informs

ICS NEWS

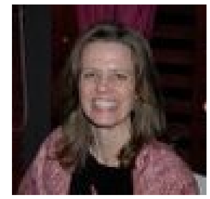
INFORMS Computing Society Newsletter

Inside This Issue

"The Society is a great opportunity to get involved, meet some new people and generally combine your professional interests with a little fun."

Message from the Chair

Robin Lougee-Heimer, T. J. Watson Research Center, IBM



"The Computer Science Technical Section is in excellent shape, with a healthy treasury, a sizable and active membership, a strong presence at the national INFORMS meetings, its own successful conference series, an excellent journal, an informative newsletter, an established and recognized prize for excellence, and attendees that know how to have a good time." So reported Past-Chair Dick Barr over 10 years ago in this newsletter, and except for the name, everything couldn't be more true today.

Treasury: At the beginning of 2008, the Society had approximately $33,000. Thanks to a generous gift from the Mica Fonden of Denmark, we are establishing a new endowment for the ICS Student Paper Award. At the current exchange rate, the endowment is valued at $15,000.

Membership: We started 2008 with 535, of which more than 100 were new members who joined under our successful student membership drive. This year, we created a new Membership Committee led by Dick Barr to continue exploring new means of enhancing the society's value to its members and growing our ranks. (Renew your membership!)

Presence at National Meeting: In Seattle, we had a record 81 sessions, a 150% increase over the previous year. All indicators point to another ICS-rich

(continued on page 29)

Remembering Alex Orden, 1916–2008

Saul Gass, University of Maryland
Linus Schrage, University of Chicago

The Optimization/Mathematical Programming community lost one of its pioneers February 9, 2008 when Alex Orden passed away in the University of Chicago Hospital.

Alex was a member of the post-World War II (WW II) generation of mathematicians who came upon Operations Research (OR), not at a university, but by finding themselves in a job that brought them to the subject. They looked at OR, tasted OR, liked OR, and made amazing pioneering scientific contributions, still used today, to not only OR, but also mathematics.

(continued on page 25)

[29] A. K. Ziliaskopoulos and D. Kotzinos, A Massively Parallel Time-Dependent Least-Time Path Algorithm for Intelligent Transportation Systems Applications, *Computer-Aided Civil and Infrastructure Engineering* 16:5 (2001), 337–346.

Dantzig's Indirect Contribution to Music Research: How the von Neumann Center of Gravity Algorithm Influenced the Center of Effect Generator key-finding Algorithm

Elaine Chew, University of Southern California ▷
echew@usc.edu (▷BibTEX entry)

Elaine Chew is the 2007–08 Edward, Frances, and Shirley B. Daniels Fellow at the Radcliffe Institute for Advanced Study at Harvard, and Associate Professor of Industrial and Systems Engineering and of Electrical Engineering at the University of Southern California Viterbi School of Engineering. At USC, she was the first holder of the Viterbi Early Career Chair, and founder of the Music Computation and Cognition Laboratory. She was honored in 2004/05 by the NSF Career and PECASE awards, respectively, for her efforts in integrating research and education at the intersection of music and engineering. Elaine earned her Ph.D. and S.M. degrees in Operations Research from the Massachusetts Institute of Technology, with an interdisciplinary dissertation on mathematical modeling of tonality. She arrived at MIT with a B.A.S. in Mathematical and Computational Sciences (honors) and in Music Performance (distinction) from Stanford University. Elaine also holds diplomas and degrees in piano performance from Trinity College, London (F.T.C.L. & L.T.C.L.), and Stanford University.

This article grew out of a presentation at Stanford University, hosted by David O. Siegmund, for the students of the Mathematical and Computational Sciences Program.

In 1991, when George Dantzig took on a summer undergraduate research student (yours truly), he had no idea that he was to indirectly impact computational music research. Then, a junior in the Mathematical and Computational Sciences Program at Stanford University, I had applied to, and received, a Summer Undergraduate Research Fellowship, which had the tongue-in-cheek acronym of SURF. When I approached Dantzig to be my research supervisor, he asked if I would prefer to do research, or data collection. Convinced that research was the more glamorous undertaking, I chose the former, to which he replied whimsically, "You know that, in research, you will be wasting a lot of paper." Thus, I began work on implementing and testing the von Neumann Center of Gravity Algorithm and Dantzig's extension to the algorithm.

