Guided Particle Adaptation PSO for Feature Selection on High-dimensional Classification

Mingshen Huang, Weiwei Yuan, Donghai Guan(✉), Mengze Lu, and Çetin Kaya Koç

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
dhguan@nuaa.edu.cn

Abstract. The Particle Swarm Optimization (PSO) algorithm, renowned for its efficiency and ease of implementation, is widely utilized in solving NP-hard problems, including feature selection. However, in high-dimensional data scenarios, most existing PSO-based feature selection methods typically employ a single filter-based approach for initializing particle populations, limiting the search range. We propose a guided particle adaptation method that integrates several filter-based methods to create guiding particles. These particles play a beneficial guiding role and expand the search range. Moreover, we introduce a new fitness factor promoting knowledge transfer under particle guidance, preventing premature convergence to global optima. Results demonstrate that this method efficiently obtains high-precision feature subsets.

Keywords: Feature Selection · Data Mining · Classification · Particle Swarm Optimization

1 Introduction

In the era of big data, data mining faces the challenge of data explosion [1], which includes dealing with high-dimensional datasets, especially in classification tasks. Feature selection (FS) is a crucial data preprocessing technique to mitigate the computational complexity of learning algorithms and improve classification accuracy [2].

Feature selection methods are typically categorized into three approaches: filter, wrapper, and embedded methods [3]. Filter methods use statistical measures for efficient FS [4], but they might offer moderate results independently of the learning algorithm. Wrapper methods treat FS as a search problem [5], often incorporating it with the learning model. Embedded methods integrate FS into the learning algorithm but can exhibit varying effectiveness [6].

Finding the optimal feature subset in FS is an NP-hard problem. In the context of n-dimensional feature data, there exist 2^n feature subset combinations. Therefore, optimization algorithms have gained attention as a means to address these challenges in FS research.

D.-S. Huang et al. (Eds.): ICIC 2024, LNCS 14862, pp. 14–26, 2024.
https://doi.org/10.1007/978-981-97-5578-3_2

Particle Swarm Optimization (PSO) is a population-based optimization algorithm with global search capabilities. PSO offers advantages such as efficiency, ease of implementation, and minimal hyperparameter tuning compared to other optimization algorithms. It is also commonly applied to FS, with its effectiveness validated in this context.

In this paper, we introduce GPAPSO (Guided Particle Adaptation PSO), a FS method that leverages multiple initialization techniques. During the particle swarm initialization, we integrate various filtering FS methods to initialize particle populations. Each populations utilizes statistical metrics for selecting feature subsets. Our contributions can be summarized as follows:

1) We present a method for guiding particle creation, integrating multiple approaches for particle swarm initialization. This method employs several populations, each applying three statistical measures to assess the feature subset of initialized particle populations, providing a beneficial guidance. Additionally, random particles are introduced to broaden the search space.
2) We establish an efficient multi-population knowledge transfer mechanism. Adaptive factors facilitate knowledge transfer among multiple populations, overcoming previous limitations. This enhancement helps particles better avoid local optima after knowledge transfer, leading to improved objective scores for the particle swarm.
3) An improved FS method is proposed based on the above two strategies, and its effectiveness is evaluated on 10 real-world high-dimensional datasets with the number of features ranging from 3000 to 12000. Experimental results demonstrate that GPAPSO outperforms several advanced EA-based FS methods and some traditional methods by producing feature subsets of higher quality.

