Acta Crystallographica Section A
Foundations of Crystallography
ISSN 0108-7673

Received 30 January 2005
Accepted 23 May 2005

Polynomial-time algorithms for the integer minimal principle for centrosymmetric structures

Anastasia Vaia and Nikolaos V. Sahinidis*

Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Urbana, IL 61801, USA. Correspondence e-mail: nikos@uiuc.edu

The minimal principle for structure determination from single-crystal X-ray diffraction measurements has recently been formulated as an integer linear optimization model for the case of centrosymmetric structures. Solution of this model *via* established combinatorial branch-and-bound algorithms provides the true global minimum of the minimal principle while operating exclusively in reciprocal space. However, integer programming techniques may require an exponential number of iterations to exhaust the search space. In this paper, a new approach is developed to solve the integer minimal principle to global optimality without requiring the solution of an optimization problem. Instead, properties of the solution of the optimization problem, as observed in a large number of computational experiments, are exploited in order to reduce the optimization formulation to a system of linear equations in the number field of two elements ($\mathbf{F}_2$). Two specialized Gaussian elimination algorithms are then developed to solve this system of equations in polynomial time in the number of atoms. Computational results on a collection of 38 structures demonstrate that the proposed approach provides very fast and accurate solutions to the phase problem for centrosymmetric structures. This approach also provided much better crystallographic R values than *SHELXS* for all 38 structures tested.

1. Introduction

Direct methods have been used extensively for structure determination for over 50 years (Hauptman & Karle, 1953; Karle & Karle, 1966; Germain & Woolfson, 1968; Debaerdemaeker & Woolfson, 1983; Bricogne, 1984; Sheldrick, 1990; Giacovazzo, 1998; Hauptman *et al.*, 1999; Massa, 2000). Many of these methods rely on the minimal principle hypothesis for the phase problem, namely that a certain function of the phases is minimized only by the set of phases corresponding to the crystal structure (Debaerdemaeker & Woolfson, 1983).

Recently, Vaia & Sahinidis (2003) provided an integer programming formulation of the minimal principle for centrosymmetric structures. With appropriate choice of origin, the center of symmetry requires the phases to take a value 0 or π. This feature of the problem was exploited by Vaia & Sahinidis (2003) in order to avoid trigonometric terms in the original minimal principle formulation of Debaerdemaeker & Woolfson (1983). With the introduction of a suitable set of binary variables, the original nonlinear and nonconvex optimization problem was reduced to a linear integer programming problem. This integer minimal principle can be solved to global optimality *via* established combinatorial optimization techniques. It is well known that integer programs are, in general, NP-hard (Nemhauser & Wolsey, 1988), thus requiring exponential computing resources for solution in the worst case. Yet computational experience with the integer minimal principle model indicates that the gap between its optimal solution and its linear programming relaxation is zero. As this linear programming relaxation typically exhibits a fractional solution, Vaia & Sahinidis (2003) resorted to a branch-and-bound algorithm for the solution of the integer minimal principle.

In this paper, we develop an approach that solves the integer minimal principle without the use of a branch-and-bound integer programming algorithm. Our algorithm is polynomial in the number of atoms in the structure. In particular, under certain assumptions, we reduce the integer minimal principle model to a system of linear equations. Since this system involves only binary variables, this paper develops a division- and multiplication-free variant of the Gauss–Jordan elimination algorithm that utilizes only binary arithmetic. In addition, we develop a Gaussian elimination algorithm that involves a sparse matrix implementation combined with a pivot rule that reduces computer memory requirements. This results in fast algorithms for solving the phase problem for centrosymmetric structures.

We solve known structures from the literature with the developed polynomial-time algorithms and compare this approach to using state-of-the-art commercial integer programming optimization software to solve the integer minimal principle. These computations demonstrate that the

doi:10.1107/S010876730501648X

algorithms developed in this paper are more efficient in terms of solution time and computer memory requirements by several orders of magnitude. We also solve the same test structures with *SHELXS* and find that not all of them are solved by *SHELXS* unless a considerable amount of user intervention and CPU time are expended.

A very important feature of the approaches developed in this paper is that they do not require the solution of an optimization problem but rely exclusively on easily implementable linear algebra techniques. As a result, the proposed algorithms can be readily incorporated in crystallographic software.

2. Integer programming formulation of the minimal principle

Consider a single-crystal X-ray experiment that provides the normalized structure factor amplitudes, $|E_m|$, for $m = 1, \ldots, M$ reflections, each of which corresponds to a reciprocal-lattice vector $\mathbf{h}_m$ and phase ϕ_m. The triplet invariants ω_t are defined as (Hauptman & Karle, 1953)

$$\omega_t = \phi_{m_t} + \phi_{m'_t} + \phi_{m''_t} \quad t = 1, \ldots, T,$$

where $\mathbf{h}_{m_t} + \mathbf{h}_{m'_t} + \mathbf{h}_{m''_t} = \mathbf{0}$ for all $t = 1, \ldots, T$.

For centrosymmetric structures, since the triplet invariants ω_t obtain values from the set $\{0, \pi, 2\pi, 3\pi\}$, the cosine of the triplet invariants can only take values from the set $\{-1, 1\}$. Based on this observation, Vaia & Sahinidis (2003) proposed the following integer minimal principle formulation for solving the phase problem for centrosymmetric structures.

Indices

m index used for reflections ($m = 1, \ldots, M$).

t index used for triplet invariants ($t = 1, \ldots, T$).

