Application of convex optimization to identification of atomic species from diffraction patterns

Ryoko Oishi-Tomiyasu[1,2,a)]
[1]Faculty of Science, Yamagata University, 1-4-12, Kojirakawa-cho, Yamagata-shi, Yamagata 990-8560, Japan
[2]JST PRESTO, 4-1-8, Honcho, Kawaguchi-shi, Saitama 332-0012, Japan

(Received 16 October 2016; accepted 30 November 2016)

Computational methods and a program to obtain crystal structures that have the perfectly identical diffraction patterns, i.e. structure factors with the same absolute values and the same lattice symmetry are discussed. This is directly related to the uniqueness of solutions in crystal structure determination of single-crystal/powder-crystal samples from diffraction data. In order to solve the problem, it is necessary to solve a system of quadratic equations. The framework of positive-semidefinite programming is used herein to solve the system efficiently. © *2017 International Centre for Diffraction Data.* [doi:10.1017/S0885715616000804]

Key words: phase retrieval, global optimization, positive-semidefinite programming, system of quadratic equations, multiple solutions

I. INTRODUCTION

How many different crystal structures can have the identical single (powder) diffraction pattern? When a crystal structure fits well to an observed pattern, how one can check whether or not there exist similarly good solutions? Although these problems have been pointed out [e.g., Engel (1988)], they have not been studied so rigorously. In powder indexing, although the problem to be solved is different, the problem about the uniqueness of solutions known as geometrical ambiguities was rigorously studied in Mighell and Santoro (1975). Recently, much progress was also made in the author's two papers (Oishi-Tomiyasu, 2014, 2016). In particular, the developed method can be used to check results of powder indexing.

For the same purpose, the uniqueness of solutions in crystal structure analysis from single-crystal diffraction patterns is investigated herein. Mathematically, crystal structures that have the same diffraction patterns as the input crystal structure are generated by: (a) enumerating all the sets of atomic coordinates with the identical set of difference vectors, and (b) determining the atomic species of every coordinates from the diffraction pattern of the original crystal structure. Both of (a) and (b) are not straightforward, considering combinatorial explosion that may occur in (a), and the necessity to solve a system of quadratic equations in (b).

Our new software succeeded in: (a) avoiding combinatorial explosion when medium-sized unit cells containing several tens atoms are input, and (b) solving the system of quadratic equations generally and efficiently by using the positive semidefinite programming (SDP) solver SDPA (Yamashita *et al.*, 2010). In this paper, the theory and some computational results are introduced.

By using the software, it is now possible to computationally answer the two questions posed in the first paragraph. The introduced results indicate that another plausible solution will not exist for ten crystal structures randomly chosen. We are now proceeding the exhaustive search for crystal structures that may have the same diffraction pattern with another crystal structure, by using crystal-structure database (see the acknowledgments for the information about the database, which currently contains more than 4000 CIF files). The results will be reported in the author's subsequent articles.

If two single crystals are judged to have the perfectly identical diffraction patterns (i.e., crystal lattice and set of structure factors), their powder samples also have the identical powder diffraction pattern. It should be noted that it is more complex mathematically to similarly check the uniqueness of solutions for powder crystal structure analysis. However, the frameworks introduced in this paper can be used for it.

II. PROBLEM TO BE SOLVED

The problem considered herein is how to generate crystal structures having structure factors with the perfectly identical absolute values as a given crystal structure. This is different from the problem normally solved in *ab initio* crystal structure analysis, in the sense that influence of observational errors can be largely ignored. This simplification drastically reduces the difficulty of the problem. Nevertheless, our problem is also useful for *ab initio* crystal structure analysis, because it can be used to check results, and understand the mathematics hidden in the phase-retrieval problems.

Throughout this article, the following ρ is used as a simple model of electron/nucleus distributions in a crystal:

$$\rho(x) = \sum_{l \in L} \sum_{i=1}^{n} C_i \delta(x - x_i - l), \tag{1}$$

where L is the lattice of a crystal, x_i $(1 \leq i \leq n)$ are positions of atoms in the unit cell. In the neutron case, C_i is the scattering factor of each atom. In the X-ray case, the scattering factors are normally approximated by the sum of exponential functions

[a)] Author to whom correspondence should be addressed. Electronic mail: ryoko_oishi@sci.kj.yamagata-u.ac.jp

