Dynamic particle swarm optimization of biomolecular simulation parameters with flexible objective functions

Marie Weiel, Markus Götz, André Klein, Daniel Coquelin, Ralf Floca and Alexander Schug

Molecular simulations are a powerful tool to complement and interpret ambiguous experimental data on biomolecules to obtain structural models. Such data-assisted simulations often rely on parameters, the choice of which is highly non-trivial and crucial to performance. The key challenge is weighting experimental information with respect to the underlying physical model. We introduce FLAPS, a self-adapting variant of dynamic particle swarm optimization, to overcome this parameter selection problem. FLAPS is suited for the optimization of composite objective functions that depend on both the optimization parameters and additional, a priori unknown weighting parameters, which substantially influence the search-space topology. These weighting parameters are learned at runtime, yielding a dynamically evolving and iteratively refined search-space topology. As a practical example, we show how FLAPS can be used to find functional parameters for small-angle X-ray scattering-guided protein simulations.

Proteins are the molecular workhouses of biological cells, with a myriad of tasks including oxygen transport, cellular communication and energy balance. As a protein’s function is linked to its structure and dynamics, its understanding requires resolving the protein’s three-dimensional shape. Misfolded proteins are associated with several neurodegenerative diseases, and deciphering the protein’s three-dimensional shape is crucial for simulation performance, and determining the bias potential’s weight is the key challenge. This selection determines how experimental and theoretical information is balanced. The bias weight is an empirical MD parameter expressing the confidence in the experimental data versus the physics-based force field. In Bayesian methods, the right weighting is derived from a statistical treatment. However, such sophisticated approaches are practically inapplicable for users with a primarily experimental background. It is still common practice to manually determine an ‘optimal’ bias weight via grid search, that is, an exhaustive search through a fixed subset of the parameter space. Adopting concepts from computational intelligence, we introduce FLAPS (‘flexible self-adapting particle swarm’ optimization), a self-learning metaheuristic based on particle swarms, to resolve this parameter selection problem. Our contributions include the following:

- A new type of flexible objective function (OF) to assess a data-assisted simulation’s plausibility in terms of simulated structures and thus the suitability of the MD parameters used.

In summary, FLAPS is a promising approach to optimizing the parameters of molecular simulations, allowing for a more accurate and reliable interpretation of experimental data.

**Acknowledgments**

The authors would like to thank the National Center for Radiation Research in Oncology (NCRO), Heidelberg, Germany, the Institute for Advanced Simulation (IAS), Jülich Supercomputing Centre (JSC), Jülich, Germany, and the Steinbuch Centre for Computing (SCC), Karlsruhe Institute of Technology (KIT), Eggenstein-Leopoldshafen, Germany, for their support in this research. The authors also acknowledge the use of the High Performance Computing Platform (HPCP) at the Steinbuch Centre for Computing (SCC), Karlsruhe Institute of Technology (KIT), and the High Performance Computing Platform (HPCP) at the Jülich Supercomputing Centre (JSC), Jülich, Germany. The research was supported by the German Federal Ministry of Education and Research (BMBF) through the project “Hardware und Software für Exaschaltungen” (HSE) and the project “Exascale Computing for Nanobiotechnology” (ExCub).
A self-adapting particle swarm optimizer for dynamically evolving environments resulting from multiple quality features of different scales in the flexible OF.

A flexible self-adapting objective function

Typically, the set of responses is mapped to a scalar score by calculating the scalar product with fixed weights. These OF parameters supposedly reflect relative importance, implicitly encoding arbitrary prior beliefs. Instead, we set up a ‘maximum-entropy’ OF with the fewest possible assumptions:

\[
f(x; z = \{\mu, \sigma\}) \triangleq \sum_{j} \frac{R_j(x) - \mu_j}{\sigma_j}
\]

