Explainable AI Insights for Symbolic Computation: A Case Study on Variable Ordering for Cylindrical Algebraic Decomposition

Lynn Pickering, **Tereso del Río**
Matthew England, Kelly Cohen

Aerospace Engineering and Engineering Mechanics, University of Cincinnati
Research Centre for Computational Science and Mathematical Modelling, Coventry University

26th October 2023

Coventry University

University of CINCINNATI | CCM

Safe Use of ML in Computer Algebra Systems

- Improve choices that don't affect correctness.
- Example: S-pair polynomial choice in **Buchberger**'s algorithm.
- CAS often rely on human-made heuristics.
- Early applications of ML in CAS showed performance improvements (Z. Huang et al., 2014).
- Question: Can ML contribute to gaining insights? (Davies et al., 2021).

Safe Use of ML in Computer Algebra Systems

- Improve choices that don't affect correctness.
- Example: S-pair polynomial choice in **Buchberger**'s algorithm.
- CAS often rely on human-made heuristics.
- Early applications of ML in CAS showed performance improvements (Z. Huang et al., 2014).
- Question: Can ML contribute to gaining insights? (Davies et al., 2021).

Safe Use of ML in Computer Algebra Systems

- Improve choices that don't affect correctness.
- Example: S-pair polynomial choice in **Buchberger**'s algorithm.
- CAS often rely on human-made heuristics.
- Early applications of ML in CAS showed performance improvements (Z. Huang et al., 2014).
- Question: Can ML contribute to gaining insights? (Davies et al., 2021).

Safe Use of ML in Computer Algebra Systems

- Improve choices that don’t affect correctness.
- Example: S-pair polynomial choice in **Buchberger**’s algorithm.
- CAS often rely on human-made heuristics.
- Early applications of ML in CAS showed performance improvements (Z. Huang et al., 2014).
- Question: Can ML contribute to gaining insights? (Davies et al., 2021).

Safe Use of ML in Computer Algebra Systems

- Improve choices that don't affect correctness.
- Example: S-pair polynomial choice in **Buchberger**'s algorithm.
- CAS often rely on human-made heuristics.
- Early applications of ML in CAS showed performance improvements (Z. Huang et al., 2014).
- Question: Can ML contribute to gaining insights? (Davies et al., 2021).

Explainable Artificial Intelligence(XAI)

- Offer explanations for AI decisions, enhancing user trust and effectiveness.
- As shown in Peifer et al. (2020), ML models can reveal insights.
- We plan to automatize these revelations.
- Extracting heuristics from complicated models.

Explainable Artificial Intelligence(XAI)

- Offer explanations for AI decisions, enhancing user trust and effectiveness.
- As shown in Peifer et al. (2020), ML models can reveal insights.
- We plan to automatize these revelations.
- Extracting heuristics from complicated models.

Explainable Artificial Intelligence(XAI)

- Offer explanations for AI decisions, enhancing user trust and effectiveness.
- As shown in Peifer et al. (2020), ML models can reveal insights.
- We plan to automatize these revelations.
- Extracting heuristics from complicated models.

Explainable Artificial Intelligence(XAI)

- Offer explanations for AI decisions, enhancing user trust and effectiveness.
- As shown in Peifer et al. (2020), ML models can reveal insights.
- We plan to automatize these revelations.
- Extracting heuristics from complicated models.

CAD and variable ordering

We have done this for CAD but could be replicated or adapted for any choice in any algorithm

CAD is an algorithm that breaks the space into regions.
Variable ordering can have a huge impact on its complexity C. W. Brown and J. H. Davenport (2007).

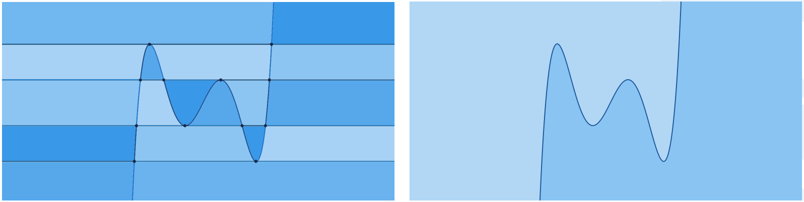

Figure: CADs of $\{x^5 + 5x^4 + 5x^3 - 5x^2 - 6x - 2y\}$. 57 cells using ordering $x \succ y$. 3 cells using the ordering $y \succ x$.

