Algorithm Selection in Short-Range Molecular Dynamics Simulations

Samuel James Newcome[1 [0000−0003−3044−6057]], Fabio Alexander Gratl[1 [0000−0001−5195−7919]], Manuel Lerchner[1 [0000−0002−5167−465X]], Abdulkadir Pazar[1 [0009−0005−9035−6821]], Manish Kumar Mishra[1 [0000−0002−5919−3260]], and Hans-Joachim Bungartz[1 [0000−0002−0171−0712]]

Chair of Scientific Computing in Computer Science, Department of Computer Science, Technical University of Munich, Boltzmannstr. 3, 85748, Garching bei Muenchen, Germany samuel.newcome@tum.de

Abstract. Recent works have highlighted the advantages of algorithm selection to optimise scientific simulations, presenting a range of approaches from classical time-series prediction, to expert-guided, to data-driven. In this work, we present novel variations upon these approaches for Molecular Dynamics simulations, implemented in the algorithm selection particle simulation library AutoPas, and compare them in terms of both performance and practicality. We demonstrate that these approaches can achieve speedups of up to 1.25 compared to an optimal single algorithm without dynamic algorithm selection.

Keywords: Algorithm Selection · Molecular Dynamics · AutoPas

1 Introduction

Molecular Dynamics (MD) simulations are an important technology with applications including thermodynamics [8] and bio-membrane simulation [5]. Molecules are simulated as point bodies moving according to Newton's equations of motion, propagated with some numerical integrator. This results in a large number of timesteps, each typically computationally dominated by inter-particle force calculations, and between them the distribution of particles changes only minimally. Where the forces are short-ranged, a common optimisation is to introduce a cutoff distance beyond which the forces are neglected. Simulations with this optimisation are well studied, resulting in the development of numerous algorithms and parallelisations, however no algorithmic configuration (AC) is optimal in all scenarios [2] and the relative performance of these ACs can vary on different hardware or with different force models [7].

Such findings motivate the development of the algorithm selection and tuning library AutoPas[1], which aims to select the optimal from a large range of ACs [2]. However, prior works [2] [10] [7] have utilised only naive selection strategies

[1] https://github.com/AutoPas/AutoPas

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35

which involve significant time spent trialling suboptimal ACs, which can perform orders of magnitude worse than the optimal.

In this work, we present and compare three methods that aim to avoid this. The **Predictive** strategy uses past time series data of an AC's performance to predict how it will perform in the future. The **Expert** strategy uses expert-written rules to determine in which scenarios an AC should be trialled. The **Random Forest** strategy uses a random forest trained on pre-collected data, which maps scenario-dependent metrics to optimal ACs. We discuss the practicalities of these methods as well as their performance in three varied simulation scenarios.

2 AutoPas

AutoPas is designed to provide a black-box particle container to the developer of an arbitrary cutoff-based particle simulator. It contains a variety of algorithms and parallelisations which combine to form an AC for efficiently calculating the forces on each particle. AutoPas aims to dynamically select the optimal of these ACs over the course of the simulation, as well as tune them. As the focus of this work is on the selection strategies, we will not discuss the different ACs themselves. For such a discussion, the reader is referred to Gratl et al. [2] and Gratl-Gaßner [3]. Overall, a total of 116 possible ACs will be used in this work.

As AutoPas primarily handles shared memory parallelism, distributed memory parallelism is intended to be implemented by the simulator developer. Seckler et al. [10] proposes decomposing the domain into a number of subdomains and assigning each an MPI rank with its own AutoPas container. The consequence of this is that different subdomains can use different ACs [10].

AutoPas' algorithm selection consists of a series of *tuning phases* that occur in regular intervals. During each tuning phase, all ACs are trialled for a small number of iterations and the average is taken. The AC with the smallest average is then used until the following tuning phase. Naively trialling all ACs is referred to as a Full Search. Selection Strategies, such as those presented in Section 5, are intended to reduce the number of ACs trialled [3].

3 Related Work

Armstrong et al. [1] investigated a very similar dynamic algorithm selection problem also applied to short-range particle simulations, using a temporal difference reinforcement learning agent and a linear regression model to predict the optimal algorithm out of two.

Mohammed et al. [6] investigated the dynamic selection of OpenMP loop-scheduling algorithms out of a range of twelve algorithms. They considered random search, exhaustive search, and expert search methods. The expert search method takes relevant measurements and applies them to expert-written Fuzzy Logic systems, which can result in changes to the algorithm.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35

Stylianou and Weiland [11] investigated the selection of one of six different sparse matrix storage formats using a decision tree and random forest. Unlike the previously discussed works, they considered independent matrices rather than a problem that evolves over time and, as such, their selection methods are solely based on features from the matrices.

