XtalOpt Version 13: Multi-Objective Evolutionary Search for Novel Functional Materials

Samad Hajinazar[a], Eva Zurek[a,*]

[a] *Department of Chemistry, State University of New York at Buffalo, Buffalo, New York 14260-3000, United States*

Abstract

Version 13 of XtalOpt , an evolutionary algorithm for crystal structure prediction, is now available for download from the CPC program library or the XtalOpt website, http://xtalopt.github.io. In the new version of the XtalOpt code, a multi-objective global optimization functionality is implemented. This feature is designed to facilitate the search for (meta)stable phases of functional materials through optimizing the enthalpy of system as well as any desired properties that are specified by the user. The code is also able to perform a constrained search by filtering the candidate structures based on user-specified criteria, while optimizing multiple objectives. Here, we present the implemented formalism and technical details about using it, and briefly overview additional improvements that are introduced to the new version of XtalOpt .

Keywords: Evolutionary structure prediction; Multi-objective global optimization; Prediction of functional materials.

NEW VERSION PROGRAM SUMMARY

Program Title: XtalOpt
CPC Library link to program files: (to be added by Technical Editor)
Developer's repository link: https://github.com/xtalopt/XtalOpt
Code Ocean capsule: (to be added by Technical Editor)
Licensing provisions: 3-clause/BSD
Programming language: C++
Journal reference of previous version: Computer Physics Communications 237 (2019) 274–275
Does the new version supersede the previous version?: Yes
*Reasons for the new version:** Implementation of multi-objective evolutionary search algorithm
Summary of revisions: A general user-friendly scheme for multi-objective search, Various improvements to user interface and functionalities, bug fixes.
Nature of problem: The XtalOpt algorithm is designed to search for new (meta)stable phases of crystal structures. Driven by the technological needs, however, finding new functional materials is an active research area in the community, and an important challenge to overcome.
Solution method: The XtalOpt algorithm is extended by introducing a generalized scalar fitness function, where a set of user-specified objectives contribute to the fitness value for candidate structures. Through using this generalized fitness value, the structural motifs that posses the desired characteristics are promoted and the search is directed towards the regions of the energy landscape of higher relevance in terms of target properties.

1. Introduction

Computational prediction of novel materials from first-principles has seen tremendous growth and success over the past decades [1, 2, 3, 4]. Owing to the abundance of computational resources, density functional theory (DFT) calculations are now more accessible for mapping the Born-Oppenheimer potential energy surface (PES) [5] of various systems. Locating the local and global minima of this energy landscape corresponds to finding the metastable and stable phases of the system, respectively.

Various global optimization (GO) algorithms have been proposed and utilized to search for the minima of the energy landscape, e.g., random search [6], simulated annealing [7], metadynamics [8], minima hopping [9], basin hopping [10], particle swarm optimization [11], and evolutionary algorithms [12, 13, 14].

Common GO strategies are traditionally designed and implemented to minimize only the enthalpy. Such a "single-objective" optimization is typically accom-

[*] Corresponding author.
E-mail address: ezurek@buffalo.edu

plished by using the enthalpy as a measure of the "suitability" of the candidate structure(s). Using this measure, the search algorithm can enhance promising structural motifs or pursue more viable directions in the PES in searching for structures of lower enthalpy.

A demonstrative example is the family of evolutionary search (ES) algorithms which are iterative stochastic approaches to global optimization. The workflow of a typical ES is as follows [15]: (i) generate a population of candidate structures, (ii) locally optimize the structures and evaluate their enthalpy, (iii) rank the structures based on their enthalpy and assign a fitness value to population members such that the structure of the lowest enthalpy is the fittest one, (iv) select a pool of parent structures from the population randomly, with the chance of each structure for entering the pool being proportional with its fitness, (v) generate a set of new structures by applying variations of bio-inspired genetic operations, i.e., randomly distorting a parent structure (mutation) and combining two parent structures (breeding). The steps (ii)-(v) are repeated until a pre-defined stopping criterion is satisfied.

Effectiveness of the GO algorithms in predicting new materials is demonstrated through numerous studies [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]. Despite the promising performance of GO algorithms in finding novel materials, technological needs require the exploration of materials that are not only (meta)stable but also possess desired characteristics, e.g., higher hardness, higher superconducting transition temperature, desired symmetry, etc. In essence, this is equivalent to searching for phases of materials that are optimum in terms of several -often conflicting- objectives. Sometimes referred to as the inverse design problem, this is an open challenge to the computational materials community.

An immediate strategy for using the GO techniques for finding functional materials consists of performing GO to optimize an objective (e.g., enthalpy) and assess the suitability of candidate structures in terms of other characteristics. A constrained search in nature, this approach has been used previously for the design and prediction of phases with desired electronic structure [30, 31], super-dense [32] and super-hard [33] materials, high dielectric [34, 35] structures, and metastable phases featuring desired crystalline order [27]. Other alternatives have been suggested, e.g., encoding the desired property into the structural information and evolving them during the search process [36]. Compared to the secondary assessment workflow, this approach is more effective in guiding the search algorithm. Nevertheless, it requires a problem-specific customization in the description of the structural properties and possibly the search algorithm itself.

A more natural integration of the desired properties in the GO process can be achieved by simultaneous optimization of all target objectives. This can be addressed using the family of "multi-objective" global optimization (MOGO) algorithms [37, 38]. Since the problem does not involve only one objective for which a global optimum is being sought, typical MOGO algorithms aim to find a set of solutions that offer the best trade-off between various objectives. A common approach is to search for a set of solution candidates, known as Pareto optimum solutions, such that for any solution no single objective can be improved without worsening other objectives [39]. Among the MOGO approaches, in particular, the multi-objective evolutionary search (MOES) algorithms [40, 41, 42] are more popular for being inherently population-based and straightforward to implement.

Various MOGO algorithms have been utilized for materials discovery [43]. Most notable are several studies dedicated to the prediction of new functional materials through implementations of Bayesian global optimization [44, 45, 46] and evolutionary and differential evolution algorithms [47, 48, 49, 50, 22, 51].

The previous version of the XtalOpt code [52, 53] included an implementation of the MOES algorithm to search for (meta)stable phases of hard materials. In this scheme, the Vickers hardness provided by the AFLOW machine learning (ML) platform [54] was used in combination with the enthalpy to evaluate the fitness of candidate structures. Global optimization of this fitness function successfully predicted a new superconducting phase of Carbon [55].

In this work, we present a general implementation of the XtalOpt MOES algorithm. The new implementation is designed to (i) accommodate an arbitrary number of target objectives, (ii) allow the user to introduce any desired objective as long as it can be represented with a numerical value, and (iii) offer a general and easy-to-setup interface that can be used with any external code. Further, the MOES in XtalOpt can perform various types of optimization (minimization or maximization) depending on the target objective and can facilitate a constraint search by filtering the pool of structures based on the user-specified criteria. This functionality is available in both the command-line interface (CLI) and graphical user interface (GUI) modes of the XtalOpt code.

The remainder of this article describes the MOES functionality, as well as a series of new features that are implemented in XtalOpt . In Section 2 a general introduction to the implemented MOES scheme is offered. The type of the generalized fitness function and general properties of the external codes to be used with XtalOpt are described, as well as the main search parameters that the user needs to provide to XtalOpt for MOES. Sections 3 and 4 introduce a detailed description of the MOES in the XtalOpt CLI mode and an example of the user-specified interface for calculating the objectives. The constrained search functionality of the code is introduced in Section 5, and Section 6 describes the changes in the -legacy- hardness optimization. A brief account of the output of the MOES along with the general error handling during the search process is given in Section 7. Finally, Section 8 introduces the search setup and the accompanied changes in the GUI mode of the code, and Section 9 introduces a series of options and functionalities that are added to the XtalOpt code.

