Chair of Scientific Computing
Univ.-Prof. Dr. Hans-Joachim Bungartz

Technische Universität München

Bachelor Thesis Final Presentation

Exploring Fuzzy Tuning Technique for Molecular Dynamics Simulations in AutoPas

Manuel Lerchner
manuel.lerchner@tum.de

Advisors:
Manish Kumar Mishra, M.Sc. (hons)
Samuel James Newcome, M.Sc.

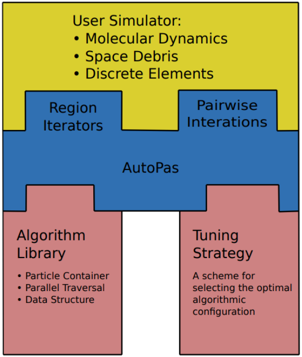
Table of Contents

What is AutoPas?

- Library for optimal node-level performance in N-body simulations
- Many different implementations for the N-body problem
- AutoTuning: Automatically switch between implementations
  - **Container:** How to find neighboring particles?
  - **Traversal:** How to efficiently handle multi-threading?
  - **Data Layout:** How to store particles in memory?
  - **Newton 3:** Can we exploit Newton's 3rd law?
  - ...
- Example applications:
  - `md_flexible` (Molecular Dynamics)
  - `sph` (Smoothed Particle Hydrodynamics)
  - Space Debris Collision Modelling

Structure of AutoPas

- Three main areas:
  - User Application
  - Algorithm Library
  - Tuning Strategies
- Algorithm Library:
  - Huge Search Space[1]
- Tuning Strategies:
  - Full Search
  - Random Search
  - Predictive Tuning
  - Bayesian Search
  - Rule Based Tuning

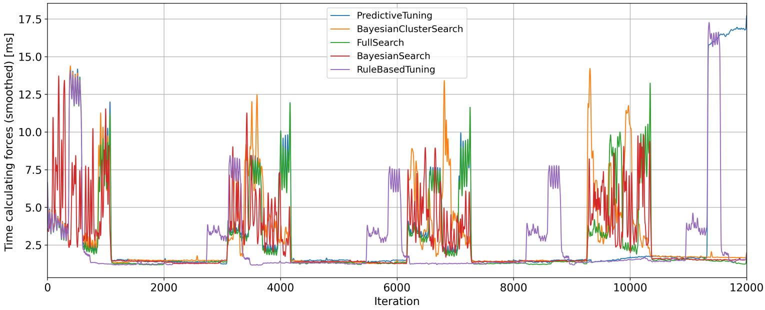


Source: [Newcome et al., 2023]

[1] Container $\times$ Traversal $\times$ Data Layout $\times$ Newton 3 $\times$ Load Estimator $\times$ Cell Size Factor

Auto-Tuning

- Tuning Phase: Find the best configuration
  - Tuning Strategies select promising configurations to evaluate
  - All those configurations are evaluated and measured
  - Huge overhead, if done badly
- Simulation Phase: Use the best configuration



Fuzzy Logic Systems

- Use human-like reasoning to model complex systems
- Linguistic Terms (*cold, hot, ...*) reduce complexity in modeling
  - Smooth transitions between terms (e.g. *cold* $\to$ *warm*)
- Fuzzy Systems are functions $f : \mathbb{R}^n \to \mathbb{R}$
- Rule based: Easy to understand, interpret and adapt
  - Interpolation effect between regions and conflicting rules
  - No hard boundaries $\to$ robust against noise

Example (Heater Control)

Input: temperature (e.g. 20°C), humidity (e.g. 50%)

Output: heater power (e.g. 50%)

Rules:

IF temp is *cold*	**AND**	humidity is *dry*	**THEN** power is *high*
IF temp is *hot*	**OR**	humidity is *wet*	**THEN** power is *low*
IF temp is *warm*			**THEN** power is *medium*

