Towards a Multimedia Implementation of an NP-Complete Dance Choreography Problem

Nigel Gwee
Department of Computer Science
Southern University and A&M College
Baton Rouge, LA 70813, USA
nigel_gwee@subr.edu

Abstract—**The Dance Choreography problem is NP-Complete. To solve non-trivial instances of such problems, heuristic algorithms are required, which we have implemented in our software we call Terpsichore©. The software creates amalgamations using figures from established International Standard Ballroom Dance syllabi, as well as our own proprietary syllabus. A multimedia-based interface provides various coaching and teaching modules, delivered in a natural, human-like manner**.

Keywords-NP-Complete; software; multimedia; ballroom; dance; choreography

I. INTRODUCTION

In his seminal work on computational complexity, Stephen A. Cook introduced the first NP-Complete problem [3]. In doing so, he opened the gates for future researchers to discover many significant computational problems for which no known efficient solutions exist. The breadth of these complex problems is indeed extensive, ranging from well-known problems in database manipulation, operations research, and yes, music and dance. These problems, once identified as NP-Complete, share one important factor: an efficient solution for any one of them, if found, implies an efficient solution for all of them, now numbering in the hundreds, if not thousands. This theoretical result justifies the intense interest in computational solutions to such problems. In this article, we add one more NP-Complete problem to the fold: dance choreography.

A. *Current Work in Dance and Computers*

Current literature that links dance and computers is not large. Work with principal dance notational systems includes that of Nakamura, Hall, Herbison-Evans, Sealey, Badler, and Smoliar [21, 14, 23, 1]. Research in using computer software to create and animate choreography in various dance genres has been carried out by Calvert, Wilke, Ryman, Fox, Yu, Johnson, Herbison-Evans, Green, Butt, Lansdowne, and Noll [2, 24, 15, 19, 22].

One difficulty is the nature of dance notation, which is unusually complex because it has to capture human motion in three dimensions. It is not at all easy to describe and perform choreography with sufficient degree of exactitude, much less conduct research and transmit concepts and results. To our knowledge, no significant research has been carried out on the use of computer software to choreograph amalgamations for ballroom dance: a unique genre in that it is the only type of dance that involves a structured partnership between a man and a woman.

In the following sections, we first describe the theoretical implications of NP-Complete theory. We then define Dance Choreography as an addition to the set of NP-Complete problems. We then introduce Terpsichore©, our software implementation of the dance choreography problem, and demonstrate its features, highlighting key design decisions.

II. DANCE CHOREOGRAPHY AND NP-COMPLETE THEORY

In this section we briefly describe the mathematical theory of NP-Complete and NP-Hard problems. We then define the Dance Choreography problem and show that it belongs to this class of problems.

A. *NP-Complete Theory in a Nutshell*

An NP-Complete problem is one for which no one has yet found an efficient computerized solution. The theory of NP-Complete problems is a central theory of computer science: its importance lies in the implication that an efficient solution to any one NP-Complete problem automatically gives a solution to the whole body of NP-Complete problems. Technically speaking, an efficient solution is generally one whose solution time increases by no more than a polynomial factor as the size of the problem increases. An NP-Complete problem is one in which the solution verification can be efficiently undertaken, and to which any other NP-Complete problem can be efficiently converted, but for which no known efficient solution current exists. Currently, only Exhaustive Algorithms are known to produce optimal solutions for such problems, but these are not practical for large instances. In these latter cases, sub-optimal algorithms such as Greedy Algorithms may have to suffice. For further details of this central topic of computer science, consult [5].

DOI: 10.5176/2251-3043_2.4.216

B. Dance Choreography is an NP-Complete Problem

We now introduce the Dance Choreography Problem and show that it is NP-Complete. It is this very nature of the Dance Choreography Problem that makes the Exhaustive Algorithm impractical for long amalgamations. As mentioned above, the Greedy Algorithm offers a practical compromise.

Since the discovery of the first NP-Complete problem [3], many hundreds of such problems have been discovered, including that of Dance Choreography [8]. For example, the NP-Complete Hamiltonian Circuit Problem asks: Does a given graph have a simple cycle containing all its vertices? [18]

Similarly one version of the Dance Choreography Problem asks: Given a dance syllabus that defines a set of figures and their permissible combinations, does it allow for an amalgamation that contains figures from a compulsory subset? Formal definitions of the Hamiltonian Circuit problem and this version of the Dance Choreography problem follow.