2 Background Study

2.1 Feature Selection

Feature selection (FS) is a data preprocessing method aimed at eliminating redundant and irrelevant features from the original feature set. Essentially, FS involves the process of identifying the optimal feature subset from the original feature set [7]. Assuming a dataset with a total of N features and M samples, where F represents the original feature set, FS can be described as the selection of n ($n < N$) features from F to enhance the performance of a given model. The goal in a classification problem can be to maximize accuracy or minimize error. During FS, the solution X can be represented as:

$$\mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N})$$

$$\mathrm{x}_\mathrm{n} \in \{0,1\} \quad \forall \mathrm{n} \in \{1,2,\ldots,\mathrm{N}\} \tag{1}$$

When $x_n = 1$, it indicates that the n-th feature is selected and retained; conversely, it is discarded. Taking the minimization of error rate $H()$ as an example, the FS problem can be formulated as:

$$\min \mathrm{H(X)}$$

$$\text{s.t.} \quad \mathrm{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N}) \tag{2}$$

2.2 Particle Swarm Optimization

Optimization algorithms have been recognized as one of the best tools for tackling combinatorial problems [8]. In recent years, researches indicate that optimization algorithms outperform exhaustive or greedy methods [9].

In PSO, each particle within the swarm has a position and velocity, representing its direction and speed for the next iteration. In an N-dimensional space, the position of the *i*th particle is denoted as $x_i = (x_{i1}, x_{i2},\ldots, x_{iN})$, and its corresponding velocity is given by $v_i = (v_{i1}, v_{i2},\ldots, v_{iN})$. Each particle has its own best-known position, denoted as $pbest_i$, and the globally best-known position *gbest* is shared among all particles. It is used to update the particle's velocity. The particle's position using the following update equations:

$$v_{in}^{t+1} = \omega * v_{in}^{t} + c_1 * r_1 * \left(pbest_{in} - x_{in}\right) + c_2 * r_2 * \left(gbest_n - x_{in}\right) \tag{3}$$

$$x_{in}^{t+1} = x_{in}^{t} + v_{in}^{t+1} \tag{4}$$

where t represents the current iteration, ω is the inertia weight, c_1 and c_2 are acceleration constants (referred to as cognitive and social parameters, respectively), and r_1 and r_2 are uniformly distributed random values within the interval [0,1].

3 Approach

To tackle the challenges posed by the curse of dimensionality in modern high-dimensional data, we present a method for creating guiding particles. We employs an adaptive factor for knowledge transfer between populations. We call this approach Guided Particle Adaptation.

3.1 GPAPSO Overall Algorithm

The flowchart of the GPAPSO algorithm is illustrated in Fig. 3. It takes three inputs: the population size n, the number of iterations (stopping criterion), and the parameter r which controls when knowledge transfer should occur. GPAPSO initially rearranges features by using filter-based methods to generate populations. Then, it computes the fitness functions of particles in each population and constructs an elite particle swarm. During the optimization process, when r generations pass without update, it will calculate the adaptive factor Pc and update the gbest of that population. This process is repeated until the maximum number of iterations is reached (Fig. 1).