Naming Conventions

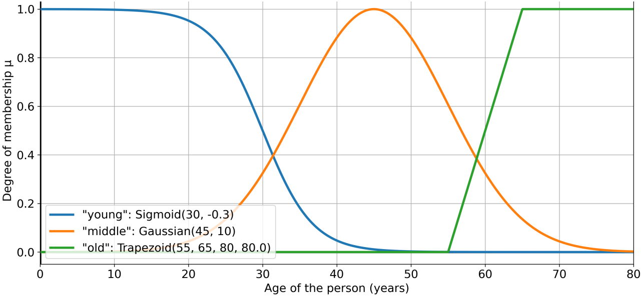
- Consider the Fuzzy Rule:

$$\underbrace{\underbrace{\textbf{IF}\ \underbrace{\text{temp}}_{\text{Ling. Var.}}\ \text{is}\ \underbrace{\textit{cold}}_{\text{Ling. Term}}\ \textbf{AND}\ \underbrace{\text{hum}}_{\text{Ling. Var.}}\ \text{is}\ \underbrace{\textit{dry}}_{\text{Ling. Term}}}_{\text{Antecedent}}\ \textbf{THEN}\ \underbrace{\underbrace{\text{power}}_{\text{Ling. Var.}}\ \text{is}\ \underbrace{\textit{high}}_{\text{Ling. Term}}}_{\text{Consequent}}}_{\text{Fuzzy Rule}}$$

- Fuzzy Logic Systems consist of:
  - Linguistic Terms / Fuzzy Sets (e.g. *cold, warm, hot*)
  - Linguistic Variables (e.g. temperature, humidity, power)
  - Fuzzy Logic Operators (e.g. **AND, OR, NOT**)
  - Fuzzy Rules (e.g. **IF** *antecedent* **THEN** *consequent*)

Fuzzy Sets vs Classical Sets

- Fuzzy Sets are generalizations of classical sets
  - U is the universe of discourse (e.g. *age* $\subseteq \mathbb{R}$)
  - Classical Set $A \subseteq U$:
    - Defined via indicator function $\chi_A : U \to \{0, 1\}$
  - Fuzzy Sets $\tilde{A} \subseteq U$:
    - Defined via **Continuous** membership function $\mu_{\tilde{A}} : U \to [0, 1]$



Linguistic Variables

- Linguistic Variables can take linguistic terms (fuzzy sets)
  - E.g. *age* can take *young*, *middle-aged* or *old*
- Instead of using crisp values (35 years), we use a combination of linguistic terms to describe the age (Fuzzyification):

$$\text{35 years} \implies \begin{cases} \text{20\% young} \\ \text{60\% middle-aged} \\ \text{0\% old} \end{cases}$$

- This allows for non-numerical reasoning:
  - E.g. **IF** age is *young* **THEN** fitness is *high*
  - Use abstract concepts instead of crisp values
- Each linguistic term represents a certain *collection* of values with varying degrees of membership

Fuzzy Logic Operators

- Fuzzy Logic Operators are used to modify/combine fuzzy sets
- Extension of boolean logic operators to real numbers
  - $\wedge : \{\textit{false}, \textit{true}\} \times \{\textit{false}, \textit{true}\} \rightarrow \{\textit{false}, \textit{true}\}$
  - **AND** $: [0, 1] \times [0, 1] \rightarrow [0, 1]$
- Fuzzy operators defined via extension of set
  - Need to maintain the classical semantics
  - Boolean logic is a special case of fuzzy logic
- Typically, Fuzzy Logic Operators are defined as:
  - **AND:** Corresponds to the intersection of fuzzy sets

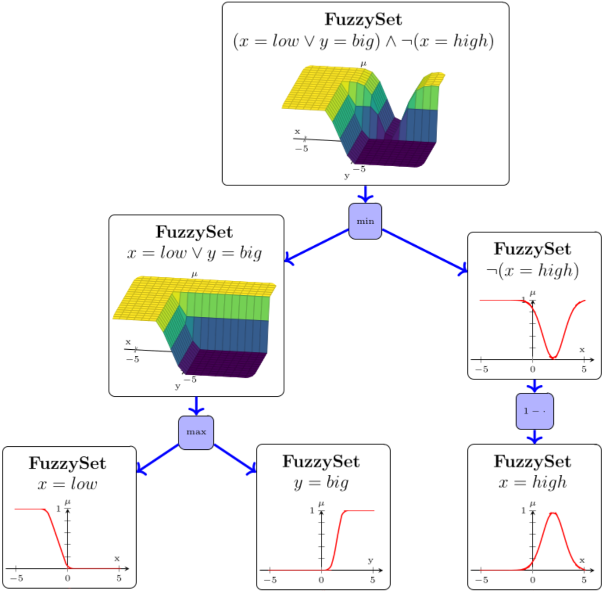
$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