HAMILTONIAN CIRCUIT

INSTANCE:
A graph G.

QUESTION:
Does G have a hamiltonian circuit (simple cycle containing all the vertices)?

DANCE CHOREOGRAPHY

INSTANCE:
A finite set *Figures*;
a set *Follow* of tuples $<f_i, f_j>, f_i, f_j \in Figures$;
$n \in \mathbf{Z}+$;
$f_{\text{start}}, f_{\text{end}} \in Figures$;
a set $Compulsory \subseteq Figures$.

QUESTION:
Is there a sequence $Amalgamation = <f_1, f_2, \ldots, f_n>$,
$f_i \in Figures$, $1 \le i \le n$, and $<f_j, f_{j+1}> \in Follow$, $1 \le j < n$
such that
$f_1 = f_{\text{start}}$ and $f_n = f_{\text{end}}$
and
$\{f_i : 1 \le i \le n\} \supseteq Compulsory$?

We can prove that the Dance Choreograph Problem is NP-Complete by using the technique of “restriction” to the Hamiltonian Circuit Problem, as formally stated below:

THEOREM 1: DANCE CHOREOGRAPHY is NP-Complete.

PROOF: Restrict to HAMILTONIAN CIRCUIT: Make $n = |Figures| + 1$, $Compulsory = Figures$, $f_{\text{start}} = f_{\text{end}}$.

The dance choreography problem as defined above can easily be extended and these extensions shown to be NP-Complete as well. For example, if we wish to include a further stipulation that the amalgamation must contain a fixed subsequence, we can define the following extension:

DANCE CHOREOGRAPHY 2

INSTANCE:
A finite set *Figures*;
a set *Follow* of tuples $<f_i, f_j>, f_i, f_j \in Figures$;
$n \in \mathbf{Z}+$;
$f_{\text{start}}, f_{\text{end}} \in Figures$;
a set $Compulsory \subseteq Figures$;
a sequence Fixed of figures $f_i \in Figures \cup \{any\}$, $1 \le i \le n$
where $any \notin Figures$.

QUESTION:
Is there a sequence $Amalgamation = <f_1, f_2, \ldots, f_n>$,
$f_i \in Figures$, $1 \le i \le n$, and $<f_j, f_{j+1}> \in Follow$,
$1 \le j < n$
such that
$\{f_i : 1 \le i \le n\} \supseteq Compulsory$
and
$f_i = Fixed_i$ if $Fixed_i \ne any$?

Having proven that DANCE CHOREOGRAPHY is NP-Complete, we can now prove that the extension problem DANCE CHOREOGRAPHY 2 is also NP-Complete by using the same restriction technique, now to DANCE CHOREOGRAPHY:

THEOREM 2: DANCE CHOREOGRAPHY 2 is NP-Complete.

PROOF: Restrict to DANCE CHOREOGRAPHY: Make $f_1 = f_n$ and $f_i = any$, $1 \ne i \ne n$.

Our proof process involved the HAMILTONIAN CIRCUIT (HC) problem which, historically, was proven to be NP-Complete by transformation from the VERTEX COVER (VC) problem. VC, in turn, was proven NP-Complete by transformation from the 3-SATISFIABILITY (3SAT) problem, and 3SAT was proven NP-Complete by transformation from SATISFIABILITY (SAT): the first NP-Complete problem discovered by Cook [3, 18]. Schematically, we can show where the Dance Choreography Problems fit in a partial NP-Completeness “proof hierarchy tree” (fig. 1).

$$\begin{tikzcd}
\text{SAT} \arrow[d, dash] \arrow[dr, dash] & \\
\text{3SAT} \arrow[d, dash] & \dots \\
\text{VC} \arrow[d, dash] & \\
\text{HC} \arrow[d, dash] & \\
\text{DANCE CHOREOGRAPHY} \arrow[d, dash] & \\
\text{DANCE CHOREOGRAPHY 2} \arrow[d, dash] & \\
\dots &
\end{tikzcd}$$

Figure 1. Partial NP-Completeness proof hierarchy tree showing the derivation of the Dance Choreography problems.

