1 Comparison of Solutions Obtained in the Training Problem

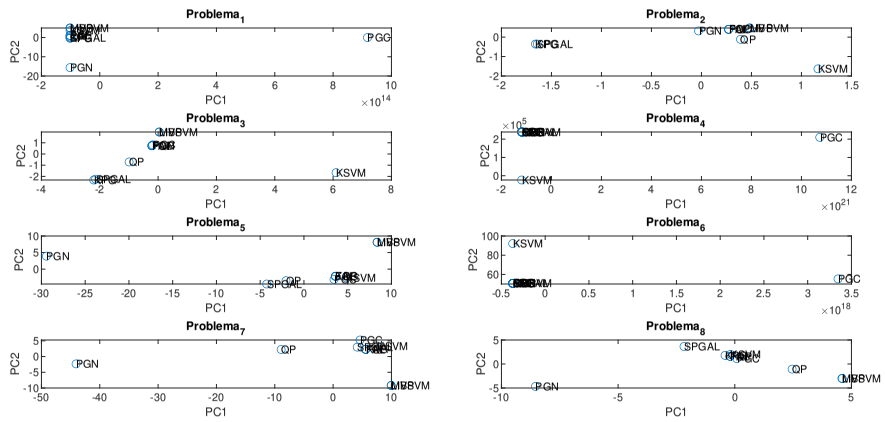
To explore the agreements between the solutions produced by different methods on the datasets used, two principal component analysis (PCA) plots were employed. PCA provides a visualization that highlights how observations in a dataset project onto the first two principal components after PCA analysis.

The graphical interpretation of these projections helps to understand the structure and variability of the data in a lower-dimensional space. Generally, the first principal component (PC1) captures the most variance in the data. When PC1 has a high value for an observation, it means that the observation has a large contribution in the direction of the greatest variability in the data. In other words, PC1 highlights the main sources of variability in the data. If two points have opposite PC1 values, it means they have high variability in the PC1 direction and are far apart in that direction. Meanwhile, the second principal component (PC2) captures the second largest amount of variance not explained by PC1. Therefore, PC2 represents a source of variability independent of PC1. When the second principal component has a high value for an observation, it means that the observation has a large contribution in the direction of the second greatest variability in the data, which is orthogonal to the PC1 direction.

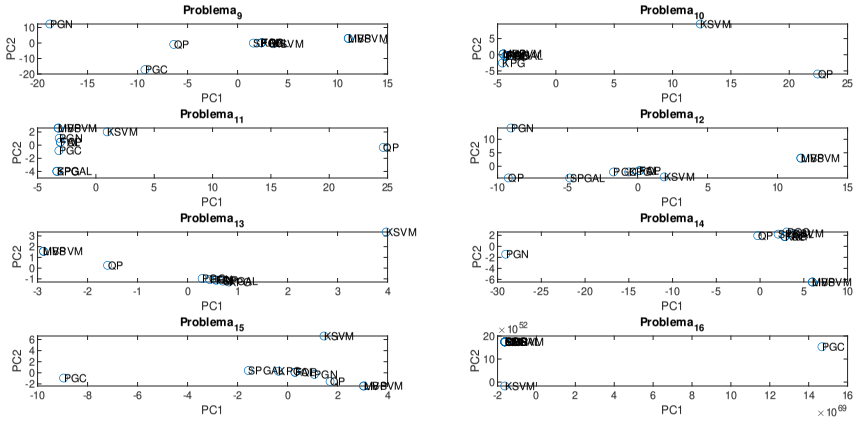
Thus, the graphical interpretation of projections onto the first two principal components can help identify clusters, trends, or differences in dataset observations. Observations close in the plot have similar characteristics or similar variability in the directions represented by the principal components, while observations that move away in the plot may indicate significant differences in the directions represented by the principal components. The direction of each principal component is defined by the original variables, and the coefficients associated with each variable can provide information on which variables contribute most to the component.

In Figures 1-4 are illustrated the comparative graphs of the first two principal components obtained through a PCA comparison stand out for Application 1, with Figures 1-2 related to linear kernel function and 3-4 related to RBF kernel function. For Figures 1-2, the algorithms KSVM, PGC, and PGN stand out because they present the highest variabilities in the first two principal components of the solutions obtained compared to the other algorithms. In the case of KSVM's solutions, it is interesting to note that it met its stopping criteria in all generated problems when the linear kernel function was used, taking more distinct directions compared to algorithms based on Active Constraints, Filter, and projections. For Figures 3-4, observing the case where the RBF kernel function was used in Application 1, the same pattern observed with the linear kernel function occurred: formations of visually well-highlighted clusters in all problems, with the points corresponding to the solutions of the KSVM, PGC, and PGN algorithms showing the highest variabilities in the first two principal components of the obtained solutions. Again, this highlights how the algorithms followed similar directions to achieve an optimized solution.