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press

depending on the d-spacings:

$$C_i(d) = \sum_{k=1}^{4} a_{ik} \exp(-b_{ik}/2d) + c_i, \tag{2}$$

which is the Fourier transform of the sum of the modified Lorenzian function:

$$S_X(x) = (4/\pi) \sum_{k=1}^{4} a_{ik} b_{ik} (b_{ik}^2 + 16\ \pi^2\ |x|^2)^{-2} + c_i \delta(x),$$

where a_{ik} and b_{ik} are real numbers, and c_i can be a complex, if anomalous dispersion is considered. From the integral

$$\int_{\mathbb{R}^3} S_X(x)\,\mathrm{d}x = (8\ \pi^2)^{-1} \sum_{k=1}^{4} a_{ik} + c_i,$$

it is seen that in the X-ray case, C_i can be approximated by

$$(8\pi^2)^{-1} \sum_{k=1}^{4} a_{ik} + \mathrm{Real}\,(c_i),$$

which enables us to use formula (1), in common to the neutron case. The Patterson function, i.e. the Fourier transform of the diffraction pattern of $\rho(x)$ is given by:

$$\sum_{l \in L} \sum_{i=1}^{n} \sum_{j=1}^{n} C_i \overline{C_j} \delta\big(x - x_i + x_j - l\big), \tag{3}$$

where $\overline{C_j}$ is the complex conjugate of C_j, thus $\overline{C_j} = C_j$, since the scattering factors are now assumed to be real.

As a consequence, two crystal structures have the same diffraction pattern if and only if:

(i) their lattices are congruent in $\mathbb{R}^3$,
(ii) they have the identical set (denoted by Λ) of difference vectors $x_i - x_j$ $(1 \le i, j \le n)$, and
(iii) For each Δ in Λ, the values of $\sum_{x_i - x_j = \Delta} C_i \overline{C_j}$ (integrated intensity of the peak of the Patterson function corresponding to Δ) are the same.

In the following computation, information about chemical formula is also used.

What follows, for precise, the primitive cell of L (not the conventional cell depending on the centring type) is always used as the unit cell for simplicity. A finite set $\{x_1, \ldots, x_n\}$ provided by all the coordinates in the unit cell is called a *configuration*, in order to mean that a configuration is the crystal structure without information about atomic species. With regard to (ii), our judgment whether $x_i - x_j = x_k - x_l$ always allows a small difference in their values.

In this situation, the following two stages are required for the computation:

(1) Enumerate all the configurations with the set of difference vectors perfectly same as that of the input crystal structure. (Let $\Delta_0 = 0$, $\pm\Delta_1, \ldots, \pm\Delta_N$ be all the elements of the set.)
(2) For each configuration enumerated in (1), determine atomic species at every coordinate, by estimating the scattering factors. More concretely, the following system of equations is solved:

$$\begin{cases} \displaystyle\sum_{i=1}^{n} x_i = \sum_{j=1}^{m} C_i, \\ \displaystyle\sum_{v_i - v_j = \Delta_k} x_i x_j = \sum_{u_i - u_j = \Delta_k} C_i C_j \quad (0\ \le k\ \le N), \end{cases} \tag{4}$$

where C_j, u_j $(1 \le j \le m)$ are the scattering factors and the coordinates of the input crystal structure, and x_i, v_i $(1 \le i \le n)$ are those of the considered configuration. If the determined x_i is close to some C_j, then the atomic specie at u_j is also disposed at v_i.

It is straightforward to solve the system of Eq. (4), if the peaks of the Patterson function are not overlapping, i.e. all the $v_i - v_j$ $(1 \le i, j \le n)$ are distinct to each other. However, overlapping peaks seems to appear more frequently than expected, owing to the symmetry of the crystal structure (see Table I). In order to generally solve the system, the method provided by the Gröbner basis will be a candidate, if the coefficients of the equations are exact. If the errors contained in the positions of the input crystal structure and coefficients of scattering factor formula (2) are ignored, this may be considered to hold in our situation. However, the following formulation based on optimization is preferable, in order to fully consider the propagated errors contained in the atom positions determined from observational data. Therefore, the following minimization is adopted in this paper:

$$\begin{aligned} &\text{Minimize} \sum_{k=0}^{N} \Bigg| \sum_{v_i - v_j = \Delta_k} x_i x_j - \sum_{u_i - u_j = \Delta_k} c_i c_j \Bigg|, \\ &\text{subject to} \sum_{i=1}^{n} x_i = \sum_{j=1}^{m} c_j. \end{aligned} \tag{5}$$

In the next section, the above minimization will be solved by using the method called SDP relaxation. Minimization of $\sum_{k=0}^{N-1} \left| \sum_{v_i - v_j = \Delta_k} x_i x_j - \sum_{u_i - u_j = \Delta_k} c_i c_j \right|^2$ is not considered herein, because a polynomial of less degrees is preferable, in order to apply the SDP relaxation in the following. This is why the sum of absolute values is minimized in (5).

