Assessing multi-objective optimization of molecules with genetic algorithms against relevant baselines

Anonymous Author(s)
Affiliation
Address
email

Abstract

Chemical design is often complex, requiring the optimal trade-off between several competing objectives. Multi-objective optimization algorithms are designed to optimally balance multiple objectives, but many chemical design approaches use the naïve weighted sum method, which is not guaranteed to give desired solutions. Here, we rigorously assess the performance of genetic algorithms for inverse molecular design, using more advanced multi-objective methods. CHIMERA and *Hypervolume* are assessed against relevant baselines for the optimization of molecules with high logP and high QED score. As a more realistic task, we also simulate a drug design campaign, optimizing for synthetically accessible molecules which bind to the 1OYT protein. We show that both methods achieve better formal optimality than the baselines and generate molecules closer to a user-specified Utopian point in property space, mimicking typical materials design objectives.

1 Introduction

Advances in machine learning (ML) and artificial intelligence (AI) have the potential to transform chemical discovery into an inverse design problem, where molecules are generated and optimized with desired properties at an accelerated rate [1]. In the context of data-driven materials discovery, AI-guided chemical design can efficiently explore chemical space while improving performance based on experimental feedback. Important application areas are drug discovery [2], catalyst design [3] and materials discovery [4]. Examples of inverse design algorithms based on deep learning (DL) include generative adversarial networks (GANs) [5, 6], variational autoencoders (VAEs) [7], and reinforcement learning [8, 9], among other approaches [1]. An older family of inverse design methods are genetic algorithms (GAs) [10–12], which have been applied in the context of chemical design for decades [13]. Despite being older, GAs often out-perform DL approaches on benchmarks [14, 15]. In particular, GAs are preferred in design campaigns where there exists insufficient data to train DL models [16], which generally require datasets on the order of 10^4–10^5 samples or more [17]. Efforts are underway to combine GAs with DL to leverage the best of both worlds [18, 19].

GAs mimic natural evolution to optimize a population of molecules against selection pressure based on a fitness function [10–12]. The fitness function is chosen to mimic the design objectives. Due to the complexity of molecular design, a single objective (*single-objective optimization*) cannot cover most multi-faceted design problems. Therefore, algorithms are needed that can optimize for multiple, potentially competing objectives simultaneously (*multi-objective optimization*). An example of such a trade-off is finding drug molecules that should both bind strongly to their intended protein target, and be synthetically accessible [20].

The goal of multi-objective optimization strategies is to find Pareto-optimal solutions, for which improvement on none of the objectives is possible without simultaneously deteriorating at least one other objective. Multi-objective optimization with GAs [21, 22, 19] have generally used variations

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

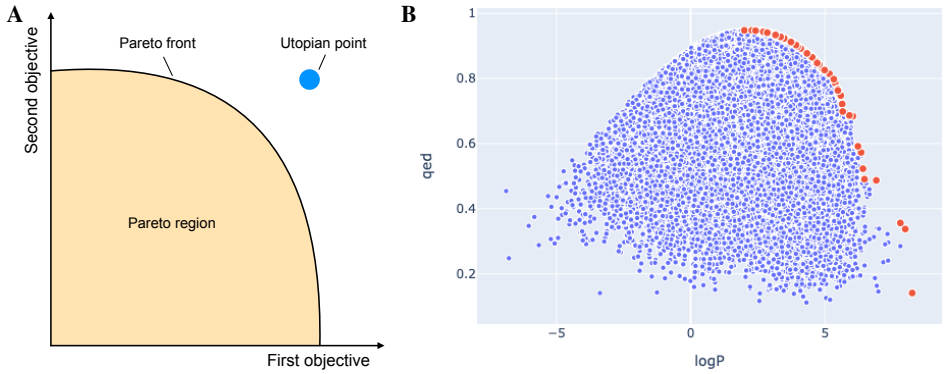

Figure 1: A) Schematic figure of the Pareto front with two objectives. B) Plot of logP and QED values for the $\texttt{ZINC}_{\texttt{red}}$ dataset, with Pareto-optimal molecules marked in red.

on the weighted sum (*WeightedSum*) or product methods [23, 24]. Although *WeightedSum* is easy to implement, it is difficult to define weights which yield desired solutions. Approaches that use more advanced multi-objective optimization algorithms often do not compare to adequate baselines [13, 3].

In this paper, we implement multi-objective optimization for the `JANUS` genetic algorithm [19] with achievement scalarizing functions (ASFs) implemented in the `OLYMPUS` benchmarking framework [25]. ASFs define a single cumulative value from the set of all objectives and encode user preferences about a multi-objective optimization problem such that their optimal solution corresponds to the desired Pareto-optimal solution. We run two inverse design tasks related to drug design and show that the more advanced multi-objective scalarizers improve upon the simple *WeightedSum* approach.

