Supplementary Materials of "Evolutionary Multitasking AUC Optimization"

Chao Wang, Kai Wu, *Member, IEEE*, and Jing Liu, *Senior Member, IEEE*

A Case Study on Automatic Hyperparameter Tuning

Hyperparameter tuning (HPT) aims to automatically select the optimal hyperparameter configuration for machine learning algorithms [1]. However, evaluating each hyperparameter configuration is often expensive, which requires multiple iterations of the corresponding machine learning algorithm. Inspired by this, the performance of our proposal on the HPT problem is preliminarily analyzed in this section.

Let $f \in \mathcal{F}$ be the machine learning model, where $\mathcal{F}$ is the model space. Let $x_f \in \mathcal{X}_f$ be a hyperparameter configuration of *f*, where $\mathcal{X}_f$ is the hyperparameter space. Combined with *n*-fold cross-validation, a common evaluation criterion for hyperparameter configuration can be expressed as follows:

$$g\left(f, x_f\right) = \frac{1}{n}\sum_{i=1}^{n} Loss\left(f, x_f, S_i^T, S_i^V\right) \tag{S1}$$

where $Loss(\cdot)$ is the loss function of the machine learning model, and S^T_i and S^V_i are the training and validation dataset in the *n*-th fold, respectively. The goal of hyperparameter tuning in this paper is to find the solution that satisfies:

$$x_f^* = \arg\min_{x_f \in \mathcal{X}_f} g\left(f, x_f\right) \tag{S2}$$

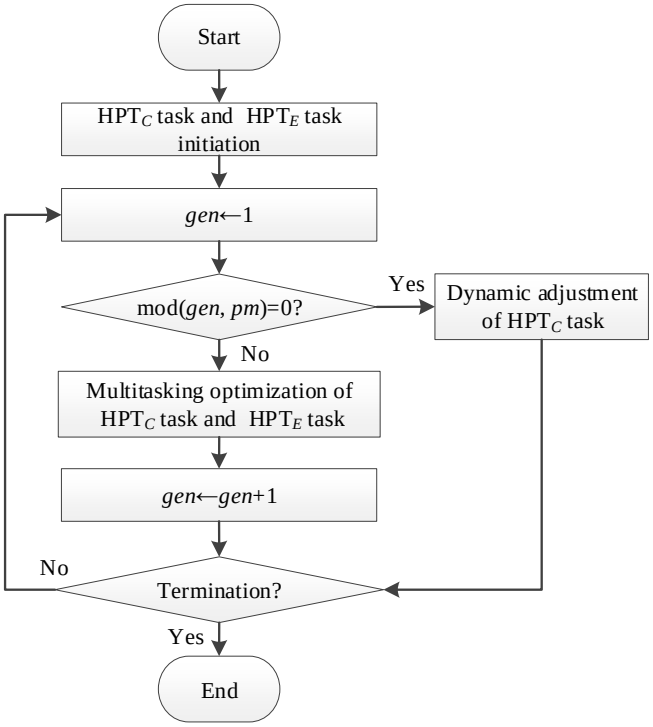

Fig.S1. The framework of EMTHPT.

In this section, our proposal is employed to solve the above HPT problem, termed as EMTHPT, as shown in Fig. S1. Firstly, a multitasking HPT optimization environment is constructed, which includes an original expensive HPT task (HPT$_E$) and a cheap HPT task (HPT$_C$) with a sampled small-scale dataset. Therefore, the objective function of multi-task HPT optimization can be defined as follows:

$$\begin{cases} \min\limits_{x_f} g\left(f, x_f\right) = \dfrac{1}{n}\sum\limits_{i=1}^{n} Loss\left(f, x_f, S_i^T, S_i^V\right) \\ \min\limits_{x_f} g\left(f, x_f\right) = \dfrac{1}{n}\sum\limits_{i=1}^{n} Loss\left(f, x_f, S_i^{T'}, S_i^{V'}\right) \end{cases} \tag{S3}$$

where $S^{T'}_i$ and $S^{V'}_i$ are two simple datasets sampled from S^T_i and S^V_i with a sampling rate *s* randomly, respectively. Then the above problem can be optimized by the general multitasking optimization algorithm to make full use of common knowledge across tasks. Next, the sampling dataset of the HPT$_C$ task is dynamically adjusted after a certain number of generations (*pm* generations). Since it is impossible to obtain the score of each data from the HPT problem like the AUC optimization problem, the random adjustment strategy (RAS) is adopted to adjust the datasets in this section dynamically. In RAS, $s\times|S^T_i|$ training instances and $s\times|S^V_i|$ validation instances are randomly selected from S^T_i and S^V_i to form new $S^{T'}_i$ and $S^{V'}_i$, where $|*|$ is the number of *. Finally, the optimal hyperparameter configuration is output when the algorithm meets the termination condition.