Variables

ϕ_m phase of the mth reflection.

φ_m normalized phase of the mth reflection equal to ϕ_m/π.

ω_t triplet invariant defined by $\omega_t = \phi_{m_t} + \phi_{m'_t} + \phi_{m''_t}$, where $\mathbf{h}_{m_t} + \mathbf{h}_{m'_t} + \mathbf{h}_{m''_t} = \mathbf{0}$.

α_t binary decision variable.

β_t binary decision variable equal to $(1 - \cos\omega_t)/2$.

Parameters

M number of reflections.

n number of atoms in the unit cell.

T number of invariants.

$|E_m|$ structure-factor amplitude associated with reflection $\mathbf{h}_m$.

A_t constant equal to $2n^{-1/2}|E_{m_t}||E_{m'_t}||E_{m''_t}|$.

$\overline{\omega}_t$ conditional expected value of the cosine of the triplet invariant, equal to $I_1(A_t)/I_0(A_t)$ (Germain *et al.*, 1970).

Model M1

$$\begin{aligned} \min \quad & f(\boldsymbol{\beta}) = \frac{\sum_{t=1}^{T} A_t[4\beta_t\overline{\omega}_t + (1 + \overline{\omega}_t{}^2 - 2\overline{\omega}_t)]}{\sum_{t=1}^{T} A_t} \\ \text{s.t.} \quad & \varphi_{m_t} + \varphi_{m'_t} + \varphi_{m''_t} = 2\alpha_t + \beta_t, \quad t = 1, \ldots, T \qquad (1) \\ & \varphi_m \in \{0, 1\}, \quad m = 1, \ldots, M \\ & \alpha_t, \beta_t \in \{0, 1\}, \quad t = 1, \ldots, T. \end{aligned}$$

For each triplet invariant, the binary variables α_t and β_t on the right-hand side of (1) force the sum of the phases on the left-hand side of (1) to a value from the set $\{0, 1, 2, 3\}$. A zero value of β_t implies that the corresponding invariant ω_t equals 0 or 2π. A $\beta_t = 1$ implies that the corresponding ω_t equals π or 3π. As shown by Vaia & Sahinidis (2003), model M1 is equivalent to the original minimal principle formulation of Debaerdemaeker & Woolfson (1983) for the case of centrosymmetric structures.

Let N denote the number of atoms in the chemical formula. Computational results by Vaia & Sahinidis (2003) demonstrated that the ratio $N : M : T = 1 : 10 : 100$ balances computational time *versus* quality of the solution for model M1. This ratio will be used throughout this paper except for cases when a sufficient number of strong reflections or triplet invariants is unavailable (such cases will be clearly pointed out). In all cases, the model is constructed by selecting the M reflections with the largest E_m values. These reflections are subsequently used to generate $T = 10M$ triplet invariants.

The values of the phases are found by identifying a nontrivial, *i.e.* nonzero, solution point of M1. Model M1 is a constrained, linear, integer programming problem. In order to solve M1 to global optimality, Vaia & Sahinidis (2003) proposed a branch-and-bound combinatorial optimization algorithm. In each iteration of the algorithm, a linear relaxation of M1 is solved whereby many (initially all) binary variables are allowed to take values in the continuous interval [0, 1]. The search space is recursively partitioned into smaller elements by restricting phases to 0 or 1. Each partition element is bound below and further refined, when necessary. Binary solutions found in the process provide upper bounds for the optimal value of M1 and facilitate pruning those partition elements whose lower bounds exceed the current best upper bound. The algorithm terminates when all subsets are eliminated. A more detailed discussion on the issues concerning this type of combinatorial optimization algorithm, including bounding and partitioning, can be found in Nemhauser & Wolsey (1988).

The above integer programming approach was implemented by Vaia & Sahinidis (2003) using commercial integer programming software and may require a computational effort that grows exponential in the number of phases. Our objective in the sequel is to develop a fast solution procedure for solving M1 to global optimality *without* requiring an exponential algorithm and commercial software.

3. Reduction to a system of equations

Consider the continuous relaxation of M1, which is obtained by allowing each variable α_t, β_t and φ_{m_t} to take values in [0, 1] instead of $\{0, 1\}$. This system can be expected to have a minimum when $\beta_t = 0$ for all $t = 1, \ldots, T$. This is because $\overline{\omega}_t > 0$ and the system of equations in (1) is under-determined (it has T equations and $2T + M$ unknowns). Therefore, the integer programming problem has a minimum with $\beta_t = 0$ for all $t = 1, \ldots, T$ if the system of equations in (1) has an integer solution with $\beta_t = 0$, $t = 1, \ldots, T$. In this case, instead of solving the integer programming problem, we can alternatively solve the system of equations in (1) for the phases after fixing β_t to 0 for all $t = 1, \ldots, T$. Therefore, the system of linear equations to be solved is the following.

Model M2

$$\begin{aligned} \varphi_{m_t} + \varphi_{m'_t} + \varphi_{m''_t} &= 2\alpha_t \quad t = 1, \ldots, T \qquad (2)\\ \varphi_m &\in \{0, 1\} \quad m = 1, \ldots, M\\ \alpha_t &\in \{0, 1\} \quad t = 1, \ldots, T. \end{aligned}$$

This is a system of T equations in $T + M$ unknowns. Its linear relaxation always has a solution, although not necessarily integral. Since all the variables are binary, one can solve this system in the $\mathbf{F}_2$ field – this is the field of integers modulo 2 with elements from $\{0, 1\}$. In this field, odd numbers are equivalent to 1 and even numbers are equivalent to 0. Therefore, the right-hand side of (2) is equivalent to zero. Consequently, model M2 is equivalent to the following model in $\mathbf{F}_2$.

