Phi modeling for Quantum Gaussian Mixture Models

*Sangyeon Kim

Abstract

In the original paper [1], phi (ϕ) has impact in the mixed case as in the form of $\cos(\phi)$. Quantum Gaussian Mixture Models (QGMM) that is presented in the original paper uses the equation $\cos(\phi)$ derived from the constraint on page 4.

However, in the researches of the validity of the objective function and lambda impact, the value of $\cos(\phi)$ was too large, therefore, we decided to change phi to a trainable variable when optimizing the objective function, so $\cos(\phi)$ should be between -1 and 1.

1. Aim

The goal of the experiments is to find some possible improvements on the training side and increase the stability of the training process.

The initialization of the parameter phi would be important for the training performance because it impacts on the mixed case. So, we'll figure out what value it would be good as an initial phi.

With the value of phi that we figured out in the previous experiment, we'll experiment the training performance as lambda changed. First of all, we'll find the value of lambda that showed good results, and with lower and higher lambda than it, we'll check the training performance so we can see what lambda it would be great to use.

2. Dataset

In this research, we use 'Old faithful' data set [2]. Because it has 2 dimensionalities we can see the training process of QGMM in 2D graphs.

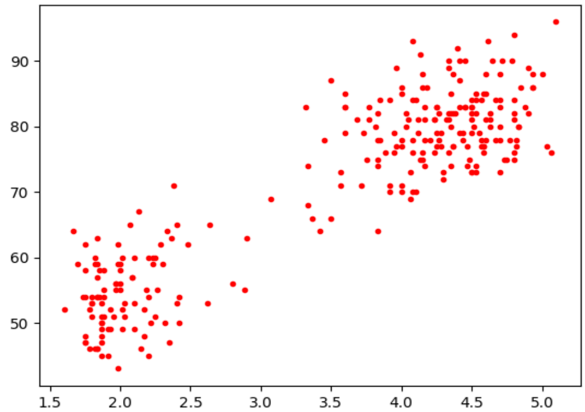


Figure 1. Old faithful data set. It has 2 dimensionalities. There are 2 bunches of observations on the bottom-left and top-right.

3. Experiments

we performed two experiments to find the possible improvements and good initial values in the training process for QGMM in 2-classes constraint environment.

In these experiments, the maximum iteration of training process is 50,000 and the tolerance is 1e-3. we used Adam optimizer with 0.001 learning rate (η). Also, as the training parameters, the initial alphas (α_k) are $\alpha_1 = 0.5$, $\alpha_2 = 0.5$, the means are randomly generated or selected from the previous research to see if the trainable phi works properly in hard cases, and the covariances (C_k) are that

$$C_1 = C_2 = \begin{pmatrix} 0.08 & 0.1 \\ 0.1 & 3.3 \end{pmatrix}$$

3.1 Initialization

To find a better initial value of phi, we trained QGMM with three different initial values of it, 0, 90 and 180, and the value of lambda was 1,500 selected as it is enough for the objective function to be constrained.

*This research was performed during Google Summer of Code 2019 for mlpack mentored by Sumedh Ghaisas.

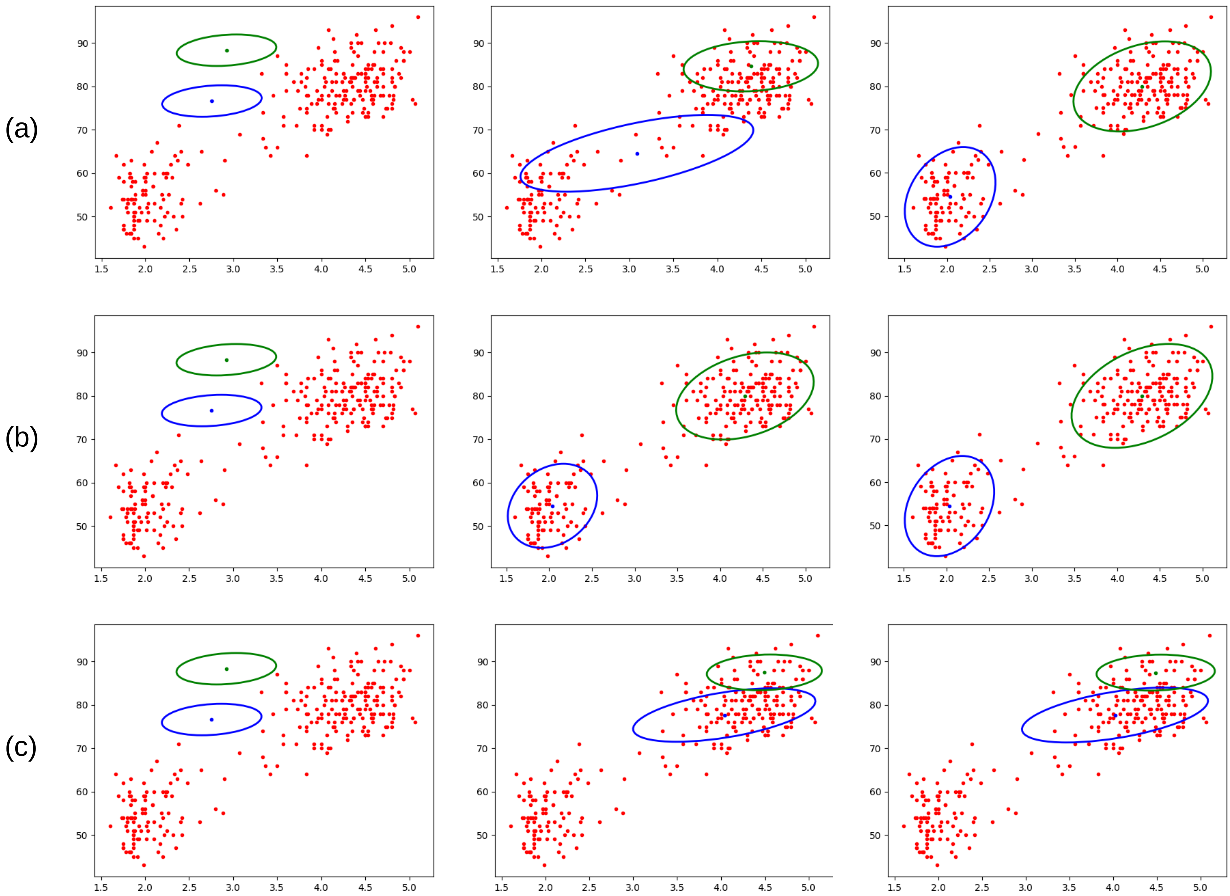


Figure 2. Training process of test case 1 with the separate phis, 0, 90 and 180. The means are [2.756031811312966, 76.62447648112042], [2.9226572802266397, 88.3509418943818]. (a) phi 0. (b) phi 90. (c) phi 180.

