

SIAG/OPT

Views-and-News

A Forum for the SIAM Activity Group on
Optimization

Volume 10 Number 1

February 1999

Contents

Case Studies

Train Control Optimization

Susanna P. Gordon and Pamela J. Williams 1

Linear Programming for Emergency

Broadcast Systems

Michael C. Ferris and Todd S. Munson 6

Improving the Optimization and Numerics

in Laue Diffraction Analysis

Zhong Ren, Rongqin Sheng, and

Stephen J. Wright 8

Comments from the Chair and Editor

Thomas Coleman and Juan Meza 12

Case Studies

Train Control Optimization

S.P.Gordon and P.J.Williams

Sandia National Laboratories, Livermore, CA 94551

1. Introduction.

A new era of automatic train control has begun, in which mass transit trains will be commanded with precision beyond the capabilities of past systems. Although transit properties such as San Francisco's Bay Area Rapid Transit (BART) have controlled their trains automatically for decades, the control

systems have limited capability. Increases in capacity now require trains to run closer together than these systems can accommodate, so new systems are being developed. These new systems, such as the Advanced Automatic Train Control (AATC) system under development by BART in collaboration with Raytheon Corporation and Harmon Industries, will increase the capacity of the system through more accurate train location and more precise control. Although the need to increase capacity has been the main driver for these new systems, they will also enable smoother service and improved energy management. However, incorporating smoother service and improved energy management into the control system will require optimization of a complex dynamic system.

Sandia has collaborated with BART to develop a simulator of the train control and power consumption of the AATC system. The simulator has enabled us to develop enhanced train control algorithms to supplement the safety-critical controller. These algorithms do not attempt to globally optimize the control system with respect to a cost function, but rather they modify the baseline vital control using heuristics to smooth out train operations, and to reduce energy consumption and power infrastructure requirements. Although enhanced train control algorithms provide a valuable first step toward optimization, they represent only a fraction of the ultimate capabilities of the system. We are now beginning work toward true optimization of the control system. Train control optimization encompasses such classes of optimization as mixed integer nonlinear programming, nonlinear discrete-time optimal control, and multi-objective optimization.

2. Automatic Train Control.

Current automatic systems employ fixed block circuitry, hard-wired into the train tracks, to locate and transmit speed commands to the trains. Due to the fixed circuitry, trains can be located and issued new speed commands at intervals of several hundred to a thousand feet. In addition, the train speed may only be commanded with roughly ten available speed commands separated by 5 to 20 mile-per-hour increments. Finally, a train may only accelerate at full or half acceleration. The AATC system, by con-

trast, will use radios to locate the trains to within 15 feet, will send new commands every half second, will transmit speed commands with one mile-per-hour precision, and will provide fully selectable acceleration rates [1]. The train commands will be generated by station (zone) controllers, which will command all of the trains on lengths of track several miles long. This architecture, shown in Figure 1, permits some centralized-control capability, with the actions of multiple trains in an area coordinated by a single controller. There is a Central control, but it will provide only limited capability, such as adjusting station dispatch times and train ‘performance levels’. However, the control system is not truly centralized, as there are multiple station controllers, each with control over a section of the overall system. Thus, the AATC system provides some of the benefits of centralized control, but simultaneously contains the complexity of distributed control.

In the initial work on this problem, the distributed nature of the control was ignored to make the problem tractable, and the control system was treated as though it was centralized. It was assumed, however, that each station controller would be aware of the trains in its neighboring zones, so that each controller could make more global decisions. Neighboring zone knowledge is particularly important for power-related algorithms, since all of the trains in an area, including trains traveling in opposite directions, share power resources.

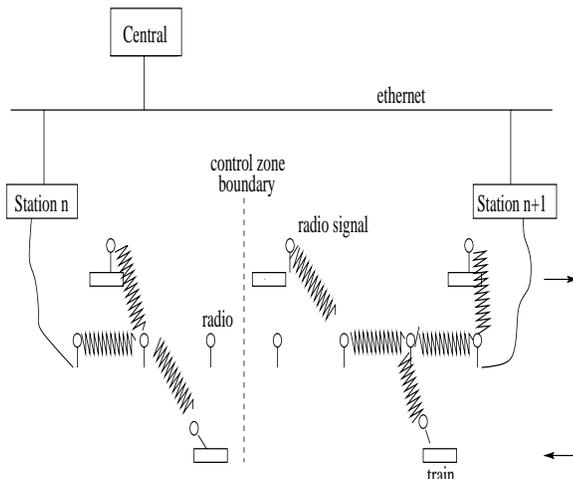


Figure 1.1: AATC communication system architecture

3. Control Optimization.

In general, there are two classes of optimization problems related to train control. The first is a scheduling problem. In this case, trip time from end-to-end of the system must be minimized, subject to some capacity constraints. The second problem treats the schedule as a constraint, and optimizes for the smoothest ride and the minimum energy requirements. Thus far, we have focused on the second problem.

3.1 Schedule optimization.

The layout of the BART system is shown in Figure 2. Since most commuter traffic is headed to and from San Francisco, the R-, C-, and A-lines all merge at the Oakland Wye into the M-line in San Francisco. This results in a high density of traffic on the M-line, as well as a severe constraint on the schedule. Trains approaching the merge are given a time slot during which they must enter the Wye, so that the trains from each line will neatly interleave. Projecting back from this merge, while allowing adequate time at each station stop, constrains the dispatch time of the trains onto each of the R-, C-, and A-lines.

