

Geraint Wiggins, The User of Constraint Systems for Musical Composition  
Shane Hoversten

This paper is something of a hodgepodge, kind of like going to a restaurant and getting foie gras for an appetizer, some chicken fingers for the main meal, and a bucket full of sugar mixed with lard for dessert.

The paper as a whole purports to be about using constraint satisfaction in music composition. The first section describes using genetic algorithms as an example of this process. This is a strange formulation of GAs, as they are not usually thought of in the same realm as constraint satisfaction algorithms. The author makes a reasonable point, though, in claiming that the fitness function, by which the genomes produced by the GAs live or die, acts as a sort of constraint, or system of constraints. Chromosomes that don't satisfy the constraint don't survive. That seems to be twisting the arm of what a GA is – I suspect one could make anything appear as a type of constraint satisfaction using similar reasoning – but ok.

Wiggins points out that GAs aren't well suited to the four part harmonization tasks to which he applies them, because the search space is too large. Particularly, the problem is that when the GAs crossover, or mutate, the interdependencies of the domain are such that local changes to the genome are likely to screw up larger-scale dependencies. Since the four part harmonization features various structural dependencies between the parts of the composition, the GA (after a point, at least) is just as likely to screw the composition up as improve it, even if it satisfies the fitness function.

This is an interesting problem in general, and stems, I think, from a weakness in the formulation of the GA algorithm itself. The idea is simple and intuitive, but doesn't capture certain properties of how biological organisms actively evolve; there's a reason that both giraffes and mice have the same number of cervical vertebrae, and that reason is because evolution works on the modular level – genotypes for modules come into being, prove evolutionarily useful, and propagate, sometimes with fine-tuning, sometimes not. The nuances of how this happens are not well understood; I suppose it's even possible that a GA might find the same solution, were the genome formulated properly, and were it given ten billion years to run on trillions of computers. But practically speaking, the simple combine-and-mutate is going to be working at a very low level of abstraction, and, as Wiggins points out, it's going to choke on structural dependencies.

Wiggins then talks about using an actual constraint system, in a PROLOG variant, on a compositional project. Considering that the paper is ostensibly about constraint satisfaction this section is remarkably brief, and the explanation of the constraint satisfaction part remarkably shallow. First, he detours and talks about the development, by Schoenberg, of serial composition, which is in opposition to the older and more popular tonal composition. A series is “a twelve note chromatic sequence in which no note is repeated.” That definition under our belts, Wiggins talks about the constraining operations that can be performed on a series: inversion, retrograde, and retrograde inversion. His particular compositional goal is to write a piece that uses four instruments,

begins in G, with the original series played with a flute, the retrograde with oboe, the rotated-4 original in cello, and the rotated-8 retrograde in harp. He plugs all of these things into a SICStus Prolog 3.5, lets it crank, and it spits out some answers, from which the author selects one. Afterwards, he concludes that this technique seems to work pretty well.

The final section seems to have nothing to do with the first two, and is about the choice of representation in computer-aided music. I didn't understand any of this at a level sufficient to say anything intelligent about it, so I won't even try.

What comes to mind after all of this is that it seems that the fundamental problem in these compositional tasks is in dealing with the inter-relationships between different sections of the music. These non-local interactions are what make the problem both hard and interesting. I'm unclear as to how complicated of structural dependencies are practical to model with this method