We have shown that any instance of the Hamiltonian Circuit Problem is a special case of the Dance Choreography Problem [8]. Consequently, the solution process for the Dance Choreography Problem will work exactly as for a similar instance of the Hamiltonian Circuit Problem. An instance of the Dance Choreography Problem is shown in fig. 2. The graph on the left represents a simplified hypothetical syllabus of four figures (Feather [Step], 3 Step, Reverse [Turn], and Natural [Turn]) with their permissible interactions (e.g., a Feather Step can be followed by either a 3 Step or a Reverse Turn, but not a Natural Turn). The graph on the right represents a solution (the dark arcs indicate a possible amalgamation, e.g., Feather Step, Reverse Turn, 3 Step, Natural Turn. Therefore, this particular syllabus does allow for an amalgamation that contains every one of its figures once and only once (a constraint imposed by making $n = |Figures| + 1$ and $Compulsory = Figures$).

Hypothetically, should we be able to solve the Dance Choreography Problem efficiently, i.e., using a polynomial time algorithm, we can also use it to solve the Hamiltonian Circuit Problem equally efficiently. The consequences are even more far reaching than this, in that we can then also solve the currently known hundreds of related NP-Complete problems, many of which are models of problems in mathematics, computer science, operations research, and industry in general [6, 7, 8, 9, 10, 11, 12]. Indeed, interest in dance choreography need not be confined to dancers, and it would be a pity if such research did not yield more practical copyrightable applications, such as the one we are about to describe.

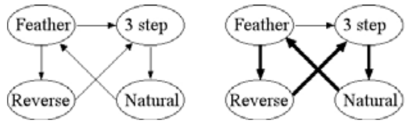


Figure 2. Solution to a Dance Choreography problem instance.

III. COMPUTERIZED DANCE TRAINING AND EDUCATION

With this theoretical basis of our design decisions, we have implemented several processes of dance choreography in our software we call Terpsichore©, named after the Greek muse of dance. Terpsichore© is a multimedia-based interactive computer software for dance training and education, copyrighted on July 1, 2011 (Registration Number TXu 1-762-479), and first demonstrated at the 85th Annual Meeting of the Louisiana Academy of Science, 2011 [7]. The copyright process of the software was presented at the 2012 National Academy of Inventors Inaugural Annual Conference [6].

We now describe the primary functions of Terpsichore© from the user's perspective. Its modules can be divided into two main categories: choreography generator / interactive trainer, and multimedia demonstrator.

A. *Terpsichore© as Choreographer*

Terpsichore© uses search algorithms that have been applied successfully in various computer science domains. It uses three types of search methods: the Exhaustive, Greedy, and Randomized searches. These searches are conducted upon a knowledge database comprising the established figures formalized by various International Standard dance associations: DVIDA (Dance Vision International Dance Association), IDTA (International Dance Teachers' Association), and ISTD (Imperial Society of Teachers of Dancing) [4, 16, 17]. To avoid any possibility of copyright infringement, we have created our own syllabus comprising these universal figures. Based on structural and pedagogical considerations, we have grouped these dance figures into three categories: "Foundational," "Intermediate," and "Advanced."

To assess proposed amalgamations during the search process, the program applies evaluation functions that use criteria such as variety and degree of difficulty. The search engine and knowledge database are internal components that are transparent to the user. More sophisticated evaluation functions using machine learning is a viable research goal.

When the user launches Terpsichore©, the following minimalist window appears (fig. 3). We feel that a first-time user would be best served by not being bombarded by an excess of choices presented in a complicated interface. Like most computer programs, Terpsichore© has a primary purpose: to create amalgamations, and it would be sufficient that just this functionality is first presented to the user. Other functions can be discovered later, through the drop-down menus and other windows that the user can choose to open. In this way, all intimidation is avoided, and the undaunted user is then encouraged to explore all that the software has to offer.

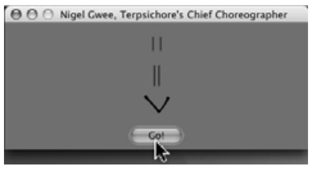


Figure 3. Commencement of the program.