2 Methodology

In `JANUS`, the fitness of a molecule is determined by a user-defined function based on the molecular properties of interest, which returns a scalar value. `JANUS` operates on the `SELFIES` representation of molecules in order to perform mutations and crossovers, and maintains parallel exploration and exploitation populations for optimization [26, 27]. More details are available in the literature [19, 28].

We perform multi-objective optimization on two tasks: (1) a drug lipophilicity-related task, in which we maximize the quantitative estimate of drug-likeness (QED) [29], and the logarithm of the water-octanol partition function (logP), both calculated using RDKit [30]; and (2) a docking task, in which we minimize both the docking score to the human 1OYT protein (calculated with SMINA [31]) and the synthetic accessibility score (SAS) [32].

For our implementation of multi-objective optimization, an arbitrary number of fitness functions are scalarized according to a specified recipe to output a single cumulative fitness value for each molecule. Molecules are then sorted based on these scalarized values, which are used to assess their viability for propagation. We use two baseline approaches and two more advanced scalarizer approaches and assess their performance according to two metrics: (1) the hypervolume of the generated Pareto front, which balances assessment of the molecules' proximity to the Pareto front, their diversity, and spread [33, 34] (see section A.2.2 for further details), and (2) the minimum distance to a pre-selected optimal goal, commonly referred to as the *Utopian* point [35] (Figure 1A).

2.1 Optimization methods

We consider three ASFs in our experiments: *WeightedSum*, *Hypervolume*, and `CHIMERA`. *WeightedSum* constitutes a baseline approach considering its typical, simple usage for multi-objective optimization. The weight vector is set as the ratio of fitness values described by the Utopian point, and is elaborated upon in Appendix Sec. A.2.1, and maintained across both analyses for consistency. *Hypervolume* aims to find solutions that maximize the dominated hypervolume indicator. `CHIMERA` is a lexicographic ASF which organizes objectives in a hierarchy of importance and constructs a cumulative function according to user-specified tolerances on individual objectives. Tolerances for `CHIMERA` were set as the Utopian point coordinates. As compared to *WeightedSum*, setting these tolerances does not

require knowledge of the relative scale of the objectives, which is not readily available without a large dataset like ZINC. More information on the ASFs used in this work is given in Appendix Sec. A.2. We also formulate a *Random* baseline, in which fitness values for each generated structure are sampled randomly from $[0, 1]$. This approach mimics a random generation of structures as all molecules have equal chance to proceed to the next generation.

2.2 Experiments

For the experiments, each approach was initialized with a starting population of 200 molecules which were allowed to evolve for 100 and 50 generations for the lipophilicity and docking tasks, respectively. To curate the starting population, the objectives for each task were computed for a reduced ∼250,000 molecule subset of the ZINC dataset [36] following Gómez-Bombarelli *et al.* [7], here dubbed $\texttt{ZINC}_{\texttt{red}}$. The starting population was chosen to represent the worst part of the dataset – hypervolume values in the lowest 10th percentile – to make the optimization more difficult. For each task, the optimization campaign was repeated ten times, the data produced being analysed for both metrics listed below, and we report both the means and standard deviations across the runs.

2.2.1 Hypervolume of Pareto front

In this analysis, all molecules generated by JANUS for each run were collected, along with the associated objective values. The Pareto front was generated and the hypervolume of the Pareto region was calculated using OLYMPUS. The worst objective values in the original $\texttt{ZINC}_{\texttt{red}}$ dataset were used as a reference point for calculating the hypervolume. As a baseline, we generated the Pareto front for the reduced $\texttt{ZINC}_{\texttt{red}}$ dataset (Figure 1B) and computed its hypervolume (Table 1). Approaches that generate Pareto fronts with larger hypervolumes are generally considered better at multi-objective optimization for a particular problem.

2.2.2 Distance to Utopian point

Based on the Pareto front extracted from the reduced $\texttt{ZINC}_{\texttt{red}}$ dataset (Fig 1B), a Utopian point outside of the front was selected by visual inspection. The Utopian points were QED of 0.6 with logP of 10, and SAS value of 1.0 with docking score of -20 for the lipophilicity and docking tasks, respectively. These can be seen as a reasonably challenging design goals which improve upon the existing dataset. For a maximization problem with n objective functions, the distance between a generated molecule's objective values and the Utopian point is computed as

$$D(\mathbf{y}, \mathbf{y}_0) = \sqrt{\sum_{i=1}^{n} \left(\frac{\max(0, y_{i,0} - y_i)}{y_{i,0}} \right)^2}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{y}_0 \in \mathbb{R}^n$ are the objective space coordinates of the generated molecule and Utopian point, respectively. The $\min$ function is used in place of $\max$ for a minimization problem. With this metric, objective values that are as good or better than the Utopian point are considered equally advantageous.