2. Multi-objective search

2.1. Generalized fitness function

The single-objective ES algorithm in XtalOpt relies on a fitness function given by:

$$f_s = \frac{H_{max} - H_s}{H_{max} - H_{min}} \tag{1}$$

where the f_s is the calculated fitness value for the s^{th} structure with an enthalpy of H_s, and H_{max} and H_{min} are the maximum and minimum of enthalpy of the candidate structures, respectively. The calculated fitness values for the structures, hence, is normalized to a value between 0 and 1.0. The parents pool is then selected by filtering the structures with fitness values above a randomly generated threshold value. This gives a relatively higher chance of being the parent of a new structure to those which are more "fit", i.e., have a lower enthalpy.

The MOES, however, needs a different measure for evaluating the suitability of candidate structures. Since a search for new stable phases is typically followed by high-accuracy local optimization of the best candidate structures, we found it sufficient to use a scalar generalized function that merges all objectives to obtain a single measure of fitness for each structure. This requires defining a set of weights for the objectives, reflecting on the relative significance of them. Assuming that $\{X\}$ and $\{Y\}$ represent two set of objectives to be minimized and

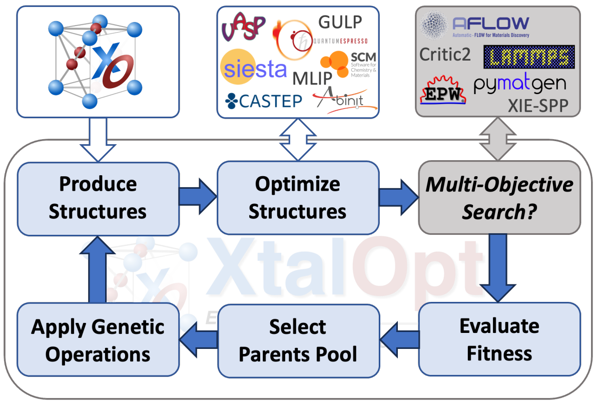


Figure 1: Workflow of the XtalOpt MOES algorithm. After producing the initial set of structures, structural optimizations can be performed through various first-principles approaches or classical potentials. The locally optimized structure is then passed on to external code(s) introduced by the user for calculating desired properties (a few are shown as examples). Subsequently, the fitness function is evaluated for all structures and the parents pool is selected accordingly. The new structures are being generated by applying various evolutionary operations to the parents pool.

maximized, respectively, the generalized fitness function for the s^{th} structure can be obtained as:

$$f_s = \sum_i w_X^i \left(\frac{X_{max}^i - X_s^i}{X_{max}^i - X_{min}^i} \right) + \sum_j w_Y^j \left(\frac{Y_s^j - Y_{min}^j}{Y_{max}^j - Y_{min}^j} \right) \tag{2}$$

where $\{X_s\}$ and $\{Y_s\}$ are the target values of the corresponding objectives calculated for the s^{th} structure, and $\{w\}$ is the set of weights associated with the objectives.

Given a total weight of 1.0 for all objectives, the above fitness measure will be normalized to $[0, 1.0]$. As a result of using this fitness value for selecting the parents pool, effectively, the MOES is converted to a single-objective search for which the above-mentioned workflow of the ES can be followed. Figure 1 illustrates an schematic of XtalOpt algorithm for MOES. Once the structures are locally optimized, the code makes structural information available to the external codes that are calculating the desired target objectives. These values are then read in by XtalOpt and used to calculate the generalized fitness function for producing the parents pool.

In practice, enthalpy is always set to be one of the objectives for minimization. While this is not necessary in our implementation, inclusion of enthalpy enables the search for lower energy structures that potentially feature the desired target properties. This promotes a series of low energy structures in the parents pool that have a higher suitability in terms of other objectives, while taking only enthalpy into account they were not as likely to be the source for generation of new structures. This, effectively, increases the chance of finding "metastable" phases with desired target properties.

For utilizing the XtalOpt MOES, the user needs to introduce an external code that produces the target value for each objective considered in the search, as well as the corresponding weight to be used in calculating the fitness values. More details about these steps is provided in the following.

2.2. User-defined objectives

For any property that the user wants to use as an objective in the MOES, there should be an executable user-provided script or code that (i) calculates that property from the output of a regular XtalOpt search, and (ii) produces an output file with a single number (integer or double) as the value of the desired objective. This output file will be read in by the XtalOpt code and will be used for determining a structure's fitness for procreation or for filtering the parents pool (see Section 5).

Essentially, after each local relaxation that can be performed with any of the total energy calculation methods available in XtalOpt , the code generates a structure file `output.POSCAR` (this file uses the Vienna *Ab initio* Simulation Package (VASP) [56, 57] format). The user-provided script should be able to read/use this file (or convert it to another structural data format if needed) to perform the intended calculations and produce the output file.

The user-provided script should be an executable script and be available in a path accessible either in the local path (for a local run) or on the cluster access path (for a remote queue). XtalOpt will call this script automatically and will read and use its output for the MOES.

The output file generated by the user-provided script should be a text file with the numerical value of the corresponding objective written as the first entry of the first line of the file.

In general, the script can be simply a sequence of commands that use the `output.POSCAR` file and call some program to produce a result; or can be a bash script that actually generates a cluster job file and submits the job to the computational cluster on its own if the intended calculations are computationally demanding (samples are provided in Section 4).

In case the particular calculations in the script need more input data (e.g., *ab initio* charge density, density of states, etc.), the user-provided script should include introductory steps to generate them, or alternatively the user can add appropriate entries to the XtalOpt job template used in the structure search to produce these data.

2.3. Multi-objective search parameters

In order to invoke the MOES functionality, for every desired objective, the user should provide the following information to the XtalOpt code:

- **optimization type**: instruction for the code on how to use the result of calculating an objective in determining the fitness function. XtalOpt can minimize or maximize an objective, use its value for filtering the parents' pool (see Section 5), or maximize the AFLOW-ML hardness (see Section 6).
- **path to the user-defined script**: the full path to the script corresponding to the introduced objective. The script will be automatically run by the XtalOpt code after local relaxation, and calculates the desired property.
- **script's output filename**: the name of the file generated by the script that contains the result of the calculation for the corresponding objective.
- **optimization weight**: a number between 0.0 and 1.0 that will be used as the weight for the corresponding objective in calculating the fitness function. The total weight of all objectives should not exceed 1.0, and the weight for optimization of the enthalpy will be calculated by XtalOpt as: "*1.0 - total weight of the objectives*".

It should be noted that: (i) the script should be placed in an accessible path where the local relaxations are performed, i.e., on the computer if XtalOpt runs the search on a local computer, or on the remote server if the energy optimization jobs are submitted to a remote queue, (ii) any objective that is explicitly assigned a weight of 0.0 will be calculated, but will not affect the optimization. Moreover, if the total weight explicitly assigned to objectives is 1.0, the enthalpy will have a weight of 0.0, i.e., the fitness will be determined only based on the objectives' value, and (iii) in case that the total weight of introduced objectives exceeds 1.0, XtalOpt will quit after producing an error message.