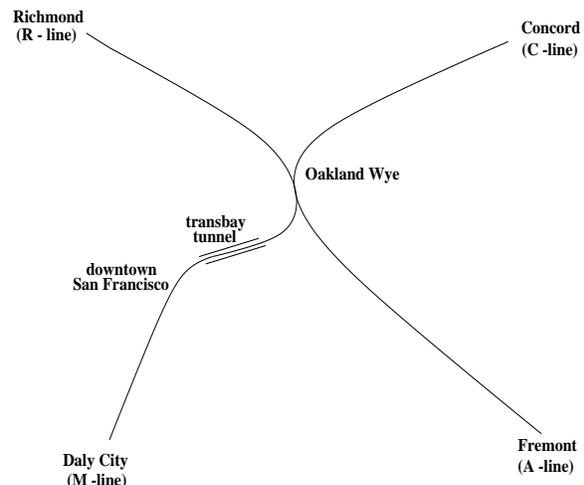


Figure 1.2: BART system diagram

In addition to satisfying the Oakland Wye constraint, the control system must simultaneously attempt to maximize system throughput and minimize scheduled headway, to provide passengers with fast and frequent service. Headway is defined as the time between consecutive trains passing a point on the

system. When trains are far enough apart, trains do not affect each other's behavior and each train's trip time from end-to-end of the line is minimized. If headway is reduced sufficiently, trains will begin to interfere and will be forced to brake occasionally in order to maintain a safe following distance. Operation at such 'interfered headways' results in increased trip time. Thus, there is a trade-off between headway and trip time minimization.

Longer trip times not only irritate passengers, but also increase infrastructure costs because more trains are required on the system. When a train reaches the end of the line, it must be turned around and prepared to begin a run back up the line. If the train arrives at the end of the line with enough time to turn around before the next dispatch time, it becomes the next scheduled train. On the other hand, if there is insufficient turn around time, then another train must be available for the next run. Thus, a small reduction in trip time can potentially reduce the number of trains required to be simultaneously operational and consequently reduce infrastructure costs.

Unfortunately, minimizing trip time and satisfying the schedule constraint at the Oakland Wye does not leave much flexibility for considering additional objectives. However, the schedule has considerable impact on other important service- and energy-related goals. On the service side, commuters appreciate the ability to transfer easily from line to line at transfer stations. For example, there is a line that runs from Richmond to Fremont. Commuters from Richmond to San Francisco would like to board the first arriving train, independent of its destination, and, if necessary, transfer in the Wye. The BART system has not always managed to create a schedule that allows for efficient transfers.

On the energy side of the equation, energy may be saved on the system from coordinated train arrival and departure times in stations. BART trains regenerate energy when they brake, and regenerative energy cannot flow back into the electric utility's power grid. Some of this energy may go unused, unless another train is accelerating nearby. Coordinated station stops and departures may save energy. However, even a schedule with perfect coordination for minimum energy usage may be confounded by random dwell-time delays due to passengers block-

ing doors and the like.

To date, service-related goals have been taken into account in the schedule, but energy-related goals in general have not. We have not pursued this type of optimization with BART, because the schedule is heavily constrained, and it may not be possible to make significant improvements. We believe there is a larger payoff in the schedule-constrained arena described below.

3.2 Schedule-constrained control optimization.

As we have explained, the train schedule is designed to minimize trip time. Thus, the nominal behavior for a train is to attain full acceleration out of a station as soon as the schedule permits, to maintain the best possible speed in between stations, and to brake for the next station stop. Any decreases in train acceleration or speed would result in increases in trip time, which could lead to missed slots in the merge or late arrivals at the end of the line necessitating additional trains on the system. Consequently, there is little room for optimization of train operations under these conditions.

Nonetheless, the system is designed to allow the scheduling of trains at close headway, so that any small delay can cause train interference. Interference occurs when a train is forced to brake or travel slower than its nominal speed because it is closely following another train. Suppose trains are scheduled at two-minute headways, and a train is delayed by over a minute. The train behind it will catch up, and will brake to maintain a safe following distance. Train interference due to delays can include anything from slight premature braking between stations, to full-fledged backups. These events can also result in wasted energy or even power shortages. On a daily basis, the current system experiences approximately 20 delays of five or more minutes. Interference will become more the rule than the exception as the scheduled headway becomes shorter, and under these conditions control optimization can play an important role. The enhanced train control algorithms that we have designed to date for the AATC system have targeted off-nominal conditions, such as backups, braking due to interference, and low train motor voltages due to excessive power demand. We

will discuss only two examples here, but more are contained in [2] and [3].

The first control algorithm relates to backup recovery. A backup occurs if a train stops for a period that is several times the scheduled headway. This can occur in either a station or between stations. An algorithm was developed to recognize backups and to reduce the speeds of approaching trains in an effort to avoid their stopping outside the station behind the stopped train. If the delay continues for a prolonged period, some trains will eventually be forced to stop in a backup behind the delayed train. When the delayed train finally begins to move, the algorithm staggers the starts of any stopped trains to avoid simultaneous acceleration, which can lead to power spikes and voltage sags. In addition, approaching trains are controlled so as to arrive after the backup clears. If additional delays occur in the station, then the algorithm again reduces the speeds of all approaching trains accordingly. As long as additional delays are on the order of the station dwell time (20 seconds) or less, this approach prevents further stoppages. However, a substantial delay to a second train can cause the backup to recur, and the algorithm restarts.

Figure 3 shows the results of simulation runs in which a train is delayed in a station for 400 seconds. Trains are approaching the backup at 120-second intervals. The graphs show the full length of each train as a shaded region along the location axis. The trajectory is flat when a train is stopped and sloped when it is in motion. The backup moves through the station with nominal control in Figure 3a, and with the delay-recovery algorithm operating in Figure 3b. Not only does this control technique provide obvious improvements in passenger comfort and reduced wear-and-tear on the motors from mode-changes (switching from propulsion to braking), it also accrues power-related benefits. If such a delay occurs in an area of limited power availability, the algorithm helps to prevent low voltages. Voltage sags can result from the large power demand of multiple accelerating trains, and can cause train motors to shut down to avoid damage.