The first work is similar to the Predictive strategy, and the latter two serve as inspiration for an Expert and a Random Forest strategy; however, the ideas must be non-trivially redeveloped into a form that is applicable to AutoPas. Unlike all of the above works, we will consider a much larger number of algorithms.

4 Live Simulation Statistics

In order to implement expert knowledge or data-driven approaches, as discussed previously, statistics need to be extracted from the simulation, from which decisions on the optimality of ACs can be made. We consider a simple yet computationally efficient scheme in which particles are binned into small cells, and the mean, standard deviation, median, and maximum numbers of particles per cell are collected. Furthermore, we also gather statistics on the number of bins, how many of these are empty, the number of OpenMP threads, and the skin[2].

5 AutoPas' Selection Strategies

In this section, we will describe three novel dynamic algorithm selection methods for AutoPas: the **Predictive**, **Expert**, and **Random Forest** methods. For a deeper discussion of the implementation of the methods, see Gratl-Gaßner [3], Lerchner [4], and Pazar [9] respectively. For details on the exact live simulation statistics used, model parameters, rules, and training data, see the repository mentioned in the Appendix.

5.1 The Predictive Strategy

In the predictive strategy, each AC is trialled during the first two tuning phases. After this, at the start of each following tuning phase, a linear model for each AC fitted to the last two data points for that AC, is used to predict the performance of that AC in that tuning phase. Only the ACs expected to perform within a relative threshold of the expected optimum are trialled.

To avoid ACs that previously performed poorly, never being trialled again, even if the simulation has changed significantly, we retrial ACs after a number of tuning phases where they have not been trialled, even if they are expected to perform worse than the threshold. On the other hand, a blacklist is applied to extremely poor-performing ACs, and they are never retrialled.

The predictive tuning strategy's simplicity is a clear benefit compared to later methods. No data is required outside of the data generated within the simulation itself, and, to a degree, no expert knowledge is required to use it.

[2] See Gratl et al. [2]

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35

5.2 The Expert Strategy

A fuzzy-logic-based system was designed to map the simulation statistics using expert-written fuzzy-logic rules to choose an optimal AC. Fuzzy logic is a good choice for such a system, as experts cannot feasibly create exact rules for this.

The clear downside of this method is the requirement for an expert to spend significant effort to develop such rules. Furthermore, Newcome et al. [7] showed that one set of rules might be suboptimal on different hardware or with a different force model. In addition, as AutoPas itself is developed, with new ACs and improvements upon existing ACs, such rules would have to be adapted.

As the above points suggest the infeasibility of a general set of expert rules that could be universally applied, we instead design rules specifically targeted towards the experiments in Section 6. The exact design of the rules was determined using data gathered by running the very experiments intended to be optimised. This could be useful, for example, when conducting several experiments with a range of parameters: one experiment is run with slow full searches to collect data to build the rules, and the rest can then use the Expert strategy. We should emphasise that these rules will not perform optimally for all experiments that could be run with AutoPas.

5.3 The Random Forest Strategy

The Random Forest strategy trains a random forest that maps the live simulation statistics to a suggested AC. To train the model, a dataset needs to be collected prior to the simulation being run.

Such a data-driven method is far easier to use than the Expert strategy, requiring only a set of simulation scenarios from which to generate performance data. This avoids much of the difficulties mentioned previously for expert-knowledge methods as, if the set of scenarios is already provided, trialling the ACs and training a model requires only a minor amount of effort.

The key issue comes from generating the set of simulation scenarios: the scenarios should be representative of the experiments intended to be optimised with the strategy, but generating the scenarios should not be so computationally expensive that it negates the benefits of the strategy nor, for the same reason, should expensive, irrelevant scenarios be included. As such, some expert decisions regarding the dataset still need to be made.

To avoid the first of these issues, we generate scenarios using simple random and fixed-grid distributions of particles, which are likely physically unrealistic (molecules too close together) but are representative at the algorithmic level. To address the second issue, these "fake" scenarios were designed to mimic a variety of different real scenarios, ultimately with the experiments in Section 6 in mind.

6 Experimental Setup

We chose three scenarios to demonstrate our strategies. Between these scenarios, there is minimal overlap in optimal AC. For brevity, we only discuss the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35

behaviour of the simulations at a high level. The full details of the software versions used and the experiment parameters can be found in the repository mentioned in the Appendix. We tested our selection strategies on HSUper[3].

Heating Sphere In the heating sphere experiment, a small sphere of molecules is placed in the centre of a large domain with reflective boundaries. Only one MPI rank is used. The simulation begins at a low temperature, which keeps the molecules together. The temperature is raised slowly, causing the sphere to start losing molecules until the domain fills sparsely. Due to the molecules dispersing, a change in the optimal AC can be seen.