LightGBM is one of the most popular gradient boosting decision trees widely used in various machine learning tasks [1]. EMTHPT is applied to solve the LightGBM's HPT problem on 11 real machine learning application datasets, including Iris, Digits, Diabetes, Breast, Musk, HTRU2, Magic04, Adult, Sensorless, Connect4, and Higgs from the LIBSVM website [3]. The details of datasets and the setting of sampling rate *s* are shown in Table S1, where *d* is the number of features. When the number of instances in the dataset is less than 30,000, the sampling rate of the HPT$_C$ task is set to 0.1; otherwise, it is set to 0.05. The seven hyperparameters in LightGBM are considered, including learning rate (*LR*), number of leaves (*NL*), minimal number of data in one leaf (*MNDL*), bagging fraction (*BF*), frequency of bagging (*FB*), feature fraction (*FF*), and λ_{l1} and its details are shown in Table S2.

TABLE SI
DATASET STATISTICS.

#ID	Dataset	S^T	S^V	d	s
1:	Iris	135	15	4	0.1
2:	Diabetes	398	44	10	0.1
3:	Breast	512	57	30	0.1
4:	Digits	1,617	179	64	0.1
5:	Musk	4,991	2,083	168	0.1
6:	HTRU2	14,318	3,580	8	0.1
7:	Magic04	15,215	3,805	10	0.1
8:	Adult	32,561	16,281	14	0.05
9:	Sensorless	40,883	17,525	48	0.05
10:	Connect4	47,504	20,053	42	0.05
11:	Higgs	10,000,000	1,000,000	28	0.05

TABLE SII
THE INFORMATION OF HYPERPARAMETERS IN LIGHTGBM

#ID	Hyperparameters	Search Space	Variable Type
1:	*LR*	(0, 1]	Continuous
2:	*NL*	[20, 200]	Discrete
3:	*MNDL*	[10, 150]	Discrete
4:	*BF*	[0, 1]	Discrete
5:	*FB*	[0.5, 1]	Continuous
6:	*FF*	[0.5, 1]	Continuous
7:	λ_{l1}	[0, 10]	Continuous

In the experiment, two representative methods, SBGA and EMEA, are considered the multitasking optimizer and embedded in EMTHPT to obtain EMTHPT-SBGA and EMTHPT-EMEA. Since there are two types of variables in LightGBM, a mixed encoding form of the real and integer number is used. SBX crossover and PM mutation are employed in our proposal. All parameter settings are the same as the experimental part (see section VI) in this paper. Table SIII shows the mean square error of each method on all datasets over 20 runs. Overall, the average performance of EMTHPT-SBGA (or EMTHPT-EMEA) is better than that of Single-task GA in most cases. This phenomenon appears because useful knowledge is transferred in our constructed multitasking optimization environment. In addition, Table SIV shows the mean square error of the EMTHPT with and without the RAS (EMTHPT and EMTHPT') on all datasets over 20 runs. It can be observed that EMTHPT-SBGA (EMTHPT-EMEA) wins 9 (10) and ties 2 (1) on 11 datasets in terms of the mean square error, which further illustrates the effectiveness of the RAS.

TABLE SIII
THE COMPARISON OF EMTHPT-SBGA (OR EMTHPT-EMEA) AGAINST SINGLE TASK GA IN TERMS OF THE MEAN SQUARE ERROR.