where \(z\) is the set of OF parameters, \(\mu_j\) is the mean and \(\sigma_j\) is the standard deviation of response \(R_j\), for a particle at position \(x\). All responses are considered equally important but can have different ranges and units. To make them comparable on a shared scale, we standardize each response’s set of values gathered over previous generations. This strategy imitates the concept of rolling batch normalization\(^{18}\). Each layer’s inputs are centred and rescaled with the aim to improve a neural network’s speed, performance and stability. Initially proposed to mitigate internal covariate shift, batch normalization is believed to introduce a regularizing and smoothing effect and promote robustness with respect to different initialization schemes. The OF in equation (1) depends not only on the parameters of interest, \(x\), in our case the MD parameters, but also on a priori unknown, context-providing OF parameters, \(z = \{\mu, \sigma\}\), from the standardization. Their values cannot be deduced from individual OF evaluations, yet fundamentally control OF performance and hence the optimization process.

Algorithm 1 FLAPS algorithm. Initialize population \(pop\) with swarm size \(S\) particles at random positions \(x_p\) \((p = 1, \ldots, S)\) between upper and lower bounds of the search space \(b_{up}\) and \(b_{lo}\), respectively.

for \(g \leftarrow 1\) to maximum generations \(G\) do

for particle in pop do

Evaluate responses at particle.position = \(x_p\):

\(\text{particle.fargs} = [\text{response}(x_p)]\)

end

Append current generation \(pop\) to history \(histp\): \(histp.append(pop)\)

Update OF parameters \(z\) based on current knowledge state of responses in \(histp\): \(z = \text{updateParams}(histp)\)

for particle in histp do

(Re-)evaluate objective function using most recent \(z\):

\(\text{particle.fitness} = f(x; z)\)

end

generate new \(histp\) do

Determine personal best \(p^g_{\text{best}}\) and update global best \(g_{\text{best}}\) accordingly.

end

for particle in pop do

Update velocity and position:

\(\text{particle.speed} + \text{rand} (0, \phi_1) (p^g_{\text{best}} - \text{particle.position}) + \text{rand} (0, \phi_2) (g_{\text{best}} - \text{particle.position})\)

Regulate velocity via \(s_{\text{max}} = 0.7 G^{-1} (b_{up} - b_{lo})\):

if \(\text{particle.speed} > s_{\text{max}}\) then

\(\text{particle.speed} = s_{\text{max}}\)

end

if \(\text{particle.speed} < -s_{\text{max}}\) then

\(\text{particle.speed} = -s_{\text{max}}\)

end

\(\text{particle.position} += \text{particle.speed}\)

end

Result: \(g_{\text{best}}\)
Our self-adapting PSO variant, FLAPS, solves this problem. Provided with a comprehensive history of all previous particles and their responses, OF parameters are learned on the fly. They are continuously refined according to the current state of the optimization, yielding a dynamically evolving and increasingly distinct OF topology. This environmental dynamism may cause convergence problems if the OF fails to approach a stable topology. As more particles are evaluated, the ranges and distributions of individual responses become better understood. Therefore, OF parameters become more accurate, improving OF performance in assessing the suitability of the actual parameters of interest, \( x \). After each generation, the values \( z = (\mu, \sigma) \) are used to reevaluate the OF for all particles in the history. Personal best positions, \( p^* \), and the swarm’s global best position, \( g^* \), are updated accordingly for propagating particles to the next generation. FLAPS uses a traditional PSO velocity formulation\(^{14}\). Strategies to prevent diverging velocities include introducing an inertia weight\(^{14}\) or a constriction factor\(^{15}\). We regulate the velocities by means of a maximum value at each particle update\(^{11}\). FLAPS’s pseudo code is shown in Algorithm 1. Inspired by the ‘simplifying PSO’ paradigm, it builds on a slim standard PSO core and can easily be complemented by concepts such as inertia weight\(^{14}\) and swarm constriction\(^{15}\) or diversity increasing mechanisms. Its time complexity is similar to that of a standard PSO with \( O\left(\frac{1}{\Delta} \cdot G \cdot Sim + S \cdot Opt\right) = O\left(\frac{1}{\Delta} \cdot G \cdot Sim + S^2 \cdot G\right) \) in Landau notation, where \( P \) is the number of simulation processors, ‘Sim’ the maximal simulation time, and all other variables as defined in Algorithm 1.

