Bio-Inspired Approaches to Anomaly and Intrusion Detection

Luis Martí[1] Marc Schoenauer[2]

[1] RIO Group, Instituto de Computação, Universidade Federal Fluminense.

[2] TAU Team, INRIA/Saclay, LRI/CNRS, Université Paris-Saclay.

2018 Genetic and Evolutionary Computation Conferece (GECCO 2018), Kyoto, Japan; July 2018.

ACM ISBN 978-1-4503-5764-7/18/07. https://doi.org/10.1145/3205651.3207853

Intrusion/anomaly detection

- Intrusion detection systems (IDSs),
- IDSs by means of anomaly detection,
- characteristics of the anomaly detection problem, and
- 'classical' approaches to anomaly detection.

Nature-inspired anomaly detection

- Artificial immune systems (AISs), and
- genetic programming (GP) approaches.

VorAIS/VorEAI & PAO

- VorEAI: Voronoi diagram evolutionary algorithm.
- PAO: Progressive addition of objectives to deal with VorEAI issues.

Cybersecurity

- Cybersecurity aims to minimize an *attack surface* over time.
- The attack surface is the portion of a system that has vulnerabilities.
- Attackers attempt to influence the system's nominal state and operation by varying their interactions with the attack surface in a non-compliant, and, usually, hard-to-detect manner.

Intrusion detection system (IDS)[1]

Methods for detection of intrusion attacks can be grouped in two main classes:

1. *Signature-based IDSs*, that look for *a priori* known patterns of attacks in system activities,
2. *Anomaly-based IDSs*, which model the normal behavior of the system/network under supervision and flag deviations from normal as anomalous, and hence, possible attacks.

[1] Northcutt, S. and Novak, J. (2002). *Network intrusion detection*. Sams Publishing

Signature-based IDSs

- Can detect known attacks for which patterns have been discerned.
- It is impossible for them to detect new or unknown attacks, as, by their very nature, they do not possess a known pattern for such attacks.
- This fact limits the applicability of this class of IDS.

The Internet of Things case

In IoT scenarios, where low or little maintenance can be expected and the multiplicity of devices implies that many more patterns should be elaborated than are practically discovered and maintained.

Network intrusion detection → anomaly detection

A Machine Learning problem

- Detecting intrusion by detecting anomalies.
- A particular case of **semi-supervised classification problem**.
- Data items must be tagged either as '*normal*' or '*anomalous*'.
- Datasets are heavily imbalanced: more '*normal*' than '*anomalous*'.
- Areas of input space with no data are also anomalous → capacity to repel unknown attacks.

A Multiobjective Optimization Problem (MOP)

Two classes of objective functions:

1. objectives quantifying the model as classifier (e.g., classification accuracy and recall).
2. objectives aiming at a compact representation of 'normal' data to a better detection of anomalies not present in the dataset.

What is an anomaly?

Anomaly detection

Definition

Anomaly Detection (or outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.

Types of anomalies

- Point anomalies,
- contextual anomalies,
- collective anomalies.

Anomaly detection: (more) formal definition

Anomaly detection can be posed as a particular case of the classification problem in which data items must be tagged as either 'normal' or 'anomalous'.

- relying on a dataset

$$\Psi = \left\{ \boldsymbol{x}^{(i)}, y^{(i)} \right\}$$

in which, without loss of generality, we have

$$\boldsymbol{x} \in \mathbb{R}^n \text{ and } y^{(i)} \in \{\text{normal}, \text{anomaly}\}$$

- we describe a **classifier** that correctly detects instances that correspond to each of the two categories.

Because of this fact, existing metrics devised to assess the quality of a classification algorithm are also applicable in this context.

Anomaly detection as a machine learning problem

Supervised anomaly detection

- Labels available for both normal data and anomalies.
- Similar to classification with high class imbalance.

Unsupervised anomaly detection

- No labels assumed.
- Based on the assumption that anomalies are very rare compared to normal data.
- Posed as a one-class classification problem[2].

Semi-supervised anomaly detection

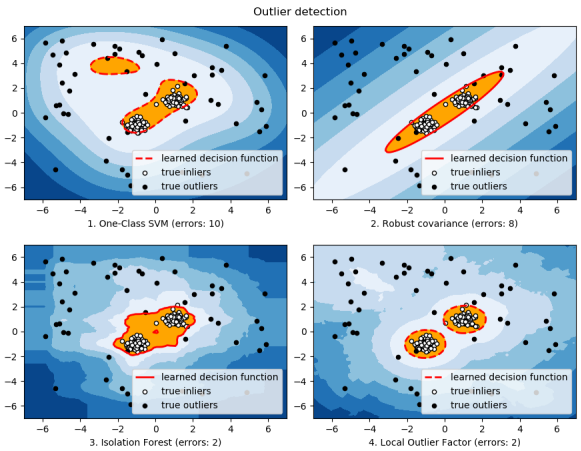
- Few labeled data,
- in some cases, labels are available only for one class of data.

The most likely case you face in a real-world scenario.

[2] Khan, S. S. and Madden, M. G. (2009). A survey of recent trends in one class classification. In *Irish Conference on Artificial Intelligence and Cognitive Science*, pages 188–197. Springer

Anomaly detection and one-class classification

- By definition, anomalies are uncommon.
- Therefore, density estimation and methods like one-class support vector machines are useful.



from `http://scikit-learn.org/stable/modules/outlier_detection.html`.

Anomaly-based IDSs have employed different statistics, machine learning and bio-inspired methods[3].

- *Distribution-based approaches*: Does data follows a pre-computed distribution?
- *Depth-based approaches*: Layers of convex hulls and flag objects in the outer layer.
- *Clustering approaches*.
- *Distance-based approaches*: How distant is an element from a subset of the elements closest to it.
- *Density-based approaches*: i.e. outlier detection by means of the local outlier factor (LOF).
- *Spectral decomposition*: Embed the data in lower dimensional subspace in which the data instances can be discriminated easily.
- *Classification approaches*: In this case, the problem is posed as the identification of which categories an observation belongs to.

[3] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15

Anomaly detection: Statistical approach

Probabilistic definition of outlier

An outlier is an object that has a low probability wrt a probability distribution model of the data.