3 Results and Discussion

The results for the lipophilicity task are shown in Figure 2A-C, and for the docking task in Figure 2D-F. For the lipophilicity task, CHIMERA allows for optimization to be guided most quickly and closely to the specified Utopian point, validating CHIMERA's use of a hierarchy of objectives and specified threshold. Importantly, CHIMERA determines when one objective is satisfied and can then focus on the remaining objectives, while the other algorithms would continue optimization of individual objectives beyond the Utopian point. Furthermore, the results over ten runs for CHIMERA produced the lowest variance. The *Hypervolume* approach gives a comparable distance, albeit with higher variance. In the docking task, the differences are less pronounced, with *WeightedSum* and CHIMERA achieving similar minimal distances to the Utopian point, on average (Figure 2D-E). This indicates that appropriate weights for *WeightedSum* were chosen for this task. However, as observed in the lipophilicity task, the same method of selecting weights does not guarantee good performance.

Looking next at the hypervolume for the lipophilicity task, molecules generated by CHIMERA and *Hypervolume* produced the Pareto fronts with the largest hypervolume on average (Table 1 in the

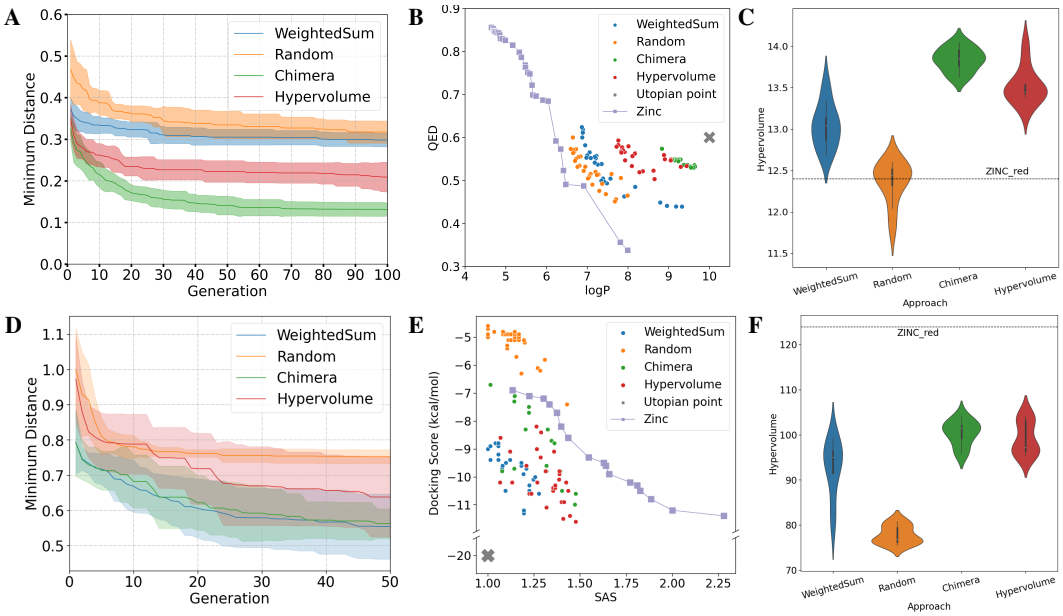

Figure 2: A) Minimum distance to Utopian point as a function of the number of generations, averaged over ten runs with standard deviation given by shaded region. Lower is better. B) The 30 closest molecules to the Utopian point in property space for each approach. The Pareto front of $\texttt{ZINC}_{\texttt{red}}$ dataset is also shown. C) Hypervolumes of generated Pareto fronts over ten runs. Higher is better. D-F) The corresponding results for the docking task. Note that $\texttt{ZINC}_{\texttt{red}}$ has large hypervolume due to points with higher SAS, which were not prioritized by the ASFs.

Appendix). While the best run for *Hypervolume* gave the Pareto front with the single highest hypervolume, the algorithm overall was less consistent (Figure 2C). Similarly for the docking task, `CHIMERA` and *Hypervolume* achieve a higher average hypervolume than both the *WeightedSum* and Random approaches (Figure 2F). Remarkably, in both tasks, `CHIMERA` seems to do be competitive with the *Hypervolume* scalarizing strategy, which explicitly tries to maximize the hypervolume itself. Both `CHIMERA` and *Hypervolume* produced significantly larger hypervolumes than the two baseline approaches *WeightedSum* and Random. A sample of the molecular structures generated by each run can be found in the Appendix, Section A.4.