The delay-recovery algorithm discussed above helps to avoid low voltages by spreading power demand over time. However, there are many potential causes of high power demand. Therefore an al-

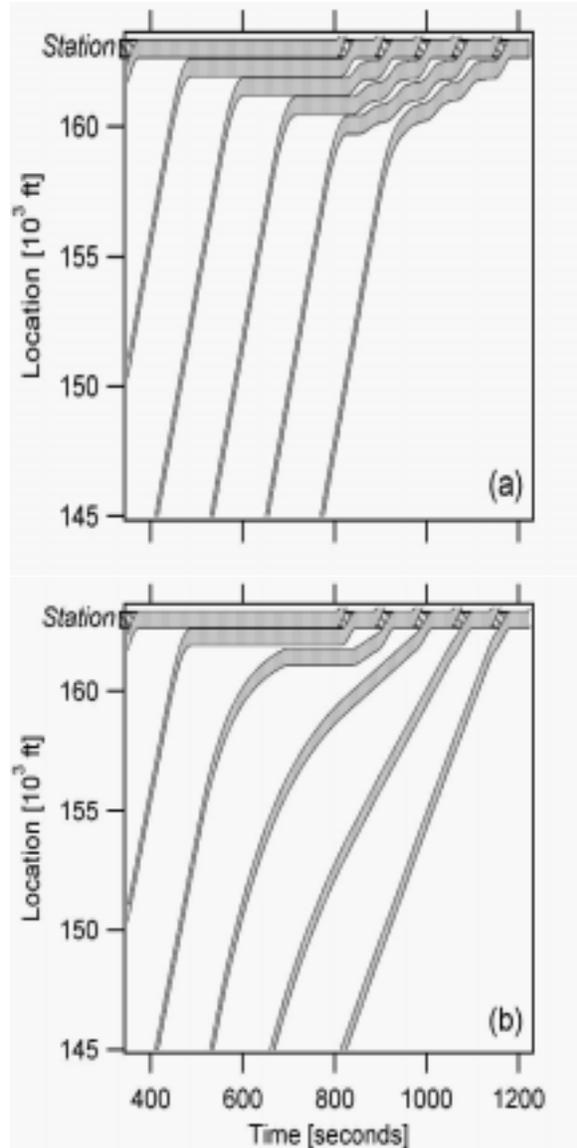


Figure 1.3: 400 second delay in a station. Approaching trains arrive at 120-second intervals. (a) Train trajectories under nominal control during a backup. (b) Train trajectories under enhanced train control. The delay recovery algorithm smoothly and efficiently controls backed-up trains, while limiting stops outside of the station.

gorithm to avoid low voltages in general would be quite valuable. Such an algorithm requires the capability to predict train voltage, which is a nonlinear function of the power demand of each train in a region. To provide this predictive capability, we have developed a neural network that can estimate the voltage at a train given the power demands in the

train's vicinity. Using this neural net, we have developed an algorithm that we believe can prevent power demand surges and their resultant voltage sags, thereby avoiding costly train motor shutdowns or damage. The only alternatives to such an algorithm are either the modification of on-board motor control to reduce power demand in response to low voltage or the installation of sufficient power infrastructure to avoid low voltages regardless of the trains' behavior. To date, BART has used the latter approach. If an algorithm such as the one we tested can modify train behavior to avoid such events, this can save tens of millions of dollars in power infrastructure costs.

4. Modeling Issues.

Thus far, only heuristic control algorithms have been designed for the AATC system. Now, we want to pursue the ultimate goal of enhanced train control by optimizing the system with respect to an objective function. To accomplish this goal with respect to the schedule-constrained control problem, we can model enhanced train control as either an optimal control problem or a dynamic programming problem.

In the first scenario, we could pose the control algorithm as a nonlinear discrete-time optimal control problem, where the control variables are the speed and acceleration commands of each train. The state variables would be the locations, speeds, and accelerations of all the trains.

As an alternative we could formulate enhanced train control as a dynamic programming problem and backward predict each train's behavior as a function of its immediate lead train. Assume we know the lead train's location, speed, and acceleration at time T . Using this predicted information, we can optimize our objective function with respect to the current speed and acceleration commands while preventing interference.

Irrespective of the problem formulation, the first step is to select a metric to optimize. Possible metrics include minimization of train-minutes of delay, reduction of energy costs, reduction of motor mode-changes, and improvement of passenger comfort. Initially, we will ignore the energy-related concepts and concentrate on improving passenger comfort. Although energy-related concepts are explicitly

ignored, a smoother ride tends to increase system reliability and avoid power spikes.

Passenger comfort may be measured in terms of the smoothness of the ride, which is related to the amount of accelerating or decelerating that a train exhibits. Our objective function will represent a measure of ride smoothness. One possibility is

$$\min \frac{1}{2} \int_0^T [\mathbf{u}(t)^T \mathbf{u}(t)] dt,$$

where $\mathbf{u}(t)$ is the acceleration command vector at time t and T is the total simulation time.

First, we will optimize the system locally between station stops instead of over the entire line. As in the heuristic control algorithm, we will initially ignore the distributed complexity of the system by assuming a single controller and a single control zone.

Speed safety checks will be incorporated as constraints.

5. Summary.

The new generation of communication-based train control systems such as the AATC will allow new possibilities in precision train control and multi-train coordination. Energy as well as control-related objectives may be pursued with these systems. We have taken a first step in this direction, using heuristic control algorithms to enhance the capabilities of the AATC. Optimization of these systems provides an important challenge, with potential benefits for transit systems worldwide.

Complete train control optimization will involve such research areas as mixed integer nonlinear programming to model the scheduling problem, nonlinear discrete-time optimal control problems to model interference and backup recovery, and finally multi-objective optimization to simultaneously optimize varied energy- and service-related objectives.

Acknowledgments

The authors are indebted to Timothy Sa, Don Sheaffer, Richard Wheeler, Paul Boggs, Patty Hough, and Kevin Long of Sandia National Laboratories and David Lehrer, John Evans, and Gene Nishinaga of BART for their many and varied contributions to this work. This work was supported under

Contract DE-AC04-94AL85000. Accordingly, the United States Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for United States Government purposes.

