# Batch Proposals for Model-Based Multi-Objective Optimization in the context of SVM tuning

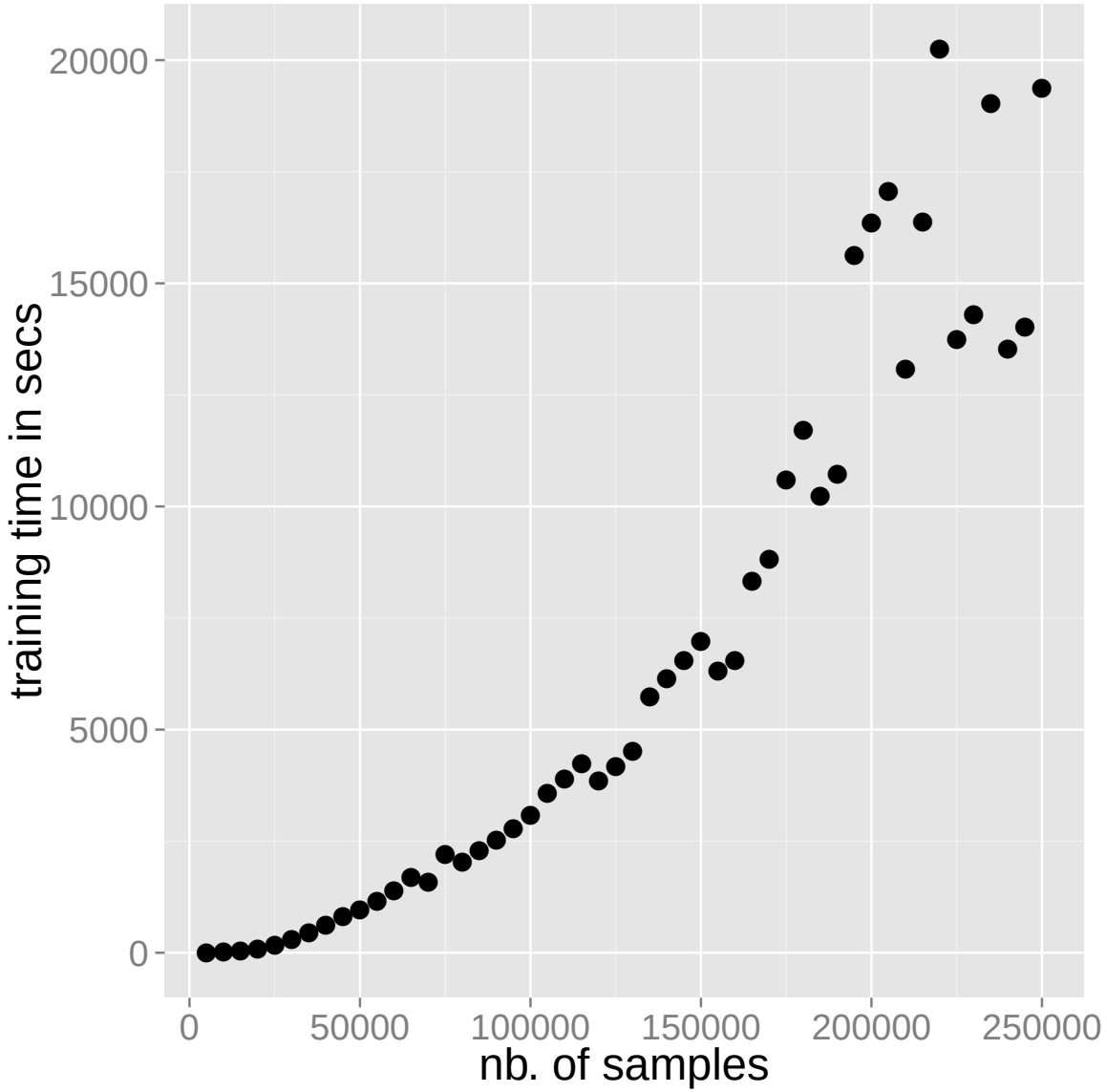
**Daniel Horn**

TU Dortmund

03-Mar-2016

# Complexity of Support Vector Machine training

- **Problem**:
  Complexity of SVM training is $\mathcal{O}(n^2)$ ... $\mathcal{O}(n^3)$
- **Example**:
  Training of LIBSVM on $\{5000, 10000, ..., 250000\}$ samples of the poker dataset
- **Solution**:
  Approximate SVM training
- **Problem**:
  Which of the many approximation algorithms to use?



- **We expect:** Every solver has a trade-off between training time and prediction error: Given more time, a solver (should) reach a better solution.
- **Our goal:** Analyze this trade-off! Solve the multi-criteria optimization problem with respect to the two objectives error and training time by varying the parameters.
- **The challenge:** Optimizing 2 expensive objectives in a 3-to-4-dimensional parameter space.
- **Our approach:** Replace standard grid search with more sophisticated PAREGO-algorithm.

# Extend the PAREGO-Algorithm

1. Scalarize objectives using the augmented Tchebycheff norm

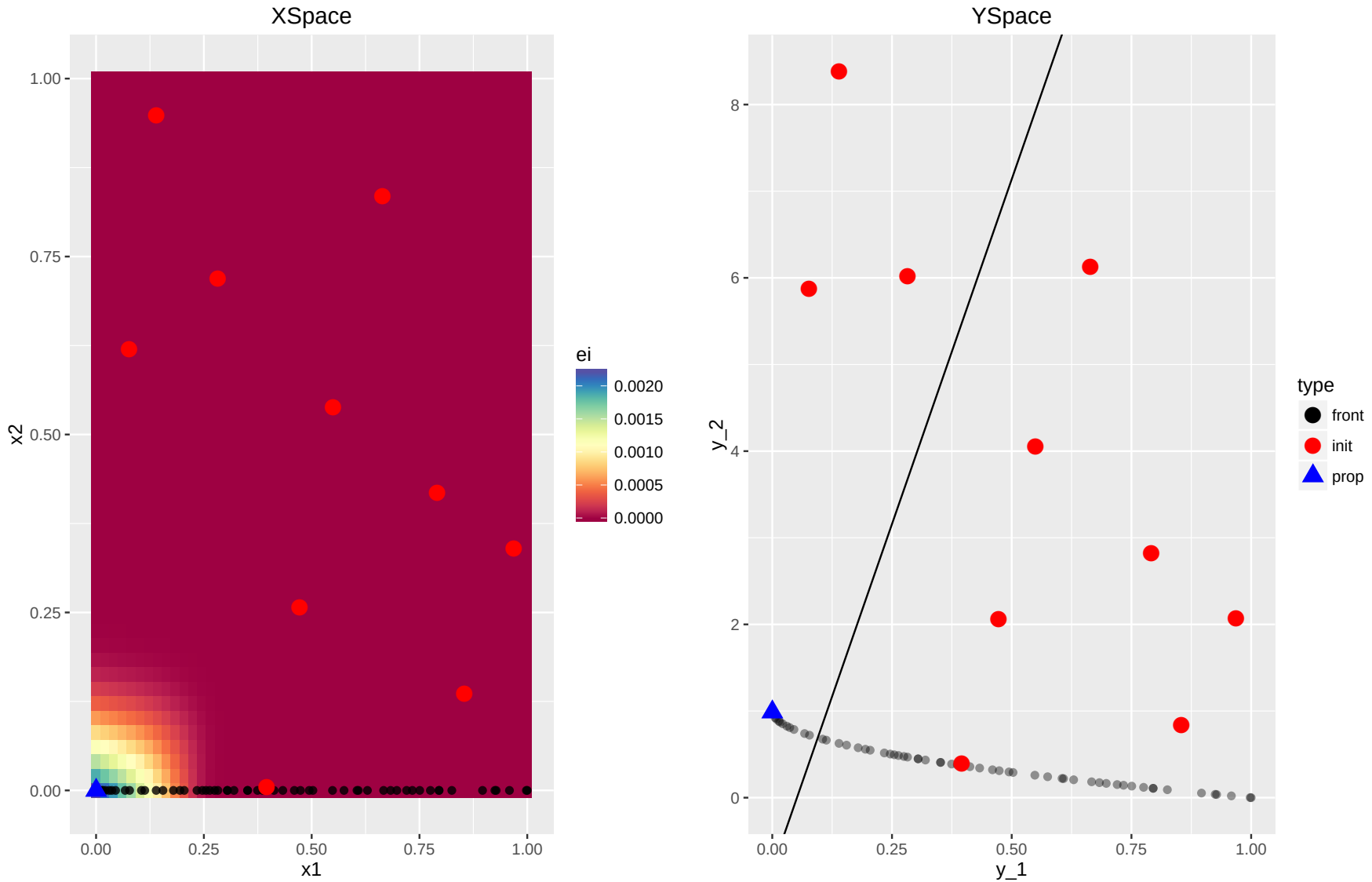
$$u(x) = -\max\left[\vec{w}(\vec{f}(\vec{x}) - \vec{i})\right] + \rho \sum \vec{w}(\vec{f}(\vec{x}) - \vec{i})$$

   with ideal point $\vec{i}$ and uniformly distributed weight vector $\vec{w}$ $(\sum w_i = 1)$ and fit surrogate model to the respective scalarization
2. Single-obj. optimization of expected improvement (EI)