  - **OR:** Corresponds to the union of fuzzy sets

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

  - **NOT:** Corresponds to the complement of a fuzzy set

$$\mu_{\neg\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x)$$


FuzzySet
$(x = low \lor y = big) \land \neg(x = high)$
min
FuzzySet
$x = low \lor y = big$
FuzzySet
$\neg(x = high)$
max
$1 - \cdot$
FuzzySet
$x = low$
FuzzySet
$y = big$
FuzzySet
$x = high$

Fuzzy Rules

- Each rule is of the form: **IF** *antecedent* **THEN** *consequent*
  - Both *antecedent* and *consequent* are fuzzy sets
  - E.g. **IF** age is *young* **THEN** fitness is *high*
- Rules are evaluated as logical implications
  - **IF** $\tilde{A}$ **THEN** $\tilde{B}$ $\iff$ $\tilde{A}$ **IMPLIES** $\tilde{B}$
  - Special form of (fuzzy) implication: Mamdani Implication
    - $\tilde{R} =$ **IF** $\tilde{A}$ **THEN** $\tilde{B}$
    - $\mu_{\tilde{R}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$
  - *Effect* of the rule is limited by the *strength* of the antecedent
- Multiple rules can act on the same linguistic variable
  - The total *effect* on the output is the aggregation/union of all individual fuzzy sets
  - Each rule *effect* is considered proportional to the *strength* of its antecedent

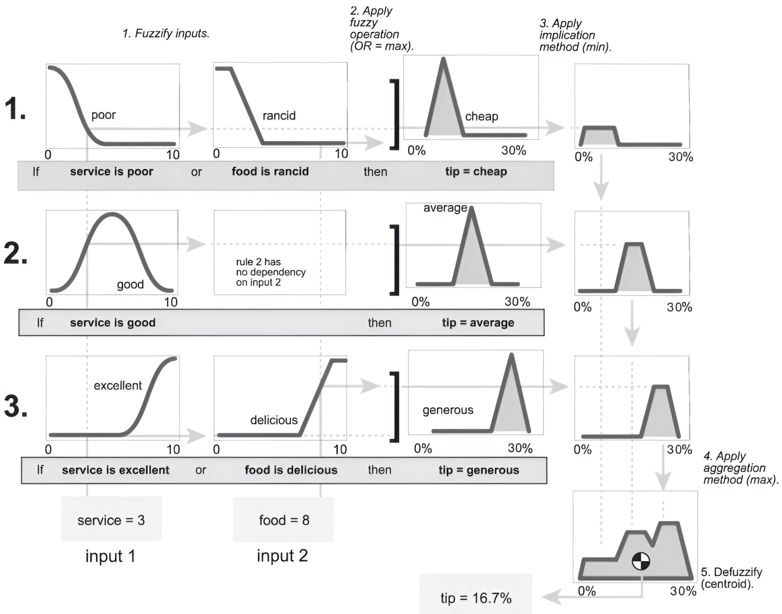
Defuzzification

- Process of converting arbitrary fuzzy sets to a crisp value
  - Special case: Fuzzy sets corresponding to a fuzzification

$$\begin{cases} \text{20\% young} \\ \text{60\% middle-aged} \\ \text{0\% old} \end{cases} \implies \text{35 years}$$

- Multiple methods for defuzzification:
  - **Center of Gravity:** Considers all values based on their membership. Finds weighted average of all possible values
  - **Mean of Maxima:** Considers only the most likely values (with highest membership). Finds the mean of all maxima
- Core idea: Represent aspects of the fuzzy set with a crisp value
  - E.g. Weighted average of all possible values (Centroid)
  - E.g. Most likely value (Mean of Maxima)