4 Conclusion

In conclusion, we have demonstrated the capacity for multi-objective optimization with achievement scalarizing functions to guide molecular evolutionary algorithms towards a fuller Pareto frontier and greater proximity to a specified Utopian point. Importantly, we show that the two most advanced scalarizers, `CHIMERA` and *Hypervolume*, outperform adequate baseline methods, the naïve *WeightedSum* approach and randomly generated molecules. These results show that the more advanced methods better position researchers to solve multi-faceted inverse molecular design problems. In particular, we find `CHIMERA` and *Hypervolume* to be most suitable for design tasks with multiple, potentially competing objectives. In upcoming work, we plan to use molecular similarity metrics to further enhance the rate at which `JANUS` can produce desirable multi-objective optimization solutions.

References

[1] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, July 2018. doi: 10.1126/science.aat2663.

[2] José Jiménez-Luna, Francesca Grisoni, Nils Weskamp, and Gisbert Schneider. Artificial intelligence in drug discovery: Recent advances and future perspectives. *Expert Opinion on*

Drug Discovery, 16(9):949–959, September 2021. doi: 10.1080/17460441.2021.1909567.

[3] Ruben Laplaza, Simone Gallarati, and Clemence Corminboeuf. Genetic optimization of homogeneous catalysts. *Chemistry–Methods*, 2(6), June 2022. doi: 10.1002/cmtd.202100107.

[4] Robert Pollice, Gabriel dos Passos Gomes, Matteo Aldeghi, Riley J. Hickman, Mario Krenn, Cyrille Lavigne, Michael Lindner-D'Addario, AkshatKumar Nigam, Cher Tian Ser, Zhenpeng Yao, and Alán Aspuru-Guzik. Data-driven strategies for accelerated materials design. *Accounts of Chemical Research*, 54(4):849–860, February 2021. doi: 10.1021/acs.accounts.0c00785.

[5] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L. Guimaraes, and Alan Aspuru-Guzik. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). Preprint, Chemistry, August 2017.

[6] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-Reinforced Generative Adversarial Networks (ORGAN) for sequence generation models, February 2018.

[7] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, February 2018. doi: 10.1021/acscentsci.7b00572.

[8] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, September 2017.

[9] Luca A Thiede, Mario Krenn, AkshatKumar Nigam, and Alán Aspuru-Guzik. Curiosity in exploring chemical spaces: Intrinsic rewards for molecular reinforcement learning. *Machine Learning: Science and Technology*, 3(3):035008, September 2022. ISSN 2632-2153. doi: 10.1088/2632-2153/ac7ddc.

[10] G Zames, NM Ajlouni, NM Ajlouni, NM Ajlouni, JH Holland, WD Hills, and DE Goldberg. Genetic algorithms in search, optimization and machine learning. *Information Technology Journal*, 3(1):301–302, 1981.

[11] John R Koza and John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[12] Mandavilli Srinivas and Lalit M Patnaik. Genetic algorithms: A survey. *computer*, 27(6):17–26, 1994.

[13] Nathan Brown, Ben McKay, François Gilardoni, and Johann Gasteiger. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of Chemical Information and Computer Sciences*, 44(3):1079–1087, May 2004. doi: 10.1021/ci034290p.

[14] Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, March 2019. doi: 10.1021/acs.jcim.8b00839.

[15] Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization, June 2022.

[16] AkshatKumar Nigam, Robert Pollice, Matthew FD Hurley, Riley J Hickman, Matteo Aldeghi, Naruki Yoshikawa, Seyone Chithrananda, Vincent A Voelz, and Alán Aspuru-Guzik. Assigning confidence to molecular property prediction. *Expert opinion on drug discovery*, 16(9):1009–1023, 2021.

[17] Michael A. Skinnider, R. Greg Stacey, David S. Wishart, and Leonard J. Foster. Chemical language models enable navigation in sparsely populated chemical space. *Nature Machine Intelligence*, July 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00368-1.

[18] Tianfan Fu, Wenhao Gao, Connor W. Coley, and Jimeng Sun. Reinforced genetic algorithm for structure-based drug design. In *ICML 2022 2nd AI for Science Workshop*, 2022.

[19] AkshatKumar Nigam, Robert Pollice, and Alán Aspuru-Guzik. Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. *Digital Discovery*, 1 (4):390–404, 2022. doi: 10.1039/D2DD00003B.

[20] Connor W. Coley. Defining and exploring chemical spaces. *Trends in Chemistry*, page S2589597420302884, December 2020. doi: 10.1016/j.trechm.2020.11.004.

[21] Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

[22] Chi Yang Cheng, Josh E. Campbell, and Graeme Matthew Day. Evolutionary chemical space exploration for functional materials: Computational organic semiconductor discovery. *Chemical Science*, 2020. doi: 10.1039/D0SC00554A.