The means μ_1, μ_2 were selected from the hard cases in the validity of the objective function research. The values of means can be seen at *Figure 2* and Appendix A

From some experiments, we figured out that Initializing phi to 180 causes bad results. Besides, the results when phi 0 and 90 were the same, but in case of phi 90, the clusters found the center of the observations faster than phi 0.

The mainly different point between phi 0 and 90 is that when we set phi to 0, phi was fixed to 0, not changed. On the other hand, in the phi 90, the value of phi has been changed little by little while the optimizer minimized the objective function. Therefore, that flexibility in an initial phi 90 seems to have had the effect on the training process positively.

3.2 Lambda selection

The value of lambda has an effect on the importance of the approximation constraint in the objective function. Thus, QGMM's training process may or may not be constrained, according to the value of lambda.

Therefore, in this experiment, we'll figure out what value it would be good as an initial lambda. Besides, we'll use phi 90 as an initial value in this experiment, as we know it is a good initial value from the precious research.

Like the initialization experiment, this test was performed with the means that didn't work properly in unconstrained cases that we figured out in the validity of the objective function research as well as randomly generated means, test cases 4~5.

In this experiment, we used three different values of lambda, 100, 1,500, and 5,000. Each lambda represents the degree of weak and strong constraint. Thus, we'll find out what value can restrict the objective function properly and how the training results vary depending on whether or not they are constrained. Also, we'll figure out what value is good as an initial lambda by finding the lower or upper bound through this experiment.

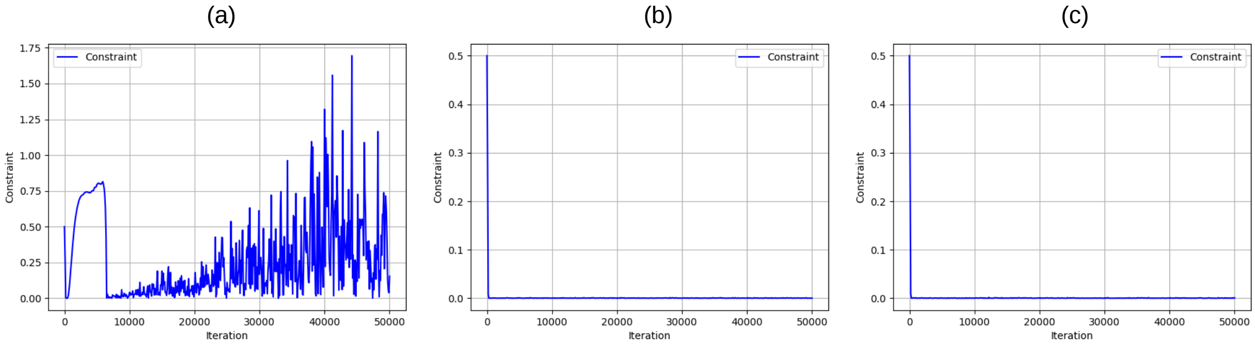


Figure 3. Constraint of test case 2 with several lambdas. (a) Lambda 100. (b) Lambda 1,500. (c) Lambda 5,000.

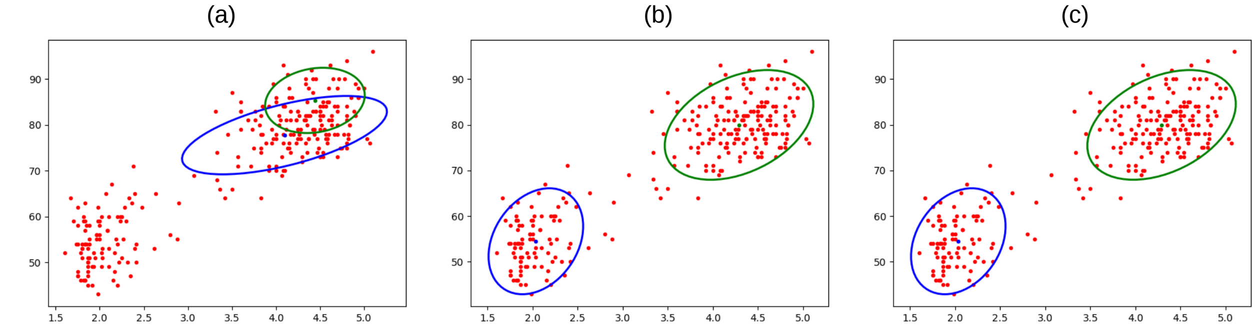


Figure 4. Training results of test case 2 with several lambdas. (a) Lambda 100. (b) Lambda 1,500. (c) Lambda 5,000.

All the training results can be seen in Appendix B.

From *Figure 3*, we can see that the optimization with lambda 100 didn't work properly on the constraint side. Also, using the weak constraint causes bad results because it is hard to converge and sometimes it diverges. In addition, we figured out that the rest of the test cases were not be trained well under the unconstrained optimization.

Therefore, if we get the value of lambda enough to make the optimization constrained, we can increase the stability of the training process.

After several experiments in this research, we found an interesting point. Because the value of cos(phi) was too high before changing phi to a trainable variable, the entire training process was greatly influenced by lambda.

On the other hand, after converting phi to be trainable, the value of cos(phi) was always between -1 and 1. As a result, the training process was more stable than before, less affected by lambda.

3.2.1 High lambda

we conducted an additional experiment with an additional lambda 20,000 to see if the higher value of lambda, the better on the view of the training performance.

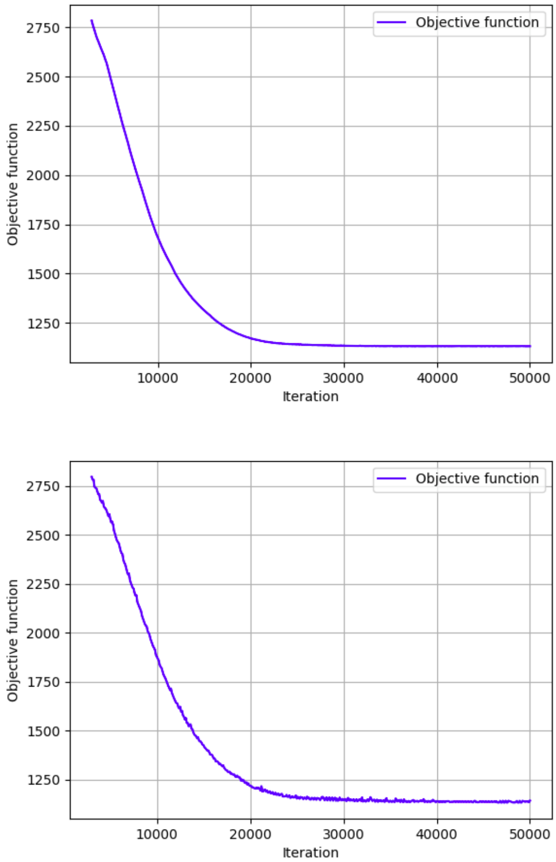


Figure 5. Objective function of test case 1 with lambda 1,500 (up) and 20,000 (down).