Exploding Liquid In the exploding liquid experiment used in prior AutoPas works [3], a thick layer of molecules is placed in the centre of an otherwise only sparsely filled tube-like domain with periodic boundaries. The thick layer explodes outwards, along the tube, in two dense "waves" of molecules in either direction. The domain is split into six subdomains along the length of the tube, each with its own MPI rank and AutoPas container. As such, there are different optimal ACs in each subdomain depending on whether it contains the dense wave, the remnants left after the wave, or if the wave has not reached it yet.

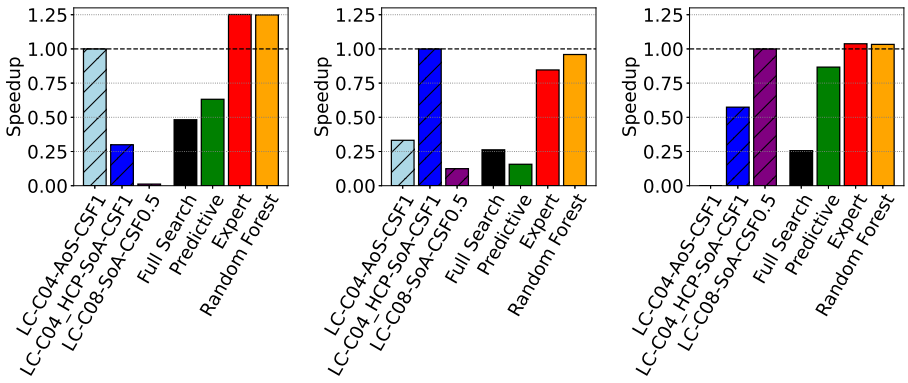
Rayleigh-Taylor In the Rayleigh-Taylor experiment, a layer of larger, lighter molecules is placed under a denser layer of smaller, heavier molecules, with the molecule layers mixing as the larger molecules rise to the top and the smaller molecules sink to the bottom. The simulation was run on 40 MPI ranks with the distribution of the two molecule types, leading to different optimal ACs on each MPI rank.

7 Results & Analysis

We first determined which single AC was optimal for each experiment (i.e. without any algorithm selection) and found that each experiment had a different optimal single AC. These make good targets to reach with our algorithm selection methods. If we achieve a speedup relative to these, we see an advantage of *dynamically* selecting and changing ACs in different regions and at different points in time. However, in the case of the Predictive and Random Forest strategies, getting close to these targets is still a success as a user without prior knowledge of which single AC is optimal will still benefit. But this second criterion of success is not valid for the Expert strategy as the methodology in Section 5.2 suggests that this single optimal AC would be known, and so a success for this strategy requires a speedup relative to this AC.

[3] Compute nodes featuring 256GB of RAM, 2 Intel Icelake sockets each with an Intel(R) Xeon (R) Platinum 8360Y processor with 36 cores; https://portal.hpc.hsu-hh.de/documentation/hsuper/

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35



(a) Heating Sphere. (b) Exploding Liquid. (c) Rayleigh Taylor.

Fig. 1: Comparison of the accumulated time spent calculating forces for each thread across the entire simulation, for the three algorithmic configurations that are the optimal single configurations for each experiment ((a) `LC-C04-N3L-AoS-CSF1`, (b) `LC-C04_HCP-N3L-SoA-CSF1`, (c) `LC-C08-N3L-SoA-CSF0.5`) and each selection strategy. Results are shown as speedups relative to the optimal single configuration for that experiment. Note that the `LC-C04-N3L-AoS-CSF1` with the Rayleigh-Taylor experiment timed out upon reaching the maximum wall time allowed on HSUper – about 7 times the wall time that `LC-C08-N3L-SoA-CSF0.5` took and only achieving approximately three-quarters of the total iterations.

In Figure 1, we see the speedup of these three optimal ACs, the naive Full Search strategy, and the three strategies presented in this work relative to each experiment's optimal single AC.

We see that the one experiment's optimal AC could perform significantly worse in other experiments. While, to an expert, such optimal ACs make sense in hindsight, suggesting such an AC with no experience running similar simulations is challenging. Furthermore, in some time periods and subdomains, other ACs perform significantly better than these (and other works have found other optimal ACs [7].) Therefore, we see benefit from dynamic algorithm selection from a wide range of ACs.

Considering now the selection strategies, we see that the Predictive strategy achieves mixed success, achieving near-1.0 speedup only in the Rayleigh-Taylor experiment. That it achieves a worse performance than Full Search in the Exploding Liquid experiment can be explained by the dramatic changes in the computational profile invalidating the linear models.

The Expert and Random Forest strategies perform similarly, with Random Forest performing slightly better in the Exploding Liquid scenario. In the Heating Sphere experiment, both strategies achieve speedups of 1.25, correctly following the switch in the optimal AC. This represents the significant potential of the methods and dynamic algorithm selection more generally. In the Exploding

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35

Liquid experiment, however, the methods achieve "speedups" of 0.85 and 0.89, respectively and in the Rayleigh-Taylor experiment, they achieve 1.04 and 1.03, respectively.