Anomaly score function

Given a data instance $\boldsymbol{x}$ from a dataset $\mathcal{D}$,

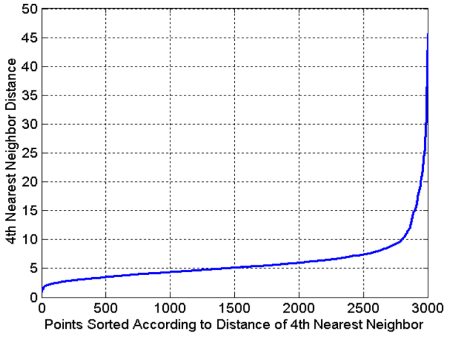
$$f(\boldsymbol{x}) = \frac{1}{P(\boldsymbol{x}|\mathcal{D})}.$$

Working principle

1. Calculate the anomaly score, $f(\boldsymbol{x})$, for each data point in the dataset.
2. Use a threshold t on this score to determine outliers. That is,

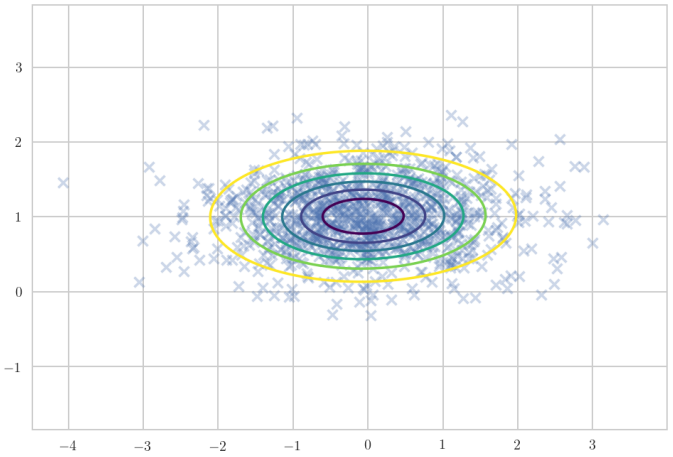
$$\boldsymbol{x} \text{ is an outlier } \iff f(\boldsymbol{x}) > t\,.$$

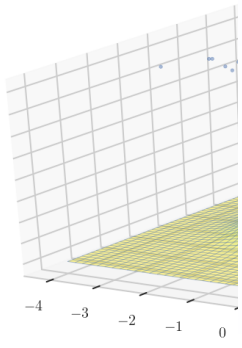
Determining threshold t



- What would be a natural choice for the value of threshold t?
- For example, assume that we want to classify 20% of the dataset instances as anomalies.
- In this case, what threshold value would you pick based on the plot above?

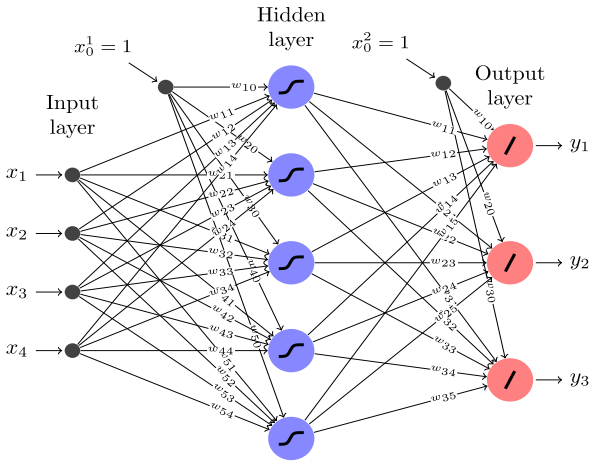
Example: Applying a normal model

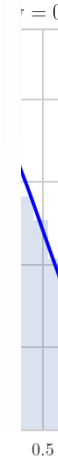




Example: Applying (stacked) MLP autoencoders

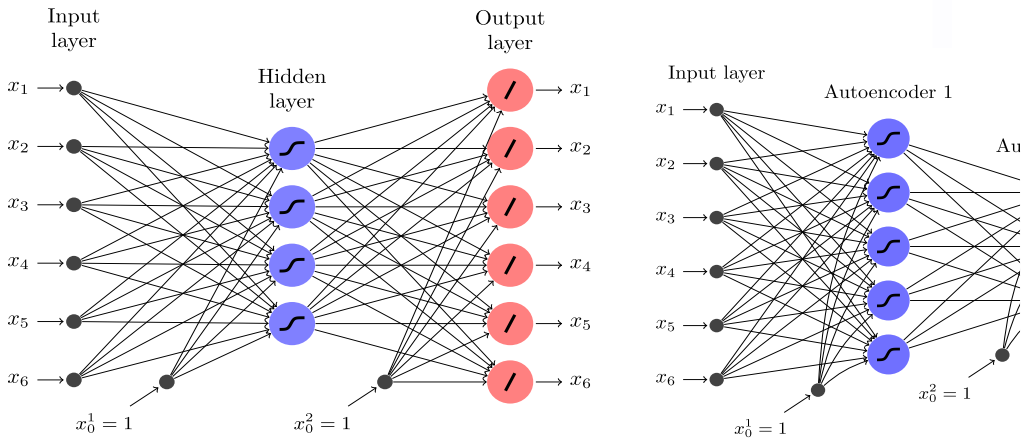
A typical Multi-Layer Perceptron (MLP)





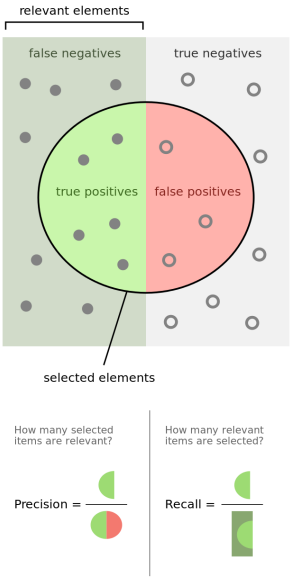
Example: Applying (stacked) autoencoders

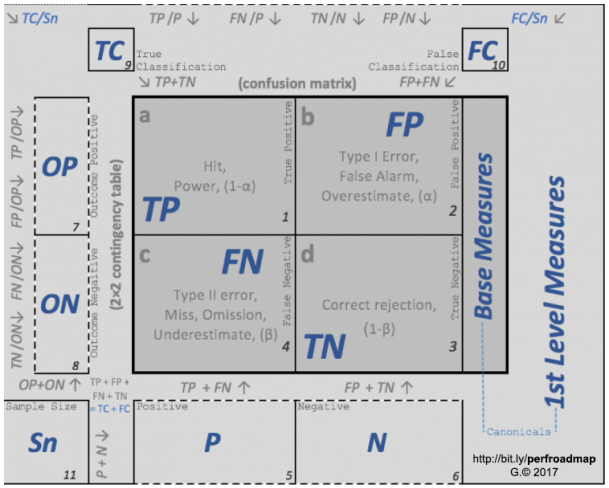
Autoencoders



If the autoencoder fails to reconstruct the input, it may be an anomaly.