Model M3

$$\begin{aligned} \varphi_{m_t} + \varphi_{m'_t} + \varphi_{m''_t} &\equiv 0 \bmod 2 \quad t = 1, \ldots, T \qquad (3)\\ \varphi_m &\in \{0, 1\} \quad m = 1, \ldots, M. \end{aligned}$$

Therefore, all that is needed in order to solve M1 is to identify a nonzero element of the null space of the matrix determined by the left-hand side of (3) in $\mathbf{F}_2$.

4. Polynomial-time algorithms

Model M3 is a system of T equations in M unknowns. Since typically $T = 10M$, this homogeneous system of linear equations is over-determined. Commercial linear algebra packages can be used to find a nonzero solution of such systems. For instance, the *MATHEMATICA* function `NullSpace[`**M**`, Modulus->2]` provides the entire null space of a matrix **M** in $\mathbf{F}_2$. Below, two highly efficient specialized algorithms are proposed for finding a single nonzero element of this null space: a variant of the Gauss–Jordan elimination algorithm and a Gaussian elimination algorithm. The latter exploits the sparsity of the constraint matrix and uses the Markowitz rule (Markowitz, 1957) to select the pivot element.

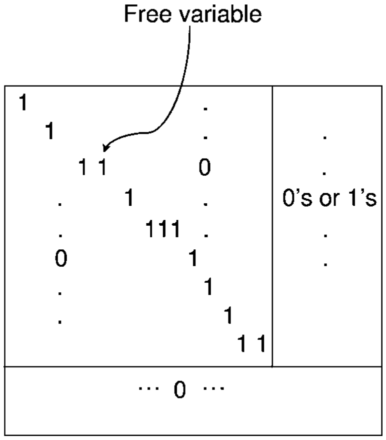

Figure 1
The constraint matrix after termination of the Gauss-Jordan elimination algorithm for a system with a nonzero solution.

4.1. Gauss–Jordan algorithm

In this variant of the Gauss–Jordan algorithm, all the elements above and below the diagonal are eliminated. We allow for row pivoting when the diagonal element is zero. If there is no nonzero element at or below the diagonal of a given column, then the phase corresponding to this column is a free variable. This means that we can assign any value (0 or 1) to this variable and still have a feasible solution to M3. If, after the algorithm is applied to all diagonal elements, all the entries of the constraint matrix from the $(M + 1)$th row and below are zero, then the system has a solution. If there is at least one free variable, then the system has a nonzero solution. To find the values of the phases, we assign arbitrarily a value 0 or 1 to each free variable and continue with back substitution. The free variables in M3 correspond to degrees of freedom in selecting different origins for the structure. Fig. 1 illustrates the constraint matrix at termination for a system of equations with a nonzero solution.

All variables are binary and the system is solved in the $\mathbf{F}_2$ field. This permits a very fast implementation of the algorithm because no multiplication or division is necessary. Indeed, consider the rth step of the algorithm, where the diagonal element is in the rth row and rth column and we must eliminate a nonzero element at the rth column and pth row. If we add the rth row to the pth row while noting that, in $\mathbf{F}_2$, even numbers are equivalent to zero, then we will have zeroed the element at the pth row and rth column without any multiplication or division. In fact, we can use logical operands to perform every calculation involved in this algorithm. This is clearly a very fast polynomial time algorithm for solving M1 to global optimality. In particular, the computational complexity of this algorithm is $O(M^2T)$.

4.2. Gaussian elimination algorithm with Markowitz score

The Gauss–Jordan algorithm described in the previous section does not take into account the sparsity of the constraint matrix in (3). Since there are at most three nonzero elements in each row of the matrix, a sparse matrix implementation would make a huge difference in terms of the memory required for solving the system. Therefore, a Gaussian algorithm is developed that employs the sparsity-preserving pivot selection rule of Markowitz (1957), combined with a sparse matrix implementation. At a given iteration of the Gaussian elimination algorithm, the Markowitz rule selects as pivot element the ijth element with the minimum value of $d_{ij} = (r_i - 1)(c_j - 1)$, where r_i and c_j denote the number of nonzeros in row i and column j, respectively, in the remaining part of the matrix.

The algorithm operates on a sparse matrix data structure describing (3). In particular, only the positions of the 1's are stored in terms of a 'column-major' format, whereby, for each column, the rows of the nonzero elements are stored in a single vector. A second vector provides an index to the first vector denoting the beginning positions of the nonzero elements of columns of (3). The following notation will be used to describe this algorithm.

Indices

ii index of the diagonal element under consideration.

$j_{\min}$ the column of the element with the smallest Markowitz score.

$i_{\min}$ the position in the row indices vector of the element with the smallest Markowitz score.

Parameters

ia integer vector containing the row indices of the nonzero elements in each column.

ja integer vector providing the next column start.

v integer vector providing the position of each variable.

c number of nonzero elements in the column from the *ii*th to the Tth row.

r number of nonzero elements in the row from the *ii*th to the Mth column.

d Markowitz score [equals $(r-1)(c-1)$].

$d_{\min}$ minimum Markowitz score.

