UPPSALA
UNIVERSITET

If Newton were a chemist

Molecular force fields & genetic algorithms

Julián R. Marrades Furquet

Project in Computational Science: Report

January 2022

PROJECT REPORT

Institutionen för informationsteknologi

UPPSALA
UNIVERSITET

Uppsala Universitet
Department of Information Technology
Project in Computational Science
1TD307

If Newton were a chemist
Molecular force fields & genetic algorithms

Julián R. Marrades Furquet[1]

[1]**Department of Information Technology, Uppsala Universitet, Uppsala, Sweden**

ABSTRACT

As Richard Feynman once said, everything that living things do can be understood in terms of the jiggling and wiggling of atoms. In this paper, we develop tools for parameterizing molecular force fields. Having one hundred years worth of biomolecular data at hand, we embark on a journey to overcome the limitations of local search heuristics by implementing a hybrid global optimizer based on genetic algorithms and Markov chain Monte Carlo methods.

Keywords: molecular force fields, hybrid optimizer, global search, Monte Carlo, genetic algorithms.

Abbreviations: **MEP** molecular electrostatic potential, **ACM** Alexandria Charge Model, **DFT** density functional theory, **EEM** electronegativity equalization method, **SQE** split charge equilibration, **MCMC** Markov chain Monte Carlo, **GA** genetic algorithm.

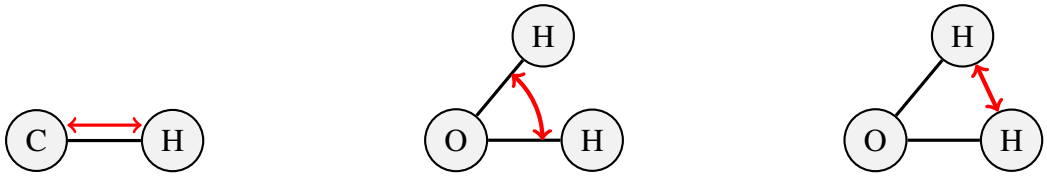
1 INTRODUCTION

If one wants to predict the movement of celestial bodies, Newton's law of universal gravitation may be the first tool that comes to mind. It states that the gravitational potential energy between two masses m_1 and m_2 can be written as

$$U = -G\frac{m_1 m_2}{r}, \tag{1}$$

where r is the distance between their mass centers, and G is the gravitational constant. The value of the latter was experimentally determined by Henry Cavendish in 1798 [1, 2].
As any mathematical model, it offers a trade-off. On the one hand, it is easy to understand and use, as it has a simple functional form and just one parameter (G). One the other hand, it is not applicable for strong gravitational fields, short distances, and high speeds. In such cases, general relativity, a more complex scheme, should be used.

When it comes to studying atomic trajectories, the situation is no different. One can use quantum mechanical methods, which are computationally expensive but accurate. Alternatively, one can employ a "Newtonian" approach by considering high-level atomic interactions (see Fig. 1), each with its own parameter. The latter models, called molecular force fields, offer simplicity at the expense of accuracy.



(a) Bond stretching. **(b)** Angle bending. **(c)** Charge.

Figure 1. Examples of high-level atomic interations.

There are vast amounts of papers that aim to improve the performance of molecular force fields. They do so by (1) altering the selection of high-level interactions, (2) using larger datasets, and (3) polishing the parameterization process. In this work, our goal is to enhance the replication of the molecular electrostatic potential (MEP) by refining the charge parameters of the Alexandria Force Field, a model under current development at the Van der Spoel Lab. To avoid convergence to a local minima during the parameterization process, we extend the previously used Markov chain Monte Carlo (MCMC) local search algorithm into a hybrid global search heuristic.

Hereinafter, this paper is organized as follows. Section 2 covers the charge component of the Alexandria Force Field. Section 3 explains the MCMC method and our hybrid approach. Section 4 carries out a simple experiment to compare both optimizers. Finally, Section 5 provides a summary and guidance for further work.