3. Multi-objective run in the XtalOpt CLI

In the MOES run in the CLI mode, for each user-defined objective a line should be added to the XtalOpt input file that starts with the keyword *objective*. This line, includes the above-mentioned information for the objective and, generally, has the following format:

```
objective = "optimization_type"
    "/path/script" "script_output_filename"
    "weight"
```

It should be noted that in the CLI mode of XtalOpt :

1. Possible options for the "**optimization_type**" are "minimization", "maximization", "hardness", and "filtration", as introduced above. This field is not case-sensitive, and only the first three letters are important in identifying the optimization type by XtalOpt (i.e., "min", "max", "har", and "fil").
2. Providing the "**script_output_filename**" is *optional*. If this is not specified, the default will be `objective#.out` in which "#" is the number of the objectives in the order that it appears in the XtalOpt input file (excluding the "hardness" objective), e.g., `objective1.out`, `objective2.out`, etc.
3. Specifying the "**weight**" for the objective is *optional*, as well. If this field is not given for a number of objectives, it will be determined via the following formula. If any weight is provided for any of the objectives, it will be subtracted from 1.0 and the remaining value will be divided between the enthalpy and objectives that don't have a specified weight.

3.1. Examples of XtalOpt *input file entries*

For a calculation that aims to minimize the volume per atom and maximize the synthesizability likelihood of the structures, the following lines can be added to the XtalOpt input file:

```
objective = min  /path/vol.sh  vol.dat  0.2
objective = max  /path/syn.sh  syn.dat  0.2
```

In this case, `vol.sh` and `syn.sh` are two executable scripts (either in a local or remote location, depending on the run) that use the `output.POSCAR` file to calculate the value of the volume per atom and synthesizability likelihood, respectively. XtalOpt expects the calculated values to be written by these scripts to the `vol.dat` and `syn.dat` files in the structure's directory. Since the weight for both objectives is 0.2, the remaining weight of 0.6 will be assigned to the enthalpy.

The following example illustrates the flexibility of the input entries:

```
objective = min  /path/vol.sh  vol.dat
objective = max  /path/syn.sh  0.2
```

Since the weight for the volume objective is not specified, the remaining total weight of 0.8 will be divided equally between enthalpy and volume per atom, and since the output filename for the synthesizability calculation is not given, XtalOpt will expect a file `objective2.out` (as this is the second objective in the list of objectives) to be present with the corresponding value.

4. Examples of user-defined scripts

In this section we provide a few examples of how the MOES can be carried out to minimize enthalpy while at the same time optimizing another quantity.

The space group number for a VASP format `output.POSCAR` structure file can be obtained using a simple Python script, e.g., `/path/spg.py`,

```
import ase.io as io
from ase import Atoms
from ase.spacegroup import get_spacegroup
s=io.read('output.POSCAR', index ='-1',
    format='vasp')
print(get_spacegroup(s, symprec=1e-3).no)
```

The output of this short Python code can be used via a simple executable bash script, e.g., `/path/spg.sh`,

```
#!/bin/sh
/path/python /path/spg.py > spg.dat
```

to produce a `spg.dat` file containing the space group number of the structure, which is readable by XtalOpt for this objective. In the XtalOpt input file for the CLI mode, this can be then introduced as:

```
objective = max  /path/spg.sh  spg.dat
```

along with the desired weight (or leaving the weight unspecified for XtalOpt to adjust it).

Alternatively, if the user desires to submit this calculation to a computational cluster, the executable script `/path/spg_queue.sh` in its most basic form can be written as:

```
#!/bin/sh
```

```
cat > fspg.slurm << EOF
#!/bin/sh
#SBATCH --nodes=1 --ntasks-per-node=1
#SBATCH --job-name=fspg
#SBATCH --output=fspg.out --error=fspg.err
#SBATCH --time=00:05:00
#SBATCH --cluster=slurm

##===== main task: calculating the objective
/path/python /paht/spg.py > spg.dat
##=====

EOF

sbatch fspg.slurm
```

This particular executable script, essentially, writes a job submission script for the slurm cluster to disk (`fspg.slurm` file, which includes everything between the lines containing the "`EOF`" keywords) and then submits this job (with "`sbatch fspg.slurm`") to the cluster. Just as a usual job submission script, the content of the job submission script (`fspg.slurm` file) includes an introductory part (job- and cluster-related settings) and the core task, basically the same as the above simple script for calculating the space group number, which is enclosed between "`##=====`" comment lines for clarity. Using this script, the optimization of the space group number will be performed similarly to the previous case, only, this time each calculation will be submitted to the cluster instead of running through a simple executable bash script.

5. Filtering the structures: Constrained search

The MOES in the XtalOpt code can facilitate a *constrained evolutionary search*. Besides maximizing or minimizing a particular objective, XtalOpt allows for filtering the relaxed structures based on some property. The result of such a constraint will prevent structures deemed unsuitable from entering the parent pool, hence promoting or prohibiting the propagation of a specific genetic characteristic.

To utilize the filtration functionality, similar to the "minimization" and "maximization" features, the user should provide a script that marks the structures to keep or discard based on the intended property. Structures that are marked for discarding will not be allowed in the parents' pool, although they will remain in the set of generated structures. The relevant entry in the XtalOpt input file for CLI mode can be:

```
objective = fil  /path/script  out_file
```

Compared to minimizing or maximizing an objective, in this case (a) regardless of the user-defined weight, the objective's weight will be set automatically to 0.0 by XtalOpt since this objective is not meant for optimization, and (b) the numerical value written to the output file by the script should be either 0 or 1. A 1 in the output file will instruct XtalOpt to keep the structure, while a 0 will instruct XtalOpt to discard it.

In the case of the filtration feature, and for a structure that is marked by the external script for discarding, XtalOpt provides the user with the option to replace the structure with a new one. For a run in the XtalOpt CLI, if the following flag is present in the input file:

```
objectivesReDo = true       # default is false
```

then XtalOpt will remove the structure from the parents' pool or replace it with a new one according to the user's instruction specified by the *jobFailAction* flag in the input file.

6. AFLOW-ML hardness

Previously, maximizing the AFLOW-ML hardness was invoked with the *calculateHardness* flag in the CLI version of the XtalOpt code (or, initiating relevant entries in the GUI). In the current version the Vickers hardness will be calculated by introducing the "hardness" objective. In the CLI mode, this can be utilized by adding a objective that has the "hardness" as the optimization type and, optionally, providing a corresponding weight, i.e.,

```
objective = hardness "weight"
```

Therefore, AFLOW-ML hardness calculations are now treated as a user-defined objective. It should be noted, however, that for a "hardness" objective: (1) the script name and the output file name inputs do not need to be provided, and (2) while an arbitrary number of objectives with "maximization", "minimization", and "filtration" type can be introduced, no more than one "hardness" objective can be added by the user. If there is more than one entry for hardness calculations in the XtalOpt input file, only the last one will be considered by the code (i.e., the weight for hardness calculation will be that of the last entry).

It should be noted that an objective of the "hardness" type performs the hardness optimization by obtaining the relevant data from AFLOW-ML through internal functions of the XtalOpt code. This is a legacy code and will be disabled in future releases of XtalOpt to avoid compatibility issues. Instead of using this type of objective, users are strongly encouraged to use a script to facilitate AFLOW-ML hardness optimization, similar to any regular "maximization" objective.

In the case that the user is interested in performing the hardness optimization as a "regular" objective, the required information can be obtained directly from the AFLOW-ML platform. An example of a script that can be used to retrieve this information is presented in the Appendix A.