In the wake of Karmarkar's much-publicized discovery of a polynomial-time algorithm for linear programming, George Dantzig himself was re-visiting the first interior-point algorithm, the von Neumann Center of Gravity Algorithm, communicated verbally to him in 1948 by John von Neumann. Dantzig had just documented proof of convergence for the von Neumann Center of Gravity Algorithm, and his proposed extension of the algorithm that had a guaranteed polynomial bound [6, 7].

The von Neumann Center of Gravity Algorithm solves the following problem: given n points on a unit sphere centered on the origin in m dimensions, find non-negative weights so that the weighted sum of the n points is the origin. Each von Neumann iteration could be calculated in relatively few computational steps. Dantzig showed that the algorithm had lower polynomial complexity (degree 2) than Karmarkar's, but a much higher constant based on the required precision.

While the von Neumann Center of Gravity Algorithm was guaranteed to converge, it did so very slowly after the initial steps. Dantzig's modification — to bracket the target — aimed to speed the convergence. A description of, and theorems related to, the von Neumann Center of Gravity Algorithm would later be incorporated into the chapter on early interior-point methods in [9, pp. 70–84].

As a Summer Undergraduate Research Fellow, I wrote a program to test the von Neumann algorithm with bracketing on a set of small problems, and presented highlights of the results in [8]. A factor affecting algorithm performance is the *radius of feasibility*, which is the radius of the largest sphere centered at the origin and inscribed in the convex hull of the n points on the unit sphere. Here is a summary of the findings: The bracketing method worked well for most problems, yielding a solution with very few von Neumann iterations. The bracketing method performed best when solving linear programs with many more variables than constraints, and that had a large radius of feasibility. When the radius of feasibility is very small and the number of constraints large, the iterations converged slowly.

Eight years later, in my Ph.D. dissertation [1], I was to propose the Spiral Array model for tonality, and the Center of Effect Generator (CEG) algorithm for key-finding. The Spiral Array model represents tonal entities in music — pitches, chords, and keys — as points on nested helices in the same three-dimensional space. Higher-level constructs are generated by successive aggregation, as weighted sums of their components.

The tonal context of a segment of music is represented by a summary point, the Center of Effect, CE, a weighted sum of the pitch set. As more music is 'heard,' the position of the CE generally gravitates toward the key, and the key is computed through a nearest-neighbor search. The CEG method was shown to converge faster, than other existing methods, to the key (in 3.75 steps, which is on par with human hearing) when applied to a classical test set.

In retrospect, the Spiral Array model and CEG algorithm was very much influenced by the kind of geometric and interior-point approach embodied in the von Neumann Center of Gravity Algorithm.

Computational Music Research

This work on the Spiral Array and the CEG algorithm marked the first of my many experiments in computational modeling of music. Researchers at the interface of Operations Research (OR) and Computing have applied computational OR techniques to areas ranging from telecommunications, to biology, to finance, so why not music? The area of Sound and Music Computing is growing exponentially due to industry interest in music search, browsing, and recommendation for both handheld and desktop applications.

In a recent article, *Music and Operations Research — the Perfect Match?* [*OR/MS Today*, June 2008], I describe the proliferation of new conferences, and the founding of a new society and journal of mathematics and computation in music. I am co-guest editing a special issue on computation for the *Journal of Mathematics and Music* [www.tandf.co.uk/journals/titles/17459737.asp], to appear in summer 2008, and co-chairing the program committee of the *Ninth International Conference on Music Information Retrieval* [ismir2008.ismir.net], to take place in Philadelphia, September 2008.

In the OR/MS article, I argued that OR has much to offer the burgeoning field of Music and Computing. I gave examples of how mathematical and computational modeling has been, and can be, employed in the three areas of music research: analysis, expressive performance, and composition/improvisation. Examples of direct application of OR techniques to music problems include François Pachet's use of Markov models [11] in his Continuator, a human-machine improvisation system, and Charlotte Truchet's use of Constraint Programming [13] in computer-assisted composition. More examples are given in [4] to demonstrate that music offers a rich area for exploration and discovery, not only in human cognition in music, but also in mathematical and computational techniques.

> ***"OR has much to offer the burgeoning field of Music and Computing... mathematical and computational modeling has been, and can be, employed in the three areas of music research: analysis, expressive performance, and composition/improvisation."***

When Harvey Greenberg extended the invitation to contribute an essay to the INFORMS Computing Society Newsletter, I decided to return to my original contribution in computational music research. This article thus puts the von Neumann Center of Gravity Algorithm and the Spiral Array CEG algorithm side-by-side for comparison and contrast. The structure of the article is as follows: I shall first describe the von Neumann Center of Gravity Algorithm, and Dantzig's bracketing technique; then, I shall present the Spiral Array model and the CEG algorithm; the connections will be discussed in the concluding section.