Algorithm

- *Step 0 (Initialization)*. Set $ii \longleftarrow 1$. Set $v_k = k$ for $k = 1, \ldots, M$.
- *Step 1 (Pivot element selection)*. For each element in vector **ia** from the *ii*th to the Mth column and from the *ii*th to the Tth row, find c. If $c = 1$, this is the pivot element; update $i_{\min}$ and $j_{\min}$ and go to Step 2. Else, if $c \neq 1$, find r for this element. Set $d = (r-1)(c-1)$. If $d = 0$, this is the pivot element; update $i_{\min}, j_{\min}$ and go to Step 2. Else, if $d < d_{\min}$, set $d = d_{\min}$ and update $i_{\min}$ and $j_{\min}$. If no pivot element is found, go to Step 5.
- *Step 2 (Switch columns)*. If $j_{\min} \neq ii$, then switch the pivot column ($j_{\min}$) with the *ii*th column.
- *Step 3 (Switch rows)*. If $\mathbf{ia}_{i_{\min}} \neq ii$, then switch the pivot row $\mathbf{ia}_{i_{\min}}$ with the *ii*th row.
- *Step 4 (Gaussian elimination)*. For every column after the *ii*th column that includes the pivot element, combine the elements of that column with the elements of the *ii*th column. If an element is present in both columns, then it is not recorded. Update **ja**.
- *Step 5 (Back substitution)*. If no more pivot elements can be found, then set values arbitrarily to the free variables. Progressively calculate the values of the unknown variables from the values of the currently known variables. Use vector **v** to map the variables after pivoting to the original variables.

The computational complexity of this algorithm is $O(M^2T)$, *i.e.* the same as that of the Gauss–Jordan algorithm. Even though the sparse implementation is more involved, it reduces enormously the amount of computer memory required with respect to the Gauss–Jordan algorithm.

4.3. Connection with Sayre's equation

Only if (3) does not have a nontrivial solution would one need to solve the integer optimization problem in M1. In order for the system of equations in (3) to have a nonzero solution point, the number of phases with a nonzero value for each constraint should be either zero or exactly two. This means that, if there is a phase in the tth constraint with a value $\varphi_{m_t} = 1$, then there should also be another phase in the same constraint with a value $\varphi_{m'_t} = 1$ or $\varphi_{m''_t} = 1$ but not all three of these phases should have a value of 1. Sayre's equation (Sayre, 1952; Hughes, 1953) provides the average value of the product

$$\langle E_{\mathbf{h}_{m'}} E_{\mathbf{h}_{m''}} \rangle_{\mathbf{h}_{m'}} = 1/N^2 E_{\mathbf{h}_m}$$

for the reciprocal vectors $\mathbf{h}_m$, $\mathbf{h}_{m'}$ and $\mathbf{h}_{m''}$, which demonstrates that (3) has a solution at least for the phases corresponding to strong reflections.

4.4. Implementation

In our implementation, we begin by using the *LEVY* and *EVAL* programs (Blessing, 1989) to obtain the normalized structure-factor amplitudes $|E_m|, m = 1, \ldots, M$. Next, a global solution of the integer minimal principle M1 is obtained by solving M3 with the proposed polynomial algorithms. This step provides the values of the phases. Then, a modified version of the *CRUNCH* system (de Gelder *et al.*, 1993) is used to calculate an E map, perform the peak-picking procedure and calculate the atomic coordinates corresponding to the phases that solve M1.

5. Computational results

In this section, we first compare the proposed variant of the Gauss–Jordan algorithm with the optimization software *CPLEX7.0* (ILOG, 2000), which employs a branch-and-bound optimization strategy optimization. The objective of this comparison is to demonstrate the benefits of using the proposed algorithms. We do so by solving a large number of structures. Then, we present computational results on a collection of structures from the literature to demonstrate that the suggested algorithms are successful in solving difficult structures. Subsequently, using one of the proposed algorithms, we experiment with different data resolutions to identify the limits of the minimal principle with respect to data resolution. Finally, we compare our approach to the popular *SHELXS* phasing software. All runs reported below were performed on a 1.5 GHz Dell Xeon workstation with 1 GB memory.

5.1. Comparison with optimization software

In this section, we compare the computational time for the branch-and-bound mixed-integer programming (MIP) algorithm implemented in *CPLEX*, the Gauss–Jordan polynomial time algorithm (PA1) and the Gaussian elimination algorithm with the sparse matrix implementation (PA2). We solve the collection of 18 structures solved in Vaia & Sahinidis (2003), where details are provided about each structure and the source of the reflection data. In Table 1, we provide the number of atoms (N) in the chemical formula, reflections used (M), triplet invariants generated (T) and total number of variables in each model.

Table 1
Model dimensions with MIP and polynomial algorithms.

Chemical structure					Variables	
No.	Formula	N	M	T	MIP	PA1/PA2
1	$C_{50}H_{66}O_6 \cdot C_3H_7NO$	61	610	6100	12810	6100
2	$C_{30}H_{22}O_6S$	37	370	3700	7770	3700
3	$C_{30}H_{32}N_2O_6$	38	380	3800	7980	3800
4	$C_{44}H_{38}O_4$	48	480	4800	10080	4800
5	$C_{34}H_{42}B_2N_2O_4$	42	420	4200	8820	4200
6	$C_{34}H_26N_2O$	37	370	3700	7770	3700
7	$C_5H_{12}NO^+ \cdot$ $C_28H_{37}B_6O_{10}^- \cdot$ $0.5C_4H_{10}O$	55.5	542	5500	11550	5500
8	$3C_{40}H_{32}O_2 \cdot 4C_6H_6$	150	1378	13780	28938	13780
9	$C_{42}H_{56}N_2O_2$	46	460	4600	9660	4600
10	$C_{36}H_{62}$	36	360	3164	6688	3164
11	$C_{17}H_{19}N_3O_2$	22	220	2200	4620	2200
12	$C_{10}H_{19}ClO$	12	150	1500	3150	1500
13	$C_{18}H_{15}NO_3$	22	220	2200	4620	2200
14	$C_{13}H_{14}N_2O_3$	18	220	1600	3420	1600
15	$C_{41}H_{78}O_{11}Si_8$	60	590	6000	12590	6000
16	$C_{44}H_{52}N_4 \cdot C_2H_6O$	51	510	5100	10710	5100
17	$C_{12}H_{10}O_3$	15	200	2000	4200	2000
18	$C_{24}H_{12}N_6 \cdot 4CHCl_3$	52	520	5200	10920	5200

