A Pilot Study of Multidimensional Scaling in the Iterated Local Search Metaheuristic with a Novel k-NN Perturbation Method.

Chou, Remy

Universiteit Leiden

Faculteit der Sociale Wetenschappen

A Pilot Study of Multidimensional Scaling in the Iterated Local Search Metaheuristic with a Novel *k*-NN Perturbation Method.

Master's Thesis Psychology,

Methodology and Statistics Unit, Institute of Psychology

Faculty of Social and Behavioral Sciences, Leiden University

Date: 30-03-2023

Supervisor: Dr. Frank M.T.A. Busing (internal)

Abstract

Multidimensional Scaling (MDS) is an iterative optimization technique used to find underlying structures of proximity data by constructing a lower-dimensional coordinate matrix. The resulting plot can provide valuable insights into the structures of the data. However, MDS is susceptible to falling into local minima, especially in datasets with measurement error or for unidimensional solutions. To address this issue, the use of Iterated Local Search (ILS) has been proposed. ILS involves using a *localsearch* algorithm to find an initial solution, then adds perturbations to this solution. The perturbed solution is used as the next initial solution in the *localsearch* to explore better local minima in the vicinity. In this thesis, effectiveness of ILS with MDS as the *localsearch* algorithm is investigated. Also, two new perturbation methods based on switching objects with one of its *k*-Nearest Neighbours (*k*-NN) were compared with random perturbation. A simulation study was conducted to tune the perturbation parameters: perturbation strength and perturbation size, on unidimensional and two-dimensional data of different dataset sizes. Multiple random starts was used as baseline to determine usefulness. The tuned ILS procedures were then compared with MDS using classical scaling and multiple random starts on four well-known real-life datasets. The results indicated that ILS is an effective metaheuristic for MDS, and that for unidimensional solutions, stronger and more perturbation is needed, whereas two-dimensional solutions benefit from weaker and less perturbation. All three ILS procedures were capable of finding similar or even better solutions than known MDS methods: classical scaling and multiple random starts. Therefore, we recommend using ILS instead of classical scaling and multiple random starts, or in combination with classical scaling. Further research can explore the use of *k*-NN perturbation in other metaheuristics such as genetic algorithms and tuning ILS parameters such as the acceptance criterion.

Key words: Multidimensional Scaling, Stress, Local minima, Iterated Local Search, Random Perturbation, *k*-NN perturbation.

1. Introduction

Multidimensional Scaling (MDS) is a method widely used for exploring structures in proximity data. Proximity data can be similarities or dissimilarities between pairs of objects: an example of dissimilarity data are the distances between cities, an example of similarity data is a correlation matrix between intelligence tests. (Mair, Borg & Rusch, 2016). MDS transforms the proximities in such a way, that the Euclidean distances in *p* dimensional space between pairs of objects represent their proximities as accurately as possible. "The Euclidean distances can then be transformed into a coordinate matrix in *p* dimensions which gives researchers an accessible visualization of the underlying structures between the variables: objects closer to each other indicate higher correlation between the variables they represent" (Borg, Groenen & Mair, 2013).

Kruskal (1964) developed a loss function (1) to assess the badness of fit between the Euclidean distances of the coordinate matrix and the observed proximities called raw Stress (σ_r):

$$\sigma_r = \sum_{i<j} w_{ij}(d_{ij}(\mathbf{X}) - \delta_{ij})^2 \quad (1)$$

,

where **X** is an *n* x *p* coordinate matrix, with *n* being the number of objects. $d_{ij}(\mathbf{X})$ is the Euclidean distance between object *i* and object *j*. δ_{ij} is the dissimilarities between object *i* and *j*, and ω_{ij} is the weight given to *ij* pair of objects. Raw Stress is not very informative: a large value does not necessarily indicate bad fit as raw Stress increases with scaling of the proximities. For example: proximities expressed in kilometres compared to meters yield the same solution, but in meters the raw Stress increases 1000 times. A remedy for this dependency on scale, is Stress-1 (σ_1) (2), in which raw Stress is expressed in relation to the size of **X** (Kruskal, 1964).

$$\sigma_1 = \sqrt{\frac{\sum_{i<j} w_{ij}(\delta_{ij} - d_{ij}(\mathbf{X}))^2}{\sum_{i<j} w_{ij} d_{ij}^2(\mathbf{X})}} \quad (2)$$

.

When the visualization perfectly reproduces the observed data, δ_{ij} - $d_{ij}(\mathbf{X})$ is zero for every pair of objects, thus the resulting Stress-1 is equal to zero. However, MDS solutions are mostly non-zero because of insufficient dimensionality: for any given dataset it may be impossible to represent the proximity data in lower dimensions. But a higher number of dimensions makes visualizing MDS solutions increasingly difficult: a solution in four

dimensions is already virtually impossible for humans to interpret. Therefore, the aim should be to find the best solution in three or lower dimensions. However, with lower dimensionality the likelihood increases of the optimization of Stress-1 to get stuck in local minima: suboptimal solutions with higher levels of Stress-1 than the global minimum, where small changes to the solutions lead to higher Stress-1 (Groenen and Heiser, 1996).

The minimization of Stress-1 is done by starting with an initial configuration (IC) and repeatedly moving the objects in Euclidean space until the improvement of Stress-1 stops. However, depending on the IC used, MDS will find different solutions with different or same levels of Stress-1. A good starting point for an IC is classical scaling (Torgerson, 1952). It takes the observed data into account: it converts the observed data by means of Torgerson Scaling (Torgerson, 1952) into an initial configuration. However, it has its weaknesses: when measurement error and outliers are present in the observed data, it no longer guarantees finding the correct underlying structure (Borg and Mair, 2017). Also, it is inflexible: classical scaling uses the same transformation over the observed data to achieve a singular solution. With these weaknesses, a MDS solution using only classical scaling should not be relied on (Mair, Borg and Rusch, 2016).

It is recommended to also check the fit of the solution using multiple random starts (MR) (Mair, Borg and Rusch, 2016). As the name implies, MR starts MDS multiple times with completely random initial configurations. By using multiple random initial configurations, many different solutions can be found. After which the solutions with the lowest Stress-1 can then be inspected and compared with the solution found by classical scaling. However, MR also has its weaknesses: it is shooting blindly without taking the observed data in account and is more susceptible to falling into local minima. Also, it is slow: even with the current computing power, many multiple random starts of MDS on big datasets can reach high runtimes.

A possible remedy for the weaknesses of the two previously mentioned IC's is the Iterated Local Search (ILS) metaheuristic. The general framework of ILS is as follows: run a problem-specific approximation algorithm, from now on called *localsearch* (in this thesis: MDS). The solution of this *localsearch* can be evaluated by a cost function and should be minimized (in this thesis: Stress-1). Instead of randomly trying the *localsearch* with random restarts, ILS starts the *localsearch* and applies perturbation to the resulting solution. The perturbed solution is then used as the starting point of the next iteration of the *localsearch.* "Adding perturbation helps retain a part of the structure and at the same time, makes it possible to escape from a local minimum towards an even better solution." (Lourenco, Martin

and Stützle, 2001). This process continues until an acceptance criterion is met, for example if the solutions no longer return a better Stress-1 after a certain number of iterations, or the maximum number of iterations is passed.

Compared to multiple random starts and classical scaling, ILS is flexible, for example: different perturbation methods, perturbation strength and perturbation size can be used and tuned depending on the chosen *localsearch*, and the dimensionality and size of the desired solution. ILS has already been applied successfully for combinatorial optimization problems such as the Traveling Salesman Problem, Vehicle Routing Problem and Aircraft Landing Problem (Kramer, 2009; Lourenco, Martin and Stützle, 2001; Cuervo, Goos, Sörensen, Arráiz, 2014). However, so far MDS has not been used as the *localsearch* in an ILS. This thesis will implement ILS with MDS as the *localsearch* and tune the three earlier mentioned parameters: Perturbation method, perturbation strength and perturbation size, with the aim to find a better local minimum compared to classical scaling and finding a better local or global minimum faster than multiple random starts.

ILS performs best when the perturbation method takes into account properties of the problem and are well matched to the *localsearch*. An example: for the Traveling Salesman Problem the *double-bridge move* performs well in multiple studies (Martin, Otto & Felten, 1992; Lin, Sun & Salous, 2016). This thesis looks into the performances of different perturbation methods for MDS with the aim to be able to make suggestions for as a good starting point for further research. The first perturbation method is random perturbation (RP) and is based on mutation in genetic algorithms as stated in Haupt & Haupt (2004): add a random value from a normal distribution to the coordinates for a number of objects. Random perturbation will be compared to two newly developed perturbation methods based on *k* nearest neighbours *(k*-NN) perturbation. In *k*-NN perturbation, the initial solution is perturbed by switching a number of objects with one of its *k* nearest neighbours. This keeps the structure of the initial solution intact as the perturbed solution keeps the same coordinates, with only the objects allocated differently. The small changes in the position of objects can help the next iteration of MDS to escape the current local minimum and continue finding even better local minima in the vicinity of the solution space. Also, if the *localsearch* gets stuck in a local minimum because two objects were switched, *k*-NN can help to jump out of this local minimum by switching these objects into opposite directions. The first *k*-NN perturbation method will use the observed data to find the *k*-NN for each object, while the second method will use the distances of the current best solution to find the *k*-NN. In the next section, all algorithms and parameters will be elaborated on further.

This thesis studies if ILS with MDS as *localsearch* can find solutions with lower Stress-1 values, faster, compared to the earlier mentioned methods (MR and CS). Also, the applicability of *k*-NN perturbation will be compared to random perturbation. This thesis will consist of two parts: a simulation study and a section with applications using real life examples. The simulation study will be used to tune perturbation strength and size for every perturbation method on different sized unidimensional and two-dimensional datasets. Secondly, ILS with the tuned perturbation methods will be compared to multiple random starts to determine the usefulness of ILS. In the second part, well-known real-life datasets will be used to compare ILS, using the tuned perturbation methods, with MDS using classical scaling and multiple random starts. Lastly, the results will be summarized, recommendations for further research and limitations of the current study will be stated in the discussion section.

2. Algorithms

Multidimensional scaling by majorization

Multidimensional scaling (MDS) input data is typically an n x n matrix Δ of proximities. Δ is symmetric, non-negative, has a zero diagonal and is filled with dissimilarities (δ_{ij}). This thesis focusses on ratio MDS of which the aim is to minimize Stress-1 (2) by finding an n x p matrix $\mathbf{X}$ of coordinates of which the Euclidean distances between object i and objects j corresponds as much as possible with the proximities δ_{ij}. A well-known optimization algorithm for MDS is Scaling by MAjorizing a COmplicated Function (SMACOF) (de Leeuw, 1977). This thesis will use SMACOF as *localsearch* because of its simplicity and the guarantee of monotonically converging of Stress. The next paragraph will give a short overview of SMACOF, more detailed information on SMACOF can be found in de Leeuw & Mair (2009).

SMACOF iteratively adjusts the positions of the objects in lower-dimensional space with the goal to minimize the pairwise dissimilarities in the original space and the pairwise distances in the lower-dimensional space. The adjustment of the positions of the objects is done by optimizing a complicated function using majorization. Majorization is done by first finding a simpler function that majorizes the complicated function, and then iteratively minimizes the simpler function. The resulting majorization can then be solved using the Guttmann transformation (Gutmann, 1968) and outputs an updated solution. Raw Stress for the new solution is then compared with the raw Stress of the current solution. SMACOF stops

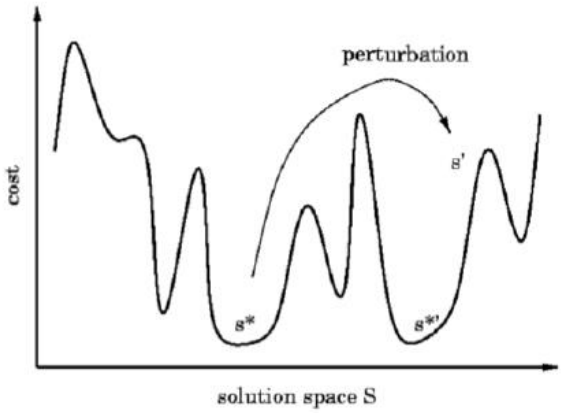
its iterative majorization when the difference between the current raw Stress and the new raw Stress is lower than a user defined acceptance criterion, when raw Stress is equal to zero or when it has reached its maximum number of iterations.

2.2. Iterated Local Search

For this thesis, ILS will be explained in context of MDS with SMACOF as *localsearch*. ILS starts with an initial solution (**X***) found using SMACOF using a random start. In the perturbation phase, **X*** is perturbed, this perturbed solution will be called **X***'* (see figure 1). In the *localsearch* phase, SMACOF is run using **X'** as the initial configuration. The resulting solution (**X*'**) is compared with **X*** on Stress-1 during the acceptance phase. If Stress-1 of **X*'**, is lower than the Stress-1 of **X***, then **X*'** will become the new best solution **X***. In the next perturbation phase, perturbation is added to **X*** and the perturbed solution will be used as initial configuration in the next *localsearch* phase. This loop continues until the maximum number of iterations is reached or an acceptance criterion is met: the solution no longer improves, or a known global minimum is found. The pseudocode of the used ILS is found in Figure 2. The perturbation methods used in the perturbation phase will be discussed further in the next section.

Figure 1

Perturbation visualized.



Note: the figure uses the more general s*, s'* and s*'* instead of **X***, **X'** and **X*'** which is used in context of MDS.

From: "A beginner's introduction to iterated local search." by H. R. Lourenço, O. Martin and T. Stützle, 2001, *Proceedings of MIC, Volume* 4, pp. 1-6).

Figure 2. *Pseudo code of an ILS using SMACOF as localsearch.*

Input: dissimilarity matrix (**Δ**), perturbation strength parameter (*PS*), perturbation size parameter *(PSi*), acceptance criterion (*ε*), maximum number of ILS iterations (*MaxIter*), known best local minimum (*global*), maximum number *localsearch* iterations (*Maxlocal*)

1. Set maxIter; Iter = 0;
2. **X*** ← generate initial solution using SMACOF with random initial configuration using **Δ** as input, acceptance criterion *ε* and maximum number of iterations *Maxlocal*.
3. While (iter < *MaxIter*) do
4. | **Perturbation phase**

   X' ← Add perturbation to **X*** using RP/*k*-NN perturbation with *PS* and *PSi*

5. | ***Localsearch* phase**

   X*' ← SMACOF using **X'** as initial configuration acceptance criterion *ε* and maximum number of iterations *Maxlocal.*

6. | **Acceptance phase**

   if: Stress-1$_{\mathbf{X}*'}$ < Stress-1$_{\mathbf{X}*}$

   X* ← **X*'**

7. | **break** if Stress-1$_{\mathbf{X}*}$ ≤ *global*
8. iter = iter + 1
9. End while
10. Return **X*** and Stress-1$_{\mathbf{X}*}$.

2.2.1 Perturbation

"Perturbation is defined as making changes in the sequence of objects by adding error to the object's coordinates." (Wagenaar & Padmos, 1971). The added perturbance may cause objects to switch places in the solution space if the perturbance is strong enough and in 'opposite' directions. The *localsearch* will continue with finding a new solution using the perturbed solution as starting point. Some important factors of perturbation are perturbation strength (PS), perturbation size (PSi) and perturbation method (PM).

How much the perturbed solution keeps the same structure as the initial solution depends on PS: how much perturbation is added to an object, and PSi: the percentage of objects that is perturbed. If the perturbation strength is too weak, the next resulting solution

will resemble the previous solution too much. This results in the next iteration of the *localsearch* to return to the previous local minimum. If the perturbation strength is too strong, the perturbed solution will no longer resemble the previous solution at all. This causes the next iteration of the *localsearch* to jump towards a different local minimum instead of exploring the neighbourhood of the current local minimum for better local minima, which is similar to using multiple random starts. Similarly, If the perturbations size is low, for example only one objects is perturbed, then the perturbed solution keeps the structure of the current solution, bar the single object. This could result in not enough difference for the next *localsearch* to escape the current local minimum. Conversely, if all objects are perturbed, the resulting perturbed solution can differ in structure completely and will, again, become similar to multiple random starts.