2 ALEXANDRIA CHARGE MODEL (ACM)

The MEP of a point in space $\boldsymbol{r_0}$ by effect of a molecule is given by

$$\Psi(\boldsymbol{r_0}) = \frac{1}{4\pi\varepsilon_0}\left[\sum_{i=1}^{N_a}\frac{z_i}{\|\boldsymbol{x_i}-\boldsymbol{r_0}\|} - \int_{\Omega}\frac{n(\boldsymbol{r})}{\|\boldsymbol{r}-\boldsymbol{r_0}\|}d\Omega\right], \tag{2}$$

where ε_0 is the permittivity of vacuum, N_a is the number of atoms, z_i are the nuclear charges, $\boldsymbol{x_i}$ are the nuclear coordinates, $n(\boldsymbol{r})$ is the electron density, and Ω is the entire space.
Eq. 2 can be evaluated using density functional theory (DFT) as well as wave-function quantum chemistry, though at a significant computational cost. Such evaluations, usually stored in databases to facilitate their re-use, act as reference for parameterizing molecular force fields. In this paper, we employ the Alexandria Library [3].
For molecular dynamics simulations, it is common to employ molecular force fields, where electrons are not taken into account in an explicit manner. Thus, instead of DFT, we take a different approach.

When no external electrostatic field is present, the electronegativity equalization method (EEM) [4, 5] defines the molecular energy in terms of its atomic charges q_i as follows:

$$E_{\mathrm{EEM}}(q_1, q_2, \ldots, q_N) = \sum_{i=1}^{N_a}\left[\chi_i q_i + \frac{1}{2}\eta_i q_i^2 + \frac{1}{2}\sum_{j\neq i} q_i q_j J_{ij}\right], \tag{3}$$

where J_{ij} is the Coulomb interaction between atoms i and j, and χ_i and η_i are the electronegativity and atom hardness parameters, respectively.
The Coulomb interaction takes the form

$$J_{ij} = \frac{1}{4\pi\varepsilon_0 r_{ij}}\mathrm{erf}\left(\beta_{ij} r_{ij}\right), \tag{4}$$

where r_{ij} is the distance between atoms i and j, $\mathrm{erf}(\cdot)$ is the Gauss error function, and β_{ij} can be written as

$$\beta_{ij} = \frac{\beta_i \beta_j}{\sqrt{\beta_i^2 + \beta_j^2}}, \tag{5}$$

introducing a new parameter β_i per atom, which represents the charge distribution width.

The EEM was extended by Nistor *et al.* [6], yielding the split charge equilibration (SQE) scheme. They introduced the concept of charge transfers p_{ij}, which stand for the charge transferred from atom i to atom j. Years later, Verstraelen *et al.* [5] presented the SQE variant that is used in the ACM:

$$E_{\mathrm{SQE}} = E_{\mathrm{EEM}} + \sum_{i,j}^{\mathrm{bonded}} \left[\frac{1}{2} \zeta_{ij} p_{ij}^2 + \Delta\chi_{ij} \left(q_i - q_j \right) \right], \tag{6}$$

where the sum only considers covalent bonds i-j, the bond hardness parameter ζ_{ij} symbolizes the resistance against charge transfer, and $\Delta\chi_{ij}$ is an electronegativity correction parameter.

