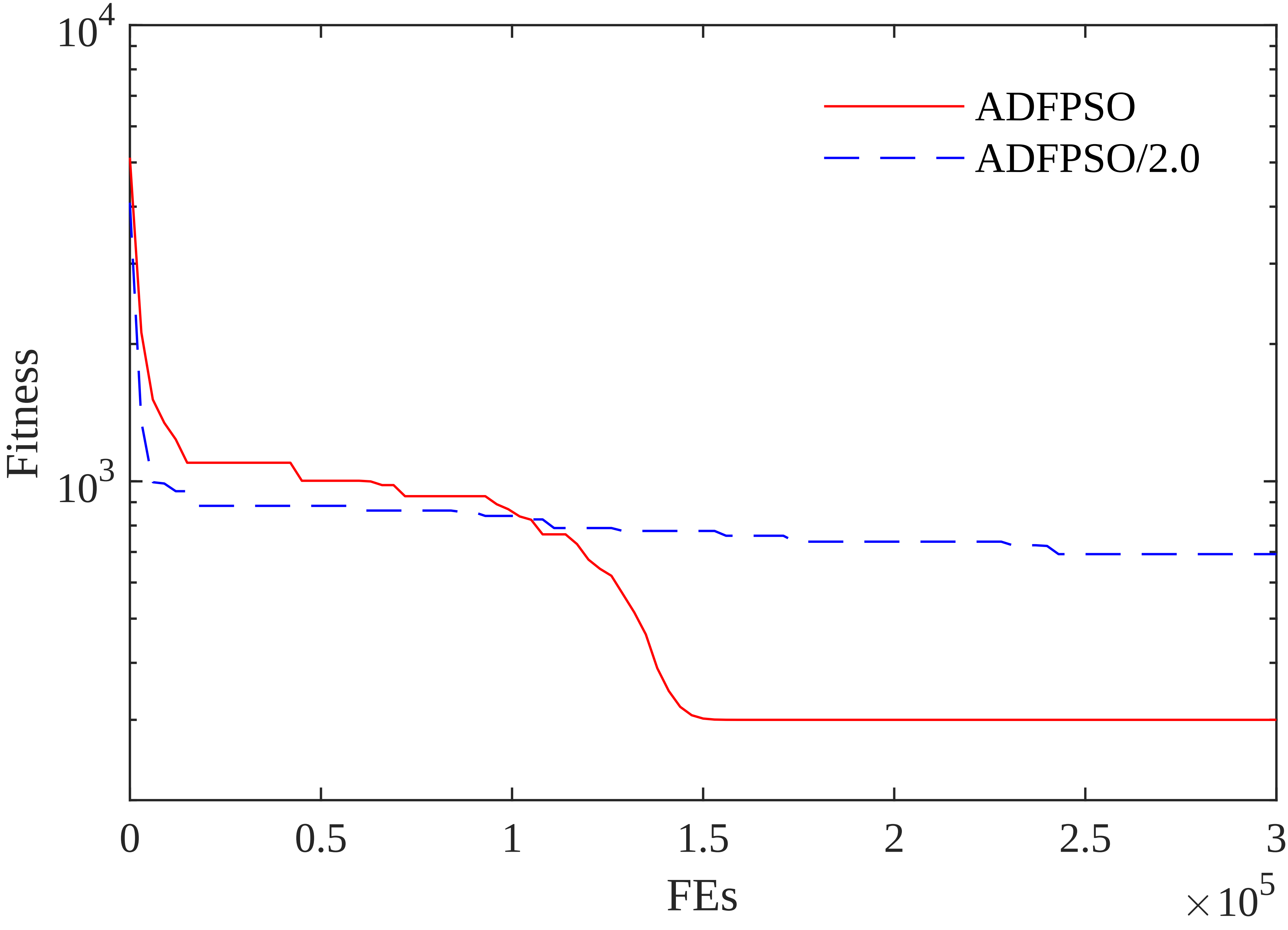
10^10
10^0
10^-10
10^-20
10^-30
Population Diversity
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
ADFPSO/2.0



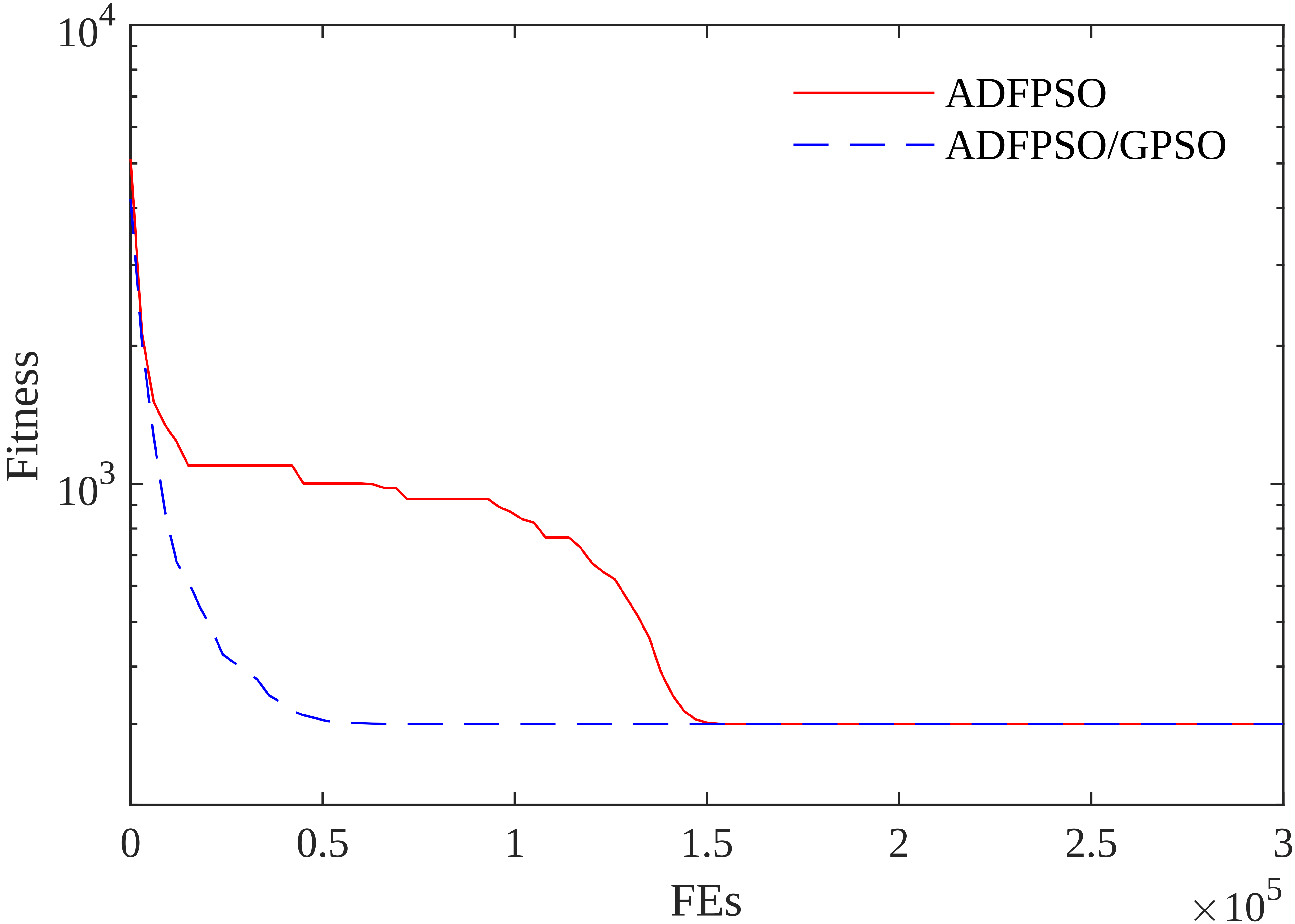
ADFPSO
ADFPSO/GPSO
Population Diversity
FEs
$\times 10^5$
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5
3