The von Neumann Center of Gravity Algorithm

The von Neumann Center of Gravity Algorithm solves linear optimization problems of the form:

$$\sum_{j=1}^{n} P_j x_j = 0, \quad \sum_{j=1}^{n} x_j = 1,$$
$$\|P_j\|_2 = 1, \quad x_j \geq 0 \qquad \forall\, j.$$

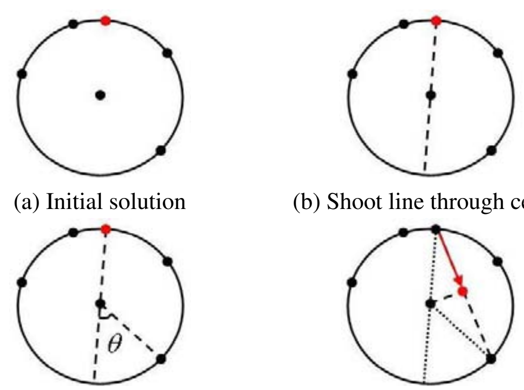
The P_j's are points on a unit ball centered on the origin, and the goal is to find the combination of weights, x_j's, on these points so that they sum to the origin. Let the set of weights at iteration t be $\{x_j^t\}$, and the approximation to the origin be $A^t = \sum_{j=1}^{n} P_j x_j^t$. By design, the sequence $\{A^t\}$ converges to the origin.

A two-dimensional example demonstrates the von Neumann Center of Gravity Algorithm. Assume that $n = 5$ and that the five P_j's are situated on the unit circle as shown in Figure C-1(a). Any of these five points on the circumference can serve as the initial solution, A^0, such as the P_j that is colored red in Figure C-1(a). The x_j^0 corresponding to this P_j is set equal to 1, and all others are set to zero.

At iteration t, draw a line from the current solution, A^t, through the center of the circle, as shown in Figure C-1(b). Find the P_j that makes the smallest acute angle, θ, with this line, P_{acute}, as illustrated in Figure C-1(c).

If there is no P_j on the opposite side of the circle, i.e., $\theta > \pi/2$, then the problem is infeasible. Otherwise, drop a perpendicular from the origin to the line through A^t and P_{acute} to get the new solution point, A^{t+1}. Update the weights on the P_j's accordingly. This step is shown in Figure C-1(d).

Dantzig [7] proved that, independent of the number of rows, m, and columns, n, in the problem, a precision of ε can be guaranteed with less than $1/\varepsilon^2$ iterations, where ε is the distance between the solution and the origin. Thus, convergence can be slow if the solution must be very close to the origin.



(a) Initial solution (b) Shoot line through center

(c) Find point with smallest angle (d) Drop perpendicular

Figure C-1. The von Neumann Center of Gravity Algorithm (▷Larger picture)

Dantzig's Bracketing Technique

The bracketing technique proposed in [6] aims to speed the convergence of the von Neumann Center of Gravity Algorithm by providing larger targets that are centered on the vertices of a simplex inside the convex hull of the P_j's.

This bracketing technique is illustrated in two dimensions in Figure C-2. Instead of targeting the origin, the bracketing technique applies the von Neumann Center of Gravity Algorithm $m + 1$ times, with each vertex of a simplex inside the unit ball as target. Such a simplex is shown in Figure C-2. For each vertex, the algorithm iterates until the approximate solution converges to a point within a given radius of the vertex.

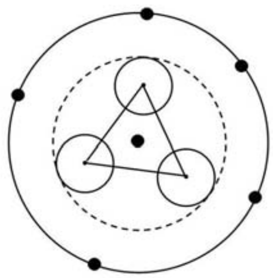


Figure C-2. The Bracketing Technique (▷Larger picture)

The vertices, and the corresponding radius defining the required precision for these targets, are chosen so that the balls circumscribing each vertex fall within the circle (or sphere) that lies inside the convex hull of the P_j's. One such circle that resides inside the convex hull is indicated by the dotted line in Figure C-2. In practice, one does not know the radius of the dotted circle, and different radii were tested empirically in the experiments that I ran.

Once approximate solutions inside the distributed targets are found, the origin is bracketed by these solutions, and one can then solve for the origin with straightforward linear operations.