REFERENCES

- [1] E. Nishinaga, J.A. Evans, and G.L. Mayhew. Wireless Advanced Automatic Train Control. *IEEE Vehicular Technology News*, 41 (p. 13), 1994.
- [2] S.P. Gordon and D.G. Lehrer. Coordinated Train Control and Energy Management Control Strategies. 1998 ASME/IEEE Joint Railroad Conference (IEEE, 1998), pp.165-176.
- [3] S.P. Gordon and D.G. Lehrer. Service- and Energy-Related Optimization of AATC. 1998 Rapid Transit Conference (APTA, 1998), CD-ROM.

Linear Programming for Emergency Broadcast Systems

Michael C. Ferris and Todd S. Munson

Computer Sciences Department, University of Wisconsin,
1210 West Dayton Street, Madison, Wisconsin 53706. ¹

This short note describes an application of linear programming embedded in emergency broadcast system hardware. In many communities, large sirens sound when tornados, or other disasters, threaten. These warning systems save many lives each year. However, the current system is typically ineffective for several reasons. Each siren covers a broad area and is difficult or even impossible to hear in many locations. Furthermore, individuals are unable to distinguish between different disaster types. The action taken during a tornado warning is very different from that taken when a chemical spill occurs. Finally, some disasters only affect a small area, while the siren blankets the entire community.

As a result, Alert Systems [1] has developed a system to overcome these limitations. In a 911-center, a simple graphical interface displays a map of the controlled area. The operator outlines a polygon containing the current danger area using a mouse

pointer device. The coordinates of the polygon's vertices and a relevant message are then transmitted as a radio signal on a controlled frequency to receivers located throughout the community. If a receiver, which knows its own location, is within the transmitted polygon, it must start beeping and display the short message detailing the nature of the disaster transmitted with the coordinates. If it is outside the affected area, the system remains silent.

This problem is easily seen to be a linear program. In essence, it is just a feasibility problem; determine if the location $b = (b_1, b_2)$ of the receiver is within the polygon defined by the vertices x^1, \dots, x^n . More simply stated, can b be represented as a convex combination of the vertices x^1, \dots, x^n ? To solve this problem we introduce artificial variables e_1, e_2 and set up the linear program:

$$\begin{aligned} \min_{\lambda, e} \quad & e_1 + e_2 \\ \text{subject to} \quad & \sum_{i=1}^n x^i \lambda_i + De = b \\ & \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, e_1, e_2 \geq 0 \end{aligned}$$

where D is a diagonal matrix with diagonal entries $D_{ii} = \text{sign}(b_i - x_i^1), i = 1, 2$. We choose these values for D in order to guarantee that the linear program has a feasible point, ($\lambda = (1, 0, \dots, 0), e = |b - x^1|$). Furthermore, the problem is bounded below by 0. Hence, it has an optimal solution which can be found by the simplex method. If the optimal value is 0, the receiver is within the polygon and the signal is activated; otherwise one of the errors e_i is positive and the receiver is outside the polygon and remains inactive.

End of story – not quite. The remaining difficulty is that the receiver must be mass produced. Hence, the manufacturer has decided to use a processing unit that has been restricted so that only integer arithmetic is allowed. While we could emulate floating point operations on this processor, we might encounter problems with numerical error. We do not consider this option because the system must be stable for all allowable inputs (we have real people depending upon the outcome). Another option is to store all of the values as rational numbers. By writing routines to perform the operations required using rationals, the code can be executed using exact arithmetic. However, a much simpler technique which only uses integer variables and the addition, subtraction, and multiplication operations is devel-

¹This research was supported in part by Alert Systems, Inc.

oped below.

The manufacturer suggested forcing the coordinates x^i to lie on an integer grid and regarding the location of the system as integer. The issue is now to implement the simplex method using only integer arithmetic. Given a canonical form linear program

$$\begin{array}{ll} \min_y & c^T y \\ \text{subject to} & Ay = b \\ & y \geq 0 \end{array}$$

we assume that an initial basic feasible solution is known with the variables partitioned into basics, B , and non-basics, N . This is easy to ensure in the above application. The steps of the revised simplex method [2] are then:

1. $y_B = A_{.B}^{-1}b$.
2. $r_N^T = c_N^T - c_B^T A_{.B}^{-1} A_{.N}$; if $r_N \geq 0$ stop optimal.
3. Choose $j = N(s)$ s.t. $r_j < 0$.
4. Calculate $d = A_{.B}^{-1} A_{.j}$. If $d \leq 0$, stop unbounded.
5. Choose r s.t. $\frac{y_{B(r)}}{d_r} = \min\{\frac{y_{B(i)}}{d_i} \mid d_i > 0\}$.
6. Swap $B(r)$ and $N(s)$ and goto 1.

In the following discussion, we assume that A , b , and c consist of integer data. Using the fact [3] that

$$A_{.B}^{-1} = \frac{1}{\det(A_{.B})} \text{adj}(A_{.B})$$

where $\text{adj}(\cdot)$ denotes the adjugate matrix, and some algebraic manipulation, we can refine the revised simplex method to use only integer arithmetic. Note that the determinant and adjugate can be evaluated in integer arithmetic using only additions and multiplications. For example, in step 1, we could evaluate

$$y_B = \frac{1}{\det(A_{.B})} \text{adj}(A_{.B})b.$$

We will now outline why the required division here is redundant in the application.

For step 2, we note that the reduced costs, r_N^T , are invariant under multiplication by a positive constant. Letting $\delta = \text{sign}(\det(A_{.B}))$, it is clear that that $\delta \det(A_{.B})$ is a positive constant. We now calculate our reduced costs as follows:

$$R_N^T := \delta \det(A_{.B}) r_N^T = \delta (\det(A_{.B}) c_N - c_B^T \text{adj}(A_{.B}) A_{.N})$$

If $R_N^T \geq 0$, then we can stop at an optimal solution. Otherwise choose $j = N(s)$ s.t. $R_N^T < 0$.