If we build a triangulation of point coordinates $\boldsymbol{r_i}$ in and around a molecule, the SQE model can be used to compute the point charge at $\boldsymbol{r_i}$, denoted by $Q(\boldsymbol{r_i})$, in terms of the charge distribution width β, the electronegativity χ and its correction $\Delta\chi$, the atom hardness η, and the bond harness ζ.
Then, the MEP at $\boldsymbol{r_i}$ can be written as

$$\Psi^{\mathrm{ACM}}(\boldsymbol{r_i}) = \sum_{i=1}^{N_a} \frac{1}{4\pi\varepsilon_0} \frac{Q(\boldsymbol{r_i})}{\|\boldsymbol{x_i} - \boldsymbol{r_i}\|}. \tag{7}$$

3 METHODS

We treat parameterization as a non-linear least-squares data-fitting problem by minimizing the following loss function:

$$L^2 = \sum_{j=1}^{N_m} \left[w_\Psi \sum_{i=1}^{N_p} \left(\Psi^{\mathrm{ACM}}(\boldsymbol{r_i}) - \Psi^{\mathrm{DFT}}(\boldsymbol{r_i}) \right)^2 + w_\alpha \sum_{k=1}^{3} \left(\boldsymbol{a}_{ii}^{\mathrm{ACM}} - \boldsymbol{a}_{ii}^{\mathrm{DFT}} \right)^2 \right], \tag{8}$$

where N_m is the number of molecules in the dataset, $\Psi^{\mathrm{DFT}}(\boldsymbol{r_i})$ is the DFT evaluation of the MEP at $\boldsymbol{r_i}$, $\boldsymbol{a}_{ii}$ are the diagonal entries of the molecular polarizability tensor from DFT and ACM, and w_Ψ and w_α are weighting factors.
The ACM contains another set of parameters α_i, representing the atomic polarizability, which are used to compute the molecular polarizability tensor.

The remainder of this section covers the MCMC local search optimizer (section 3.1), and the hybrid heuristic based on genetic algorithms (GAs) and MCMC (section 3.2).
In this paper, we assume that the reader is somewhat acquainted with Monte Carlo methods and GAs. Otherwise, we refer them to [7, 8] as well as the great deal of available online resources.

3.1 Markov chain Monte Carlo (MCMC)

The MCMC scheme receives an initial vector of parameters and, by changing one value at a time, attempts to minimize L^2. Algorithm 1 describes the method.
On lines 21-24, the temperature parameter `T` controls the probability of accepting a parameter change that does not decrease L^2 (see Fig. 2). This is our application of the Metropolis criterion [9], which allows us to explore the parameter space and lower the chances of converging prematurely to a local

Algorithm 1 MCMC

Input:
`lb`, `ub`: vectors of lower/upper bounds per parameter
`nParam`: number of force field parameters to optimize
`initParam`: inital parameter vector
`nIter`: number of iterations
`step`: maximum allowed parameter change as fraction of its range
`T`: temperature
Output:
`param`: final parameters
`bestParam`: best parameter vector found

▷ Create initial vector of parameters within their bounds and compute loss
`param` ← `initParam`
`loss` ← **L2** `(param)`
▷ Declare best parameter and loss structures
`bestParam` ← `param`
`bestLoss` ← `loss`
▷ Iterate
for `k` ← `0` **to** `nIter - 1` **do**
  for `j` ← `0` **to** `nParam - 1` **do**
    `newParam` ← `param`
    ▷ Randomly pick a parameter index
    `i` ← $U\{$`0, nParam - 1`$\}$
    `maxChange` ← `step*(ub[i]-lb[i])`
    ▷ Draw a random parameter change
    `change` ← U`[-maxChange, maxChange]`
    `newParam[i]` ← `newParam[i]+change`
    ▷ Constrain the new value between its bounds
    `constrain(newParam[i], lb[i], ub[i])`
    `newLoss` ← **L2** `(newParam)`
    `deltaLoss` ← `newLoss - loss`
    if `deltaLoss < 0` **or** U`[0, 1]` $\leq$ `exp(-deltaLoss/T)` **then**
      `param` ← `newParam`
      `loss` ← `newLoss`
    end if
    if `newLoss < bestLoss` **then**
      `bestParam` ← `newParam`
      `bestLoss` ← `newLoss`
    end if
  end for
end for

minima.