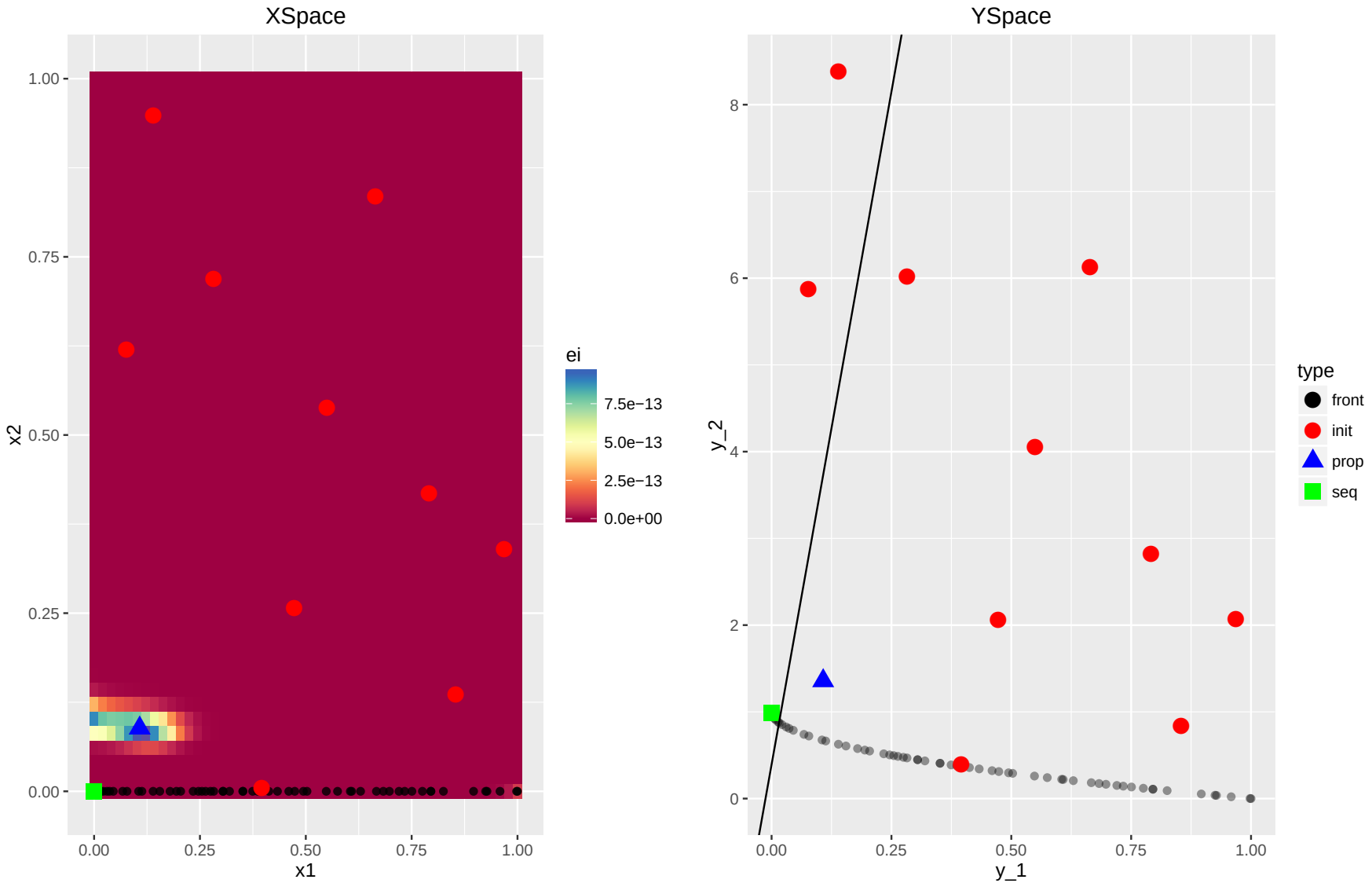
Modification: **Increase the number and diversity of randomly drawn weight vectors**

- If $N$ points are desired, $cN$ $(c > 1)$ weight vectors are considered
- Greedily reduce set of weight vectors by excluding one vector of the pair with minimum distance
- Scalarizations implied by each weight vector are computed
- Fit and optimize models for each scalarization
- Optima of each model build the batch to be evaluated