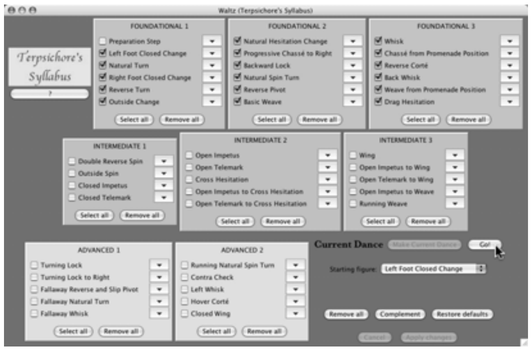
From this window, the user will most likely click the default button, or type the Return key, whereupon an (almost) instant amalgamation appears (fig. 6). Hopefully, the user will discover the various options open, subsequently arriving at the basic runtime process (figs. 4–6).

In this process, the user chooses a set of figures from which to create an amalgamation (fig. 4), and selects the search method (fig. 5). The user also decides the quantity and length of amalgamations (interface for these choices not shown here). From among the very numerous possible solutions, the program chooses the required number of amalgamations, ensuring a virtually unlimited variety of outputs across successive trials. Fig. 6 shows just one of the many amalgamations possible.

Of the three algorithms implemented, the Random Algorithm simply outputs randomly the required number of amalgamations. The results are not uniform in quality, a trade-off for its speed. In contrast, the Exhaustive Algorithm, as its name implies, searches all possible amalgamations and uses the evaluation functions to determine the best among them. As we have shown, the Dance Choreography Problem is NP-Complete, and only the processor speeds of modern hardware prevent the Exhaustive Algorithm from taking too long to produce its output, at least for relatively short amalgamations. The algorithm actually solves an NP-Hard problem, but for simplicity we do not elaborate on that here. A compromise between the unpredictable output of the Random Algorithm, and the expensive thoroughness of the Exhaustive Algorithm is the Greedy Algorithm, which undertakes a heuristics-based search that attempts to make smart choices while taking shortcuts. Encouragingly, it often, although not always, discovers amalgamations that are just as good as those from the Exhaustive Algorithm, but takes far less time to do so (see the next section for a comparison between these two algorithms).

Besides functioning as an amalgamation generator, Terpsichore© provides for various other interactive activities. The user can ask to create amalgamations, as shown above, as well as to help create user-defined amalgamations. The user is provided only the figures that could possibly fit in the respective part of the sequence (fig. 7). Program verification has shown that our software can reproduce all the amalgamations given in Moore, an acknowledged authority and pioneer in the codification of the International Ballroom Dance syllabus [20]. In another mode, users can test their knowledge of the syllabus by creating amalgamations without the benefit such guidance; in this instance Terpsichore© annotates the user's efforts, pointing out incorrect sequences.

We may summarize the capabilities of the software in a use-case diagram (fig. 8). The software enables the user to conduct three different types of search: Random, Thorough (using the Exhaustive Algorithm), and Quick (using the Greedy Algorithm; available in advanced user mode). The user is also able to create customized amalgamations: "Make my own" and "Test myself"—the former provides guided construction of amalgamations, the latter quizzes the user (advanced user mode). The multimedia capabilities provide aural and visual playback ("Play audio" and "Play video"), and the saving of files to disk ("Create audio" and "Create video").



FOUNDATIONAL 1
- ☐ Preparation Step
- ✓ Left Foot Closed Change
- ✓ Natural Turn
- ✓ Right Foot Closed Change
- ✓ Reverse Turn
- ✓ Outside Change

FOUNDATIONAL 2
- ✓ Natural Hesitation Change
- ✓ Progressive Chassé to Right
- ✓ Backward Lock
- ✓ Natural Spin Turn
- ✓ Reverse Pivot
- ✓ Basic Weave

FOUNDATIONAL 3
- ✓ Whisk
- ✓ Chassé from Promenade Position
- ✓ Reverse Corté
- ✓ Back Whisk
- ✓ Weave from Promenade Position
- ✓ Drag Hesitation

INTERMEDIATE 1
- ☐ Double Reverse Spin
- ☐ Outside Spin
- ☐ Closed Impetus
- ☐ Closed Telemark

INTERMEDIATE 2
- ☐ Open Impetus
- ☐ Open Telemark
- ☐ Cross Hesitation
- ☐ Open Impetus to Cross Hesitation
- ☐ Open Telemark to Cross Hesitation