How perturbation strength is defined, differs between the proposed perturbation methods. Random perturbation intends to switch objects given opposite direction and a proper size for the perturbation. The perturbation is a randomly picked value from a normal distribution, which is added to the coordinates (x_{ij}) of a number of objects to achieve a perturbed coordinate (x'_{ij}) (3) (Haupt & Haupt, 2004):

$$x'_{ij} = x_{ij} + N(0, \theta\sigma_{X*}) \quad (3)$$

where

σ_{X*} = standard deviation of the current best solution **X***.

θ = factor used to increase or decrease perturbation strength.

For random perturbation, the perturbation strength is defined by the θ factor. Increasing θ increases the standard deviation of the normal distribution, resulting in a wider range from which values can be chosen.

2.3. *k*-NN perturbation

In this thesis, a new method of perturbation has been developed: *k*-NN perturbation. A number of objects will be switched with one of its *k* nearest neighbours. In *k*-NN perturbation, perturbation strength is defined as the size of *k:* with a higher *k*, the chosen objects have the chance of switching with a more distant object, resulting in a more strongly perturbed solution. Compared to random perturbation, *k*-NN perturbation has a couple of benefits. The first benefit is a higher chance of switching outliers: For random perturbation it is possible that the added of the perturbation is not enough for the outlier to jump the gap between it and the rest of the objects, or if the added perturbation moves objects in the same

direction on the solution space. Whereas, if the outlier is selected for *k*-NN perturbation, it will always switch with one of its *k* nearest neighbours even if these objects are quite distant. The second benefit is that *k*-NN perturbation will always make a switch. Random perturbation does not guarantee a switch if the perturbation is not strong enough or, again, in the same direction as the nearest neighbour.

In this study two different methods of determining *k* nearest neighbours for every object are applied:

1. *k-NN$_{\Delta}$: k*-NN perturbation using the observed data. Determine the *k* nearest neighbours for a number of objects by using the dissimilarities found in the observed data (**Δ**). Randomly switch the coordinates of one of the *k* nearest neighbours with the chosen object to perturb.
2. *k-NN$_{X}$: k*-NN perturbation using the coordinate matrix (**X**) of the current best solution. Order the Euclidean distances of **X**, to determine the *k* nearest neighbours for a number of objects. Randomly switch the coordinates of one of the *k* nearest neighbours with the chosen object to perturb.

The difference lies between when and how many times the *k*-nearest neighbours are determined: in the first method, this is done every time ILS reaches the perturbation phase. In the second method, this is done only once at the start of the ILS algorithm. The first method has the benefit of using the structure of the observed data as the starting point of the perturbation. The second method makes changes using the structure of the current best solution, which might help the next iteration of the *localsearch* to keep exploring better local minima in the vicinity of the current best local minimum.

3. Simulation study

3.1. Methods

The effectiveness and applicability of ILS with the three different perturbation methods will be studied using a Monte Carlo simulation study. The first goal is to find the best combination of perturbation strength and perturbation size for every perturbation method for different sized unidimensional and two-dimensional datasets. Then, a comparison will be made between ILS with the three tuned perturbation methods, MR will be used as baseline to determine usefulness.

In Table 1, an overview of the used factors and factor levels are presented. Perturbation strength for ILS with both *k*-NN perturbation methods equates to the number of

k nearest neighbours an object can switch with. In the simulation study *k* was varied between 5%, 10%, 25%, 50% of the total number of objects, with a minimum of $k = 1$. For ILS with random perturbation, perturbation strength will be determined by the factor θ, which will be varied between .05, .25, 1 and 2. Perturbation size is defined by the percentage of objects that will be perturbed. In this simulation study, 10%, 25%, 50% and 100% of the objects will be perturbed. The last two factors are dataset size (*n*) and the dimensionality of the dataset (*p*).

All simulations were done in R (R Core Team, 2020). The MDS function from the smacof package (v2.1-5: P. Mair, J. De Leeuw, P. J. F. Groenen & Ingwer Borg, 2022) will be used as the *localsearch* in ILS and to perform multiple random starts. MDS will be used with the ratio MDS-type, a maximum number of iterations of 500 and an acceptance criterion of ε equalling $1*10^{-12}$. All three ILS procedures will be run one hundred times for every combination of PS, PSi, *n* and *p*, with a maximum number of iterations of 500. For the baseline, multiple random starts, the MDS function will be run five hundred times using a random initial configuration for all dataset sizes and dataset dimensionalities. Again, the MDS function will have a maximum number of iterations of 500.

Table 1

Overview of simulation study factors.

Factor	Levels
Procedures	*k-NN$_X$, k-NN$_\Delta$, RP, MR*
PS	k = 5%, 10%, 25%, 50%
	θ = .05, .25, 1, 2
PSi	10%, 25%, 50%, 100%
n	20, 50, 100
p	1, 2

3.1.2 Data generation

Six different datasets will used: three two-dimensional and three one-dimensional datasets. A coordinate matrix (**X**) is generated with random coordinates picked from a normal distribution, $N(0,1)$. The Euclidean distance matrix of **X** will be used as the observed dissimilarity data (**Δ**). The generated data has zero error and three different sizes ($n = 20$, $n =$

50, n = 100). Findings of Smit (2018) found no increasing problems with local minima at a higher number of objects, therefor 100 is chosen as the maximum number of objects.

3.1.3 Outcome variables

An overview of all outcome variables is available in Table 2. The first and main outcome variable is the ability to find the global minimum (*Global*). With zero error and an acceptance criterion of $\varepsilon = 1 \bullet 10^{-12}$, a solution with Stress-1 $< 1 \bullet 10^{-6}$ will be classified as the global minimum. The percentage of times a method can find the global minimum (*%Global*) will be used to determine the consistency of each method. Minimum, mean and maximum Stress-1 (*Stress*) values of the best local minima per method are used as the second outcome variable to determine the consistency in finding better solutions. The third outcome variable is the expired runtime at the moment of finding the best local minimum or global minimum. Lastly, #*MDS* is the number of times the ILS performed the *localsearch* phase before finding its best solution. *#MDS* will be used to evaluate the efficiency of ILS compared to multiple random starts.

Table 2

Overview of outcome variables.

Variable	Definition
Global	Whether the global minimum was found (Stress-1 $< 1 \bullet 10^{-6}$) (yes/no).
%Global	Percentage of times the global minimum was found.
Stress	Stress-1 of the best local minimum/global minimum.
Runtime	Mean elapsed time between the start of the procedure and the moment of finding the lowest Stress-1 value of all runs.
#MDS	Average number of times the MDS function was called at the moment of finding the best solution.

3.1.4 Data analysis.

ILS using the three perturbation methods are evaluated on their ability to find the global minimum per combination of the factors PS and PSi. If multiple combinations are able to find the global minimum per dataset size and dimensionality, then the percentage of times the global minimum was found will be used to evaluate the consistency. If procedures are not able to find the global minimum, then the best combination of factors is determined with

Stress as outcome variable. If no significant differences in *Stress* are observed, then *Runtime* as the outcome. Lastly, *#MDS* will be used to determine how efficient a single ILS is compared to multiple random starts when looking at how many times the MDS function was called.

The ILS methods stop when the global minimum is found, therefor it is expected that *Stress* is not normally distributed: *Stress* will mostly be zero if a method is able to find the global minimum in most ILS runs. Therefore, non-parametric tests will be used for data analysis: Friedman tests will be used to study if PS and PSi have a significant influence within the same method on *Stress* and *Runtime*. If significant differences are found, a Wilcoxon signed-rank test will be done to find between which levels this difference is found.

After finding every best combination of factors levels, comparisons between ILS with each perturbation method and multiple random starts will be done using Kruskal-Wallis tests. A Wilcoxon rank sum test will be used as post hoc test to compare each procedure on *Stress* and *Runtime*. All comparisons will be evaluated on significance using $p < .05$.

3.2. Results 1D

3.2.1 Tuning the parameters for each perturbation method.

The found *Stress* per combination of perturbation strength, perturbation size and dataset size are found in figures 3, 4 and 5. In Table 3 the best combination of factors for each perturbation method are available. The other outcome variables are visualised and found in Appendix A and B**.**

ILS using *k-*NN_{Δ} perturbation was able to find the global minimum with every combination of PS and PSi, for every dataset size. When looking at each panel of Figure 3, a clear decrease in local minima is visible when perturbation strength increases. At the highest level of PS, the global minimum is even found every time whilst using less *#MDS*, independent of PSi. Differences in *Runtime* between PS levels were significant in the medium ($H(3) = 80{,}49$; $p < .05$) and the largest dataset ($H(3) = 236{,}22$; $p < .001$). With higher perturbation strength resulting in lower runtime. No significant difference in *Stress* between PSi levels were found.

ILS using *k-*NN_X perturbation was also able to find the global minimum at every level of PS and PSi. Again, the percentage of the runs that found the global minimum increases with higher levels of perturbation strength. This is even more clear in the medium and large

datasets, where wide distributions of *Stress* values were found at *k* equalling 5% and 10%. At *k* equalling 50%, the global minimum was found every time whilst using the lowest *#MDS*, independent of the perturbation size. Significant differences for *Runtime* were found between the levels of PS for all three datasets: smallest dataset ($H(3) = 14{,}48$; $p < .05$), medium dataset ($H(3) = 481{,}07$; $p < .001$) and the largest dataset ($H(3) = 151{,}00$; $p < .001$), with higher levels of PS being lower in runtime. Only in the smallest dataset, a significant difference between PSi levels was found ($H(3) = 4{,}96$; $p < .05$), with higher perturbation size resulting in lower *runtime.*

Lastly, ILS using random perturbation was also able to find the global minimum at every level of PS and PSi. But again, higher levels of perturbation strength resulted in finding the global minimum more often: at the highest level of PS, the global minimum is found every time. Significant differences in *Runtime* were found between levels of PS in every dataset size, with higher perturbation strength resulting in lower *Runtime*. Again, no significant differences in *Stress* between levels of PSi were found. *Runtime* did significantly differ between PSi levels, with higher PSi resulting in lower *Runtime* in the medium ($H(3) = 16{,}48$; $p < .001$) and large dataset *(*$H(3) = 61{,}94$; $p < .001$). Again, the combination of high PSi and PSi resulted in the lowest *#MDS* needed to find the global minimum.

In conclusion, each perturbation method benefits from high perturbation strength, which increases the consistency of finding the global minimum. If the perturbation strength is high, perturbation size shows no influence on the consistency of finding the global minimum. Perturbation size does have an effect when looking at *Runtime*: higher levels of PSi results in lower runtimes needed to find the global minimum. Therefor it is recommended to use a perturbation size of 50% of all objects and a high level of perturbation strength *k* equalling 50% of the total number of objects for both *k*-NN perturbation methods and $\theta = 2$ for random perturbation.

Unidimensional solutions have a bumpier solutions space, which need a stronger perturbation on more objects for the next *localsearch* to escape the current local minima. For example, when the current solution misplaces two objects on completely opposite sides on the solution space, a weak perturbation will not cause these objects to move in opposite direction enough for the next *localsearch* to escape the current local minimum.

Figure 3.

Stress-1 values of unidimensional solutions found by ILS with k-NN_{Δ} perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

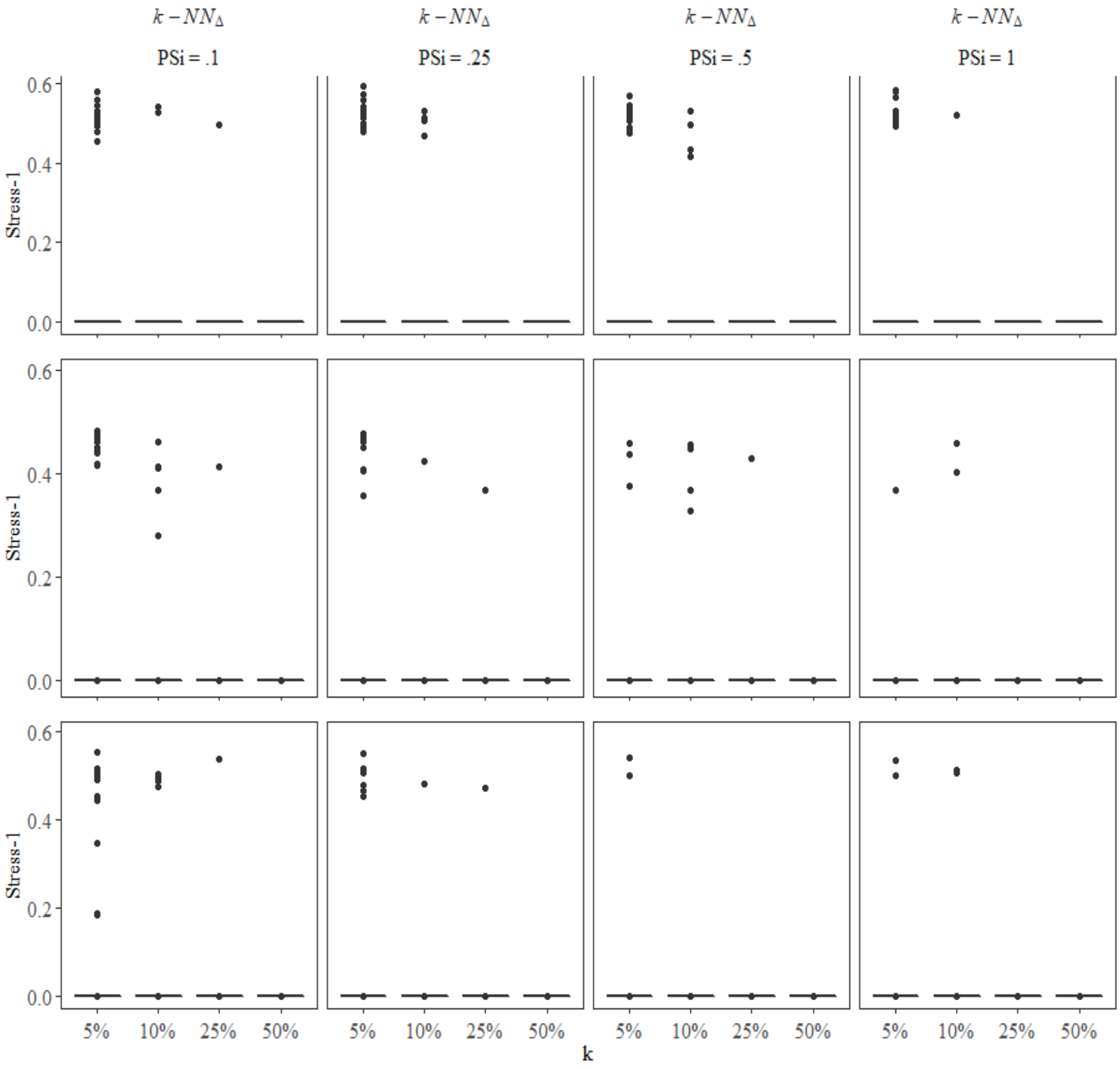


Figure 4.

