Two Musical CSPs

Marc Chemillier, Charlotte Truchet
GREYC, Université de Caen, France
Equipe Représentation Musicales, IRCAM and LIP6, Université de Paris 6, Paris, France
marc@info.unicaen.fr and Charlotte.Truchet@ircam.fr

We present two musical problems which are interesting examples of musical CPs. The first one deals with harp music from Nzakara people of Central African Republic, where canon structures have been discovered. The computation of such structures is a critical problem for constraint solvers in the sense that there are no exact solutions, the only possible way to solve the problem being to accept approximate solutions. The second problem deals with the analysis of Ligeti's textures. It is a kind of variation on the classical "choral harmonization problem" which is to some extent adapted to the atonal music of Ligeti. But unlike the choral harmonization problem where a great amount of solutions can be found, the Ligeti's problem is critical in the sense that it has only very few solutions. For both these real musical problems, we mainly use an adaptive solver belonging to the family of local search methods and developed at Ircam, which seems to be quite efficient. For a description on the solver, see (Codognet 2000) or (Truchet, Assayag, Codognet 2001). It works by successive improvement of an initial configuration. For this algorithm, a constraint C is written as a cost-function f_C, with $f_C(V) = 0 \Leftrightarrow C(V)$ for an instanciation V, and the algorithm minimizes the sum of these cost-functions. Thus, it allows to deal with approached solutions and display the intermediate solutions if needed.

1. Nzakara harp

This problem has been formulated by Marc Chemillier (Chemillier 1995). One of its interesting properties is that it has no exact solutions because of two conflicting constraints. The goal is to construct a sequence of n bichords, each bichord in a domain of size 5. In midi values, the allowed bichords are (60,64), (60,67), (62,67), (62,70), (64,70). The first constraint C1 states that the lower voice of the sequence reproduces the upper voice (apart from a transposition), with a time gap of p (n and p are fixed integer). Formally, we have a transposition-like function which maps 60 to 64, 62 to 67, and 64 to 67. For every k < n, the lower note of the (k+p)-th bichord is the transposition of the upper note of the k-th bichord. The second constraint C2 is to avoid trivial sequences such as (a a a…) or (a b a b a b...). It can be written as bichord (k+p) $\neq$ bichord k and bichord (k+1) $\neq$ bichord k (the Nzakara never repeat a bichord). Notice that the two constraints are antagonistic. An example is shown figure 1. The five strings of the harp are represented graphically as horizontal lines, and mapped to the midi codes 60, 62, 64, 67, 70 (approximately a scale played by Nzakara musicians).

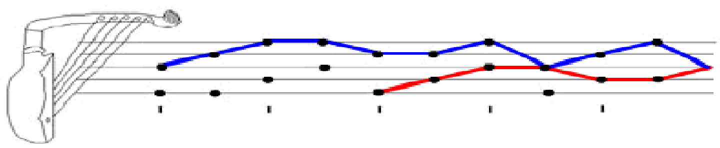

Figure 1. The lower voice reproduces the upper voice with a time gap and a translation (n = 10, p = 4).

In the original Nzakara harp repertoire, one can find different values for integers n and p (respectively the total length of the repeated sequence and the distance of the canon). Sequences with n = 10, p = 4 (as in figure 1) and n = 20, p = 4 belong to the category called *ngbakia. S*equences with n = 30, p = 6 belong to the category called *limanza* (figure 2 below). These sequences are played as *ostinato*, each piece of Nzakara poetry being sung with the accompagniment of such formula played on the harp. The categories *ngbakia* and *limanza* also refer to traditional dances, the harp formulas being adapted from rhythms and musical elements borrowed from the dance-repertoire played on the portable xylophone or the drum. Some of these sequences can heard on the two CDs mentionned in the references below.

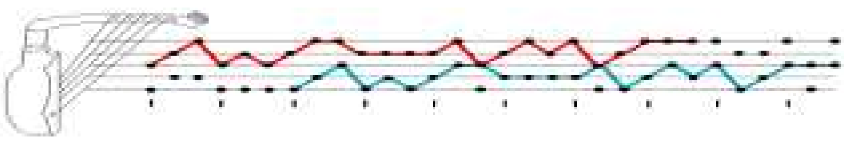

Figure 2. A *limanza* sequence with n = 30, p = 6.