III. ALGORITHMS

A. Method to enumerate configurations with an identical set of difference vectors.

The following is the algorithm implemented in the new software:

Input: $\Lambda = \{\pm\Delta_1, \ldots, \pm\Delta_N\}$: set of all the difference vectors of a given crystal structure, not including the vector 0, and assumed to be sorted in the descending order of lengths.

Output: all the configurations in $\mathbb{R}^3/\mathbb{Z}^3$ that have Λ as the set of difference vectors.

Algorithm:

(i) Set the stage number $s = 1$, which is incremented, while the algorithm is carried out.
(ii) As the initial step of the stage $s = 1$, make the following configuration:

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press

(iii) At the stage $s = k$, for each node (denoted by v) taken from each configuration that has been constructed, make new configurations, by repeating the following procedures (a), (b):
   (a) form a configuration by adjoining v and an end point of Δ_k as in Figure 3,
   (b) form a configuration by connecting the v and an end point of Δ_k by some Δ_l with $k < l$ as in Figure 3.

The formed configurations are saved, as long as they are distinct from the others, and all of their difference vectors are in Λ. Proceed to the next stage $s = k + 1$, if it becomes impossible to add another Δ_k, without violating the rule that all difference vectors are in Λ.

A graph is called a *complete graph*, if every pair of distinct nodes is connected by a unique edge. If $C_1, \ldots, C_m$ are all the configurations that have Λ as the set of difference vectors, each C_i can be identified with a complete graph G with the edges assigned to one of $\pm\Delta_1, \ldots, \pm\Delta_N$. A complete subgraph $H_k \subset G$ is obtained, by connecting all the points in G that are endpoints of $\pm\Delta_1, \ldots, \pm\Delta_k$ in G.

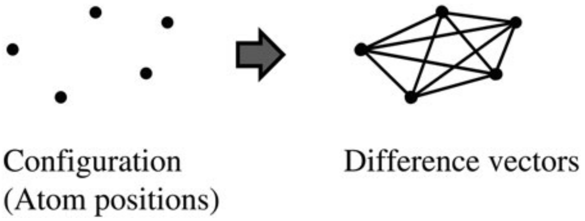


Figure 1. Difference vectors of a configuration of atoms.

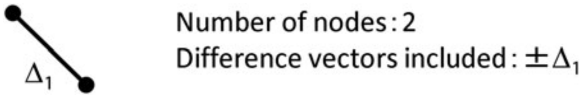


Figure 2. Configuration consisting of two atoms with the different vector Δ_1.

In short, in the stage k, the above algorithm aims to generate the H_k that is uniquely determined for each C_i. This is in order to avoid combinatorial explosion; distinct subgraphs of G_i should not be simultaneously saved in each stage. This use of H_k succeeded in suppressing the memory usage for medium-sized unit cells containing not more than several tens of atoms.

B. Method to solve quadratic optimization problem (QP)

What follows, the set of all the difference vectors of a given crystal structure $\Lambda = \{\Delta_0, \pm\Delta_1, \ldots, \pm\Delta_N\}$ is assumed to include $\Delta_0 = 0$. It is straightforward to verify that the following optimization problem is equivalent to (5):

$$\text{Minimize} \sum_{k=0}^{N} \varepsilon_k$$
$$\text{subject to} \begin{cases} \displaystyle\sum_{i=1}^{n} x_i = \sum_{j=1}^{m} c_j \\ \displaystyle -\varepsilon_k \leq \sum_{v_i - v_j = \Delta_k} x_i x_j - \sum_{u_i - u_j = \Delta_k} c_i c_j \leq \varepsilon_k \quad (1 \leq k \leq N). \end{cases} \tag{6}$$

If we put $\varepsilon_{1,k} = \varepsilon_k + \sum_{v_i - v_j = \Delta_k} x_i x_j - \sum_{u_i - u_j = \Delta_k} c_i c_j$ and $\varepsilon_{2,k} = \varepsilon_k - \sum_{v_i - v_j = \Delta_k} x_i x_j + \sum_{u_i - u_j = \Delta_k} c_i c_j$, the problem (6)