Stress-1 values of unidimensional solutions found by ILS with k-NN_X perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

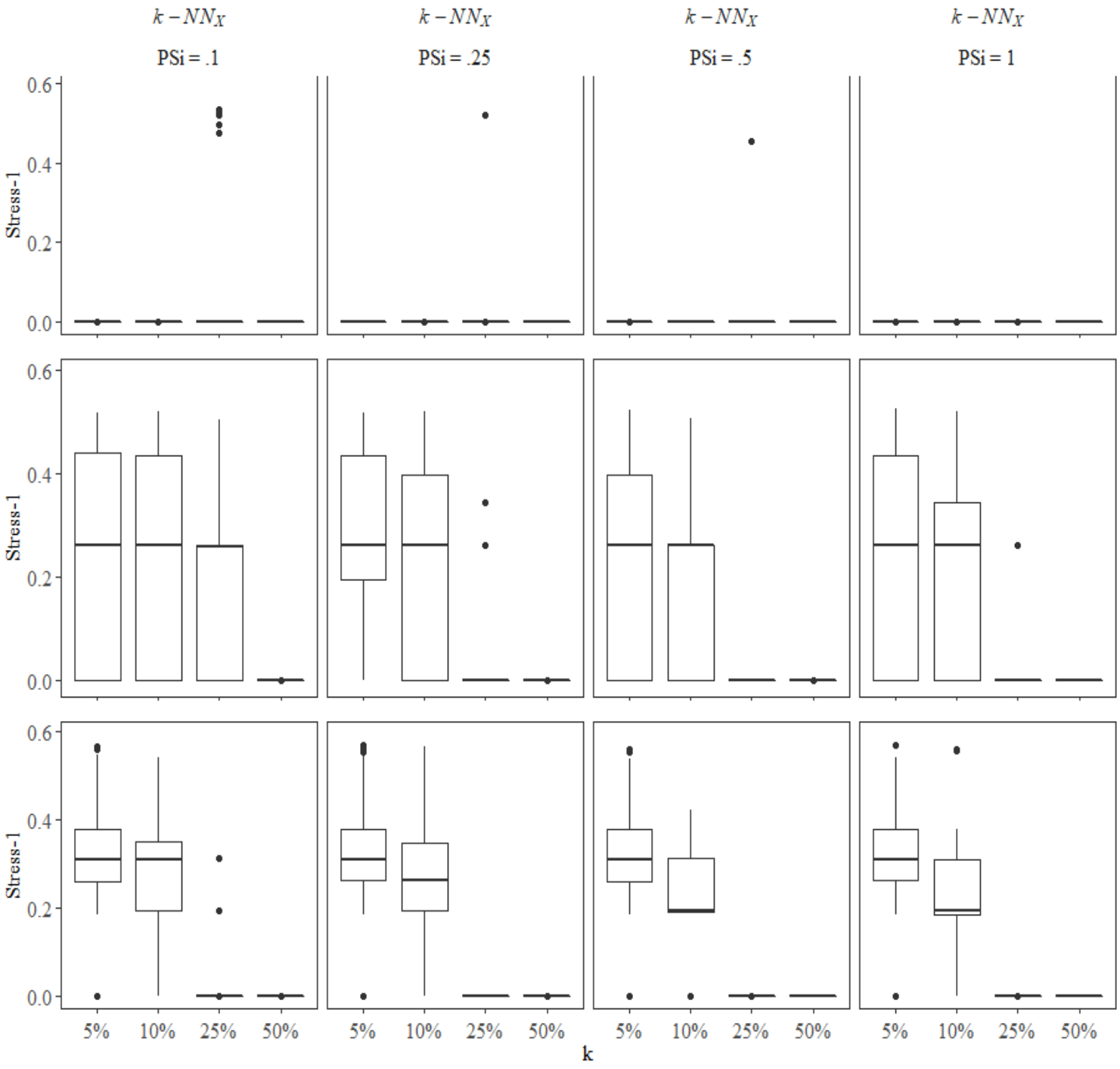
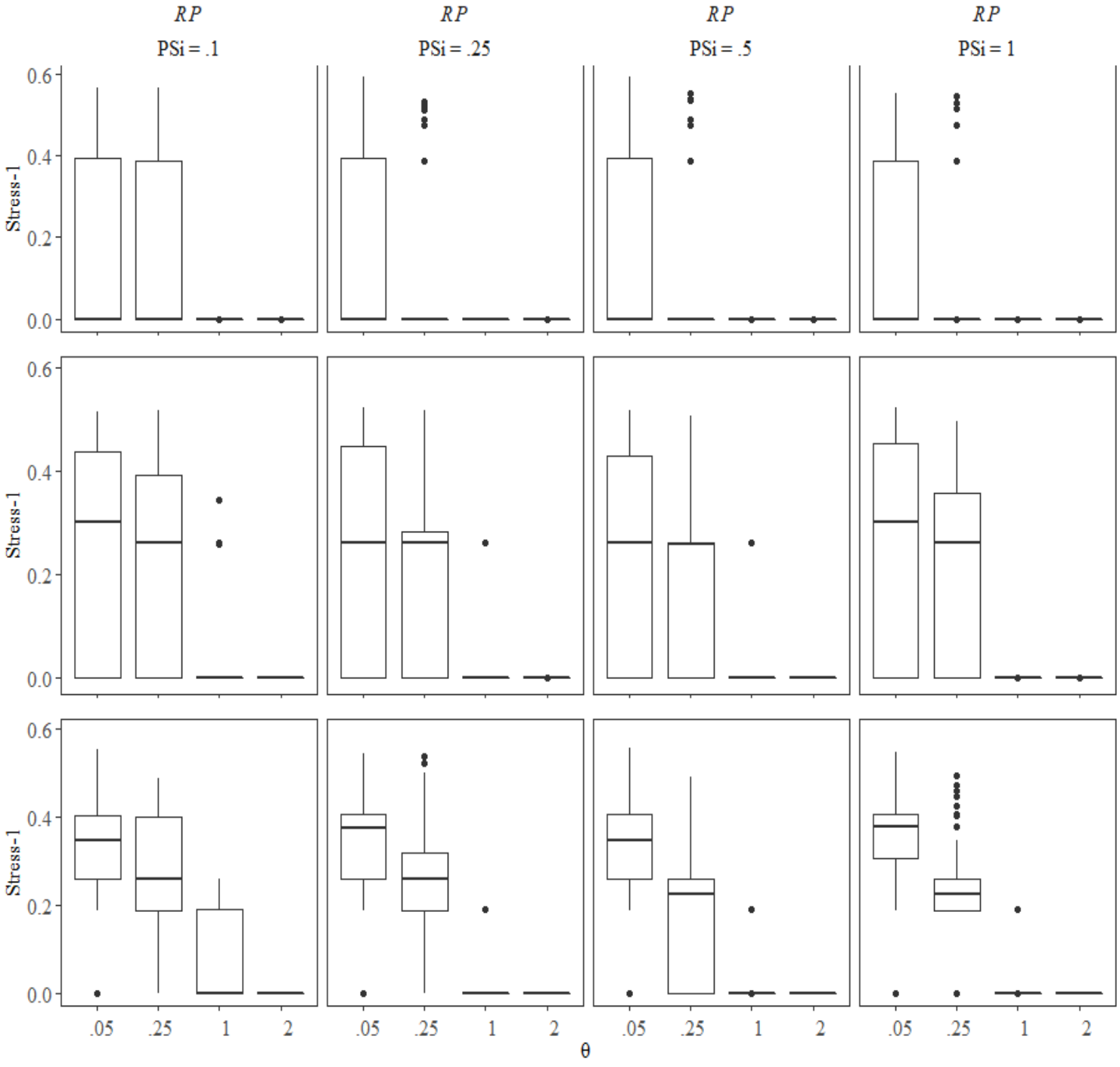


Figure 5.

Stress-1 values of unidimensional solutions found by ILS with random perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.



3.2.2 Comparing ILS with each perturbation method and multiple random starts.

The following PS and PSi levels per perturbation method were used as comparison with multiple random starts on the unidimensional dataset (see Table 3).

Table 3

Overview of the best parameter per dataset size and perturbation method for unidimensional datasets.

Perturbation method	*n*	*PS*	*PSi*	*n*	*PS*	*PSi*	*n*	*PS*	*PSi*
k-NN$_{\Delta}$	20	k = 50%	50%	50	k = 50%	50%	100	k = 50%	50%
k-NN$_{X}$	20	k = 50%	50%	50	k = 50%	50%	100	k = 50%	50%
RP	20	θ = 2	50%	50	θ = 2	50%	100	θ = 2	50%

All results can be found in Table 4. ILS with every perturbation method and multiple random starts were able to find the global minimum. However, MR was much less consistent: finding the global minimum in a lower percentage of its run. The ILS procedures were able to find the global minimum every time, therefor *Runtime* will be used as outcome variable to compare which ILS procedure found the global minimum the fastest. A significant difference was found between the procedures in the small *(*H(2) = 40,59; p < .001), the medium *(*H(2) = 13.08; p < .01) and the large dataset (H(2) = 21,67; p < .001). ILS with *k-NN*$_{\Delta}$ and *k-NN*$_{X}$ perturbation only showed significant differences in *Runtime* in the small dataset (U = 2870,00; p < .001), with *k-NN*$_{X}$ perturbation having lower *Runtime*. RP is faster in the medium dataset than *k-NN*$_{X}$ (U = 3818,00; p < .01) and *k-NN*$_{\Delta}$ (U = 3890,00; p < .01). In the other datasets, both *k*-NN ILS procedures were faster in finding the global minimum.

In conclusion: ILS with the three different perturbation methods is better in finding the global minimum compared to multiple random starts. No clear difference between each perturbation method were found on runtime: random perturbation was faster in the medium dataset, but both *k*-NN perturbation methods were faster in the small and large dataset. Also, no clear difference between the two k-NN perturbation methods was found. Using the current best solution to find the *k*-nearest neighbours, was faster in the small dataset, but in the medium and large dataset, both *k*-NN perturbation methods performed similarly on runtime. The number of times the ILS needs to call the MDS function during the *localsearch* phase also indicates that ILS is more efficient: needing on average thirteen or less *localsearch*

phases compared to multiple random which runs MDS 500 times. All three ILS procedures found the global minimum and were more consistent than MR, indicating that ILS is effective for unidimensional datasets, without error.

Table 4

Summary of the results of ILS with k-NN_{Δ} perturbation, ILS with k-NN_X perturbation and ILS with RP using tuned parameters, with MR as baseline for unidimensional solutions.

Procedure	n	Min Stress	Max Stress	Mean Stress	Runtime	SD runtime	Global	%Global	#MDS
MR	20	0	.6820	.3827	.0028	.010	Yes	28,4%	-
k-NN_{Δ}	20	0	0	0	.0098	.007	Yes	100%	7
k-NN_X	20	0	0	0	.0072	.006	Yes	100%	7
RP	20	0	0	0	.0069	.011	Yes	100%	8
MR	50	0	.6260	.4068	.0084	.002	Yes	10,2%	-
k-NN_{Δ}	50	0	0	0	.0273	.011	Yes	100%	6
k-NN_X	50	0	0	0	.0284	.025	Yes	100%	7
RP	50	0	0	0	.0666	.076	Yes	100%	13
MR	100	0	.6038	.4396	.0263	.016	Yes	1,2%	-
k-NN_{Δ}	100	0	0	0	.0889	.028	Yes	100%	5
k-NN_X	100	0	0	0	.0122	.055	Yes	100%	7
RP	100	0	0	0	.1877	.188	Yes	100%	13

Note: MR was run 500 times, ILS was run 100 times per perturbation method.

3.3. Results 2D

3.3.1. Tuning the parameters for each perturbation method.

All Stress values found by ILS with *k-NN_{Δ}*, *k-NN_X* and random perturbation are found in figures 6**,** 7 and 8. The best combination of perturbation strength and perturbation size can be found in Table 5. Runtime and #MDS are visualised in appendices A and B.

A clear pattern is visible for ILS with each perturbation method when looking at figures 6**,** 7 and 8 from left to right**:** *Stress* increases with higher perturbation size, independent of perturbation strength. ILS with *k-NN_{Δ}* perturbation is able to find the global

minimum for every run in every dataset size while using a combination of low perturbation strength and perturbation size. The effects of PS on *Stress* were significant for every dataset: small ($H(3) = 17{,}60$; $p < .05$), medium ($H(3) = 146{,}01$; $p < .001$), and large ($H(3) = 112.27$; $p < .001$), increasing *Stress* at higher perturbation strength. Similarly, *Runtime* and *#MDS* increases with higher PS and PSi.

Similar findings were present for ILS with *k-NN*$_X$ perturbation: the lowest level of PSi is the most consistent in finding the global minimum (left column in Figure 7). At the lowest perturbation size, *k-NN*$_X$ performs best using the lowest perturbation strength in the medium and large dataset. In contrary, in the small dataset a PS equalling 10% performed significantly better than PS equalling 5% on both *Stress* ($U = 53.00$, $p < .001$) *Runtime* ($U = 49445{,}00$; $p < .001$).

Again, random perturbation also found lower *Stress* values, faster at the lowest level of PS and PSi in every dataset. ILS RP is even only able to find the global minimum in the medium dataset with the lowest levels of PS and PSi. A higher *#MDS* is needed in every dataset than ILS using both *k*-NN perturbation methods.

Concluding, for the two-dimensional datasets, ILS procedures were able to find the global minimum in every dataset size. Contrary to the results for unidimensional solutions, lower perturbation strength and size resulted finding the global minimum more consistently for two-dimensional solutions. Also, lower levels of PS and PSi resulted in lower mean Stress-1 and runtime needed to find the best local minimum/global minimum. Therefore, it is recommended to use k-NN perturbation with *k* equalling 5% of the total number of objects and random perturbation with a θ of .05. Apply this perturbation strength on 10% of all objects.

The weaker perturbation needed can be explained by the increase in flexibility of the objects to move around in a two-dimensional space, compared to a unidimensional space. This results in *localsearch* having an easier time to explore the space around of the current local minimum, without dropping back into the current local minimum. If the PS and PSi increases, each next *localsearch* has to start over in a completely new neighbourhood in the solution space instead of finding better local minima near the current local minimum.

Figure 6.

*Stress-1 values of two-dimensional solutions found by ILS with k-NN_Δ perturbation at different levels of PS. In the columns rom left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100. The dashed line indicates the global minimum of Stress-1 equalling 1*10^{-6}.*

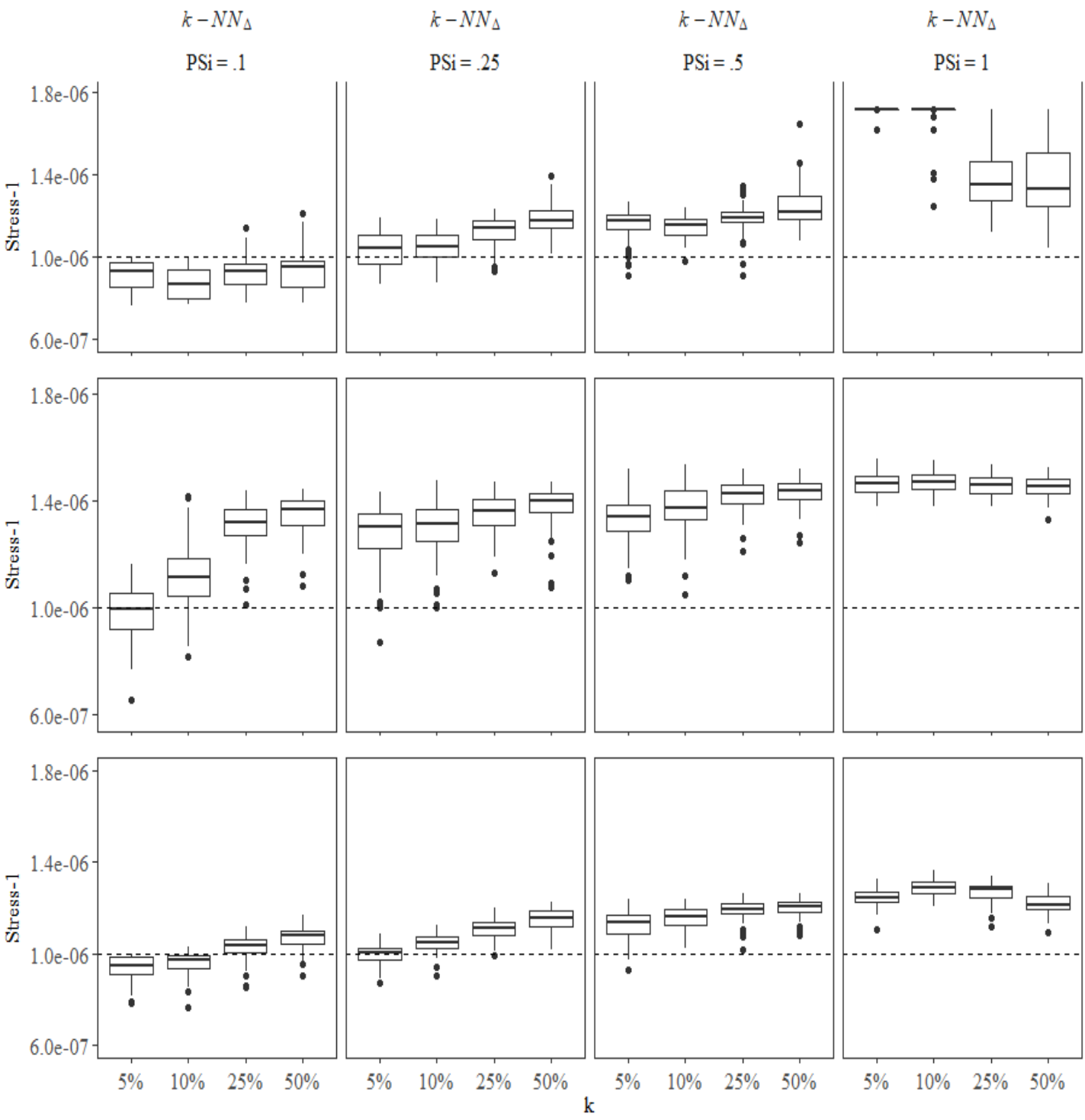


Figure 7.