There are some errors regarding the canon structure in the sequence represented on figure 2, a few bichords having their lower note outside the lower broken line (exactly 6 bichords). As it has been said previously, there is no way to avoid errors because of incompatible constraints. By formalizing the construction of Nzakara canons, one can calculate the minimal number of such errors. As shown on figure 3 below, couples of strings plucked simultaneously are placed in a very regular manner. Bichords separated by p positions always appear in the same order A B C D E, the lower voice reproducing the melodic ascending motion of the upper voice with a shift of one position, excepted at the end of the sequence where it goes back to the beginning and breaks the ordered structure. At this ending point, there is exactly one error regarding the canon constraint. This construction leads us to a property which can be stated as follows : *the number of errors in a Nzakara canon is at least gcd(n,p) where n is the length of the sequence, and p the distance of the canon.*

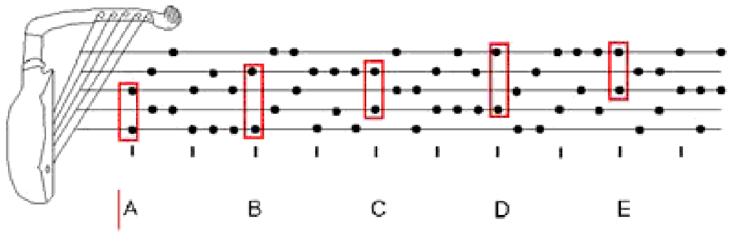

Figure 3. Ordered five bichord sequence.

An algorithm is described in (Chemillier 1995) to compute all the canons constructed with the minimal number of errors. In the case n = 10, p = 4, a surprising result is obtained, as there is only one solution with the minimal number of error (which is this case is two). This unique solution is the Nzakara sequence which was shown on figure 1. Another sequence with values n = 10, p = 4 can be obtained with only two errors in the canon structure. But this sequence is "degenerated" in the sense that it is a repetition of two equal shorter subsequences. Both of these sequences (the Nzakara one and the degenerated one) are computed by the adaptive solver, as shown on figure 4. The computation time is very short (0,3 seconds on a Mac G4 on an average of ten runs).

Figure 4. Two sequences with two errors (n = 10, p = 4).

The interest of adaptive search in this case is thus to get approximate solutions. But this CSP shows also the utility of weighing the constraints. It is possible, and easy, to play with the balance of the constraints since they are only two and they are antagonistic (one can find a solution with C1, or with C2, but not with both). It is sort of a deal : how much do you want of C1, how much are you ready to release on C2? This example shows the advantage of adaptive search for over-constrained CSP. Experimentally, the adaptive solver finds good approximate solutions. Figure 5 shows a resolution with n = 30, p = 6, and zero errors for C2, seven for C1. The Nzakara make 6 errors on C1 with these values, as we have seen before.

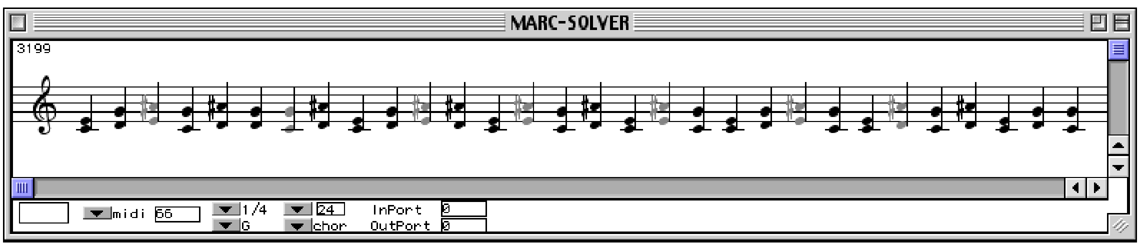

Figure 5. Light gray shows bad bichords with regard to C1.