Table 2
Objective function value and CPU times with MIP, PA1, and PA2.

		CPU time (s)		
Structure	f	MIP	PA1	PA2
1	0.0625	196	3	4
2	0.0351	83	3	3
3	0.0862	89	2	1
4	0.1347	254	3	2
5	0.0965	127	2	1
6	0.0586	112	3	2
7	0.0316	238	4	3
8	0.3594	28	25	6
9	0.0872	130	3	2
10	0.0193	74	6	6
11	0.0053	63	2	2
12	0.0016	10	1	1
13	0.0044	10	2	2
14	0.0243	16	1	1
15	0.0542	601	6	4
16	0.1258	261	5	3
17	0.0041	6	1	3
18	0.0351	134	3	2
Average		135	4	3
Standard deviation		143	5	2

In Table 2, we report the global optimal value (f) of the minimal principle, along with the running time in s for MIP, PA1 and PA2. The branch-and-bound algorithm as well as our PA1 and PA2 algorithms provided exactly the same global minimum values (f) of the minimal principle model. As shown in Table 2, the PA1 and PA2 algorithms are, respectively, an average of 34 and 45 times faster than the MIP. The running time for PA1 averages 4 s. Even for the largest structure (structure 8), the running time of PA1 does not exceed 25 s. Similarly, the running time for PA2 averages 3 s. PA2 is faster than PA1 since it minimizes fill-in and the number of pivots required for solution. For MIP, there are cases in which smaller structures require larger running times than larger structures. This is because the running time for the MIP depends not only on the model dimensions but also on the selection of the origin. On the other hand, the CPU time for both polynomial time algorithms is independent of the size of the structure as illustrated in Fig. 2. Note also that the integer optimization model M1 involves $2T$ more variables than the set of equations M3, which results in much higher computer memory requirements for the MIP.

5.2. Application to an additional collection of structures

We have solved 20 additional structures by applying the PA1 and PA2 algorithms. For each of these structures, Table 3 provides the number of atoms (N) in the chemical formula, the space group, the number of molecules in the unit cell (Z) and the source of the data. The first 13 structures of this table are structures reported in *Acta Cryst.* Sections C and E. The sizes of these structures range from 41 to 119 atoms. The last seven structures of Table 3 were obtained from various sources and are considered as difficult structures either because of low-resolution data or because of the presence of a highly disordered solvent. The sizes of these structures range from 42 to 117 atoms.

Table 4 provides the number of phases (M) and invariants (T) used in the system of equations M3. Even for the larger structures, the number of variables in M3 is relatively small. For instance, for structure 33, which has 110 atoms, model M3 has $M = 1100$ variables and $T = 11\,000$ equations, while model M1 has $2T + M = 22\,110$ variables and $T = 11\,000$ equations. Table 4 also presents the minimal principle value f, the crystallographic coefficient R and the CPU s for each structure. Even though there are thousands of equations (exactly T) in the model, the running time for PA1 ($t_{\rm PA1}$) and PA2 ($t_{\rm PA2}$) averages only 12 and 5 s, respectively. PA2 is faster because it exploits sparsity of the constraint matrix in a way that minimizes fill-in and subsequent Gaussian elimination iterations.

The polynomial algorithms were successful in determining all 20 structures. In all cases, they resulted in the same objective function value as the linear programming relaxation of M1. Our computational experience with all 38 structures solved in this paper shows that, in all cases, there is a nonzero solution point to M3 and, thus, application of the specialized algorithms to the minimal principle model reveals the crystal

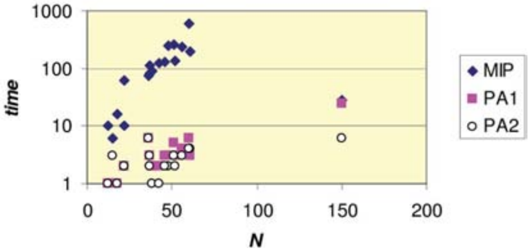

Figure 2
Computational time (s) as a function of number of atoms N.

Table 3
Additional test problems.