When optimizing the objective function with a high lambda, we observed the curve of the objective function trembled, although the training result was nice like when lambda 1,500 and 5,000. With the constraint that is not entirely zero when minimizing it, the constant of lambda, which is big, rather interferes with the convergence of the objective function. Therefore, we found out that it doesn't guarantee the higher lambda we get, the better performance it is.

3.3 Various distances

In the 3.1 Initialization experiment, we found some interesting point. Seeing the training process (c) in *Figure 2*, the two clusters have converged on a pile of observations. It is interesting because the initial value of phi in the training process (c) was 180 or -180 ($\phi_1=90$, $\phi_2=-90$). Therefore, we did guess they have converged on a pile of observations because of the property of the subtraction between two distributions according to the value of phi.

Thus, for further experiment, we'll see additional cases about *Figure 2* as we change the distance between the initial clusters little by little. From this experiment, we expect we'll figure out the effects phi has on the distance.

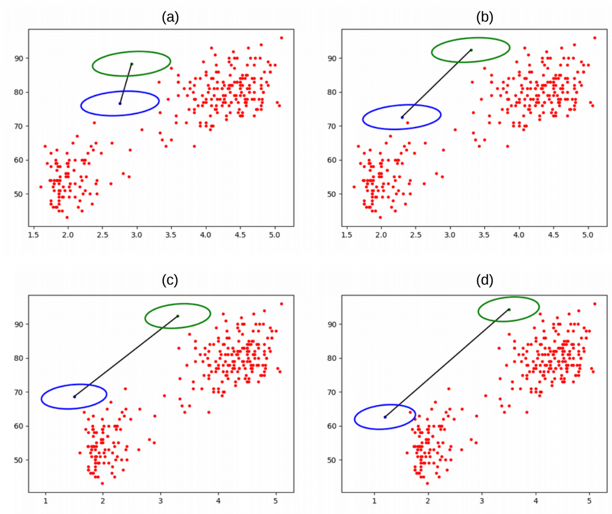


Figure 6. Initial distance. (a) Test case 1, distance is 11.78. (b) Test case 2, distance is 19.83. (c) Test case 3, distance is 23.87. (d) Test case 4, distance is 31.88

There are 4 cases that have a separate distance initially. With the initial states, we trained QGMM and the results of the training can be seen in Appendix C.

Seeing the training results, in the case of phi 180, the results of training were bad except the test case 4 (d). The reason the training in the test case 4 was successful is initially clusters were distributed close to each piles of the observations, seeing the result of the test case 3. However, we figured out an interesting finding.

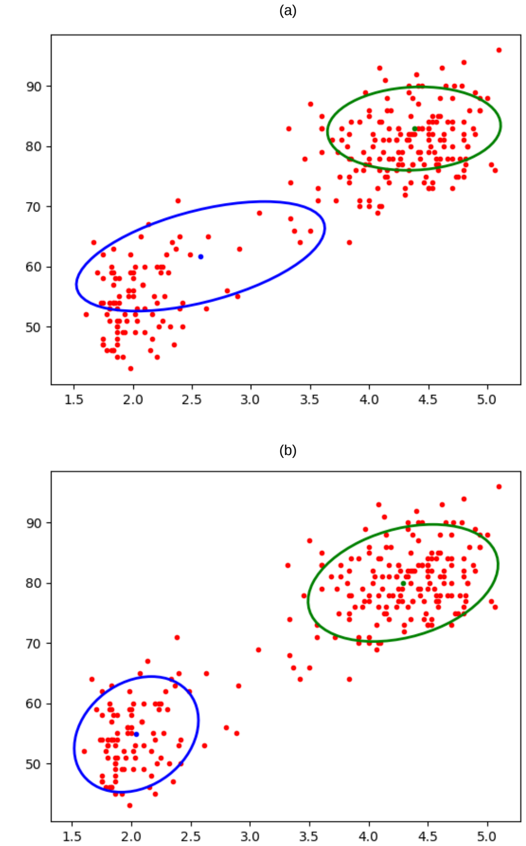


Figure 7. Middle distance in the test case 2. (a) Phi 0 (b) Phi 90

The more negative cos(phi) is, the closer the clusters to each other and the more positive cos(phi) is, the father the clusters to each other.

Also, the initial value of phi matters on the view of the training performance because according to the initial phi, the value of lambda required to restrict the objective function varies.

Conclusions

In this research, we figured out when the initial value of phi is 0, it isn't changed well. Besides, when phi is 180 or lambda is low, the constraint sometimes doesn't work properly, so it causes some bad training results in the experiments.

Therefore, it is good to use the value 90 as an initial phi, and if lambda is enough large for the objective function to be constrained, the training process is more stable than an unconstrained case.