Experimentally, we observed that the two sequences with minimal number of errors shown on figure 4 do not appear with the same frequency in the adaptive solver research. Both sequences could have had equal probabilities, but this is not the case. The following curve represents the frequency of apparition of the Nzakara solution during about 500 successive computations. The slope of this curve is only 0.32, which is less than the slope 1/2 of the upper line corresponding to equal probabilities. This experimental result seems not connected to algorithmic aspects, but may be related to the fact that the two possible sequences have not the same number of circular permutations. Since the degenerated sequence is made of two sublists, it appears twice as much as the Nzakara one. This shows that the representation with cost-functions that we've chosen is unbiased.

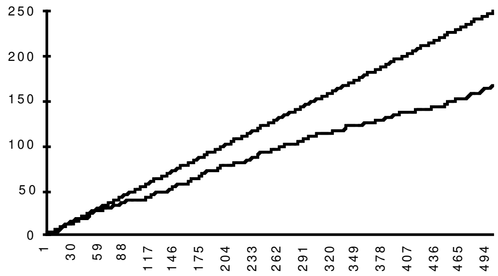

Figure 6. Frequency of the apparition of Nzakara solution during adaptive search.

We have also tried a modelisation in Sicstus Prolog. Since we know there are no solutions, it is necessary to find a way to get answers anyway. We need some predicates to define what is a bichord (conso/2) and what is a transposition (transpo/2). Since the Nzakara always introduce the same kind of error, we just need to add transpo(60,70). With this modelisation, the Nzakara solution is obtained after a half an hour calculation.

2. Ligeti's textures

The second problem is the analysis of Ligeti's textures. Given a sequence of n aggregates A_i with less than k notes each, try to find a k voice polyphony denotated as an array X_{ij} so that :

1) each A_i is included in the corresponding row X_i,

2) voices which are not playing any note from the corresponding aggregate must stay on the same pitch $X_{ij} = X_{(i-1)j}$,

3) melodic motions in each voice are limited to the following intervals : -2, -1, 0 or 1, which means that $X_{ij} - X_{(i-1)j}$ belongs to the set {-2,-1,0,1}.

This problem is related to many Ligeti's works which are called "pattern-meccanico" (Clendinning 1993). The aggregates A_i are played as short motives quickly repeated in a mechanical fashion, such as in the piece *Continuum* for harpsichord. We will concentrate here on a texture taken from the beginning of the orchestral piece *Melodien*. In this piece, k has value 10. The following graphical representation (figure 7) displays the notes of the aggregates as black squares, and the analysis as ten broken lines passing through these notes. The midi codes of the notes are ranging from 66 to 93.

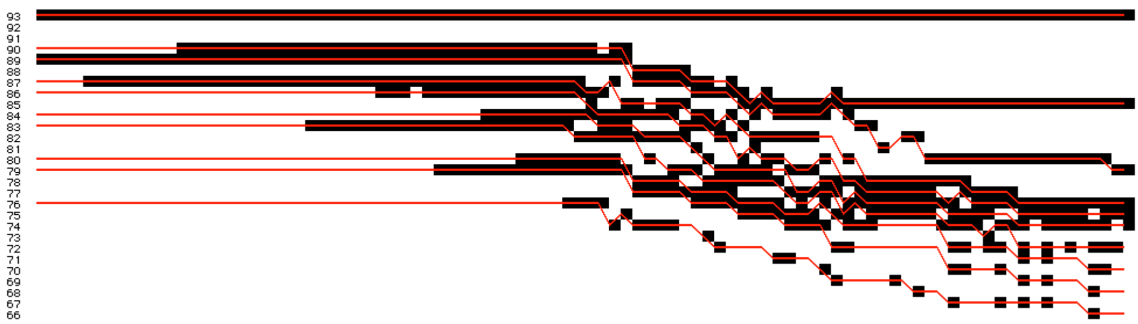

Figure 7. Texture from *Melodien* and its ten line analysis (mes. 14 to 30).