Since

$$d = \frac{1}{\det(A_{.B})} \text{adj}(A_{.B}) A_{.j},$$

step 4 becomes: if $\delta \text{adj}(A_{.B}) A_{.j} \leq 0$ stop unbounded. Otherwise, we need to perform the ratio test of step 5. Looking at $\frac{y_{B(r)}}{d_r}$ we immediately note that the quantity $\frac{1}{\det(A_{.B})}$ factors out of each term.

We implicitly store the resultant $\frac{\text{adj}(A_{.B}) b_r}{(\text{adj}(A_{.B}) A_{.j})_r}$ as a rational number. We find the minimum of all the eligible values by calculating a common denominator, and comparing the integer numerator.

To summarize, our refined revised simplex method is as follows:

1. Calculate δ and R_N^T as above. If $R_N^T \geq 0$, stop optimal. Otherwise choose $j = N(s)$ s.t. $R_N^T < 0$.
2. If $\delta \text{adj}(A_{.B}) A_{.j} \leq 0$ stop unbounded. Let $I = \{i \mid \delta (\text{adj}(A_{.B}) A_{.j})_i > 0\}$.
3. While $|I| > 1$ do
 - (a) Let $\bar{i}, \hat{i} \in I$.
 - (b) Let

$$\Delta := \text{sign}((\text{adj}(A_{.B}) A_{.j})_{\bar{i}} (\text{adj}(A_{.B}) A_{.j})_{\hat{i}}).$$

(c) If

$$\Delta (\text{adj}(A_{.B}) b)_{\bar{i}} (\text{adj}(A_{.B}) A_{.j})_{\hat{i}} \leq \Delta (\text{adj}(A_{.B}) b)_{\hat{i}} (\text{adj}(A_{.B}) A_{.j})_{\bar{i}}$$

then $I = I \setminus \hat{i}$.

(d) Otherwise, $I = I \setminus \bar{i}$.

4. Let r be the remaining element in I . Swap $B(r)$ and $N(s)$ and goto 1.

Note in particular that no divisions are required. The solution values y_B are not needed since for the application considered we only need to test whether the optimal solution of the linear program is zero

or positive. Furthermore, for the production version

of the code we have included the smallest subscript anti-cycling rule [2] to prevent the simplex method from failing to terminate because of cycling.

For large problems, this method is impractical and not recommended. However, for the feasibility problem that Alert Systems needs to solve where typically n is less than 10, the above method works extremely well. Furthermore, by looking at our specific grid size, we can symbolically evaluate all of the information needed and determine the maximum magnitude of the integers required. This information helps in the actual implementation of the algorithm in hardware.

REFERENCES

- [1] pageALERT. <http://www.alertsys.com/>.
- [2] V. Chvátal. *Linear Programming*. W.H. Freeman and Company, New York, 1983.
- [3] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, England, 1985.

Improving the Optimization and Numerics in Laue Diffraction Analysis

Zhong Ren,¹ Rongqin Sheng,² and Stephen J. Wright³

1. Introduction.

When X-rays are beamed onto a crystal, they are diffracted to produce a regular array of “spots” of varying intensity on an area detector. The locations of the spots are determined by the crystal lattice, while their intensities depend on the spatial and temporal average conformation of all molecules in the crystal and during data collection. By processing

¹Department of Biochemistry and Molecular Biology, The University of Chicago, 920 East 58th Street, Chicago, IL 60637, USA; renz@cars.uchicago.edu

²Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA; sheng@mcs.anl.gov

³Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439, USA; wright@mcs.anl.gov

the data in these images, we can extract information about the structure of the molecules. For large biomolecules, this computational problem is one that tests the limits of current capabilities.

We are interested in the Laue method of X-ray diffraction, in which the incident X-ray is not monochromatic, but rather is made up of a spread of different wavelengths. The Laue technique produces complex diffraction images, but is more suitable in situations in which data must be gathered quickly, such as when our intention is to observe reactions in progress (see, for example, Genick et al. [1]). Another major advantage is that Laue diffraction is more suited to synchrotron sources (such as the Advanced Photon Source at Argonne National Laboratory), which naturally produce bright X-rays with a spread of wavelengths.

The purpose of the LaueView code is to process the raw data to obtain a set of corrected, accurate structure factor amplitudes. LaueView was written by one of the authors of this article (Zhong Ren) and is described in detail by Ren and Moffat [5, 6]. Our mission in the current project was to enhance the optimization and numerical techniques used by LaueView, to make it produce results of the same or better quality in less computing time.

The focus of our investigations was on the scaling part of the code, in which the measured intensities are corrected to account for distortion effects. This operation accounts for about half the execution time on a typical data set and involves the most challenging algorithmic issues. (The remaining operations in the code either require minimal time or are easily parallelized.)

2. Outline of X-ray Diffraction.

Diffraction of X-rays by a crystal admits a beautiful mathematical explanation in terms of lattice theory, simple geometry, and other classical tools. Consider a three-dimensional lattice with basis vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} in \mathbf{R}^3 , so that each point in the lattice can be expressed as

$$x\mathbf{a} + y\mathbf{b} + z\mathbf{c}, \quad \text{where } x, y, \text{ and } z \text{ are integers.} \quad (1)$$

The parallelepiped whose sides are the vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} is referred to as the *unit cell*. The *reciprocal*