#ID	EMTHPT-SBGA	EMTHPT-EMEA	Single-task GA
1	**2.15E−01(5.94E−02)**+	**2.02E−01(1.48E−02)**+	2.43E−01(8.59E−02)
2	**5.29E+01(3.94E−01)**+	**5.37E+01(4.02E−01)**+	6.11E+01(3.32E−01)
3	**2.00E−01(4.57E−02)**+	**1.66E−01(1.76E−02)**+	2.10E−01(3.56E−02)
4	**9.34E−01(6.08E−02)**+	**9.44E−01(7.79E−02)**+	1.13E+00(2.68E−01)
5	**1.05E−01(2.04E−02)**+	**1.06E−01(1.00E−02)**+	1.54E−01(2.02E−02)
6	**1.24E−01(4.91E−03)**+	**1.24E−01(2.52E−03)**+	1.33E−01(1.56E−03)
7	**2.99E−01(3.32E−03)**+	**3.01E−01(2.14E−03)**+	3.11E−01(1.25E−03)
8	**3.06E−01(8.15E−03)**+	**3.04E−01(7.47E−03)**+	3.15E−01(6.48E−03)
9	**6.94E−02(2.27E−03)**+	**6.87E−02(1.15E−02)**+	8.24E−02(1.12E−02)
10	**2.75E−01(2.26E−04)**+	**2.74E−01(1.30E−03)**+	2.98E−01(3.63E−03)
11	**4.11E−01(2.23E−04)**≈	**4.11E−01(2.38E−04)**≈	4.12E−01(2.10E−04)
w/t/l	10/1/0	10/1/0	–

TABLE SIV
THE COMPARISON OF EMTHPT WITH AND WITHOUT THE RAS (EMTHPT AND EMTHPT') IN TERMS OF THE MEAN SQUARE ERROR.

#ID	EMTHPT-SBGA	EMTHPT -SBGA	EMTHPT-EMEA	EMTHPT -EMEA
1	**2.15E−01(5.94E−02)**+	2.36E−01(1.35E−01)	**2.02E−01(1.48E−02)**+	2.22E−01(1.67E−01)
2	**5.29E+01(3.94E−01)**+	6.05E+01(3.54E−01)	**5.37E+01(4.02E−01)**+	5.94E+01(3.97E−01)
3	**2.00E−01(4.57E−02)**+	2.11E−01(5.89E−02)	**1.66E−01(1.76E−02)**+	1.94E−01(4.86E−02)
4	**9.34E−01(6.08E−02)**+	1.02E+00(1.48E−01)	**9.44E−01(7.79E−02)**+	1.03E+00(8.77E−02)
5	**1.05E−01(2.04E−02)**+	1.10E−01(2.15E−02)	**1.06E−01(1.00E−02)**+	1.14E−01(2.44E−03)
6	**1.24E−01(4.91E−03)**+	1.32E−01(1.16E−03)	**1.24E−01(2.52E−03)**+	1.36E−01(5.62E−03)
7	**2.99E−01(3.32E−03)**+	3.05E−01(3.40E−03)	**3.01E−01(2.14E−03)**+	3.07E−01(3.46E−03)
8	**3.06E−01(8.15E−03)**≈	3.09E−01(4.51E−03)	**3.04E−01(7.47E−03)**+	3.13E−01(5.61E−03)
9	**6.94E−02(2.27E−03)**+	7.12E−02(6.70E−03)	**6.87E−02(1.15E−02)**+	7.68E−02(9.23E−03)
10	**2.75E−01(2.26E−04)**+	2.89E−01(1.71E−03)	**2.74E−01(1.30E−03)**+	2.89E−01(2.68E−04)
11	**4.11E−01(2.23E−04)**≈	4.12E−01(2.10E−04)	**4.11E−01(2.38E−04)**≈	4.12E−01(2.10E−04)
w/t/l	9/2/0	–	10/1/0	–

B EXPERIMENTAL RESULTS IN SECTION VI

Firstly, the average convergence trends of four compared methods, i.e., EMTAUC-MFEAII, EMTAUC-SBGA, EMTAUC-EMEA, and Single-task GA, on different datasets are shown in Fig. S2. In these figures, the x-axis represents the computational cost, and the y-axis represents the average objective value on a log scale.