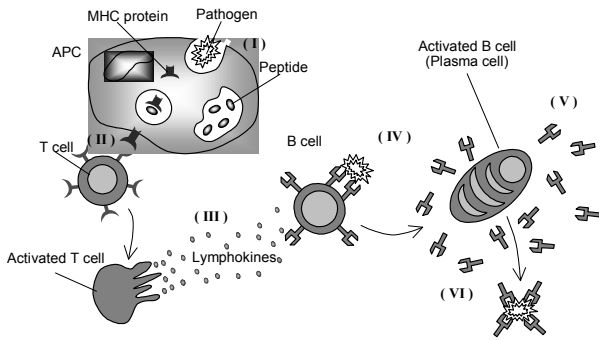
Assessing anomaly detection methods





Nature-inspired anomaly detection

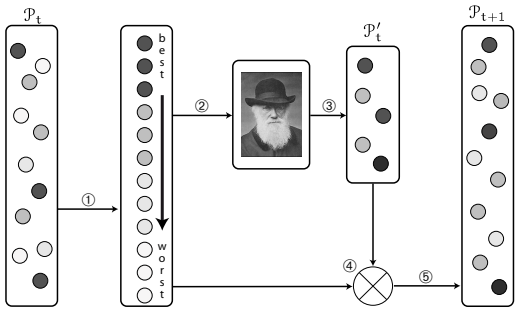
The natural immune system as an anomaly detector



from De Castro, L. N., & Von Zuben, F. J. (1999). Artificial immune systems: Part I–basic theory and applications. Universidade Estadual de Campinas, Tech. Rep, 210(1).

- Innate vs acquired immunity.
- Notion of self/non-self.
- Representation.
- Affinity.
- Negative selection.

A 'generic' evolutionary algorithm

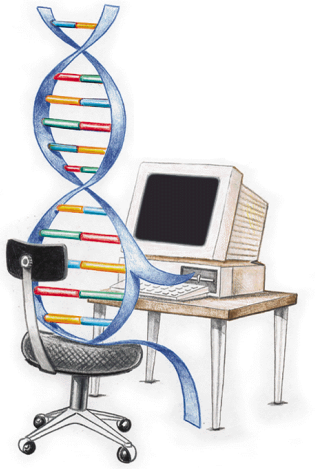


- A *population* of *individuals*,
- individuals are ranked and selected relying on a *fitness function*;
- *variation operators* inspired by the natural evolutionary process are applied, and
- individuals with better fitness have a more active role.

Requirements for an anomaly detection EA

- Spatial representation of 'normal' and 'anomaly' areas of the space,
- take into account different classification metrics → *multi-objective*,
- ability to deal with the imbalance in the dataset, and
- learn in a semi-supervised way.

Genetic programming[4]



- Individuals can be interpreted as programs.
- The results are computer programs able to perform well in a predefined task.
- Adequacy of a given individual (program) is defined by the fitness function.
- Programs can be encoded in multiple complex representation languages, like linear structures, trees and graphs.

from `http://www.genetic-programming.org`.

[4] Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*, volume 1. MIT press

Evolving anomaly detectors with genetic programming

- GP can be used to evolve classifiers,[5] and
- those classifiers can be applied for anomaly detection.
- Tree-based representation seemed the most apt representation.

- Hybrid Flexible Neural Trees for Intrusion Detection:[6] mixes genetic programming and particle swarm optimization.
- Genetic Programming Ensemble for Distributed Intrusion Detection Systems:[7] concurrently evolve decision trees using an ensemble.
- Stream Genetic Programming:[8] boosting-based distributed ensemble methods to learn from streaming data.
- One-Class Multi-Objective Genetic Programming:[9] aims to create classifiers from datasets containing only positive (non-anomalous) examples.

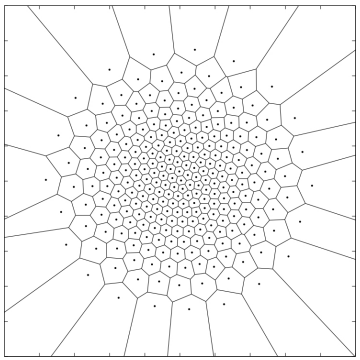
[5] Espejo, P. G., Ventura, S., and Herrera, F. (2010). A survey on the application of genetic programming to classification. *Trans. Sys. Man Cyber Part C*, 40(2):121–144

[6] Chen, Y., Abraham, A., and Yang, J. (2005). Feature selection and intrusion detection using hybrid flexible neural tree. In *International Symposium on Neural Networks*, pages 439–444. Springer

[7] Folino, G., Pizzuti, C., and Spezzano, G. (2005). Gp ensemble for distributed intrusion detection systems. In *International Conference on Pattern Recognition and Image Analysis*, pages 54–62. Springer

[8] Folino, G., Pizzuti, C., and Spezzano, G. (2007). Mining distributed evolving data streams using fractal gp ensembles. In *European*

Using Voronoi diagrams as individuals



- Voronoi diagrams are geometrical constructs that were known by ancient Greeks.
- A set of points $\{S_1, \ldots, S_m\}$, known as *Voronoi sites*, in a given n-dimensional Euclidean space $\mathcal{E}$ defines a *Voronoi diagram*, i.e.,
- a partition of the space into *Voronoi cells*.

Multi-objective optimization problem

$$\text{minimize } \boldsymbol{F}(\boldsymbol{x}) = \langle f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}) \rangle,$$
$$\text{with } \boldsymbol{x} \in \mathcal{D}.$$

- $\mathcal{D}$: feasible set — can be defined as constraints;
- $\mathcal{O}$: objective set;
- optimality — *Pareto dominance*;
- $\mathcal{D}^*$: Pareto-optimal set;
- $\mathcal{O}^*$: Pareto-optimal front, and;
- $\mathcal{P}^*$: optimizer solution.

Optimality is defined in terms of the *Pareto dominance relation*.

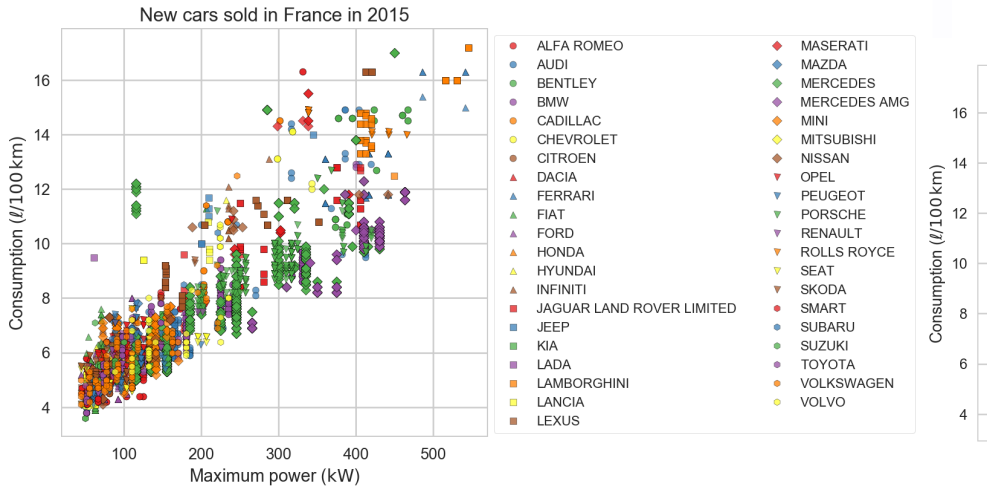
Many-objective problems
Problems with four or more objectives.