[23] I. Y. Kim and O. L. de Weck. Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, 31(2): 105–116, February 2006. ISSN 1615-1488.

[24] Carlos A. Coello Coello, Silvia Gonz&aacute, Lez Brambila, Josu&eacute, Figueroa Gamboa, Ma Guadalupe Castillo Tapia, Raquel Hern&aacute, Ndez G&oacute, and mez. Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems*, 6(2):221–237, July 2020. ISSN 21986053. Publisher: Springer.

[25] Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, Melodie Christensen, Elena Liles, Jason E Hein, and Alán Aspuru-Guzik. Olympus: A benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2(3): 035021, September 2021. doi: 10.1088/2632-2153/abedc8.

[26] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[27] AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *Chemical science*, 12(20):7079–7090, 2021.

[28] Janus. URL `https://github.com/aspuru-guzik-group/JANUS`.

[29] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[30] Rdkit: Open-source cheminformatics. URL `https://www.rdkit.org`.

[31] David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904, 2013.

[32] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):1–11, 2009.

[33] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999. doi: 10.1109/4235.797969.

[34] Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Comput. Surv.*, 52(2), mar 2019. ISSN 0360-0300. doi: 10.1145/3300148.

[35] Jasbir S Arora. 15—discrete variable optimum design concepts and methods. *Introduction to Optimum Design, 2nd ed.; Arora, JS, Ed*, pages 513–530, 2004.

[36] Michael M. Mysinger Erin S. Bolstad John J. Irwin, Teague Sterling and Ryan G. Coleman. Zinc: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012.

[37] Jieyu Lu Jianing Lu, Song Xia and Yingkai Zhang. Dataset construction to explore chemical space with 3d geometry and deep learning. *Journal of Chemical Information and Modeling*, 61 (3):1095–1104, 2021. doi: 10.1021/acs.jcim.1c000072.

[38] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms — A comparative case study. In Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature — PPSN V*, Lecture Notes in Computer Science, pages 292–301, Berlin, Heidelberg, 1998. Springer. ISBN 978-3-540-49672-4. doi: 10.1007/BFb0056872.

[39] J.D. Knowles, D.W. Corne, and M. Fleischer. Bounded archiving using the lebesgue measure. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 4, pages 2490–2497 Vol.4, 2003. doi: 10.1109/CEC.2003.1299401.

[40] Andreia P. Guerreiro, Carlos M. Fonseca, and Luís Paquete. The Hypervolume Indicator: Problems and Algorithms. *ACM Computing Surveys*, 54(6):1–42, July 2021. ISSN 0360-0300, 1557-7341. doi: 10.1145/3453474. URL `http://arxiv.org/abs/2005.00515`. arXiv:2005.00515 [cs].

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We have discussed the potential for the models to be used to design chemical weapons.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See https://github.com/natkusanda/multiobjective-janus.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 2 and the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See plots in Section 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Narval credit in Section A.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See ZINC credit in Section 2.2.
   (b) Did you mention the license of the assets? [N/A] ZINC is free to use for everyone. See license at `https://wiki.docking.org/index.php/UCSF_ZINC_License`
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See appendix section A.1, available under Apache 2.0 license
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] ZINC is free to use, while the original `JANUS` code is available under Apache 2.0 license - See A.1

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Supplemental code

All code written for this paper, including implementation of multi-objective optimisation and relevant results, can be found at `https://drive.google.com/drive/folders/1oXpu6PEkY11JzRgdq-28YPU81Pb0iU-I?usp=sharing`. The code is licensed under Apache 2.0. This code builds upon the original `JANUS` code, which is also made available under Apache 2.0 license, and can be found at `https://github.com/aspuru-guzik-group/JANUS`. ZINC is free to use for everyone. See license at `https://wiki.docking.org/index.php/UCSF_ZINC_License`. The reduced version of ZINC was downloaded from `https://github.com/aspuru-guzik-group/chemical_vae`

Please note that an expanded mutation alphabet was used to increase the presence of ring groups in all runs and approaches. This alphabet can be found in `/janus_chimera/janus/mutate.py` in the repository linked above, and the ring groups were extracted using Extended Functional Groups (EFGs) software written by Lu, et. al [37].

A.2 Details of achievement scalarizing functions

In this section, we formally define each of the achievement scalarizing functions (ASFs) used in this work. For the purpose of this discussion, we consider a molecular design task which features a set of n objective functions, $\mathbf{f} = \{f_i\}_{i=1}^n$, to be *minimized* concurrently. For molecule $\mathbf{x}$, noiseless objective values are obtained as $y_i = f_i(\mathbf{x})$. Furthermore, we assume a dataset of objective values $\mathcal{D} = \{\mathbf{y}_i\}_{i=1}^K$ corresponding to K generated molecular candidates has been collected by our optimization procedure.