Stress-1 values of two-dimensional solutions found by ILS with k-NN_X perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100. The dashed line indicates the global minimum of Stress-1 equalling $1*10^{-6}$.

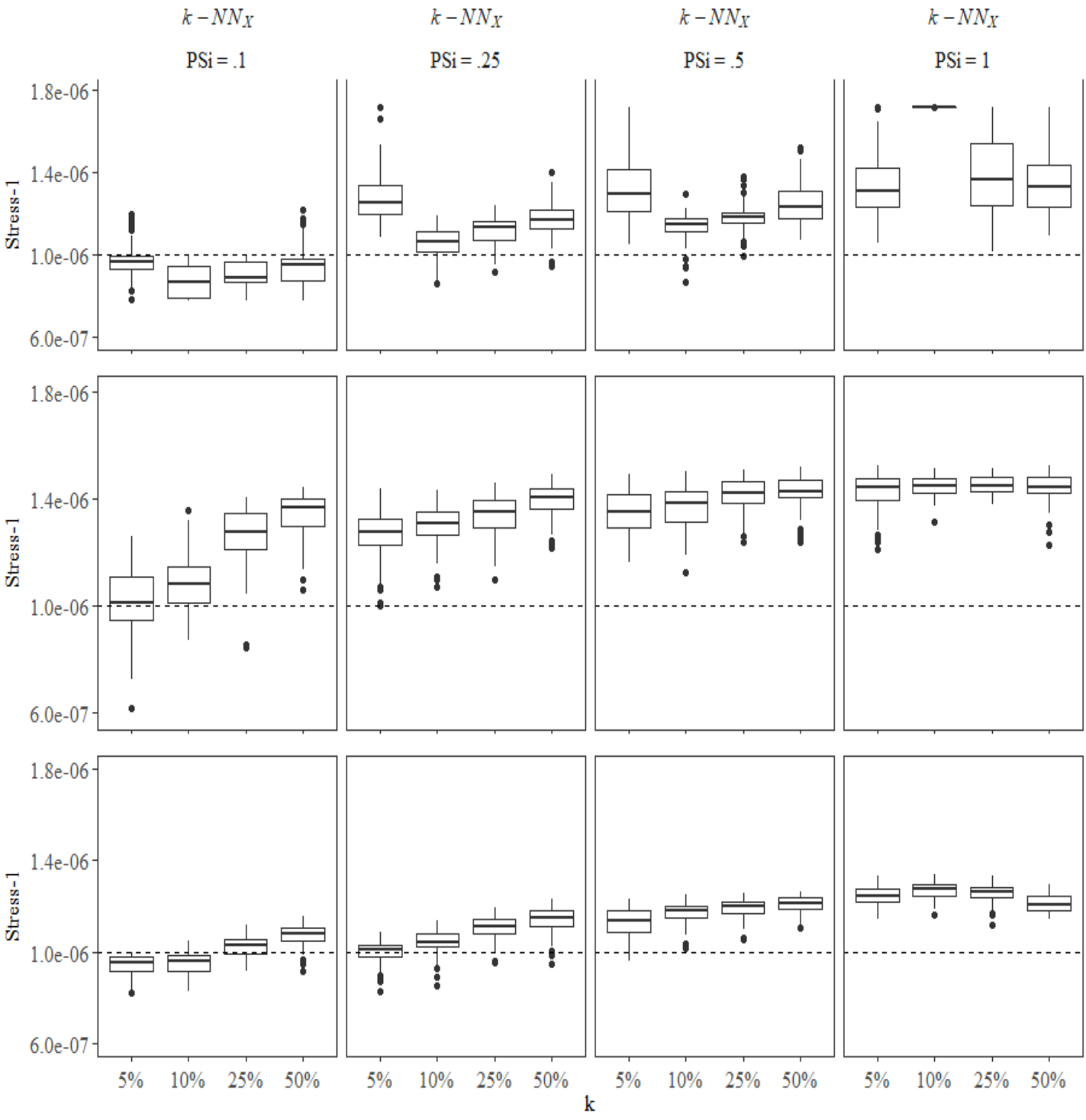
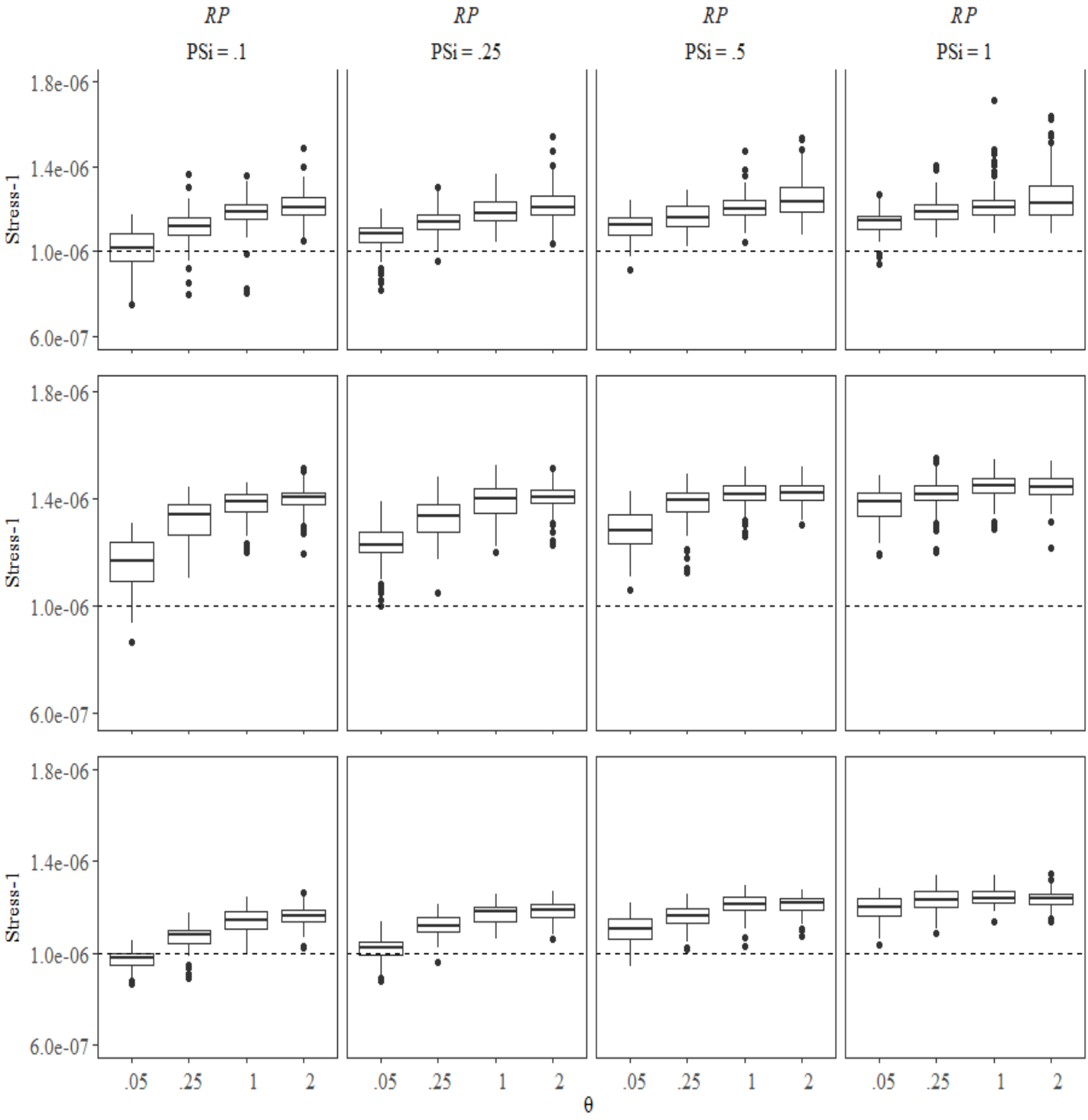


Figure 8

*Stress-1 values of two-dimensional solutions found by ILS with random perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100. The dashed line indicates the global minimum of Stress-1 equalling 1*10^{-6}.*



3.2.4. Comparing ILS with each perturbation method and multiple random starts

The following PS and PSi levels per procedure were used as comparison with MR on the two-dimensional dataset (see Table 5).

Table 5

Overview of the best parameter per dataset size and procedure for p = 2.

Procedure	*n*	*PS*	*PSi*	*n*	*PS*	*PSi*	*n*	*PS*	*PSi*
k-NN$_{\Delta}$	20	*k* = 5%	10%	50	*k* = 5%	10%	100	*k* = 5%	10%
k-NN$_{X}$	20	*k* = 10%	10%	50	*k* = 5%	10%	100	*k* = 5%	10%
RP	20	θ = .05	10%	50	θ = .05	10%	100	θ = .05	10%

The results of the comparison between ILS with the different perturbation methods and MR are available in Table 6.

Even though ILS was able to find the global minimum with each perturbation method, random perturbation was less consistent in finding the global minimum compared to the two *k*-NN perturbation methods in every dataset size. No clear difference was found on *runtime* between each perturbation method: both *k*-NN perturbation methods were faster than random perturbation in the small ($H(2) = 153{,}59$; $p < .001$) and large dataset ($H(2) = 67{,}25$; $p < .001$), whereas random perturbation was faster in the medium dataset compared to both *k*-NN perturbation methods ($H(2) = 9{,}49$; $p < .01$). ILS using random perturbation had to call the MDS function the most times before finding its best solution compared to *k-NN*$_{\Delta}$ and *k-NN*$_{X}$ perturbation. Indicating that the random nature of RP hinders it from perturbing the chosen objects enough for the next *localsearch* phase to escape the current local minimum. Both *k*-NN perturbation methods needed to call the MDS function a similar number of times before finding the global minimum. Multiple random starts was less consistent in finding the global minimum compared to ILS with the three different perturbation methods. Also, multiple random got stuck in worse local minima resulting in a higher maximum *Stress* compared to ILS with each perturbation method. This leads MR to find higher mean *Stress* than the three ILS procedures.

These results indicate that ILS using random and *k*-NN perturbation are valid methods for finding MDS solutions in normally distributed, simulated two-dimensional datasets without error.

Table 6.

Summary of the results of ILS with k-NN_{Δ} perturbation, k-NN_X perturbation and random perturbation using tuned parameters, with MR as baseline for two-dimensional datasets.

Procedure	n	Mean Stress	Min Stress	Max Stress	Runtime	SD runtime	Global	%Global	#MDS
MR	20	.017209	$8.5*10^{-7}$	.1905	.0028	.010	Yes	12,8%	-
k-NN_{Δ}	20	.000001	$8.7*10^{-7}$	$9.9*10^{-7}$	.4787	.356	Yes	100%	54
k-NN_X	20	.000001	$7.7*10^{-7}$	$9.9*10^{-7}$	.7556	1.72	Yes	100%	85
RP	20	.000001	$7.5*10^{-7}$	$1.2*10^{-6}$	4.893	2.69	Yes	44%	265
MR	50	.004940	$9.2*10^{-7}$	.2534	.0084	.019	Yes	2,4%	-
k-NN_{Δ}	50	.000001	$6.5*10^{-7}$	$1.2*10^{-6}$	3.902	2.29	Yes	55%	238
k-NN_X	50	.000001	$6.2*10^{-7}$	$1.3*10^{-6}$	4.014	2.41	Yes	48%	246
RP	50	.000001	$8.7*10^{-7}$	$1.3*10^{-6}$	3.024	1.74	Yes	6%	240
MR	100	.001502	$9.1*10^{-7}$	.2778	.0263	.040	No	1,6%	-
k-NN_{Δ}	100	.000001	$9.4*10^{-7}$	$9.9*10^{-7}$	2.174	1.82	Yes	100%	52
k-NN_X	100	.000001	$8.3*10^{-7}$	$9.9*10^{-7}$	2.192	2.23	Yes	100%	57
RP	100	.000001	$8.7*10^{-7}$	$1.1*10^{-6}$	6.563	4.46	Yes	82**%**	216

Note: MR was run 500 times, ILS was run 100 times per perturbation method.

4. Real life applications

In the simulation study, the best combinations of perturbation parameters for *k-NN_{Δ}*, *k-NN_X* and random perturbation were found for both unidimensional and two-dimensional datasets. In the following section, the effectiveness and consistency of ILS with the three tuned perturbation methods, will be compared with two commonly used MDS methods: classical scaling (CS) and multiple random starts. Four real-life small (≤ 20 objects) and medium (≤ 50 objects) sized datasets will be used as examples. The medium sized Morse code dataset (Rothkopf, 1957) and the Food dataset (Murphy & Ross, 1999) will be used for both unidimensional and two-dimensional MDS. The small Churchill dataset (Churchill, 1995) will be used for MDS in two dimensions. Lastly, the small Kabah dataset (Robinson, 1951) will be used for MDS in one dimension.

Effectiveness is defined as the ability of ILS with the different perturbation methods to find solutions with lower Stress-1 than MDS using classical scaling. Also, the percentage of runs in which a solution with a lower Stress-1 values than CS (%Stress-$1_{<CS}$) is found, will be used as a measure of consistency. Lastly, the number of unique local minima found per procedure is also reported as a measure of consistency. Multiple random starts will be used as comparison for the performances of ILS with the three perturbation methods. If the best local minimum is known for each dataset, this minimum will also be used for comparison with the ILS procedures.

As in the simulation study, the MDS function from the smacof package (v2.1-5: P. Mair, J. De Leeuw, P. J. F. Groenen & Ingwer Borg, 2022) will be used as *localsearch* in ILS and to perform multiple random starts and classical scaling. ILS will be run 100 times per perturbation method, using the PS and PSi values in Table 7. ILS will use MDS with a random start to achieve an initial solution. MDS as *localsearch* in the three ILS procedures will have a maximum number of iterations of 500. ILS will have a maximum of 500 iterations. For multiple random starts, the MDS function will be run 500 times, also with a maximum number of iterations of 500. An acceptance criterion of ε equalling $1*10^{-12}$ will be used for all MDS functions.

Table 7.

Overview of tuned perturbation parameters for k-NN_{Δ}, k-NN_X and random perturbation per dataset dimensionality and size.

Perturbation method	p	n	PS	PSi	p	n	PS	PSi
k-NN_{Δ}	1	Small	k = 50%	50%	1	Medium	k = 50%	50%
k-NN_X	1	Small	k = 50%	50%	1	Medium	k = 50%	50%
RP	1	Small	θ = 2	50%	1	Medium	θ = 2	50%
k-NN_{Δ}	2	Small	k = 5%	10%	2	Medium	k = 5%	10%
k-NN_X	2	Small	k = 10%	10%	2	Medium	k = 5%	10%
RP	2	Small	θ = .05	10%	2	Medium	θ = .05	10%

4.2 Morse dataset

The Morse code dataset (Rothkopf, 1957) is a 36 x 36 confusion matrix resulting from an experiment in which subjects were asked whether pairs of Morse codes were identical. The Morse codes are the 26 letters of the Latin alphabet and the numbers zero to nine. Stress-1

values for unidimensional and two-dimensional solutions have been found in earlier research. Palubeckis (2013) and Brusco, Köhn and Stahl (2008) found similar Stress-1 for unidimensional solutions, being .4798. However, both studies required high computational times. Buja and Swayne (2002) found a two-dimensional solution with the lowest Stress-1 equalling .2836.