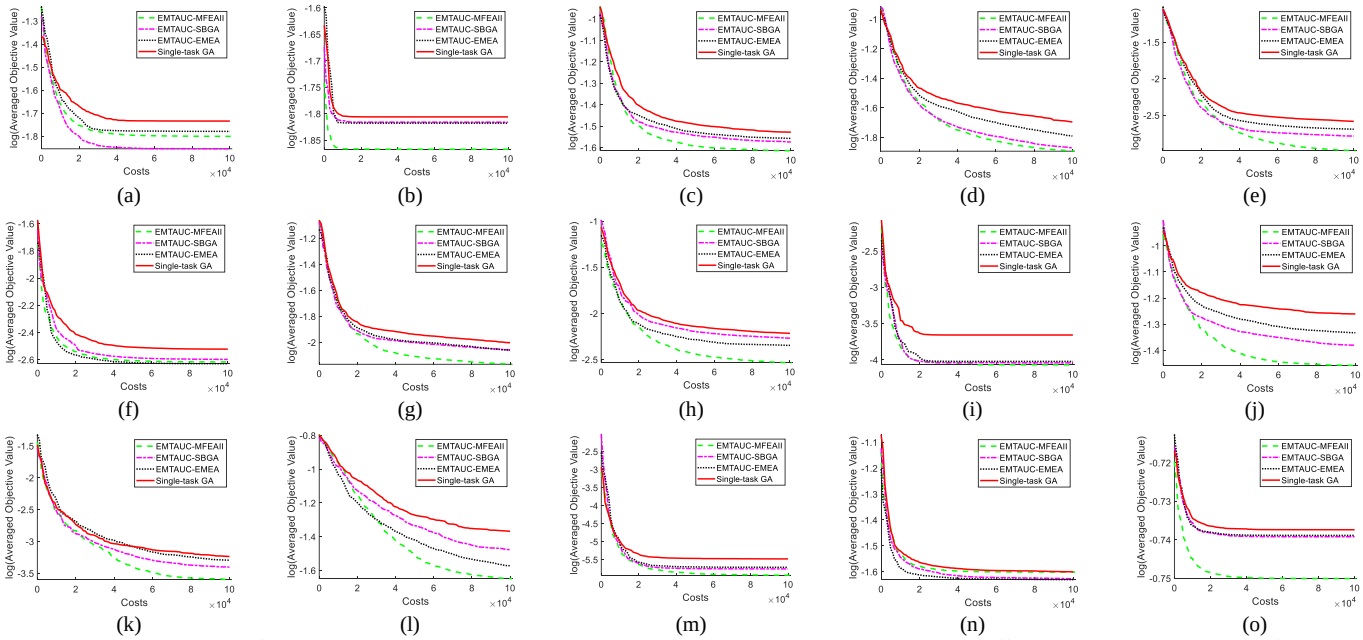

Fig. S2. The convergence curve of EMTAUC-MFEAII, EMTAUC-SBGA, EMTAUC-EMEA, and Single-task GA on different datasets, (a) diabetes, (b) fourclass, (c) german, (d) splice, (e) usps, (f) Australian, (g) a9a, (h) sonar, (i) svmguide1, (j) svmguide3, (k) segment, (l) ijcnn1, (m) satimage, (n) vowel, and (o) poker.