A.2.1 WeightedSum

Weighted sum ASFs maps multiple objectives onto a cumulative scalar objective using a vector of weights $\mathbf{w} \in \mathbb{R}^n$ to produce the weighted sum

$$J(\mathbf{y}; \mathbf{w}) = \sum_{i=1}^{n} w_i y_i \,. \tag{2}$$

Here, we select each weight w_i according to the ratio specified by the Utopian point, scaling the relative magnitude of the ith objective.

As the Utopian point was specified as 0.6 for QED and 10 for logP, the weights were selected as 10/0.6 = 16.66 for the former objective, and 1 for the latter. As the same experimental data was used for both analyses, these weights were reflected in the Hypervolume of Pareto front analyses as well.

A.2.2 Hypervolume indicator

The hypervolume indicator is an example of a set-quality indicator, which facilitate assessment of Pareto fronts by summarizing their characteristics (such as proximity to the Pareto front, diversity and spread) with a single scalar value. Owing to its ease of interpretation, hypervolume is one of the most widely employed set-quality indicators [34].

The hypervolume indicator maps a set of objective values $\mathcal{D}$ to a measure of the region dominated by that set and bounded above by some reference point $\mathbf{r} \in \mathbb{R}^n$. Intuitively, the indicator provides a notion of the size of the covered objective space or the size of the dominated space [38, 39, 34, 40] Formally, the hypervolume indicator H given a dataset of objective value measurements $\mathcal{D}$ is

$$H\left(\mathcal{D}; \mathbf{r}\right) = \Lambda\left(\{\mathbf{q} \in \mathbb{R}^n \mid \exists \mathbf{p} \in \mathcal{D} \ : \ \mathbf{p} \leq \mathbf{q} \wedge \mathbf{q} \leq \mathbf{r}\}\right) , \tag{3}$$

where Λ is the Lebesgue measure. Here, for two objective space points $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{q} \in \mathbb{R}^n$, the expression $\mathbf{p} \leq \mathbf{q}$ is used to indicate that $\mathbf{p}$ *weakly dominates* $\mathbf{q}$, that is $p_i \leq q_i \ \forall \ 1 \leq i \leq n$. H can also be described as the union of hyperrectangles

$$H\left(\mathcal{D}; \mathbf{r}\right) = \Lambda\left(\bigcup_{\mathbf{p} \in \mathcal{D},\ \mathbf{p} \leq \mathbf{r}} [\mathbf{p}, \mathbf{r}]\right) , \tag{4}$$

where $[\mathbf{p}, \mathbf{r}]$ represents the hyperrectangle fixed from above by reference point $\mathbf{r}$ and below by $\mathbf{p}$.

In most cases (including this work), the hypervolume indicator is used as an analysis tool to asses the quality and diversity of a Pareto set after an optimiziation campaign has transpired. However, it can also be used as an ASF to which aims to find solutions which maximize the dominated hypervolume. In this case, the solutions in $\mathcal{D}$ are considered one at a time, which simplifies the calculation of H to the volume of the hyperrectangle with corners at $\mathbf{y}$ and $\mathbf{r}$,

$$H\left(\mathbf{y}\right) = \prod_{i=1}^{n} r_i - y_i \, . \tag{5}$$

Importantly, using the hypervolume indicator as a ASF this way produces a constraint that $\mathbf{y} \leq \mathbf{r}$. As such, we update $\mathbf{r}$ at each iteration such that each of its elements correspond to the maximum observed value for that objective in the optimization history $\mathcal{D}$.

A.2.3 Chimera

`CHIMERA` is an achievement scalarizing function which combines *a priori* scalarizing with lexicographic approaches, and is created for optimization problems where the objective is expensive to evaluate. `CHIMERA` allows users to organize multiple objectives into a hierarchy, i.e., $\mathbf{f} = \{f_i\}_{i=1}^n$ is replaced by $\mathbf{f} = (f_1, \ldots, f_n)$ ordered according to a descending hierarchy of importance (f_1 is more important than f_2, f_2 than f_3, and so on).