Chemical structure					
No.	Formula	*N*	*Z*	Space group	Reference
19	$4C_{19}H_{23}N_5O_3 \cdot 2C_2H_6OS \cdot 3H_2O$	119	2	$P\bar{1}$	(*a*)
20	$C_{33}H_{59}N_{11}{}^{8+} \cdot 8ClO_4{}^{-} \cdot 5H_2O$	89	2	$P\bar{1}$	(*b*)
21	$C_{54}H_{72}O_{10} \cdot CH_4O$	66	4	$P2_1/c$	(*c*)
22	$C_{51}H_{59}NO_{11}S$	64	4	$P2_1/c$	(*d*)
23	$C_{34}H_{32}N_8O_4S_2 \cdot 3C_5H_5N$	68	1	$P\bar{1}$	(*e*)
24	$C_{39}H_{47}O_4 \cdot C_6H_{16}N^+ \cdot C_2H_3N$	53	4	$P2_1/c$	(*f*)
25	$C_{24}H_{42}N_6{}^{4+} \cdot C_{10}H_2O_8{}^{4-} \cdot 6H_2O$	54	1	$P\bar{1}$	(*g*)
26	$C_{16}H_{38}N_4{}^{2+} \cdot 2C_8H_4NO_6{}^{-}$	50	1	$P\bar{1}$	(*h*)
27	$3C_{10}H_8N_2 \cdot 2C_6H_4O_6$	60	1	$P\bar{1}$	(*i*)
28	$C_{48}H_{42}$	48	1	$P\bar{1}$	(*j*)
29	$C_{44}H_{38}O_2$	46	2	$P2_1/c$	(*k*)
30	$C_{47}H_{32}$	47	2	$P\bar{1}$	(*l*)
31	$C_{21}H_{19}N_3O_7 \cdot 2C_3H_7NO$	41	2	$P\bar{1}$	(*m*)
32	$C_{109}H_{73}N \cdot CHCl_3$	114	4	$P2_1/c$	(*n*)
33	$C_{110}H_{74}$	110	4	$P\bar{1}$	(*o*)
34	$C_{83}H_{82}N_8O_{20}Cl_6$	117	2	$P\bar{1}$	(*p*)
35	$C_{84}H_{70}N_{10}O_7$	101	4	$P2_1/n$	(*p*)
36	$C_{51}H_{44}N_5O_8Cl_3$	67	2	$P\bar{1}$	(*p*)
37	$C_{29}H_{35}N_4O_6Cl_3$	42	2	$P\bar{1}$	(*p*)
38	$C_{36}H_{48}O_9$	44	4	$P2_1/c$	(*p*)

References: (*a*) Hempel *et al.* (2000); (*b*) McKee & Morgan (2003); (*c*) Thuéry *et al.* (2000); (*d*) Kim *et al.* (2000); (*e*) Chantrapromma *et al.* (2001); (*f*) Leverd *et al.* (2000); (*g*) Zhu *et al.* (2002); (*h*) MacLean *et al.* (2002); (*i*) Cowan *et al.* (2001); (*j*) Frampton *et al.* (2000); (*k*) Robinson *et al.* (1999); (*l*) Perera *et al.* (2003); (*m*) Zou *et al.* (2003); (*n*) de Graaff (2003); (*o*) Ho (2003); (*p*) de Gelder (2003).

Table 4
Model dimensions and computational results.

Structure	*N*	*M*	*T*	*f*	*R*	$t_{\rm PA1}$	$t_{\rm PA2}$
19	119	1190	11900	0.1057	0.12	45	12
20	89	890	8900	0.1194	0.14	13	8
21	66	660	6600	0.0547	0.07	7	4
22	64	640	6400	0.1163	0.15	9	3
23	68	680	6800	0.0669	0.15	10	10
24	53	530	5300	0.0356	0.05	7	3
25	54	540	5400	0.1776	0.04	3	2
26	50	500	5000	0.0568	0.08	4	3
27	60	600	6000	0.1546	0.03	4	3
28	48	480	4800	0.0837	0.03	3	2
29	46	460	4600	0.1362	0.13	2	1
30	47	470	4700	0.054	0.06	3	2
31	41	410	4100	0.062	0.14	2	1
32	114	1100	11000	0.202	0.17	24	5
33	110	1100	11000	0.0853	0.13	30	10
34	117	1170	11700	0.0534	0.07	31	9
35	101	1010	10100	0.1173	0.25	19	5
36	67	670	6700	0.1746	0.15	7	3
37	42	420	4200	0.1137	0.19	5	2
38	45	450	4500	0.0623	0.22	4	2
Average						12	5
Standard deviation						12	3

structure without the need to invoke a branch-and-bound algorithm to solve M1.

5.3. Sensitivity to data resolution

For eight of the structures in Vaia & Sahinidis (2003) (structures 1–8 of Table 1) with data resolution ranging from 0.76 to 0.842 Å, respectively, we have truncated the reflection data to progressively lower their resolution to 0.9, 1.0, 1.1, 1.2 and 1.3 Å. Table 5 provides the *R* values based on the normalized structure-factor amplitudes, and the fraction (*k*) of the total number of independent atoms correctly identified for these structures at the five different resolution values.

Since our algorithm provides the global solution to the minimal principle model and there is no false minima issue, we are able to identify *unambiguously* the resolution where the minimal principle, in conjunction with the *CRUNCH* peak-picking procedures, fails to separate atoms in the electron-density map. In most of the cases, we were able to identify correctly all the atoms in the asymmetric unit cell up to 1.2 Å resolution. At resolution 1.3 Å, despite the fact that there was an insufficient number of strong reflections and, therefore, we had to rely on data with small *E* values, a significant fraction of the structure was often identified. Yet these computational experiments demonstrate that the minimal principle, in conjunction with the *CRUNCH* peak-picking procedures, fails at resolutions lower than approximately 1.2 to 1.3 Å.