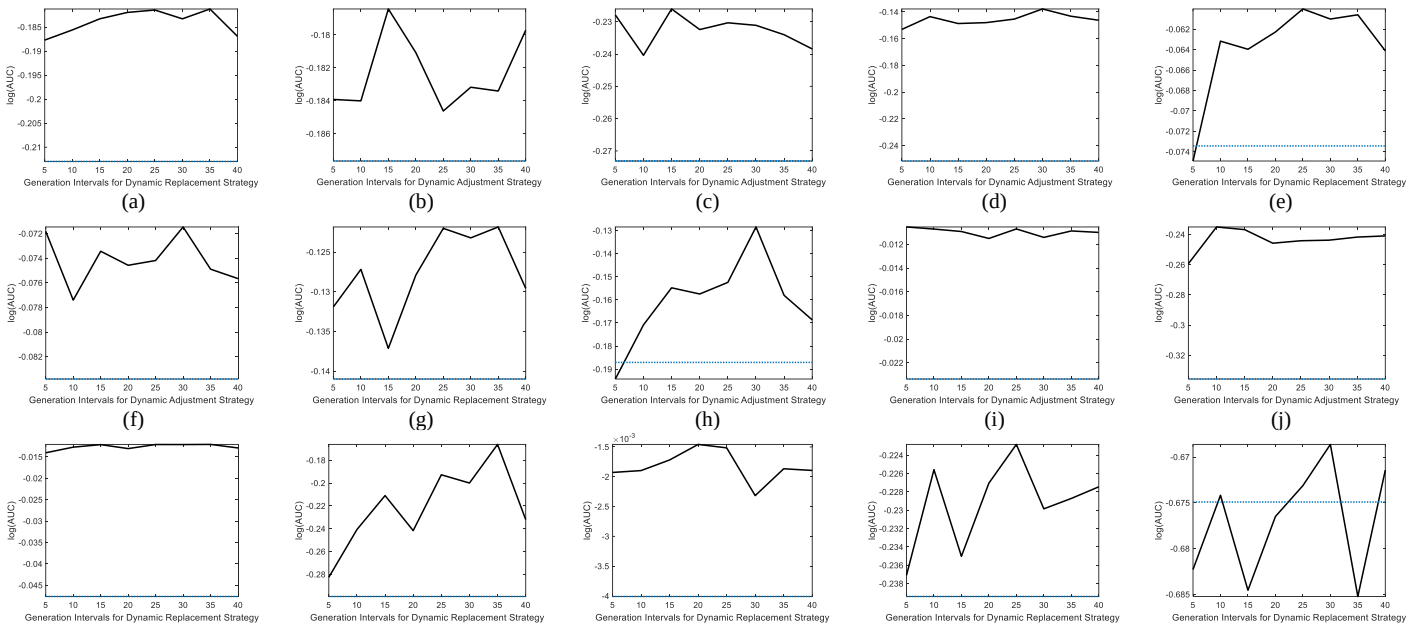

Fig. S3. The relationships between the generation intervals for dynamic adjusting strategy and the average AUC value obtained by EMTAUC-SBGA on different datasets, where the dotted line represents the average AUC value obtained by Single-task GA, (a) diabetes, (b) fourclass, (c) german, (d) splice, (e) usps, (f) australian, (g) a9a, (h) sonar, (i) svmguide1, (j) svmguide3, (k) segment, (l) ijcnn1, (m) satimage, (n) vowel, and (o) poker.

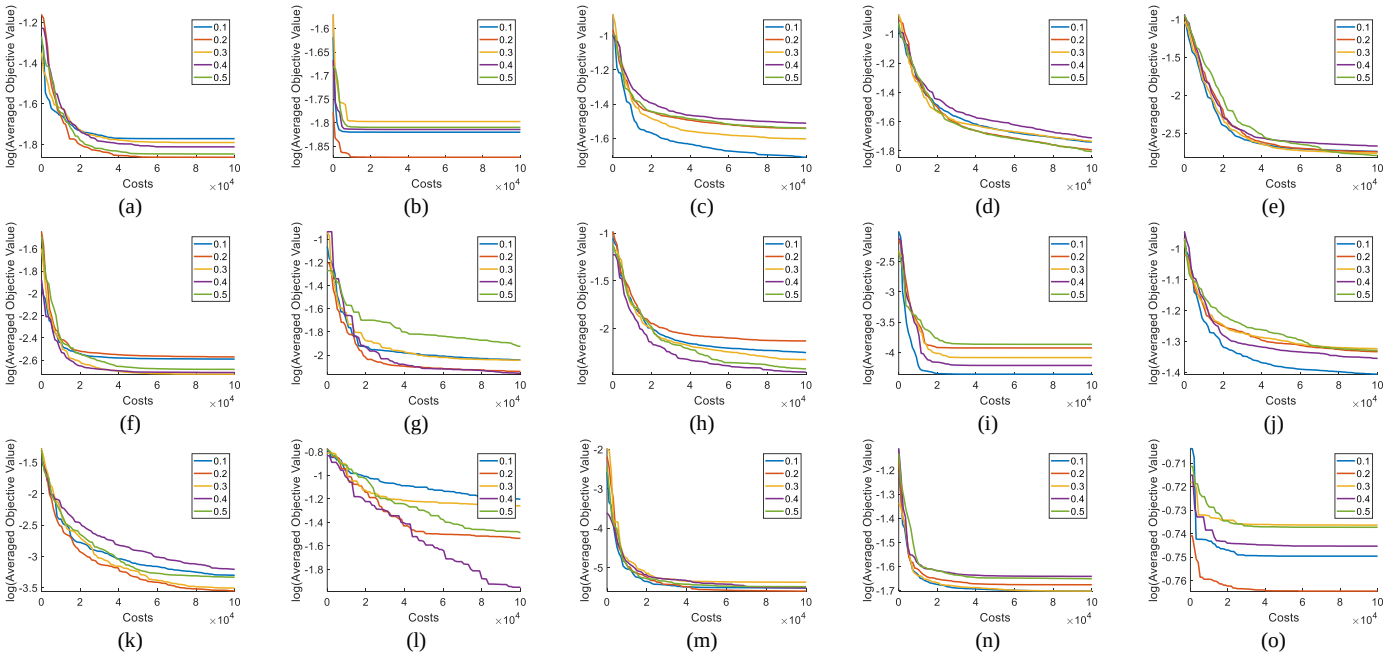

Fig. S4. The performance of EMTAUC with varying ratio of the whole dataset for the cheap task, (a) diabetes, (b) fourclass, (c) german, (d) splice, (e) usps, (f) australian, (g) a9a, (h) sonar, (i) svmguide1, (j) svmguide3, (k) segment, (l) ijcnn1, (m) satimage, (n) vowel, and (o) poker.