Pareto dominance

The optimality of a set of solutions can be defined based on the so-called *Pareto dominance relation*[10].

- For the optimization problem specified, and
- having $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}$.
- $\boldsymbol{x}_1$ is said to *dominate* $\boldsymbol{x}_2$ (expressed as $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$) iff
- $\forall f_j, f_j(\boldsymbol{x}_1) \leq f_j(\boldsymbol{x}_2)$ and
- $\exists f_i$ such that $f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$.

The multi-objective car example revisited[11]



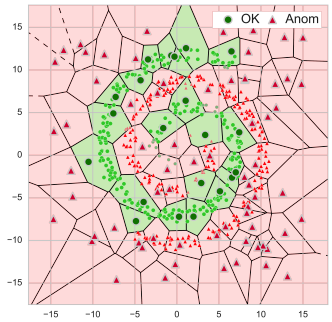
[10] Pareto, V. (1896). *Cours D'Économie Politique*. F. Rouge, Lausanne

[11] https://github.com/lmarti/evolutionary-multiobjective-optimization

Anomaly detection with VorEAl

VorEAl: Voronoi-based Evolutionary Algorithm[12]

- Evolves Voronoi diagrams.
- Encodes areas of '*normal*' or '*anomalous*' data.
- Multi-objective *alla* NSGA-II.



Classification metrics:

- accuracy and recall.

Objectives for representation:

- compactness of representation,
- total empty volume.

Individual: set of Voronoi sites, $\mathcal{I} = \{\boldsymbol{S}_i\}$ with $\boldsymbol{S_i}.\ell \in \{\text{OK}, \text{Anom}\}$

Classification: label of nearest site,

$$\text{clfy}(\mathcal{I}, \boldsymbol{x}) = \boldsymbol{S}^*.\ell \text{ with}$$

$$\boldsymbol{S}^* = \arg\min_{\boldsymbol{S}_i \in \mathcal{I}} \|\boldsymbol{x} - \boldsymbol{S}_i\| .$$

[12] Martí, L., Fansi-Tchango, A., Navarro, L., and Schoenauer, M. (2016). Anomaly detection with the Voronoi diagram evolutionary algorithm. In Handl, J., Hart, E., Lewis, R. P., López-Ibáñez, M., Ochoa, G., and Paechter, B., editors, *Proceedings of the 14th International Conference Parallel Problem Solving from Nature (PPSN XIV)*, pages 697–706, Berlin/Heidelberg. Springer International Publishing

Need of volume-based objectives

- ▷ A data instance that falls in an area not covered by learning dataset should be interpreted as an anomaly.
- ▷ Must represent known data as compact as possible.
- ▷ Relation between the volumes of the Voronoi cell and the convex hull of the training data that it contains.

Volume-based objectives: Compactness

Represent know data as compact as possible.

Relation between **volumes** of **Voronoi cells** and the **convex hulls of the data** that they contain.

$$c(\mathcal{I}) = \begin{cases} \sum_{\boldsymbol{S}_i \in \mathcal{I}} \frac{\text{vol}\left(\text{convex\_hull}\left(\mathcal{D}_{\boldsymbol{S}_i}\right)\right)}{\text{vol}(\text{cell}(\boldsymbol{S}_i))} & \text{if } |\mathcal{D}_i| > n_{\min}\,, \\ 0 & \text{in other case}\,. \end{cases}$$

- $\text{vol}(c)$: volume of convex hull c,
- $\text{cell}(\boldsymbol{S})$: Voronoi cell corresponding to site $\boldsymbol{S}$,
- $n_{\min}$: minimum diagram length, and
- $\mathcal{D}_{\boldsymbol{S}}$: subset of learning dataset classified by site $\boldsymbol{S}$

$$\mathcal{D}_{\boldsymbol{S}} = \{\boldsymbol{x} \in \Psi;\, d(x, S) \leq d(x, \boldsymbol{S}^*),\ \forall \boldsymbol{S}^* \in \mathcal{I}\}$$

Volume-based objectives: Total empty volume

Promote (big) cells that do not contain data to be labeled as 'Anom'.

- Volume of cells labeled as anomaly rated it by the number of data instances it contains.
- Sites with few data inside should become empty as the evolution takes place.

$$v(\mathcal{I}) = \sum_{\substack{\boldsymbol{S}_i \in \mathcal{I}, \\ \boldsymbol{S}_i.\ell = \text{Anom}}} \frac{\text{vol}(\text{cell}(\boldsymbol{S}_i))}{1 + 2\ln(|\mathcal{D}_{\mathcal{S}_i}| - n_{\min} + 1)}.$$

- $\text{vol}(c)$: volume of convex hull c,
- $\text{cell}(\boldsymbol{S})$: Voronoi cell corresponding to site $\boldsymbol{S}$, and
- $n_{\min}$: minimum diagram length.

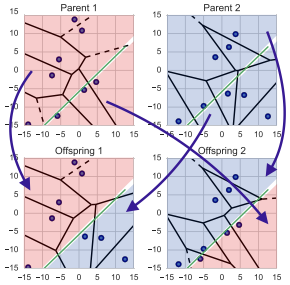
VorEAI variation operators

Mating operator

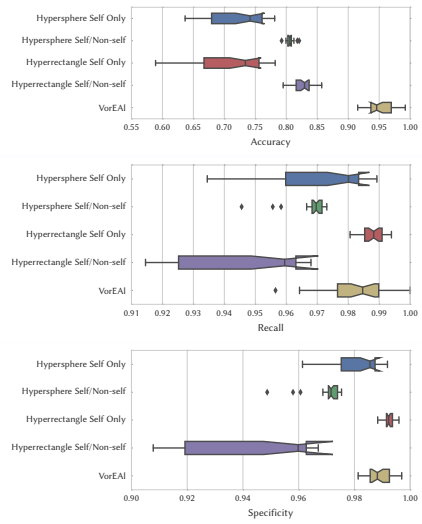
- A random cutting hyperplane is generated.
- Parents sites lying in each side of the hyperplane are exchanged.

Mutation operator

- Mutate sites' locations, similar to evolutionary strategies,
- switch site labels,
- add sites, and
- remove sites.