5.4. Comparison to *SHELXS*

All 38 structures of Tables 1 and 3 were also tried with *SHELXS* (Sheldrick, 1990). The results are reported in Table 6 in comparison with the results using PA2. As seen in this table, PA2 takes an average of about 4 s over this collection of compounds and so does *SHELXS*. In addition to the time spent on PA2, our approach requires a relatively small amount of time to be spent on the *LEVY*, *EVAL* and peak-searching operations. As seen in Table 6, PA2 provides much better crystallographic *R* values than *SHELXS* for all these compounds. Furthermore, there are many cases for which the crystallographic *R* values indicate that *SHELXS* was unable to solve these structures. Refinement will most likely improve the results from the application of *SHELXS* but no refinement was used by PA2 either.

In all computations reported in Table 6, we used *SHELXS* with default algorithmic options. A trained user could certainly improve the performance of *SHELXS via* the selection of different options. For instance, after solving structure 13 with default *SHELXS* options, we set the values of the *SHELXS* options `ns`, `nE` and `np` to two, two and four times, respectively, the values used by *SHELXS* in the default run. Doing so resulted in a much improved crystallographic *R* value of 0.22 and a considerable increase of CPU time to 18 s. According to Sheldrick (2005), only six structures (2, 32, 33, 35, 36 and 37) cannot be solved using *SHELXS*, at least without time-consuming expert intervention, and all structures can be solved with *SHELXD* but take at least ten times as long as PA2.

Our interpretation of the results of these computational experiments is that the higher CPU times required by the *SHELX* codes are because these codes use stochastic global optimization algorithms that do not exploit the mathematical structure of the phase problem to the extent that deterministic global optimization algorithms do. On the other hand, our approach solves the integer minimal principle model to global optimality *via* deterministic algorithms that do not require

Table 5
R values and fraction of independent atoms identified (k) at different resolutions.

	Resolution									
	0.9 Å		1.0 Å		1.1 Å		1.2 Å		1.3 Å	
Structure	R	k	R	k	R	k	R	k	R	k
1	0.07	61/61	0.07	61/61	0.07	61/61	0.07	61/61	0.18	60/61
2	0.14	37/37	0.14	37/37	0.14	37/37	0.14	36/37	0.21	24/37
3	0.04	19/19	0.06	19/19	0.06	19/19	0.06	19/19	0.25	16/19
4	0.04	24/24	0.04	24/24	0.04	24/24	0.04	23/24	0.18	9/24
5	0.06	42/42	0.06	42/42	0.07	42/42	0.07	40/42	0.18	18/42
6	0.08	37/37	0.08	37/37	0.08	37/37	0.38	18/37	0.37	7/37
7	0.08	55/55	0.11	55/55	0.14	54/55	0.14	32/55	0.15	17/55
8	0.11	73/75	0.11	73/75	0.10	73/75	0.11	73/75	0.19	11/75

user intervention or multiple trials from different starting points.

6. Conclusions

In this paper, we propose a new formulation of the integer minimal principle previously developed for centrosymmetric structures and we develop two specialized algorithms. These algorithms not only take into account the special mathematical structure of the minimal principle model but also go further and, in accordance with Sayre's equation, exploit the theoretically expected characteristics of the phases themselves.

The solution of the minimal principle optimization problem reduces to the solution of a system of linear equations that involves a smaller number of variables than the integer minimal principle. In order to solve this system of equations, we developed two algorithms. The first algorithm is a variant of the Gauss–Jordan algorithm and uses binary arithmetic to reduce computational time. This algorithm provides fast and accurate results for all 38 structures to which it was applied. This approach reduces the running time by an average of 34 times in comparison to the integer programming approach. Most importantly, the computation time depends only on the size of the structure and no longer depends on the selection of origin. In particular, a global optimum of the minimal principle model is obtained in $O(M^2T)$ time. We also propose a Gaussian elimination algorithm that combines a sparse matrix implementation with the Markowitz pivot selection rule that preserves sparsity. Compared to the Gauss–Jordan variant, the sparse implementation is somewhat faster in practice because it minimizes fill-in. More importantly, it reduces drastically the computer memory requirements.

When applied to truncated reflection data for eight structures from the literature, the binary Gauss–Jordan polynomial-time algorithm, in conjunction with the *CRUNCH* peak-picking procedure, was able to solve structures for resolutions of up to 1.3 Å. Our algorithms also provided much better crystallographic R values than *SHELXS* on all 38 structures that we tested.

In crystallographic computing practice with small molecules, a ratio of $N : M : T = 1 : 10 : 100$ is commonly used. Then, our $O(M^2T)$ algorithms require $O(N^3)$ operations. Even if M is increased in proportion to N^2, the proposed algorithms are still polynomial in N and require $O(N^6)$ operations.

Table 6
CPU s and crystallographic R values for PA2 and *SHELXS*.