**Application to data-assisted protein simulations**

We applied FLAPS to the optimization of MD parameters in SAXS-guided protein simulations.\(^{15}\) SAXS data are integrated into computationally efficient structure-based models, which probe dynamics arising from a protein’s native geometry\(^{36,37}\). To assess the utility of different MD parameter sets, we need a metric for simulation quality in terms of physically reasonable structures matching the data. Designing such an OF in advance is non-trivial and has two major aspects: (1) physical plausibility of a simulated ensemble of protein structures and (2) its agreement with the target data, that is, how well the data are reproduced by simulated structures. To represent these aspects, we use the Rosetta energy function 2015 (REF15)\(^{36,37}\) and the least-squares deviation \( \chi^2 \) of simulated data from the target data\(^{16}\).

Protein structure determination relies on quick and reliable scoring of many models to select those closest to the native state. Structures are rated by energetic scores associated with their conformational state. REF15 is a weighted sum of energy terms efficiently approximating the energy of a biomolecular conformation as a function of geometric degrees of freedom and chemical identities\(^{15}\). With a protein’s native fold corresponding to the state with minimal free energy, a lower-scoring structure is expected to be more native-like. Because the scores do not have a direct conversion to physical energies, REF15 and structural stability are not correlated across different proteins. Similarly, \( \chi^2 \) values without context are inconclusive and must be compared for each protein system. Both REF15 and \( \chi^2 \) are available from a simulation, yielding a molecular system’s atom positions over time.

For SAXS-guided structure-based simulations, two MD parameters are particularly important: bias weight \( k_b \) and temperature \( T \). \( k_b \) balances information in the SAXS data with the physical model, and \( T \) is a measure of available thermal energy and controls the system’s conformational flexibility. Thus, a particle corresponds to a simulation using a particular MD parameter set, \( x = (k_b, T) \). The OF is set up as

\[
\begin{align*}
  f(x = (k_b, T); z) &= [\text{REF15}_{\text{av}}]_{\text{std}} \\
                  &+ [\chi^2_{\text{med}}]_{\text{std}} \\
                  &+ [\chi^2_{\text{av}}]_{\text{std}} \\
\end{align*}
\]

The first response evaluates the average physical stability of simulated structures, the second is the median \( \chi^2 \) deviation of simulated data from the target data. Owing to the ill-posed nature of the SAXS inverse problem, globally distinct protein structures can possess the same scattering intensity. As shown in Fig. 1, this can lead to a pronounced ambiguity in \( \chi^2 \). To resolve the resulting non-injectivity in the OF, we introduce a third response, the inverse average \( \chi^2 \) deviation. This acts as a regularizer, rewarding deviations from the target data and thus preventing possible overfitting. Combining these responses yields a surrogate model of a simulated ensemble’s similarity to the desired target structure. The smaller the OF, the more physical, data-consistent and (likely) similar to the target state the simulated structures are.

In physico-empirical structure-based models, different combinations of bias weight and temperature can equally yield useful results. There is no MD parameter ground truth for this type of simulation, so a purely evidence-based evaluation according to the similarity of simulated protein structures to the target is to be applied.

We use the global distance test (GDT)\(^{40}\) to quantify differences between two conformations of a protein (Methods section ‘Root-mean-square deviation’). To estimate how similar two superimposed structures are, the placement of each alpha carbon is compared to various distance cutoffs. Percentages, \( P_i \), of alpha carbons with displacements below cutoffs of \( x \) Å are used to calculate the total score:

\[
\text{GDT} = 0.25 \times (P_1 + P_2 + P_3 + P_4) \in [0, 100].
\]

Higher GDT values indicate a stronger similarity between two models. Structures with GDT > 50 are considered topologically accurate\(^{11}\). The GDT is used to validate the OF as a surrogate model of an ensemble’s similarity to the target structure.