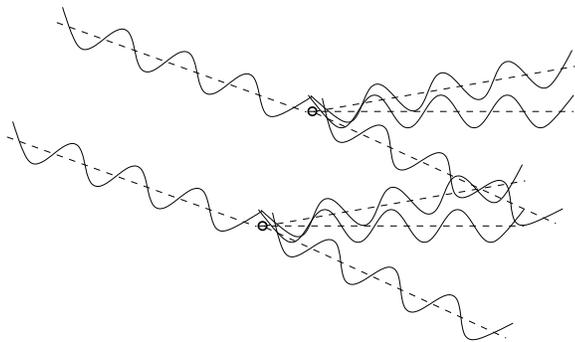


Figure 1.4: Diffraction from two points in the lattice (beams diffract in all directions from all points; we show just a few directions here)

lattice is characterized by three basis vectors $\bar{\mathbf{a}}$, $\bar{\mathbf{b}}$, and $\bar{\mathbf{c}}$ with the following properties:

$$\begin{aligned} \mathbf{a} \cdot \bar{\mathbf{a}} &= 1, & \mathbf{a} \cdot \bar{\mathbf{b}} &= 0, & \mathbf{a} \cdot \bar{\mathbf{c}} &= 0, \\ \mathbf{b} \cdot \bar{\mathbf{a}} &= 0, & \mathbf{b} \cdot \bar{\mathbf{b}} &= 1, & \mathbf{b} \cdot \bar{\mathbf{c}} &= 0, \\ \mathbf{c} \cdot \bar{\mathbf{a}} &= 0, & \mathbf{c} \cdot \bar{\mathbf{b}} &= 0, & \mathbf{c} \cdot \bar{\mathbf{c}} &= 1, \end{aligned} \quad (2)$$

where “ \cdot ” denotes the standard (Euclidean) inner product. The difference vector \mathbf{R} between any two lattice points is expressible in a form similar to (1), that is,

$$\mathbf{R} = x\mathbf{a} + y\mathbf{b} + z\mathbf{c}, \quad x, y, \text{ and } z \text{ integers.} \quad (3)$$

In a crystal, the molecules are arranged in a lattice. Each molecule in the crystal contains a number of electrons, distributed in a cloud about the atomic nuclei. When each of these electrons encounters the incident X-ray, the electron is set in motion and becomes an oscillating dipole, and therefore a source of secondary radiation. This process is known as “scattering.” Interference between the X-rays scattered from the electrons in the crystal gives rise to the diffraction patterns observed on the detector.

Suppose for the moment that scattering takes place from a single scattering center at the same location in each of the molecules. Because of the crystalline structure, these scattering centers make up a lattice, whose basis vectors we denote as above by \mathbf{a} , \mathbf{b} , and \mathbf{c} . Consider *any two* centers, as shown in Figure 1.4, with displacement \mathbf{R} of the form (3). Suppose that the incident beam has direction \mathbf{t} , which we express in terms of the reciprocal lattice basis by

$$\mathbf{t} = x_t\bar{\mathbf{a}} + y_t\bar{\mathbf{b}} + z_t\bar{\mathbf{c}}, \quad \|\mathbf{t}\| = 1/\lambda, \quad (4)$$

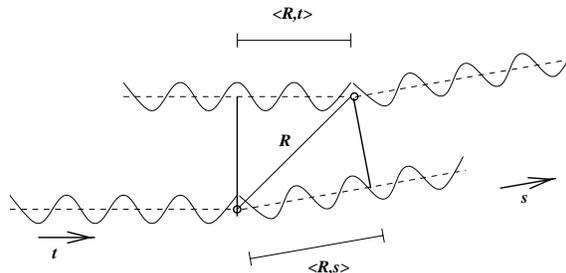


Figure 1.5: Beam from direction \mathbf{t} diffracted in direction \mathbf{s} from two lattice points separated by \mathbf{R} , showing difference in path length

for some coefficients x_t , y_t , and z_t . (The normalization condition $\|\mathbf{t}\| = 1/\lambda$ ensures that each direction is uniquely specified by the coefficient triple (x_t, y_t, z_t) .) Suppose we investigate a particular direction of scattering \mathbf{s} , also defined in terms of the reciprocal basis vectors with the same normalization condition as in (4) by

$$\mathbf{s} = x_s\bar{\mathbf{a}} + y_s\bar{\mathbf{b}} + z_s\bar{\mathbf{c}}, \quad \|\mathbf{s}\| = 1/\lambda. \quad (5)$$

In Figure 1.5, we illustrate scattering in the direction \mathbf{s} from the two lattice points separated by the displacement \mathbf{R} of the form (3). The scattered beams will remain in phase provided that *the difference in path length is an integer multiple of the wavelength λ* . Since the path difference is $\mathbf{R} \cdot (\mathbf{s} - \mathbf{t})$, this condition can be expressed as

$$\begin{aligned} \lambda \mathbf{R} \cdot (\mathbf{s} - \mathbf{t}) &= \lambda(x\mathbf{a} + y\mathbf{b} + z\mathbf{c}) \cdot \\ & \quad [(x_s - x_t)\bar{\mathbf{a}} + (y_s - y_t)\bar{\mathbf{b}} + (z_s - z_t)\bar{\mathbf{c}}] \quad (6) \\ &= \lambda x(x_s - x_t) + \lambda y(y_s - y_t) + \lambda z(z_s - z_t) \\ &= \lambda M, \quad \text{for some integer } M, \quad (7) \end{aligned}$$

where we used the relations (2) to derive the second equality. Recall that if \mathbf{s} is to yield a spot on the detector, the relation (7) must hold for *all* pairs of points in the lattice, that is for *all integers* x , y , and z . This is possible only if the coefficients of \mathbf{s} satisfy the relations

$$x_s - x_t = h, \quad y_s - y_t = k, \quad z_s - z_t = \ell, \quad (8)$$

where h , k , and ℓ are *integers*.

The integer triple (h, k, ℓ) , along with the fixed direction \mathbf{t} of the incident beam, completely characterizes the direction \mathbf{s} . We refer to (h, k, ℓ) as the *Miller indices*.