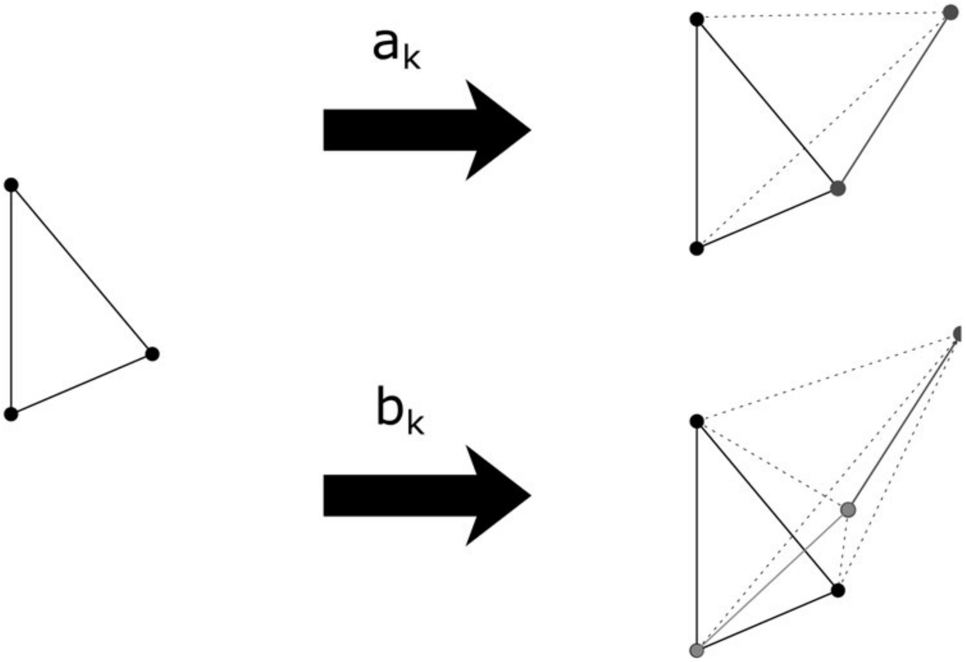


Figure 3. New configuration formed in the stage k.

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press

is also equivalent to:

$$\text{Minimize} \sum_{k=0}^{N} \frac{\varepsilon_{1,k}+\varepsilon_{2,k}}{2}$$

$$\text{Subject to} \begin{cases} \sum_{i=1}^{n} x_i = \sum_{j=1}^{m} c_j \\ \sum_{v_i - v_j = \Delta_k} x_i x_j - \sum_{u_i - u_j = \Delta_k} c_i c_j = \frac{\varepsilon_{1,k}-\varepsilon_{2,k}}{2} \\ \varepsilon_{1,k} \geq 0, \quad \varepsilon_{2,k} \geq 0 \quad (1 \leq k \leq N). \end{cases} \tag{7}$$

The problems (6) and (7) belong to the category of QPs, because the objective function and the constraints are polynomials of degree ≤2. The SDP relaxation is one of the standard methods to solve QP problems. According to the notation of SDP, the symbol • is used to represent the inner product of symmetric matrices $X = (X_{ij})$ and $Y = (Y_{ij})$ of the same size n:

$$X \cdot Y \underset{\text{def}}{=} \text{Trace}(XY) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} y_{ij} \tag{8}$$

What follows, $X \geq 0$ is used to mean that the symmetric matrices X is positive semidefinite.

An optimization problem is classified in the category of positive SDP, if it can be represented as either of the following forms:

(i) Minimize $C \cdot X$ subject to

$$\begin{cases} A_k \cdot X = b_k \quad (k = 1, \ldots, N) \\ X \geq 0. \end{cases} \tag{9}$$

(ii) Maximize $\sum_{k=1}^{N} b_k y_k$ subject to

$$\begin{cases} Y + \sum_{k=1}^{N} y_i A_i = C \\ Y \geq 0. \end{cases} \tag{10}$$

If the symmetric matrices A_k $(k = 1, \ldots, N)$, C and the vector $(b_1, \ldots, b_N)$ are common in (i), (ii), the problem (i) [*resp.* (ii)] is called the *primal problem*, when (ii) [*resp.* (i)] is called the *dual problem*. This is owing to the following facts known as the duality theorems:

Theorem 1 (weak duality theorem)

If some X, $(Y, y_1, \ldots, y_N)$ satisfies (9), (10), respectively, $C \cdot X \geq \sum_{k=1}^{N} b_k y_k$ always hold.

Theorem 2 (strong duality theorem)

Suppose that some X, $(Y, y_1, \ldots, y_N)$ satisfies (9), (10), respectively. In this case:

- If X is positive-definite, there exists an optimum solution $(Y^*, y_1^*, \ldots, y_N^*)$ of (ii), and the optimum value coincides with the lower limit of $C \cdot X$ under the constraint (9).
- If Y is positive-definite, there exists an optimum solution X^* of (i), and the optimum value coincides with the upper limit of $\sum_{k=1}^{N} b_k y_k$ under the constraint (10).
- If both X and Y are positive-definite, then there exist optimum solutions X^*, $(Y^*, y_1^*, \ldots, y_N^*)$, and $C \cdot X^* = \sum_{k=1}^{N} b_k y_k^*$ holds.