One might notice that, as long as $\mathtt{T} > 0$, convergence is not guaranteed. Hence, we apply simulated annealing. Given a starting point `annealStart` as a fraction of the total number of iterations, we lower `T` according to

$$\mathtt{T} = \begin{cases} \mathtt{T}_* \left(1 - \frac{\mathtt{iter}}{\mathtt{nIter}+1}\right) & \text{if } \mathtt{iter} < \mathtt{nIter} \\ 10^{-6} & \text{if } \mathtt{iter} = \mathtt{nIter} \end{cases}, \tag{9}$$

where $\mathtt{T}_*$ is the initial temperature and `iter` is the current iteration number. Fig. 3 illustrates this process.

3.2 Hybrid GA-MCMC

In essence, our hybrid approach consists in using the MCMC search algorithm as the mutation operator within the GA evolution process. Since we employ it as a *random alteration* engine, we will not perform simulated annealing. Algorithm 2 explains the steps of the HYBRID scheme.

4 EXPERIMENTS & RESULTS

To compare the new hybrid method with the raw MCMC heuristic, we use a selection of ten alcohols as fitting targets. Namely, 1-butanol, 1-isopropoxy-2-propanol, 1-methylcyclohexanol, 1-

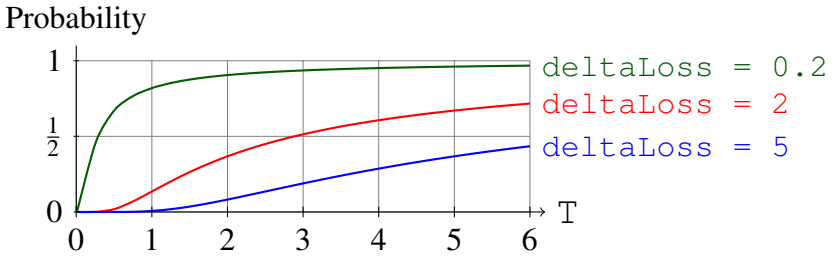
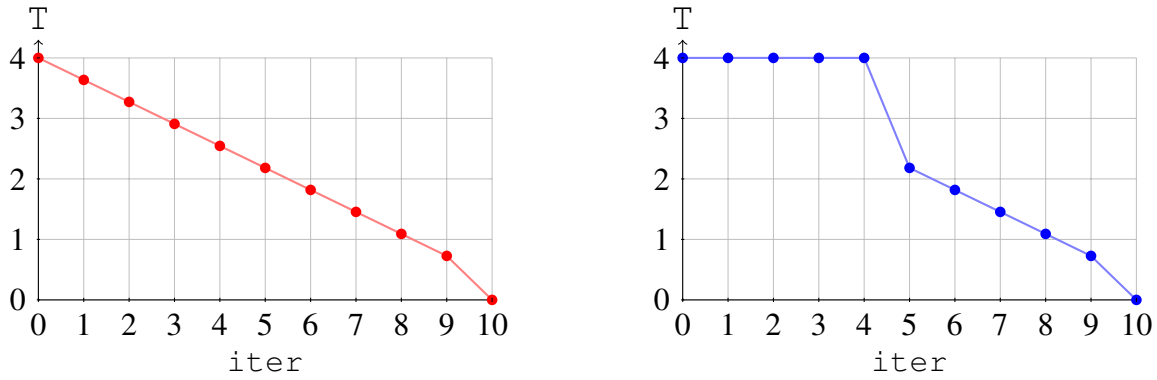


Figure 2. Probability of accepting a parameter change that does not decrease L^2 in terms of the temperature `T` for different non-negative `deltaLoss`.



(a) `annealStart` $= 0.0$. **(b)** `annealStart` $= 0.5$.