7. Error handling and outputs

There might be a situation in which the external script (or the code that it is called by) fails to operate properly or fails to produce the output in a format that is readable by the XtalOpt code. In general, XtalOpt considers the output file to be correct and contain a valid value only if the "first entry" in the "first line" of the file is a numerical value. Otherwise, e.g., the file is not produced, is empty, or its first entry is not a legitimate numerical value, the calculation will be considered as failed by XtalOpt , and the structure will not be considered as a candidate to enter the pool.

During the XtalOpt evolutionary search, generally, the search settings including the MOES settings are written to the `xtalopt.state` and `xtaloptSettings.log` files, which can be used to verify the initialization of the search. In a run in the CLI mode, the `xtalopt-runtime-options.txt` file is also produced. It includes the set of search parameters that can be modified during the search. Among the MOES-related entries, however, only the *objectivesReDo* flag is outputted to this file and is allowed to be changed once the run is started.

For each structure, besides the corresponding output files generated by the scripts, a summary of the objective-related info (overall status of its calculations and their value) is written to the `structure.state` file.

Finally, just as with any run with the XtalOpt code, the live status of the overall process is available at any moment in the `results.txt` file, which summarizes the ranking of structures generated during the course of the evolutionary search. In the case of MOES runs, the status of a structure that has successfully finished local relaxation steps changes to "ObjectiveCalculation". Once calculating the objectives are finished, the status will change to "Optimized", "ObjectiveDismiss", or "ObjectiveFail" depending on whether the calculations finished successfully, the structure was discarded by a filtration objective, or the calculations failed, respectively. Moreover, in the `results.txt` file, for each objective introduced by the user, there will be an extra column in this file for the values of the corresponding objective.

8. Multi-objective runs in the XtalOpt GUI

XtalOpt GUI in this new release differs from the previous version of the code. In the **Search Settings** tab the AFLOW-ML harness related entries have been removed (green box in the Figure 2a), and a new **Multiobjective Search** tab is introduced (red outline).

In the **Multiobjective Search** tab, all entries relevant to a MOES run (described in Section 2.3) can be entered in the corresponding fields by (1) choosing the MOES run type from a drop-down menu, (2) and (3) entering the user-provided script and output file names, (4) entering or setting the weight, and (5) specifying whether a structure discarded by a filtration feature should be replaced with a new one or not. Then, the selected objective can be added to the list of objectives for the run (Figure 2b).

The aforementioned MOES run specifications in the CLI mode apply to the GUI case, as well. For instance:

- The types of MOES runs are "maximization", "minimization", "filtration", and "hardness"; and setting up an AFLOW-ML hardness calculation only requires its weight to be specified (Figure 2c),
- No more than one instance of "hardness" objective can be introduced in each run. If more than one instance of the "hardness" objective is present, only the corresponding weight (if entered differently) will be updated,
- Desired objectives can be arbitrarily added or removed before the run is started. However, once the search begins, the only MOES-related parameter that can be altered is the one instructing the code to handle the structures that are discarded by a "filtration" objective.

Further, in the MOES run in the XtalOpt GUI,

- The common errors in the input parameters (e.g., leaving script or file name fields empty, having space in text entries, total weight exceeding 1.0), result in an error message from the code (Figure 2d),

- In the **Progress** tab, the status of a structure that has successfully finished local relaxations changes to “Calculating objectives...”. After finalizing the calculations, the status will change to “Optimized”, “ObjectiveDismiss”, or “ObjectiveFail” according to the results of the calculations (Figure 2e),
- In the **Plot** tab, once a MOES run is started, the list of introduced objectives appears among the available options for the x and y axes of the trend plots, as well as the list of labeling symbols (Figure 2f).

9. Miscellaneous

In the new release of the XtalOpt code, a number of options are implemented to address special situations that the user might encounter. These options are briefly introduced in the following.

9.1. Scaled volume limits (CLI and GUI mode)

An important step in conducting evolutionary search is to come up with reasonable values for minimum and maximum limits of volume for the produced unit cells. The previous versions of XtalOpt provided two options of explicitly specifying these limits or introducing a fixed value for the generated unit cells’ volume. A new option is added to the current release of XtalOpt to aid in making a more educated guess for these values.

In the CLI mode, the user can optionally specify the pair of flags:

```
volumeScaleMin = ####

volumeScaleMax = ####
```

with the corresponding values being “real numbers greater than zero” (e.g., 0.8 and 1.2 for the minimum and maximum values, respectively).

If these flags are present with proper values, XtalOpt first calculates the total volume of spheres of van der Waals radius for all atoms in a formula unit. Then, it multiplies that total volume in the scaling factors to obtain the minimum and maximum limits of volume. One can check the final calculated values in the run output (i.e., the `xtalopt.state` or `xtalopt-runtime-options.txt` files).

The GUI mode includes this option in the **Structure Limits** tab, where the user can set the scaling factor and access a live update of calculated minimum and maximum volume limits while adjusting the scaling factors (Figure 3).

9.2. Running XtalOpt *locally on a cluster (CLI mode)*

Often when lengthy evolutionary searches are performed, the XtalOpt code is executed on the cluster where the jobs are being submitted (i.e., running XtalOpt locally while submitting the jobs to a queue). As the jobs are being submitted to the cluster, a remote queue interface should be specified in the XtalOpt input (e.g., slurm, pbs, etc.). This, however, requires a ssh connection to the cluster itself, which depending on the ssh configuration of the user’s account might not be allowed. For these type of XtalOpt runs, in the CLI mode, the user can add the following flag to the input file and run the code as usual:

```
localQueue = true      # default is false
```

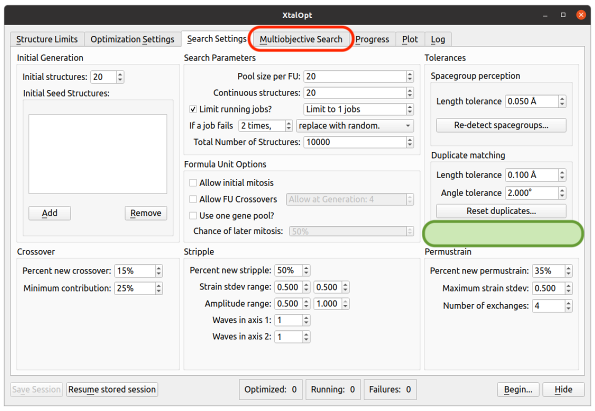
9.3. Termination of the XtalOpt *run (CLI mode)*

In a regular XtalOpt run, and once the maximum number of structures (specified by the user) is generated, the user can resume the run by increasing this maximum number (among other run-time flags that can be changed). This functionality requires the code to not quit automatically, and the user needs to terminate the application manually after the desired output is obtained. There are, however, situations that the user prefers the code to exit after producing a specified number of structures (e.g., in a systematic run for multiple input files). For a run in the CLI mode, the code can be instructed to exit after producing the specified number of structures by adding the following flag to the input file:

```
softExit = true       # default is false
```

or by setting its value to true in the run-time setting file `xtalopt-runtime-options.txt` during the run. With this flag set to true, the code quits after all running (and pending) jobs are finished and the output files are updated.