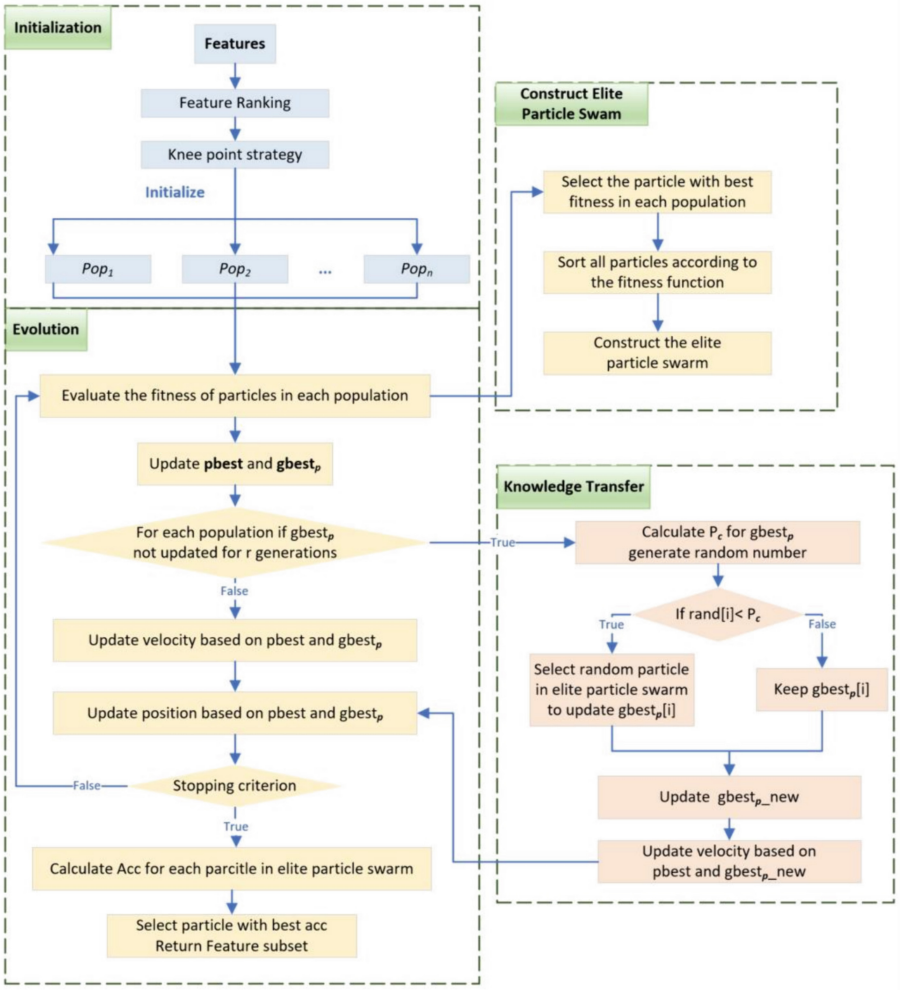

Fig. 1. Flowchart of GPAPSO algorithm.

3.2 Representation of GPAPSO's Particle

In previous methods like binary PSO [10], particle positions are restricted to either 0 or 1. Continuous PSO demonstrate superior performance. The GPAPSO algorithm proposed in this paper is based on a continuous PSO algorithm. Each particle, as illustrated in Fig. 2, is represented by two continuous vectors: velocity, which guides particle updates, and position, which is an n-dimensional vector within the range of [0,1].

This paper combines three different kinds of filter-based method to create guiding particles, significantly enhancing their guiding role.

The ANOVA F-value is computed by comparing the average variances between groups and within groups. Specifically, the F-value is represented by the following

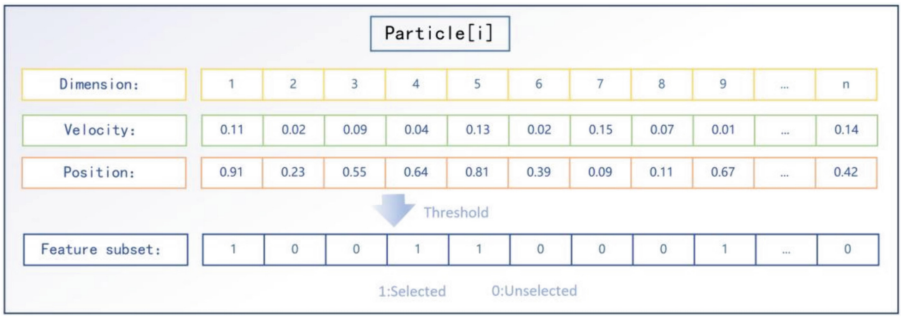

Fig. 2. Representation of GPAPSO's Particle.

formula:

$$F = \frac{MSB}{MSW} \tag{5}$$

where *MSB* stands for the between-group average variance, and *MSW* represents the within-group average variance.

The ReliefF algorithm randomly selects a sample R from the training dataset and identifies h nearest neighbors, both from the same class (H_j, where $j = 1,2,\ldots,h$) and different classes (M_j, where $j = 1,2,\ldots,h$). The calculation formula for feature α is as follows:

$$W(\alpha)^{t+1} = W(\alpha)^{t} - \sum_{j=1}^{h} \frac{\mathrm{diff}(\alpha, R, H_j)}{mh} + \frac{\sum_{c \notin s(R)} \left[\frac{p(c)}{1-p(s(R))} \sum_{j=1}^{h} \mathrm{diff}(\alpha, R, M_j) \right]}{(mh)} \tag{6}$$

where $W(\alpha)$ represents the weight of feature α; t denotes the number of iterations; m represents the number of samples; $s(R)$ is the class label of sample R; $p(c)$ and $p(s(R))$ indicate the proportions of class c and class R; $diff(a,R_1,R_2)$ denotes the difference between the values of feature α for samples R_1 and R_2.