Source: MathWorks - Fuzzy Inference Process



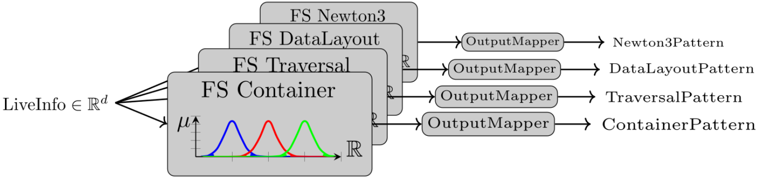
Fuzzy Tuning Strategy

- Main Idea: Use Fuzzy Logic to tune AutoPas
- Make use of LiveInfoData[2] to perform tuning
- Advantages:
  - Expert knowledge can provide powerful tuning
  - Potentially more robust against noise
  - May require fewer rules than rule-based tuning
  - Easy to interpret and understand
- Challenges and Questions:
  - What are the output variables? How to predict Configurations?
  - How to interpret the result? (Fuzzy System : $f : \mathbb{R}^n \to \mathbb{R}$)
  - How to create the fuzzy rules and linguistic terms?
    - Expert knowledge?
    - Machine Learning?

[2] Simulation state: `avgParticles/Cell`, `homogeneity`, `threadCount` ...

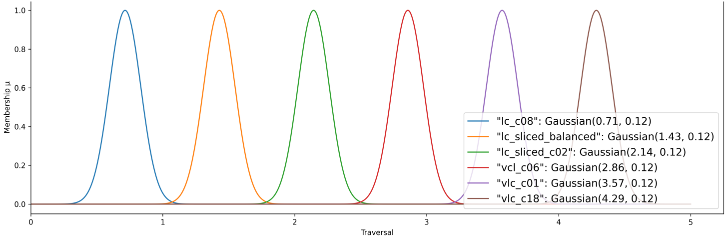
Approach 1: Component Tuning

- Predict good values/patterns for each tunable parameter separately
- A Fuzzy System for each tunable parameter
  - Output Variable: Class values of the parameter
  - Idea: Map numerical output of the Fuzzy System to the *closest* class value [Mohammed et al., 2022]
- Combine all results to find resulting configuration(s)



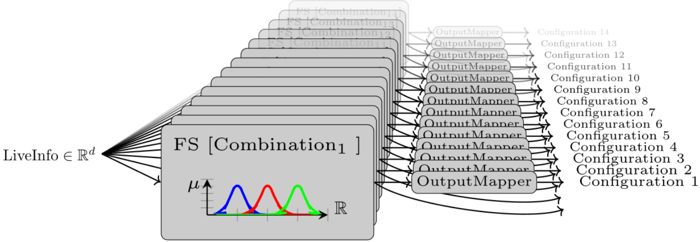
Approach 1: Component Tuning (Example)

- Style: **IF** avgParticlesPerCell is *low* **AND** threadCount is *low*
  THEN traversal is *vcl_c06*
- **Benefits:**
  - Few fuzzy systems
  - Few and natural rules
- **Drawbacks:**
  - Unused power of fuzzy logic (MoM defuzzification)
  - Independence assumption



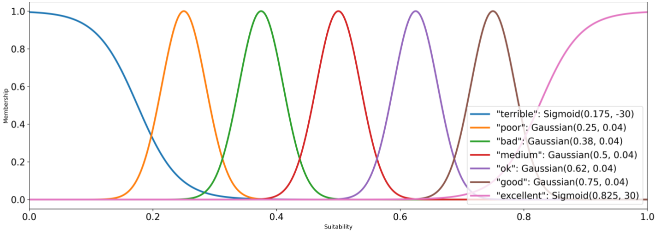
Approach 2: Suitability Tuning

- Predict the suitability of *each* configuration
- A Fuzzy System for each possible configuration
  - Output Variable: Suitability classes of the configuration
  - Directly use numerical output as suitability value
- Configuration with high suitabilities are chosen



Approach 2: Suitability Tuning (Example)