On the other hand, and at any moment during a run, the user can force the XtalOpt process (hence, the run) to quit immediately by adding the following line to the run-time settings, i.e., the `xtalopt-runtime-options.txt` file,



Initial Generation
Initial structures: 20
Initial Seed Structures:

Search Parameters
Pool size per FU: 20
Continuous structures: 20
✓ Limit running jobs? Limit to 1 jobs
If a job fails 2 times, replace with random.
Total Number of Structures: 10000

Formula Unit Options
Allow initial mitosis
Allow FU Crossovers Allow at Generation: 4
Use one gene pool?
Chance of later mitosis: 50%

Tolerances
Spacegroup perception
Length tolerance 0.050 Å
Duplicate matching
Length tolerance 0.100 Å
Angle tolerance 2.000°

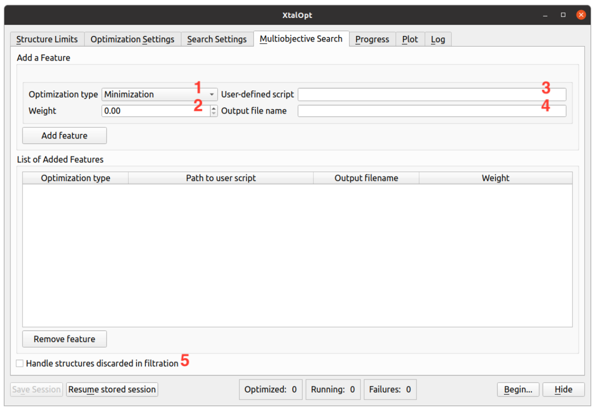
Crossover
Percent new crossover: 15%
Minimum contribution: 25%

Stripple
Percent new stripple: 50%
Strain stdev range: 0.500 0.500
Amplitude range: 0.500 1.000
Waves in axis 1: 1
Waves in axis 2: 1

Permustrain
Percent new permustrain: 35%
Maximum strain stdev: 0.500
Number of exchanges: 4

Optimized: 0 Running: 0 Failures: 0

(a) The XtalOpt GUI in the new version: "Search Settings" does not include AFLOW-ML hardness entries, and a new "Multiobjective Search" tab is added.



Add a Feature
Optimization type Minimization
Weight 0.00
User-defined script
Output file name

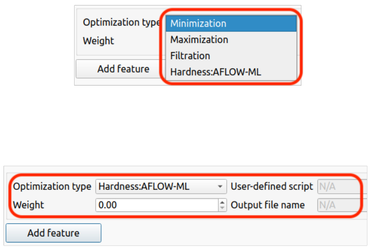
List of Added Features

Optimization type	Path to user script	Output filename	Weight

Handle structures discarded in filtration

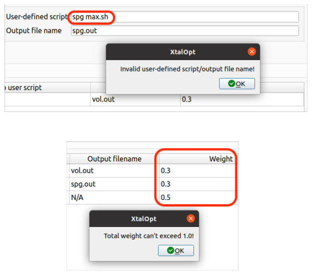
Optimized: 0 Running: 0 Failures: 0

(b) The "Multiobjective Search" tab where user can specify the search settings.



Optimization type Hardness:AFLOW-ML User-defined script N/A
Weight 0.00 Output file name N/A

(c) Multi-objective search types and relevant entries for the AFLOW-ML hardness calculations.



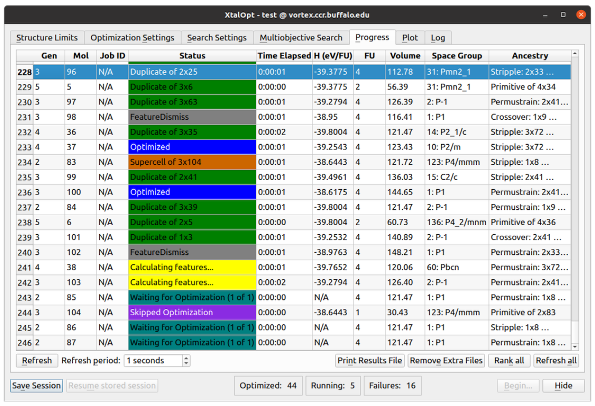
User-defined script spg max.sh
Output file name spg.out

Invalid user-defined script/output file name!

Output filename	Weight
vol.out	0.3
spg.out	0.3
N/A	0.5

Total weight can't exceed 1.0!

(d) Input errors for non-acceptable text entries and total weight of more than one.

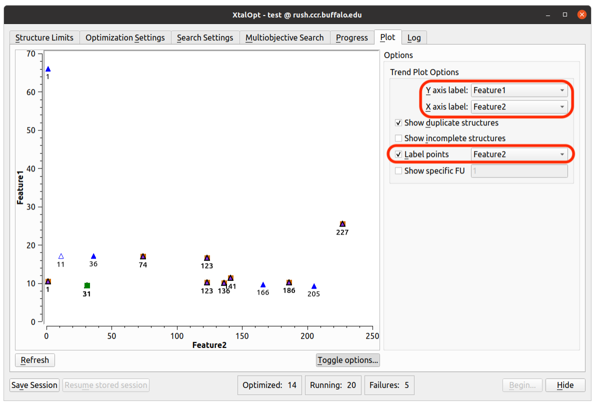


	Gen	Mol	Job ID	Status	Time Elapsed	H (eV/FU)	FU	Volume	Space Group	Ancestry
228	3	96	N/A	Duplicate of 2x25	0:00:01	-39.3775	4	112.78	31: Pmn2_1	Stripple: 2x33 ...
229	5	5	N/A	Duplicate of 3x6	0:00:00	-39.3775	2	56.39	31: Pmn2_1	Primitive of 4x34
230	3	97	N/A	Duplicate of 3x63	0:00:01	-39.2794	4	126.39	2: P-1	Permustrain: 2x41...
231	3	98	N/A	FeatureDismiss	0:00:01	-38.95	4	116.41	1: P1	Crossover: 1x9 ...
232	4	36	N/A	Duplicate of 3x35	0:00:02	-39.8004	4	121.47	14: P2_1/c	Stripple: 3x72 ...
233	4	37	N/A	Optimized	0:00:01	-39.2543	4	123.43	10: P2/m	Stripple: 3x72 ...
234	2	83	N/A	Supercell of 3x104	0:00:01	-38.6443	4	121.72	123: P4/mmm	Stripple: 1x8 ...
235	3	99	N/A	Duplicate of 2x41	0:00:01	-39.4961	4	136.03	15: C2/c	Stripple: 2x41 ...
236	3	100	N/A	Optimized	0:00:01	-38.6175	4	144.65	1: P1	Permustrain: 2x41...
237	2	84	N/A	Duplicate of 3x39	0:00:01	-39.8004	4	121.47	2: P-1	Permustrain: 1x9 ...
238	5	6	N/A	Duplicate of 2x5	0:00:00	-39.8004	2	60.73	136: P4_2/mnm	Primitive of 4x36
239	3	101	N/A	Duplicate of 1x3	0:00:01	-39.2532	4	140.89	2: P-1	Crossover: 2x41 ...
240	3	102	N/A	FeatureDismiss	0:00:01	-38.9763	4	148.21	1: P1	Permustrain: 2x33...
241	4	38	N/A	Calculating features...	0:00:01	-39.7652	4	120.06	60: Pbcn	Permustrain: 3x72...
242	3	103	N/A	Calculating features...	0:00:02	-39.2794	4	126.40	2: P-1	Permustrain: 2x41...
243	2	85	N/A	Waiting for Optimization (1 of 1)	0:00:00	N/A	4	121.47	1: P1	Permustrain: 1x8 ...
244	3	104	N/A	Skipped Optimization	0:00:00	-38.6443	1	30.43	123: P4/mmm	Primitive of 2x83
245	2	86	N/A	Waiting for Optimization (1 of 1)	0:00:00	N/A	4	121.47	1: P1	Stripple: 1x8 ...
246	2	87	N/A	Waiting for Optimization (1 of 1)	0:00:00	N/A	4	121.47	1: P1	Permustrain: 1x8 ...