Secondly, Fig. S3 shows the average AUC value of EMTAUC versus varying key parameter *pm*. The value of *pm* is set from 5 to 40 in steps of 5.

Thirdly, Fig. S4 shows the average objective value of EMTAUC versus varying key parameter *s*. In these figures, the x-axis represents the computational cost, and the y-axis represents the average objective value on a log scale.

Finally, Fig. S5 shows the average AUC value of EMTAUC versus varying key parameter λ. The value of λ is set to 2^i, i=−5, −4, −3, …, 1.

REFERENCES

[1] Y.-Q. Hu, Y. Yu, W.-W. Tu, Q. Yang, Y. Chen, and W. Dai, "Multi-fidelity automatic hyperparameter tuning via transfer series expansion," In *AAAI*, vol. 33, pp. 3846–3853, 2019.

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp.1871–1874, 2008.

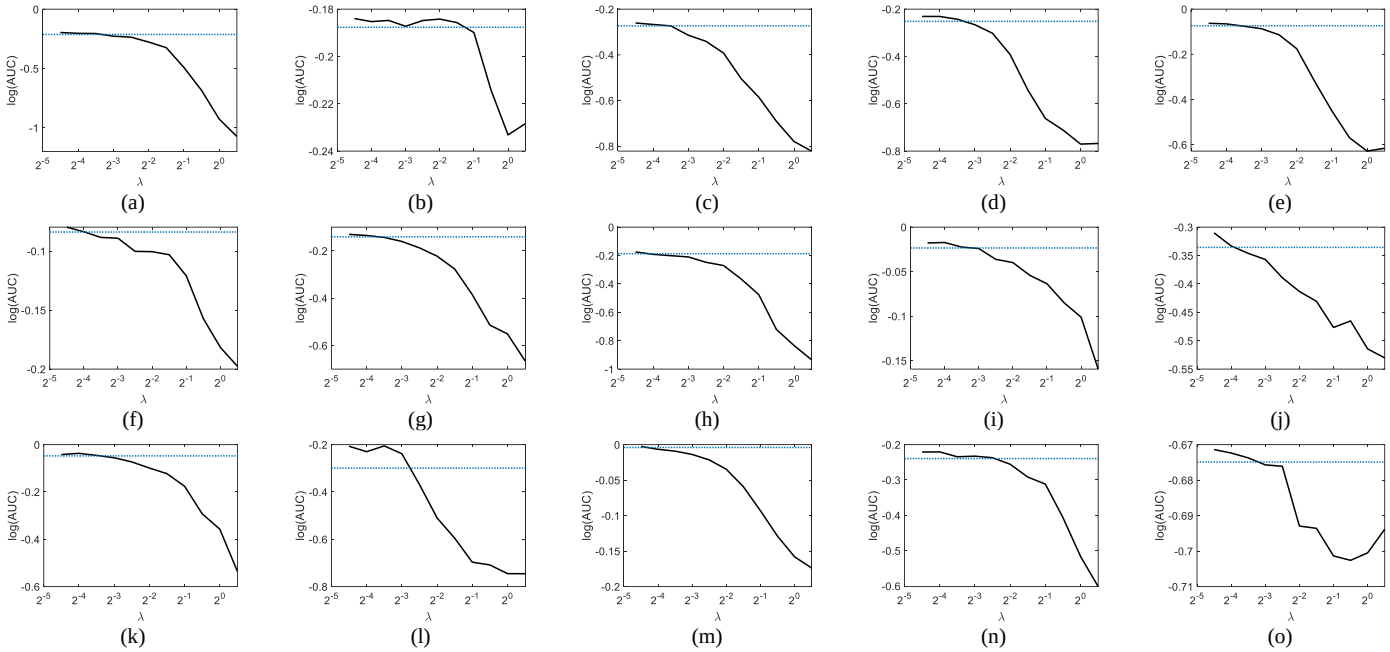

Fig. S5. The relationships between the penalty parameter and the average AUC value obtained by EMTAUC-SBGA on different datasets, where the dotted line represents the average AUC value obtained by Single-task GA, (a) diabetes, (b) fourclass, (c) german, (d) splice, (e) usps, (f) australian, (g) a9a, (h) sonar, (i) svmguide1, (j) svmguide3, (k) segment, (l) ijcnn1, (m) satimage, (n) vowel, and (o) poker.