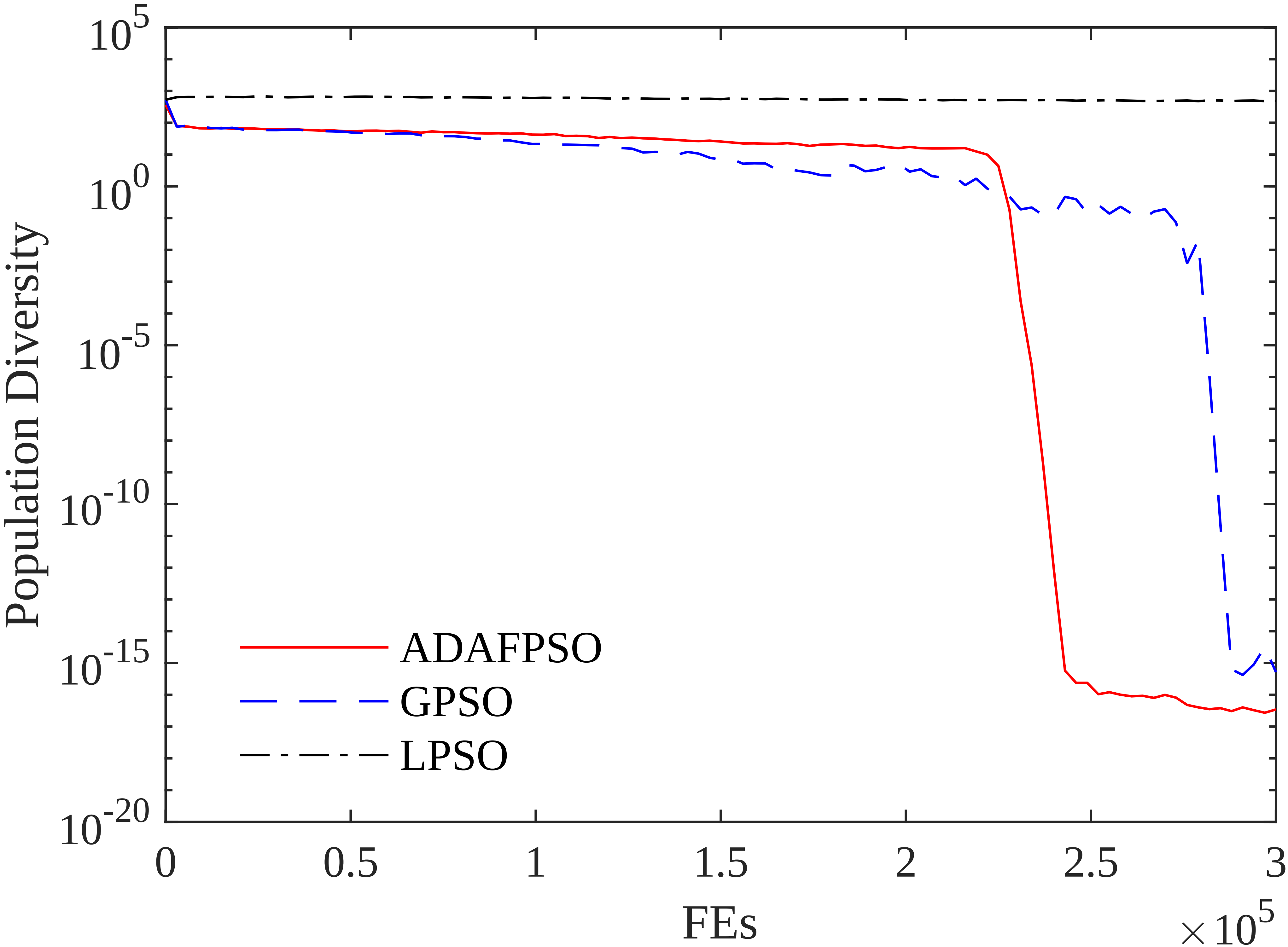
10^4
10^3
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$
ADFPSO
GPSO
LPSO



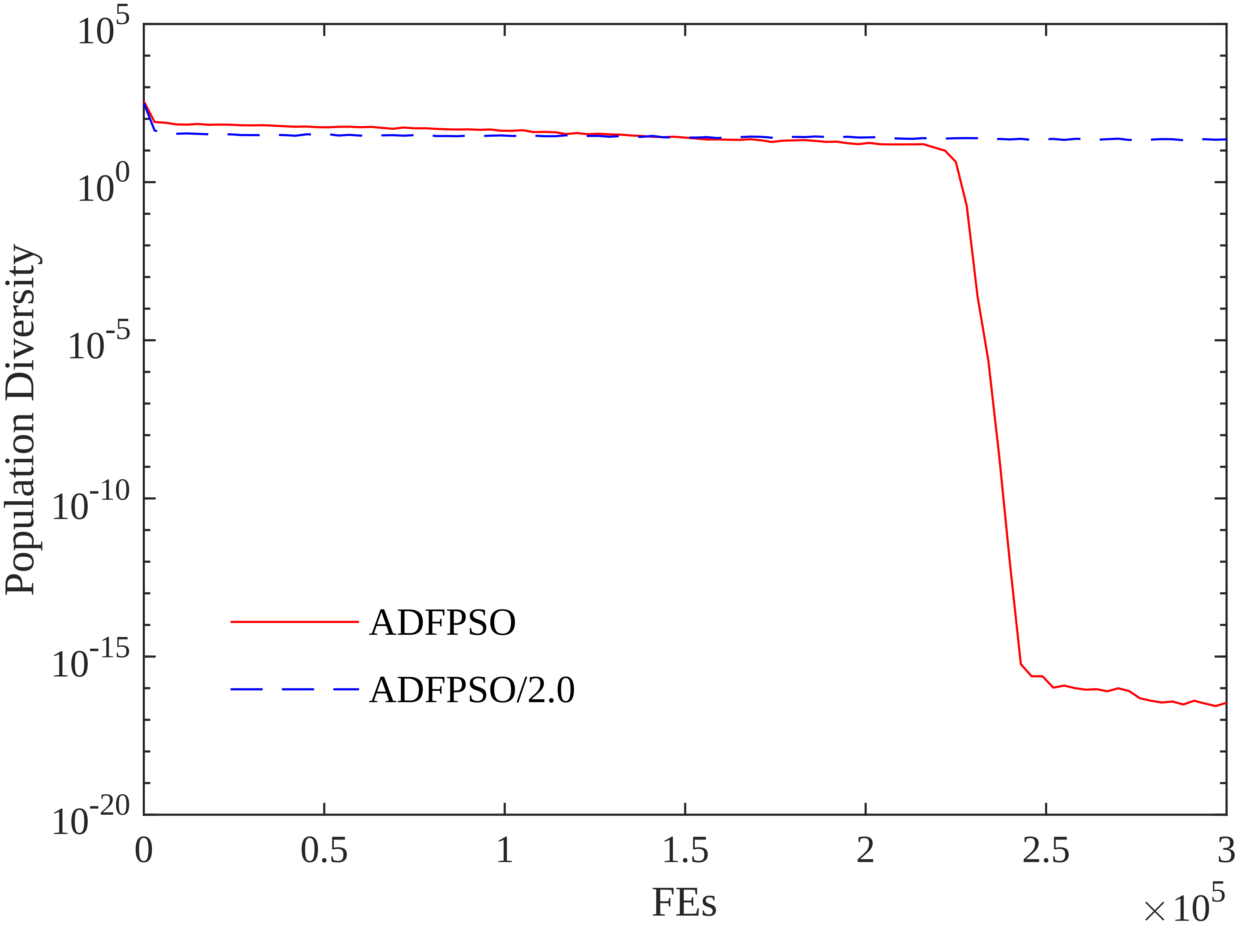
10^4
10^3
Fitness
ADFPSO
ADFPSO/2.0
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$