Undergraduate Research Experience

For my SURF, I implemented the von Neumann Center of Gravity Algorithm, and Dantzig's bracketing extension. The program was written in Pascal (recall that this was in 1991) and run on a SPARC station. I tested 47 files of data, each a matrix, varying in size from 3×4 to 28×40, recording the minimum radius of the simplex tested in the bracketing method (an approximation to the radius of feasibility), the number of operations and function calls (including von Neumann iterations), and whether a feasible solution was found.

We found, on these small test problems, that the von Neumann Center of Gravity Algorithm with bracketing works well for most problems, yielding a solution with relatively few von Neumann steps. For problems that have a small radius of feasibility, the iterations converged slowly, and the bracketing extension does little to improve on the original von Neumann Center of Gravity Algorithm.

Eight years later, as a PhD student at the Operations Research Center at MIT, I proposed the Spiral Array model for tonality, and the Center of Effect Generator (CEG) algorithm for key-finding [1]. While I did not set out to deliberately employ ideas from the von Neumann algorithm and its extension, in retrospect, my undergraduate experience nevertheless influenced my later work in mathematical modeling of tonality.

Tonality in a Nutshell

The Spiral Array [1] is a mathematical model for tonality. More specifically, it is a geometric model that represents elements of the tonal system underlying the music with which most of us are familiar. These elements include: (1) pitches, sounds having a fundamental frequency; (2) chords, simultaneous sounding of multiple pitches (the triad, which consists of three pitches, is a kind of chord); and, (3) keys, a collection of pitches, the ordered sequence of which exhibits a distinct pattern of perceived stability. The pitch set of a key can be unambiguously defined by three triads. The name of the key is the pitch name of the most stable tone. Tonality refers to this set of interrelations amongst pitches and sets of pitches.

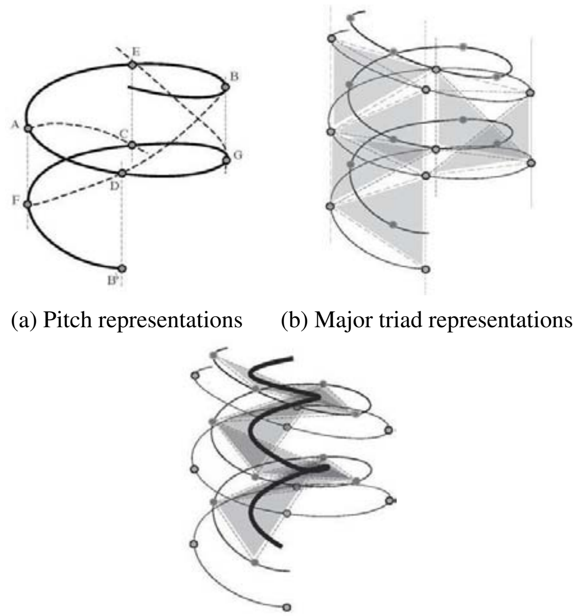
The existence of the tonal system is one of the main reasons why we can form expectations, and have them resolved, when listening to music — for example, we can hear when a melody is ended, and when it is not and wants to go on (try singing only the first three phrases of 'Happy Birthday.') There are many reasons to model tonality — for example, to create a representation on which to base algorithms for computational music analysis, and to understand human perception and cognition. Examples of investigations into mental representations of tonality include [12, 14]. My work on the Spiral Array has focused primarily on the motivations of computational analysis. Algorithms for automated analysis drive the systems for automatic accompaniment and computer-assisted composition, and for analysis and synthesis of expressive music performances.

The Spiral Array Model

The Spiral Array model aims to model aspects of tonality. It consists of an array of nested helices that each represents a kind of tonal entity — pitches, major/minor chords, and major/minor keys — in music. Higher-level constructs are generated successively as the convex sum of their components.

Figure C-3 shows some of the components of the Spiral Array model. Figure C-3(a) depicts the outermost pitch class helix. Pitch representations are spaced evenly at each quarter turn of the helix. Each node represents a pitch class, i.e., a C is not just the middle C on a keyboard, but all C's at different octaves above and below it. Frequencies of pitches in the same class are related by powers of two. Neighboring pitch classes along the helix contain pitches with frequency ratios of approximately 2:3, and vertical neighbors have ratios of approximately 4:5.