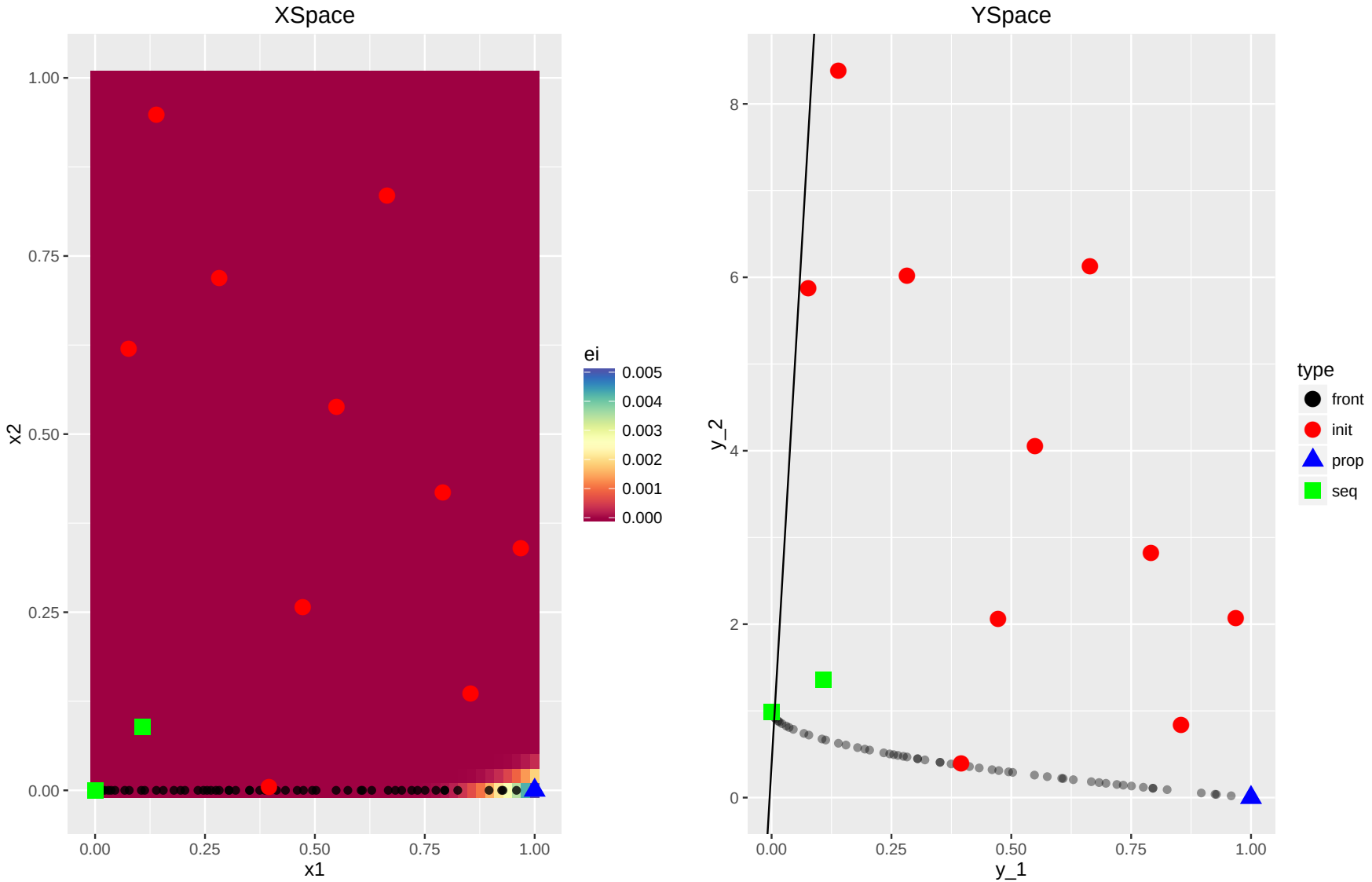
# Animation of PAREGO



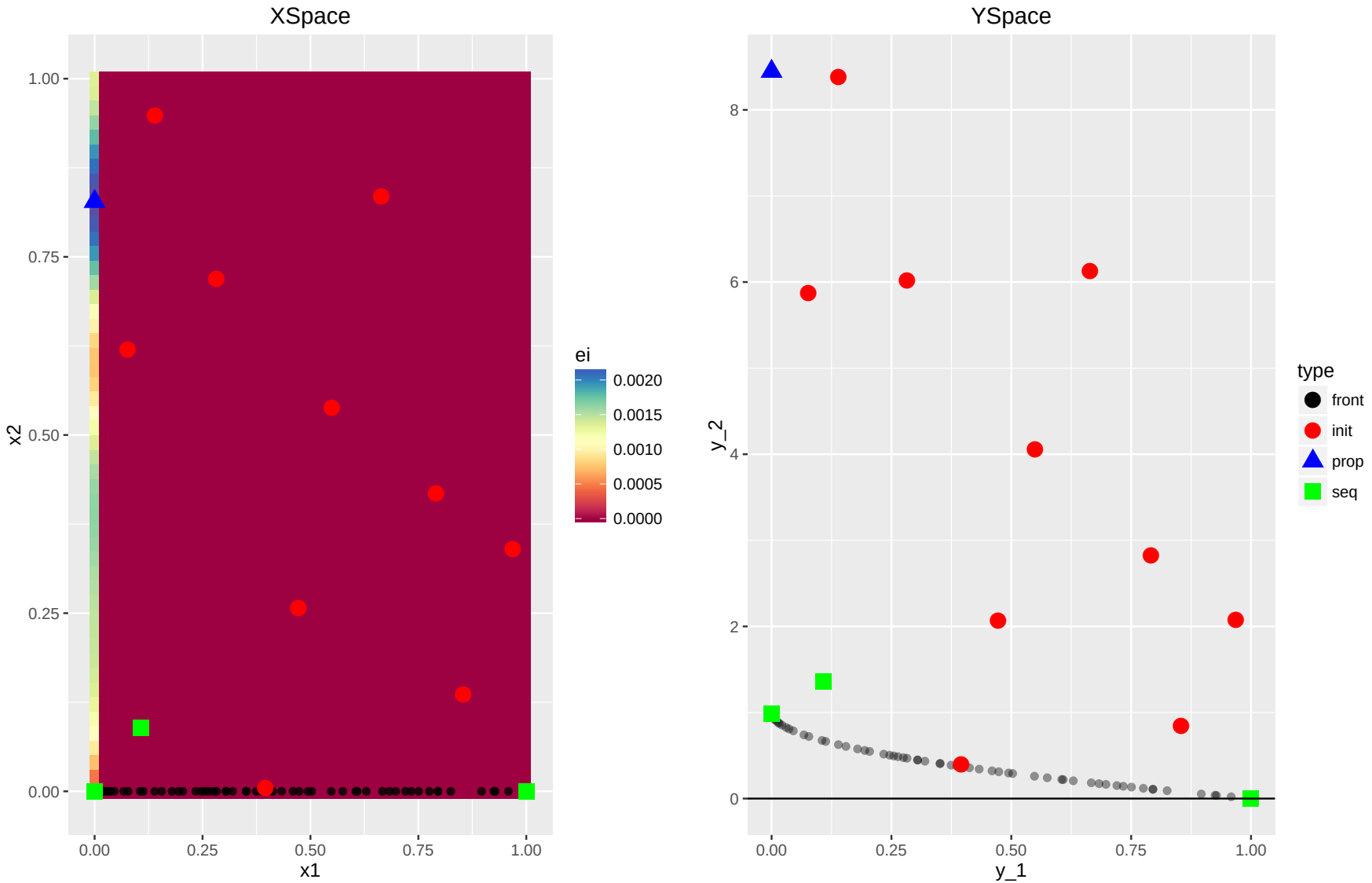
# Animation of PAREGO



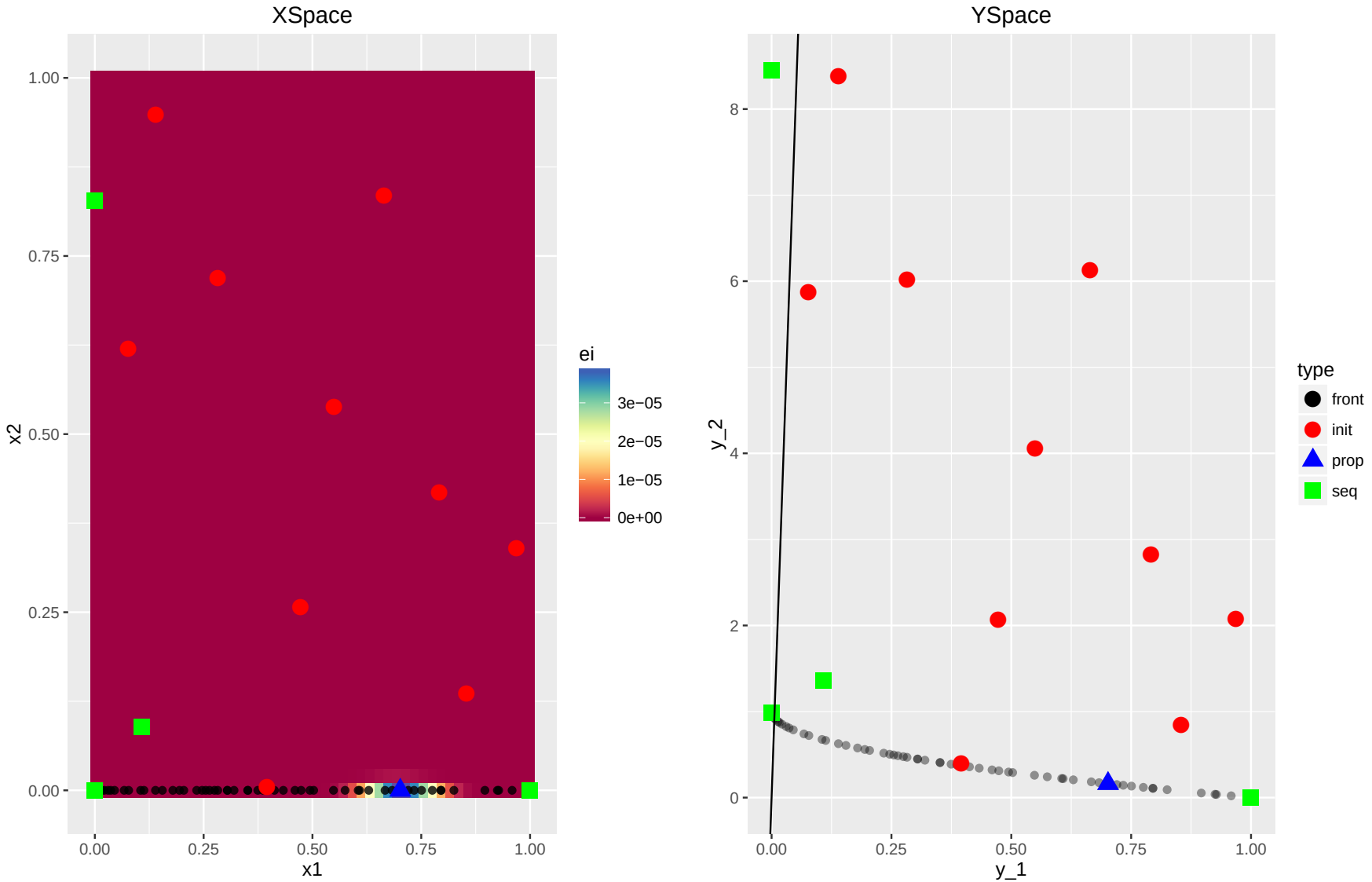
# Animation of PAREGO



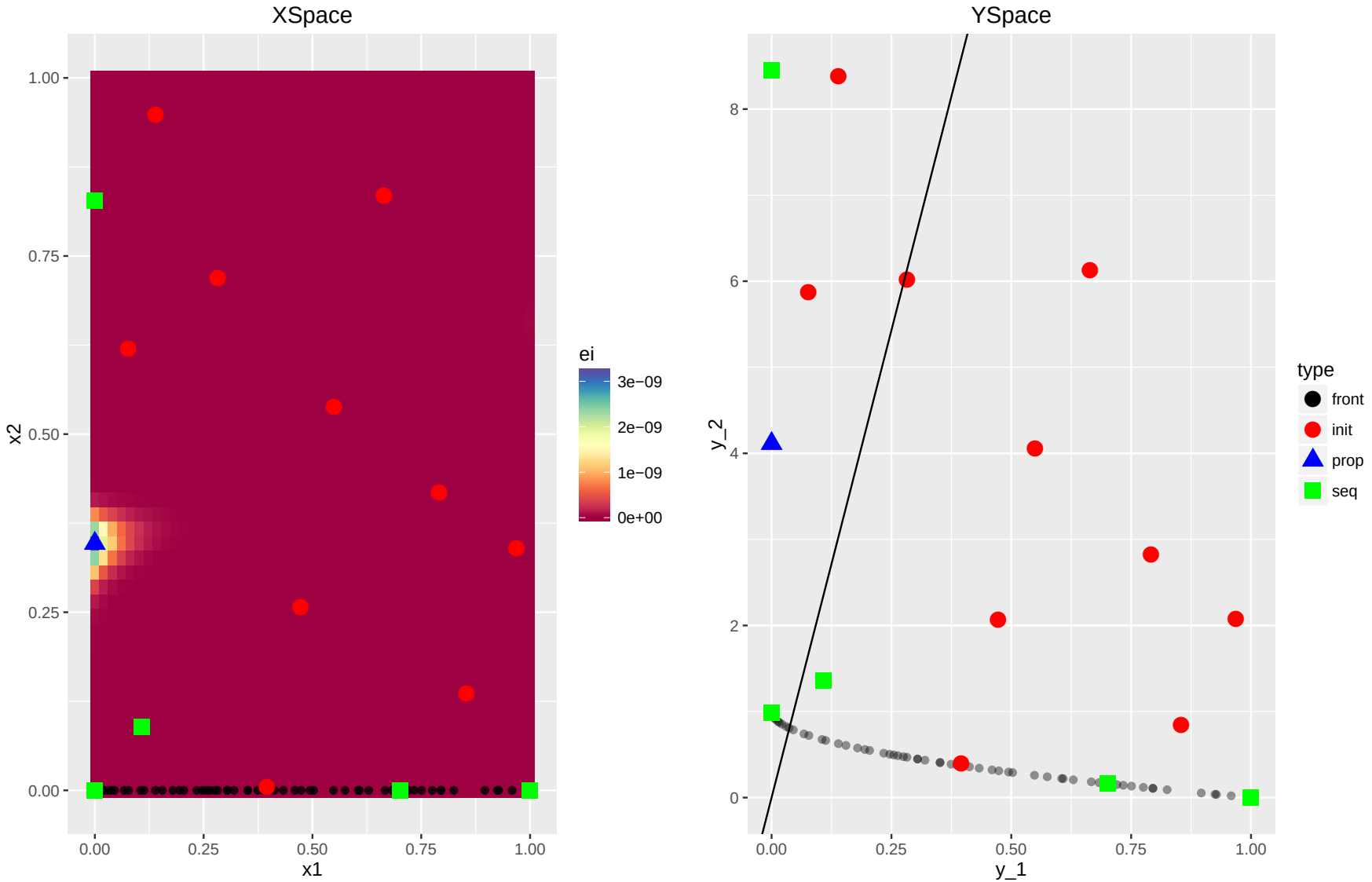
# Animation of PAREGO



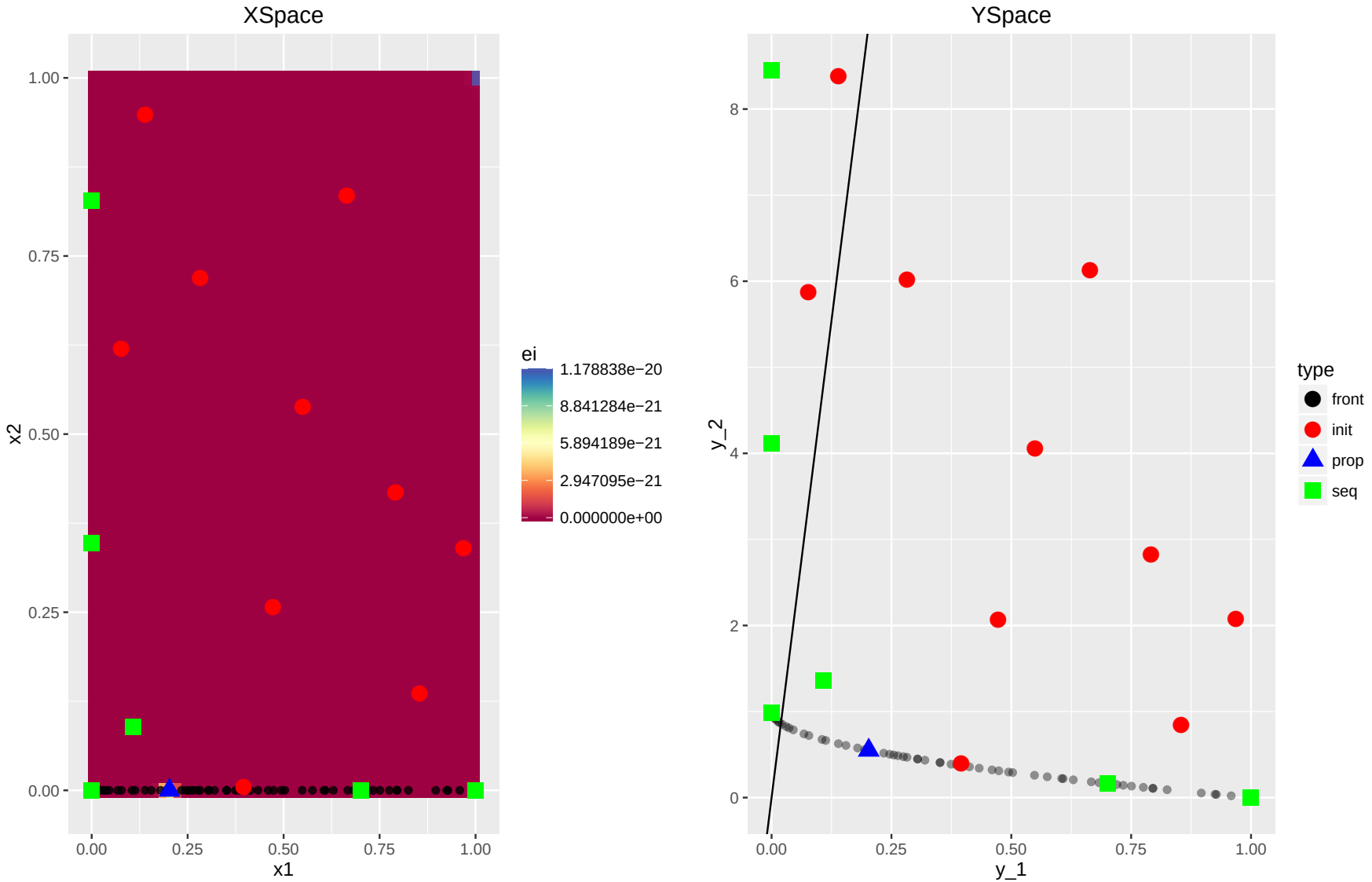
# Animation of PAREGO



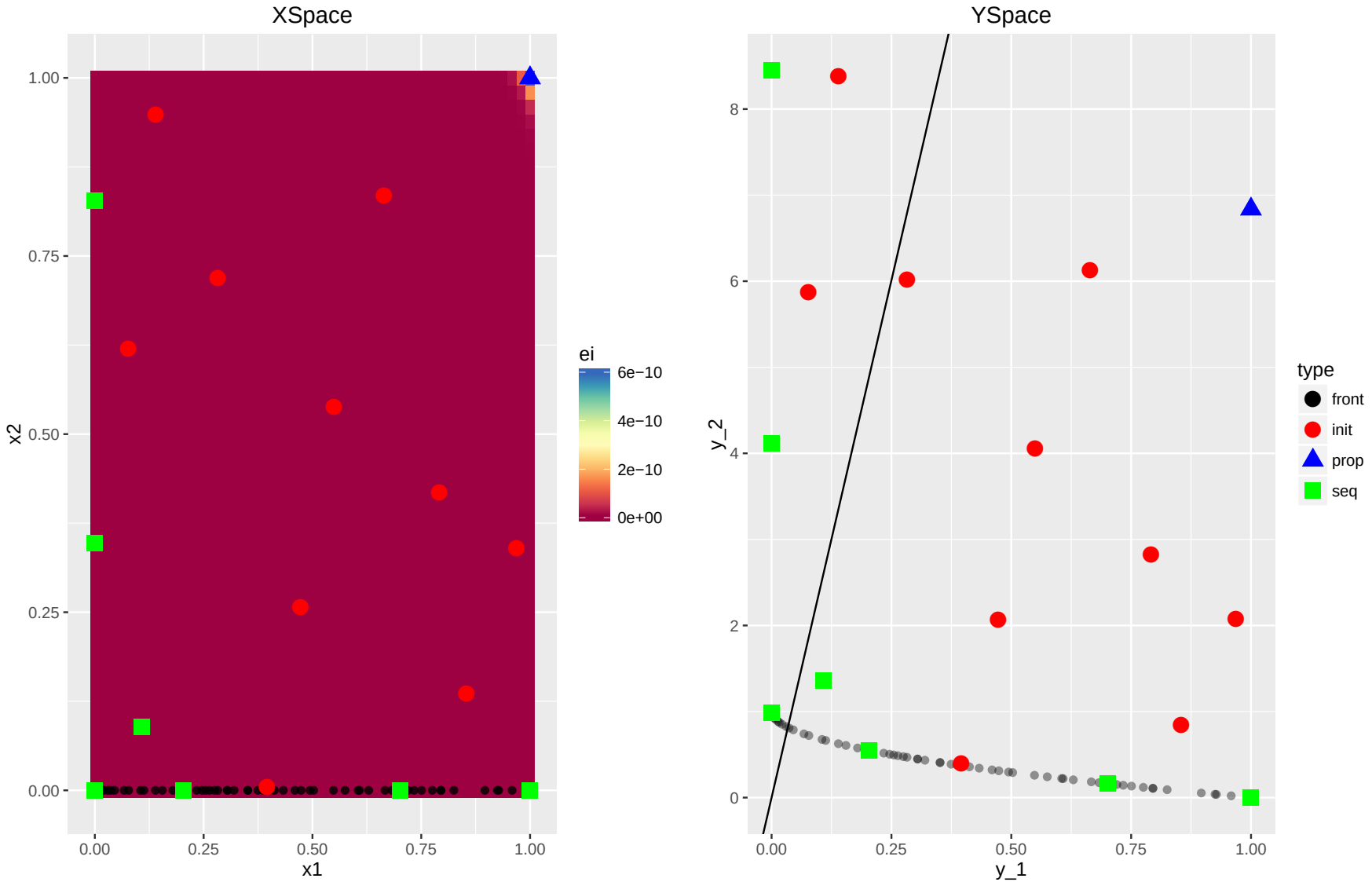
# Animation of PAREGO



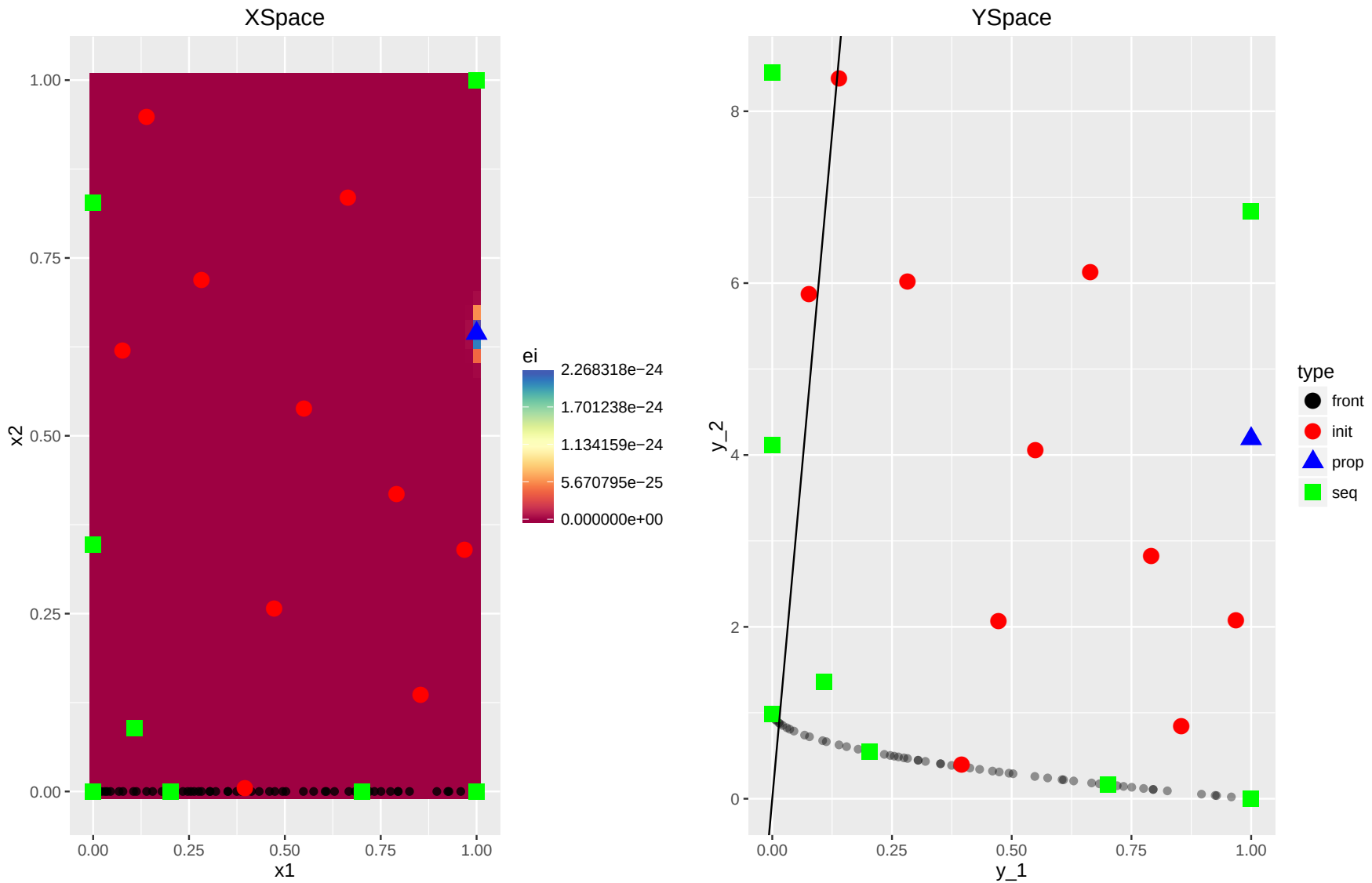