CAD and variable ordering

CAD is an algorithm that breaks the space into regions.
Variable ordering can have a huge impact on its complexity C. W. Brown and J. H. Davenport (2007).

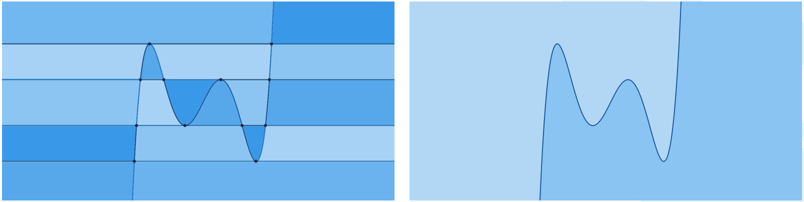

Figure: CADs of $\{x^5 + 5x^4 + 5x^3 - 5x^2 - 6x - 2y\}$. 57 cells using ordering $x \succ y$. 3 cells using the ordering $y \succ x$.

CAD and variable ordering

CAD is an algorithm that breaks the space into regions.
Variable ordering can have a huge impact on its complexity
C. W. Brown and J. H. Davenport (2007).

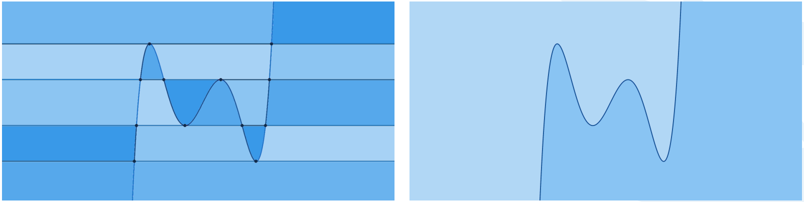

Figure: CADs of $\{x^5 + 5x^4 + 5x^3 - 5x^2 - 6x - 2y\}$. 57 cells using ordering $x \succ y$. 3 cells using the ordering $y \succ x$.

Explain basic heuristics

Brown proposed in C. W. Brown (2004) chooses the variable that minimizes these features, breaking ties with the next one:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
  (highest degree with which the variable appears)
- $\max_{polys}(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
  (highest degree of a monomial in which the variable appears)
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
  (number of monomials in which the variable appears)
- randomly

Explain basic heuristics

`Brown` proposed in C. W. Brown (2004) chooses the variable that minimizes these features, breaking ties with the next one:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
  (highest degree with which the variable appears)
- $\max_{polys}(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
  (highest degree of a monomial in which the variable appears)
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
  (number of monomials in which the variable appears)
- randomly

Explain basic heuristics

`Brown` proposed in C. W. Brown (2004) chooses the variable that minimizes these features, breaking ties with the next one:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
  (highest degree with which the variable appears)
- $\max_{polys}(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
  (highest degree of a monomial in which the variable appears)
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
  (number of monomials in which the variable appears)
- randomly

Explain basic heuristics

`Brown` proposed in C. W. Brown (2004) chooses the variable that minimizes these features, breaking ties with the next one:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
  (highest degree with which the variable appears)
- $\max_{polys}(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
  (highest degree of a monomial in which the variable appears)
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
  (number of monomials in which the variable appears)
- randomly

Explain basic heuristics

`gmods` proposed by Río and England (2022) chooses the variable that minimizes:

- $\sum_{polys}(\max_{monomials}(Degree_{x_i}))$
  (highest degree with which the variable appears in the product of the polynomials)
- randomly

Explain basic heuristics

`gmods` proposed by Río and England (2022) chooses the variable that minimizes:

- $\sum_{polys}(\max_{monomials}(Degree_{x_i}))$
(highest degree with which the variable appears in the product of the polynomials)
- randomly

Results of basic heuristics

`Brown` proposed in C. W. Brown (2004) and `gmods` proposed by Río and England (2022).