- Style: **IF** threadCount is *high* **AND** avgParticlesPerCell is *low* **THEN** suitability_LinkedCells_AoS_lc_c18_disabled is *bad*
- **Benefits:**
  - Utilizes the full power of fuzzy logic (CoG defuzzification)
  - Dependencies and incompatibilities can be modeled
- **Drawbacks:**
  - Huge number of fuzzy systems
  - Impossible to maintain by hand

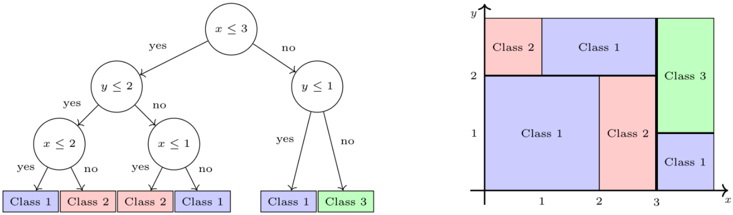


Data-Driven Rule Extraction

- Creating the Fuzzy Rules and Linguistic Terms manually is difficult
  - Expert Knowledge is required
  - Formalization of knowledge is difficult
  - Potentially many rules required
- Use Machine Learning to extract rules from data
  - Decision Tree $\rightarrow$ Fuzzy Decision Tree $\rightarrow$ Fuzzy Rules [Crockett et al., 2006]
  - Does not require expert knowledge
- Human experts can still validate the rules

Decision Trees

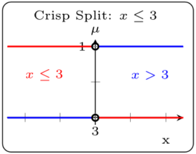
- **Idea**: Split data with axis-aligned splits to best separate classes
- Corresponds to nested *if-then-else* rules
- Easy to understand and interpret
- TFinal tree contains the entire expert knowledge
- Can be trained automatically, given enough data
    - No expert knowledge required!

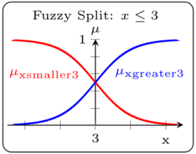


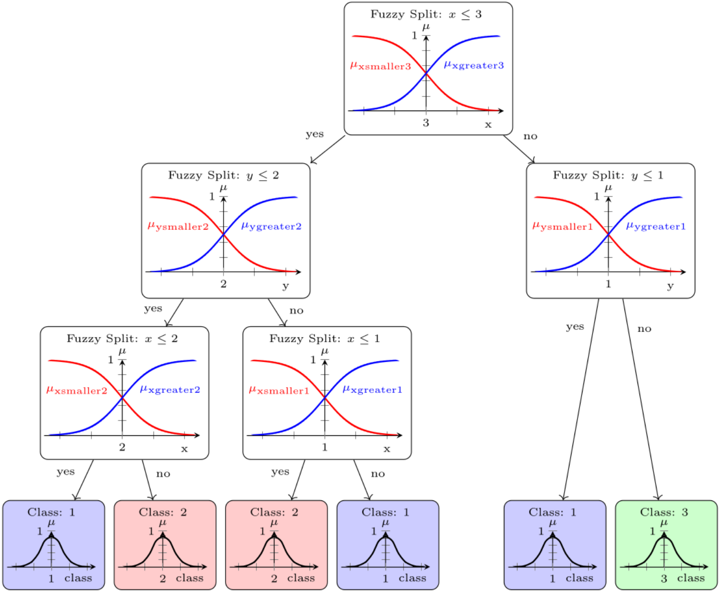
(a) Example decision tree (b) Decision surface over $\mathcal{D} = [0, 4] \times [0, 3]$

Decision Trees → Fuzzy Decision Trees

- Fuzzy Decision Trees
  - Each decision is a linguistic term
  - E.g. Traverse left if temperature is *cold*
- Conversion: Each (crisp) split is turned into two fuzzy sets
  - Fuzzy sets should maintain the semantics of the split
  - Provides robustness against noise
- Leaf nodes are represented with Linguistic terms