# Animation of PAREGO



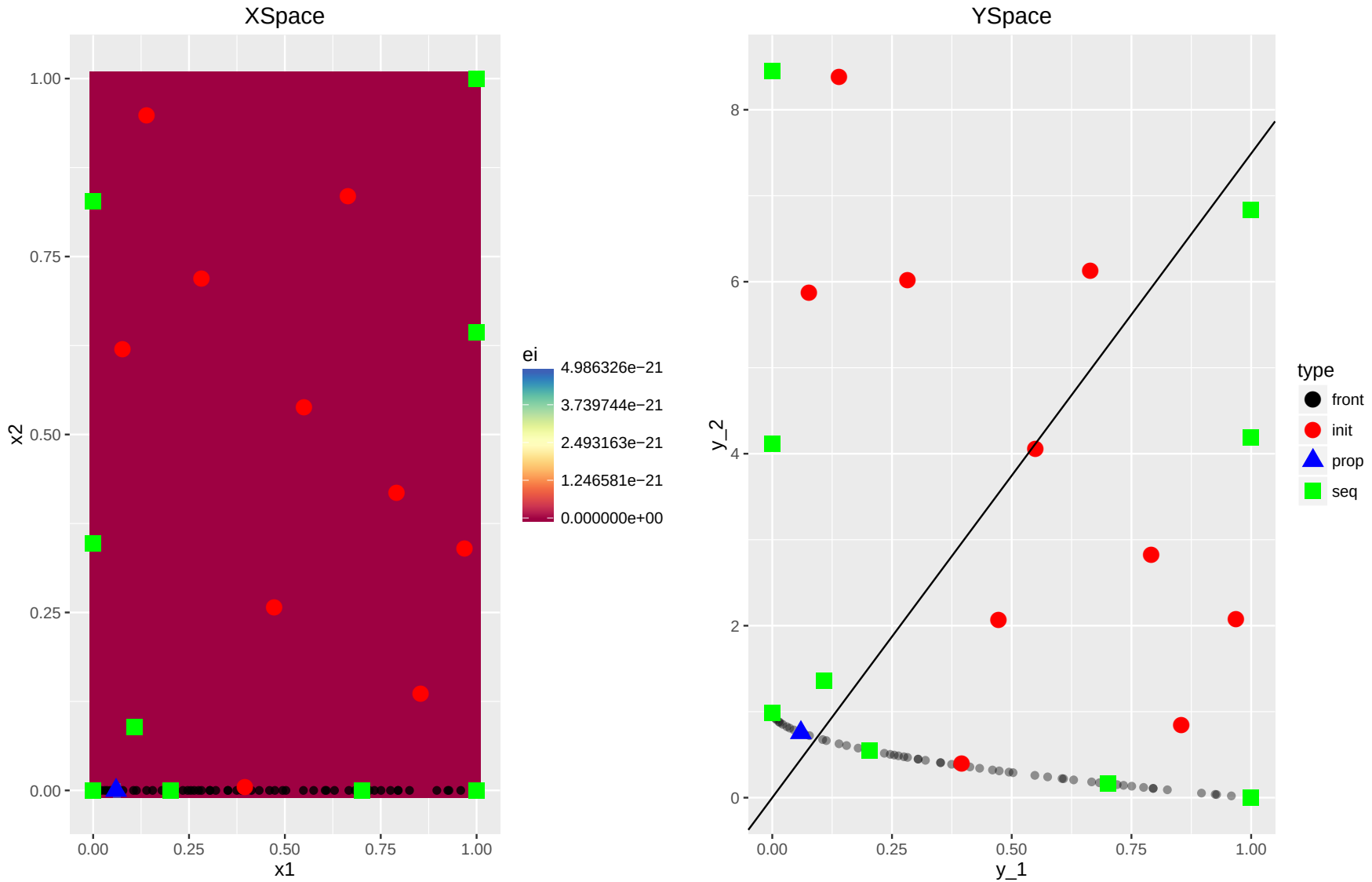
# Animation of PAREGO



# Animation of PAREGO



# Animation of PAREGO



# The design of our study

The parameters $(C, \gamma)$ of the SVM itself were optimized over $2^{[-15,15]}$ respectively. Every solver has further approximation parameters:

| SVM solver | Parameters | Optimization Space |
|---|---|---|
| LLSVM | Matrix rank | $2^{[4,11]}$ |
| LIBSVM | $\epsilon$ (Accuracy) | $2^{[-13,-1]}$ |