Figure 3. Simulated annealing for different `annealStart` when $\texttt{T}_* = 4$ and $\texttt{nIter} = 10$. The solid lines only act as visual aid, since $\texttt{iter} \in \mathbb{Z}^+$.

nonanol, 1-pentalon, 1-phenyl-1-propanol, 1-phenylethanol, methanol, phenylmethanol, and trans-2-methylcyclohexanol. This set yields 12 force field parameters.
Using an informed guess by domain experts as the starting force field parameter vector, we perform 50 runs of the MCMC optimizer using the hyperparameters described in Table 1. Then, we run the HYBRID scheme as specified in Table 2. We choose these particular values so that a candidate solution in both pure MCMC and HYBDRID undergoes 150 MCMC iterations.

nIter	step	T	annealStart
150	0.02	10	0.5

Table 1. Hyperparameters for the MCMC optimizer.

popSize	nGen	order	prCross	nIter	step	T
50	5	1	0.35	30	0.02	10

Table 2. Hyperparameters for the HYBRID optimizer.

Fig. 4 shows the minimum loss achieved by each MCMC run and individual in HYBRID. Although the interquartile ranges overlap, HYBRID offers an improvement thanks to its left-skewed minima distribution. Given that we only produce 50 samples, the Central Limit Theorem is not likely to hold. Therefore, we have to restrict ourselves to a visual analysis, since the results of inferential statistical

Algorithm 2 `HYBRID`

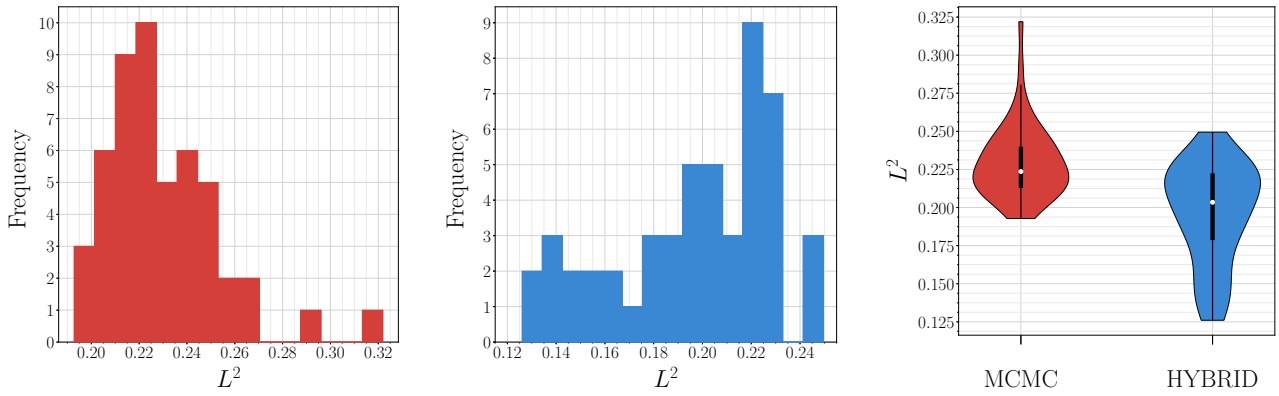
Input:
▷ Input for `GA`
`popSize`: number of individuals in the population
`nGen`: number of generations
`order`: order of the k-point crossover operator **`KPC`**
`prCross`: probability of crossover
▷ Input for `MCMC`
`lb`, `ub`, `nParam`, `nIter`, `step`, `T`
Output:
`bestInd`: best individual found