User-specified tolerance values $\mathbf{y}^{tol}$ are provided to `CHIMERA` which indicate objective value thresholds at which the user is satisfied (e.g., one may want to generate a molecule with a QED of at least 0.6, before prioritizing optimizing its lipophilicity). Whether or not some measured value of the ith objective function y_i satisfies its corresponding tolerance can be indicated by the Heaviside function, $\Theta\left(y_i^{tol} - y_i\right) = 0$ if $y_i \geq y_i^{tol}$ and 1 if $y_i < y_i^{tol}$. Alternatively, discontinuities in the cumulative function can be avoided by utilizing a smooth logistic function approximation to Θ, parameterized by smoothing parameter τ,

$$\theta\left(y_i^{tol} - y_i\right) = \left[1 + \exp\left(-\frac{y_i^{tol} - y_i}{\tau}\right)\right]^{-1} . \tag{6}$$

`CHIMERA` constructs an ASF using approximate Heaviside functions to weight objective functions $\mathbf{f}$. Importantly, the resulting ASF is sensitive to only one objective function at a time in any given parameter space region. As such, objective values y_i are shifted based on the minimum of y_{i-1} (the next most important objective function in $\mathbf{f}$) in the parameter space regions where y_{i-1} does not satisfy its corresponding tolerance y_i^{tol}. This minimum value is denoted y_{i-1}^{min}. `CHIMERA`, $\chi(\mathcal{D}; \mathbf{y}^{tol})$ is then formulated as

$$\chi(\mathcal{D}; \mathbf{y}^{tol}) = y_1 \theta_1^+ + \prod_{i=1}^{n} \left(y_1 - y_{i-1}^{min}\right) \theta_i^- + \sum_{i=2}^{n} \left(y_i - y_{i-1}^{min}\right) \theta_i^+ \prod_{j=0}^{i-1} \theta_m^- \, , \tag{7}$$

where θ_i^+ and θ_i^- are used to abbreviate $\theta\left(y_i^{tol} - y_i\right)$ and $\theta\left(y_i - y_i^{tol}\right)$, respectively. We set the smoothing parameter $\tau = 0.001$ in all our experiments.

The tolerances for the lipophilicity task were 0.6 for QED, and 1000 for logP, which entails unconstrained maximization of logP once the tolerance for QED is fulfilled. For the docking task, the tolerances were 2.0 for SAS and -1000 for docking score.

A.3 Additional details of optimization experiments

Table 1: Hypervolumes of generated Pareto fronts for the lipophilicity and docking tasks. The results are averages over ten repeats with standard deviations. Higher is better.

	Hypervolume	
Approach	QED-logP ↑	SAS-docking ↑
$\texttt{ZINC}_{\texttt{red}}$	12.40	123.91
Random	12.36 ± 0.24	77.83 ± 2.73
Weighted sum	13.04 ± 0.27	92.63 ± 6.23
Hypervolume	13.52 ± 0.22	99.27 ± 3.66
CHIMERA	13.84 ± 0.14	100.40 ± 2.00