| LASVM | $\epsilon$ (Accuracy), #Epochs | $2^{[-13,-1]}$, $2^{[0,7]}$ |
| LIBBVM/CVM | $\epsilon$ (Accuracy) | $2^{[-19,-1]}$ |
| SVMperf | $\epsilon$ (Accuracy), #Cutting planes | $2^{[-13,-1]}$, $2^{[4,11]}$ |

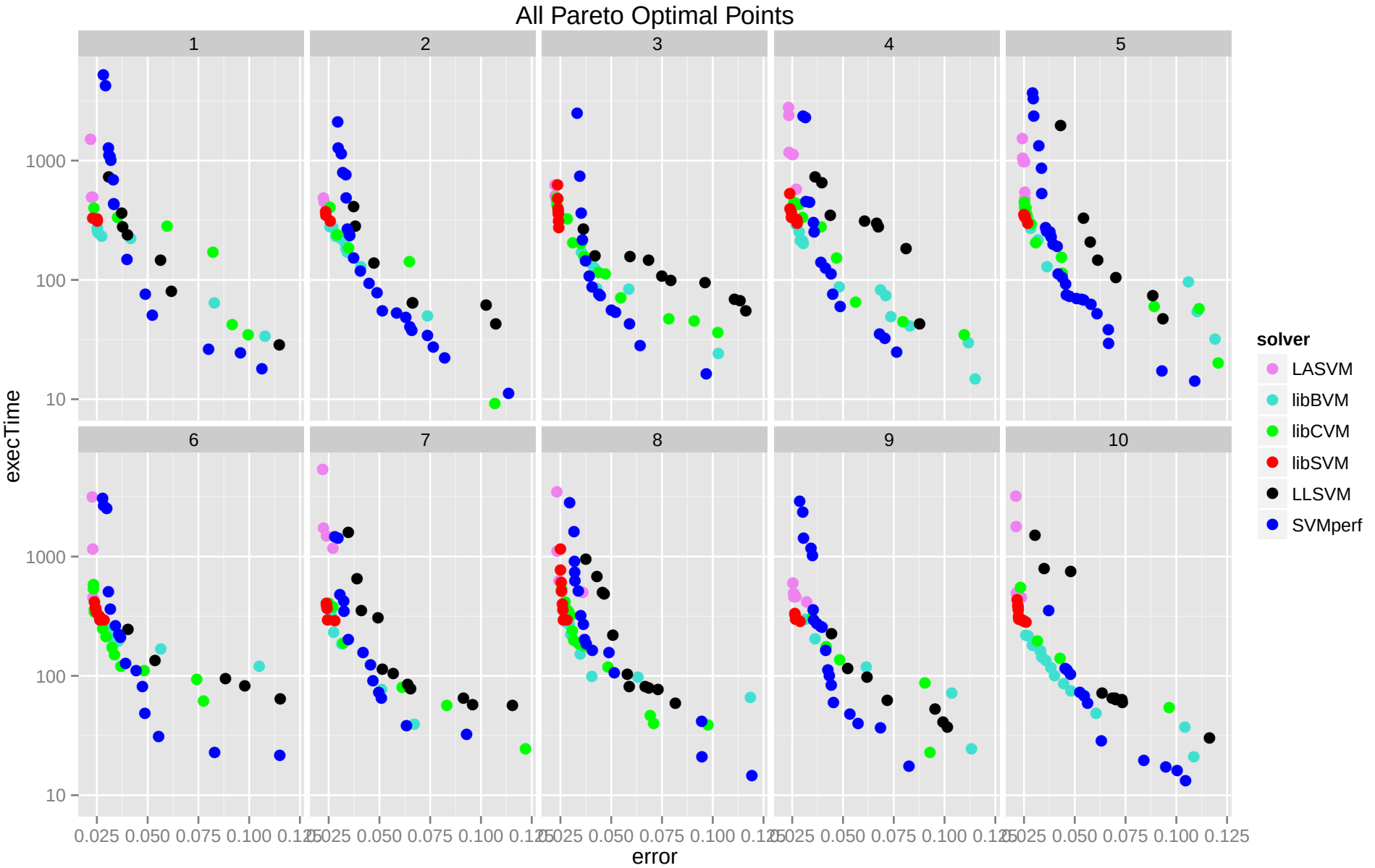
Additional parameters set to default values.

# Datasets

| data set | # points | # features | class ratio | sparsity |
|---|---|---|---|---|
| wXa | 34 780 | 300 | 34.45 | 95.19 % |
| aXa | 36 974 | 123 | 3.17 | 88.72 % |
| protein | 42 153 | 357 | 1.16 | 71.46 % |
| mnist | 70 000 | 780 | 0.96 | 80.76 % |
| vehicle | 98 528 | 100 | 1.00 | 0 % |
| shuttle | 101 500 | 9 | 0.27 | 0.23 % |
| spektren | 175 090 | 22 | 0.80 | 0 % |
| ijcnn1 | 176 691 | 22 | 9.41 | 40.91 % |
| arthrosis | 262 142 | 178 | 1.19 | 0.01 % |
| cod-rna | 488 565 | 8 | 2.00 | 0.02 % |
| covtype | 581 012 | 54 | 1.05 | 78 % |
| poker | 1 025 010 | 10 | 1.00 | 0 % |

Table : Overview of the data sets.