Fuzzy Split: $x \leq 3$
$\mu_{xsmaller3}$
$\mu_{xgreater3}$
yes
no
Fuzzy Split: $y \leq 2$
$\mu_{ysmaller2}$
$\mu_{ygreater2}$
Fuzzy Split: $y \leq 1$
$\mu_{ysmaller1}$
$\mu_{ygreater1}$
Fuzzy Split: $x \leq 2$
$\mu_{xsmaller2}$
$\mu_{xgreater2}$
Fuzzy Split: $x \leq 1$
$\mu_{xsmaller1}$
$\mu_{xgreater1}$
Class: 1
Class: 2
Class: 2
Class: 1
Class: 1
Class: 3

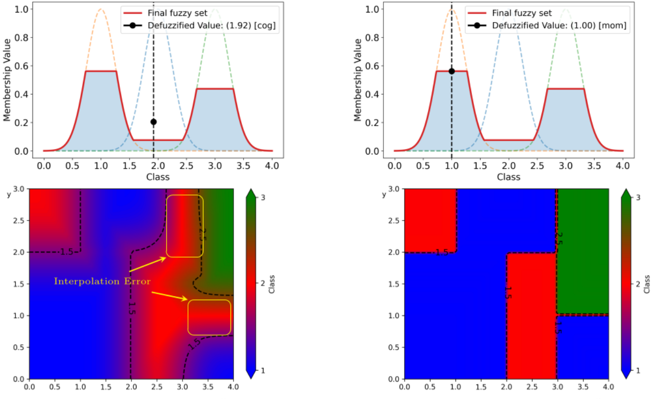
Fuzzy Decision Trees → Fuzzy Rules

- Depth-First traversal of the Fuzzy Decision Tree
  - Each node corresponds to a condition ofthe antecedent
  - Leaf node corresponds to the consequent
- One rule per path from root to leaf
- Corresponds to *unnesting* the decision tree

Rule	Antecedent	Consequent
1	x is smaller3 $\wedge$ y is smaller2 $\wedge$ x is smaller2	$class$ is 1
2	x is smaller3 $\wedge$ y is smaller2 $\wedge$ x is greater2	$class$ is 2
3	x is smaller3 $\wedge$ y is greater2 $\wedge$ x is smaller1	$class$ is 2
4	x is smaller3 $\wedge$ y is greater2 $\wedge$ x is greater1	$class$ is 1
5	x is greater3 $\wedge$ y is smaller1	$class$ is 1
6	x is greater3 $\wedge$ y is greater1	$class$ is 3

Fuzzy Decision Surfaces - CoG vs. MoM

- CoG: Interpolation effect + errors, smooth boundaries
- MoM: Hard boundaries, similar to Decision Trees



Fuzzy Rule Extraction for `md_flexible`

- Collect huge dataset of `md_flexible` simulations
- LiveInfoData: `maxDensity, homogeneity, threadCount, ...`
- TuningData: `Container, Traversal, Newton3, ..., Time`
- Introduce notion of *relative speed* for each configuration
  - $t_{best}^{(i)}$: Best configuration time in tuning phase i
  - $t_{config}^{(i)}$: Time of configuration in tuning phase i

$$\text{relative speed}_{config}^{(i)} = \frac{\mathrm{t}_{best}^{(i)}}{\mathrm{t}_{\mathrm{config}}^{(i)}}$$

  - Allows for fair comparison between different tuning phases

Resulting Dataset

- Performance of configurations in different environments
- Can be used to extract rules for both approaches
- Goal: Find configurations with high relative speed

ParticlesPerCell			Miscellaneous			Configuration			
avg	max	stddev	homo-genity	max-density	threads	Container DataLayout	Traversal	Newton3	Relative Speed
0.905	23	0.0129	0.0354	0.531	1	LinkedCells_AoS	lc_sliced	enabled	0.450641
2.201	13	0.0144	0.0861	0.627	24	VerletListsCells_AoS	vlc_sliced	disabled	0.594117
0.905	18	0.0136	0.0431	0.319	4	LinkedCells_AoS	lc_sliced_c02	enabled	0.454632
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Generate Rules: Approach 1

- Naively remove bad configurations (relative speed $< 70\%$)
- Group remaining dataset by tuning-phase (same live-info)
- Aggregate present (*good*) values for each parameter
- Apply Rule Extraction algorithm to each parameter