A.4 Sample of Generated Molecules

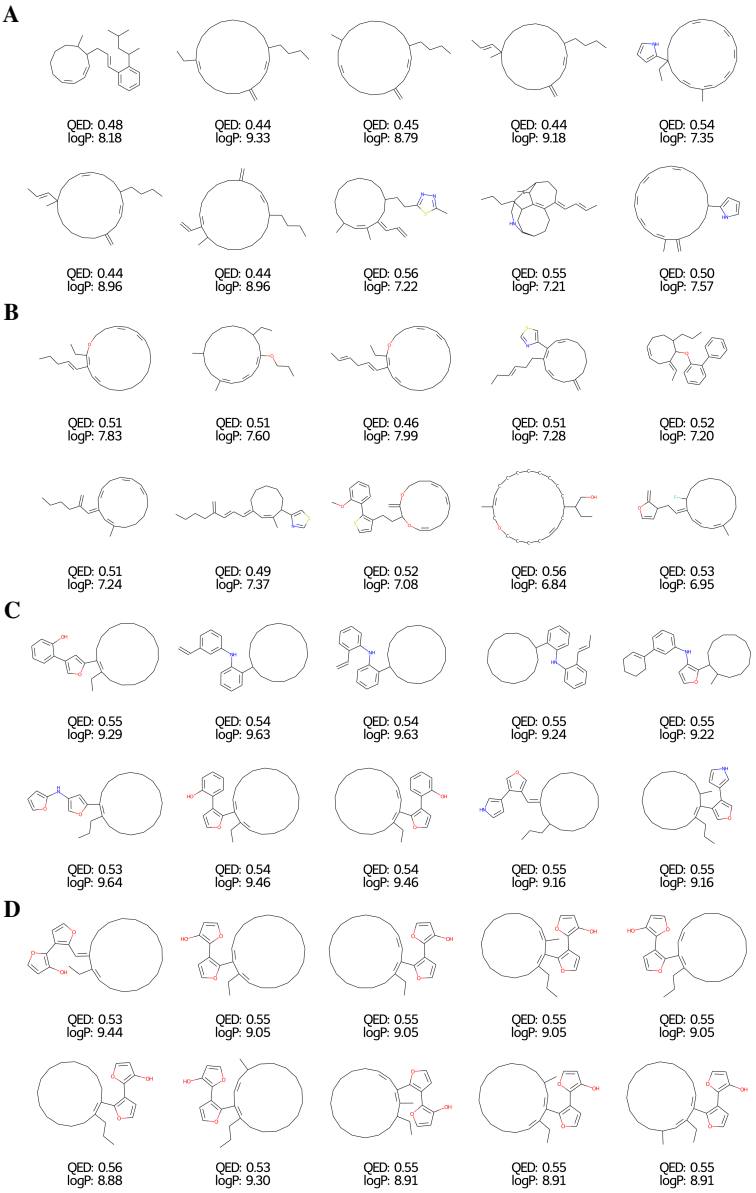

Figure 3: Sample of top ten generated molecules for the task of maximising logP and QED metrics, with minimum distance to Utopian point for A) WeightedSum B) Random Search C) Chimera D) Hypervolume. Note that the mutation alphabet was augmented to include more ring groups.

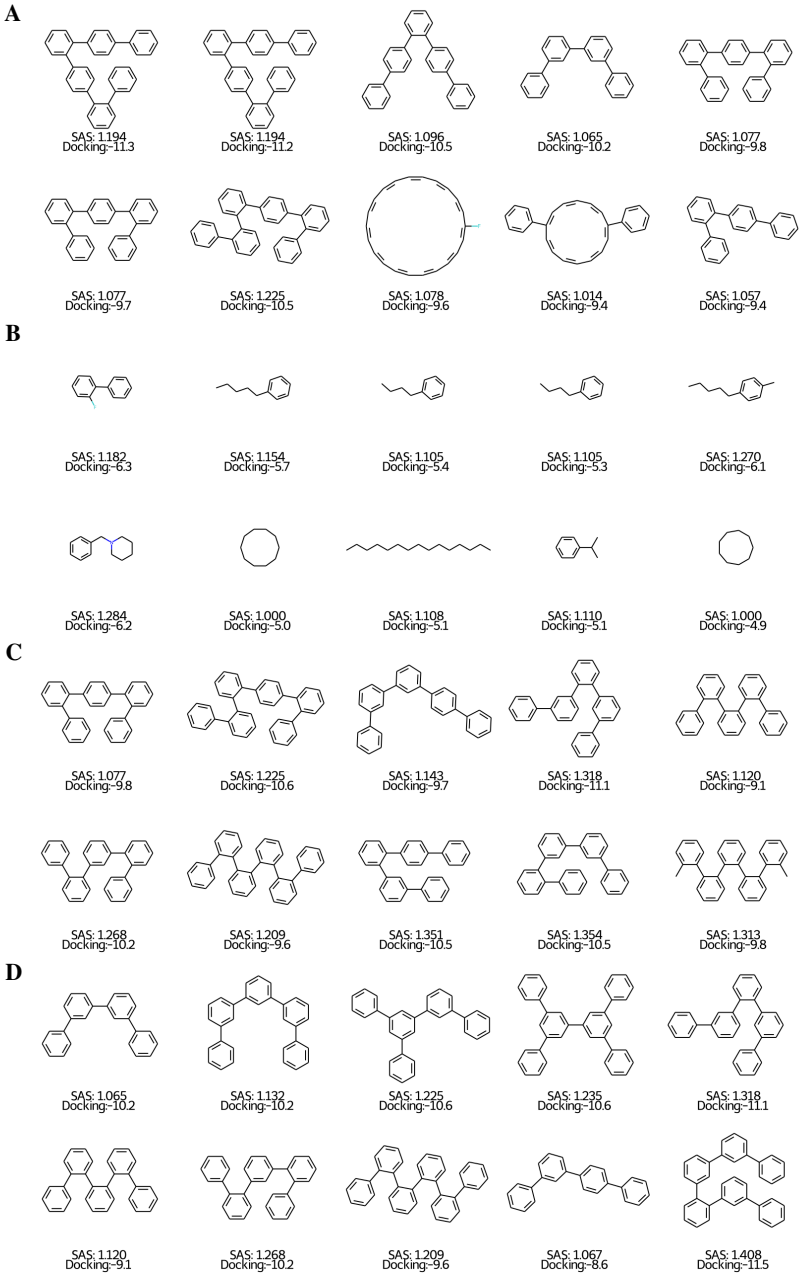

Figure 4: Sample of top ten generated molecules for the docking task of minimising SAS and the docking score metrics, with minimum distance to Utopian point for A) WeightedSum B) Random Search C) Chimera D) Hypervolume. Note that the mutation alphabet was augmented to include more ring groups. Here, we observe the use of SAS as a measure of synthetic accessibility prioritizes hydrocarbons, phenyl groups, and alternating double and single bonds. A more sophisticated measure of synthetic accessibility will give more realistic molecules.