NSL-KDD'99 benchmark[13]



[13] NSL-KDD dataset. http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html. Accessed: 2016-02-03

NSL-KDD'99 and current state of the art

Classifier name	Accuracy (%)	FPR (%)
NSA_{sp}	72.31 ± 4.73	1.88 ± 0.94
$\text{NSA}^{+}_{\text{sp}}$	80.58 ± 0.56	2.94 ± 0.55
NSA_{re}	71.09 ± 5.57	0.76 ± 0.18
$\text{NSA}^{+}_{\text{re}}$	82.62 ± 1.60	5.46 ± 2.25
VorEAI	97.34 ± 2.54	2.95 ± 0.32
Decision Tree	81.05	N/A
Naive Bayes	76.56	N/A
Random Forest	80.67	N/A
SVM	69.52	N/A
AdaBoost	90.31	3.38
SOM	75.49	5.77
ANN	81.20	3.23
MNB+N2B	38.89	27.80
DMNB+RP	81.47	12.85
DMNB+PCA	94.84	4.40
DMNB+N2B	96.50	3.00
ERB-ANN	94.70	N/A
ERB-ANN + VQ	97.06	N/A
ANN + indicator variable and rough set	96.7	3.00

Directions for VorEAl improvement

VorEAl is currently being applied by Thalés as part of their network intrusion detection probe.

Better selection methods

- VorEAl was becoming many-objective, therefore
- better selection methods are needed.
- **SMS-EMOA**[14] selection based on hypervolume contribution.
- **NSGA-III**[15] selection based on reference points.

Adaptation in high-dimensional domains

- Number of sites/cells in individuals (Voronoi diagrams) is variable.
- Upper limit, $n_{\max}$, is impossible to set for complex problems.
- Substitute $n_{\max}$ by an objective that minimizes number of cells.

[14] Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS–EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669

[15] Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601

Adding a new objective function

This new Voronoi diagrams size minimization objective was formulated as

$$l(\mathcal{I}) = \frac{1}{1 + 0.01\left(|\mathcal{I}| - n_{\min}\right)},$$

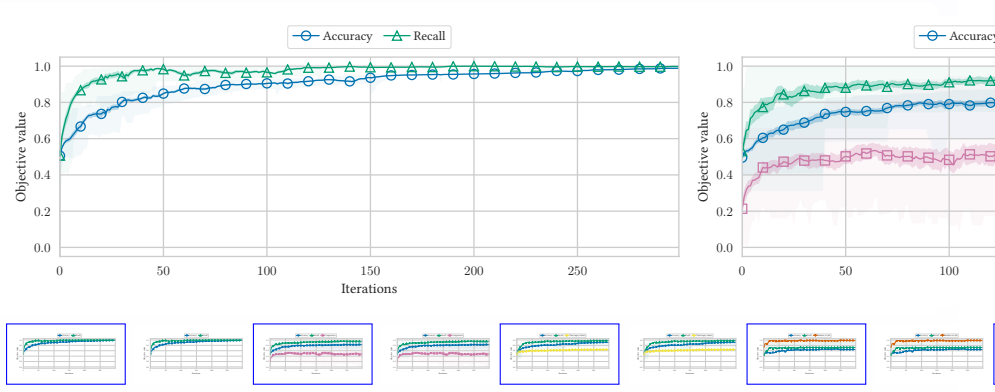
- $n_{\min}$: lower bound for individual size.

$l(\cdot)$ characteristics

- Bounded in $[0, 1]$, and
- is to be maximized -as the other objectives.
- In those aspects, better than directly using the number of sites, $|\mathcal{I}|$.

...and this is when spooky things started to happen!

Effect of adding the number of sites/cells objective



Introducing the number of sites/cells minimization objective lead to severe diversity loss and poor performance.

Reflections

The new objective degraded diversity and performance significantly.

In a 'perfect world' we could just wait as selection preserves all non-dominated individuals...

...but we are in the 'real world' and we need solutions in a viable time frame.

Hypothesis

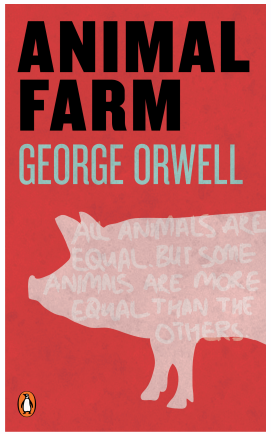
- Reducing the number of sites is very easy to attain → just create small individuals...
- ...but small individuals do not yield good performance in terms of the other objectives.

...and, this could also be happening in other contexts:

- Genetic programming → reducing bloat might create small but useless programs.
- Evolutionary machine learning → most compact structures tend to not perform well.

Our proposal: Progressive Addition of Objectives

Starting point



Paraphrasing Orwell[16]:
All objectives are important, but some objectives are more important than others!

- In real-world practice there are generally some primary objectives: the main features we want to optimize.
- Other objectives express desirable features, like minimum model size.

What if we start with the primary objectives and progressively add the secondary ones?

[16] Orwell, G. (1945). *Animal Farm.* Secker and Warburg, London, UK

MOP performance indicators

Hypervolume indicator[17]

For a set of solutions $\mathcal{A}$,

$$I_{\text{hyp}}(\mathcal{A}) = \text{volume}\left(\bigcup_{\forall \boldsymbol{a} \in \mathcal{A}} \text{hypercube}(\boldsymbol{a}, \boldsymbol{r})\right),$$

- $\boldsymbol{r}$, *reference point.*

Additive epsilon indicator[18]

- Relies on the ε-dominance concept.
- Minimum value of ε that makes set $\mathcal{A}$ ε-dominate set $\mathcal{B}$,

$$I_{\varepsilon+}(\mathcal{A}, \mathcal{B}) = \inf_{\varepsilon \in \mathbb{R}} \{\forall \boldsymbol{y} \in \mathcal{B},\ \exists \boldsymbol{x} \in \mathcal{A} \text{ such that } \boldsymbol{x} \preccurlyeq_{\epsilon+} \boldsymbol{y}\}.$$

[17] Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009). Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA'09, pages 87–102, New York, NY, USA. ACM

[18] Knowles, J., Thiele, L., and Zitzler, E. (2006). A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. revised version

Detecting convergence

Sophisticated heuristic stopping criteria are subject of intensive research[19].

On-line convergence detection criterion (OCD)[20]

- Robust and well understood method for convergence detection.
- Computes a (set of) performance indicators on consecutive populations.
- Determines if they have remained stable in a non-progress state applying a statistical hypothesis tests.

We use OCD to determine if the evolution is stagnating and, therefore, it is time to add a secondary objective.