This problem is closely related to the choral harmonization problem. The melodic choral line is replaced by a sequence of aggregates, and the four part polyphony is replaced by a ten line polyphony. But the constraints are very similar in the way they formalize restrictions concerning the melodic motions. Of course, the four parts of the choral polyphony are real parts which are supposed to be sung, whereas in the texture analysis, the ten line polyphony is not supposed to be played. It is just an implicit structure expressing the logic of the texture. From a constraint programming point of view, there is another important difference. The choral harmonization problem is known to have many solutions (Dellacherie 1998), so that it takes not too many time to compute the first solution. Furthermore, it is possible to deal with musical abstract objects such as harmonic degrees. This allows us to divide the search process into several phases and to reduce the number of variables in each phase (Pachet 2001). On the contrary, the texture analysis is a critical problem, because it has very few solutions, and a lot of variables, as we shall see.

Let's first describe the modelisation to get a solution from line to line (i.d. from row to row in ten voice polyphonic array). The departure line is supposed to be already completed and will be written $D = (D_1 ... D_{10})$. The arrival line may not be complete and will be written $A = (A_1 ... A_k)$. The variables are $X_1 ... X_{10}$, they represent the full set of midi values corresponding to the arrival. There are four constraints :

C1 = $A \subset X$

C2 = $\forall i, X_i < X_{i+1}$

C3 = if $X_i \notin A$, then $X_i = D_i$

C4 = $X_i \in \{D_i - 2, D_i - 1, D_i, D_i + 1\}$

The last constraint can easily be reduced by a restriction of the domains. So we take, as domain for Xi, $Dom_i = \{D_i - 2, D_i - 1, D_i, D_i + 1\}$. The other constraints are transformed in cost-functions as follows.

$f_1(i, X) = k - \#\{j \in [1,10], X_j \notin A\}$

Note that this function is independent from i. This represents well the constraint, which concerns the whole set of variables. It doesn't disturb the resolution, since f_1 will still be minimized in the global cost. Anyway, we could have chosen to add it not in the costs of each variables but in the global cost.

$f_2(i, X) = \sum_{j<i} \max(0, 1 + X_j - X_i) + \sum_{j<i} \max(0, 1 + X_i - X_j)$

This function counts nothing if X_j is well ordered with regard to X_i, and the distance between X_i and X_j if not.

$f_3 =$ if $X_i \in A$ then 0 else $| X_i - D_i |$

Then we have many ways to represent the whole problem. The first and simplest one would be to take 49*10 variables and try to solve a CSP with around 500 variables. To reduce the number of variables, we could rely on the fact that we know how many notes are hidden (83 in *Melodien*). We could then write the constraints only on those variables. This is not satisfying, because it doesn't allow to verify the process on every line and for every note. We have chosen an half-way possibility. It is very easy to solve the CSP from one line to another : 10 variables, little domains (size 4 if we reduce C4 as a restriction of domains). So we can iteratively solve from line i to line i+1, giving the solution of the line i as a departure for line i+1. From a mere CSP point of view, we loose information doing this, because of the domain reduction. But it doesn't matter since we don't perform an exhaustive search. Adding a maximum number of iterations to the algorithm, we could decide that a search has failed for line i and backtrack to find another solution for line i-1. Anyway we don't need to do so because the algorithm performs well enough. It also allows to check the validity of the process for every note in the score : even in the case where there is no surprise, if lines i and i+1 have 10 notes, finding a solution means that lines i and i+1 verify the iterative process described above. Experimentally, the calculation time is very stable, around 7 seconds on a Mac G4 in OpenMusic.

A specific algorithm described in (Chemillier 1999) has proved that the analysis of the texture has only six solutions. More precisely, there are three possible transitions from A_{21} to A_{22}, and two possible transitions from A_{36} to A_{37}. The latter corresponds to the following midi values (where notes outside the A_i are marked with parenthesis) :

A_{36} = ((67) (70) 72 73 75 76 77 80 85 93)
A_{37} = (67 70 72 74 (75) 76 77 80 85 93)

A_{36} = ((67) (70) 72 73 75 76 77 80 85 93)
A_{37} = (67 70 72 (73) 74 76 77 80 85 93)

and these two solutions are represented on figure 8 from A_{33} to A_{39}.