	CPU s		Crystallographic R	
Structure	PA2	*SHELXS*	PA2	*SHELXS*
1	4	2	0.09	0.21
2	3	1	0.14	0.41
3	1	1	0.05	0.17
4	2	4	0.04	0.18
5	1	1	0.06	0.21
6	2	1	0.13	0.18
7	3	6	0.06	0.27
8	6	6	0.19	0.25
9	2	1	0.08	0.16
10	6	2	0.10	0.21
11	2	1	0.06	0.20
12	1	1	0.10	0.21
13	2	1	0.05	0.33
14	1	1	0.04	0.22
15	4	7	0.11	0.18
16	3	1	0.19	0.28
17	3	1	0.14	0.20
18	2	1	0.25	0.26
19	12	10	0.12	0.41
20	8	8	0.14	0.20
21	4	3	0.07	0.19
22	3	2	0.15	0.22
23	10	3	0.15	0.23
24	3	2	0.05	0.18
25	2	3	0.04	0.18
26	3	5	0.08	0.22
27	3	4	0.03	0.18
28	2	5	0.03	0.17
29	1	1	0.13	0.20
30	2	4	0.06	0.17
31	1	4	0.14	0.24
32	5	5	0.17	0.45
33	10	23	0.13	0.49
34	9	10	0.07	0.35
35	5	2	0.25	0.36
36	3	5	0.15	0.38
37	2	4	0.19	0.40
38	2	1	0.22	0.25
Min.	1	1	0.03	0.16
Max.	12	23	0.25	0.49
Average	4	4	0.11	0.25

This work was supported in part by the Joint NSF/NIGMS Initiative to Support Research in the Area of Mathematical

Biology under NIH award GM072023. We are grateful to Alexander B. Smith and Dr Scott R. Wilson for discussions as well as for helping us with the computations with *SHELXS*. Finally, we thank Professor G. M. Sheldrick for useful comments regarding the *SHELX* codes, and two anonymous referees for numerous suggestions that helped us improve the quality of this manuscript.

References

Blessing, R. H. (1989). *J. Appl. Cryst.* **22**, 396–397.

Bricogne, G. (1984). *Acta Cryst.* A**40**, 410–445.

Chantrapromma, S., Razak, I. A., Fun, H. K., Karalai, C., Zhang, H., Xie, F. X., Tian, Y. P., Ma, W., Zhang, Y. H. & Ni, S. S. (2001). *Acta Cryst.* C**57**, 289–290.

Cowan, J. A., Howard, J. A. K. & Leech, M. A. (2001). *Acta Cryst.* C**57**, 1196–1198.

Debaerdemaeker, T. & Woolfson, M. M. (1983). *Acta Cryst.* A**39**, 193–196.

Frampton, C. S., Gall, J. H. & MacNicol, D. D. (2000). *Acta Cryst.* C**56**, e22.

Gelder, R. de (2003). Personal communication.

Gelder, R. de, de Graaff, R. A. G. & Schenk, H. (1993). *Acta Cryst.* A**49**, 287–293.

Germain, G., Main, P. & Woolfson, M. M. (1970). *Acta Cryst.* B**26**, 274–285.

Germain, G. & Woolfson, M. M. (1968). *Acta Cryst.* B**24**, 91–96.

Giacovazzo, C. (1998). *Direct Phasing in Crystallography. Fundamentals and Applications.* Oxford University Press.

Graaff, R. A. G. de (2003). Personal communication.

Hauptman, H. A. & Karle, J. (1953). *Am. Monograph* 3. *Solution of the Phase Problem. I. The Centrosymmetric Crystal*. Michigan: American Crystallographic Association.

Hauptman, H. A., Xu, H., Weeks, C. M. & Miller, R. (1999). *Acta Cryst.* A**55**, 891–900.

Hempel, A., Camerman, N., Mastropaolo, D. & Camerman, A. (2000). *Acta Cryst.* C**56**, 1225–1227.

Ho, D. M. (2003). Personal communication.

Hughes, E. W. (1953). *Acta Cryst.* **6**, 871.

ILOG (2000). *CPLEX 7.0 User's Manual.* ILOG CPLEX Division, Incline Village, NV, USA.

Karle, J. & Karle, I. L. (1966). *Acta Cryst.* **21**, 849–859.

Kim, J. S., Lee, W. K., Rim, J. A., Jensen, W. P., Lee, J.-H., Kim, M.-J. & Suh, I. H. (2000). *Acta Cryst.* C**56**, 1369–1371.

Leverd, P. C., Weber, D., Habicher, W. D. & Nierlich, M. (2000). *Acta Cryst.* C**56**, 997–998.

MacLean, E. J., Teat, S. J., Farell, D. M. M., Ferguson, G. & Glidewell, C. (2002). *Acta Cryst.* C**58**, o470–o473.

McKee, V. & Morgan, G. G. (2003). *Acta Cryst.* C**59**, o150–o152.

Markowitz, H. M. (1957). *Manage. Sci.* **3**, 255–269.

Massa, W. (2000). *Crystal Structure Determination.* Berlin: Springer-Verlag.

Nemhauser, G. L. & Wolsey, L. A. (1988). *Integer and Combinatorial Optimization. Series in Discrete Mathematics and Optimization.* New York: Wiley Interscience.

Perera, K. P. U., Abboud, K. A., Smith, D. W. & Krawiec, M. (2003). *Acta Cryst.* C**59**, o107–o110.

Robinson, P. D., Hou, Y. & Meyers, C. Y. (1999). *Acta Cryst.* C**55**, IUC9900147.

Sayre, D. (1952). *Acta Cryst.* **5**, 60–65.

Sheldrick, G. M. (1990). *Acta Cryst.* A**46**, 467–473.

Sheldrick, G. M. (2005). Personal communication.

Thuéry, P., Nierlich, M., Asfari, Z. & Vicens, J. (2000). *Acta Cryst.* C**56**, 343–344.

Vaia, A. & Sahinidis, N. V. (2003). *Acta Cryst.* A**59**, 452–458.

Zhu, L. G., Ellern, A. M. & Kostić, N. M. (2002). *Acta Cryst.* C**58**, o129–o130.

Zou, W., Chen, P., Gao, Y. & Meng, J. (2003). *Acta Cryst.* E**59**, o337–o339.