[19] Wagner, T., Trautmann, H., and Martí, L. (2011). A taxonomy of online stopping criteria for multi-objective evolutionary algorithms. In Takahashi, R. H. C., Deb, K., Wanner, E. F., and Greco, S., editors, *6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011)*, volume 6576, pages 16–30, Berlin/Heidelberg. Springer. 10.1007/978-3-642-19893-9_2

[20] Wagner, T., Trautmann, H., and Naujoks, B. (2009). OCD: Online convergence detection for evolutionary multi-objective algorithms based on statistical testing. In Ehrgott, M., Fonseca, C. M., Gandibleux, X., Hao, J.-K., and Sevaux, M., editors, *5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009)*, volume 5467 of *Lecture Notes in Computer Science*, pages 198–215, Berlin/Heidelberg. Springer

Progressive Addition of Objectives (PAO)[21]

A general approach usable in any (many-objective) MOP.

- A greedy methodology.
- Starts with a set of primary objectives.
- Progressively select which objectives to add from the set of secondary objectives by selecting the least *disruptive* one.
- Select the objective that degrades as little as possible the convergence and diversity of the population.

- We need a function $\lambda(\cdot)$ that can be defined relying on performance indicators.
  - $\mathcal{S}$-PAO: PAO based on the hypervolume indicator.
  - ε-PAO: PAO based on the additive ε indicator.

[21] Martí, L., Fansi-Tchango, A., Navarro, L., and Schoenauer, M. (2017). Progressively adding objectives: A case study in anomaly detection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 593–600, New York, NY, USA. ACM

Some PAO notation

$$\text{minimize } f_1, \ldots, f_M;\ \mathcal{F} := \{f_1, \ldots, f_M\}.$$

- $\mathcal{F}^{\text{prim}}$, set of **primary objectives** and $\mathcal{F}^{\text{sec}}$, **set of secondary objectives**,

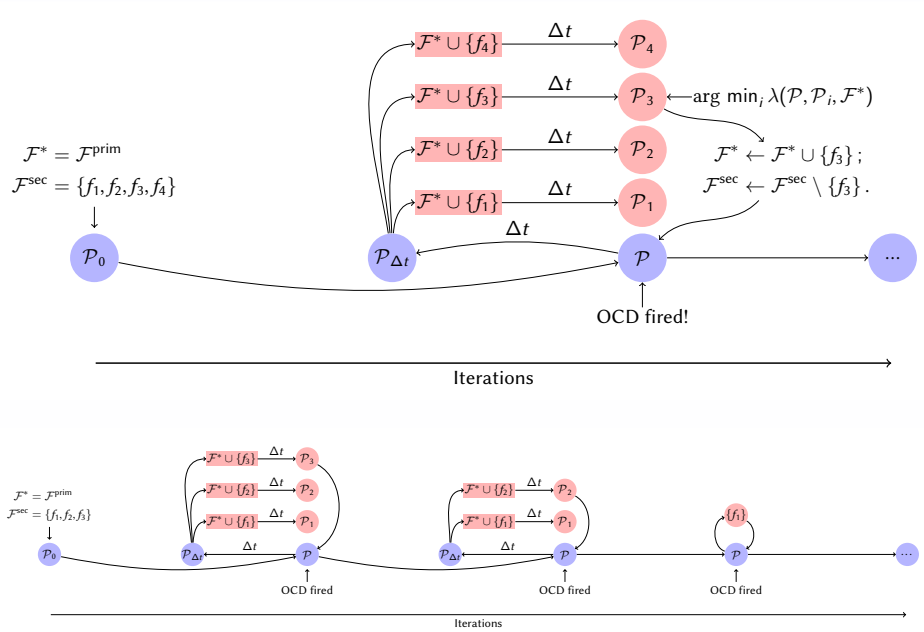
$$\mathcal{F}^{\text{prim}} \cup \mathcal{F}^{\text{sec}} = \mathcal{F};\ \mathcal{F}^{\text{prim}} \cap \mathcal{F}^{\text{sec}} = \varnothing.$$

- EA$(\mathcal{P}, \mathcal{F})$: EMOA instance with population $\mathcal{P}$ and $\mathcal{F}$ objective functions.

- $\mathcal{P}, \mathcal{P}_{\Delta t} = \texttt{evolve}(e, \Delta t)$: evolves an instance, e, until convergence is detected by the OCD method.
  - Returns the last population, $\mathcal{P}$, and $\mathcal{P}_{\Delta t}$, the one obtained Δt iterations before.
- $\mathcal{P} = \texttt{evolve}_{t_{\max}}(e, t_{\max})$: evolves an instance, e, for $t_{\max}$ iterations.
  - Returns $\mathcal{P}$, population of the last iteration.

PAO Algorithm

function PAO($\mathcal{F}^{\text{prim}}, \mathcal{F}^{\text{sec}}, \lambda, \Delta t$)
  $\mathcal{F}^{\text{prim}}$: primary objectives.
  $\mathcal{F}^{\text{sec}}$: secondary objectives.
  $\lambda(\cdot)$: performance indicator.
  Δt: rollback iterations.
  $\mathcal{P} \leftarrow$ `rand_init`.
  $\mathcal{F}^* \leftarrow \mathcal{F}^{\text{prim}}$.
  $\mathcal{P}, \mathcal{P}_{\Delta t} \leftarrow \text{evolve}\left(\text{EA}\left(\mathcal{P}, \mathcal{F}^*\right), \Delta t\right)$.
  while $\mathcal{F}^{\text{sec}} \neq \varnothing$ **do**
    for all $f_i \in \mathcal{F}^{\text{sec}}$ **do**
      $\mathcal{P}_i \leftarrow \text{evolve}_{t_{\max}}\left(\text{EA}\left(\mathcal{P}_{\Delta t}, \mathcal{F}^* \cup \{f_i\}\right), \Delta t\right)$.
    $i = \arg\min_i \lambda(\mathcal{P}, \mathcal{P}_i, \mathcal{F}^*)$.
    $\mathcal{F}^* \leftarrow \mathcal{F}^* \cup \{f_i\}$; $\mathcal{F}^{\text{sec}} \leftarrow \mathcal{F}^{\text{sec}} \setminus \{f_i\}$
    $\mathcal{P}, \mathcal{P}_{\Delta t} \leftarrow \text{evolve}\left(\text{EA}\left(\mathcal{P}, \mathcal{F}^*\right), \Delta t\right)$.
  return $\mathcal{P}$



Experimenting with PAO

Applying PAO to the improved VorEAIs

Objective

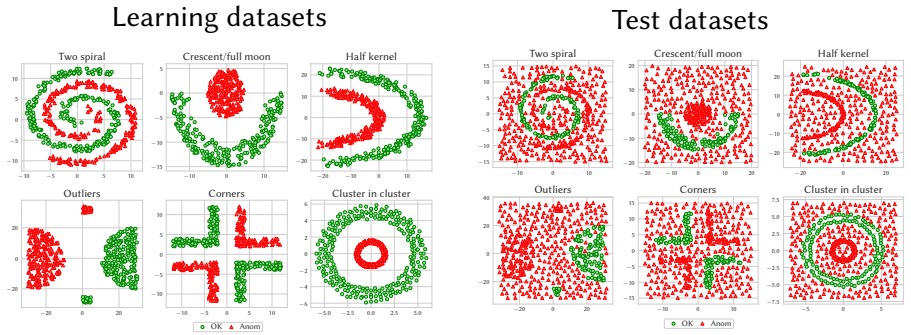
- Impact of using PAO on VorEAI.
- Involve NSGA-III and SMS-EMOA selection.
- Also included some baseline methods:
  - negative selection algorithm (NSA)[22],
  - one-class vector machines (SVMs)[23], and
  - naive Bayes classifier[24].