Refresh period: 1 seconds

Optimized: 44 Running: 5 Failures: 16

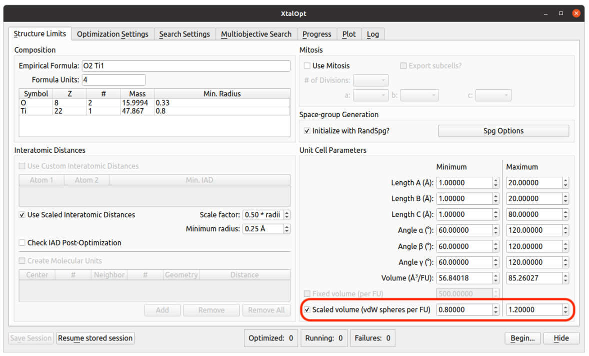
(e) The "Progress" tab with new MOES-related status.



Options
Trend Plot Options
Y axis label: Feature1
X axis label: Feature2
✓ Show duplicate structures
Show incomplete structures
✓ Label points Feature2
Show specific FU 1

Optimized: 14 Running: 20 Failures: 5

(f) The "Plot" tab in MOES run with user-defined objectives as labels and tags.


XtalOpt

Structure Limits | Optimization Settings | Search Settings | Multiobjective Search | Progress | Plot | Log

Composition

Empirical Formula: O2 Ti1

Formula Units: 4

Symbol	Z	#	Mass	Min. Radius
O	8	2	15.9994	0.33
Ti	22	1	47.867	0.8

Interatomic Distances

☐ Use Custom Interatomic Distances

Atom 1	Atom 2	Min. IAD

✓ Use Scaled Interatomic Distances — Scale factor: 0.50 * radii — Minimum radius: 0.25 Å

☐ Check IAD Post-Optimization

☐ Create Molecular Units

Center	#	Neighbor	#	Geometry	Distance

Add | Remove | Remove All

Mitosis

☐ Use Mitosis ☐ Export subcells?

# of Divisions:

a: b: c:

Space-group Generation

✓ Initialize with RandSpg? — Spg Options

Unit Cell Parameters

	Minimum	Maximum
Length A (Å):	1.00000	20.00000
Length B (Å):	1.00000	20.00000
Length C (Å):	1.00000	80.00000
Angle α (°):	60.00000	120.00000
Angle β (°):	60.00000	120.00000
Angle γ (°):	60.00000	120.00000
Volume (Å³/FU):	56.84018	85.26027
☐ Fixed volume (per FU)	500.00000	
✓ Scaled volume (vdW spheres per FU)	0.80000	1.20000

Save Session | Resume stored session | Optimized: 0 | Running: 0 | Failures: 0 | Begin... | Hide

Figure 3: Using the scaled volume option to specify the volume per formula unit limits in the new XtalOpt GUI

```
hardExit = true
```

It should be noted that (1) the *hardExit* flag terminates the XtalOpt running process regardless of any running or pending jobs and without updating the output files, and (2) this is only a run-time option and the presence of the flag in the input file is ignored by XtalOpt .

9.4. *Run-time log file and debug options (CLI and GUI modes)*

In the previous versions of XtalOpt , the code outputs a comprehensive list of messages regarding the progress of the run in the CLI mode, while only a subset of this information (i.e., key updates about the status of the run) are available in the **Log** tab for a run in the GUI mode. In the new release of the XtalOpt code, once the code is compiled with the

```
-DXTALOPT_DEBUG=ON
```

configuration option, a detailed list of output messages is saved to the log file `xtaloptDebug.log` in the local working directory for both the CLI and GUI modes.

Further, if the code is compiled with the

```
-DMOES_DEBUG=ON
```

configuration option, running MOES produces a set of output messages regarding the calculation of objectives and the generalized fitness function. These lines, starting with the keywork `NOTE`, contain information useful in monitoring the sanity of the calculations and MOES run.

9.5. *New options for the VASP optimizer (CLI and GUI modes)*

Recent versions of the VASP code allow for training and using ML interatomic potentials. The latest version of the XtalOpt code allows for using VASP ML models as an interatomic potential by automatically identifying the type of the calculations output file. Moreover, the user can provide only one POTCAR file for a multi-element system in the XtalOpt input file, instead of one per element which was required in the previous versions of XtalOpt . This can be done by introducing the POTCAR as "system" type, e.g.,

```
potcarFile system = /path_to/potcar
```

This last option is especially useful in interfacing XtalOpt with external codes (e.g., an arbitrary optimizer which is scripted to produce VASP format output files). It should be noted that (1) as XtalOpt arranges the chemical elements in alphabetical order, individual POTCAR files should be combined in the same order to prodcue the correct results, and (2) if a "system" POTCAR is introduced, other entries of *potcarFile* flag in the XtalOpt input file will be ignored by the code.

Appendix A. Retrieving AFLOW-ML data

The AFLOW-ML platform can be used to obtain electronic and vibrational properties [58, 59], as well as the superconducting critical temperature of a chemical compound [60]. The following script [61] can be used to obtain the AFLOW-ML data for a structure file input `POSCAR` with the VASP format.

```
#!/usr/bin/python3
import json, sys, os
from time import sleep
from urllib.parse import urlencode
from urllib.request import urlopen
from urllib.request import Request
from urllib.error import HTTPError
SERVER="http://aflow.org"
API="/API/aflow-ml"
MODEL="plmf"
poscar=open('POSCAR', 'r').read()
encoded_data = urlencode({'file':
    poscar,}).encode('utf-8')
url = SERVER + API + "/" + MODEL +
    "/prediction"
request_task = Request(url, encoded_data)
task = urlopen(request_task).read()
task_json = json.loads(task.decode('utf-8'))
```

```
results_endpoint =
    task_json["results_endpoint"]
results_url = SERVER + API + results_endpoint
incomplete = True
while incomplete:
  request_results = Request(results_url)
  results = urlopen(request_results).read()
  results_json = json.loads(results)
  if results_json["status"] == 'PENDING':
    sleep(10)
    continue
  elif results_json["status"] == 'STARTED':
    sleep(10)
    continue
  elif results_json["status"] == 'FAILURE':
    print("Error: prediction failure")
    incomplete = False
  elif results_json["status"] == 'SUCCESS':
    print("Successful prediction")
    print(results_json)
    incomplete = False
```

References

[1] S. M. Woodley, R. Catlow, Crystal structure prediction from first principles, Nature Materials 7 (12) (2008) 937–946. doi:10.1038/nmat2321.

[2] G. Hautier, A. Jain, S. P. Ong, From the computer to the laboratory: Materials discovery and design using first-principles calculations, Journal of Materials Science 47 (21) (2012) 7317–7340. doi:10.1007/s10853-012-6424-0.

[3] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, Nature Materials 12 (3) (2013) 191–201. doi:10.1038/nmat3568.

[4] A. R. Oganov, C. J. Pickard, Q. Zhu, R. J. Needs, Structure Prediction Drives Materials Discovery, Nature Reviews Materials 4 (5) (2019) 331–348. doi:https://doi.org/10.1038/s41578-019-0101-8.