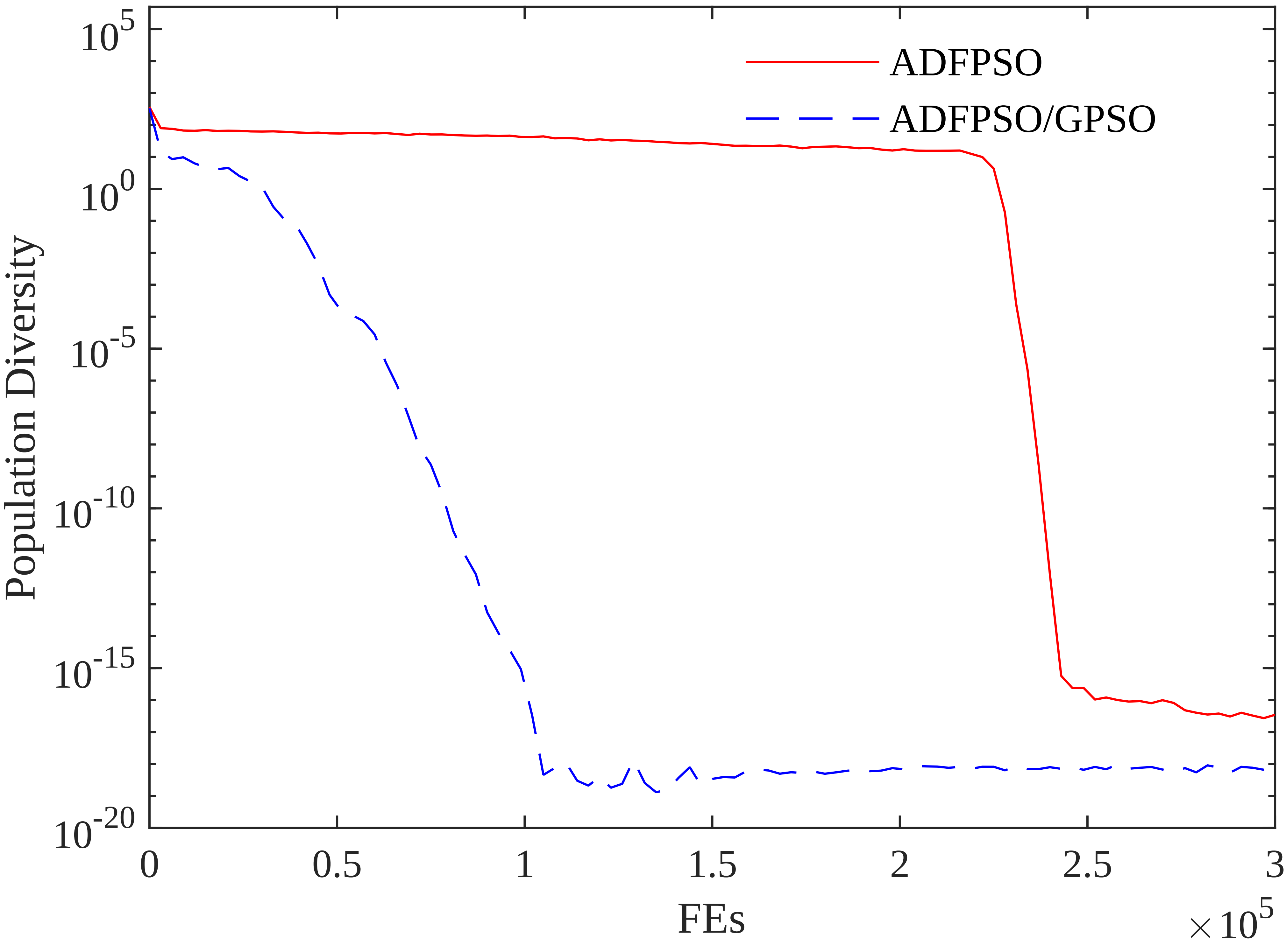


ADFPSO
ADFPSO/GPSO
Fitness
10^4
10^3
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$

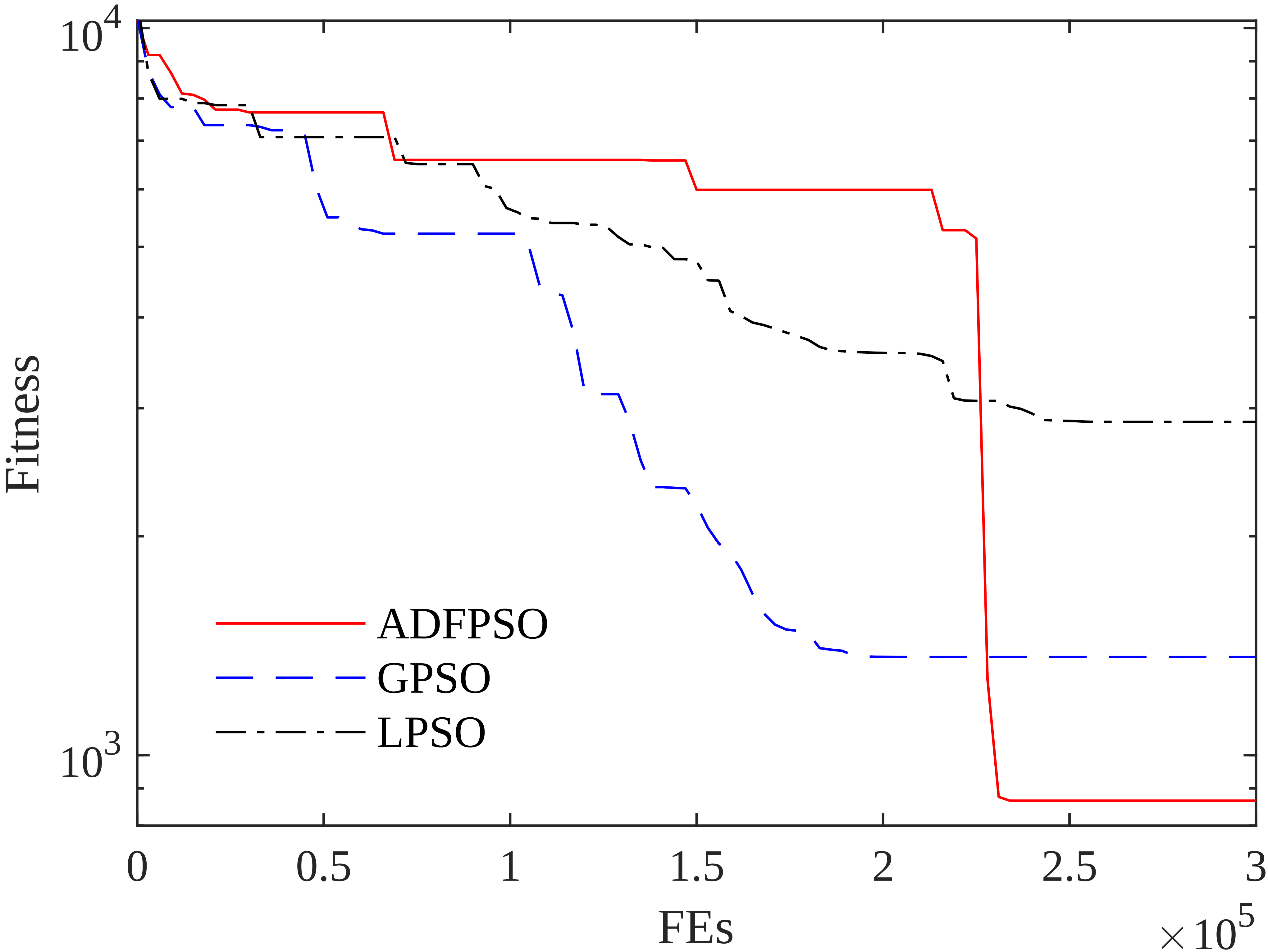


Population Diversity
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-20}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$
ADAFPSO
GPSO
LPSO

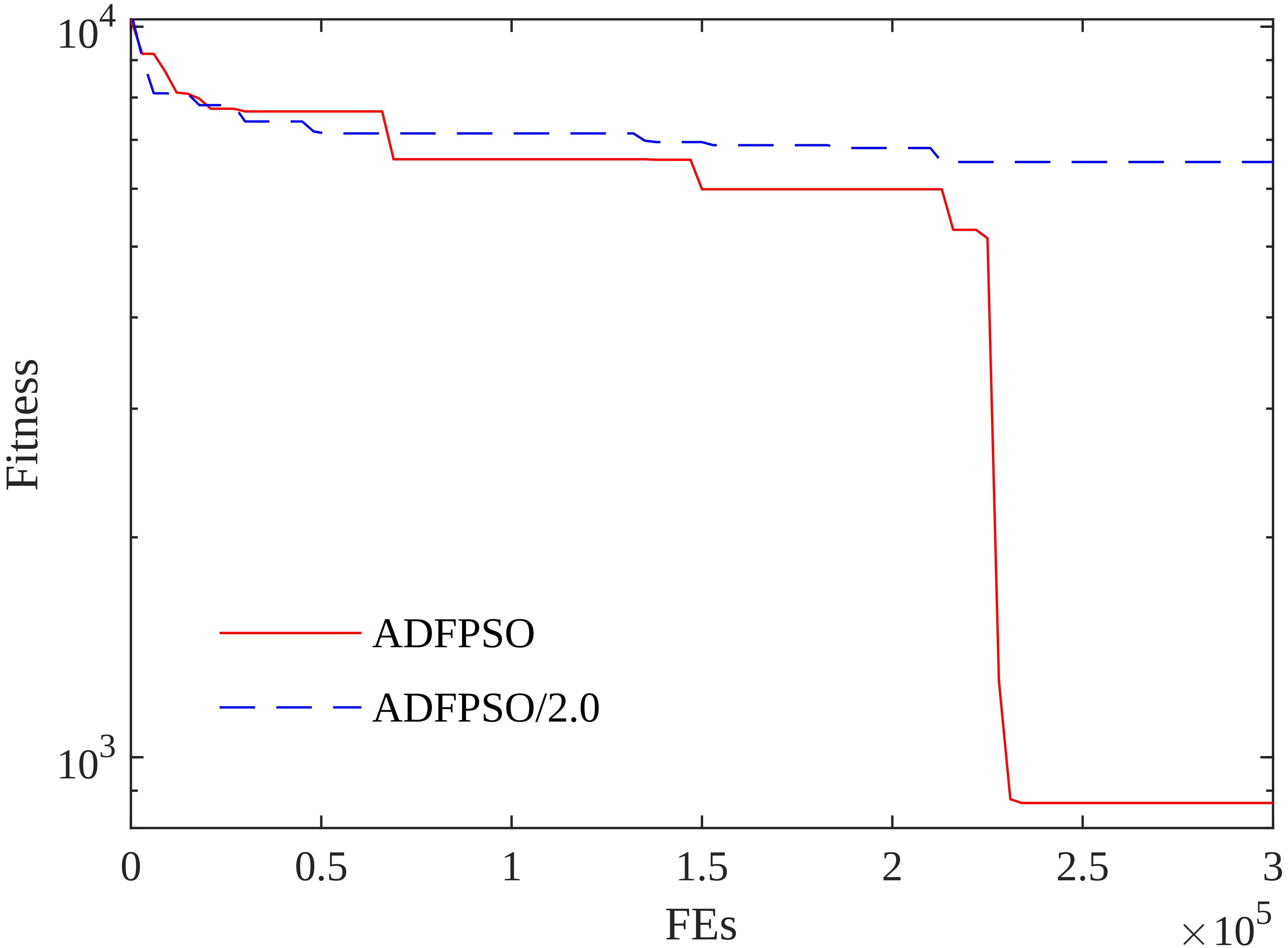

Population Diversity
10^5
10^0
10^-5
10^-10
10^-15
10^-20
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
ADFPSO/2.0