In these experiments, the optimal single AC is the AC, which is optimal on the MPI ranks with the most work (dense regions of particles), even if these denser regions move and thus, the ranks with this high workload change. As a dense region moves out of a rank, the optimal AC changes, but as the workload of the rank also reduces, selecting a new AC has a minimal impact overall. We nevertheless should remember that, for the Random Forest approach, a near-1.0 speedup is still successful, and we expect that improved MPI load balancing would lead to a greater potential for above-1.0 speedup in such experiments and the current results suggest the Expert and Random Forest strategies could realise this potential.

The overhead of evaluating all the models and rules was generally found to be negligible.

8 Conclusion and Outlook

Of the three methods presented in this work, the Random Forest strategy appears the most successful – balancing performance with accessibility – and is able to achieve speedups of 1.25 compared to simply picking a single AC. If we instead compare against the prior naive Full Search strategy, we see a speedup of 4.05, demonstrating a significant improvement. Whilst it is expected that the Expert strategy could be improved to match or even beat the Random Forest strategy, this is not a tractable solution to the problem and is unsustainable. The Predictive strategy struggles significantly, particularly in dramatically changing scenarios; however, the further development of such an accessible, data-free approach could be valuable in some situations.

As previously mentioned, a key issue with such data-driven models is the need for good data, the generation of which is itself an expert decision. A more accessible solution could be to combine online learning with an understanding of confidence: in scenarios where the model has not been trained on similar data, the model becomes unconfident in its suggestions, triggering exploration of the search space and thus online learning, and therefore better performance in further similar simulation runs. Thus, the user generates data relevant to the simulations they want to optimise.

Acknowledgements This work has been financially supported by, as well as computational resources (HSUper) from, the project hpc.bw, which is funded by dtec.bw – Digitalisation and Technology Research Centre of the Bundeswehr. dtec.bw is funded by the European Union — NextGenerationEU.

The authors would like to thank Amartya Das Sharma and Ruben Horn, of Helmut-Schmidt-University, for their invaluable assistance and advice with the Python-C interface on HSUper. The authors would like to express their deep thanks to all contributors to the open-source AutoPas project.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35

Appendix

All input files, outputs, job scripts for training data and the experiments, models, and fuzzy rule files, as well as the software versions used can be found at `https://github.com/SamNewcome/Algorithm-Selection-in-Short-Range-Molecular-Dynamics-Simulations`.

References

[1] Warren Armstrong et al. "Dynamic algorithm selection using reinforcement learning". In: *2006 international workshop on integrating ai and data mining*. IEEE. 2006, pp. 18–25.

[2] Fabio Alexander Gratl et al. "N ways to simulate short-range particle systems: Automated algorithm selection with the node-level library AutoPas". In: *Computer Physics Communications* 273 (2022), p. 108262.

[3] Fabio Alexander Gratl-Gaßner. "AutoPas: Automated Dynamic Algorithm Selection for HPC Particle Simulations". Doctoral thesis. Technical University of Munich, 2025.

[4] Manuel Lerchner. "Exploring Fuzzy Tuning Technique for Molecular Dynamics Simulations in AutoPas". Bachelor's thesis. Technical University of Munich, 2024.

[5] Siewert J Marrink et al. "Computational modeling of realistic cell membranes". In: *Chemical reviews* 119.9 (2019), pp. 6184–6226.

[6] Ali Mohammed et al. "Automated scheduling algorithm selection and chunk parameter calculation in openmp". In: *IEEE Transactions on Parallel and Distributed Systems* 33.12 (2022), pp. 4383–4394.

[7] Samuel James Newcome et al. "Towards auto-tuning Multi-Site Molecular Dynamics simulations with AutoPas". In: *Journal of Computational and Applied Mathematics* 433 (2023), p. 115278.

[8] Isabel Nitzke et al. "ms2: A molecular simulation tool for thermodynamic properties, release 5.0". In: *Computer Physics Communications* (2025), p. 109541.

[9] Abdulkadir Pazar. "Optimizing Algorithm Selection in AutoPas with Decision Trees and Random Forests". Master's thesis. Technical University of Munich, 2024.

[10] Steffen Seckler et al. "AutoPas in ls1 mardyn: Massively parallel particle simulations with node-level auto-tuning". In: *Journal of Computational Science* 50 (2021), p. 101296.

[11] Christodoulos Stylianou and Michele Weiland. "Optimizing Sparse Linear Algebra Through Automatic Format Selection and Machine Learning". In: *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE. 2023, pp. 734–743.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97635-3_35