Unlike previous models that employ network (or their dual) representations of pitch classes, for example [10], the Spiral Array uses the interior space to define spatial representations of chords and keys as weighted sums of their components. Each triad is represented as a point on the triangle outlined by its component pitches; Figure C-3(b) shows the major triad representations, which themselves lie on an inner helix. Minor triads and their corresponding helix are defined in a similar fashion. The major key representations are generated by convex combinations of their defining triads, which would be three adjacent major triads for a major key, as shown in Figure C-3(c). The minor key helix is produced in a similar way. Figure C-3(c) also depicts the nested helices for pitch classes, major triads, and major keys, in decreasing order of radii.



(a) Pitch representations (b) Major triad representations

(c) Major key representations

Figure C-3. The Spiral Array Model (reproduced from [3])
(▷Larger picture)

Center of Effect Generator Key-Finding Method

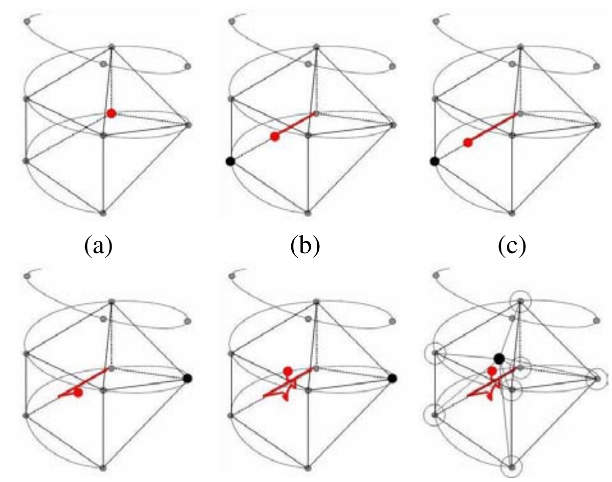
The CEG algorithm [1, 2] can be illustrated simply with a melody. A melody consists of a sequence of note events, each note having the properties of pitch and duration. The algorithm generalizes to more complex music with simultaneous tones at any given time. In the Spiral Array, the tonal context of a segment of music is represented by a summary point, the Center of Effect, CE, of the pitches. A CE of a collection of notes can be generated as the sum of the pitch positions, weighted by their respective durations.

Given a melody, the CEG algorithm successively generates CEs as each note event occurs, thus updating itself as it gravitates towards the key representation. The distance between CE and key need not decrease monotonically; the CE trace can move toward, or away from, a key. The key at any given time is determined by a nearest neighbor search for the closest key representation.

Figure C-4 provides a pictorial guide to the CEG algorithm. Figure C-4(a) shows the initial CE at the first note of the melody, which is the pitch representation of the first note. At the second note, which is of the same duration as the first, and represented by the black disc in Figure C-4(b), the CE moves to the midpoint between the first and second pitches. Suppose the third note is the same as the second; then the CE simply moves closer to the pitch of the second/third note, as shown in Figure C-4(c). The next step is shown in Figure C-4(d). As the iterations continue, a trajectory is traced in the interior of the model.

Suppose that, at the state shown in Figure C-4(e), one wishes to determine the key. The key is found by searching for the nearest key representation. The solution key and the convex hull of its pitch set are shown in Figure C-4(f).

When tested on a small test set of the fugue subjects of all 24 fugues in the *Well-Tempered Clavier* Book I, ignoring correct answers on the first note, the algorithm found the key of the fugue in 3.75 note events on average, which is on par with key-finding performance by humans, and faster than previous methods.



(a) (b) (c)

(d) (e) (f)

Figure C-4. The Center of Effect Generator Algorithm
(▷Larger picture)

Inspirations

The most obvious similarity between the von Neumann Center of Gravity Algorithm and the Spiral Array CEG algorithm is the geometric and interior-point approach that underlies both methods. The von Neumann Center of Gravity Algorithm is an early interior point algorithm, while the Spiral Array CEG algorithm can be loosely considered an interior point approach to key-finding. A difference is that, in the CEG algorithm, convergence is not guaranteed; the goal of music is not to converge monotonically to a key, but to create interest through the varying of distances to different keys.

The von Neumann algorithm works best in problems where the number of variables is extremely large, compared to the number of constraints. The Spiral Array is presently defined only in three dimensions, thus there exists a fixed limit to the problem size. The CEG algorithm is thus highly amenable to real-time applications, as in the MuSA.RT analysis and visualization system [5].

Direct implementation of the idea of bracketing in key-finding still requires some thought. The idea exists in music cognition: notes that sound imply certain chords, and these chords in turn point to the key context. The challenge in employing parallel optimization to multiple targets in music analysis is the management of time. Music unfolds in a single stream that is experienced over time. It is unclear whether the iterations toward multiple targets should be implemented simultaneously or in series in a real-time system that mimics human key-finding abilities. A possible CEG algorithm with bracketing could be to determine the recent chords (the distributed targets) and use them to determine the key (the bracketed ultimate target).