[5] M. Born, R. Oppenheimer, Zur Quantentheorie der Molekeln, Annalen der Physik 389 (20) (1927) 457–484. doi:10.1002/andp.19273892002.

[6] C. J. Pickard, R. J. Needs, Structures at high pressure from random searching, physica status solidi (b) 246 (3) (2009) 536–540. doi:10.1002/pssb.200880546.

[7] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, Science 220 (4598) (1983) 671–680. doi:10.1126/science.220.4598.671.

[8] R. Martoňák, A. Laio, M. Parrinello, Predicting Crystal Structures: The Parrinello-Rahman Method Revisited, Physical Review Letters 90 (7) (2003) 075503. doi:10.1103/PhysRevLett.90.075503.

[9] S. Goedecker, Minima Hopping: An Efficient Search Method for the Global Minimum of the Potential Energy Surface of Complex Molecular Systems, J. Chem. Phys. 120 (21) (2004) 9911–9917. doi:10.1063/1.1724816.

[10] D. J. Wales, J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms, J. Phys. Chem. A 101 (28) (1997) 5111–5116. doi:10.1021/jp970984n.

[11] Y. Wang, J. Lv, L. Zhu, Y. Ma, Crystal structure prediction via particle-swarm optimization, Physical Review B 82 (9) (2010) 094116. doi:10.1103/PhysRevB.82.094116.

[12] A. R. Oganov, C. W. Glass, Crystal structure prediction using ab initio evolutionary techniques: Principles and applications, The Journal of Chemical Physics 124 (24) (2006) 244704. doi:10.1063/1.2210932.

[13] D. C. Lonie, E. Zurek, XtalOpt: An open-source evolutionary algorithm for crystal structure prediction, Computer Physics Communications 182 (2) (2011) 372–387. doi:10.1016/j.cpc.2010.07.048.

[14] S. Hajinazar, A. Thorn, E. D. Sandoval, S. Kharabadze, A. N. Kolmogorov, MAISE: Construction of neural network interatomic models and evolutionary structure optimization, Computer Physics Communications 259 (2021) 107679. doi:10.1016/j.cpc.2020.107679.

[15] E. Zurek, Discovering New Materials via A Priori Crystal Structure Prediction, in: A. L. Parrill, K. B. Lipkowitz (Eds.), Reviews in Computational Chemistry, Vol. 29, John Wiley \& Sons, Inc., Hoboken, New Jersey, 2016, pp. 274–326. doi:10.1002/9781119148739.ch5.

[16] A. N. Kolmogorov, S. Shah, E. R. Margine, A. F. Bialon, T. Hammerschmidt, R. Drautz, New Superconducting and Semiconducting Fe-B Compounds Predicted with an Ab Initio Evolutionary Search, Physical Review Letters 105 (21) (2010) 217003. doi:10.1103/PhysRevLett.105.217003.

[17] X. Luo, J. Yang, H. Liu, X. Wu, Y. Wang, Y. Ma, S.-H. Wei, X. Gong, H. Xiang, Predicting Two-Dimensional Boron–Carbon Compounds by the Global Optimization Method, Journal of the American Chemical Society 133 (40) (2011) 16285–16290. doi:10.1021/ja2072753.

[18] J. Zhang, Q. Zeng, A. R. Oganov, D. Dong, Y. Liu, High throughput exploration of ZrxSi1-xO2 dielectrics by evolutionary first-principles approaches, Physics Letters A 378 (47) (2014) 3549–3553. doi:10.1016/j.physleta.2014.09.019.

[19] C. Xie, Q. Zeng, A. R. Oganov, D. Dong, Discovering low-permittivity materials: Evolutionary search for MgAl2O4 polymorphs, Applied Physics Letters 105 (2) (2014) 1–5. doi:10.1063/1.4890464.

[20] A. N. Kolmogorov, S. Hajinazar, C. Angyal, V. L. Kuznetsov, A. P. Jephcoat, Synthesis of a predicted layered LiB via cold compression, Physical Review B 92 (14) (2015) 144110. doi:10.1103/PhysRevB.92.144110.

[21] M. Mayo, A. J. Morris, Structure Prediction of Li–Sn and Li–Sb Intermetallics for Lithium-Ion Batteries Anodes, Chemistry of Materials 29 (14) (2017) 5787–5795. doi:10.1021/acs.chemmater.6b04914.

[22] C. Y. Cheng, J. E. Campbell, G. M. Day, Evolutionary chemical space exploration for functional materials: computational organic semiconductor discovery, Chemical Science 11 (19) (2020) 4922–4933. doi:10.1039/D0SC00554A.

[23] J. A. Flores-Livas, Crystal structure prediction of magnetic materials, Journal of Physics: Condensed Matter 32 (29) (2020) 294002. doi:10.1088/1361-648X/ab7e54.

[24] L. T. Nguyen, G. Makov, High-Pressure Phases of SnO and PbO: A Density Functional Theory Combined with an Evolutionary Algorithm Approach, Materials 14 (21) (2021) 6552. doi:10.3390/ma14216552.

[25] N. Geng, T. Bi, E. Zurek, Structural Diversity and Superconductivity in S–P–H Ternary Hydrides under Pressure, The Journal of Physical Chemistry C 126 (16) (2022) 7208–7220. doi:10.1021/acs.jpcc.1c10976.

[26] L. T. Nguyen, G. Makov, GeS Phases from First-Principles: Structure Prediction, Optical Properties, and Phase Transitions

upon Compression, Crystal Growth & Design 22 (8) (2022) 4956–4969. doi:10.1021/acs.cgd.2c00497.

[27] B. Wang, K. P. Hilleke, S. Hajinazar, G. Frapper, E. Zurek, Structurally Constrained Evolutionary Algorithm for the Discovery and Design of Metastable Phases, Journal of Chemical Theory and Computation 19 (21) (2023) 7960–7971. doi:10.1021/acs.jctc.3c00594.

[28] K. P. Hilleke, X. Wang, D. Luo, N. Geng, B. Wang, F. Belli, E. Zurek, Structure, stability, and superconductivity of N-doped lutetium hydrides at kbar pressures, Physical Review B 108 (1) (2023) 014511. doi:10.1103/PhysRevB.108.014511.

[29] Y. Xu, J. M. Marrett, H. M. Titi, J. P. Darby, A. J. Morris, T. Friščić, M. Arhangelskis, Experimentally Validated Ab Initio Crystal Structure Prediction of Novel Metal–Organic Framework Materials, Journal of the American Chemical Society 145 (6) (2023) 3515–3525. doi:10.1021/jacs.2c12095.

[30] A. Franceschetti, A. Zunger, The inverse band-structure problem of finding an atomic configuration with given electronic properties, Nature 402 (6757) (1999) 60–63. doi:10.1038/46995.

[31] S. V. Dudiy, A. Zunger, Searching for Alloy Configurations with Target Physical Properties: Impurity Design via a Genetic Algorithm Inverse Band Structure Approach, Physical Review Letters 97 (4) (2006) 046401. doi:10.1103/PhysRevLett.97.046401.

[32] Q. Zhu, A. R. Oganov, M. A. Salvadó, P. Pertierra, A. O. Lyakhov, Denser than diamond: Ab initio search for superdense carbon allotropes, Physical Review B 83 (19) (2011) 193410. doi:10.1103/PhysRevB.83.193410.

[33] A. O. Lyakhov, A. R. Oganov, Evolutionary search for superhard materials: Methodology and applications to forms of carbon and TiO2, Physical Review B 84 (9) (2011) 092103. doi:10.1103/PhysRevB.84.092103.