[22] Ji, Z. and Dasgupta, D. (2004). Real-valued negative selection algorithm with variable-sized detectors. *Lect Notes Comput Sc*, 3102:287–298

[23] Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Machine learning*, 54(1):45–66

[24] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130

Experimental setup

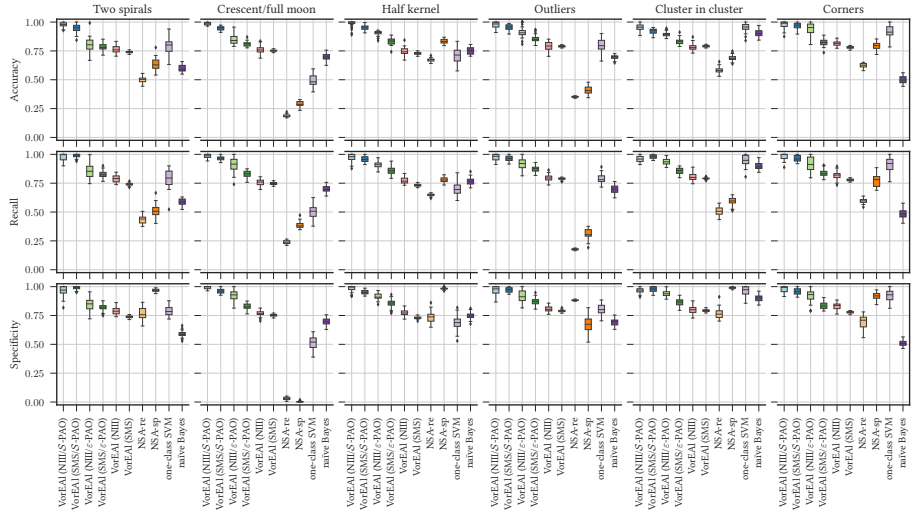


- Each problem poses a different challenge.
- Added noise in test problems validate the concept of adding volume-based objectives.

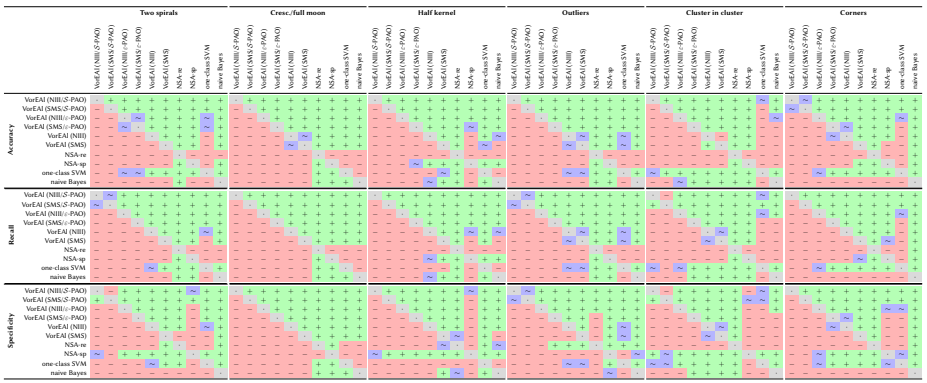
- Compute accuracy, recall and specificity.
- Bergmann–Hommel[25] statistical test procedure for assessing classifiers.

[25] Bergmann, B. and Hommel, G. (1988). Improvements of general multiple test procedures for redundant systems of hypotheses. In *Multiple Hypothesenprüfung/Multiple Hypotheses Testing*, pages 100–115. Springer

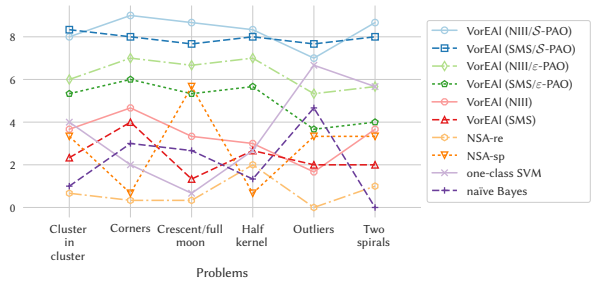
Box plots of the results

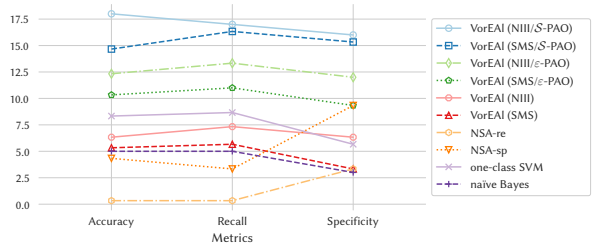


Bergmann–Hommel tests



Summarized tests results by problem and metric[26]