Finally, we utilize the feature weight Gini coefficient. For a feature α, its Gini coefficient can be computed using the following formula:

$$GINI(\alpha) = 1 - \sum_{i=1}^{n} p_i^2 \tag{7}$$

where p_i represents the proportion of samples from the i-th class within the current node's samples.

3.3 Particle Initialization

In GPAPSO, a knee-point strategy [11] is used to categorize all features into estimated important features and estimated redundant features [12]. This approach, widely applied

in many optimization problems for determining knee-points [13], has been employed in recent algorithms for generating initial particles. In the knee-point strategy, features are sorted in descending order based on their weights. A line is drawn between the highest and lowest feature, and the point on the curve with the maximum distance to line L is set as the knee-point. Features beyond the change point, as well as those ranked before it, are referred to as estimated important features, while the remaining features are labeled as estimated redundant features.

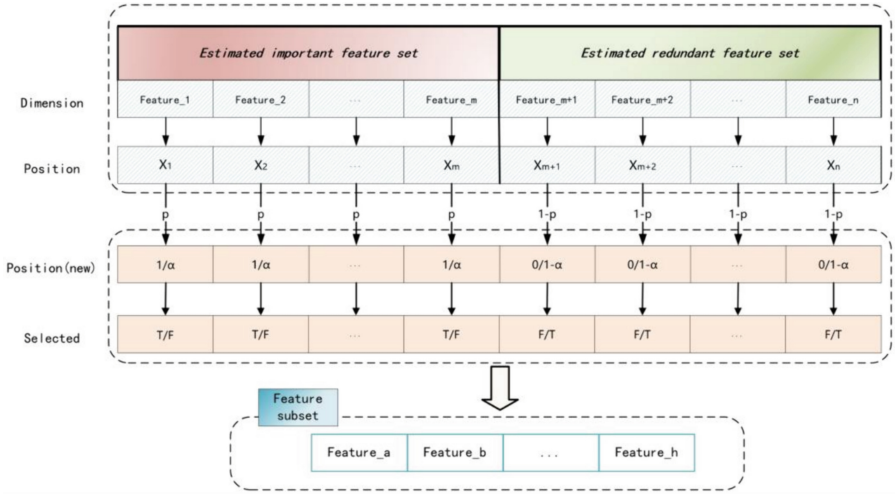

Fig. 3. Particle Initialization.

Features in the estimated important feature set are obviously more likely to be important than those in the estimated redundant feature set. Therefore, during particle initialization, estimated important features are more likely to be chosen than estimated redundant features. The calculation formula for the selection probability p is as follows:

$$\mathrm{p} = \frac{\mathrm{N_i}}{\mathrm{N_i} + \mathrm{N_r}} \tag{8}$$

where N_i is the number of features in the estimated important feature set, and N_r is the number of features in the estimated redundant feature set.

Figure 3 shows particle initialization. Features will be selected based on whether it belongs to the estimated important feature set. If the feature is part of the estimated important feature set and the random value is less than p, set it to 1; otherwise, it is set to α. For the features in the estimated redundant feature set, they are set to 0 if the random value is less than $1 - p$; otherwise, they are set to $1 - \alpha$. Here, α is half the difference between the threshold and 1.

3.4 Knowledge Transfer Mechanism

Before knowledge transfer, it is essential to clarify how the populations are partitioned. Three methods generate different guiding particles, and random particles are also generated to broaden the search space. Within a population, we ensure an equal number of these three types of guiding particles and an equal number of random particles. Previous research indicates that when using PSO for FS, the number of particles in a population is typically around 1/20 of the total feature count, usually not exceeding 200.