`gen` ← `0`
▷ The population is a collection of individuals, each with a parameter vector (`p`), L^2 (`L2`), and selection probability (`prob`)
`oldPop` ← `[]`
for `i` ← `0` **to** `popSize - 1` **do**
  ▷ Randomly initialize a new individual between the bounds
  `newInd` ← **INIT**(nParam, lb, ub)
  ▷ Compute L^2 of the individual
  `L2``(newInd)`
  `oldPop.add(newInd)`
end for
▷ Get best individual of the population based on L^2
`bestInd` ← `findBest(oldPop)`
repeat
  `gen` ← `gen + 1`
  `newPop` ← `[]`
  ▷ Sort population in ascending order of L^2
  `SORT``(oldPop)`
  ▷ Compute the selection probability per individual based on its rank in the sorted population
  `PROB``(oldPop)`
  for `i` ← `0` **to** `popSize - 1` **with** `i` ← `i+2` **do**
    ▷ Select parents based on their probability
    `parent1, parent2` ← **`SELECT`**`(oldPop)`
    ▷ Do crossover
    if $U[0,1] \leq$ `prCross` **then**
      `child1, child2` ← **`KPC`**`(order, parent1, parent2)`
    else
      `child1, child2` ← `parent1, parent2`
    end if
    ▷ Do mutation by calling `MCMC` with the children as starting parameter vectors
    `child1` ← **`MCMC`**`(..., child1, ...)`
    `child2` ← **`MCMC`**`(..., child2, ...)`
    `newPop.add(child1, child2)`
  end for
  `oldPop` ← `newPop`
  ▷ Compute L^2 for the new generation
  for `ind` **in** `oldPop` **do**
    `L2``(ind)`
  end for
  ▷ Check for a new best individual
  `tmpBestInd` ← `findBest(oldPop)`
  if `tmpBestInd.L2` $<$ `bestInd.L2` **then**
    `bestInd` ← `tmpBestInd`
  end if
until `gen` $\geq$ `nGen`

methods such as the t-test would not be significant.

Fig. 5 plots the trajectory of the best MCMC run and the best individual in HYBRID. After only one generation, the HYBRID individual already outperforms its competitor. Although the lack of simulated annealing causes a setback in the fourth generation, survival of the fittest quickly resolves it.
In principle, MCMC could achieve the same minimum as HYBRID in longer real time. However, when constrained to 150 iterations, MCMC reaches a loss of 0.192795 while HYBRID goes down to 0.126024.

Fig. 6 shows the HYBRID evolution as a 2D histogram. Albeit the loss distribution quickly shifts downwards, we maintain solution diversity and prevent premature convergence thanks to rank-based selection probabilities and the lack of simulated annealing during mutation.



(a) MCMC. **(b)** HYBRID. **(c)** Distribution comparison.

Figure 4. Minimum loss achieved by each run of MCMC and individual in HYBRID.

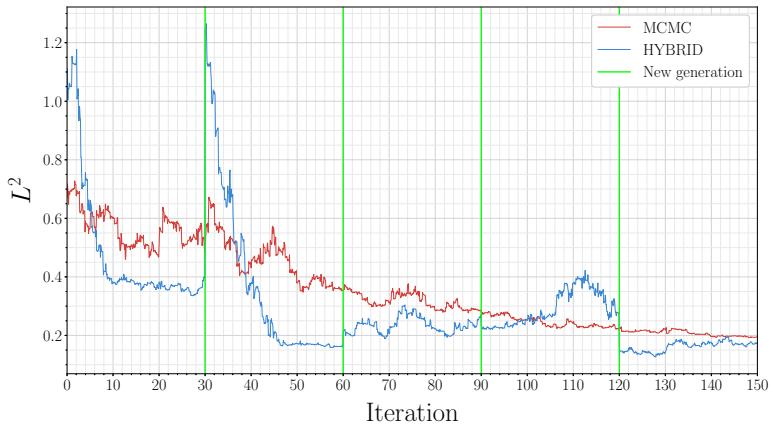


Figure 5. Best run of MCMC and best individual in HYBRID.

5 CONCLUSIONS & FUTURE WORK

In this paper, we have proposed a hybrid global optimizer based on genetic algorithms and Markov chain Monte Carlo methods. While its performance in simple scenarios is promising, further examination, in terms of quantity of runs/individuals and size of the molecule dataset, is required to draw statistically significant conclusions. For such tests, techniques such as niche penalty may be necessary to keep solution diversity.