In conclusion, I have given an overview of the von Neumann Center of Gravity Algorithm and the Spiral Array CEG algorithm, and demonstrated the conceptual similarities between them. It is without a doubt that many other OR and computational techniques can be applied directly to, or inspire, the solution of problems in music analysis and synthesis. So, this is a call to all closet musicians out there in the OR and Computing community: you too can play a part in furthering computational music research, and have fun doing it!

Acknowledgments

Apart from the influence of my undergraduate research experience with Dantzig, this work would not have been possible without my dissertation supervisor, Jeanne Bamberger, who introduced to me early work on Music and Artificial Intelligence, and my OR advisor, Georgia Perakis, for shepherding my interdisciplinary Ph.D. dissertation.

This material is based in part upon work supported by the Stanford Summer Undergraduate Research Fellowship in 1991, the Josephine de Kármán Dissertation Fellowship in 1999, the Edward, Frances, and Shirley B. Daniels Fellowship at the Radcliffe Institute for Advanced Study, and by the National Science Foundation under Grant No. 0347988. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the above funding organizations.

Elaine relaxes on the steps of MIT, where she performed many of her concerts before 2000 and will do so again May 12, 2008, in a program exploring *The Mathematics in Music* through contemporary compositions that employ rhythmic, melodic, and tonal combinations, permutations, and transformations.

Photo by Paul Keel

"The challenge in employing parallel optimization to multiple targets in music analysis is the management of time."

References

[1] E. Chew, *Towards a Mathematical Model of Tonality*, Ph.D. dissertation, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2000.

[2] E. Chew, Modeling Tonality: Applications to Music Cognition, in *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, Edinburgh, Scotland, 2001, 206–211.

[3] E. Chew, Regards on Two Regards by Messiaen: Post-Tonal Music Segmentation Using Pitch Context Distances in the Spiral Array, *Journal of New Music Research* 34:4 (2005), 341–351.

[4] E. Chew, Music and Operations Research — The Perfect Match? *OR/MS Today*, June 2008.

[5] E. Chew and A. R. J. François, Interactive multi-scale visualizations of tonal evolution in MuSA.RT Opus 2, *ACM Computers in Entertainment* 3:4 (2005), article 3E, 16 pages.

[6] G. B. Dantzig, *Converting a converging algorithm into a polynomially bounded algorithm*, Report SOL 91-5, System Optimization Lab, Stanford University, 1991.

[7] G. B. Dantzig, *An ε-Precise feasible solution to a constraint in $1/\varepsilon^2$ iterations independent of problem size*, Report SOL 92-5, System Optimization Lab, Stanford University, 1992.

[8] G. B. Dantzig, *Bracketing to speed convergence illustrated on the Von Neumann algorithm for finding a feasible solution to a linear program with a convexity constraint*, Report SOL 92-6, System Optimization Lab, Stanford University, 1992.

[9] G. B. Dantzig and M. N. Thappa, *Linear Programming 2: Theory and Extensions*, Springer-Verlag, New York, NY, 2003.

[10] D. Lewin, *Generalized Musical Intervals and Transformations*, Yale University Press, New Haven, CT, 1987.

[11] F. Pachet, The Continuator: Musical Interaction With Style, *Journal of New Music Research* 32:3 (2003), 333–341.

[12] R. N. Shepard, Structural Representations of Musical Pitch, in D. Deutsch (ed.), *The Psychology of Music*, Academic Press, London, UK, 1982, 344–390.

[13] C. Truchet, Some Constraint Satisfaction Problems in Computer Assisted Composition and Analysis, in E. Lluis-Puebla, G. Mazzola, and T. Noll (eds.), *Perspectives of Mathematical and Computer-Aided Music Theory*, epOs Music, Osnabrück, FRG, 2003.

[14] R. J. Zatorre and C. L. Krumhansl, Mental Models and Musical Minds, *Science* 298:5601 (2002), 2138–2139.

Remembering Alex Orden, 1916–2008

(continued from page 1 ◁)

Alex received his undergraduate degree in optics from the University of Rochester in 1937 and an M.S. degree in physics from the University of Michigan in 1938. During WW II he worked on the design of proximity fuses at the National Bureau of Standards (NBS) in Washington, D.C. After the war, he went to MIT and received his Ph.D. in mathematics