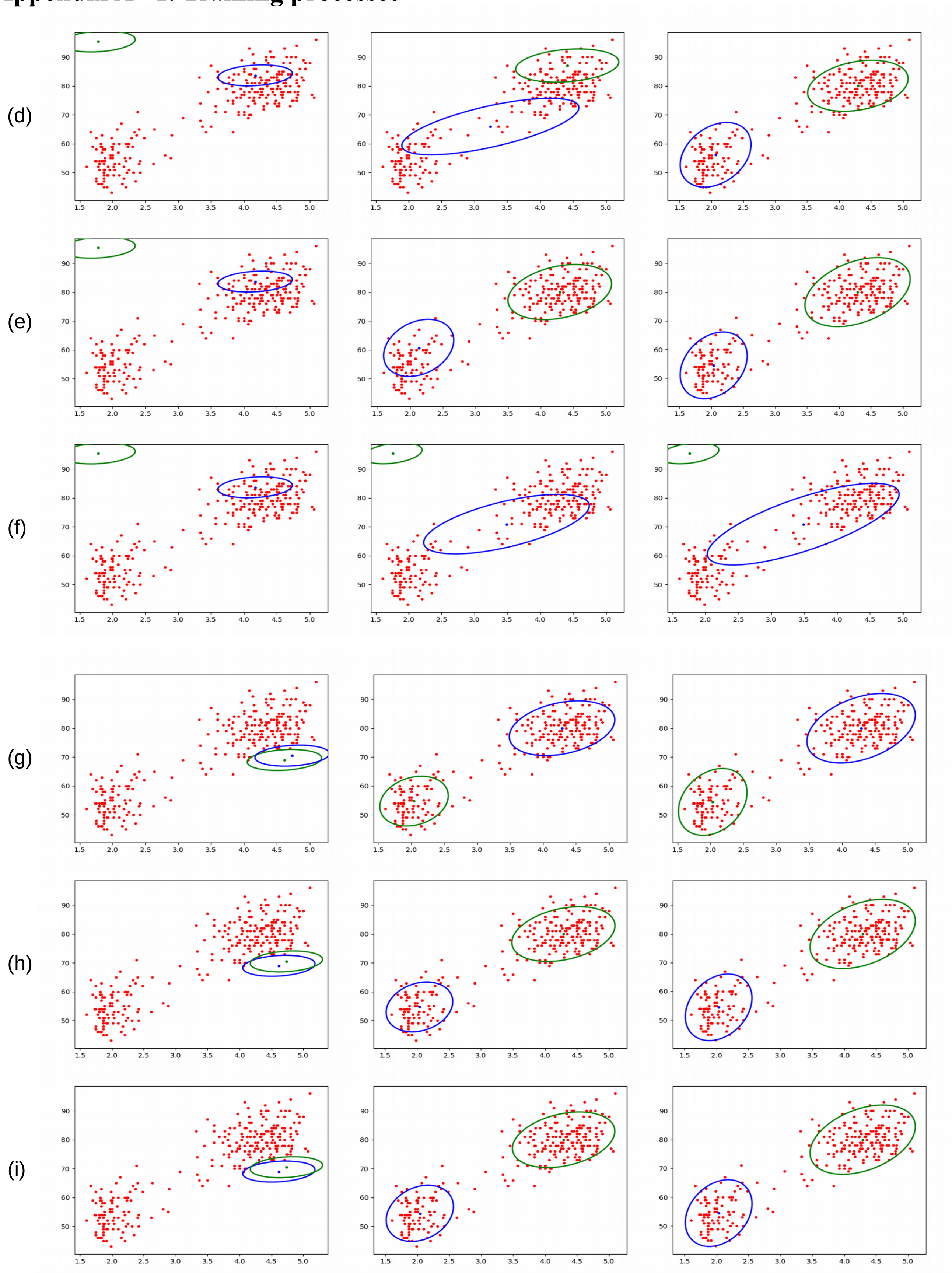
However, we found that lambda, which is too large, rather interferes with the convergence of the objective function. Also, as we changed phi to a trainable variable, it showed more stable training process, less affected by lambda.

References

[1] https://arxiv.org/pdf/1612.09199.pdf

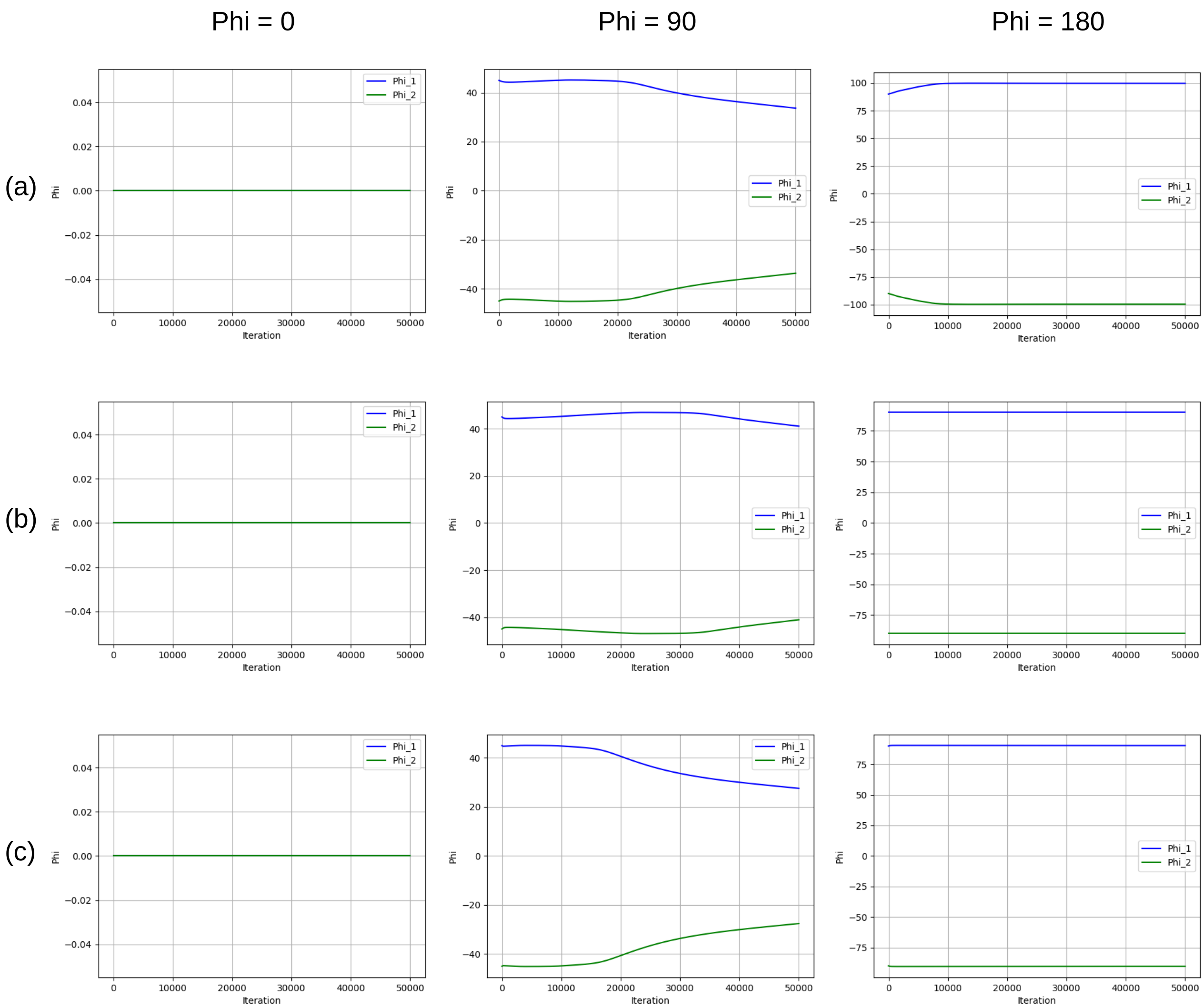
[2] https://www.kaggle.com/janithwanni/old-faithful