A summary of the results is available in Table 8. For unidimensional solutions, ILS with every perturbation method was able to find lower values than classical scaling. ILS with *k-NN*$_X$ perturbation performed most consistent: finding lower Stress-1 values than CS in 93% of all runs. ILS with random perturbation was able to find a lower Stress-1 than CS in 84% and ILS with *k-NN*$_\Delta$ perturbation in 78%. MR performed worse than each ILS: it was not able to find lower Stress-1 values than classical scaling, with the lowest being .52945.

For two-dimensional solutions, ILS with every perturbation method found the same lowest Stress-1 (.29992). ILS using *k-NN*$_\Delta$ perturbation was the most consistent: finding its lowest Stress-1 value in each of its one hundred runs. ILS using *k-NN*$_X$ perturbation found this value in 55% of the runs. In the other runs, it got stuck in nine different local minima. ILS using random perturbation was the least consistent of every ILS: it found the lowest Stress-1 in only 4% of all runs. It got stuck in even more local minima: seventy-nine different local minima. Still, every ILS performed better than multiple random starts. MR was the least consistent, finding 409 different local minima of its 500 runs and found the same Stress-1 value as classical scaling in only 1% of its runs.

Table 8

Summary of outcome variables for unidimensional and two-dimensional solutions for ILS with k-NN_{Δ}, k-NN_X and random perturbation using tuned parameters compared to classical scaling (CS) and multiple random starts (MR) for the Morse dataset.

Procedure	p	Lowest Stress-1	%Stress-$1_{<CS}$	Number of local minima
Palubeckis (2013)	1	**.47980**	-	-
CS	1	**.50139**	-	-
k-NN_{Δ}	1	.48393	78%	100
k-NN_X	1	.48216	93%	99
RP	1	.48476	84%	100
MR	1	.52945	0%	499
Buja and Swayne (2002)	2	**.28360**	-	-
CS	2	**.29992**	-	-
k-NN_{Δ}	2	.29992	100%	1
k-NN_X	2	.29992	55%	9
RP	2	.29992	4%	80
MR	2	.29992	1%	409

Note: MR was run 500 times, ILS was run 100 times per perturbation method.

4.2.2 Food dataset

The Food dataset originates from a study from Ross and Murphy (1999) in which 94 undergraduates categorized food items without restrictions on the number of possible groups. The resulting dataset was made into a dissimilarity matrix by Hubert, Arabie and Meulman (2001), in which the dissimilarities represent the percentage of subjects who did not place the same food item in the same group. Different numbers of categorization are possible: Palubeckis (2013) performed unidimensional scaling, and Köhn, Steinley and Brusco (2010) used *p*-median clustering to categorize all items in two groups. Palubeckis (2013) had to use a subset of 35 objects from the food data, as the complete dataset too computationally heavy. No known best local minima are yet published for both unidimensional and two-dimensional solutions of the complete dataset.

The results of the comparison between ILS with each perturbation method, classical scaling and multiple random starts can be found in Table 9. ILS with the three perturbation

methods was able to find solutions with lower Stress-1 values than classical scaling. The lowest value was found using ILS with *k-NN*$_X$ perturbation: a Stress-1 of .47431. *k-NN*$_X$ perturbation was also the most consistent in finding solutions with lower Stress-1 values than classical scaling (83% of total runs). Followed by ILS with random perturbation (72% of total runs) and *k-NN*$_\Delta$ perturbation (48% of total runs). This is better than multiple random starts, which was not able to find lower Stress-1 values than CS in any of its runs.

Classical scaling found a two-dimensional solution with a Stress-1 of .28459. The three ILS procedures and MR were able to find solutions with a lower Stress-1 of .28211. Again, ILS performed more consistent than multiple random starts. Multiple random starts was only able to find a lower Stress-1 than CS in 7% of its runs, whereas ILS using *k-NN*$_\Delta$ perturbation was able to find a lower Stress-1 in 47% of its runs. Followed by ILS with *k-NN*$_X$ perturbation with 40% and random perturbation with 17%.

Table 9.

Summary of outcome variables for unidimensional and two-dimensional solutions for ILS with k-NN$_\Delta$*, k-NN*$_X$ *and random perturbation using tuned parameters compared to classical scaling (CS) and multiple random starts (MR) for the Food dataset.*

Procedure	p	Lowest Stress-1	%Stress-1$_{<CS}$	Number of local minima
CS	1	**.48990**	-	-
ILS *k-NN*$_\Delta$	1	.47856	48%	100
ILS *k-NN*$_X$	1	.47431	83%	100
ILS RP	1	.47768	72%	99
MR	1	.52231	0%	495
CS	2	**.28459**	-	-
ILS *k-NN*$_\Delta$	2	.28211	47%	10
ILS *k-NN*$_X$	2	.28211	40%	17
ILS RP	2	.28211	17%	78
MR	2	.28211	7%	368

Note: MR was run 500 times, ILS was run 100 times per perturbation method.

4.2.3 Churchill dataset

The Churchill dataset consists of correlation coefficients between ten factors that influence the image of a department store. The data is acquired from responses to of a random selection of shoppers (Churchill, 1995). The data has been used in many MDS methods, such as

PERMAP (Heady & Lucas, 1997). The two-dimensional configuration found by Heady & Lucas (1997), had a Stress-1 value of .0793. Which they concluded to be the true global minimum.

A summary of the results can be found in Table 10**.** All three ILS procedures were able to find lower Stress-1 values than classical scaling. *k-NN$_X$* perturbation performed most consistent: finding the same solution with a of Stress-1 equalling .079065 in each run. *k-NN$_\Delta$* perturbation performed less consistent: finding the same solution in 51% of the runs, in the other runs it got stuck in three different local minima. ILS with random perturbation performed the worse of all perturbation methods, as it found the same lowest minimum in only10% of its runs. ILS with random perturbation also got stuck in the most different local minima. Multiple random starts performed worse compared to ILS with each perturbation method: it was able to find the lowest minimum in only 3,2% of the runs and found worse solutions than CS in 72,8% of its runs.

Table 10.

Summary of outcome variables for ILS with k-NN$_\Delta$, k-NN$_X$ and random perturbation using tuned parameters compared to classical scaling (CS) and multiple random starts (MR) for the Churchill dataset.

Procedure	Lowest Stress-1	%Stress-1$_{<CS}$	Number of local minima
Heady & Lucas	**.07930**	**-**	**-**
CS	.08557	-	-
ILS *k-NN$_\Delta$*	.07906	93%	4
ILS *k-NN$_X$*	.07906	100%	1
ILS RP	.07906	52%	44
MR	.07906	100%	124

Note: MR was run 500 times, ILS was run 100 times per perturbation method.

4.2.4 Kabah dataset

The Kabah dataset (Robinson, 1951) contains dissimilarities between deposits of pottery found at 17 archaeological dig sites. As similar types of pottery were developed in the same period, it can be inferred that high dissimilarities between sites indicate that the pottery was made during different points in time. A unidimensional solution can be made which places the deposits in chronological order, from oldest to newest. It has one measurement error: the ninth deposit is positioned between the seventh and the eight deposit (Pliner, 1996). Lau, Leung and

Tse (1998) used nonlinear programming to solve the unidimensional solution but could not guarantee the global minimum. Simantirakis (1996) used mixed integer programming and could guarantee the global minimum, but at the cost of high computation time.

The results of ILS with each perturbation method compared to classical scaling and multiple random starts can be found in Table 11. Classical scaling could not place the deposits in the correct order: deposits seventeen and fifteen are switched, resulting in a solution with a Stress-1 of .26623. Independent of perturbation method, ILS was able to put the deposits in the correct order, resulting in a solution with a Stress-1 of .26382. Also, ILS with each perturbation method found solutions with a lower Stress-1 values than CS in every run. MR performed worse, as it found a solution with a lower Stress-1 than CS in only one run. Still, even the solution with the lowest Stress-1 did not order the deposits correctly as deposits, switching deposits seven and eight.

Table 11

Summary of outcome variables for ILS with k-NN_Δ, k-NN_X and random perturbation using tuned parameters compared to classical scaling (CS) and multiple random starts (MR) for the Kabah dataset.

Procedure	Lowest Stress-1	%Stress-$1_{<CS}$	Number of local minima
CS	**.26623**	-	-
ILS *k*-NN_Δ	.26382	100%	43
ILS *k*-NN_X	.26382	100%	20
ILS RP	.26382	100%	12
MR	.26508	0.2%	499

Note: MR was run 500 times, ILS was run 100 times per perturbation method.

5. Conclusion and discussion

In this thesis, an explorative study was performed on the application of the ILS metaheuristic with MDS as *localsearch.* Also, two new *k*-NN perturbation methods were introduced and compared to random perturbation. ILS with the three perturbation methods was compared with known MDS methods: multiple random starts and classical. The simulation study concluded that unidimensional MDS solutions benefit from higher perturbation strength and size, whereas two-dimensional MDS solutions benefit from lower perturbation strength and size. Compared to multiple random starts, each perturbation method was more consistent in finding the global minimum. Between perturbation methods, both *k*-NN perturbation methods

performed more consistent than random perturbation on two-dimensional MDS solutions.

Next, examples of applications were given on unidimensional and two-dimensional datasets using ILS with the three tuned perturbation methods. The aim was to find solutions with similar or even lower Stress-1 than classical scaling and multiple random starts. For unidimensional solutions, ILS performed better than classical scaling and multiple random starts. For two-dimensional solutions, ILS performed equally well as classical scaling and better than multiple random starts. ILS with random perturbation performed the least consistent of the three perturbation methods, having the lowest percentage of runs resulting in a better solution than classical scaling. This is possibly due to the random nature of RP: even though the distribution, from which the added perturbation is picked gets wider, the chance of picking a value close to zero is still possible. If this occurs, no actual perturbation is added, or the perturbation is too small which causes the next iteration of MDS to return to the current local minimum.

Based on the findings of this thesis, the ILS metaheuristic is proven to be effective with MDS as *localsearch.* It is able to find similar or even better solutions than classical scaling and is more flexible: perturbation parameters can be tuned based on the dimensionality of the desired solution. Also, it is more efficient and consistent than multiple random starts. *k*-NN perturbation also is an effective perturbation method for MDS, performing more consistent than random perturbation.

Earlier research by Hubert, Arabie and Meulman (2002) investigated the effectiveness of pairwise interchange for unidimensional scaling problems. Pairwise interchange iteratively swaps one object with its closest neighbour, swaps two pairs of objects or places one object between each pair of objects, until the solution no longer improves. Pairwise interchange has similarities with *k*-NN perturbation using a small *k* and a small perturbation size. In their study, this method was studied with datasets with a maximum of 25 objects. The proposed ILS with both *k*-NN perturbation methods was able to find unidimensional solutions for bigger datasets with a maximum of 45 objects, by swapping 50% of all objects with the closest 50% of all objects. This indicates that a high perturbation strength and size is better for the optimization of unidimensional solutions.

A possible reason for finding higher Stress-1 values for the Morse dataset than Palubeckis (2013) and Buja & Swayne (2002), was the usage of data without error to tune the perturbation parameters. If the parameters were tuned on the real-life examples instead, ILS might find better solutions. However, ILS proved to be much faster than the methods used in the studies of Palubeckis (2013) and Buja & Swayne (2002). Similarly, Palubeckis (2013) had

to use a subset of the Food dataset as it would be too computationally heavy to run the full dataset, whereas ILS with each perturbation method was able to find unidimensional solutions using the complete dataset within reasonable runtimes. Also, high computation times were needed to find the global minimum for the Kabah dataset in studies from Leung and Tse (1998) and Simantirakis (1996), whereas ILS was able to find the global minimum every time, in low runtimes. Lastly, for the Churchill dataset, ILS with each perturbation method was able to find better solutions than Heady and Lucas (1997).

Follow-up studies can look into comparisons between the proposed ILS procedure with other known metaheuristic such as the Quadratic Assignment Interchange heuristics (Hubert, Arabie & Meulman, 2002), pairwise interchange (Brusco & Stahl, 2000) and the multiple perturbation operator method used in Sabar and Kendal (2015). As this thesis found better results for *k*-NN perturbation than RP, further study into the effectiveness of *k*-NN perturbation as part of the mutation phase in genetic algorithm is also recommended. *k*-NN perturbation has proven to be effective for MDS, follow-up studies can also look into its effectiveness on other optimization problems such as the Traveling Salesman Problem.

Lastly, the proposed ILS with MDS as *localsearch* should also be studied further with the methods:

- Look into how well the proposed ILS performs with nonmetric-MDS. How well is it able to find better solutions for ordinal data than MR and CS? And how prone is the proposed ILS to degenerate solutions while using ordinal data.
- Only the perturbation method, perturbation strength and perturbation size were varied in this thesis. The implementation of a dynamic acceptance criterion and the history parameter (Lourenco, Martin and Stützle, 2001) could be studied.
- Dynamically decrease perturbation strength: start off with high perturbation strength and decrease it after each better solution. This also helps the search for better solutions in the current neighbourhood of the solution space. Increase the perturbation strength if it is apparent that the perturbation is too weak to escape the current local minimum.
- Perturbation in this thesis was added to randomly chosen objects. Further research could look into the effect of adding perturbance to the object(s) which cause(s) the most Stress-1.

Acknowledgements

I did not think my thesis would be such a long journey. A journey filled with problems, bright moments, less bright moment, moments of intense motivation and losing it all together. During my journey I was met with new challenges: juggling a traineeship and my first professional job with writing my thesis. Something I could not combine at times. But thanks to my always patient and helpful supervisor, Prof. Frank Busing, I eventually could. Without him I would not have been where I am now.

I would like to thank my roommates for supporting me and helping me vent my frustrations.

References

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Borg, I., Groenen, P. J., & Mair, P. (2013). MDS algorithms. In *Applied Multidimensional Scaling* (pp. 81-86). Springer, Berlin, Heidelberg.

Borg, I., & Mair, P. (2017). The choice of initial configurations in multidimensional scaling: Local minima, fit, and interpretability. *Austrian Journal of Statistics*, *46*(2), 19-32.

Brusco, M. J., Köhn, H. F., & Stahl, S. (2008). Heuristic implementation of dynamic programming for matrix permutation problems in combinatorial data analysis. *Psychometrika*, 73(3), 503.

Brusco, M. J., & Steinley, D. (2014). Model selection for minimum-diameter partitioning. *British Journal of Mathematical and Statistical Psychology*, *67*(3), 471-495.

Brusco, M. J., & Stahl, S. (2000). Using Quadratic Assignment Methods to Generate Initial Permutations for Least-Squares Unidimensional Scaling of Symmetric Proximity Matrices. *Journal of Classification*, 17(2).

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., & Chen, L. (2008). Data visualization with multidimensional scaling. *Journal of computational and graphical statistics*, *17*(2), 444-472.

Churchill, G. A., & Iacobucci, D. (1995). Marketing research. *Methodological foundations*, 6.

De Leeuw, J. (1977). Recent developments in statistics, ch. Applications of convex analysis to multidimensional scaling. *North-Holland, Amsterdam*, (1), 1-5.

Dong, X., Nowak, M., Chen, P., & Lin, Y. (2015). Self-adaptive perturbation and multi-neighborhood search for iterated local search on the permutation flow shop problem. *Computers & Industrial Engineering*, *87*, 176-185.

France, S. L., & Akkucuk, U. (2021). A review, framework, and R toolkit for exploring, evaluating, and comparing visualization methods. *The Visual Computer*, 37(3), 457-475.

Groenen, P. J. (1993). The majorization approach to multidimensional scaling: some problems and extensions. *DSWO Press*.

Guttman, L. (1968). A general nonmetric technique for finding the smallest coordinate space for a configuration of points. *Psychometrika*, *33*(4), 469-506.