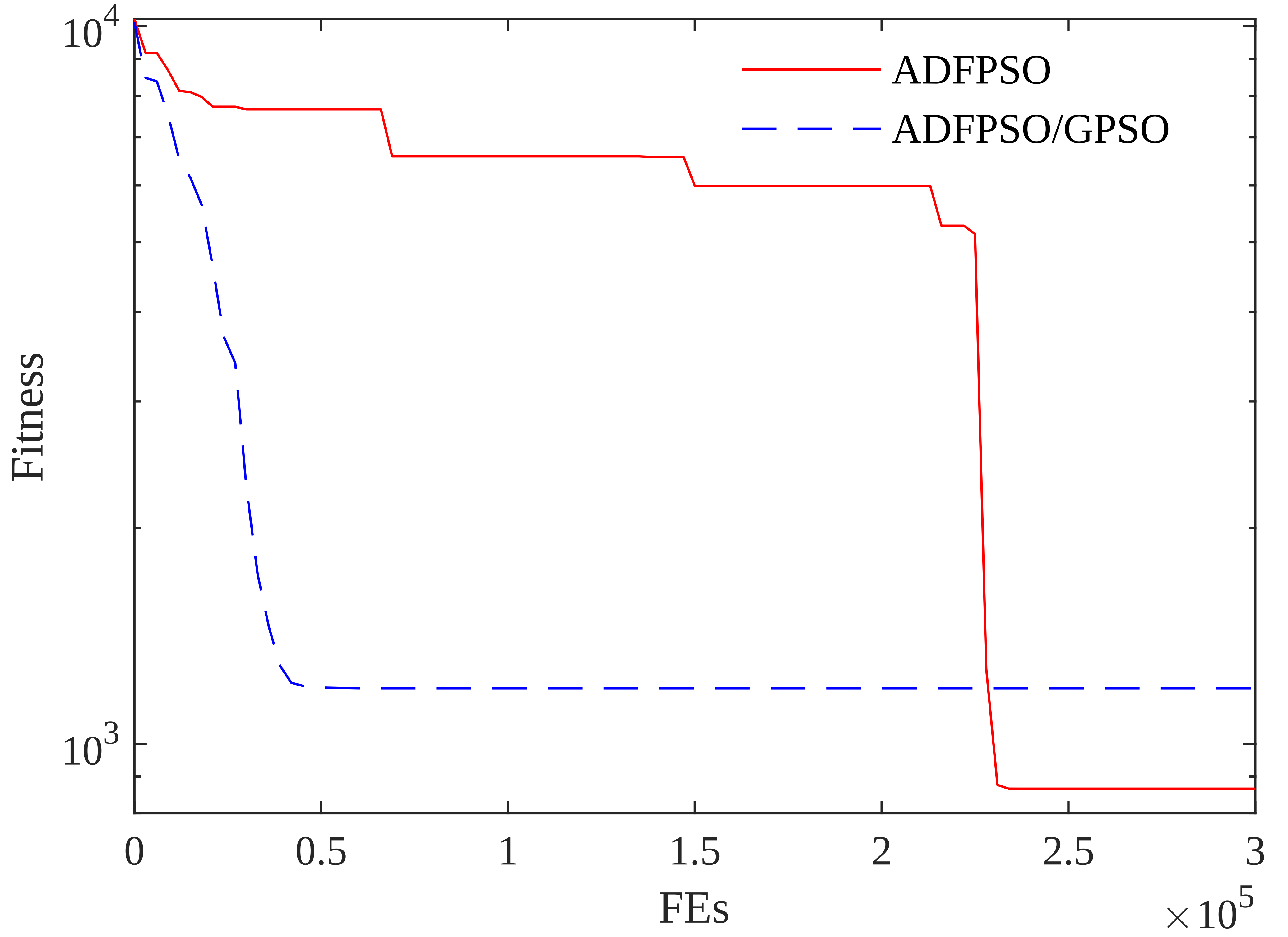
ADFPSO
ADFPSO/GPSO
Population Diversity
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-20}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$



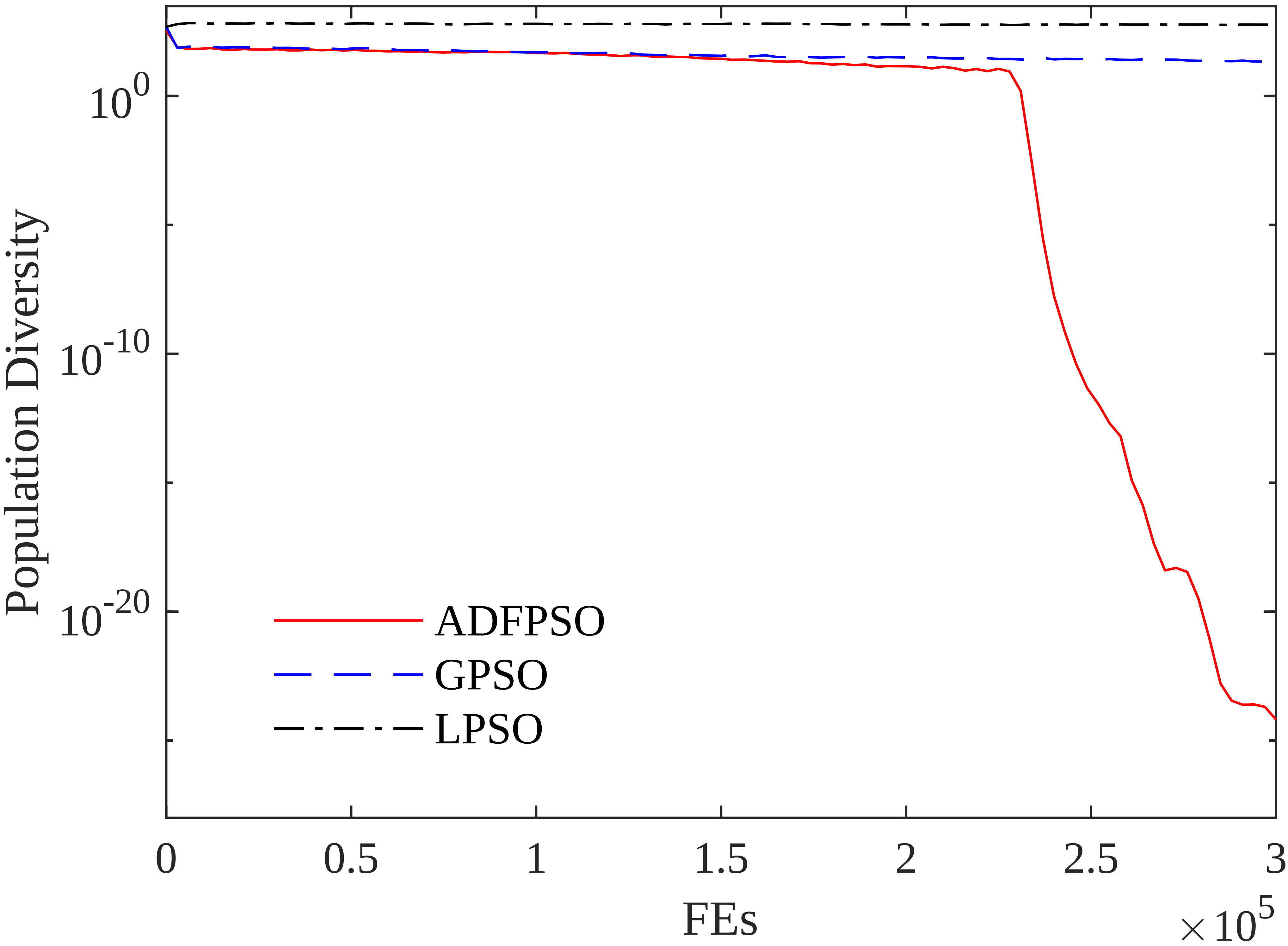
10^4
10^3
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
GPSO
LPSO