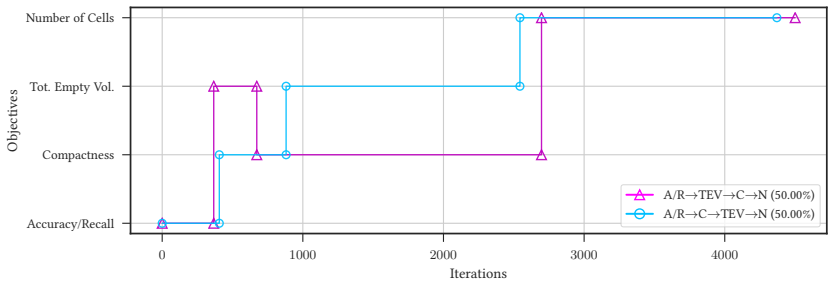




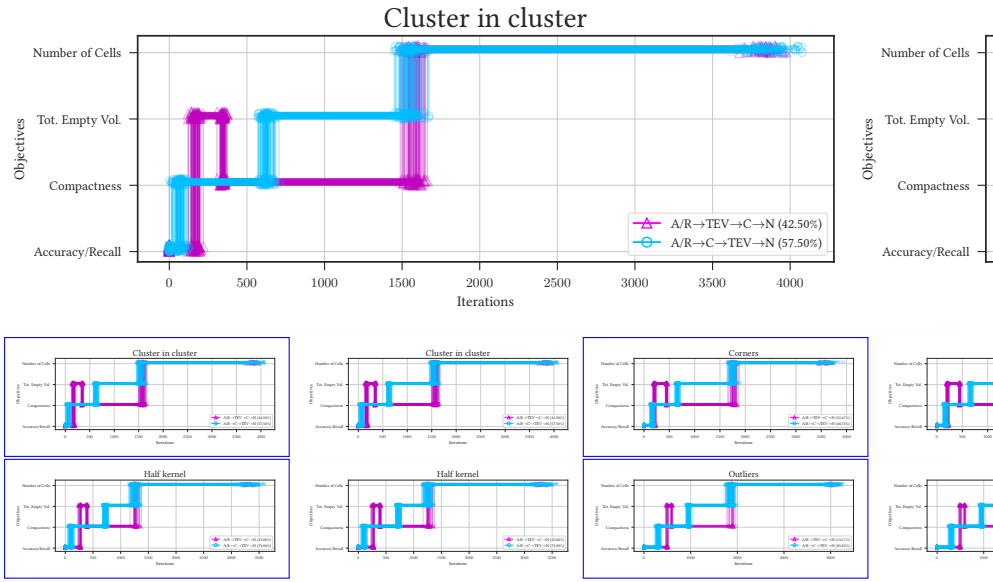
[26] Bader, J. (2010). *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. PhD thesis, ETH Zurich, Switzerland

Understanding when objectives are added

Ploting the iteration(s) when objective functions were selected by PAO in each run.



Objective addition sequences



Final remarks

- We have examined VorEAl and PAO.
- VorEAl is in use by Thalés.
- PAO provides a methodology for progressive adding objectives to complex and/or many-objective problems.
- Applied PAO on extended versions of VorEAl with substantial positive results.

Next steps

- Is PAO a new approach to many-objs?
- Can out choice for $\lambda()$ be improved?
- How to extend to other areas? GP, ML, etc.
- How to incorporate results from other areas like objective reduction?
- Preparing MOP benchmarks that can be used to analyze PAO.
- Currently applying PAO+VorEAl in more realistic datasets (NSL-KDD, ISCX 2012).
- **More theory is needed!**

Thank you! Danke sehr! Merci beaucoup! Obrigado!
¡Gracias!
Questions?

Bibliography I


NSL-KDD dataset.
`http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html`.
Accessed: 2016-02-03.


Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009).
Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point.
In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA'09, pages 87–102, New York, NY, USA. ACM.


Bader, J. (2010).
Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods.
PhD thesis, ETH Zurich, Switzerland.


Bergmann, B. and Hommel, G. (1988).
Improvements of general multiple test procedures for redundant systems of hypotheses.
In *Multiple Hypothesenprüfung/Multiple Hypotheses Testing*, pages 100–115. Springer.


Beume, N., Naujoks, B., and Emmerich, M. (2007).
SMS–EMOA: Multiobjective selection based on dominated hypervolume.
European Journal of Operational Research, 181(3):1653–1669.


Chandola, V., Banerjee, A., and Kumar, V. (2009).
Anomaly detection: A survey.
ACM Computing Surveys (CSUR), 41(3):15.


Chen, Y., Abraham, A., and Yang, J. (2005).
Feature selection and intrusion detection using hybrid flexible neural tree.
In *International Symposium on Neural Networks*, pages 439–444. Springer.


Curry, R. and Heywood, M. I. (2009).
One-class genetic programming.
In *European Conference on Genetic Programming*, pages 1–12. Springer.

Bibliography II


Deb, K. and Jain, H. (2014).
An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints.
IEEE Transactions on Evolutionary Computation, 18(4):577–601.


Domingos, P. and Pazzani, M. (1997).
On the optimality of the simple bayesian classifier under zero-one loss.
Mach. Learn., 29(2-3):103–130.


Espejo, P. G., Ventura, S., and Herrera, F. (2010).
A survey on the application of genetic programming to classification.
Trans. Sys. Man Cyber Part C, 40(2):121–144.


Folino, G., Pizzuti, C., and Spezzano, G. (2005).
Gp ensemble for distributed intrusion detection systems.
In *International Conference on Pattern Recognition and Image Analysis*, pages 54–62. Springer.


Folino, G., Pizzuti, C., and Spezzano, G. (2007).
Mining distributed evolving data streams using fractal gp ensembles.
In *European Conference on Genetic Programming*, pages 160–169. Springer.


Ji, Z. and Dasgupta, D. (2004).
Real-valued negative selection algorithm with variable-sized detectors.
Lect Notes Comput Sc, 3102:287–298.


Khan, S. S. and Madden, M. G. (2009).
A survey of recent trends in one class classification.
In *Irish Conference on Artificial Intelligence and Cognitive Science*, pages 188–197. Springer.

Bibliography III


Knowles, J., Thiele, L., and Zitzler, E. (2006).
A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers.
214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland.
revised version.


Koza, J. R. (1992).
Genetic programming: On the programming of computers by means of natural selection, volume 1.
MIT press.


Martí, L., Fansi-Tchango, A., Navarro, L., and Schoenauer, M. (2016).
Anomaly detection with the Voronoi diagram evolutionary algorithm.
In Handl, J., Hart, E., Lewis, R. P., López-Ibáñez, M., Ochoa, G., and Paechter, B., editors, *Proceedings of the 14th International Conference Parallel Problem Solving from Nature (PPSN XIV)*, pages 697–706, Berlin/Heidelberg. Springer International Publishing.


Martí, L., Fansi-Tchango, A., Navarro, L., and Schoenauer, M. (2017).
Progressively adding objectives: A case study in anomaly detection.
In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 593–600, New York, NY, USA. ACM.


Northcutt, S. and Novak, J. (2002).
Network intrusion detection.
Sams Publishing.


Orwell, G. (1945).
Animal Farm.
Secker and Warburg, London, UK.


Pareto, V. (1896).
Cours D'Économie Politique.
F. Rouge, Lausanne.

Bibliography IV


Tax, D. M. J. and Duin, R. P. W. (2004).
Support vector data description.
Machine learning, 54(1):45–66.


Wagner, T., Trautmann, H., and Martí, L. (2011).
A taxonomy of online stopping criteria for multi-objective evolutionary algorithms.
In Takahashi, R. H. C., Deb, K., Wanner, E. F., and Greco, S., editors, *6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011)*, volume 6576, pages 16–30, Berlin/Heidelberg. Springer.
10.1007/978-3-642-19893-9_2.


Wagner, T., Trautmann, H., and Naujoks, B. (2009).
OCD: Online convergence detection for evolutionary multi-objective algorithms based on statistical testing.
In Ehrgott, M., Fonseca, C. M., Gandibleux, X., Hao, J.-K., and Sevaux, M., editors, *5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009)*, volume 5467 of *Lecture Notes in Computer Science*, pages 198–215, Berlin/Heidelberg. Springer.