Heady, R. B., & Lucas, J. L. (1997). PERMAP: An interactive program for making perceptual maps. *Behavior Research Methods, Instruments, & Computers*, *29*(3), 450-455.

Hubert, L., Arabie, P., & Meulman, J. (2001). Combinatorial data analysis: Optimization by dynamic programming. *Society for industrial and Applied Mathematics*.

Hubert, L. J., Arabie, P., & Meulman, J. J. (2002). Linear unidimensional scaling in the L 2-norm: Basic optimization methods using MATLAB. *Journal of Classification*, 19(2), 303-328.

Karimi-Mamaghan, M., Pasdeloup, B., Mohammadi, M., & Meyer, P. (2021, June). A Learning-Based Iterated Local Search Algorithm for Solving the Traveling Salesman Problem. In *International Conference on Optimization and Learning* (pp. 45-61). Springer, Cham.

Köhn, H. F., Steinley, D., & Brusco, M. J. (2010). The p-median model as a tool for clustering psychological data. *Psychological methods*, 15(1), 87.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1-27.

Lau, K. N., Leung, P. L., & Tse, K. K. (1998). A Nonlinear Programming Approach to Metric Unidimensional Scaling. *Journal of classification*, 15(1).

Lin, B. L., Sun, X., & Salous, S. (2016). Solving travelling salesman problem with an improved hybrid genetic algorithm. *Journal of computer and communications.*, *4*(15), 98-106.

Lourenço, H. R., Martin, O., & Stützle, T. (2001, July). A beginner's introduction to iterated local search. In *Proceedings of MIC* (Vol. 4, pp. 1-6).

Lourenço, H. R., Martin, O. C., & Stützle, T. (2019). Iterated local search: Framework and applications. In *Handbook of metaheuristics* (pp. 129-168). Springer, Cham.

Mair, P., Borg, I., & Rusch, T. (2016). Goodness-of-fit assessment in multidimensional scaling and unfolding. *Multivariate behavioral research*, *51*(6), 772-789.

Mair P, Groenen PJF, de Leeuw J (2022). “More on multidimensional scaling in R: smacof version2.” *Journal of Statistical Software*, **102**(10), 1–47.

Martin, O., Otto, S. W., & Felten, E. W. (1992). Large-step Markov chains for the TSP incorporating local search heuristics. *Operations Research Letters*, *11*(4), 219-224.

Palubeckis, G. (2013). An improved exact algorithm for least-squares unidimensional scaling. *Journal of Applied Mathematics*, 2013.

Pliner, V. (1996). Metric unidimensional scaling and global optimization. *Journal of classification*, 13(1), 3-18.

Robinson, W. S. (1951). A method for chronologically ordering archaeological deposits. *American antiquity*, 16(4), 293-301.

Ross, B. H., & Murphy, G. L. (1999). Food for thought: Cross-classification and category organization in a complex real-world domain. *Cognitive psychology*, *38*(4), 495-553.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning tasks. *Journal of Experimental Psychology*, *53*(2), 94.

RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA URL http://www.rstudio.com/.

Sabar, N. R., & Kendall, G. (2015). An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem. *Omega*, *56*, 88-98.

Simantiraki, E. (1996). Unidimensional scaling: A linear programming approach minimizing absolute deviations. *Journal of classification*, 13, 19-25.

Smit, J (2018). *Avoiding Local Minima in Nonmetric Multidimensional Scaling by Means of Deterministic Annealing and Iterated Local Search* (Unpublished Masters's thesis). Leiden University, Leiden.

Stützle, T. (1998). Applying iterated local search to the permutation flow shop problem. *Technical Report AIDA*-98-04, FG Intellektik, TU Darmstadt.

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, *17*(4), 401-419.

Vansteenwegen, P., & Mateo, M. (2014). An iterated local search algorithm for the single-vehicle cyclic inventory routing problem. *European Journal of Operational Research*, *237*(3), 802- 813.

Wagenaar, W. A., & Padmos, P. (1971). Quantitative interpretation of stress in Kruskal's multidimensional scaling technique. *British Journal of Mathematical and Statistical Psychology*, *24*(1), 101-110.

Žilinskas, A., & Žilinskas, J. (2006). Parallel hybrid algorithm for global optimization of problems occurring in MDS-based visualization. *Computers & Mathematics with Applications*, 52(1-2), 211-224.

Appendix A

Runtimes found by ILS per perturbation method in the simulation study.

Figure 9.

Runtimes at the moment of finding unidimensional solutions with the lowest Stress-1 using ILS with k-NN$_\Delta$ perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100

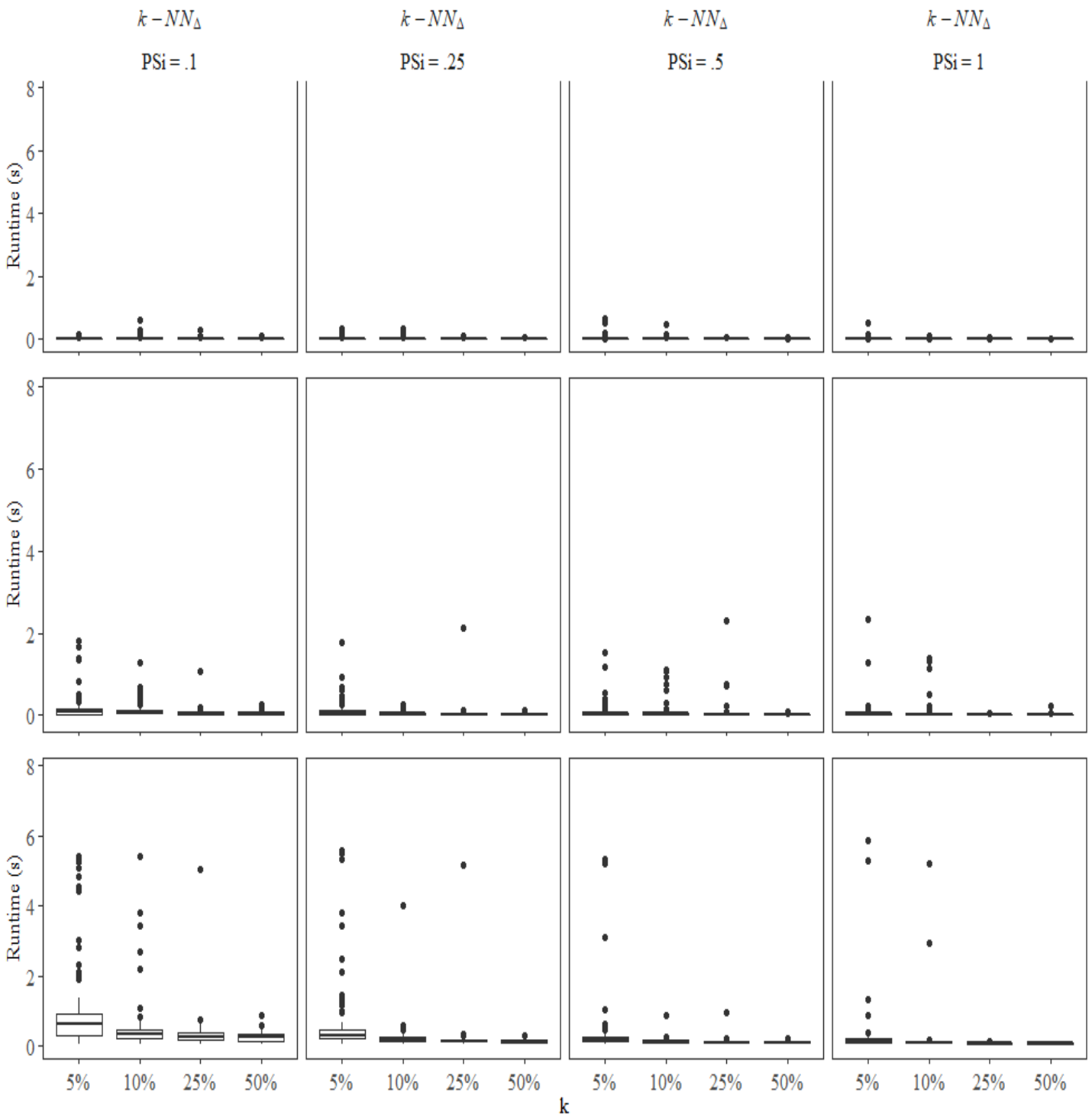


Figure 10.

Runtimes at the moment of finding unidimensional solutions with the lowest Stress-1 using ILS with k-NN_X perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

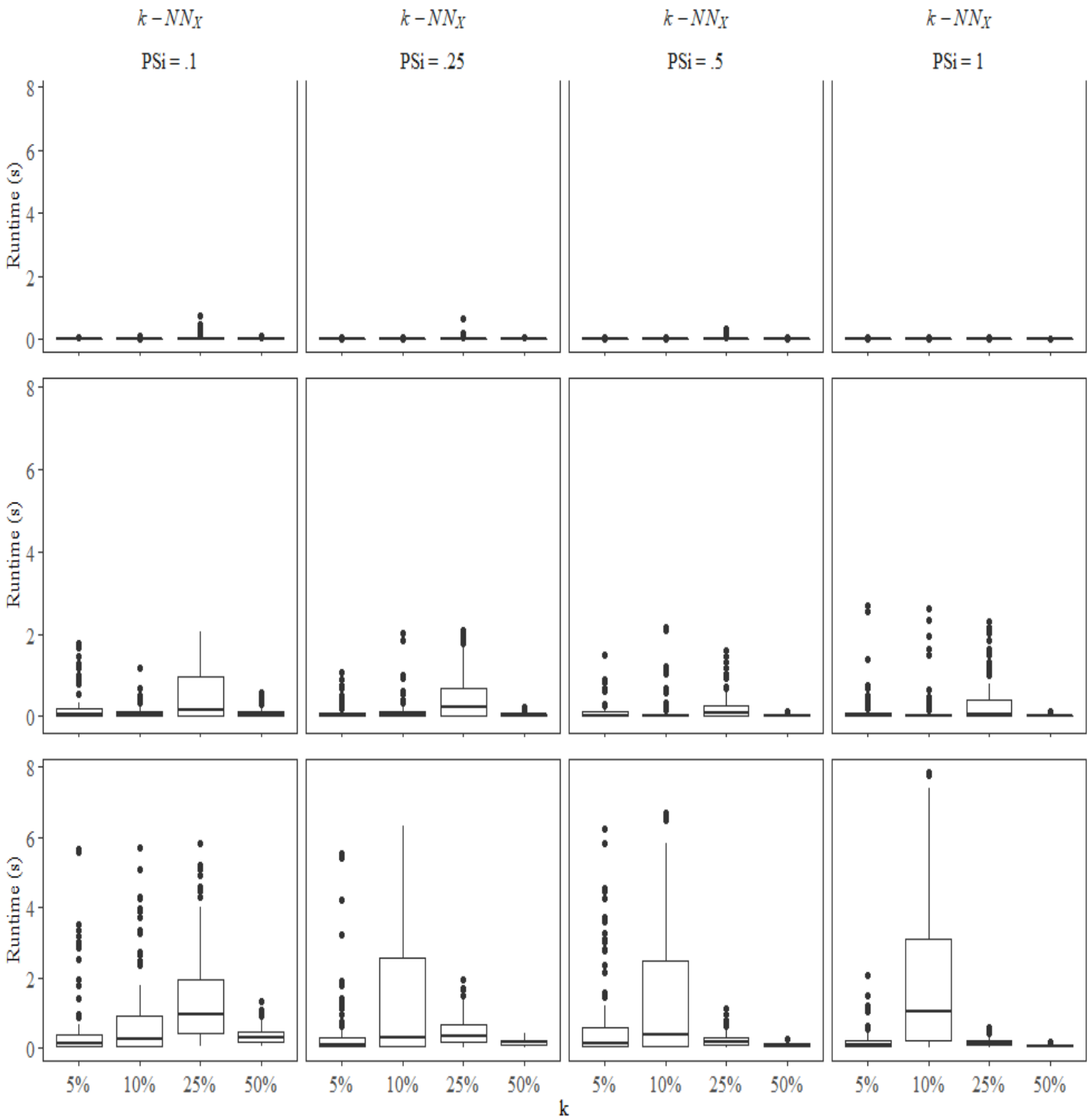


Figure 11.

Runtimes at the moment of finding unidimensional solutions with the lowest Stress-1 using ILS with random perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

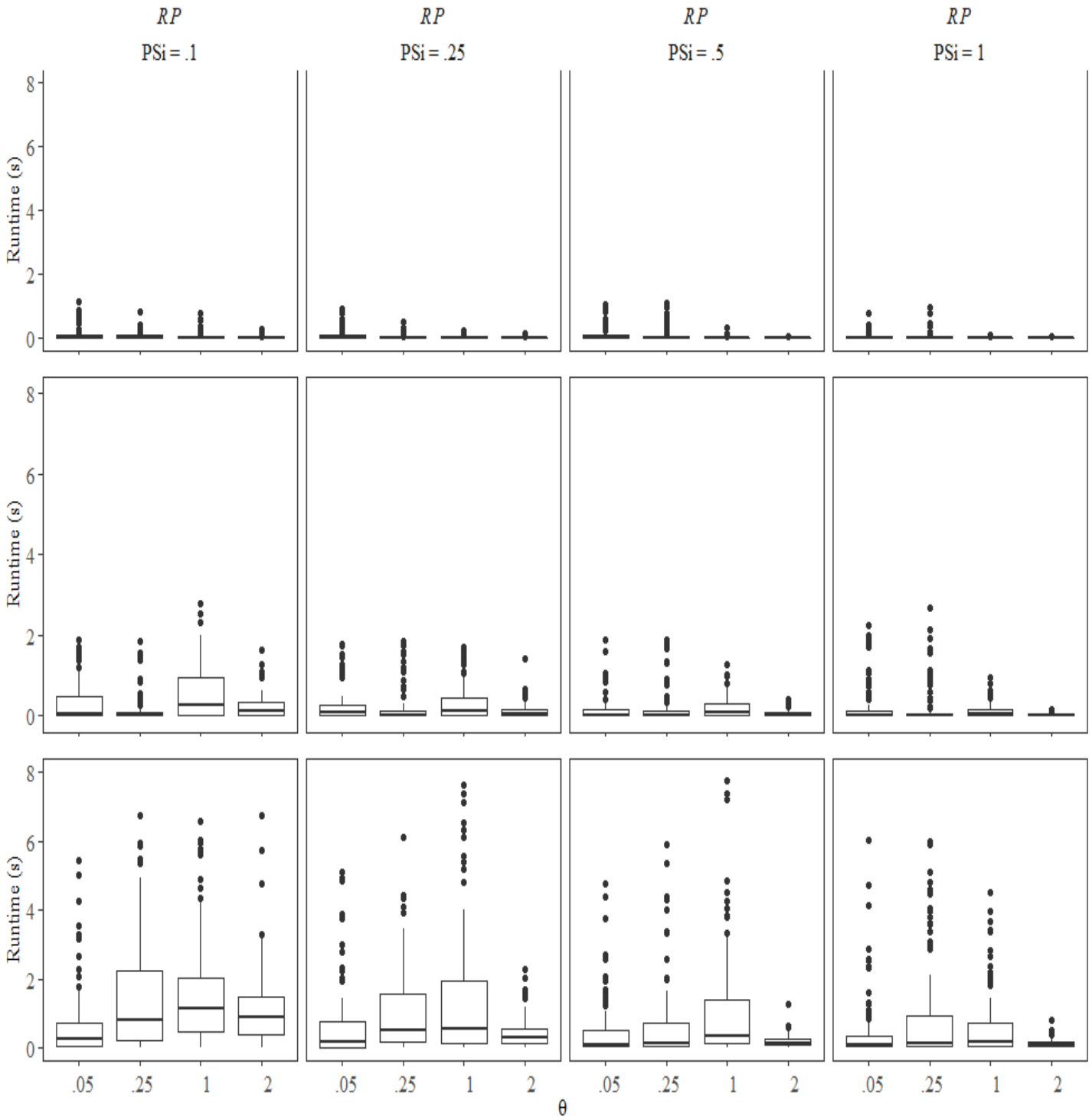


Figure 12.