Name	Accuracy	Total time	# Completed
`gmods`	**0.563**	**7192.2**	**982.6**
`Brown`	0.553	7842.6	968.9
`random`	0.167	20797.3	262.5
`virtual-best`	1	4822.7	1019

Table: Metrics of existing heuristics in our testing dataset.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- $\ldots$
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- ...
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- ...
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- $\ldots$
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- $\ldots$
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- $\ldots$
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

Machine learning models

Being inspired by human-made heuristics, D. Florescu and M. England, 2019 proposed representing sets of polynomials as a list of features.
For example:

- $\max_{polys}(\max_{monomials}(Degree_{x_i}))$
- $\max_{polys} Sign(\max_{monomials}(TotalDegree * Sign(Degree_{x_i})))$
- $\sum_{polys}(\sum_{monomials}(Sign(Degree_{x_i})))$
- $\ldots$
- $op_{polys} op(op_{monomials}(op(Degree_{x_i})))$

Using these features, various machine learning models were trained.

XAI - One decision

From each ML decision, XAI models tell us which features had the biggest impact on the result.

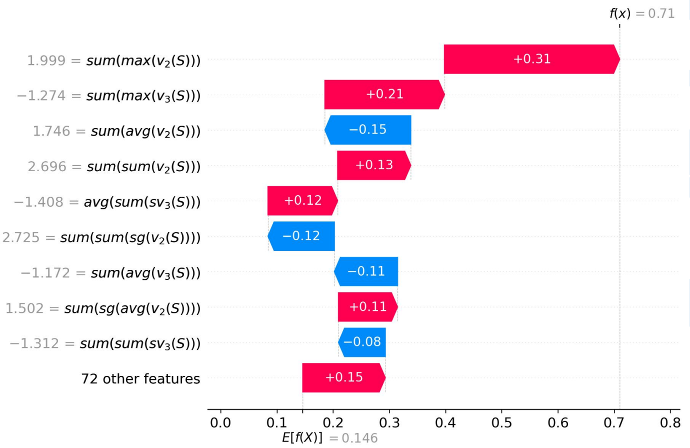

Figure: An explanation of a decision made by the MLP model on an example CAD problem instance, for the selected output ordering, ordering 5: $x_3 \succ x_1 \succ x_2$

XAI - All decisions - One model

Feature Name	Summed Impact
$sum(max(v_i(S)))$	102.58
$avg(avg(v_i(S)))$	86.814
$sum(max(sv_i(S)))$	72.671
$sum(sum(sv_i(S)))$	63.515
$avg(avg(sg(v_i(S))))$	53.735
$avg(avg(sv_i(S)))$	47.913
$sum(sum(sg(v_i(S))))$	46.66
$sum(sum(v_i(S)))$	45.988
$\ldots$	$\ldots$

Table: Features in Multi Layer Perceptron after merging those that would generate the same heuristic.

XAI - Models vote most impactful features

Feature Name	Voted Score
$sum(max(v_i(S)))$	3.333
$avg(avg(v_i(S)))$	2.167
$sum(sum(v_i(S)))$	1.158
$avg(avg(sg(v_i(S))))$	1.15
$sum(sg(sum(v_i(S))))$	0.794
$sum(max(sv_i(S)))$	0.787
$avg(avg(sv_i(S)))$	0.583
$sum(sum(sg(v_i(S))))$	0.554
⋮	⋮

Table: Voted score of merged and aggregated features across all models.

Using them as heuristics

Name	Accuracy	Total time	# Completed
`SumMaxV`	**0.563**	7192.2	982.6
`AvgAvgV`	0.544	**7138.7**	**983.5**
`SumSumV`	0.549	7524.8	975.3
`AvgAvgSgV`	0.535	8682.6	956.3
`SumSgSumV`	0.45	10836.7	922.5
`SumMaxSV`	0.509	8771.7	956.5

Table: Evaluation metrics for the new heuristics to choose the variable orderings for CAD. In bold, the best measure of the metric out of all the heuristics.