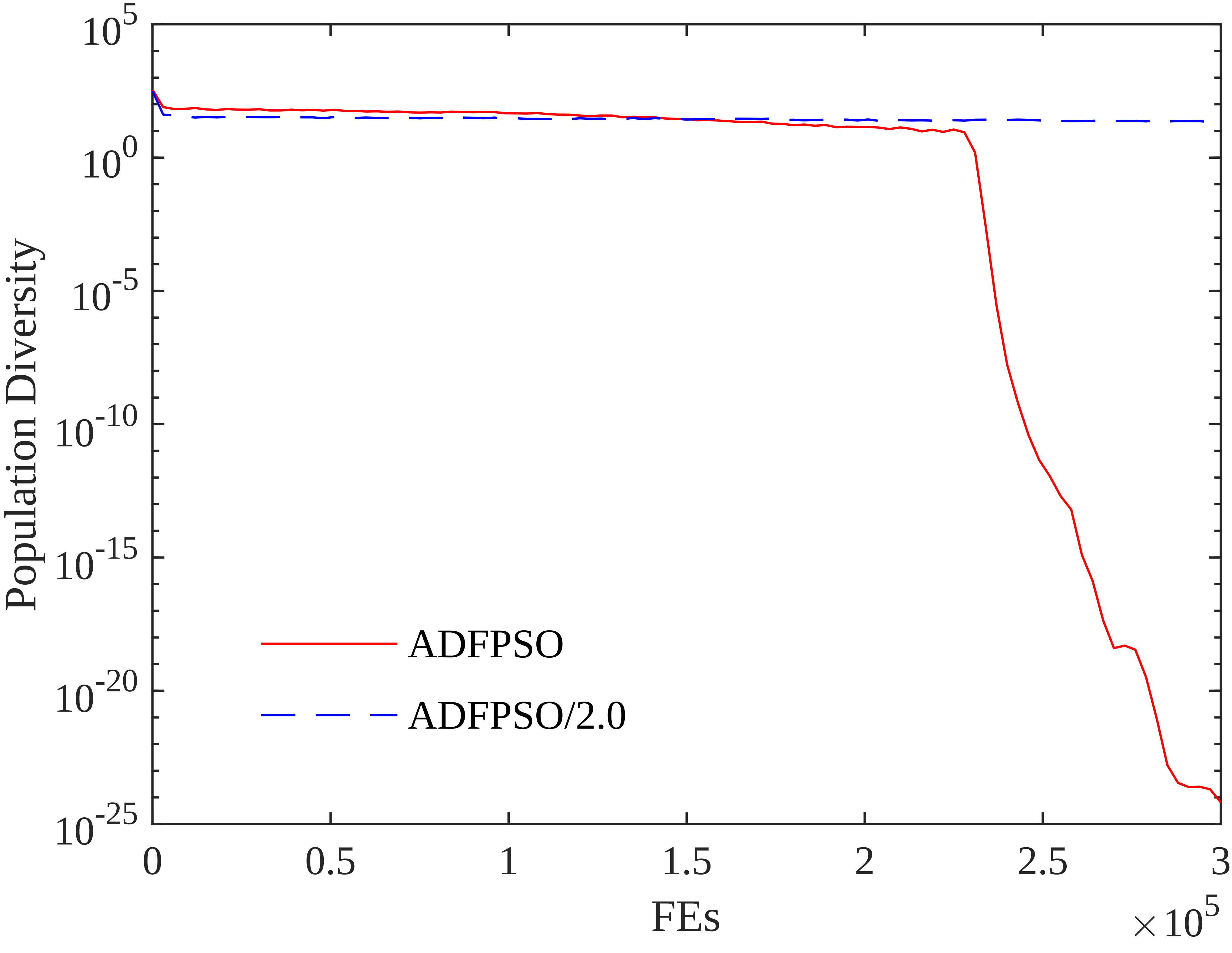


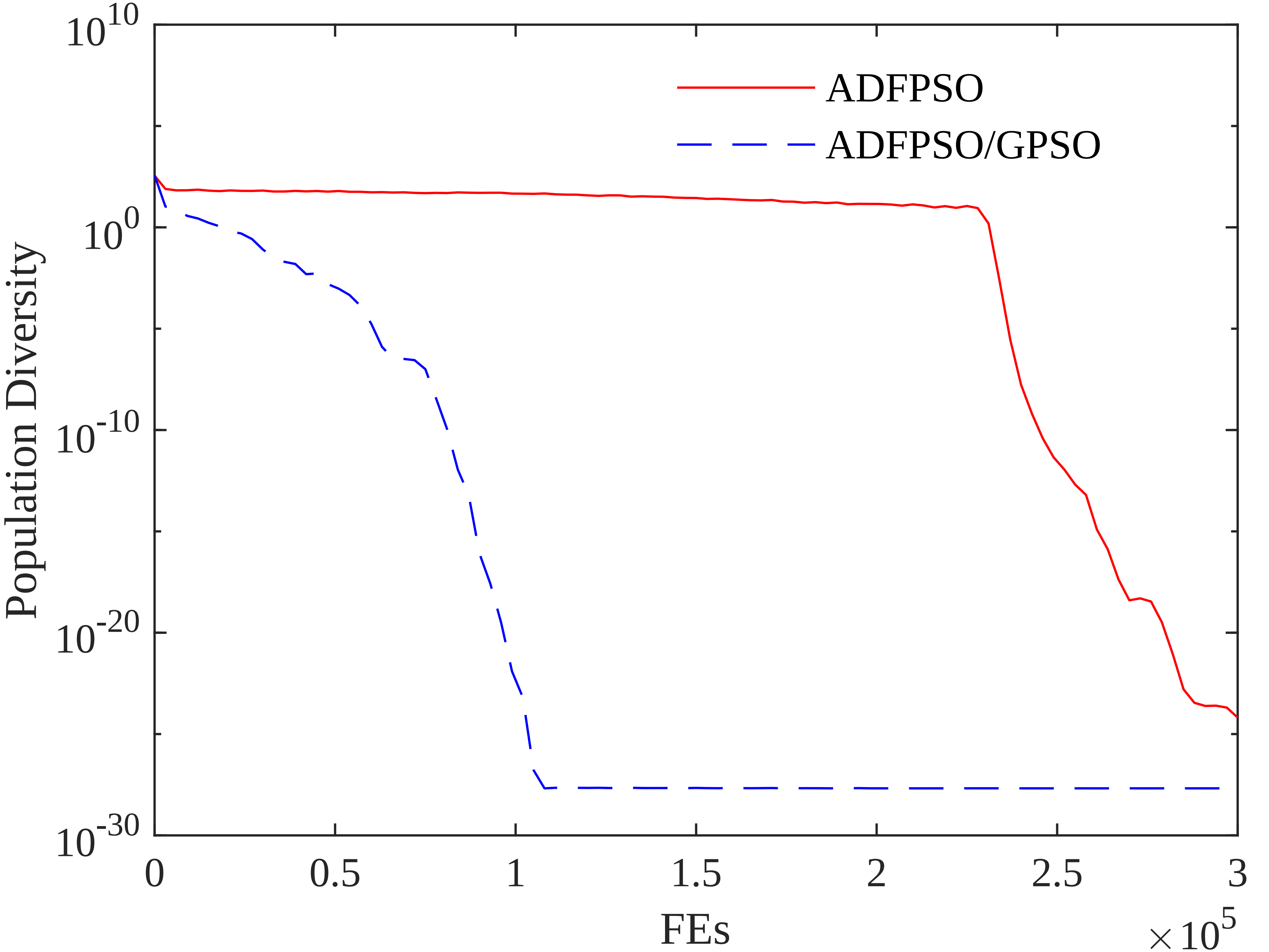
10^4
10^3
Fitness
ADFPSO
ADFPSO/2.0
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$