In general, positive SDP belongs to the category of convex optimization. It is known that interior point methods can efficiently find the global optimum solution.

According to the following steps, a SDP relaxation of (7) is obtained as a result:

1. By using the constraint $\sum_{i=1}^{n} x_i = \sum_{j=1}^{m} c_j$, remove the variable x_n from the problem.
2. Now the problem has the variables $x_1, \ldots, x_{N\text{-}1}$ and $\varepsilon_{1,k} \geq 0$, $\varepsilon_{2,k} \geq 0$ $(1 \leq k \leq N)$. The variable set can be identified with the following matrix X, by putting $x = (1, x_1, \ldots, x_{N\text{-}1})$ and $\tilde{X} = x^T x$ (x^T is the transposition of x):

$$X = \begin{pmatrix} \tilde{X} & & & & & & \\ & \varepsilon_{1,1} & & & & & \\ & & \ddots & & & & \\ & & & \varepsilon_{1,N} & & & \\ & & & & \varepsilon_{2,1} & & \\ & & & & & \ddots & \\ & & & & & & \varepsilon_{2,N} \end{pmatrix} \geq 0,$$

$$\tilde{X}_{11} = 1, \tilde{X} \text{ is rank } 1, \tag{11}$$

where $\tilde{X}_{ij}$ is the (i, j)-entry of $\tilde{X}$, and all the empty entries of X are assumed to be zero.

TABLE I. Result of Algorithm III.A; the input parameter ε used for judgment whether any vectors (v_1, v_2, v_3) in $\mathbb{R}^3/\mathbb{Z}^3$ equal 0 (i.e., if $\sum_{i=1}^{3} \left|\exp(2\pi\sqrt{-1}v_i) - 1\right|^2 < \varepsilon^2$) is set to 0.001.

No.	Materials (space group)	Number N of atoms in a cell (N ($N-1$)/2)	Number of difference vectors	Number of enumerated configuration
1	NaCl (F m -3 m)	2 (1)	1	1
2	BaFBr ($P4/n$ m m)	6 (15)	9	1
3	SrFCl ($P4/n$ m m)	6 (15)	9	1
4	$TaCl_4$ (C 1 2/m 1)	10 (45)	25	1
5	NiAs (C m c 2_1)	12 (66)	54	1
6	$Y(PO_4)$ (I $4_1/a$ m d)	12 (66)	34	3
7	CsI_3 (P m c n)	16 (120)	64	1
8	$BaCrO_4$ (P n m a)	24 (276)	141	1
9	Be_2 (SiO_4) (R -3)	42 (861)	441	1
10	$Co_2V_2O_7$ (P $2_1/c$)	44 (946)	484	1

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press

As a result, (7) is equivalent to the following problem:

$$\text{Minimize } C\cdot X$$

$$\text{subject to}\begin{cases} A_k\cdot X = b_k (k = 1, \ldots, N), \quad Z\cdot X = 1 \\ X \text{ is the block-diagonalized matrix : } \begin{pmatrix} \tilde{X} & & & & & & \\ & \varepsilon_{1,1} & & & & & \\ & & \ddots & & & & \\ & & & \varepsilon_{1,N} & & & \\ & & & & \varepsilon_{2,1} & & \\ & & & & & \ddots & \\ & & & & & & \varepsilon_{2,N} \end{pmatrix}, \\ X \geq 0,\ \tilde{X} \text{ is rank1}, \end{cases} \tag{12}$$

where C is the following symmetric matrix:

$$\begin{pmatrix} O & & & & & & \\ & 1/2 & & & & & \\ & & \ddots & & & & \\ & & & 1/2 & & & \\ & & & & 1/2 & & \\ & & & & & \ddots & \\ & & & & & & 1/2 \end{pmatrix}, \tag{13}$$

A_k, b_k are the symmetric matrix and the real number uniquely determined from the constraint:

$$\sum_{v_i - v_j = \Delta_k} x_i x_j - \sum_{u_i - u_j = \Delta_k} c_i c_j = \frac{\varepsilon_{1,k} - \varepsilon_{2,k}}{2}. \tag{14}$$

Z is the matrix with 1 in the (1, 1)-entry and 0 in the others; hence, $Z\cdot X = 1$ means $\tilde{X}_{11} = 1$.