[34] Q. Zeng, A. R. Oganov, A. O. Lyakhov, C. Xie, X. Zhang, J. Zhang, Q. Zhu, B. Wei, I. Grigorenko, L. Zhang, L. Cheng, Evolutionary search for new high- k dielectric materials: methodology and applications to hafnia-based oxides, Acta Crystallographica Section C Structural Chemistry 70 (2) (2014) 76–84. doi:10.1107/S2053229613027861.

[35] J. Qu, D. Zagaceta, W. Zhang, Q. Zhu, High dielectric ternary oxides from crystal structure prediction and high-throughput screening, Scientific Data 7 (1) (2020) 81. doi:10.1038/s41597-020-0418-6.

[36] E. J. Higgins, P. J. Hasnip, M. I. J. Probert, Simultaneous Prediction of the Magnetic and Crystal Structure of Materials Using a Genetic Algorithm, Crystals 9 (9) (2019) 439. doi:10.3390/cryst9090439.

[37] I. Giagkiozis, P. Fleming, Methods for multi-objective optimization: An analysis, Information Sciences 293 (2015) 338–350. doi:10.1016/j.ins.2014.08.071.

[38] N. Gunantara, A review of multi-objective optimization: Methods and its applications, Cogent Engineering 5 (1) (2018) 1502242. doi:10.1080/23311916.2018.1502242.

[39] M. T. M. Emmerich, A. H. Deutz, A tutorial on multiobjective optimization: fundamentals and evolutionary methods, Natural Computing 17 (3) (2018) 585–609. doi:10.1007/s11047-018-9685-y.

[40] J. Horn, N. Nafpliotis, D. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Vol. 1, IEEE, 1994, pp. 82–87. doi:10.1109/ICEC.1994.350037.

[41] N. Srinivas, K. Deb, Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation 2 (3) (1994) 221–248. doi:10.1162/evco.1994.2.3.221.

[42] K. Deb, Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction, in: Multi-objective Evolutionary Optimisation for Product Design and Manufacturing, Vol. 121, Springer London, London, 2011, pp. 3–34. doi:10.1007/978-0-85729-652-8_1.

[43] T. W. Liao, G. Li, Metaheuristic-based inverse design of materials – A survey, Journal of Materiomics 6 (2) (2020) 414–430. doi:10.1016/j.jmat.2020.02.011.

[44] A. Solomou, G. Zhao, S. Boluki, J. K. Joy, X. Qian, I. Karaman, R. Arróyave, D. C. Lagoudas, Multi-objective Bayesian materials discovery: Application on the discovery of precipitation strengthened NiTi shape memory alloys through micromechanical modeling, Materials and Design 160 (2018) 810–827. doi:10.1016/j.matdes.2018.10.014.

[45] A. M. Gopakumar, P. V. Balachandran, D. Xue, J. E. Gubernatis, T. Lookman, Multi-objective Optimization for Materials Discovery via Adaptive Design, Scientific Reports 8 (1) (2018) 3738. doi:10.1038/s41598-018-21936-3.

[46] D. Khatamsaz, B. Vela, P. Singh, D. D. Johnson, D. Allaire, R. Arróyave, Multi-objective materials bayesian optimization with active learning of design constraints: Design of ductile refractory multi-principal-element alloys, Acta Materialia 236 (2022). doi:10.1016/j.actamat.2022.118133.

[47] L. Yang, D. Robin, F. Sannibale, C. Steier, W. Wan, Global optimization of the magnetic lattice using genetic algorithms, Conf. Proc. C 0806233 (2008) THPC033.

[48] H.-Z. Chen, Y.-Y. Zhang, X. Gong, H. Xiang, Predicting New TiO 2 Phases with Low Band Gaps by a Multiobjective Global Optimization Approach, The Journal of Physical Chemistry C 118 (5) (2014) 2333–2337. doi:10.1021/jp411437f.

[49] Y.-Y. Zhang, W. Gao, S. Chen, H. Xiang, X.-G. Gong, Inverse design of materials by multi-objective differential evolution, Computational Materials Science 98 (2015) 51–55. doi:10.1016/j.commatsci.2014.10.054.

[50] J. J. Maldonis, Z. Xu, Z. Song, M. Yu, T. Mayeshiba, D. Morgan, P. M. Voyles, StructOpt: A modular materials structure optimization suite incorporating experimental data and simulated energies, Computational Materials Science 160 (2019) 1–8. doi:10.1016/j.commatsci.2018.12.052.

[51] J. Meng, M. Abbasi, Y. Dong, C. Carlos, X. Wang, J. Hwang, D. Morgan, Experimentally informed structure optimization of amorphous TiO 2 films grown by atomic layer deposition, Nanoscale 15 (2) (2023) 718–729. doi:10.1039/D2NR03614B.

[52] P. Avery, C. Toher, S. Curtarolo, E. Zurek, XtalOpt Version r12: An open-source evolutionary algorithm for crystal structure prediction, Computer Physics Communications 237 (2019) 274–275. doi:10.1016/j.cpc.2018.11.016.

[53] Z. Falls, P. Avery, X. Wang, K. P. Hilleke, E. Zurek, The XtalOpt Evolutionary Algorithm for Crystal Structure Prediction, The Journal of Physical Chemistry C 125 (3) (2021) 1601–1620. doi:10.1021/acs.jpcc.0c09531.

[54] P. Avery, X. Wang, C. Oses, E. Gossett, D. M. Proserpio, C. Toher, S. Curtarolo, E. Zurek, Predicting superhard materials via a machine learning informed evolutionary structure search, npj Computational Materials 5 (1) (2019) 89. doi:10.1038/s41524-019-0226-8.

[55] X. Wang, D. M. Proserpio, C. Oses, C. Toher, S. Curtarolo, E. Zurek, The Microscopic Diamond Anvil Cell: Stabilization of Superhard, Superconducting Carbon Allotropes at Ambient Pressure, Angewandte Chemie International Edition 61 (32) (2022). doi:10.1002/anie.202205129.

[56] G. Kresse, J. Hafner, Ab initio molecular dynamics for liquid metals, Physical Review B 47 (1) (1993) 558–561. doi:10.

1103/PhysRevB.47.558.

[57] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, Physical Review B 54 (16) (1996) 11169–11186. doi:10.1103/PhysRevB.54.11169.

[58] O. Isayev, C. Oses, C. Toher, E. Gossett, S. Curtarolo, A. Tropsha, Universal Fragment Descriptors for Predicting Properties of Inorganic Crystals, Nat. Commun. 8 (2017) 15679. doi:10.1038/ncomms15679.

[59] F. Legrain, J. Carrete, A. van Roekeghem, S. Curtarolo, N. Mingo, How Chemical Composition Alone Can Predict Vibrational Free Energies and Entropies of Solids, Chemistry of Materials 29 (15) (2017) 6220–6227. doi:10.1021/acs.chemmater.7b00789.

[60] V. Stanev, C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, I. Takeuchi, Machine learning modeling of superconducting critical temperature, npj Computational Materials 4 (1) (2018) 29. doi:10.1038/s41524-018-0085-8.

[61] E. Gossett, C. Toher, C. Oses, O. Isayev, F. Legrain, F. Rose, E. Zurek, J. Carrete, N. Mingo, A. Tropsha, S. Curtarolo, AFLOW-ML: A RESTful API for machine-learning predictions of materials properties, Computational Materials Science 152 (2018) 134–145. doi:10.1016/j.commatsci.2018.03.075.