ParticlesPerCell			Miscellaneous			Aggregated Configuration Terms		
avg	max	stddev	homo-genity	max-density	threads	Container DataLayout	Traversal	Newton3
0.906	15	0.015	0.055	0.297	4	"LinkedCells_SoA, VerletClusterLists_SoA, VerletListsCells_AoS"	"lc_sliced, lc_sliced_balanced, lc_sliced_c02"	"enabled"
0.945	25	0.041	0.084	0.673	24	"LinkedCells_SoA, VerletClusterLists_SoA, VerletListsCells_AoS"	"lc_c04, lc_c08, lc_sliced, lc_sliced_balanced"	"disabled, enabled"

Antecedent				Consequent
avgParticlesPC	homogeneity	particlesPCStdDev	threadCount	Traversal
lower than 1.553	higher than 0.047	lower than 0.023	higher than 2.5	"lc_sliced, vlc_c18, lc_sliced_c02"
	lower than 0.037	lower than 0.023	lower than 26.0	"vcl_c06, vlc_c18, vlc_sliced_c02"
⋮	⋮	⋮	⋮	⋮

Generate Rules: Approach 2

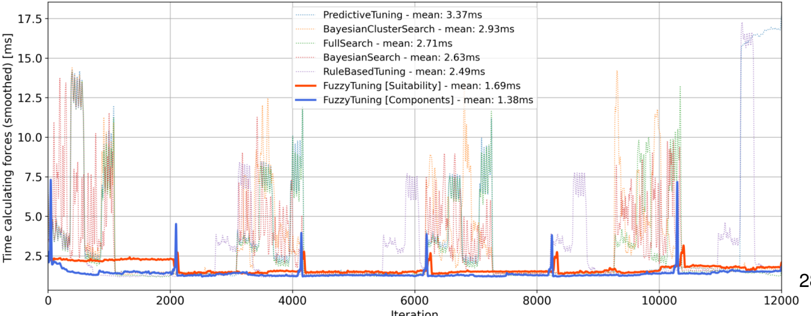
- Assign suitability-classes to each entry in the dataset
- Classes represent ranges of relative speed
- Apply Rule Extraction algorithm for the suitability class

ParticlesPerCell			Miscellaneous			Configuration				
avg	max	stddev	homo-genity	max-density	threads	Container DataLayout	Traversal	Newton3	Relative speed	Suitability
0.905	15	0.012	0.035	0.531	1	LinkedCells_AoS	lc sliced	enabled	0.450	"bad"
0.944	25	0.012	0.083	0.691	28	VerletClusterLists_AoS	vcl_c06	disabled	0.319	"poor"
0.944	20	0.012	0.079	0.041	12	LinkedCell_SoA	vlc_ sliced	enabled	0.989	"excellent"

Antecedent				Consequent
avgParticlesPC	homogeneity	particlesPCStdDev	threadCount	Suitability LinkedCells_AoS lc_c01_disabled
	lower than 0.084	higher than 0.029	higher than 26.0	"medium"
	higher than 0.084	higher than 0.029	higher than 26.0	"bad"
		higher than 0.02	lower than 2.5	"poor"
⋮	⋮	⋮	⋮	⋮

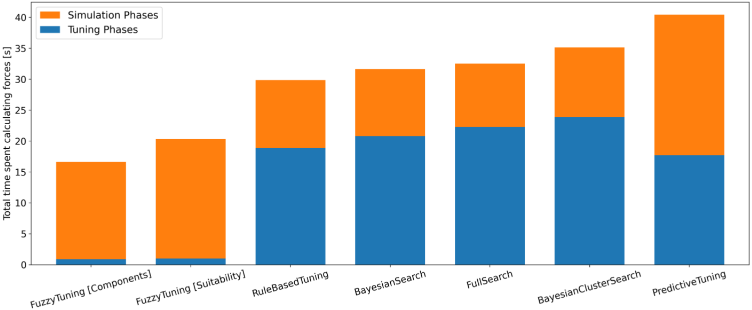
Benchmark 1: Exploding Liquid

- Exploding Liquid Benchmark (Included in Dataset)
  - Both fuzzy approaches look promising
  - Very short tuning phases (not much noise)
  - Selected configurations perform well (tiny spikes)
  - Winning configurations are (mostly) equivalent