If the rank-1 condition is removed from (12), the obtained problem is a SDP problem, because it is represented as in (9). Owing to the convexity, it is not necessary to worry about "local minimums" when the SDP problem is solved. If the objective functions of P and P^* converge to a value larger than zero, it is computationally proved that the original problem (5) has no solutions. Otherwise, if the objective function converges to a value close to 0, and X converges to a rank-1 matrix $\tilde{X}$, it is proved that the vector $\boldsymbol{x} = (1, x_1, \ldots, x_{N-1})$ satisfying $\tilde{X} = {}^T\boldsymbol{xx}$ is the "unique" solution of (5).

In general, one of the obstacles in utilizing the SDP relaxation is that the size n of the variables X, Y and the number N of constraints have a tendency to increase to a large number, although the numbers must be under practical limits. For example, SDPARA-C can handle the cases of $n < 20\,000$ and $N < 20\,000$. In the considered situation, n and N are normally <60 and <2000; hence, public SDPA solvers can efficiently solve the problem on a home PC. Another difficulty is that it is sometimes not straightforward to evaluate results, when the objective function converges to a value close to 0 and X converges to a matrix of rank >1. If influence of the errors is now assumed to be little, it is normally possible to obtain a finite set containing all the solutions from $\tilde{X}$ and A_k by using the Gröbner basis or some other methods.

IV. IMPLEMENTATION AND RESULTS

The methods described in III.A and III.B were implemented into programs with C++. The program for the latter calls SDPA (Yamashita *et al.*, 2010) as a solver of the SDP problem. Although the software was able to deal with only unit cells of primitive centring as of June, 2016 when EPDIC-15 was held, it can now handle all the centring types.

In addition to the NaCl sample (case of two atoms in a unit cell), the test data were randomly chosen from a database including more than 2000 CIF files. In Table I, results of Algorithm III.A are presented. It is seen that the Patterson functions have a number of overlapping peaks; otherwise, the value of $N\,(N-1)/2$ in the table equals the number of difference vectors.

On the computer with i7-6500 CPU@2.50 GHz, 8 GB memory, the computation times did not exceed 5 s for all the test data. Although memory allocation error did not happen for the unit cells in the tables, the memory usage becomes rather severe, as the number of atoms becomes more than several tens. Except for sample 6, the results indicate that the crystal structures do not have the same set of difference vectors as distinct configurations.

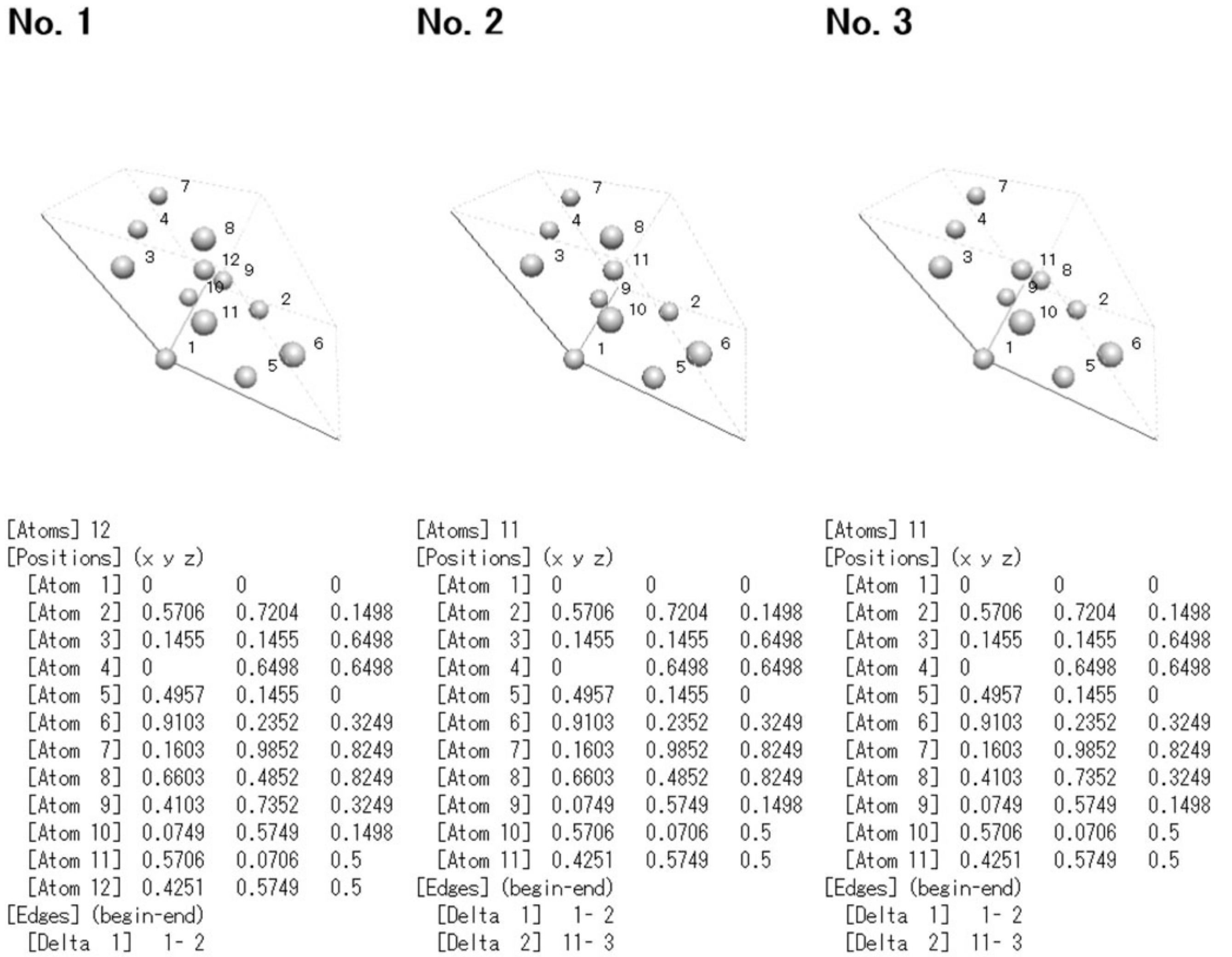
The program outputs the generated configurations in an html file, so that users can see the results with the 3D viewer of web browsers (Figure 4).