Runtimes at the moment of finding two-dimensional solutions with the lowest Stress-1 using ILS with k-NN_Δ at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

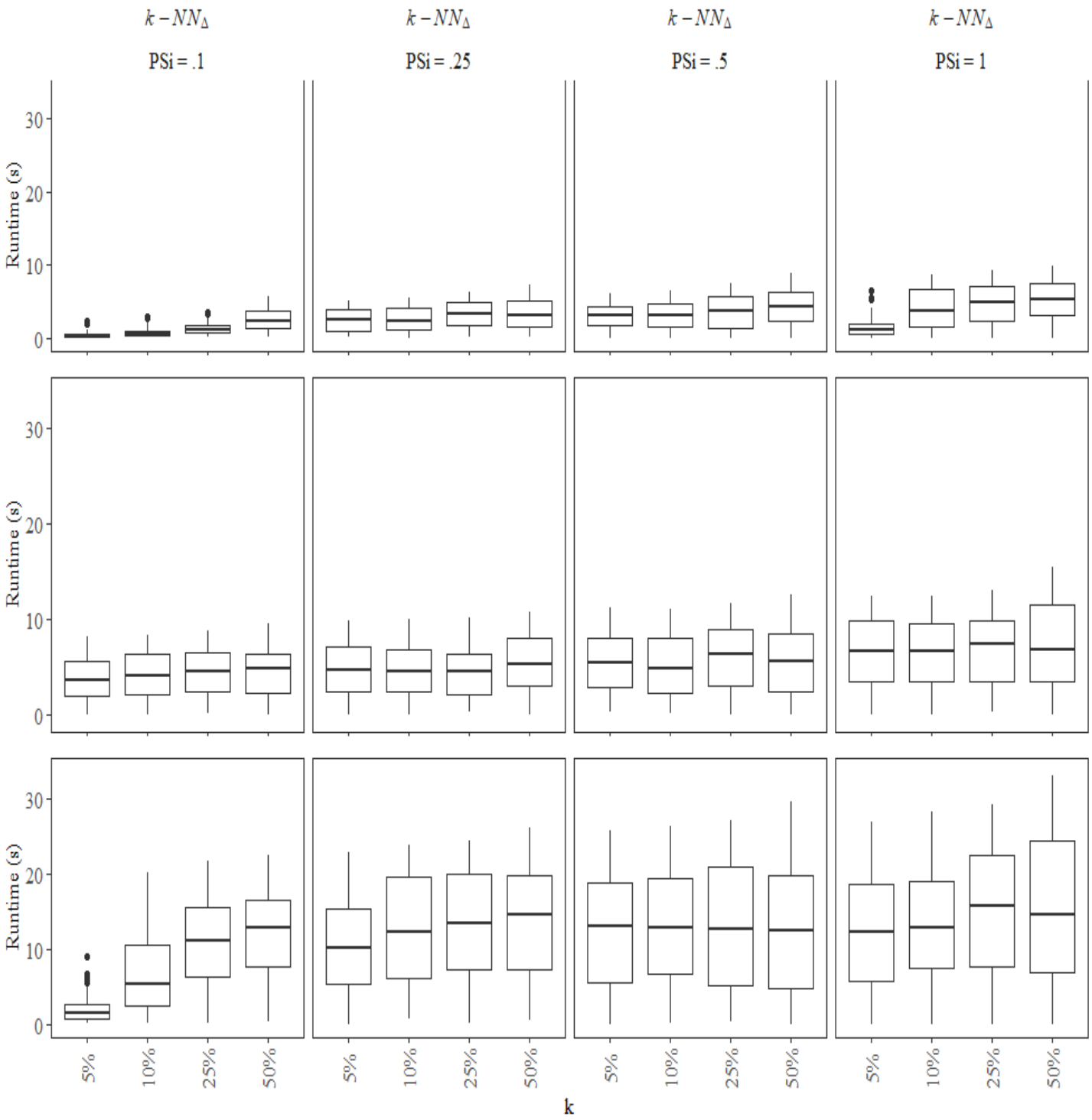


Figure 13.

Runtimes at the moment of finding two-dimensional solutions with the lowest Stress-1 using ILS with k-NN_X at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

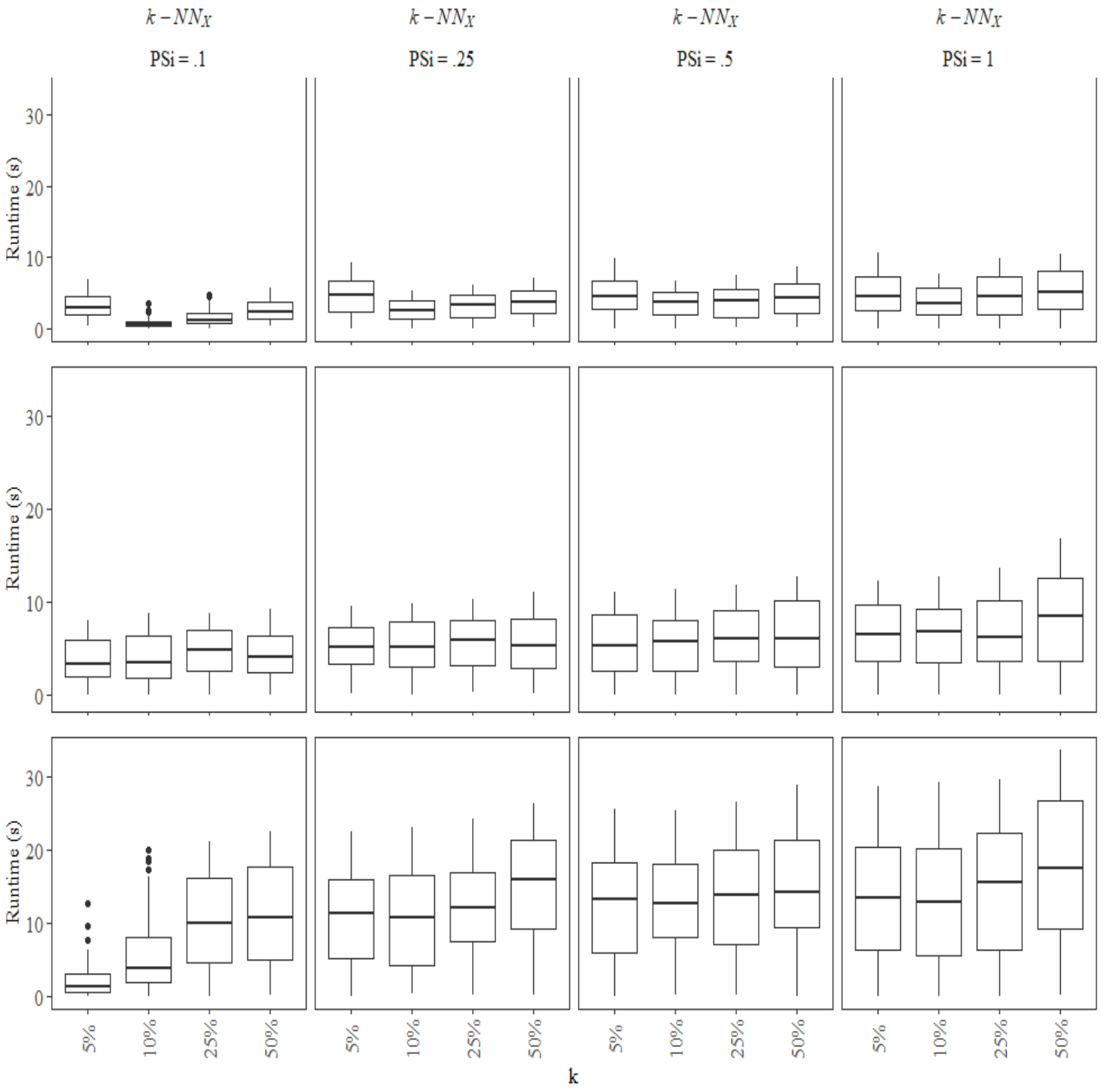
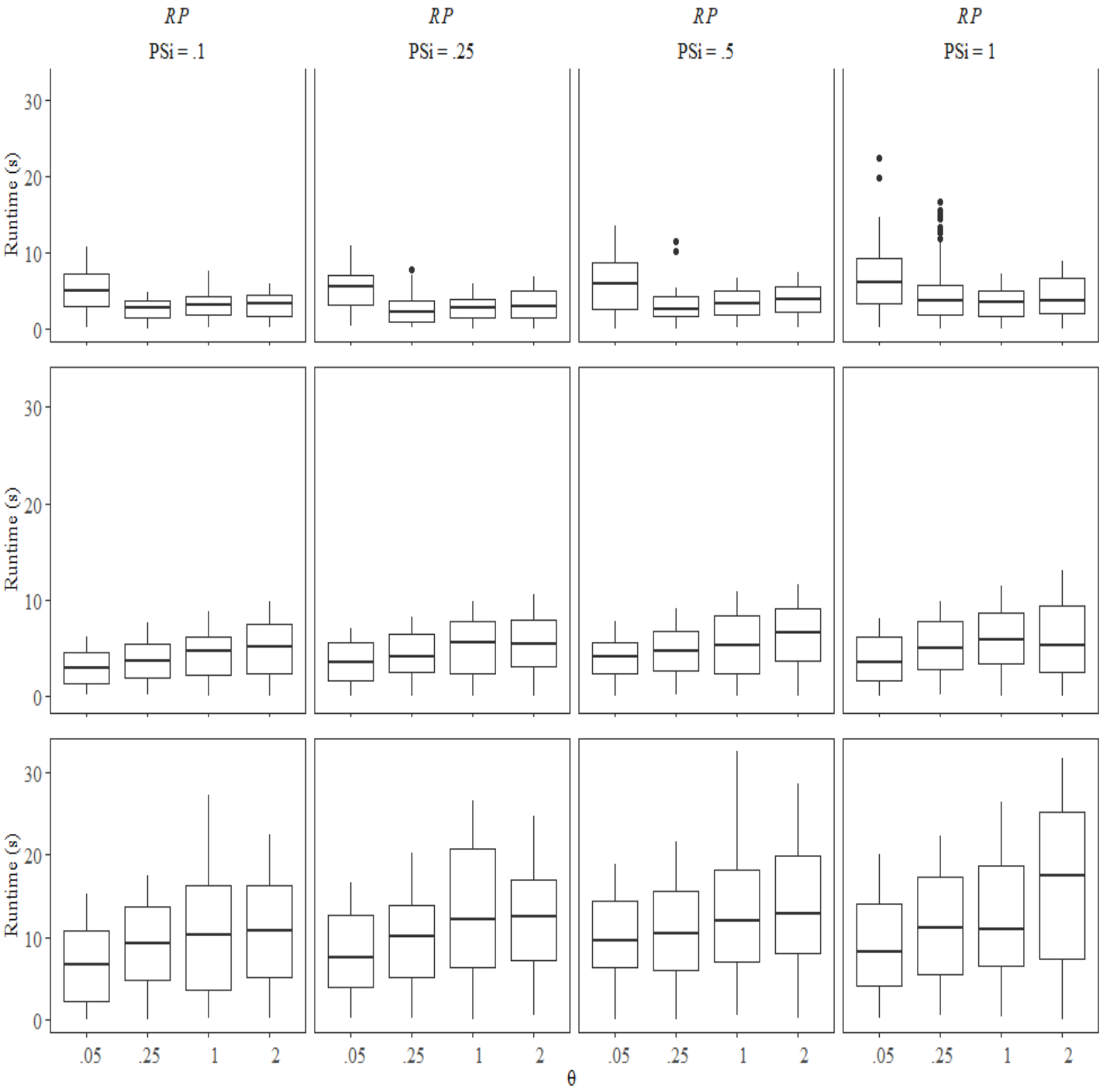


Figure 14.

Runtimes at the moment of finding two-dimensional solutions with the lowest Stress-1 using ILS with random perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.



Appendix B

Number of calls of the MDS function needed by ILS per perturbation method in the simulation study.

Figure 15.

Number of calls of the MDS function needed at moment of finding the best local minimum or global minimum for unidimensional datasets using ILS with k-NN_Δ perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

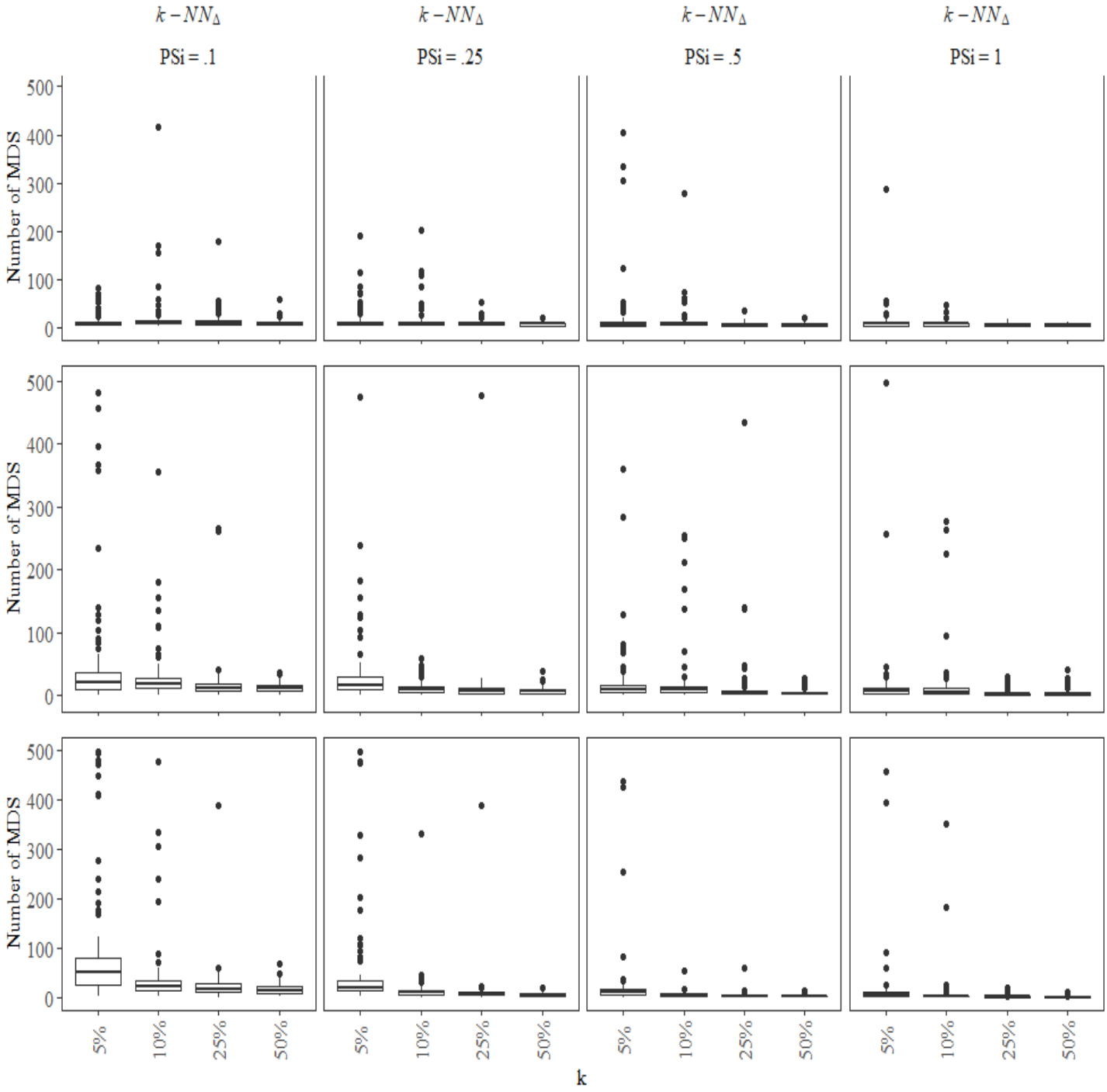


Figure 16.