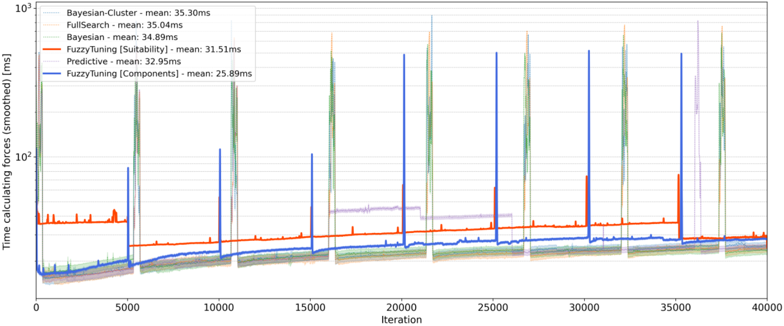
Total Time: Exploding Liquid

- Fuzzy tuning has lowest total time
- Mostly due to the very efficient tuning phases
  - Few configurations evaluated
  - Evaluated configurations are expected to perform well
- Benefits of tuning with tiny overhead



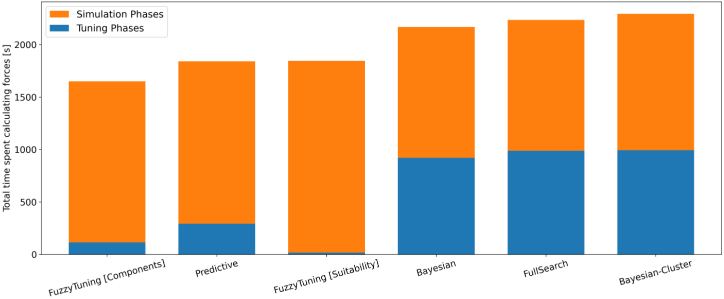
Benchmark 2: Spinodal Decomposition MPI

- Spinodal Decomposition MPI (Indirectly included in Dataset)
  - Fuzzy tuning promising again
  - Suitability approach struggles, often misses the best configuration
  - However, both approaches have efficient tuning phases



Comparison and Evaluation: Spinodal Decomposition

- Component approach performs well
- Suitability approach still quite fast
  - Fast tuning phases compensate for suboptimal configurations!
  - Misses caused by to optimisitic suitability threshold
    - Too few configurations are slected for evaluation
    - **Solution:** Increase the suitability threshold (Top 10% → Top 30%)



Future Work

- Dynamic Rule Generation
  - Update the expert knowledge on the fly
  - Adapt to new scenarios
  - No need for giant datasets
- Improving Tuning Strategies
  - Implement early stopping mechanism
  - Stop evaluating extremely bad configurations early
- Simplification of the Fuzzy System to Decision Trees
  - Use Decision Trees instead of Fuzzy Systems
  - Could potentially perform equally well, while being easier to understand

Conclusion

- Fuzzy Logic is very promising for tuning AutoPas
- Data-Driven Rule Extraction is a powerful tool
- However:
  - Requires a lot of prior data (current training data is 1.1GB)
  - Users cannot be expected to collect this data beforehand
  - A more user-friendly approach is needed for broader adoption
- Future work could focus on simplifying/streamlining the rule extraction process
- Alternatively: Investigate Early Stopping, to solve the tuning overhead once and for all

Thank you for your attention!

Questions?

References I

Crockett, K., Bandar, Z., Mclean, D., and OShea, J. (2006).
On constructing a fuzzy inference framework using crisp decision trees.
Fuzzy Sets and Systems, 157(21):2809–2832.

Mohammed, A., KorndÃ¶rfer, J. H. M., Eleliemy, A., and Ciorba, F. M. (2022).
Automated scheduling algorithm selection and chunk parameter calculation in openmp.
IEEE Transactions on Parallel and Distributed Systems, 33(12):4383–4394.

Newcome, S. J., Gratl, F. A., Neumann, P., and Bungartz, H.-J. (2023).
Towards the smarter tuning of molecular dynamics simulations.
Amsterdam, The Netherlands.