The proposed GPAPSO algorithm uses the adaptive factor in multi-population knowledge transfer. We first extract the global best particles from each population and construct an elite particle swarm. During the optimization search process, when the gbest of the current population remains unchanged for several generations, the gbest is updated using an adaptive factor. This method involves calculating an adaptive factor value, denoted as *Pc*, for each gbest of a particle, and it is computed as follows:

$$Pc = 0.05 + 0.45 * \frac{\exp \frac{10(rank(i)-1)}{S-1}}{\exp^{10} - 1} \tag{9}$$

where S is the elite particle swarm size and rank(i) is the rank of elite-particle[i]. The best particle in the swarm will be ranked 1.

Subsequently, a random number is generated for each dimension of gbest to determine whether the particle's position in that dimension needs to be updated. If the generated random number is less than *Pci*, a particle is randomly selected from the elite particles for updating. When a particle has a smaller *rank*(*i*), its *Pci* is smaller, resulting in a lower probability of learning from other particles.

3.5 Fitness Function

In FS tasks, the two primary objectives are to maximize classification accuracy and minimize the size of the feature subset. In researches [14], a combined fitness function considers both of these objectives. The fitness function is defined as follows:

$$\text{fitness} = \beta * \text{ER} + (1 - \beta)\frac{\text{Selected}}{\text{All}} \tag{10}$$

where *ER* represents the classification error rate of the learning algorithm, *Selected* is the number of selected features, and *All* is the total number of available features in the training dataset.

4 Experiment

4.1 Datasets

We use datasets, each containing over a thousand features. These datasets are publicly available at http://featureselection.asu.edu. Table 1 provides information for each dataset.

4.2 Comparative Methods

To evaluate the proposed method, we selected the following advanced methods and traditional optimization algorithms as comparison methods:

1) PSO-based FS with the variable-length strategy (VLPSO) [15].
2) VS-CCPSO approach for FS [16].
3) Chaotic binary PSO FS (CBPSO) [17].
4) PSO (4–2) [3].
5) Differential evolution (DE) [8].
6) Multitasking PSO (MTPSO) [13].

Table 1. Datasets.

Dataset	features	Instance	Classes	%Smallest	%Largest
PCMAC	3289	1943	2	49	51
Lung Cancer	3312	203	5	3	68
BASEHOCK	4862	1993	2	50	50
RELATHE	4322	1247	2	45	55
Brain Tumor	5920	90	5	4	67
Prostate Tumor	5966	102	2	49	51
Leuk2	7129	72	4	6	53
GLI85	22283	85	2	31	69
9Tumor	5726	60	9	3	15
11Tumor	12533	174	11	4	16

4.3 Result

Classification accuracy and the number of selected features are the most important indicators for evaluating the effectiveness of FS methods. In this experiment, a higher classification accuracy and a lower number of selected features indicate that the algorithm has an advantage in addressing FS problems.

Table 2 shows the average (AvgAcc) of the classification accuracies of GPAPSO and the other 6 compared algorithms on the 10 high-dimensional classification datasets. As can be seen from Table 3, the average classification accuracies of the proposed GPAPSO method are better than those of the compared methods. This illustrates that GPAPSO can achieve higher classification accuracy compared to other methods.

Table 3 compares GPAPSO with the other nine algorithms in terms of the number of selected features. As can be seen from Table 3, VLPSO selects fewer features than GPAPSO on 7 out of the 10 datasets. The main reason is that VLPSO uses a length-changing mechanism at the cost of precision to reduce the size of features during the

Table 2. Average of the classification accuracy.