Appendix A - 1. Training processes



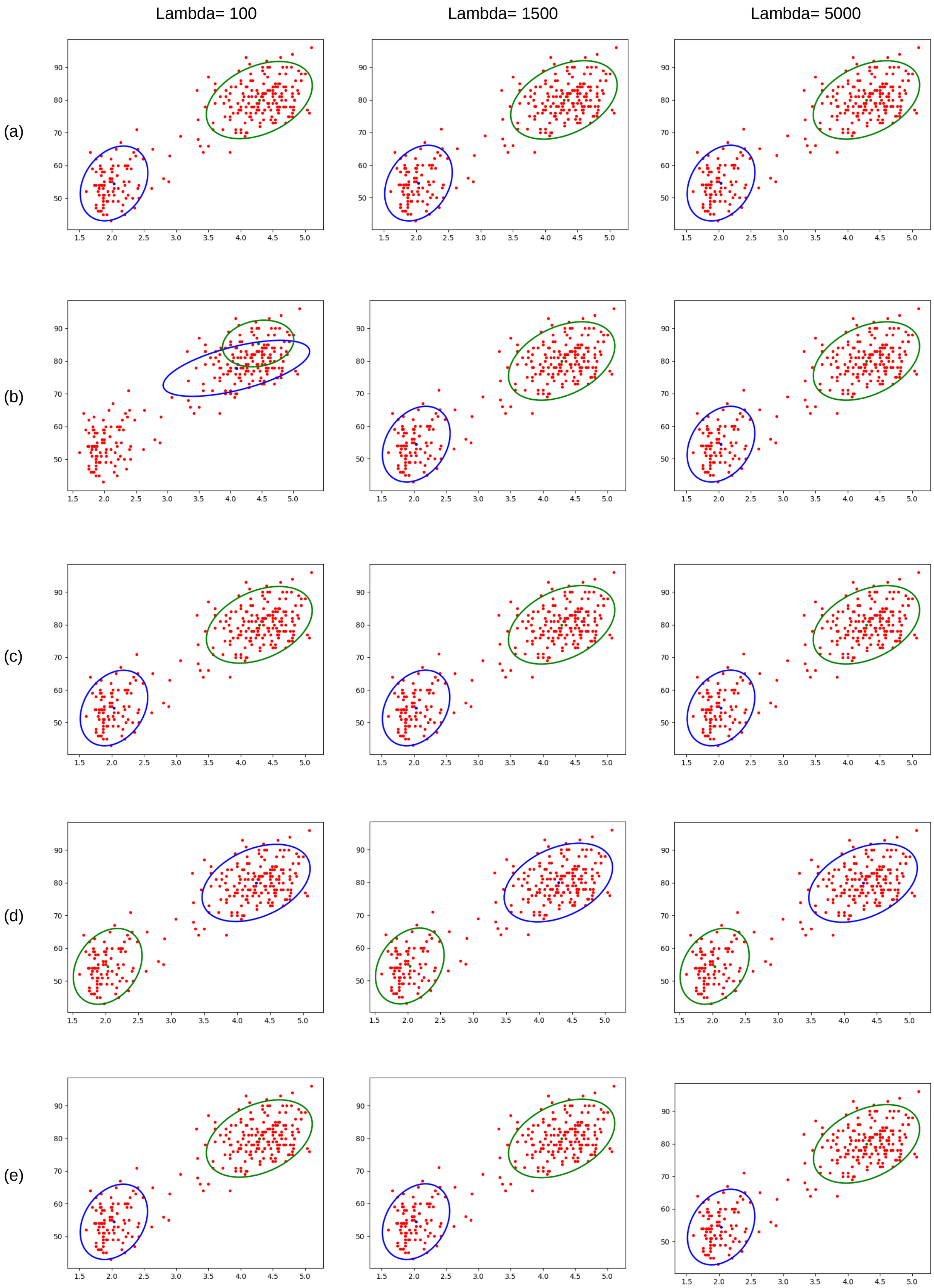
Appendix A-1. Training processes of test case 2 and 3 with the separate phis, 0, 90 and 180. (d) ~ (f) Test case 2 whose means are [4.171021823127277, 83.66322004888708], [1.781079954983019, 95.411542531776]. (g) ~ (i) Test case 3 whose means are [4.616385494792178, 68.97139287485163], [4.73416217991247, 70.48443049223583]. (d), (g) Phi 0. (e), (h) Phi 90. (f), (i) Phi 180.