As the amount of runs/individuals increases, a highly efficient parallel implementation will be needed to counteract the high cost of loss computation.

ACKNOWLEDGMENTS

This paper is part of a collaborative effort by researchers at the Van der Spoel Lab. The author would like to thank David van der Spoel, Marie-Madeleine Walz, Alfred Andersson, and Lisa Schmidt for their help and support.

Section 2 is partially based on Shuyi Qin's master's thesis [7].

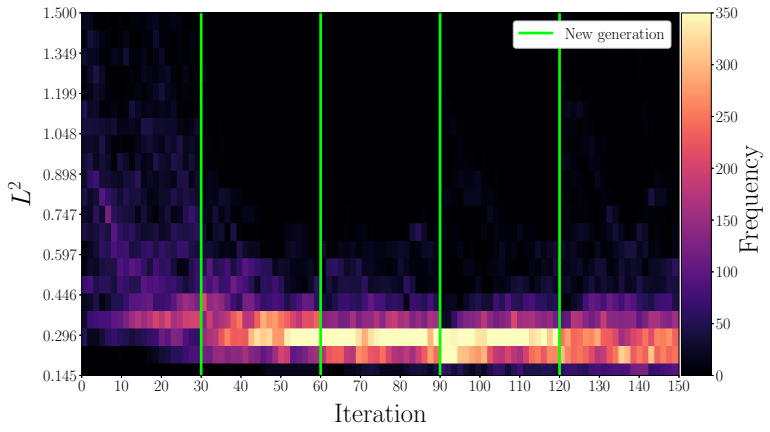


Figure 6. Evolution of HYBRID.

REFERENCES

[1] Boys, C. V. On the Newtonian Constant of Gravitation. *Nat.* **50**, 330–334 (1894). URL https://doi.org/10.1038/050330a0. DOI 10.1038/050330a0.

[2] Poynting, J. H. Gravitation. In Chisholm, Hugh (ed.). In *Encyclopædia Britannica*, vol. 12, 384–389 (Cambridge University Press, 1911), 11th edn.

[3] Ghahremanpour, M. M., van Maaren, P. J. & van der Spoel, D. The Alexandria library, a quantum-chemical database of molecular properties for force field development. *Sci. Data* **5**, 180062 (2018). URL https://doi.org/10.1038/sdata.2018.62. DOI 10.1038/sdata.2018.62.

[4] Mortier, W. J., Ghosh, S. K. & Shankar, S. Electronegativity-equalization method for the calculation of atomic charges in molecules. *J. Am. Chem. Soc.* **108**, 4315–4320 (1986). URL https://doi.org/10.1021/ja00275a013. DOI 10.1021/ja00275a013.

[5] Verstraelen, T., Van Speybroeck, V. & Waroquier, M. The electronegativity equalization method and the split charge equilibration applied to organic systems: Parametrization, validation, and comparison. *The J. Chem. Phys.* **131**, 044127 (2009). URL https://doi.org/10.1063/1.3187034. DOI 10.1063/1.3187034.

[6] Nistor, R. A., Polihronov, J. G., Müser, M. H. & Mosey, N. J. A generalization of the charge equilibration method for nonmetallic materials. *The J. Chem. Phys.* **125**, 094108 (2006). URL https://doi.org/10.1063/1.2346671. DOI 10.1063/1.2346671.

[7] Qin, S. *Sensitivity analysis in high-dimensional space*. Master's thesis, Uppsala Universitet (2021). URL http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-459514.

[8] van Dijk, S. F. *Genetic Algorithms for Map Labeling*, PhD thesis. Chapter 2 (Universiteit Utrecht, 2001). URL http://dspace.library.uu.nl/bitstream/handle/1874/864/full.pdf?sequence=1.

[9] Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biom.* **57**, 97–109 (1970). URL https://doi.org/10.1093/biomet/57.1.97. DOI 10.1093/biomet/57.1.97.