Dataset	VL-PSO	VS-CCPSO	CBPSO	PSO (4-2)	DE	MTPSO	GPAPSO
PCMAC	83.21	79.86	76.44	76.64	76.21	85.46	**88.68**
Lung Cancer	89.62	95.89	92.32	92.03	95.33	96.07	**98.53**
BASEHOCK	91.31	89.31	83.72	86.17	82.07	93.69	**94.73**
RELATHE	69.76	83.42	81.07	78.44	74.77	80.47	**86.01**
Brain Tumor	65.85	85.57	84.92	82.34	84.64	87.22	**88.64**
Prostate Tumor	82.43	91.20	86.33	87.82	77.42	93.71	**95.23**
Leuk2	81.21	85.08	86.70	83.98	74.16	90.42	**93.33**
GLI85	79.81	86.14	79.11	81.07	80.71	89.68	**94.11**
9Tumor	59.12	65.03	48.71	50.33	48.92	65.12	**74.99**
11Tumor	77.67	87.71	81.82	79.90	78.16	90.37	**91.99**

Table 3. Number of features.

Dataset	VL-PSO	VS-CCPSO	CBPSO	PSO (4-2)	DE	MTPSO	GPAPSO
PCMAC	**65.9**	762.0	1524.1	498.1	143.1	117.5	112.9
Lung Cancer	367.1	566.6	1116.7	460.6	1405.2	373.3	**350.1**
BASEHOCK	**172.0**	1404.9	2511.0	1437.1	2107.1	298.8	382.8
RELATHE	**122.3**	950.6	1852.2	874.2	1598.0	198.4	337.2
Brain Tumor	**34.8**	1289.0	2800.1	758.2	2473.1	547.7	198.8
Prostate Tumor	50.2	752.1	2403.9	801.6	2158.9	401.7	**38.0**
Leuk2	**67.1**	1193.7	3003.8	997.2	3103.0	420.4	247.0
GLI85	597.2	4396.0	9120.7	3244.2	8267.5	1067.9	**333.7**
9Tumor	**66.8**	875.2	2522.5	878.1	2483.4	409.1	224.2
11Tumor	**301.2**	1879.8	5429.4	1769.3	4763.2	1321.2	463.0

evolutionary process. Nonetheless, GPAPSO still selected the fewest features on two datasets and chose the second fewest features compared to the other benchmark methods. Moreover, it is evident that GPAPSO achieved higher accuracy.

We can observe that in the problem of FS in high-dimensional classification, GPAPSO performs exceptionally well in the two most important metrics: accuracy and the number of selected features. Table 4 records the average training time consumed during the experiments. From Table 4, it can be observed that on three out of ten datasets,

Table 4. Training time.

Dataset	VL-PSO	VS-CCPSO	CBPSO	PSO (4-2)	DE	MTPSO	GPAPSO
PCMAC	16292.11	29137.12	23124.67	11070.72	29730.05	**4424.78**	6510.37
Lung Cancer	1783.66	380.18	460.93	**231.01**	738.37	344.14	1987.77
BASEHOCK	28567.44	40747.85	37343.45	23829.99	24785.87	8403.21	**8301.73**
RELATHE	10063.93	22186.60	17477.32	8756.52	31647.24	**3317.25**	6596.48
Brain Tumor	1312.27	243.42	302.67	170.29	532.92	156.63	**118.39**
Prostate Tumor	446.31	**182.28**	392.66	216.20	679.63	186.19	1957.26
Leuk2	643.13	**126.32**	263.27	147.50	413.13	128.22	918.94
GLI85	3410.78	693.80	1192.18	**611.21**	1952.51	307.93	2996.89
9Tumor	383.40	88.80	147.48	89.55	278.09	**71.21**	974.99
11Tumor	4277.87	1350.91	2563.47	1331.37	3773.15	790.54	**1207.32**

GPAPSO had the shortest training time. Moreover, no single algorithm exhibits consistently excellent performance in this regard. GPAPSO's performance is the best. This is because GPAPSO consumes time during initialization due to the use of multiple methods.

Table 5. Ablation Experiment.