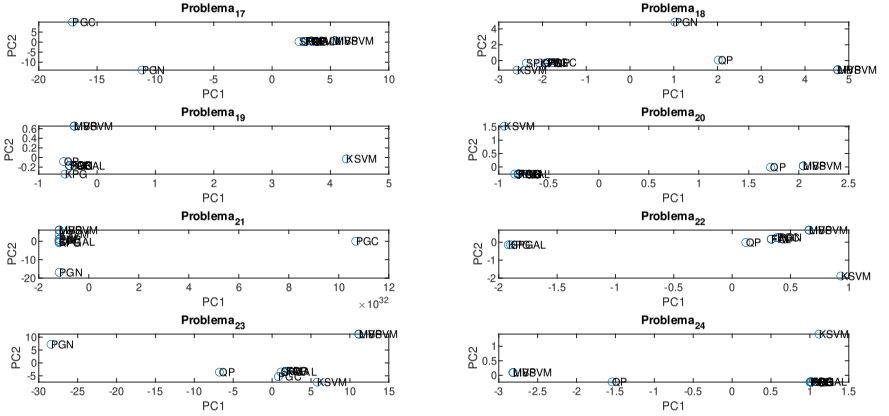
As for the results for the Application 2, the results obtained followed the same pattern as in Application 1, regardless of the kernel function considered. In Application 2, regardless of the kernel function, Figures 5-6 show the formation of distinctly visible clusters in the 32 problems, with some points exhibiting slightly larger displacements. In Figures 5-6, the algorithms KSVM, PGC, and PGN also stand out due to their greater variability in the first two principal components compared to the other algorithms. It is worth noting that these algorithms did not meet the imposed stopping criteria, adopting more distinct directions compared to the algorithms based on Active Constraints, Filter, and projections. The algorithms that clustered the most were MVP and LIBSVM, while FAL, FQP, SPGAL, and QP remained as peripheral points to the clusters of MVP and LIBSVM. This highlights that the solutions of the algorithms took the same direction for all problems until they reached their respective stopping criteria.



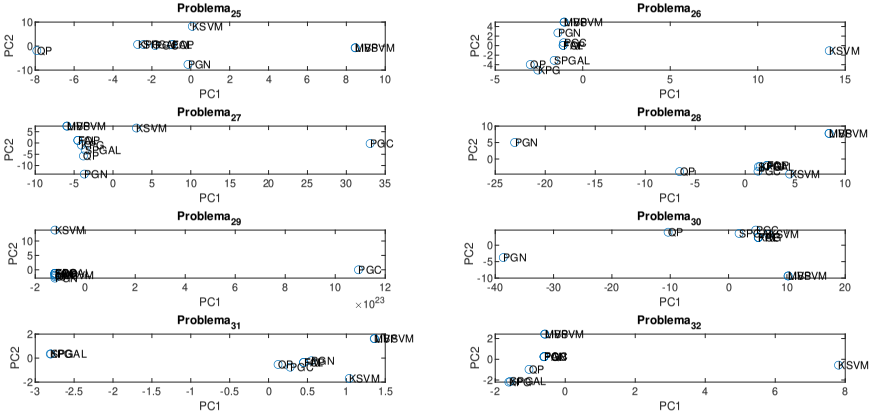
(a) Problems 1-8.



(b) Problems 9-16.

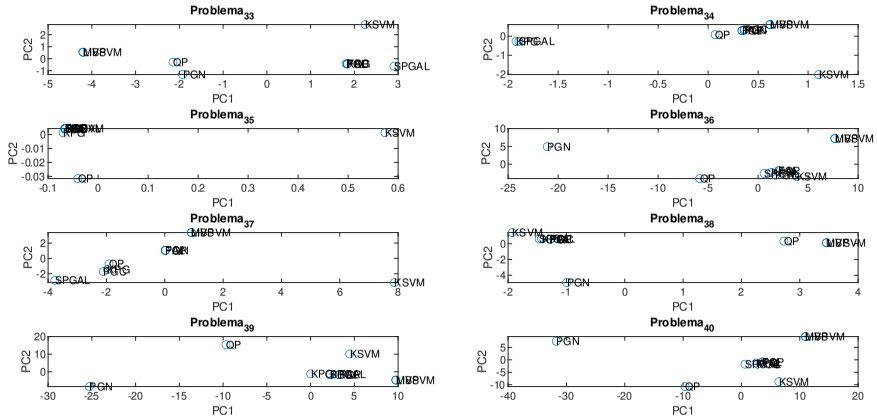


(c) Problems 17-24.

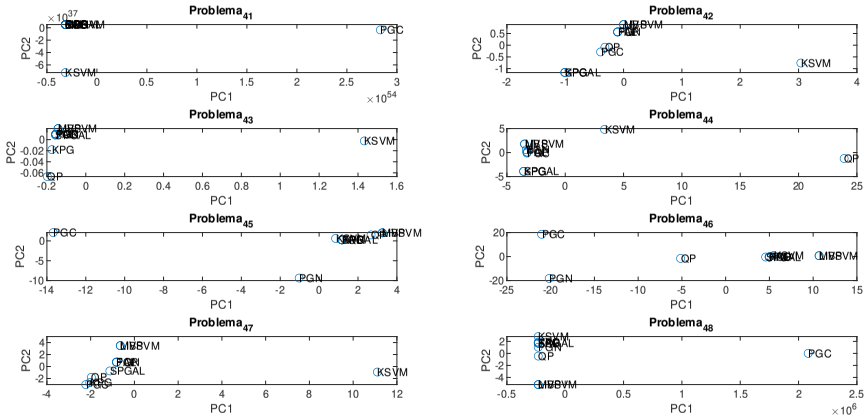


(d) Problemas 25-32.

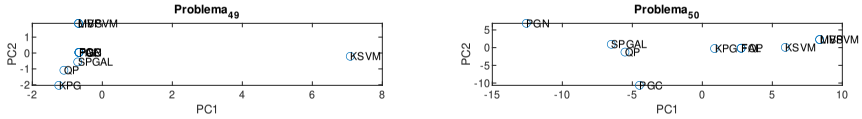
Figure 1: Application 1: Comparison of the main components of the solutions to randomly generated data sets - Linear *kernel* function (Part 1).



(a) Problems 33-40.