**Results**

We optimized MD parameters of SAXS-guided structure-based simulations for two well-characterized proteins: lysine-, arginine- , ornithine-binding (LAO) protein and adenylate kinase (ADK).
Small ligands such as sugars and amino acids are actively transported into bacteria across cell membranes. Dedicated transport systems comprise a receptor (that is, the binding protein) and a membrane-bound protein complex. Interactions of the ligated binding protein with the membrane components induce conformational changes in the latter, forming an entry pathway for the ligand. We study LAO protein (Fig. 2a), which undergoes a conformational change from an apo (unligated; Protein Data Bank [PDB] code 2LAO) to a holo (ligated; PDB code 1LST) state upon ligand binding. The structures have a GDT of 39.39. SAXS-guided simulations started from the open conformation and aimed at the closed one. Adenosine triphosphate (ATP) is the universal energy source in cells and is vital to processes such as muscle contraction and nerve impulse propagation. By continuously checking ATP levels, ADK provides the cell with a mechanism to monitor energetic levels and metabolic processes. The transition between an open (PDB code 4AKE) and closed (PDB code 1AKE) state is quintessential to its catalytic function. The structures have a GDT of 33.06. SAXS-guided simulations started from the open conformation and aimed at the closed one.

Artificial target data were calculated from known structures with CRYSOL. Statistical uncertainties were modelled following ref. 48. We performed seven FLAPS runs with different initial conditions for each protein. Swarm-based metaheuristics such as FLAPS have hyperparameters influencing the optimization behaviour, and their efficacy can only be demonstrated empirically by a finite number of computational experiments. We present results for a swarm of 10 particles and 15 generations as a workable trade-off between optimization performance and compute time for the considered application. This set-up was found to be sufficient for convergence in preceding trial runs. Calculations were performed on 1,000 cores of a supercomputer. One run cost ~40,000 core hours. The results of the three best runs are listed in Table 1 (complete results are provided in Supplementary Tables 1 and 2).

As shown for LAO protein in Fig. 3, the OF consistently converged to a stable topology (Supplementary section ‘Analyzing swarm convergence’).

For each simulation, we calculated the median GDT with respect to the target state from all structures in the trajectory. To validate the OF, we state its Pearson correlation $\rho$ with the median GDT as a measure of linear correlation. Because minimizing the OF should be equivalent to maximizing the GDT, negative correlations, ideally $-1$, are expected. The OF’s suitability is confirmed for both LAO protein and ADK with correlations up to $-0.94$ and $-0.85$, respectively.

Discussion

The inverse problem of reconstructing molecular structures from low-resolution SAXS data is still unsolved. Biomolecular simulations are among the most powerful tools for eliminating the arising ambiguity and access the valuable structural information content of such data. However, data-assisted simulations rely on MD parameters, where, most importantly, experimental information must be weighted accurately with respect to the physical model.

Here, we have shown how computational intelligence can be used to systematically explore MD parameter spaces and optimize the performance of complex physics-based simulation techniques. We introduced FLAPS, a data-driven solution for a fully automatic and reproducible parameter search based on particle swarms.

To identify the best MD parameters for SAXS-guided protein simulations, we designed an OF as an accurate surrogate of simulation quality in terms of physical structures matching the target data. A suitable OF will typically depend on multiple quality features of different scales to equally reflect a data-assisted simulation’s physical plausibility and its agreement with the data. To handle multiple responses in classical PSO, they need to be mapped to a scalar score via multiplication by fixed weights. These additional OF parameters must either be chosen manually (and probably suboptimally) in advance or be absorbed into the search space, resulting in a massive increase in dimensionality. FLAPS solves this problem by intelligently learning OF parameters in the optimization.
process, avoiding the need to set them as ‘magic numbers’, while reducing the search-space dimensionality to a minimum. Various responses are automatically balanced with respect to each other to enable a meaningful and unbiased comparison on a shared scale. Implemented in FLAPS, our conceptual OF reliably identified useful MD parameters for two different proteins, where we observed

For each protein system, the best three runs are listed. OF, objective function $f$; $\rho$, Pearson correlation of $f$ and median global distance test; $k$, bias weight; $T$, temperature; GDT$^{med}$, median global distance test (total score).