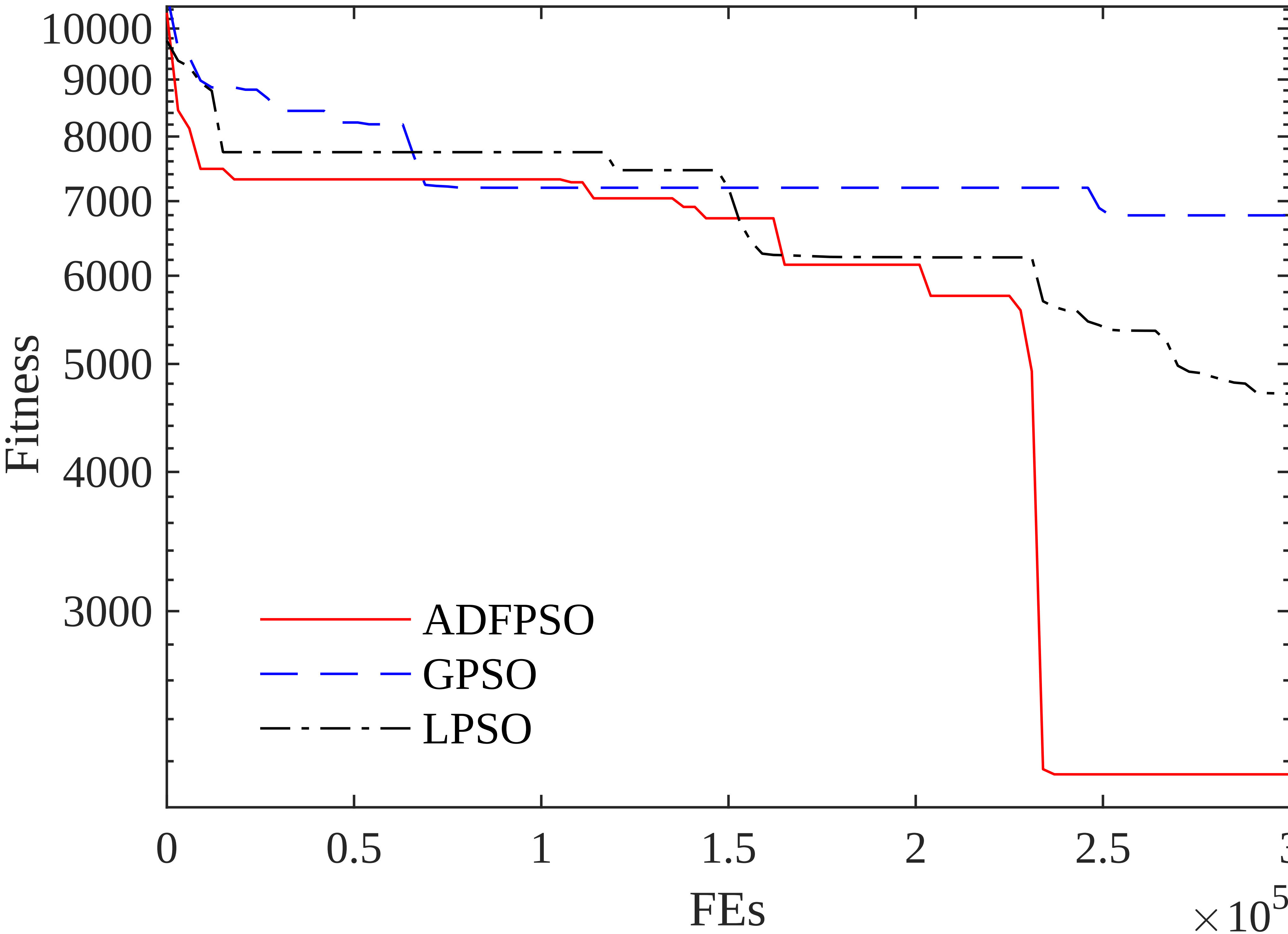
ADFPSO
ADFPSO/GPSO
Fitness
10^4
10^3
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$


Population Diversity
10^0
10^-10
10^-20
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
GPSO
LPSO

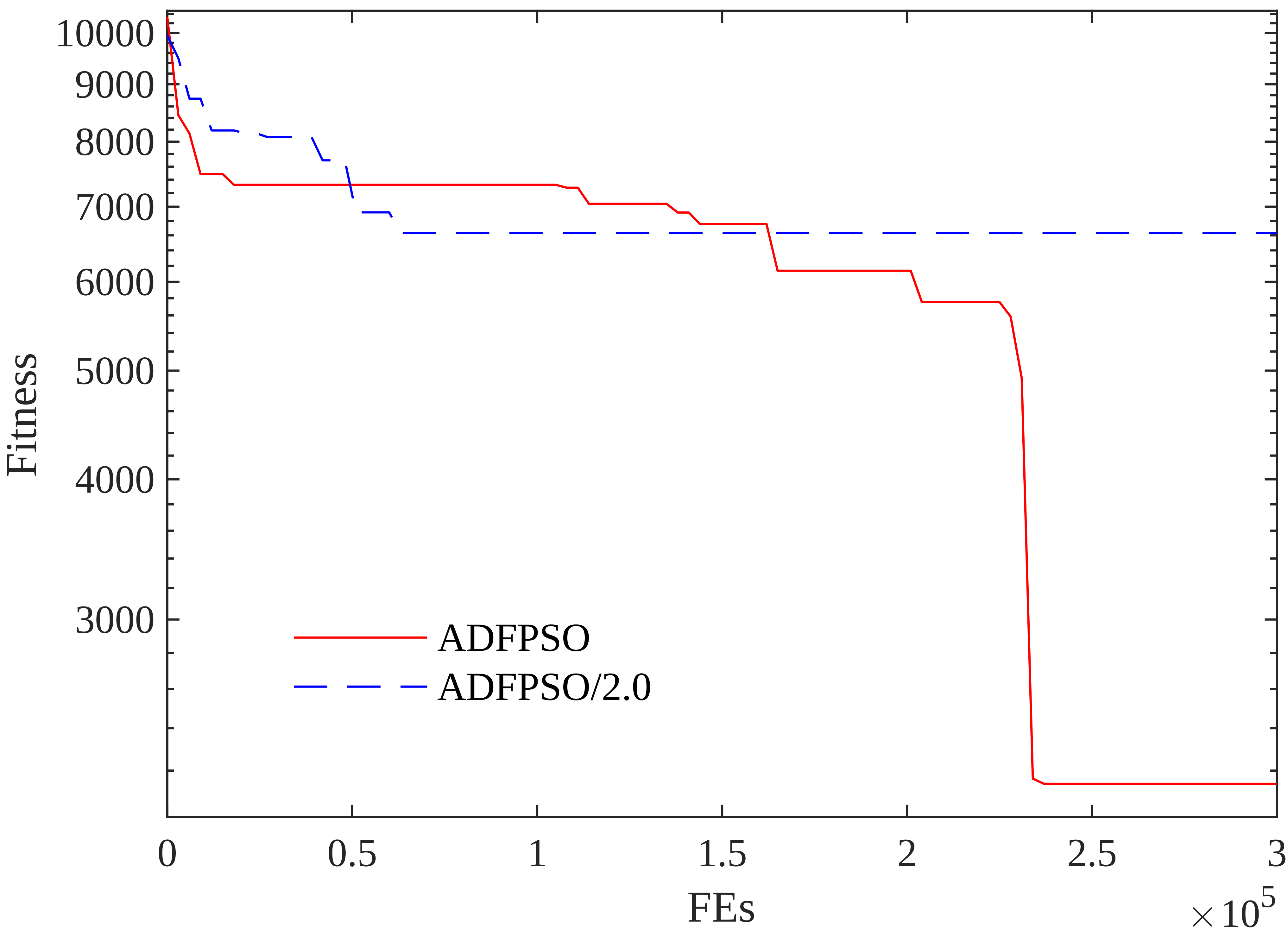

Population Diversity
FEs
$\times 10^5$
ADFPSO
ADFPSO/2.0
10^5
10^0
10^{-5}
10^{-10}
10^{-15}
10^{-20}
10^{-25}
0
0.5
1
1.5
2
2.5
3