With regard to sample 6, newly obtained configurations are subsets of the original configuration, which indicates that their chemical formula must be different from the original one. Therefore, they will not be judged as good candidates experimentally.

Subsequently, Algorithm III.B was applied to the data 1–10 by using original crystal structures as the configuration to determine the atomic species. In this setting, the scattering factors should be the same as those of the original structure. In addition, with regard to the NaCl sample, at least two solutions should be found, because, if a good solution exists, another good solution is generated just by switching the two scattering factors x_1, x_2.

The results of our numerical test are presented in Table II. In all the cases, SDPA carried out its global search very quickly. Except for the NaCl sample, a rank-1 matrix was returned as the optimum solution $\tilde{X}$ in (11), and the scattering factors obtained as a result were very close to those of the atoms assigned in the original crystal structure, which

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press



```
[Atoms] 12
[Positions] (x y z)
  [Atom  1] 0       0       0
  [Atom  2] 0.5706  0.7204  0.1498
  [Atom  3] 0.1455  0.1455  0.6498
  [Atom  4] 0       0.6498  0.6498
  [Atom  5] 0.4957  0.1455  0
  [Atom  6] 0.9103  0.2352  0.3249
  [Atom  7] 0.1603  0.9852  0.8249
  [Atom  8] 0.6603  0.4852  0.8249
  [Atom  9] 0.4103  0.7352  0.3249
  [Atom 10] 0.0749  0.5749  0.1498
  [Atom 11] 0.5706  0.0706  0.5
  [Atom 12] 0.4251  0.5749  0.5
[Edges] (begin-end)
  [Delta  1]   1- 2
```

```
[Atoms] 11
[Positions] (x y z)
  [Atom  1] 0       0       0
  [Atom  2] 0.5706  0.7204  0.1498
  [Atom  3] 0.1455  0.1455  0.6498
  [Atom  4] 0       0.6498  0.6498
  [Atom  5] 0.4957  0.1455  0
  [Atom  6] 0.9103  0.2352  0.3249
  [Atom  7] 0.1603  0.9852  0.8249
  [Atom  8] 0.6603  0.4852  0.8249
  [Atom  9] 0.0749  0.5749  0.1498
  [Atom 10] 0.5706  0.0706  0.5
  [Atom 11] 0.4251  0.5749  0.5
[Edges] (begin-end)
  [Delta  1]   1- 2
  [Delta  2]  11- 3
```

```
[Atoms] 11
[Positions] (x y z)
  [Atom  1] 0       0       0
  [Atom  2] 0.5706  0.7204  0.1498
  [Atom  3] 0.1455  0.1455  0.6498
  [Atom  4] 0       0.6498  0.6498
  [Atom  5] 0.4957  0.1455  0
  [Atom  6] 0.9103  0.2352  0.3249
  [Atom  7] 0.1603  0.9852  0.8249
  [Atom  8] 0.4103  0.7352  0.3249
  [Atom  9] 0.0749  0.5749  0.1498
  [Atom 10] 0.5706  0.0706  0.5
  [Atom 11] 0.4251  0.5749  0.5
[Edges] (begin-end)
  [Delta  1]   1- 2
  [Delta  2]  11- 3
```

Figure 4. Results for sample 6 (the output displayed on a web browser); No. 1 is the configuration of the original crystal structure and confirmed to have the same set of difference vectors as its two subsets Nos 2 and 3.