To summarize, we conclude from (5) and (8) that the direction \mathbf{s} will produce a spot on the detector if it satisfies the following conditions for some set of Miller indices (h, k, ℓ) :

$$\mathbf{s} = (x_t + h)\bar{\mathbf{a}} + (y_t + k)\bar{\mathbf{b}} + (z_t + \ell)\bar{\mathbf{c}}, \quad (9a)$$

$$\|(x_t + h)\bar{\mathbf{a}} + (y_t + k)\bar{\mathbf{b}} + (z_t + \ell)\bar{\mathbf{c}}\| = 1/\lambda. \quad (9b)$$

These relations constitute *Bragg's law*. Laue diffraction uses a number of wavelengths so there are many more triples (h, k, ℓ) that satisfy these relations for *some* value of λ in the range of the incident beam's wavelengths.

Variation in intensity between the spots occurs because the scattering can happen from any point in the electron cloud, not just the point scattering centers in our simplified discussion above. Information about the electron density function and the locations of the atoms within each molecule then can be deduced from the measured intensities. We refer the reader to [4] for additional details.

3. The LaueView Code.

LaueView, the code with which we worked in this study, analyzes the measured intensity data from the detector and outputs the structure factor amplitudes for each set of Miller indices (h, k, ℓ) . For details, see Ren and Moffat [6]. After determining the measured intensities (a process that itself involves numerical integration and nonlinear least-squares techniques), corrections are applied to account for such effects as temperature factors, radiation damage, general absorption, and, most important, the varying intensity of the incident beam across its range of wavelengths. The resulting nonlinear least-squares data-fitting problem involves a large number of observations (of the order of 10^5 – 10^6 for a protein data set) and a relatively small number of parameters (typically 100–200).

LaueView2.5 is a Fortran 77 code of approximately 50,000 lines. Most of its arithmetic is performed in single precision. A Unix shell script is used to set up each run. The user edits this script to select the parameters to be varied and those to be fixed on a particular run and to indicate whether the code should start “cold” or use the approximate solution generated by a previous run as its starting

point. LaueView was used to produce the structures reported in Ren et al. [7] and Genick et al. [1]

LaueView seeks a multiplicative scaling factor f to be applied to each measured intensity I , to obtain the corrected intensity fI . The factor f depends on the Miller indices (h, k, ℓ) and an index i of the particular observation of this reflection. In addition, f depends on various parameters whose values are to be recovered from the data-fitting process. It is composed of a product of 12 factors, that is,

$$f_L f_P f_\lambda f_{\text{isoS}} f_{\text{anisoS}} f_{\text{isoB}} f_{\text{anisoB}} f_{\text{isoD}} f_{\text{anisoD}} f_A f_U f_O,$$

where each factor represents a correction for a different effect, such a polarization, temperature factors, isotropic and anisotropic scale factors, and wavelength normalization. Each of these factors depends on parameters whose values are recovered by fitting the model to the set of measured intensities by a nonlinear least-squares technique. The wavelength normalization factor f_λ corrects for the fact that the incident beam contains a spread of wavelengths, of varying intensity. The curve that relates wavelength to intensity, known as the λ curve, is not known directly—it must be parameterized and reconstructed by fitting our intensity data and by using our knowledge of the crystal symmetry. Redundant measurements at different wavelengths and knowledge of the symmetry are instrumental in reconstructing the λ curve.

In LaueView2.5, the λ curve is defined in terms of the Chebyshev basis functions $\cos(i \arccos z)$, $i = 1, 2, 3, \dots$, for $z \in [-1, 1]$. In our modified version, LaueView3.1, we switched to piecewise quadratic basis functions with local support, which gave an more compact parameterization of the curve and hence reduced the processing time.

4. Numerical Computing Issues.

LaueView makes use of a number of numerical techniques in curve-fitting, nonlinear least-squares, and numerical linear algebra. In addition, it lends itself quite readily to parallel implementation.

We mentioned above that a change to the curve-fitting procedure yielded significant savings. In the nonlinear least-squares computation, LaueView2.5 used a Levenberg-Marquardt algorithm from *Numerical Recipes* [3] which was not very efficient, par-

ticularly as it involved computation of the Jacobian even at candidate iterates that were rejected because of higher function value. We changed to an implementation based on the trust-region Levenberg-Marquardt algorithm described by Moré [2]. One special feature of our problem was that it was not practical to compute and store the full Jacobian J of our residual function r , since the dimensions of this matrix could be as large as $10^6 \times 200$. Rather, we evaluated the rows of J and the elements of r in sequence, and accumulated the products $J^T J$ and $J^T r$, using these small data structures as the basis of the step computation.

In LaueView2.5, single precision accumulation of $J^T J$ caused this matrix to be numerically indefinite, and the negative gradient vector $-J^T r$ to not be a descent direction for the objective function $\frac{1}{2}\|r\|^2$. These features caused the least-squares method to stall and terminate prematurely. We switched to double-precision accumulation of these quantities.

The chief linear algebra operation was a decomposition of $J^T J$. Though this operation was not so critical, we replaced the *Numerical Recipes* routine with the LAPACK routine DSYEV. The most expensive part of the calculation is evaluation of the function and derivative quantities $r^T r$, $J^T J$ and $J^T r$, at a given parameter vector x . As mentioned above, LaueView evaluates these terms r_i^2 , $r_i \nabla r_i$, and $\nabla r_i (\nabla r_i)^T$ in sequence for $i = 1, 2, \dots, m$ and accumulates the quantities $\|r\|^2$, $J^T r$, and $J^T J$.

A parallel version of this process proceeds in the obvious way: The indices $i = 1, 2, \dots, m$ are “dealt out” to the P available processors. Each processor forms its own partial sums, and a global summation operation then recovers $r^T r$, $J^T J$ and $J^T r$. The other operations associated with LaueView, including computation of the candidate Levenberg-Marquardt step, take place concurrently (and redundantly) on all processors; their relatively low computational cost makes it not worthwhile to parallelize them.