Combining them

We can combine them as tiebreakers

Name	**Accuracy**	**Total time**	**# Completed**
`Brown`	0.553	7842.6	968.9
`T1`	0.567	**6896.3**	**985.7**
`T2`	**0.583**	6896.7	984.8

Table: Evaluation metrics for the different heuristics to choose the variable orderings for CAD. In bold, the best measure of the metric out of all the heuristics. `T1`=`SumMaxV>AvgAvgV>SumSumV` and `T2`=`SumMaxV>SumSumSgV>SumSumV`

Comparison with previous state of the art

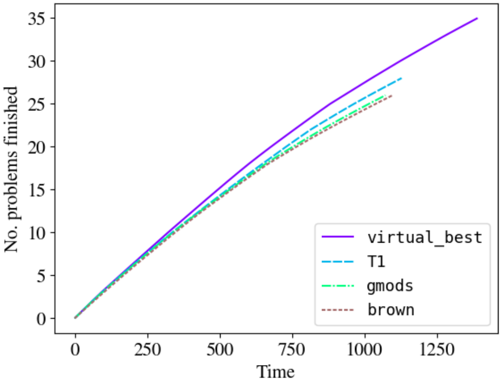

Figure: Survival plot for best-existing heuristics and new heuristics proposed in the difficult problems.

Moral of this piece of work

XAI was able to return very relevant features, and **there is nothing special about CAD!**
In Symbolic Computation, we can use XAI tools to deduce simple heuristics.
The entire paper can be read in `https://doi.org/10.48550/arXiv.2304.12154`. And the code used is freely available in Zenodo: `https://doi.org/10.5281/zenodo.8229298`

References I

C. W. Brown (2004). *Companion to the Tutorial "Cylindrical Algebraic Decomposition", presented at ISSAC '04*. URL: https://www.usna.edu/Users/cs/wcbrown/research/ISSAC04/handout.pdf.

C. W. Brown and J. H. Davenport (2007). "The complexity of quantifier elimination and cylindrical algebraic decomposition". In: *Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation*. ISSAC '07. ACM, pp. 54–60. URL: https://doi.org/10.1145/1277548.1277557.

References II

D. Florescu and M. England (2019). “Algorithmically generating new algebraic features of polynomial systems for machine learning”. In: *Proceedings of the 4th Workshop on Satisfiability Checking and Symbolic Computation (SC^2 2019)* (Bern, Switzerland). Ed. by J. Abbott and A. Griggio. CEUR Workshop Proceedings 2460, p. 12. URL: http://ceur-ws.org/Vol-2460/.

Davies, Alex et al. (2021). “Advancing mathematics by guiding human intuition with AI”. In: *Nature* 600, pp. 70–74. URL: https://doi.org/10.1038/s41586-021-04086-x.

References III

- Peifer, D., M. Stillman, and D. Halpern-Leistner (2020). “Learning Selection Strategies in Buchberger’s Algorithm”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*. Ed. by H. Daumé III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 7575–7585. URL: https://proceedings.mlr.press/v119/peifer20a.html.
- Río, Tereso del and Matthew England (2022). “New Heuristic to Choose a Cylindrical Algebraic Decomposition Variable Ordering Motivated by Complexity Analysis”. In: *Computer Algebra in Scientific Computing*. Ed. by François Boulier et al. Vol. 13366. Lecture Notes in Computer Science. Springer International Publishing, pp. 300–317. URL: https://doi.org/10.1007/978-3-031-14788-3_17.

References IV

Z. Huang et al. (2014). "Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition". In: *Intelligent Computer Mathematics*. Ed. by S. M. Watt et al. Vol. 8543. Lecture Notes in Artificial Intelligence. Springer International, pp. 92–107. URL: http://dx.doi.org/10.1007/978-3-319-08434-3_8.