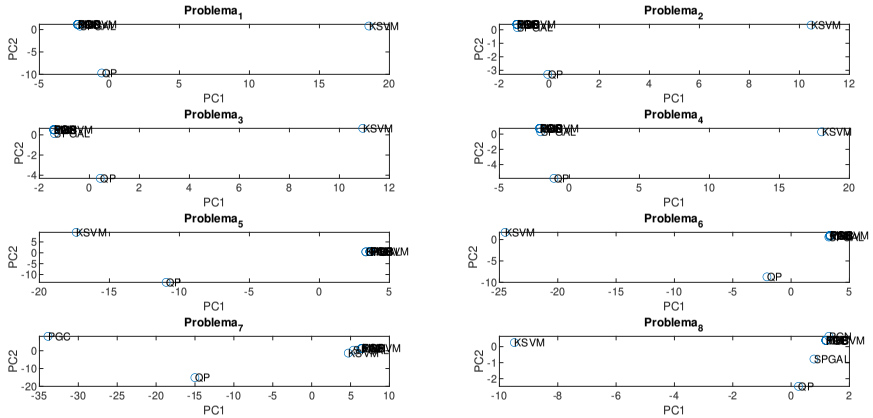


(b) Problems 41-48.

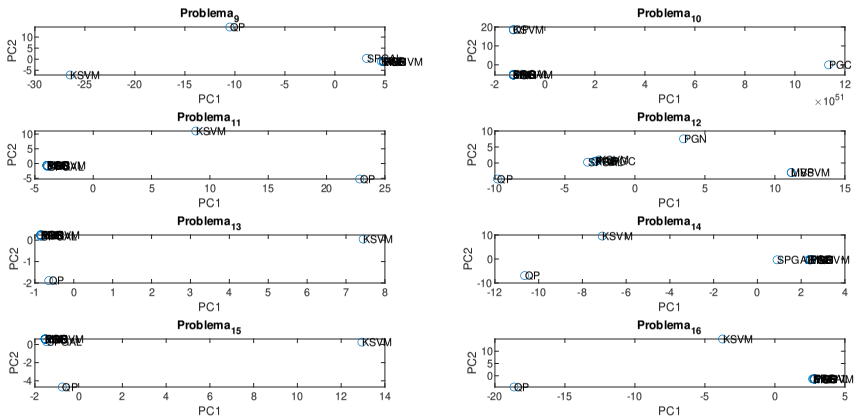


(c) Problems 49-50.

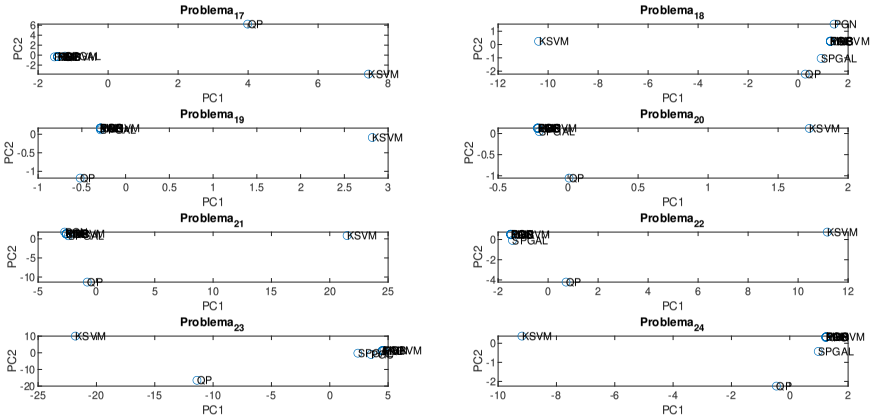
Figure 2: Application 1: Comparison of the main components of the solutions to randomly generated data sets - Linear *kernel* function (Part 2).



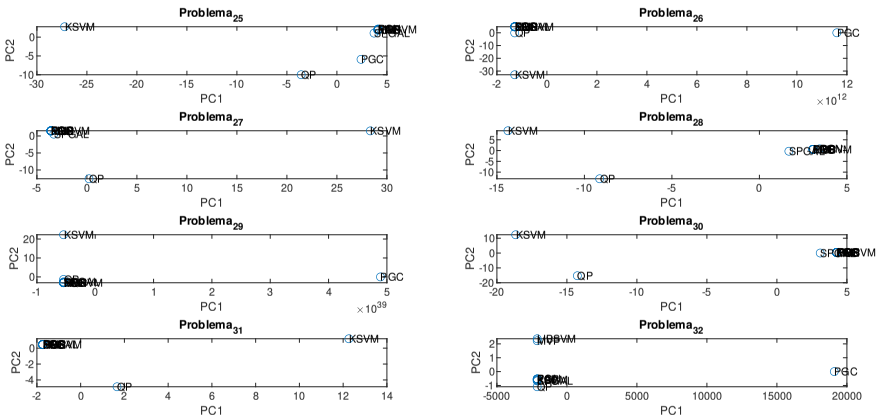
(a) Problems 1-8.



(b) Problems 9-16.

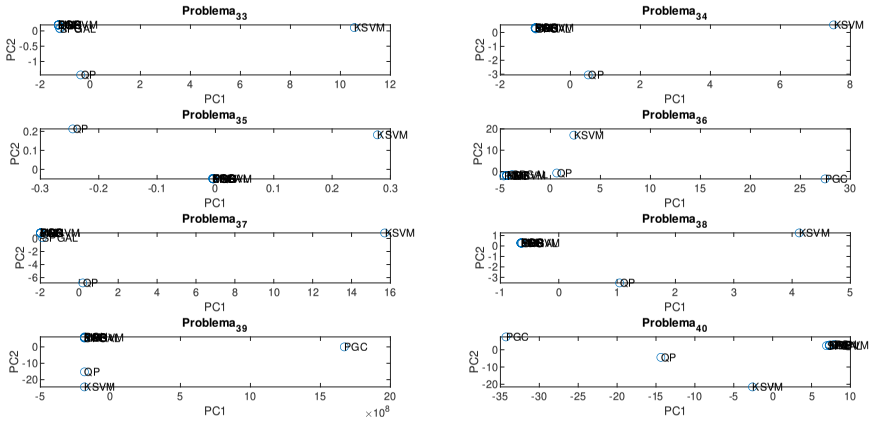


(c) Problems 17-24.