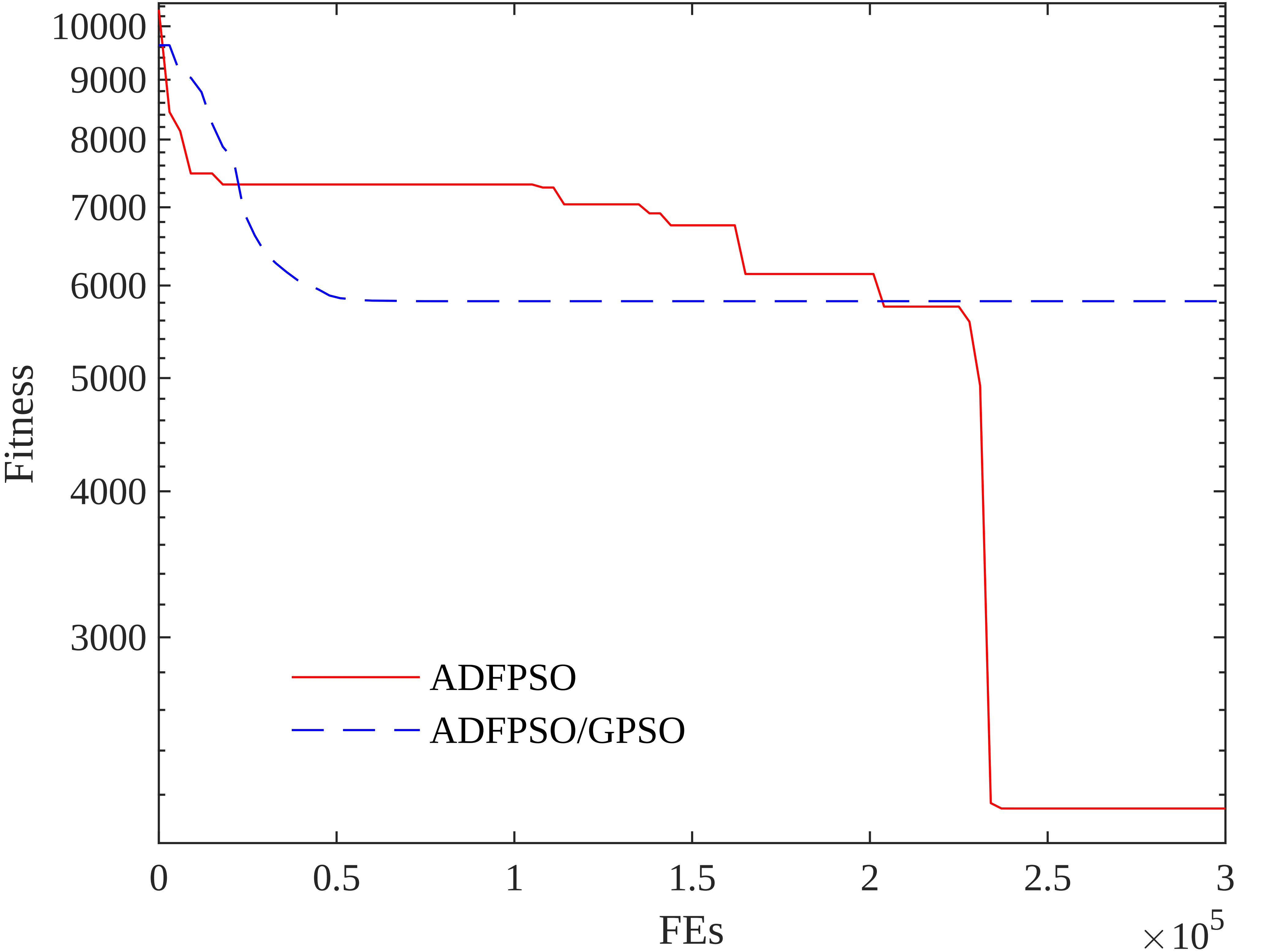


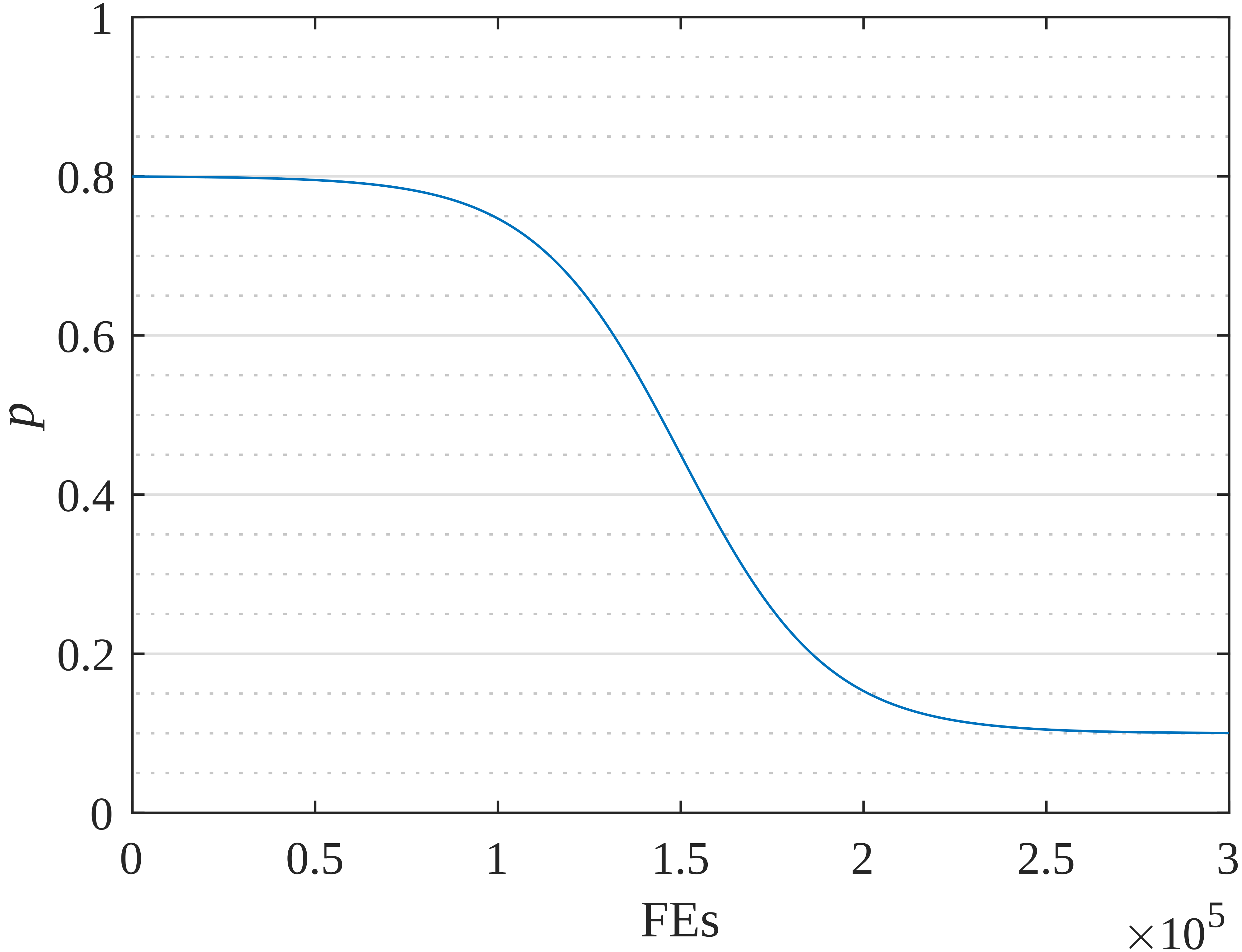
ADFPSO
ADFPSO/GPSO
Population Diversity
10^{10}
10^{0}
10^{-10}
10^{-20}
10^{-30}
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^{5}$



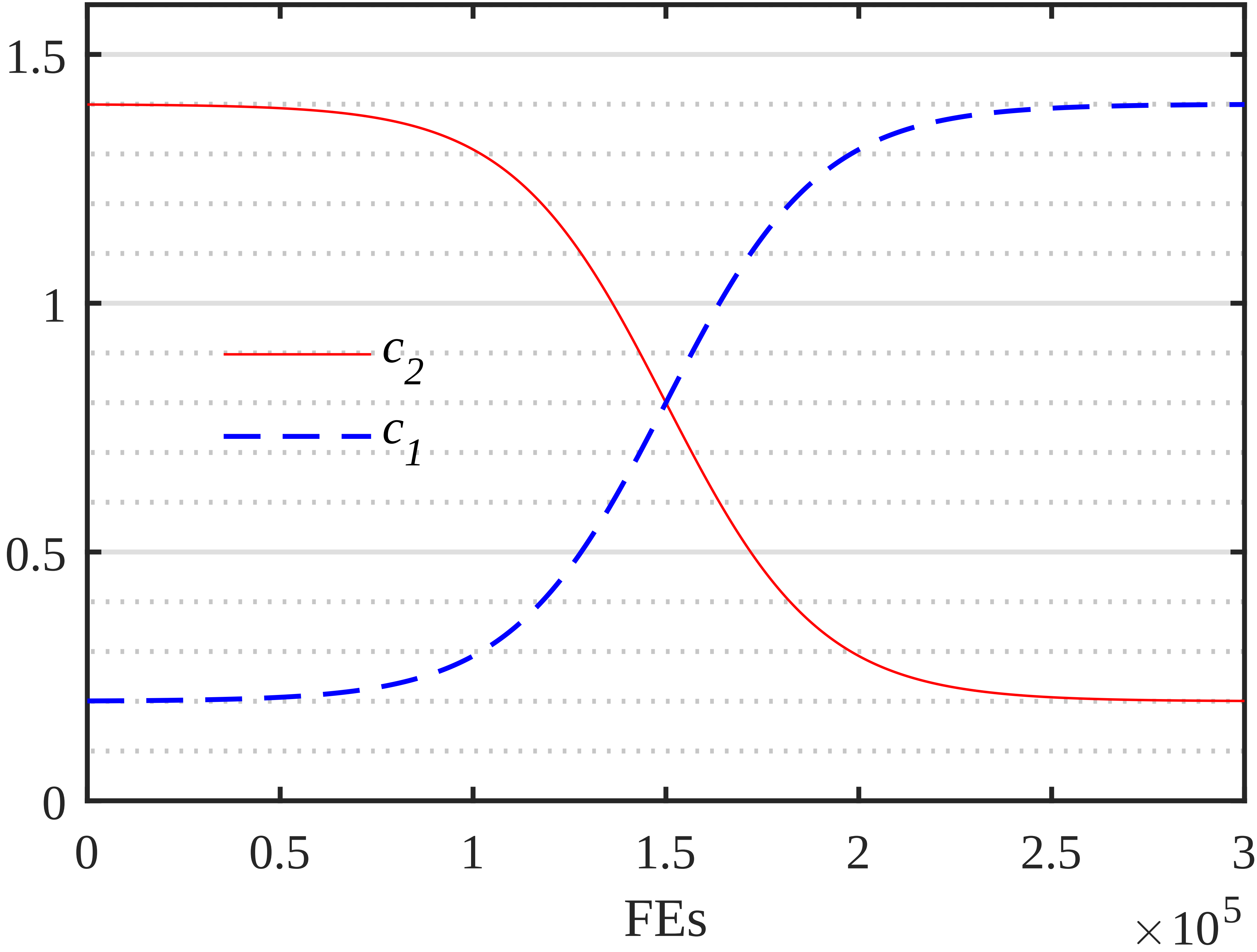
10000
9000
8000
7000
6000
5000
4000
3000
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$
ADFPSO
GPSO
LPSO