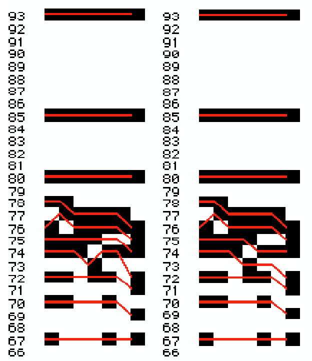

Figure 8. Two solutions from A_{33} to A_{39}.

Once again, we observed experimentally that the two solutions do not have equal probabilities. The graphical representation shown on figure 9 proves that the first solution on figure 8 appears more frequently during the adaptive search process.

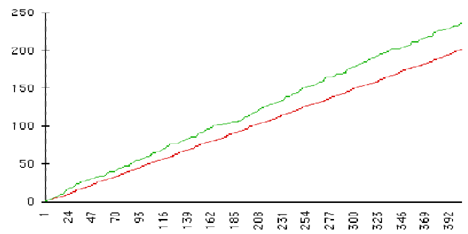

Figure 9. Frequency of the apparition of the first solution during adaptive search.

3. Conclusion

We have presented two examples of musical CSPs. Both of them are non trivial and linked to musical analysis. The first one is interesting because it has no solution, which is very often the case in Computer Assisted Composition, and thus a good example of musical CSPs. It requires to be solved either with some modification, or by an algorithm which deals with approximate solution.

The second one has mainly two features : much of the solution is already given in the problem itself (when the line A_i is already full), so we just need to check it, and it can be reduced in some way without loss of generality, allowing to reduce drastically the problem's complexity. A second feature of the problem is that the ten voices of the polyphony are independant (excepted the fact that they do not cross each other). In a certain sense, it is pure atonal counterpoint. According to the harmony vs. counterpoint classification, one could say that there is no harmony here. More precisely, there is no opportunity to use intermediate variables such as harmonic degres, in order to reduce the complexity of the problem (as it can be done in the classic "choral harmonization problem").

4. References

G. Assayag, CAO vers la partition potentielle, *Les cahiers de l'Ircam, Recherche et Musique,* **3** (1993) 13-41.

J.P. Clendinning, "The Pattern-Meccanico Compositions of György Ligeti", *Perspectives of New Music* 31 (1:) (1993) 192-234.

M. Chemillier, La musique de la harpe, E. de Dampierre (ed.), *Une esthétique perdue* (Paris, Presses de l'Ecole Normale Supérieure, 1995) 99-208.

M. Chemillier, Générateurs musicaux et singularités, *JIM 99, 6èmes journées d'informatique musicale* (CNET-CEMAMu, Issy-les-Moulineaux, 1999) 167-177.

M. Chemillier, György Ligeti et la logique des textures, *Analyse musicale* 38 (février 2001) 75-85.

M. Chemillier, Ethnomusicology, Ethnomathematics. The Logic Underlying Orally Transmitted Artistic Practices, Fourth Diderot Forum, European Mathematical Society (Springer Verlag, Berlin, to appear 2002).

P. Codognet, Adaptive Search, Preliminary Results, *BOOK of the 4th ERCIM/CompulogNet Workshop*, 2000.

P. Dellacherie, Modélisation et optimisation d'un harmoniseur automatique de chorals en PLC(FD), mémoire de DEA, université de Caen, 1998.

K. Ebcioglu, An expert system for harmonizing four-part chorals, *Computer Music Journal,* **12** (3), 1988.

F. Pachet (éd.), Constraints techniques for artistic applications, Workshop ECAI-98, Brighton, 1998.

F. Pachet, P. Roy, Musical Harmonization with Constraints: a Survey, *Constraints*, 6, 2001, 7-19.

C. Truchet, G. Assayag, P. Codognet, Visual and Adaptive Constraint Programming in Music, *Proceedings of the ICMC,* La Havane, 2001.

Audio CDs

Central African République. Music of the former Bandia courts, recordings, texts and photographs by M. Chemillier & E. de Dampierre, Paris, CNRS/Musée de l'Homme, Le Chant du Monde, CNR 2741009, 1996.

La parole du fleuve. Harpes d'Afrique Centrale, Paris, Cité de la musique, CM001, 1999.