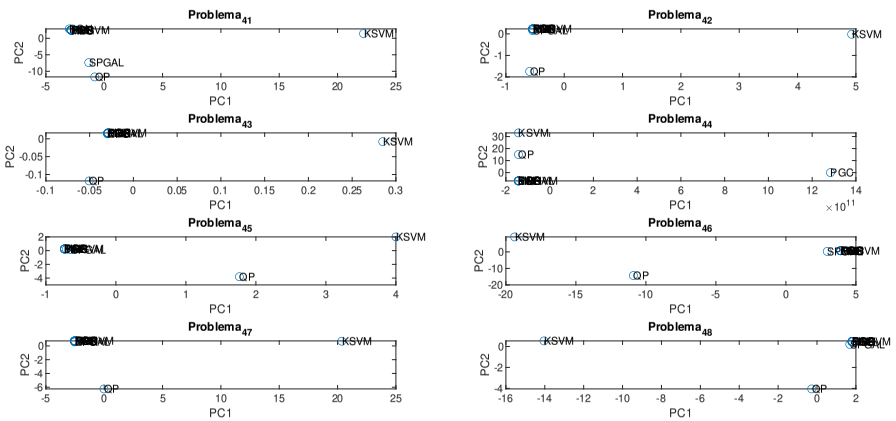


(d) Problems 25-32.

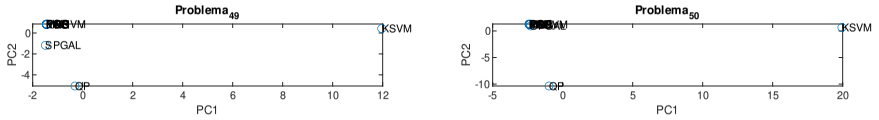
Figure 3: Application 1: Comparison of the main components of the solutions to randomly generated data sets - RBF *kernel* function (Part 1).



(a) Problems 33-40.

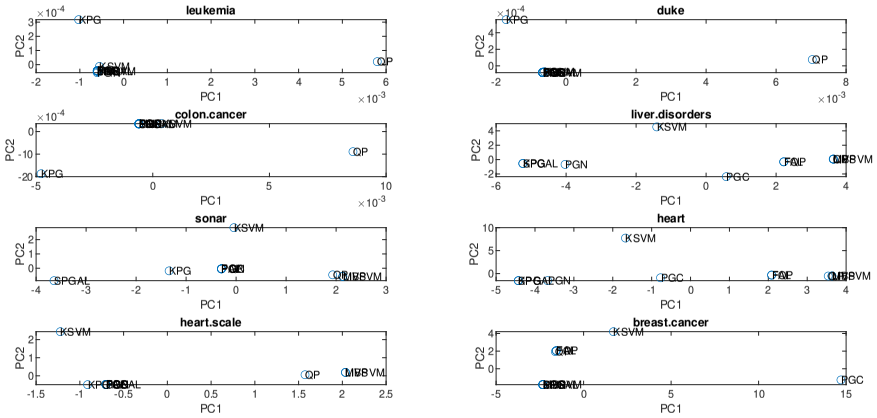


(b) Problems 41-48.

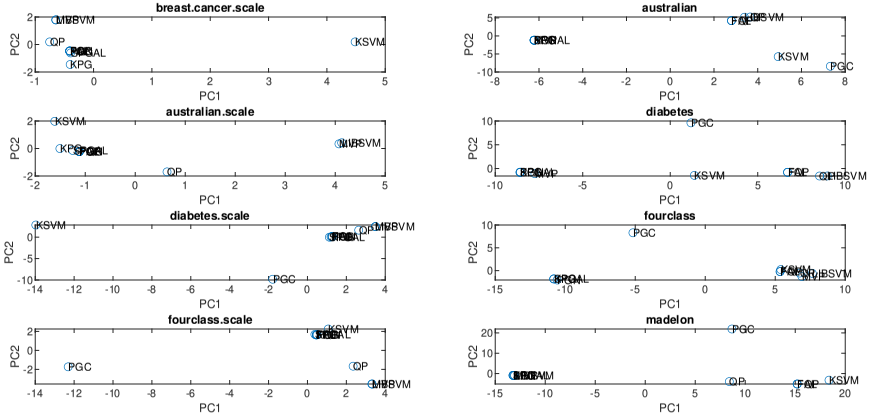


(c) Problems 49-50.

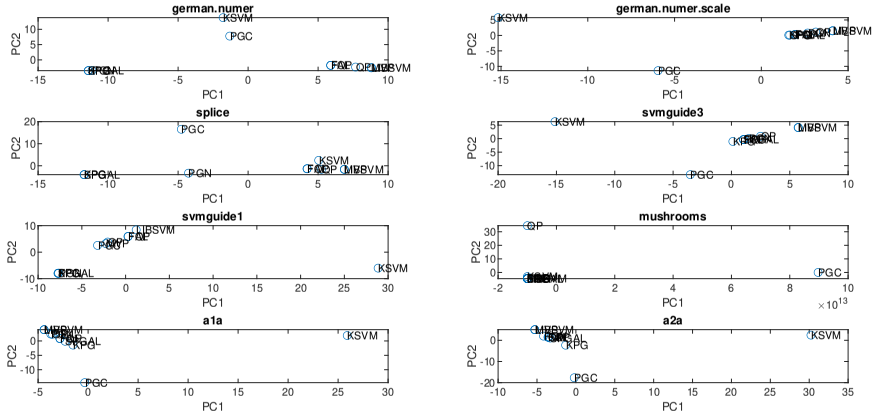
Figure 4: Application 1: Comparison of the main components of the solutions to randomly generated data sets - RBF *kernel* function (Part 2).