TABLE II. Result of Algorithm III.B.

No.	Materials (space group)	Optimum value of primal problem (9) [Optimum value of dual problem (10)]	Rank of the optimum solution X
1	NaCl (*F m -3 m*)	5.11×10^{-9} (-6.64×10^{-9})	2
2	BaFBr (*P4/n m m*)	2.08×10^{-8} (-6.03×10^{-8})	1
3	SrFCl (*P4/n m m*)	2.72×10^{-9} (-7.91×10^{-9})	1
4	$TaCl_4$ (*C* 1 2/*m* 1)	5.15×10^{-9} (-1.91×10^{-8})	1
5	NiAs (*C m c* 2_1)	9.01×10^{-8} (-4.98×10^{-7})	1
6	$Y(PO_4)$ (*I* 4_1/*a m d*)	5.34×10^{-8} (-2.42×10^{-7})	1
7	CsI_3 (*P m c n*)	$8.03e \times 10^{-8}$ (-4.76×10^{-7})	1
8	$BaCrO_4$ (*P n m a*)	1.76×10^{-7} (-1.62×10^{-6})	1
9	Be_2 (SiO_4) (*R* -3)	6.95×10^{-7} (-8.04×10^{-6})	1
10	$Co_2V_2O_7$ ($P2_1/c$)	1.01×10^{-6} ($-1.18 \times 10-5$)	1

indicates that the diffraction patterns will not be the same, if the atomic species of the input crystal structures are replaced.

Although the solution was not uniquely determined for Sample 1, nothing more difficult than solving quadratic equations is necessary for this case. For further investigation, it will be also necessary to find crystal structures that provide cases of rank >1.

Overall, it may be concluded that our project to develop programs to check the uniqueness of solutions in crystal structure analysis from diffraction patterns, has gained satisfactory results and software. As applications of the methods developed herein, the following researches are planned:

- Exhaustive search for distinct crystal structures with the identical diffraction patterns, using crystal-structure database.
- Extension of the methods and the programs to the cases of powder diffraction.
- Implementation of a new method to handle the cases when the objective function of the SDP problem converges to 0, and the optimum solution a tilde should be on top of the $\tilde{X}$ as in equation 11 is rank >1.

V. CONCLUSION

Methods to check the uniqueness of the solution in single-crystal structure analysis was provided. The developed program works well for unit cells containing not more than several tens of atoms, and was able to provide a check that the crystal structure is the only possible solution. The theory

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press

of positive-semidefinite programming and the solver SDPA were used to solve the systems of quadratic equations required for the computation.

ACKNOWLEDGEMENTS

The author would like to thank Professor Kamiyama of High Energy Accelerator Research Organization for offering his private database to this software project. She would also like to extend her gratitude to Mr. Tamura and Mr. Kurosawa of Visible Information Inc. for their collaboration in implementing the software.

Engel, P. (**1988**). “Mathematical problems in modern crystallography,” Comput. Math. Applic. **16**, 425–436.

Mighell, A. D. and Santoro, A. (**1975**). “Geometrical ambiguities in the indexing of powder patterns,” J. Appl. Crystallogr. **8**, 372–374.

Oishi-Tomiyasu, R. (**2014**). “Method to generate all the geometrical ambiguities of powder indexing solutions,” J. Appl. Crystallogr. **47**, 2055–2059.

Oishi-Tomiyasu, R. (**2016**). “A table of geometrical ambiguities in powder indexing obtained by exhaustive search,” Acta Crystallogr. A: Found. Crystallogr. **72**, 73–80.

Yamashita, M., Fujisawa, K., Nakata, K., Nakata, M., Fukuda, M., Kobayashi, K., and Goto, K. (**2010**). *A High-performance Software Package for Semidefinite Programs: SDPA 7* (Research Report B-460). Dept. of Math. and Comp. Science, Tokyo Institute of Technology.

https://doi.org/10.1017/S0885715616000804 Published online by Cambridge University Press