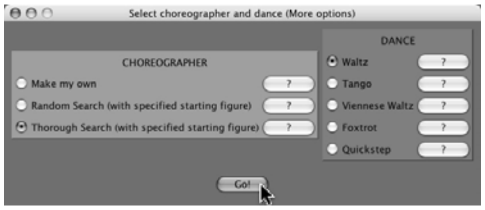
INTERMEDIATE 3
- ☐ Wing
- ☐ Open Impetus to Wing
- ☐ Open Telemark to Wing
- ☐ Open Impetus to Weave
- ☐ Running Weave

ADVANCED 1
- ☐ Turning Lock
- ☐ Turning Lock to Right
- ☐ Fallaway Reverse and Slip Pivot
- ☐ Fallaway Natural Turn
- ☐ Fallaway Whisk

ADVANCED 2
- ☐ Running Natural Spin Turn
- ☐ Contra Check
- ☐ Left Whisk
- ☐ Hover Corté
- ☐ Closed Wing

Starting figure: Left Foot Closed Change

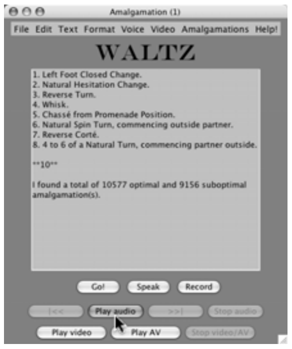
Figure 4. Amalgamation figures.



CHOREOGRAPHER
- ○ Make my own
- ○ Random Search (with specified starting figure)
- ● Thorough Search (with specified starting figure)

DANCE
- ● Waltz
- ○ Tango
- ○ Viennese Waltz
- ○ Foxtrot
- ○ Quickstep

Fig. 5. Search method (choreographer) and dances.

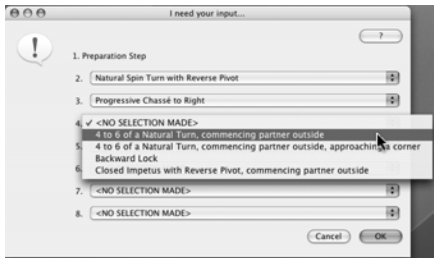


1. Left Foot Closed Change.
2. Natural Hesitation Change.
3. Reverse Turn.
4. Whisk.
5. Chassé from Promenade Position.
6. Natural Spin Turn, commencing outside partner.
7. Reverse Corté.
8. 4 to 6 of a Natural Turn, commencing partner outside.

10

I found a total of 10577 optimal and 9156 suboptimal amalgamation(s).

Figure 6. A Waltz amalgamation.



1. Preparation Step
2. Natural Spin Turn with Reverse Pivot
3. Progressive Chassé to Right
4. ✓ <NO SELECTION MADE>
   4 to 6 of a Natural Turn, commencing partner outside
   4 to 6 of a Natural Turn, commencing partner outside, approaching a corner
   Backward Lock
   Closed Impetus with Reverse Pivot, commencing partner outside
7. <NO SELECTION MADE>
8. <NO SELECTION MADE>

Figure 7. Creating user-defined amalgamations.

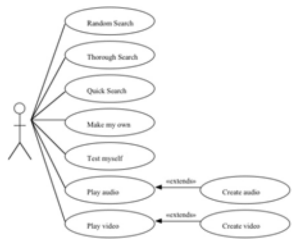


Figure 8. Use-case diagram showing the capabilities of Terpsichore©.

B. Comparing Algorithm Performance Using Terpsichore©

Using our software, we took the opportunity to test and compare the performances of the Exhaustive and Greedy algorithms. To eliminate futile searches, we implemented the Exhaustive Algorithm to incorporate tree pruning.

In the following experiment, we ran the Exhaustive and Greedy algorithms on a set of seventeen figures from the three Foundational categories of our proprietary syllabus, and ran the algorithms to produce amalgamations of between one and ten figures in length. The amalgamation optimality measure was the degree of variety of figures chosen: the fewer repetitions of figures, the better. Running time was measured in clock ticks (60 clock ticks = 1 second). The experiment was performed on a Macintosh PowerPC computer with dual 2.3 GHz G5 processors.

As expected, the Exhaustive Algorithm required exponential time relative to amalgamation length, whereas the Greedy Algorithm took almost constant time throughout. On the other hand, the Greedy Algorithm did not always produce optimal results, generating amalgamations with fitness values varying from 8.5/10 to 10/10 (Tables I and II).