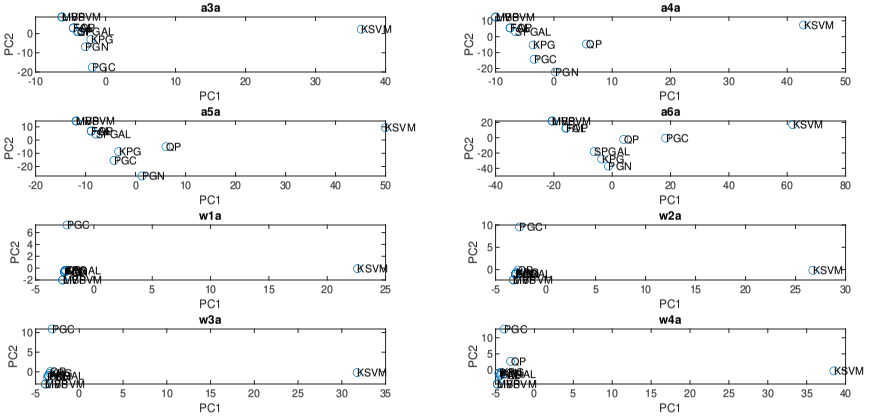
(a) Conjunto de dados: *leukemia*, *duke*, *colon.cancer*, *liver.disorders*,*sonar*, *heart*, *heart.scale* e *breast.cancer*.



(b) Conjunto de dados: *breast.cancer.scale*, *australian*, *australian.scale*,*diabetes*, *diabetes.scale*, *fourclass*, *fourclass.scale* e *madelon*.

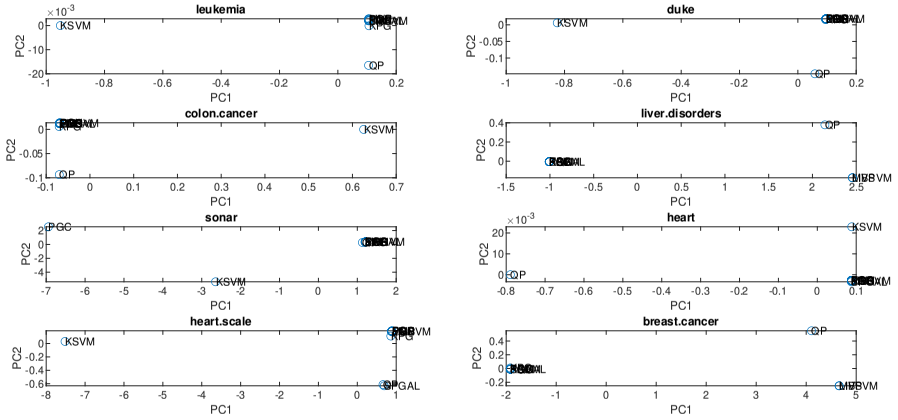


(c) Conjunto de dados: *german.numer*, *german.numer.scale*, *splice*, *svmguide1*, *mushrooms*, *a1a* e *a2a*.

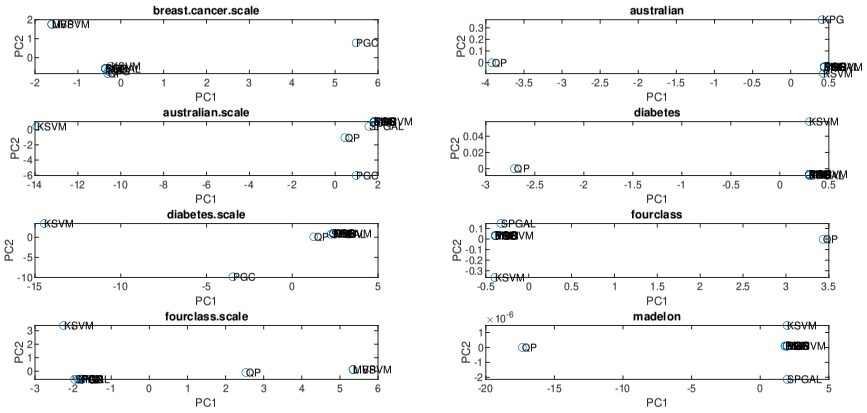


(d) Conjunto de dados: *a3a*, *a4a*, *a5a*, *a6a*, *w1a*, *w2a*, *w3a* e *w4a*.

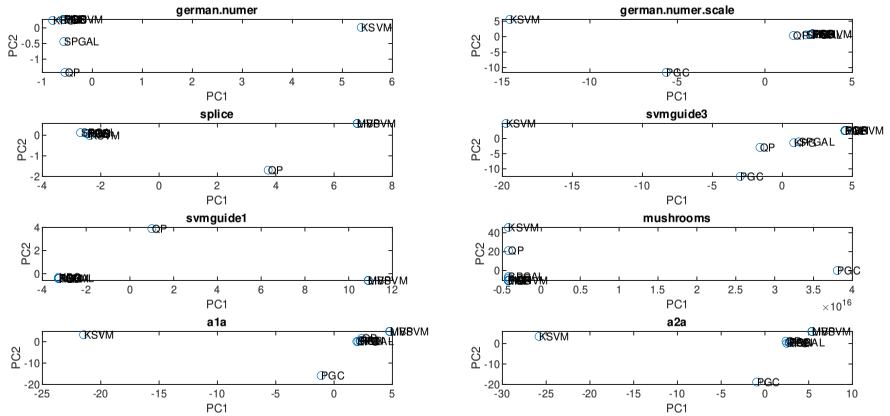
Figure 5: Aplicação 2: Comparativo das componentes principais das soluções aos conjuntos de dados gerados aleatoriamente - Função *kernel* linear.