Dataset	Method	Time(s)	Size	AvgAcc	
Lung Cancer	GPAPSO^{m-}	1726.79	590.8	95.12	
	GPAPSO^{k-}	1018.03	966.7	97.07	(+)
	GPAPSO	1987.77	350.1	**98.53**	(+)
9Tumor	GPAPSO^{m-}	658.46	410.3	69.39	
	GPAPSO^{k-}	514.28	626.5	64.52	(+)
	GPAPSO	1174.09	224.2	**74.99**	(+)
11Tumor	GPAPSO^{m-}	1127.55	1056.7	88.57	
	GPAPSO^{k-}	690.73	1479.6	85.71	(+)
	GPAPSO	1207.32	463	**91.99**	(+)
Prostate Tumor	GPAPSO^{m-}	1626.84	198.4	93.28	
	GPAPSO^{k-}	851.83	286.2	90.47	(+)
	GPAPSO	2157.26	38	**95.23**	(+)

(*continued*)

Table 5. (*continued*)

Dataset	Method	Time(s)	Size	AvgAcc	
Leuk2	GPAPSO^{m-}	838.64	385.4	91.21	
	GPAPSO^{k-}	441.23	582	86.67	(+)
	GPAPSO	1218.94	247	**93.33**	(+)
Brain Tumor	GPAPSO^{m-}	566.45	312.6	83.83	
	GPAPSO^{k-}	230.99	424.1	80.36	(+)
	GPAPSO	1108.39	198.8	**88.64**	(+)
GLI85	GPAPSO^{m-}	1720.83	909.2	88.94	
	GPAPSO^{k-}	990.6	1398.9	82.35	(+)
	GPAPSO	3296.89	333.7	**94.11**	(+)
PCMAC	GPAPSO^{m-}	7002.57	265.2	85.29	
	GPAPSO^{k-}	4557.78	372.8	82.27	(+)
	GPAPSO	9510.37	132.9	**88.68**	(+)
RELATHE	GPAPSO^{m-}	5198.42	740.9	83.91	
	GPAPSO^{k-}	3081.27	848.7	82.16	(+)
	GPAPSO	8596.48	582.8	**86.01**	(+)
BASEHOCK	GPAPSO^{m-}	10063.86	548.7	93.98	
	GPAPSO^{k-}	5362.69	895.6	87.96	($\approx$)
	GPAPSO	14301.73	180.7	**94.73**	(+)

4.4 Ablation Experiment

To investigate the effects of the proposed multi-method population initialization strategy and the adaptive factor-based knowledge transfer strategy throughout the entire optimization process, we conducted ablation experiments by comparing GPAPSO with versions where these two strategies were removed. The first version doesn't utilize multi-method population initialization at the start of optimization, denoted as GPAPSO^{m-}. The second version doesn't employ the adaptive factor-based knowledge transfer during the evolutionary process, denoted as GPAPSO^{k-}.

Table 5 presents the training time, classification accuracy, and the number of selected features obtained by training with these three methods. It can be observed from the table that the use of the proposed guided particle adaptive method increased training time on most datasets. However, it resulted in achieving better average classification accuracy with a smaller subset of features. Figure 4 illustrates the fitness of three methods throughout the iterative process of training. It is evident from the graph that guided particle adaptation effectively avoids falling into local optima, thereby achieving higher fitness.

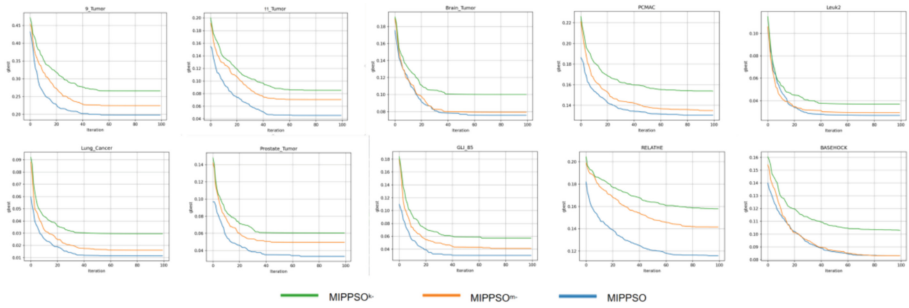

Fig. 4. Average fitness in 100 iterations.