TABLE I. PERFORMANCE OF THE EXHAUSTIVE ALGORITHM

length	time	optimal	suboptimal	total
1	1	1	0	1
2	1	10	0	10
3	2	24	1	25
4	8	83	14	97
5	36	314	77	391
6	156	1021	391	1412
7	643	3399	1905	5304
8	2666	10577	9156	19733
9	10911	32428	41369	73797
10	44432	96105	179945	276050

TABLE II. PERFORMANCE OF THE GREEDY ALGORITHM

length	time	fitness
1	<1	10
2	<1	10
3	<1	10
4	<1	8.8
5	<1	9
6	<1	10
7	1	10
8	1	9.4
9	1	9.4
10	1	8.5

C. Terpsichore© as Multimedia Dance Trainer

Terpsichore© acts as a dance trainer by providing several coaching modules: video presentations, displays of foot positions and alignments (fig. 9), and real-time delivery of amalgamations that comply with the timing and sequential requirements of the various figures in the recognized ballroom dance syllabi. These modules can be used in conjunction with a human coach or dance teacher by providing a practically inexhaustible variety of amalgamations, and a variety of presentation modes that are designed to retain student attention and interest.

Enhancements to its multimedia capabilities can take advantage of evolving computer graphics techniques. We have in mind exploiting the capabilities of 2D- and 3D-modeling, and the rendering of the generated choreography into Labanotation [21].

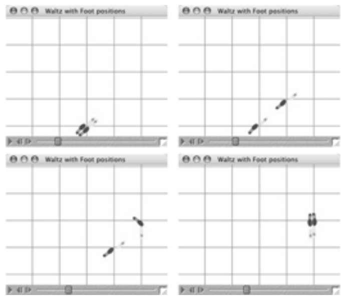

Waltz with Foot positions
Waltz with Foot positions
Waltz with Foot positions
Waltz with Foot positions

Figure 9. Screenshots of a video sequence showing foot positions.

D. Implementation and Development of the Software

Several programming languages were used to implement Terpsichore©, in order to exploit the unique capabilities of each of them. At the heart of the application is the search engine, implemented in C++ (ISO/IEC TR 19768:2007). For the graphical user interface we used Java (J2SE 5.0), exploiting its Swing components extensively. The Java Native Interface was the glue for the C++ and Java code. For the multimedia features we used Objective-C 2.0 and Object-C++ to take advantage of the Macintosh® OS X Operating System's advanced capabilities.

The software has been designed to facilitate enhancement of various aspects of its interface, search engine, and knowledge database to accommodate new syllabi figures and even new dance genres. Its software structure can also be easily adapted for related artistic endeavors, such as music composition. We do so by modifying the knowledge database to comprise, say, the rules of sixteenth-century musical counterpoint. We could also modify the search engine to use machine-learning algorithms, such as artificial neural networks and genetic algorithms in order to reflect human preferences. For applications of these algorithms to music composition, see [9, 10, 11, 12].

IV. Conclusion

We have shown that the Dance Choreography problem is NP-Complete and hence is closely related to many other significant problems in industry. In addition to exhaustive search algorithms, which are practical only in small instances of the problem, heuristic based algorithms are necessary. The implementation of these algorithms has led to our dance choreography software Terpsichore©. We showed how Terpsichore© not only creates vast numbers of quality amalgamations, but how it can also be configured to provide coaching and self-teaching capabilities.

The software was designed to be readily extensible. For example, once copyright clearance can be obtained with various dance instruction schools, more real-time videos can be incorporated. Realistic synthesized voices can also be easily added to provide a more natural experience for the user.

Future work on Terpsichore© will also include the addition of more realistic computer animation including 2D- and 3D-modeling featuring partner dancing, a uniquely difficult but not insurmountable task [15, 23]. More sophisticated evaluation functions and algorithms to measure aesthetics in the various amalgamations produced is another promising avenue of research.

Acknowledgment

We wish to thank Mr. William ("Bill") Davies, Dancesport Adjudicator and Examiner, and former North American and United States Ballroom Dance Champion, for invaluable artistic advice. We are also grateful to Attorney Frances Ball for legal assistance during the copyright registration process of this software.