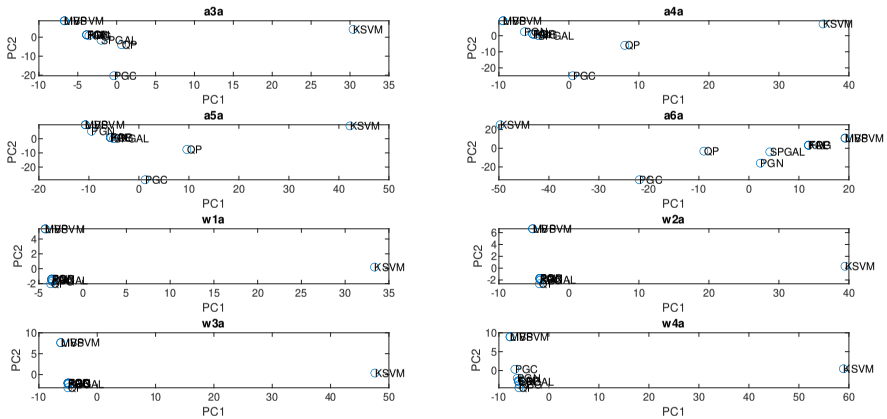
(a) Conjunto de dados: *leukemia*, *duke*, *colon.cancer*, *liver.disorders*,*sonar*, *heart*, *heart.scale* e *breast.cancer*.



(b) Conjunto de dados: *breast.cancer.scale*, *australian*, *australian.scale*,*diabetes*, *diabetes.scale*, *fourclass*, *fourclass.scale* e *madelon*.



(c) Conjunto de dados: *german.numer*, *german.numer.scale*, *splice*, *svmguide1*, *mushrooms*, *a1a* e *a2a*.



(d) Conjunto de dados: *a3a*, *a4a*, *a5a*, *a6a*, *w1a*, *w2a*, *w3a* e *w4a*.

Figure 6: Aplicação 2: Comparativo das componentes principais das soluções aos conjuntos de dados gerados aleatoriamente - Função *kernel* RBF.

2 Survival Analysis of Algorithms Involving Execution Times

Survival analysis is a statistical methodology widely used in medical research, social sciences, and many other fields to investigate and model events involving the duration of a certain phenomenon, such as time until the occurrence of an adverse event like death, equipment failure, or any other system failure. This approach is particularly useful when dealing with censored data, where information about the event duration is not available for all individuals or observation units.

One of the most common methods for conducting survival analysis is the use of Kaplan-Meier curves. Kaplan-Meier curves provide a graphical representation of survival functions over time, allowing visualization of survival rates at different intervals, i.e., the survival function provides valuable information about the probability of an event occurring over time. As time progresses, the survival curve shows how this probability decreases. It is especially useful for comparing different groups and assessing the impact of independent variables on survival. By analyzing Kaplan-Meier curves, researchers can identify if there are significant differences in survival rates between groups and, if necessary, conduct statistical tests to determine if these differences are statistically significant.

For the research conducted, Kaplan-Meier curves represent the probability that the execution of a certain algorithm has not yet ended relative to the execution time. To construct a Kaplan-Meier curve, we start from an empirical survival function S as a function of time t. This function calculates the probability of an individual or object surviving beyond a specific point in time t. The function S is defined as

$$S(t) = \prod_{i=1}^{j} \left(1 - \frac{d_i}{n_i}\right),$$

where j is the number of distinct times at which the event of interest occurs, d_i is the number of events occurring at time t_i, and n_i is the number of individuals (or objects) at risk of experiencing the event at time t_i. The survival probability is calculated by multiplying the probability of not experiencing an event at that time, considering all previous times. The Kaplan-Meier curve is constructed by illustrating the values of S relative to time t and connecting the points. Therefore, in the graphs to be illustrated in this paper, the x-axis represents the execution time, while the y-axis represents the probability that the algorithm's execution is active. As time passes, the survival probability decreases as algorithm executions end. Hence, a curve is generated for each algorithm, according to the parameters of the training problem (**??**).

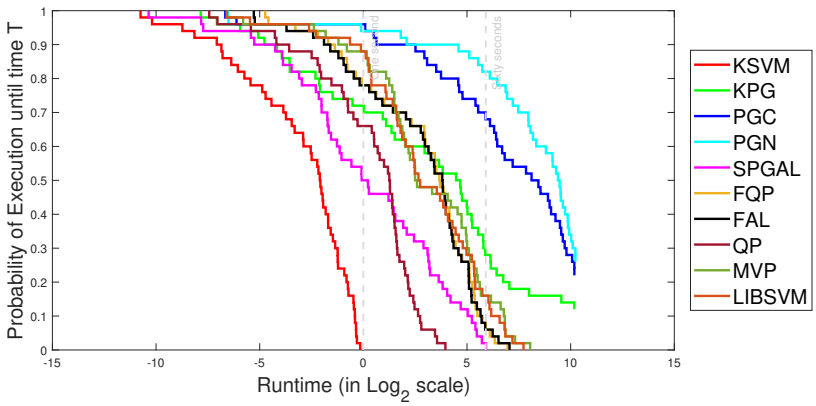
It is worth noting that when the curves of different algorithms intersect, it indicates significant differences in efficiency or execution time between these algorithms. To highlight such possible differences, a Log-Rank hypothesis test was also conducted considering a significance level for the test. The p-value of this test indicates whether there are significant differences in a pairwise comparison of algorithms. For the tests performed, the hypotheses to be tested were:

$$\begin{cases} H_0: & \text{There is no difference in survival curves between algorithms} \\ H_1: & \text{At least one pair of groups has significant differences} \end{cases}.$$