5. Computational Results.

We report on the effects of the improvements in LaueView on a real data set from a crystal of photoactive yellow protein, for which structure results obtained with LaueView2.5 are presented in [1].

Table 1.1: CPU Times (seconds) for LaueView 2.5 and 3.1 on SGI Reality Monster, for Six Runs with Different Starting Points and Different Fixed Parameters

LaueView2.5		LaueView3.1 ($n_\lambda = 32$)	
time (s)	optimal f	time (s)	optimal f
2096	1045598	191	1017473
3725	485746	247	509350
7700	576934	386	591343
failed		559	492059
not tested		1617	491801
10841	633712	1171	610565

The value of m for this set is approximately 120,000, while the number of parameters n is small, between about four and seventy in our experiments. Numerical experiments were conducted from a number of different starting points, with different parameters being allowed to vary on each run.

We performed experiments on two computational platforms. The first was a single processor of a SGI Onyx2 Reality Monster running IRIX 6.4, equipped with sixteen MIPS R10000 processors and 4 GB of memory. The second was an 80-node IBM SP in which each node is an RS/6000 workstation equipped with a 120 MHz P2SC chip and 256 MB of memory. Enhanced file systems were used on both systems, because the code needs to write (and in some cases to read) very large files.

Table 1.1 summarizes performance on the SGI machine. LaueView2.5 is the original version of the code prior to numerical improvements, while LaueView3.1 incorporates all the improvements described above. In all cases, LaueView3.1 obtained comparable final objective values to the original code, while greatly improving the robustness and efficiency of the computation. The larger functions values may be attributable in part to our use of a different parametrization of the λ curve (using fewer parameters), and may also be due to the code identifying a different local minimizer.

Results of the parallel code running on the IBM SP are presented in Table 1.2. We show results on the first and last runs from Table 1.1. LaueView3.1 was used with the number of basis functions set to 64.

Table 1.2: Parallel Performance of LaueView3.1 on IBM-SP Multiprocessor

Procs.	Time (s)	Speedup	(excl. output)
1	551		
2	377	1.5	1.6
4	274	2.0	2.6
1	4443		
2	2522	1.8	1.8
4	1334	3.3	3.5
8	765	5.8	6.5
16	485	9.2	11.3
32	351	12.7	17.3

Since we parallelized only the critical section of the code in which the matrix $J^T J$ and the vector $J^T r$ are evaluated and accumulated, the speedups are considerably less than linear in the number of processors; the non-parallel parts of the computation (particularly the 100 seconds spent in writing the output file to disk) become relatively more significant as the number of processors is increased. However, the wall-clock time is reduced considerably on multiple processors. The final column in Table 1.2 indicates the speedup figure obtained when the 100 seconds spent on writing the output file is subtracted from the total CPU time for each run. A more sophisticated parallel code could parallelize this operation by having each processor write a section of the output to its own file, while any subsequent run could read these files in parallel, in an analogous fashion. We feel that the run-time advantages obtained with our current parallelization technique are sufficient to meet the needs of experimentalists for faster turnaround time, however.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38, by a DOE Grand Challenge Application Grant, B&R No. KJ-01-01-03-0, and in part by NIH Grant RR07707 to Keith Moffat.

REFERENCES

- [1] U. K. Genick, G. E. O. Borgstahl, K. Ng, Z. Ren, C. Pradervand, P. M. Burke, V. Srajer, T.-Y. Teng, W. Schildkamp, D. E. McRee, K. Moffat, and E. Getzoff. Structure of a protein photocycle intermediate by millisecond time-resolved crystallography. *Science*, 275:1471–1475, March 1997.
- [2] Jorge J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G.A. Watson, editor, *Lecture Notes in Mathematics, No. 630—Numerical Analysis*, pages 105–116. Springer-Verlag, 1978.
- [3] W. H. Press, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in Fortran: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.
- [4] Z. Ren, R. Sheng, and S. J. Wright. Advanced computational techniques for Laue diffraction analysis. Preprint ANL/MCS-P740-0199, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., January 1999.
- [5] Zhong Ren and Keith Moffat. Deconvolution of energy overlaps in Laue diffraction. *Journal of Applied Crystallography*, 28:482–493, 1995.
- [6] Zhong Ren and Keith Moffat. Quantitative analysis of synchrotron Laue diffraction patterns in macromolecular crystallography. *Journal of Applied Crystallography*, 28:461–481, 1995.
- [7] Zhong Ren, Kingman Ng, Gloria E. O. Borgstahl, Elizabeth D. Getzoff, and Keith Moffat. Quantitative analysis of time-resolved Laue diffraction patterns. *Journal of Applied Crystallography*, 29:246–260, 1996.

Comments from the Chair and Editor

After a rather lengthy interval, we are happy to say the SIAG/OPT Newsletter is back with a wonderful set of articles concerned with applications of optimization. We will carry this application theme forward with the next issues. Indeed, the very next issue will be focused entirely on optimization applied to the problems of finance. We aim to release this issue before the SIAM Optimization meeting in Atlanta where Coleman and Dembo (CEO, Algorithms) will offer a short course on Financial Optimization on May 9.

For more information on the conference, check out the web site for the Sixth SIAM Conference on Optimization, <http://www.siam.org/meetings/op99/index.htm>.

We would like to remind you that this is your newsletter. Please consider submitting an expository article on your favorite optimization application (or any other interesting aspect of optimization.). Such articles can be very useful in spreading the word about optimization and its importance, throughout the SIAG/OPT community and beyond. They can also be wonderful classroom aids, helping students see the importance of optimization and applied mathematics more generally. Please contribute.

Thomas F. Coleman, SIAG/OPT Chair
Computer Science
Cornell University
Ithaca, NY 14853
coleman@tc.cornell.edu

Juan C. Meza, Editor
Sandia National Laboratories
P.O. Box 969, MS 9011
Livermore, CA 94551
meza@ca.sandia.gov