References

[1] N. I. Badler and S. W. Smoliar, "Digital representations of human movement," Computing surveys, vol. 11, no. 1, pp. 19–38, 1979.

[2] T. Calvert, L. Wilke, R. Ryman, and I. Fox, "Applications of computers to dance," IEEE Computer Graphics and Applications, vol. 25, no. 2, pp. 6–12, 2005.

[3] S. A. Cook, "The complexity of theorem-proving procedures," in Proc. 3rd Annual ACM Symposium on Theory of Computing, Shaker Heights, Ohio: Association for Computing Machinery, 1971, pp. 151–158.

[4] Dance Vision International Dance Association, Bronze (Silver/Gold) Level International Style Standard Manual, Las Vegas, Nevada: Dance Vision, 2008.

[5] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, 20th printing. New York: Freeman, 1999.

[6] N. Gwee, "Copyrighting Terpsichore©: Multimedia dance software," National Academy of Inventors Inaugural Annual Conference, Tampa, Florida, 2012.

[7] N. Gwee, "Implementation of an NP-Complete dance choreography problem," Louisiana Academy of Science 85th Annual Meeting, University of Louisiana at Monroe, Monroe, Louisiana, 2011.

[8] N. Gwee, "The theory of NP-Completeness: Applications and implications," Louisiana Academy of Science 83rd Annual Meeting, Southeastern Louisiana University. Hammond, Louisiana, 2009.

[9] N. Gwee, "Feasibility of genetic algorithms for music composition," Proc. International Conference on Artificial Intelligence, Las Vegas, Nevada, vol. I., pp. 428–433, 2003.

[10] N. Gwee, "Composing species counterpoint with genetic algorithms," Proc. 41st ACM-SE Conference, Savannah, Georgia, pp. 235–240, 2003.

[11] N. Gwee, "Effective heuristics for algorithmic music composition," Proc. International Conference on Artificial intelligence, Las Vegas, Nevada, vol. III, pp. 1027–1033, 2002.

[12] N. Gwee, "Capturing human musical preferences with fuzzy application of rules," Proc. 40th ACM-SE Conference, Raleigh, North Carolina, pp. 139–146. 2002.

[13] N. Gwee, "An NP-complete music problem, " Proc. 39th Annual ACM-SE Conference, Athens, Georgia, pp. 184–190, 2001.

[14] N. L. Hall and D. Herbison-Evans, "BALLONES: A ballet animation language," Proc. Australian Computer Graphics Association 1990 Conference, 1990.

[15] D. Herbison-Evans, R. D. Green, and A. Butt, "Computer animation with NUDES in dance and physical education," Australian Computer Science Communications, vol. 4, no. 1, pp. 324–331, 1982.

[16] G. Howard, Technique of Ballroom Dancing, Brighton, England: KenadS Design & Print, 2007.

[17] Imperial Society of Teachers of Dancing, The Ballroom Technique, London, England: Lithoflow Ltd, 1994.

[18] R. M. Karp, "Reducibility among combinatorial problems," In Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, pp. 85–103, 1972.

[19] J. Lansdowne, "The computer and choreography," IEEE Computer, vol. 11, no. 8, pp. 19–30, 1978.

[20] A. Moore. Ballroom Dancing, 10th edition. London: A & C Black, 2002.

[21] M. Nakamura, "Text representation of Labanotation data for computer based motion analysis," World Dance Association/Council of Kinetography Laban/Congress on Research in Dance International Conference, 2004.

[22] A. M. Noll, "Choreography and computers," Dance Magazine, pp. 43–45, January1967.

[23] D. Sealey, "Notate: Computerized programs for Labanotation," Journal for the Anthropological Study of Human Movement, vol. 1, no. 2, pp. 70–74, 1982.

[24] T. Yu and P. Johnson, "Tour Jeté, Pirouette: Dance choreographing by computers," YLEM Journal: Artists Using Science and Technology, vol. 23, no. 6, pp. 8–10, 2003.

Nigel Gwee is an associate professor of computer science at Southern University, Baton Rouge, Louisiana. He received his Ph.D. in musicology and his Ph.D. in computer science from Louisiana State University, Baton Rouge. He has written papers and computer programs on music and dance. Recently, he won the Best Research Paper Award at the Third International Conference on Software Engineering & Applications, 2012, Singapore.