Appendix A - 3. Phi



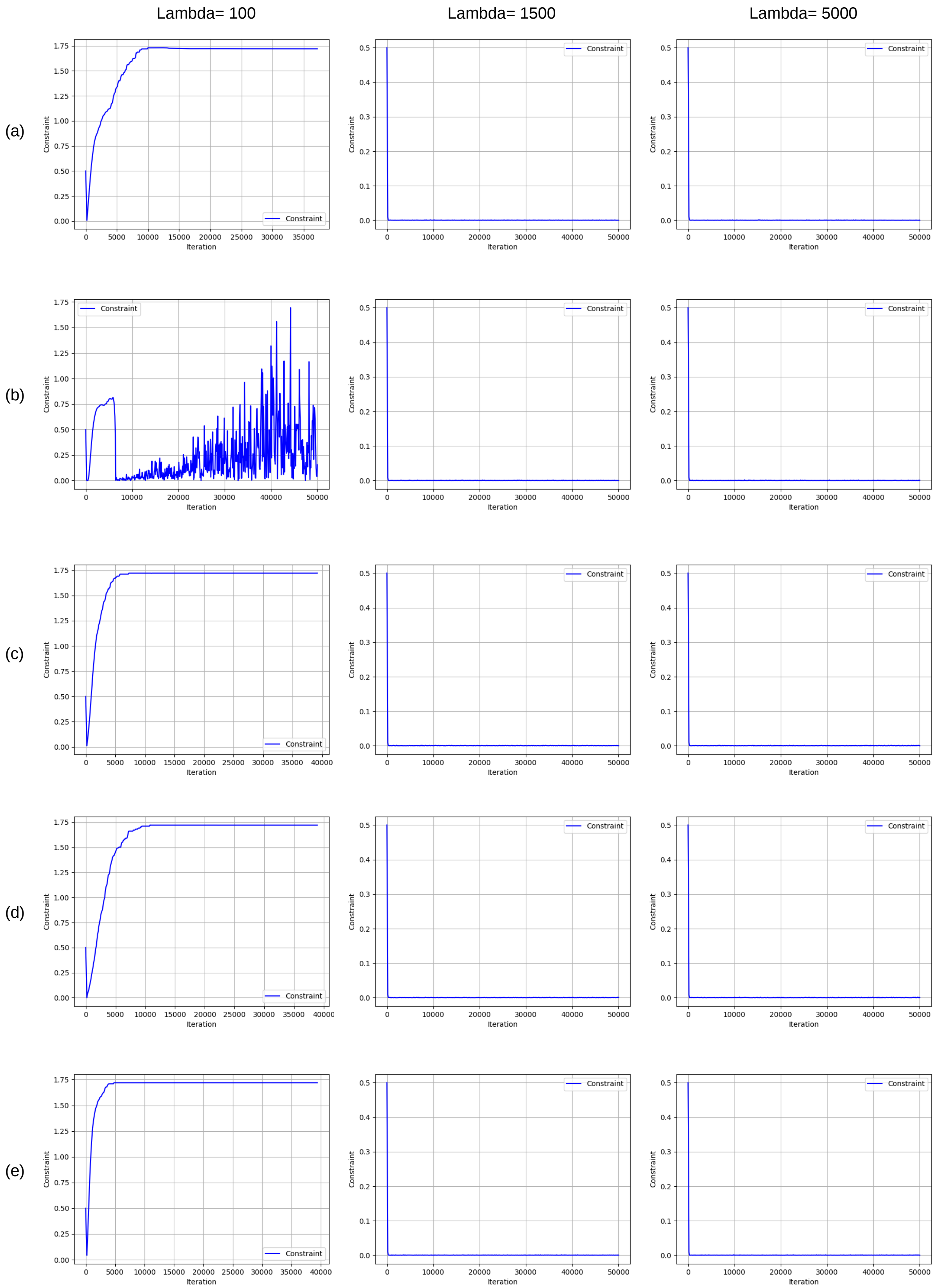
Appendix A-3. Phi of test case 1 ~ 3 with the separate phis. (a) Test case 1. (b) Test case 2. (c) Test case 3.

Appendix B - 1. Training results



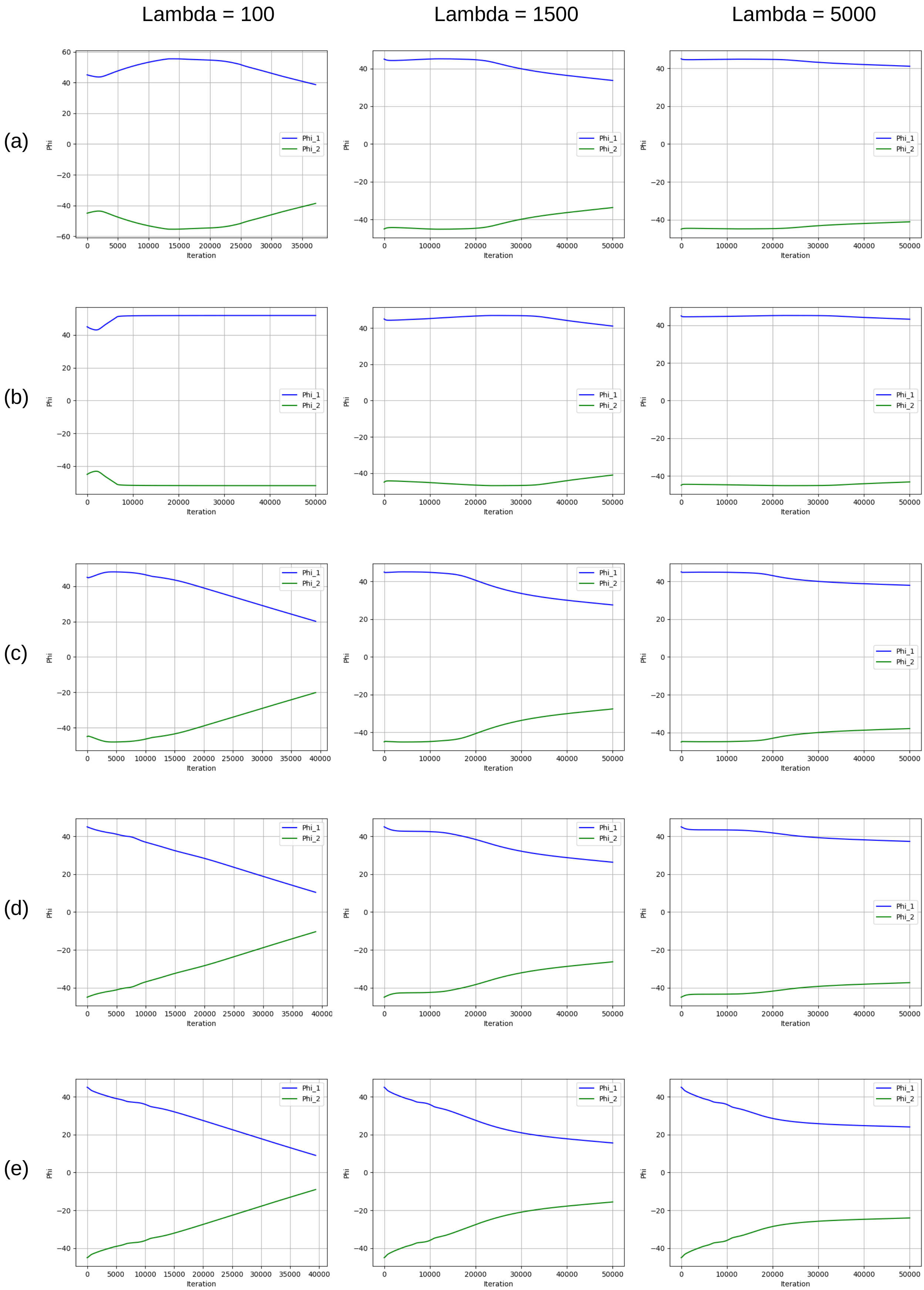
Appendix B-1. Training results of each test case. (a) Test case 1, means are [2.756031811312966, 76.62447648112042], [2.9226572802266397, 88.3509418943818], (b) Test case 2, means are [4.171021823127277, 83.66322004888708], [1.781079954983019, 95.411542531776], (b) Test case 3, means are [4.616385494792178, 68.97139287485163], [4.73416217991247, 70.48443049223583] (d) Test case 4, means are [3.5335808453329793, 60.79723193882826], [3.748786959785587, 46.017018024467745] (e) Test case 5, means are [4.399318766072071, 63.982790484402784], [2.511548424664534, 90.2446329311453]