# Example: MNIST (appr. 2 weeks computation time)



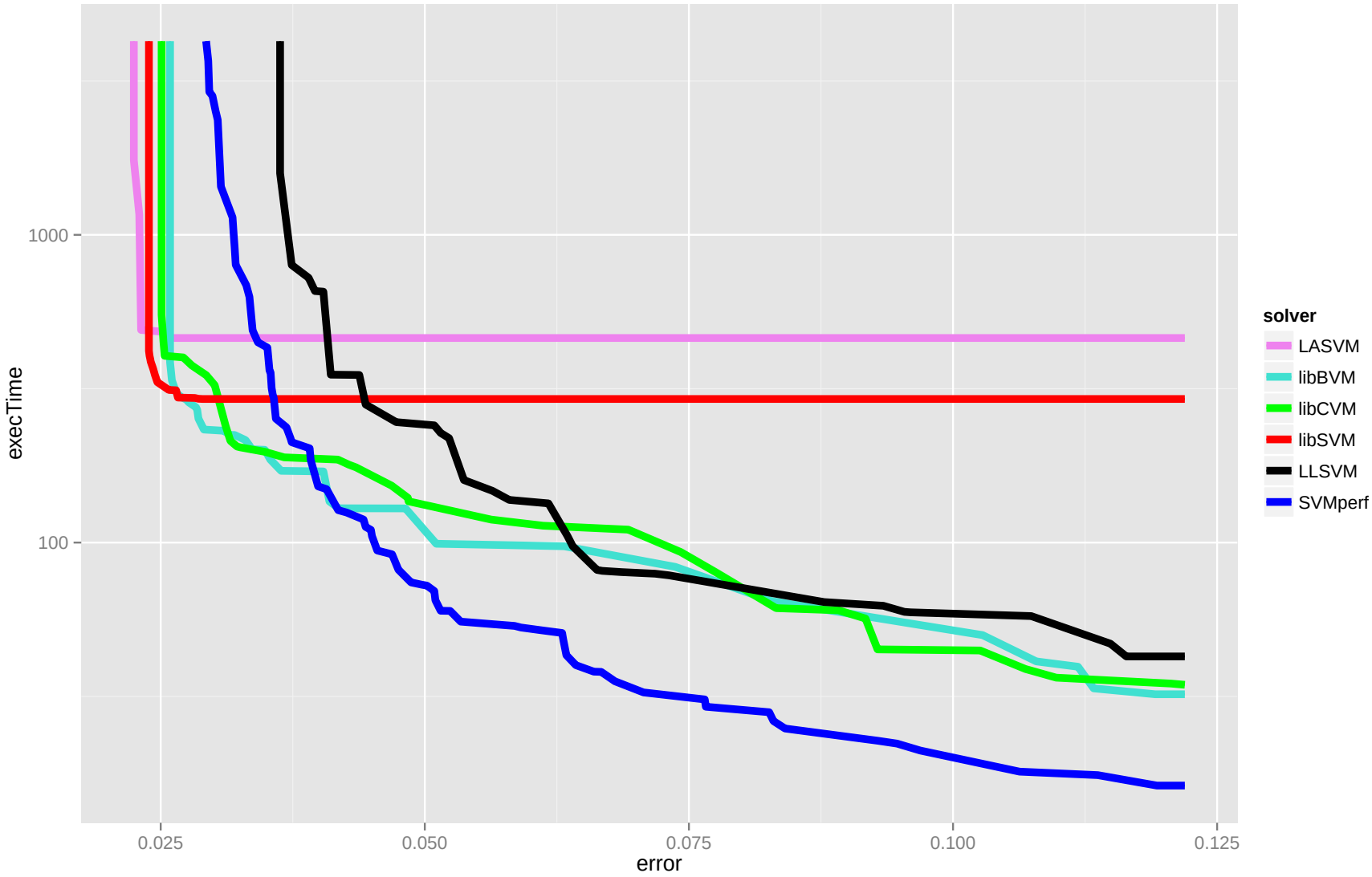
# Further Analysis of the Results

- Too much information – systematic analysis needed
- Look at the common Pareto front

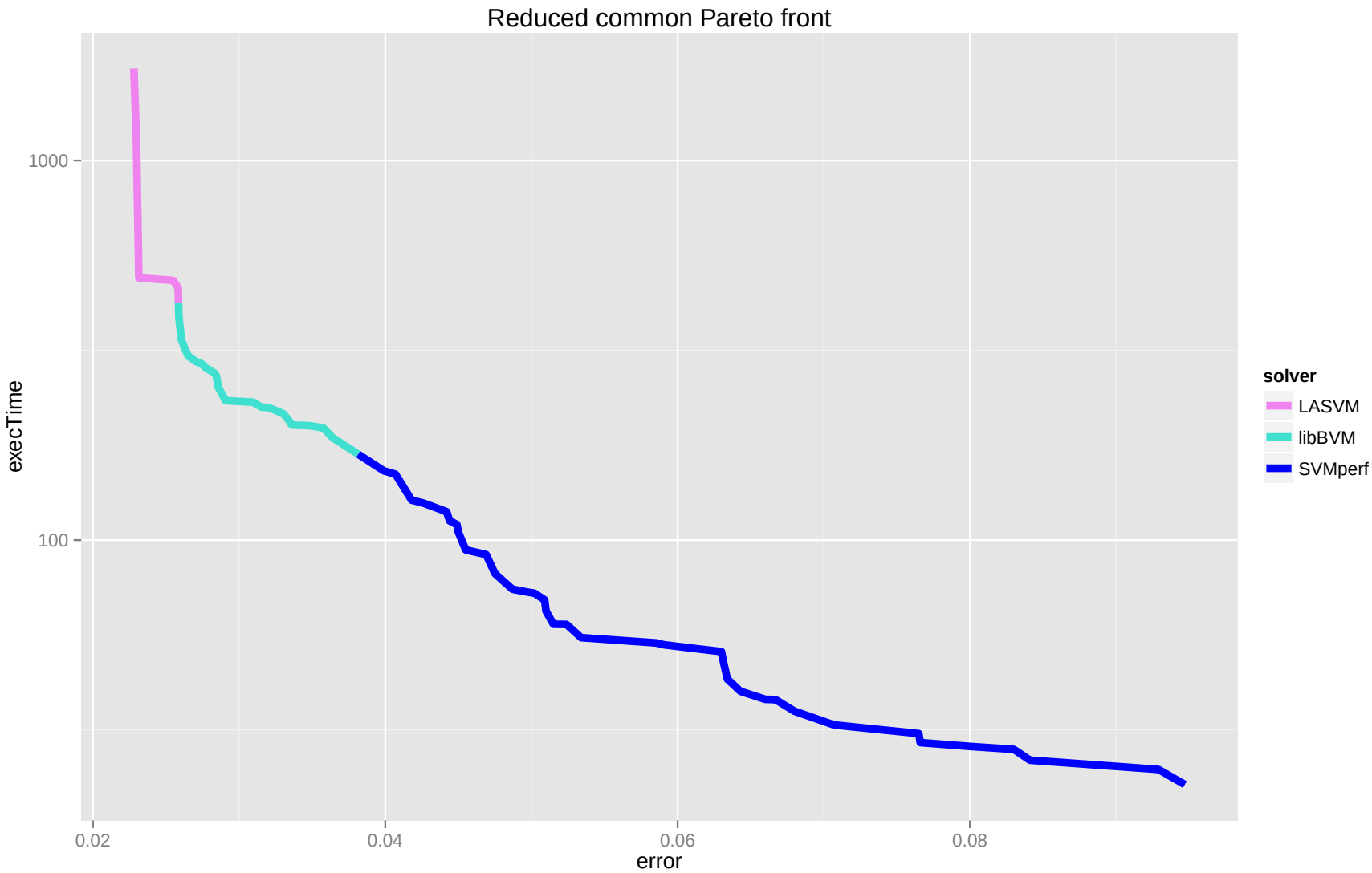
**The Common Pareto Front:** For every given trade-off give the best algorithm from a portfolio of algorithm. The portfolio should be as small as possible while the corresponding front should be as good as possible.

- Remove completely dominated solvers
- Calculate the empirical attainment function (eaf)
- Select *best* subset of solvers with respect to the Augmentend Tschebyscheff Norm
- Decide, which solver covers which part of the common front

# Example: MNIST



# Example: MNIST



# Batch Proposals

**Sequential Computation Time of the MNIST example:**

- 10 Replications · 6 Algorithms · 220 Function Evaluations · 1 hour per function evaluation = 13 200 hours ≈ 2 years
- Parallelization over experiments: 220h per experiment → our Batch Computer would alles only ≈ 10 parallel experiments
- Make 20 function evaluation at the same time

**Idea of Batch Proposals:**

- Propose $N_1$ points in a Batch → Evaluate all points on $N_2$ parallel systems → speed up of factor $\min(N_1, N_2)$ (ideally ...)
- Super computer allow very high $N_2$ → we need algorithms that allow high values of $N_1$
- We proposed batch mechanisms for known algorithms and benchmarked them for $N_1 = 4$

# Batch Proposal for MSPOT

1. Individual models for each objective
2. Multi-obj. optimization of mean response on each model, e.g. using NSGA-II
3. Select final candidates from NSGA-II result based on hypervolume contribution to the current approximation

Modification: **Iterated indicator-based candidate selection**

- Select point of the candidate set having the highest contribution
- Add point to the Pareto front approximation
- Update the contributions of the remaining points
- Repeat until $N$ points for a batch evaluation have been selected