Number of calls of the MDS function needed at moment of finding the best local minimum or global minimum for unidimensional datasets using ILS with k-NN_X perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

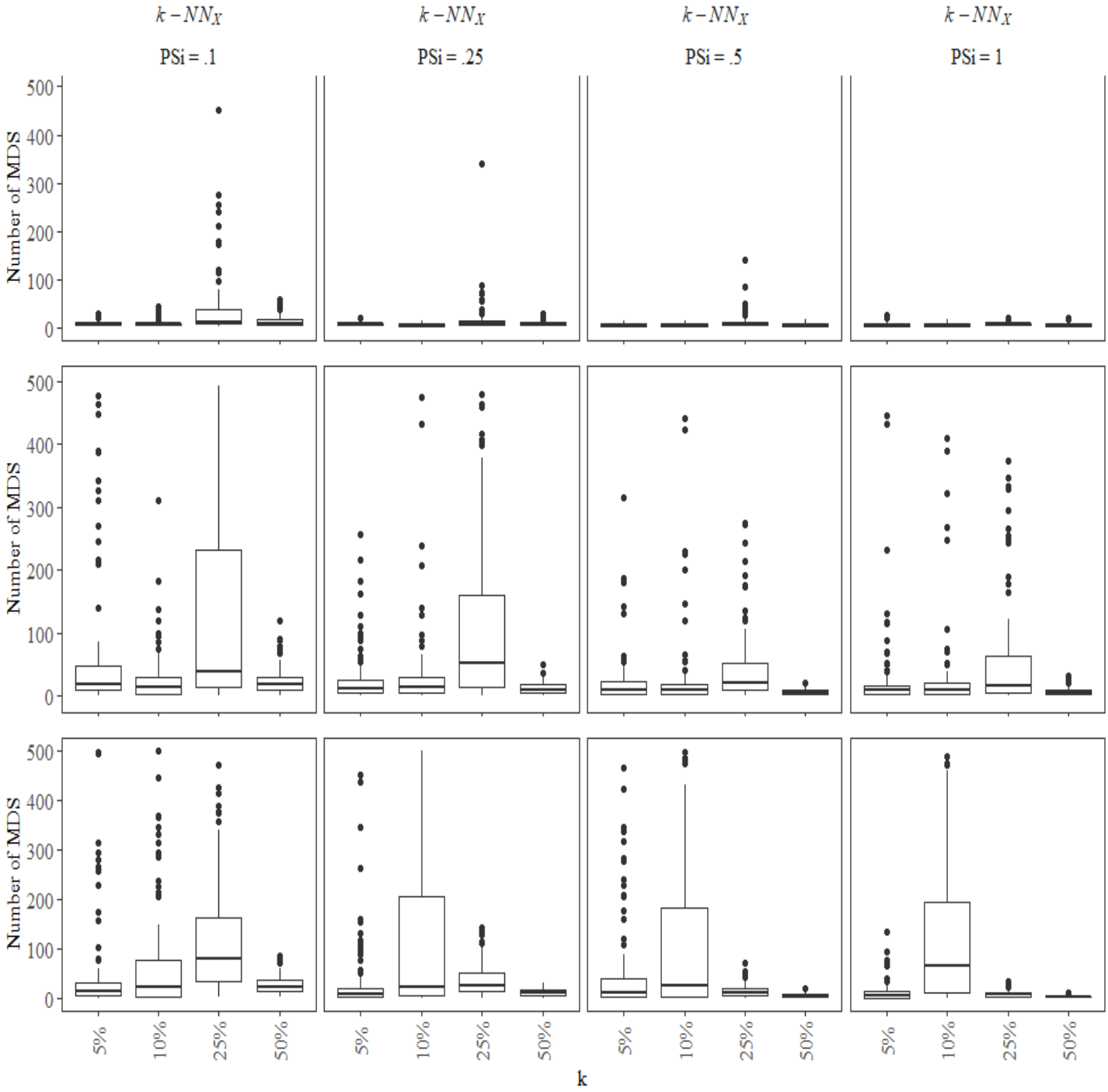


Figure 17.

Number of calls of the MDS function needed at moment of finding the best local minimum or global minimum for unidimensional datasets using ILS with random perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

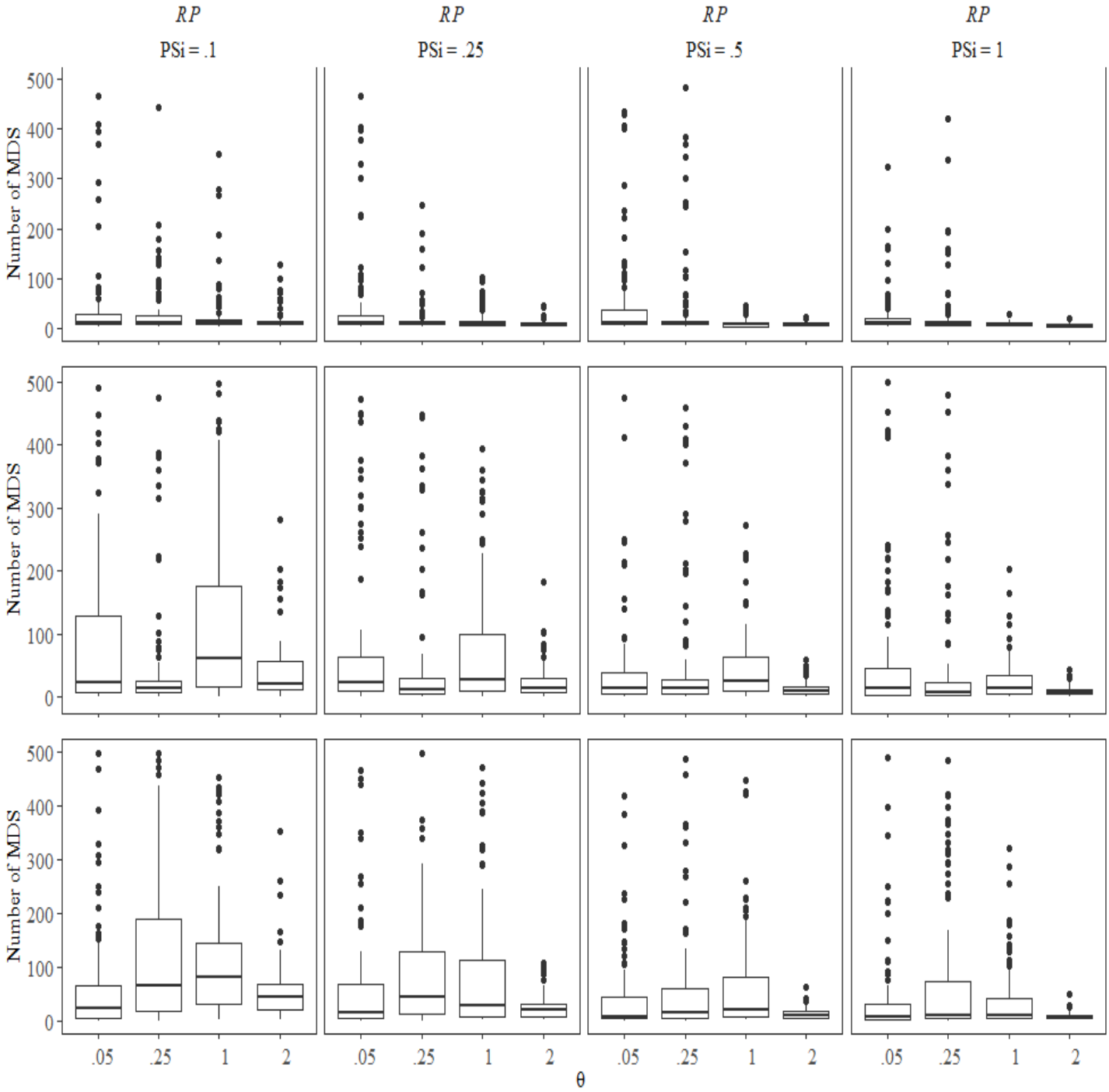


Figure 18.

Number of calls of the MDS function needed at moment of finding the best local minimum or global minimum for two-dimensional datasets using ILS with k-NN$_\Delta$ perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

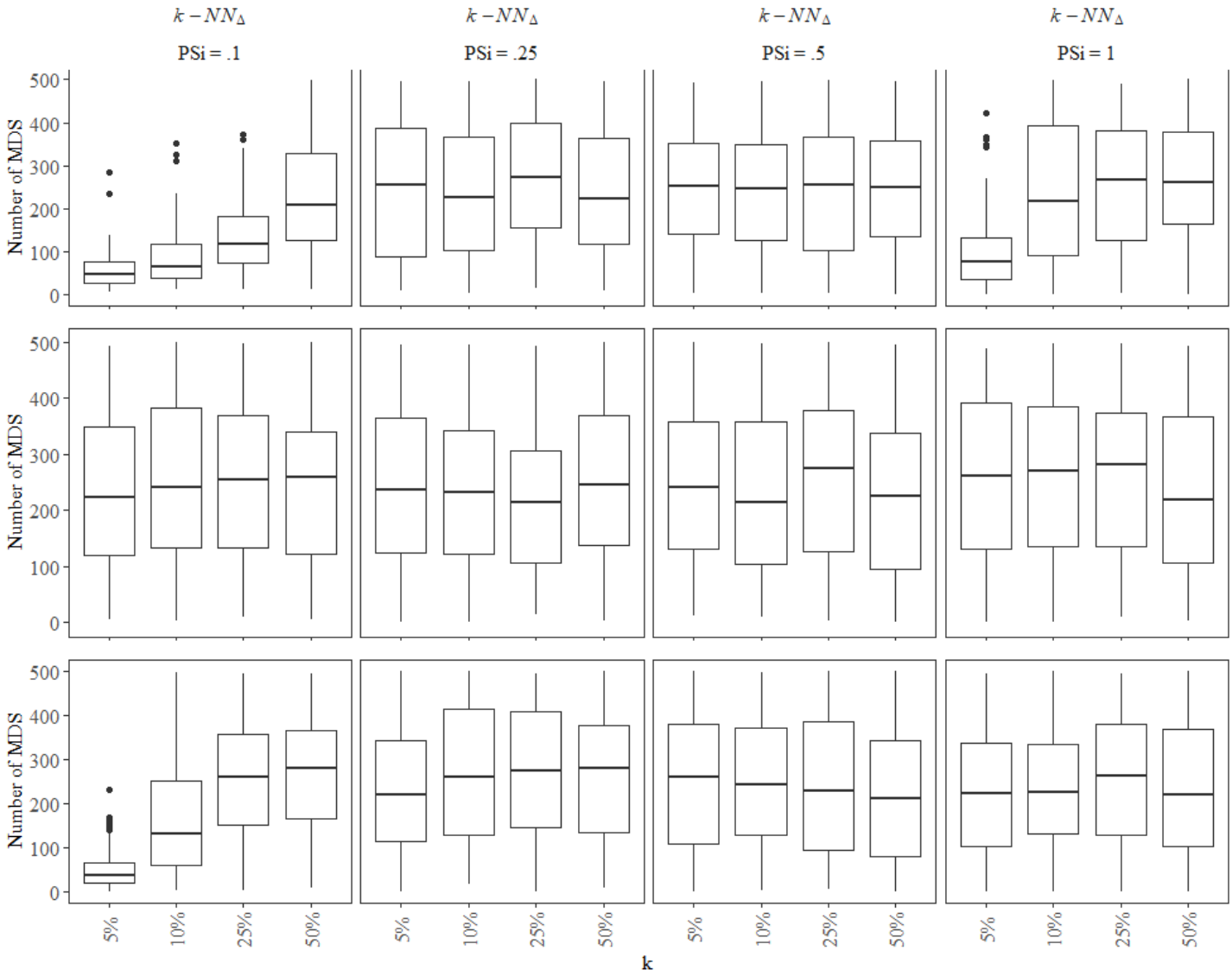


Figure 19.

Number of calls of the MDS function needed at moment of finding the best local minimum or global minimum for two-dimensional datasets using ILS with k-NN$_X$ perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.

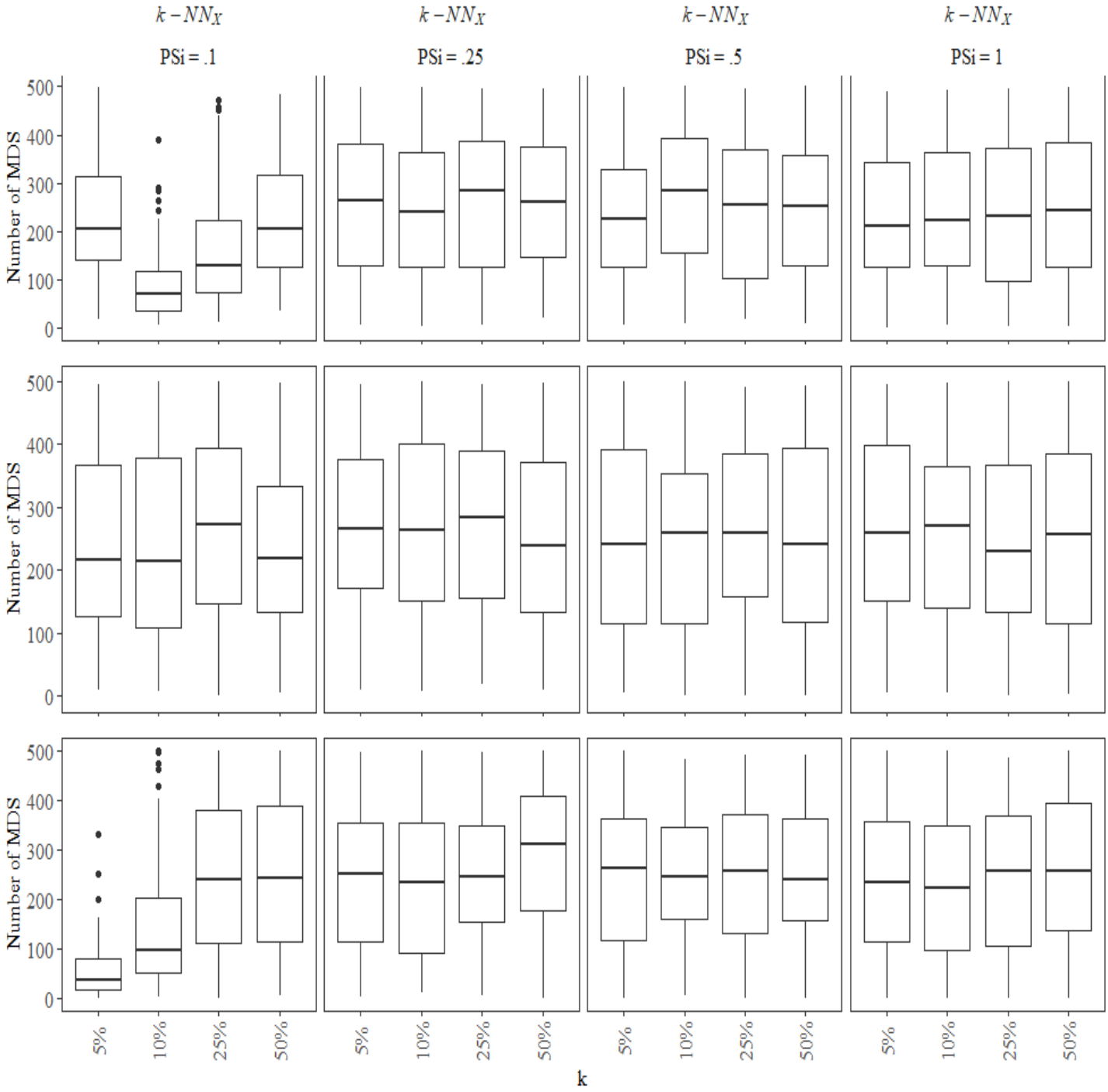


Figure 20.

Number of calls of the MDS function needed at moment of finding the best local minimum or global minimum for two-dimensional datasets using ILS with random perturbation at different levels of PS. In the columns from left to right: different levels of PSi. In the rows from top to bottom: dataset sizes 20, 50 and 100.