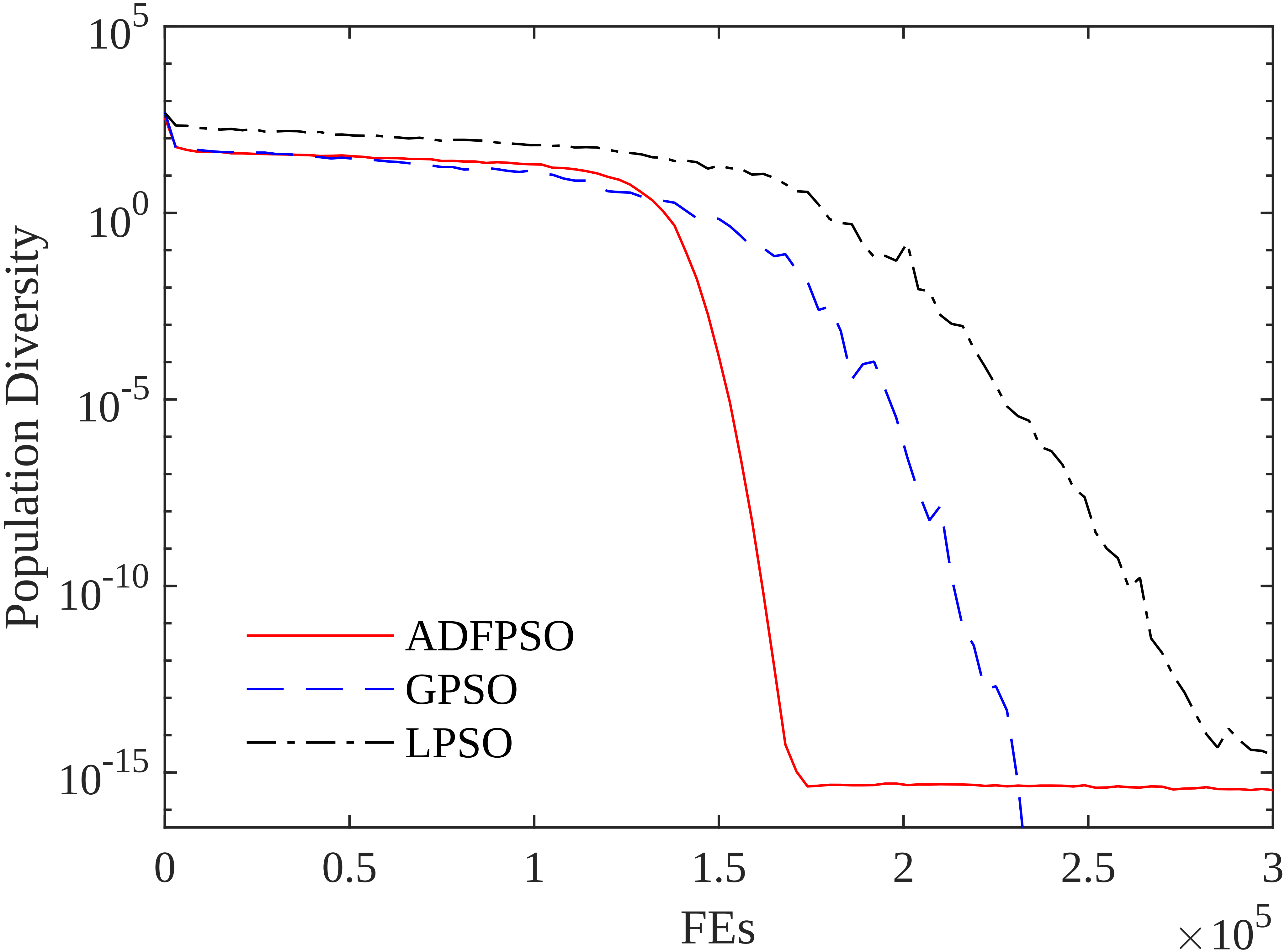

10000
9000
8000
7000
6000
5000
4000
3000
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$
ADFPSO
ADFPSO/2.0


10000
9000
8000
7000
6000
5000
4000
3000
Fitness
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$
ADFPSO
ADFPSO/GPSO

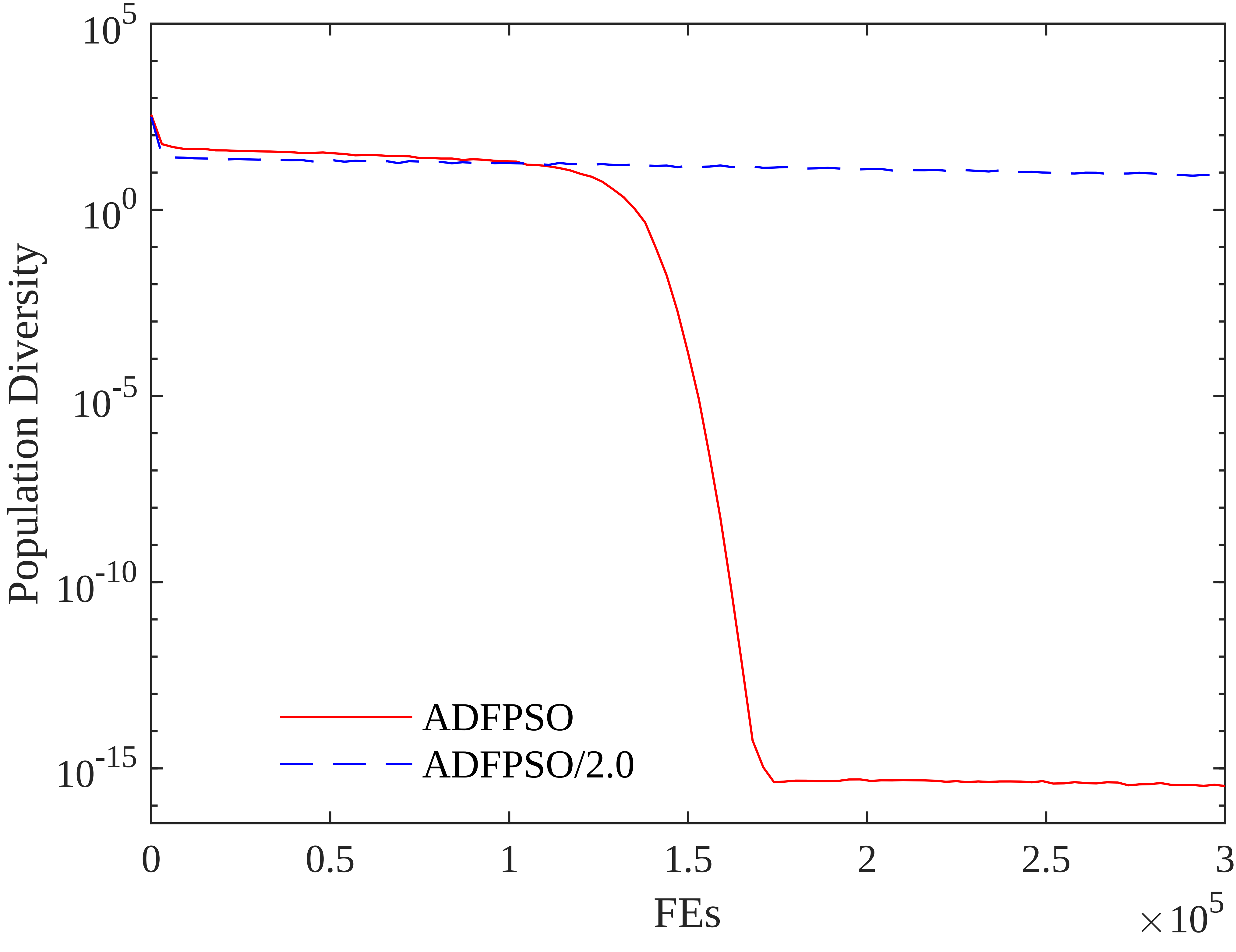


1
0.8
0.6
0.4
0.2
0
p
0
0.5
1
1.5
2
2.5
3
FEs
×10^5


1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$
c_2
c_1



Population Diversity
10^5
10^0
10^-5
10^-10
10^-15
0
0.5
1
1.5
2
2.5
3
FEs
×10^5
ADFPSO
GPSO
LPSO



10^5
10^0
10^{-5}
10^{-10}
10^{-15}
Population Diversity
0
0.5
1
1.5
2
2.5
3
FEs
$\times 10^5$
ADFPSO
ADFPSO/2.0

Dear Editors:

We would like to submit the manuscript entitled "Adjustable driving force based particle swarm optimization algorithm", which we wish to be considered for publication in "Information Sciences". No conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

We deeply appreciate your consideration of our manuscript, and we look forward to receiving comments from the reviewers. If you have any queries, please don't hesitate to contact me at the address below.

Thank you and best regards.

Yours sincerely,

Xuewen Xia, Fei Yu

xwxia@whu.edu.cn, yufei@whu.edu.cn