# Batch-Proposal for DIB (SMS-EGO, $\epsilon$-EGO)

1. Singl-obj. optimization of aggregating infill criterion:
   Calculate contribution of the lower confidence bound $I(\vec{x}) = \hat{\vec{y}}(\vec{x}) + \lambda \hat{\vec{s}}(\vec{x})$ (LCB) of representative solution to the current front approximation
   - **SMS-EGO:** Contribution with regard to the hypervolume indicator. For $\varepsilon$-dominated ($\preceq_\varepsilon$) solutions, and a respective penalty $\Psi(\vec{x}) = -1 + \prod_{j=1}^{m} \left(1 + (I(\vec{x}) - y_j^{(i)})\right)$ is added
   - $\varepsilon$-**EGO:** Contribution with regard to the additive $\varepsilon$-indicator

Modification: **Use simulated evaluations for candidate generation**

- The proposed point $\vec{x}^*$ is not directly evaluated, but the LCB $I(\vec{x}^*)$ is added to the current approximation without refitting the model
- Repeat until $N$ points for a batch evaluation have been found

# Benchmark: Test functions

| **Name** | $d$ | $m$ | **Internal test functions** |
|---|---|---|---|
| GOMOP-22 | 2 | 2 | Branin, 3-Hump-Camel ($\vec{x} \in [-2, 2]^2$) |
| GOMOP-25 | 2 | 5 | Branin, 3-Hump-Camel ($\vec{x} \in [-2, 2]^2$), Hartman, Goldstein-Price, 6-Hump-Camel ($x_1 \in [-2, 2]$, $x_2 \in [-1, 1]$) |
| GOMOP-52 | 5 | 2 | Hartman, Rastrigin ($\vec{x} \in [-0.5, 0.5]^5$]) |
| GOMOP-55 | 5 | 5 | Hartman, Rastrigin ($\vec{x} \in [-0.5, 0.5]^5$), Rosenbrock, Zahkharov ($\vec{x} \in [-1, 1]^5$), Powell ($\vec{x} \in [-1, 1]^5$) |
| ZDT1 | 5 | 2 | |
| ZDT2 | 5 | 2 | |
| ZDT3 | 5 | 2 | |
| DTLZ2 | 5 | 2 | |
| DTLZ2 | 5 | 5 | |

Expensive setting: Total budget of $n_{\text{total}} = 40d$, init.design $= 4d$

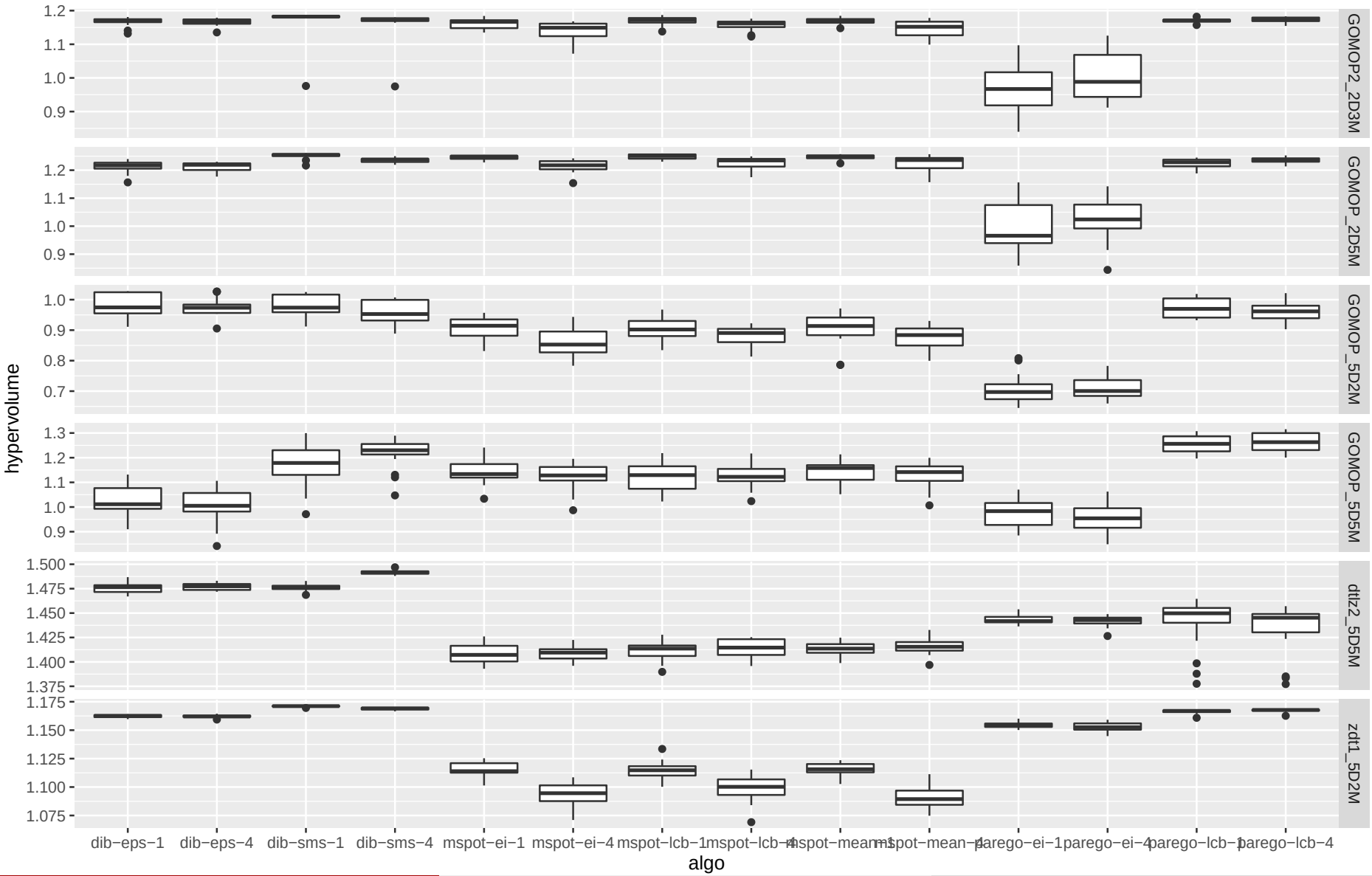
# Benchmark: Experimental Setup

- In addition to the four MBMO algorithms, Random Search and NSGA2 were run as baselines
- Every MBMO algorithm was run as single-point and 4-point variant
- ParEGO was run with LCB and EI as infill criterion, MSPOT with Mean, LCB and EI

- Reference set: union of all Pareto-optimal solutions
- All approximations and reference sets are normalized to the interval $[1, 2]^m$
- 3 indicators: unary R2, unary hypervolume, and additive $\varepsilon$

- 20 replications per run
- Significant improvements ($p = 0.05$) with respect to a pairwise Wilcoxon test
- Tests vs. baselines Random Search and NSGA-II
- Tests Singlepoint versus Multipoint

# Results


hypervolume
GOMOP2_2D3M
GOMOP_2D5M
GOMOP_5D2M
GOMOP_5D5M
dtlz2_5D5M
zdt1_5D2M
dib−eps−1
dib−eps−4
dib−sms−1
dib−sms−4
mspot−ei−1
mspot−ei−4
mspot−lcb−1
mspot−lcb−4
mspot−mean−1
mspot−mean−4
parego−ei−1
parego−ei−4
parego−lcb−1
parego−lcb−4
algo

# Toolbox: mlrMBO

We implemented the taxonomy and the 4 MBMO algorithms ParEGO, SMS-EGO, $\varepsilon$-EGO, and MSPOT as its instantiations in our R-package. Our package supports:

- Algorithms for single- und multi-objective optimization
- A large number of different infill criteria, infill optimizers, surrogate modells, ...
- A modular structure, easy to extend

# My Literature


Horn, D., Demircioglu, A., Bischl, B., Glasmachers, T., Wagner, T., and Weihs, C. (201Xa). Multi-objective selection of algorithm portfolios. *Archives of Datascience*. Submitted.

Horn, D., Demircioglu, A., Bischl, B., Glasmachers, T., and Weihs, C. (201Xb). A comparative study on large scale kernelized support vector machines. *Advances in Data Analysis and Classification*. Under Revision.

Horn, D., Wagner, T., Biermann, D., Weihs, C., and Bischl, B. (2015). Model-Based Multi-objective Optimization: Taxonomy, Multi-Point Proposal, Toolbox and Benchmark. In *Evolutionary Multi-Criterion Optimization*, volume 9018 of *Lecture Notes in Computer Science*, pages 64–78. Springer International Publishing.