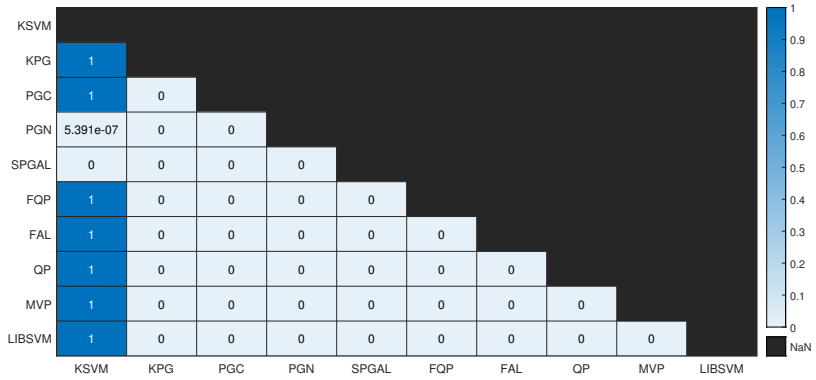
For the tests conducted, the significance level adopted was 0.05, where p-values below this level suggest significant differences in survival curves. For more information about the adopted hypothesis test and/or survival analysis, we suggest reading the works of [**?**, **?**], and [**?**].

For the Application 1, the Figures 7a and 7c illustrate the Kaplan-Meier curves of the algorithms in terms of the probability of execution at a certain time. It is again highlighted that the KSVM algorithm was the fastest in terms of training, followed by the LIBSVM, MVP, SPGAL, and QP algorithms. The Filter-based algorithms achieved similar performances, with their curves almost overlapping each other, but indicating a longer training time to obtain a solution. The algorithms with the Kaplan-Meier curve more shifted to the right in this experiment were PGC and PGN. As for Figures 7b and 7d, which illustrate the results obtained in the Log-Rank hypothesis test to verify the existence of significant differences in training time between two algorithms at a p-value of 0.05, they corroborate that the KSVM algorithm obtained the most differentiated training values compared to the other algorithms, regardless of the kernel function used in the experiment. In the case of the linear kernel function, the other algorithms did not show statistically significant differences between the presented values, while in the case of the RBF kernel function, there were differences in training time in the QP, MVP, and LIBSVM algorithms (besides KSVM) compared to the other algorithms.

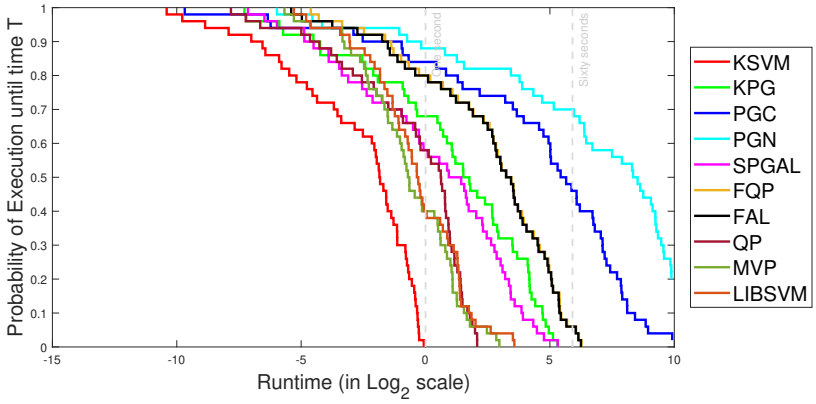
As for Application 2, the Figures 8a and 8c illustrates the Kaplan-Meier curves of the algorithms in terms of the probability of execution at a certain time. It is again highlighted that the SPGAL and KSVM algorithms were the fastest in terms of training, followed by the LIBSVM, MVP, and QP algorithms. The Filter-based algorithms achieved similar performances, with their curves almost overlapping each other, but indicating a longer training time to obtain a solution. The algorithms with the Kaplan-Meier curve more shifted to the right in this experiment were PGC and PGN. It is worth noting that there are curves of different algorithms intersecting in Figures 8a and 8c, giving an indication of significant differences in efficiency or execution time between these algorithms. Figures 7b and 7d, which illustrate the results obtained in the Log-Rank hypothesis test to highlight the existence of significant differences in training time between two algorithms at a p-value of 0.05, corroborate that the KSVM and SPGAL algorithms obtained the most differentiated training values compared to the other algorithms, for the case of the linear kernel function, and that the QP, MVP, and LIBSVM algorithms (besides KSVM) obtained differences compared to the other algorithms in the case of using the RBF kernel function. The other algorithms did not show statistically significant differences between the presented values.