Appendix B - 2. Constraint



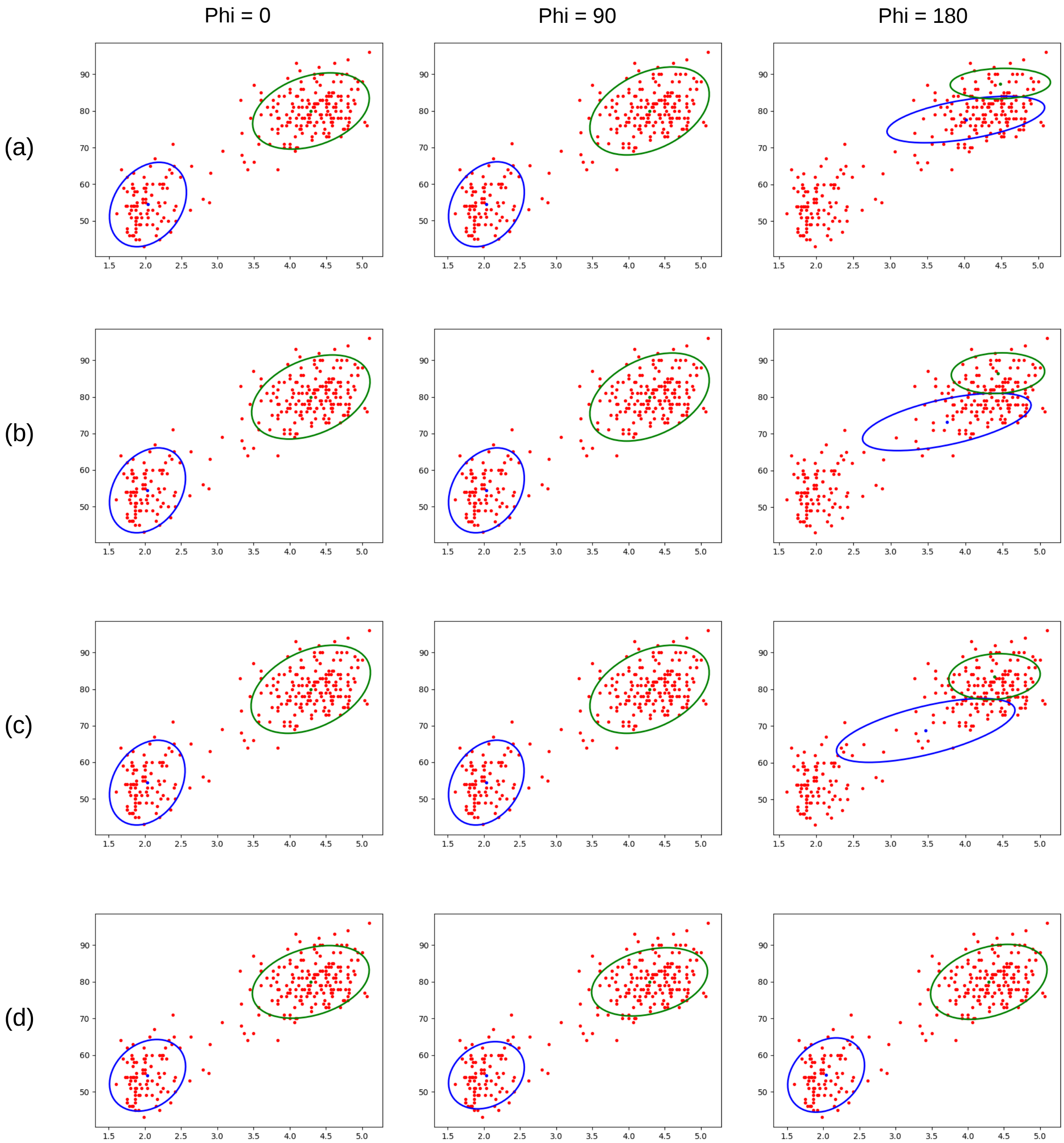
Appendix B-2. Constraint of each test case. (a) Test case 1. (b) Test case 2. (c) Test case 3. (d) Test case 4. (e) Test case 5.

Appendix B - 3. Phi



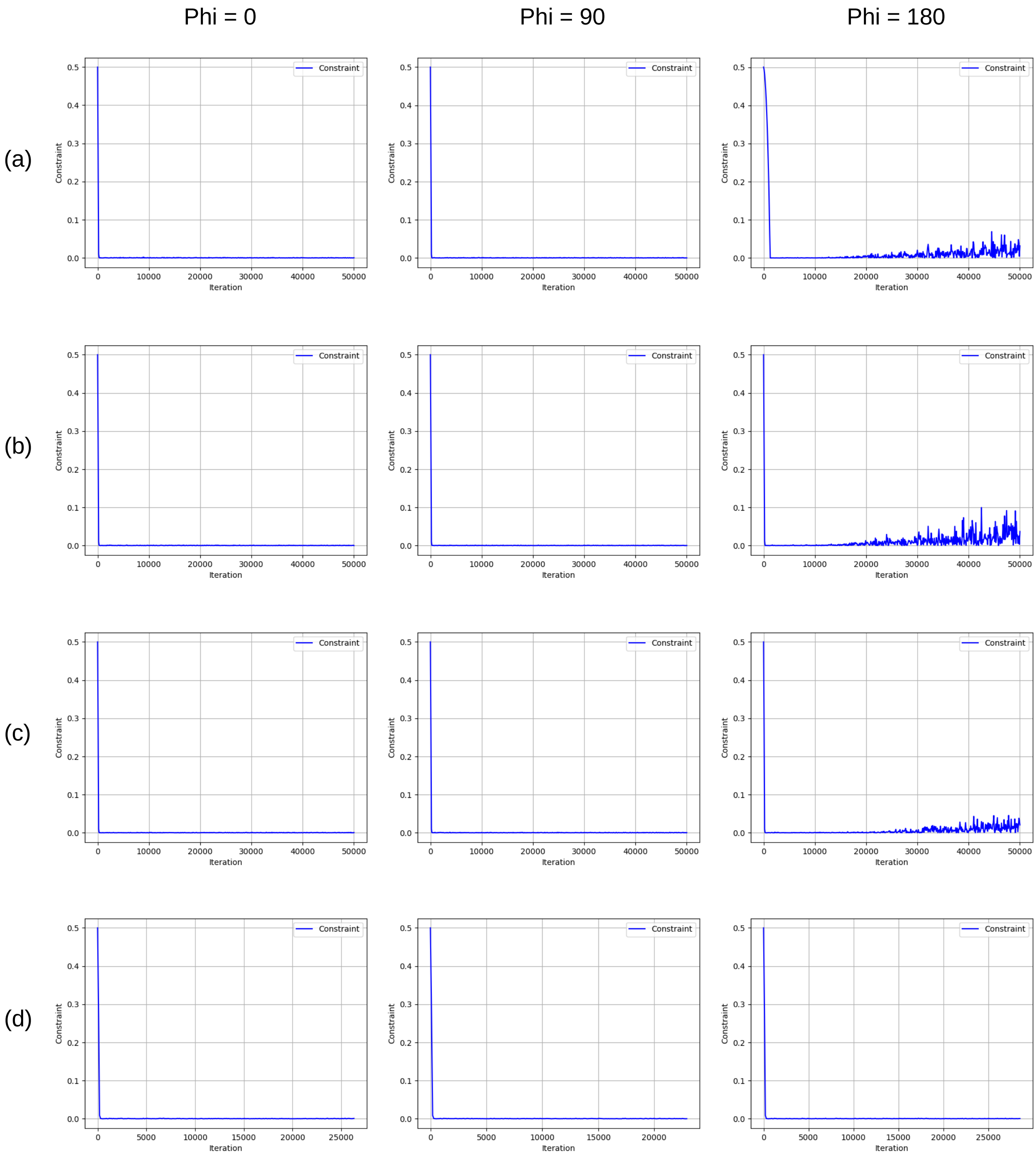
Appendix B-3. Phi with the initial value [45, -45]. (a) Test case 1. (b) Test case 2. (c) Test case 3. (d) Test case 4. (e) Test case 5.