### Table 1 | FLAPS optimization results

| System | LAO protein (holo to apo) | | Adenylate kinase (open to closed) | |
|--------|----------------------------|----------------------------|
| Seed   | 1790954                    | 1791104                    | 1791106 | 1795691 | 1798723 | 1810891 |
| $\rho$ | $-0.94$                    | $-0.87$                    | $-0.87$ | $-0.85$ | $-0.81$ | $-0.74$ |
| $f^{min}$ | $-2.34$ | $1.79$ | $1.99$ | $1.42$ | $1.57$ | $1.62$ |
| $f^{max}$ | 8.32   | 5.86   | 4.41   | 6.92   | 9.05   | 7.47   |

Best simulation in terms of OF

| $f$ | $k$ | $T$ | $\text{GDTmed}$ | $\text{OF}$ | $\text{GDTmed}$ |
|-----|-----|-----|-----------------|-------------|---------------|
| $k$ | $-2.03$ | $-1.73$ | $-1.55$ | $1.969 \times 10^{-9}$ | $1.970 \times 10^{-9}$ |
| $T$ | 11.98 | 29.63 | 10.03 | 10.84 | 10.56 |

Best simulation in terms of GDT$^{med}$

| $f$ | $k$ | $T$ | $\text{GDTmed}$ | $\text{OF}$ | $\text{GDTmed}$ |
|-----|-----|-----|-----------------|-------------|---------------|
| $k$ | $3.001 \times 10^{-10}$ | $3.422 \times 10^{-11}$ | $4.190 \times 10^{-10}$ | $2.030 \times 10^{-9}$ | $1.970 \times 10^{-9}$ |
| $T$ | 7.095 | 63.78 | 63.78 | 63.78 | 63.78 |

Fig. 3 | Dynamically evolving OF topology. Results are shown for LAO protein (seed 1790954). The current global best position is marked by a star. $\varepsilon$ is the energy scale of the structure-based model.

Fig. 4 | OF versus median GDT. a,b. OF versus median GDT for LAO protein (a) and ADK (b). GDT$^{sys}$, GDT between initial and target structure of each test system; $\varepsilon$, energy scale of the structure-based model.
Fig. 5 | Representative structures from global best simulations. a, b, LAO protein (seed 1790954) (a) and ADK (seed 1795691) (b). Structures with maximum GDT are shown (coloured) and are almost identical to the respective target states (grey). The colouring indicates the displacement of each alpha carbon in the simulated structure with respect to the target state. The average alpha-carbon displacement in each coloured structure with respect to each grey structure is given. Structures are visualized with PyMOL.8.

The RMSE is the minimal mass-weighted average distance between N atoms (usually backbone or alpha carbon) of two superimposed structures over all possible spatial translations and rotations,

\[
\text{RMSE} = \min_{\text{trans, rot}} \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left( \sum_{n=1}^{N} m_n |r_{i,n} - r_{0,n}|^2 \right)}.
\]

where \(M = \sum_{n=1}^{N} m_n\) and \(m_i, r_{i,n}\) is the mass of atom \(i\), \(r_{i,n}\) and \(r_{0,n}\) are the positions of atom \(i\) in the mobile and reference structure, respectively. Holo/apo LAO protein and open/closed ADK have alpha-carbon RMSD values of 4.7 Å and 7.1 Å, respectively.

A disadvantage is that RMDS correlates strongly with the largest displacement between two structures, and small numbers of displaced atoms induce large changes. We use GDT\textsubscript{SA35}\textsuperscript{61} as the main target metric as it more accurately accounts for local misalignments.

**Implementation.** FLAPS is implemented as a stand-alone solver in Hyppopy, a Python-based hyperparameter optimization package available at https://github.com/MIC-DKFZ/Hyppopy. Hyppopy provides tools for blackbox optimization. It has a simple, unified application programming interface (API) that can be used to access a collection of solver libraries. Our implementation of FLAPS is available on GitHub\textsuperscript{8}. We implemented a Message-Passing Interface (MPI)-parallel version of the code using a sophisticated parallelization architecture as described in Supplementary Fig. 13. Available compute nodes comprising a given number of processors are divided into blocks, each of which corresponds to one particle in the warm. Within one block, the simulation itself runs on a single core, while all the other cores process the generated frames in the trajectory on the fly. This results in a massive reduction in runtime.