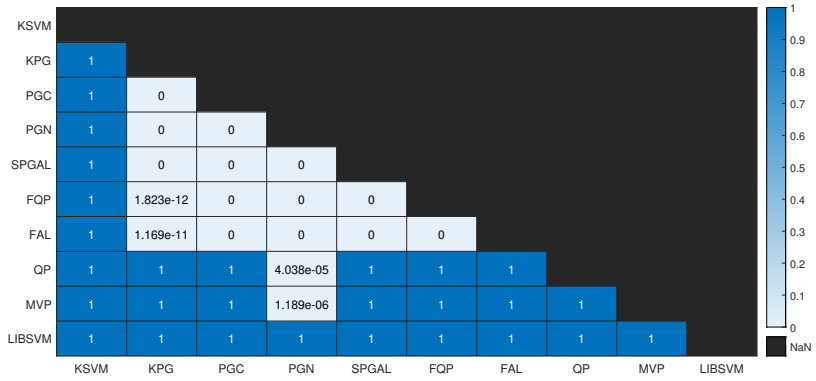
(a) Kaplan-Meier curves - linear *kernel* function.



(b) Log-Rank test heat map - linear *kernel* function.

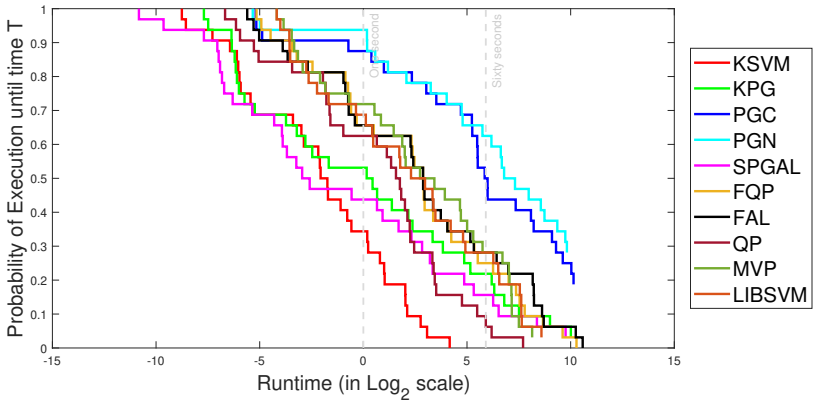


(c) Kaplan-Meier curves - RBF *kernel* function.

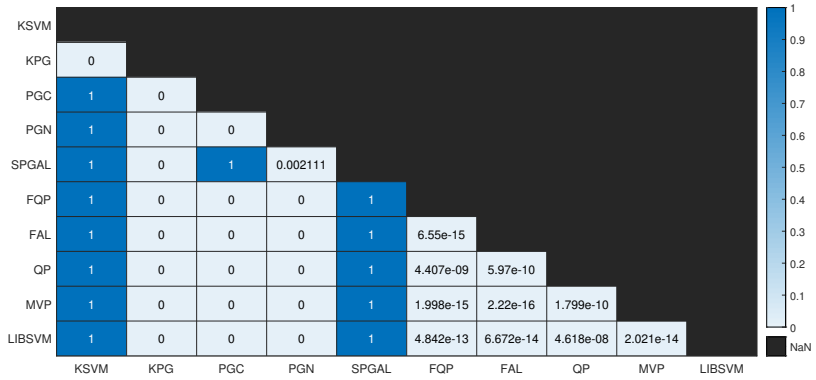


(d) Log-Rank test heat map - RBF *kernel* function.

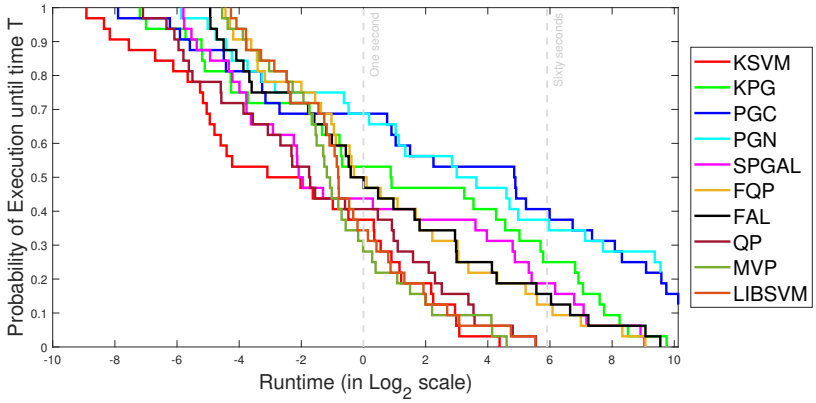
Figure 7: Application 1: Algorithm survival analysis.



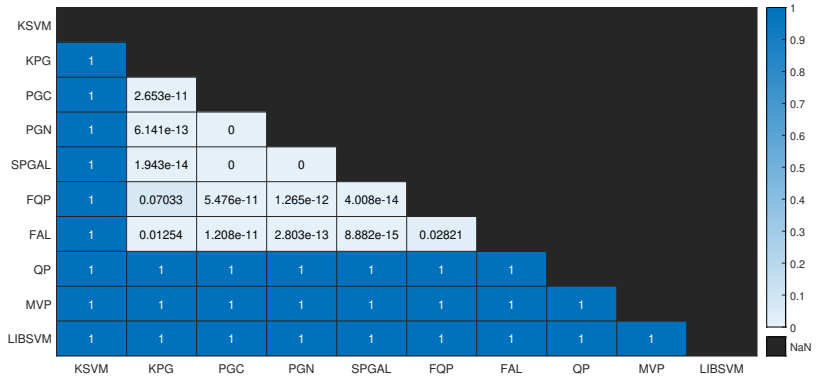
(a) Kaplan-Meier curves - linear *kernel* function.



(b) Log-Rank test heat map - linear *kernel* function.



(c) Kaplan-Meier curves - RBF *kernel* function.



(d) Log-Rank test heat map - RBF *kernel* function..

Figure 8: Application 2: Algorithm survival analysis.