Appendix C - 1. Training results



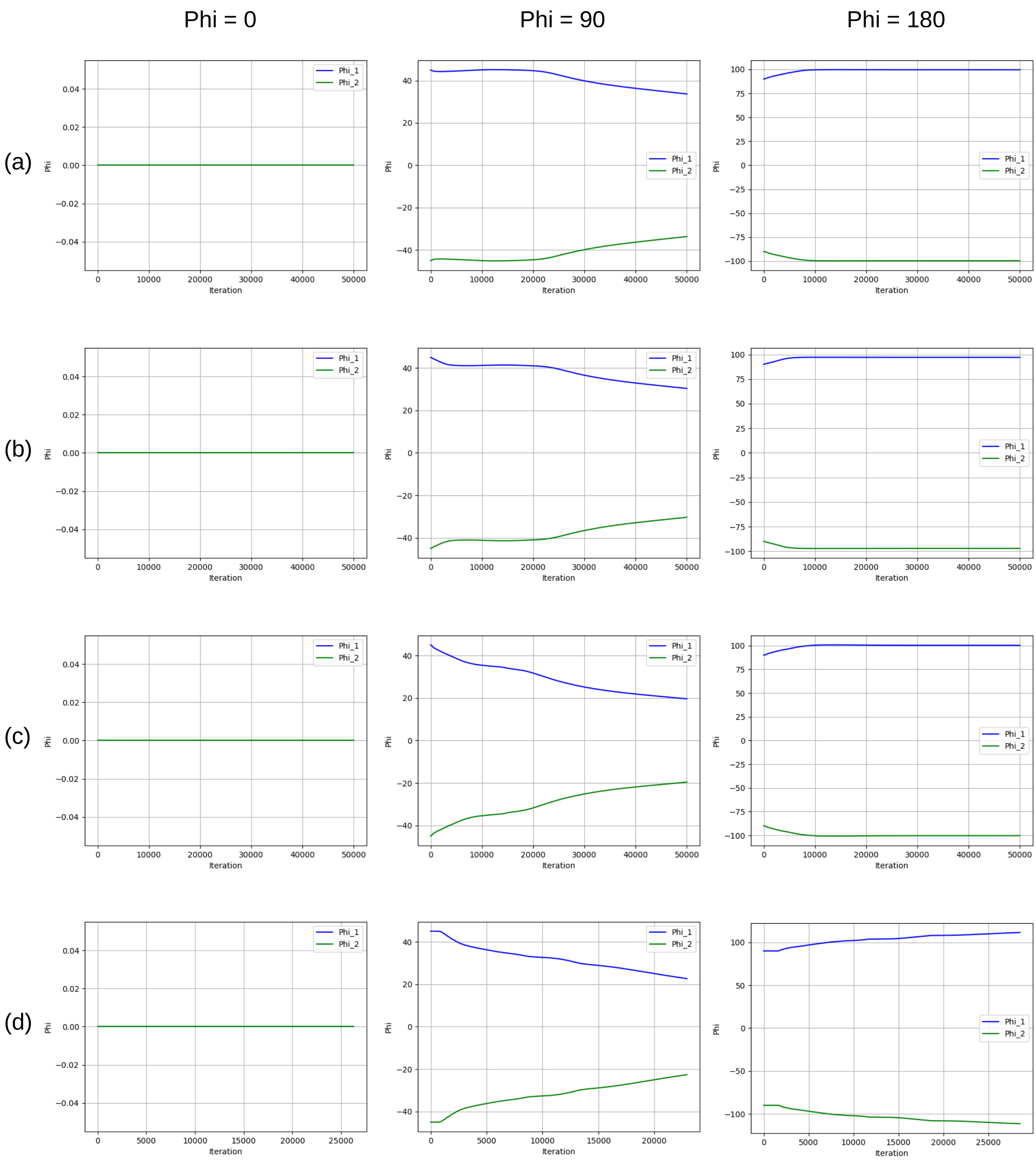
Appendix C-1. Training results of each test case. (a) Test case 1, means are [2.756031811312966, 76.62447648112042], [2.9226572802266397, 88.3509418943818], (b) Test case 2, means are [2.3, 72.6], [3.3, 92.4], (c) Test case 3, means are [1.5, 68.6], [3.3, 92.4] (d) Test case 4, means are [1.2, 62.6], [3.5, 94.4].

Appendix C - 2. Constraint



Appendix C-2. Constraints of each test case. (a) Test case 1. (b) Test case 2. (c) Test case 3. (d) Test case 4.

Appendix C - 3. Phi



Appendix C-3. Phis of each test case.
(a) Test case 1 and the final values are [0, 0], [33.670788, -33.670788], and [99.6261, -99.6261]
(b) Test case 2 and the final values are [0, 0], [30.318073, -30.318073], and [97.10504, -97.10504]
(c) Test case 3 and the final values are [0, 0], [19.598967, -19.598967], and [100.24093, -100.24093]
(d) Test case 4 and the final values are [0, 0], [22.647099, -22.647099], and [111.47447 -111.47447].