The experiments were run on the ForHLR II cluster system located at the Steinbuch Centre for Computing at Karlsruhe Institute of Technology. The system comprises 1,152 thin, that is, solely central processing unit (CPU)-based, compute nodes. Each node is equipped with two 10-core Intel Xeon E5-2660 v3 Haswell CPUs at 3.3 GHz, 64 GB of DDR3 main memory and 4x Mellanox 100-Gbit EDR InfiniBand links. The software packages used were a RHEL Linux with kernel version 4.18.0 and Python 3.6.8.

Each run used 51 compute nodes (1,020 cores in total). Owing to the magnitude of metadata and I/O operations, we used a private on-demand file system (BeetFS On-Demand) with a stripe count of 1, where one node was reserved for the metadata server\textsuperscript{41}. Each block in the underlying simulator-worker scheme consisted of five nodes, that is, 100 cores (one simulator, 99 workers). Each run cost ~40,000 CPU hours. For the presented application, we used cognitive acceleration coefficient \(\phi = 2.0\) and social acceleration coefficient \(\phi_s = 1.5\) in the particle update (Algorithm 1). The complete set-up, including all PSO hyperparameters, is available on GitHub\textsuperscript{5}.

**Data availability.** The software for the metaheuristic molecular dynamics parameter optimization of SAXS-guided structure-based protein simulations used in this work\textsuperscript{60} is publicly available at https://github.com/FLAPS-NNMI/FLAPS-sim_setups/releases/tag/v1.0 and published under the Creative Commons Attribution 4.0 International Public License.

**Code availability.** All code used in this work is publicly available at https://github.com/FLAPS-NNMI/FLAPS-sim_setups/releases and published under the New BSD Licence. The MPI-parallelized FLAPS solver is implemented in Optunity, a hyperparameter tuning package for Python. Our extended version of Optunity\textsuperscript{58} is available at https://github.com/FLAPS-NNMI/FLAPS-optunity/releases/tag/v1.0 and integrated into Hyppopy, a Python-based toolbox for blackbox optimization. Our extended version of Hyppopy\textsuperscript{57} is available at https://github.com/FLAPS-NNMI/FLAPS-Hyppopy/releases/tag/v1.0.
Articles

57. Weiel, M. et al. Minimal dataset repository for reproduction of presented results (FLAPS-NMI@Github/FLAPS-sim_setups, 2021); https://doi.org/10.5281/zenodo.4773999; https://github.com/FLAPS-NMI/FLAPS-sim_setups/releases/tag/v1.0

58. The PyMOL Molecular Graphics System, Version 1.8 (Schrödinger, 2015).

Acknowledgements

We are thankful for computing time on the ForHLR II computer cluster, funded by the Ministry of Science, Research and Arts Baden-Württemberg and by the Federal Ministry of Education and Research. M.W., M.G., A.K. and D.C. acknowledge support by the Helmholtz Association’s Initiative and Networking Funds under project no. ZT-I-0003. M.G. and D.C. acknowledge support by the Helmholtz AI platform grant. The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript. M.W. gratefully acknowledges support by the DFG Research Training Group 2450.

Author contributions

M.W., M.G. and A.S. conceived the study. M.W., M.G., A.K. and R.F. developed the methodology. M.W., A.K. and D.C. implemented the FLAPS solver in Hyppopy. M.W. conducted the optimization runs under the supervision of M.G. and A.S. All authors discussed the results. M.W., M.G. and D.C. wrote the manuscript. M.W. designed and produced figures. All authors read and approved the manuscript.

Funding

Open access funding provided by Forschungszentrum Jülich GmbH (4205).

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-021-00366-3.

Correspondence and requests for materials should be addressed to A.S.

Peer review information Nature Machine Intelligence thanks the anonymous reviewers for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2021