5 Conclusion

This paper presents a novel particle initialization method aimed at enhancing the guidance of initial particles and expanding the meaningful search range. To achieve this goal, a collective filter-based initialization approach is proposed, where initial particles in the population are generated by various methods, and particles within the population employ an adaptive factor for knowledge transfer during evolution. Results demonstrate that, compared to traditional single initialization methods and simple learning from other populations' best particles, the proposed method achieves better classification performance with smaller feature subsets.

The proposed method holds promising applications in feature selection, with broader potential across various tasks. Updating at the feature dimension level in the context of adaptive learning may offer improved performance, but implementation poses challenges due to computational costs. However, this direction will be considered in future work.

References

1. Obermeyer, Z., Emanuel, E.J.: Predicting the future-big data, machine learning, and clinical medicine. N. Engl. J. Med. **375**(13), 1216 (2016)
2. Zhou, L., Pan, S., Wang, J.: Machine learning on big data: opportunities and challenges. Neurocomputing **237**, 350–361 (2017)
3. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms. Appl. Soft Comput. **18**, 261–276 (2014)
4. Mwangi, B., Tian, T.S., Soares, J.C.: A review of feature reduction techniques in neuroimaging. Neuro Inform. **12**(2), 229–244 (2014)
5. Zhou, Y., Zhang, W., Kang, J., Zhang, X., Wang, X.: A problem specific nondominated sorting genetic algorithm for supervised feature selection. Inf. Sci. **547**, 841–859 (2021)
6. Chen, K., Zhou, F., Xue, B.: Particle swarm optimization for feature selection with adaptive mechanism and new updating strategy. In: Proceedings of the Australasian Joint Conference Artificial Intelligence, pp. 419–431 (2018)
7. Tabakhi, S., Moradi, P., Akhlaghian, F.: An unsupervised feature selection algorithm based on ant colony optimization. Eng. Appl. Artif. Intell. **32**, 112–123 (2014)

8. Khushaba, R.N., Al-Ani, A., Al-Jumaily, A.: Differential evolution based feature subset selection. In: Proceedings of the International Conference on Pattern Recognition, pp. 1–4 (2008)
9. Deniz, A., Kiziloz, H.E., Dokeroglu, T.: Robust multi-objective evolutionary feature subset selection algorithm for binary classification using machine learning techniques. Neurocomputing **241**, 128–146 (2017)
10. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. Proc. IEEE Int. Conf. Syst. Man Cybern. **5**, 4104–418 (1997)
11. Das, I.: On characterizing the 'knee' of the pareto curve based on normal-boundary intersection. Struct. Optim. **18**, 107–115 (1999)
12. Jiang, M., Wang, Z., Hong, H., Yen, G.G.: Knee point based imbalanced transfer learning for dynamic multi-objective optimization. IEEE Trans. Evol. Comput. **25**(1), 117–129 (2021)
13. Chen, K., Xue, B., Zhang, M., Zhou, F.: An evolutionary multitasking based feature selection method for high-dimensional classification. IEEE Trans. Cybern. **52**(7), 7172–7186 (2022)
14. Chen, K., Xue, B., Zhang, M., Zhou, F.: Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimisation. IEEE Trans. Evol. Comput. **26**(3), 446–460 (2022)
15. Tran, B., Xue, B., Zhang, M.: Variable-length particle swarm optimization for feature selection on high-dimensional classification. IEEE Trans. Evol. Comput. **23**(3), 473–487 (2019)
16. Song, X., Zhang, Y., Guo, Y.N.: Variablesize cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. IEEE Trans. Evol. Comput. **24**(5), 882–895 (2020)
17. Chuang, L.Y., Yang, C.H., Li, J.C.: Chaotic maps based on binary particle swarm optimization for feature selection. Appl. Soft Comput. **11**(1), 239–248 (2011)