

# Oito Reflexões sobre a Criatividade na Composição Auxiliada por Computadores

Rodolfo Coelho de Souza<sup>1</sup>, Regis Rossi Alves Faria<sup>1</sup>

<sup>1</sup> Departamento de Música – Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto  
– Universidade de São Paulo

Av. Bandeirantes, 3900 – 14040-030 – Ribeirão Preto – SP – Brazil

rcoelho@usp.br, regis@lsi.usp.br

**Resumo.** *Este artigo reflete sobre uma variedade de tópicos relativos à criatividade na composição algorítmica, tomando o partido de que os algoritmos atuais não exibem genuína criatividade, embora sejam ferramentas auxiliares eficientes para a composição musical. Entre os tópicos abordados estão: a lógica do pensamento musical, a barreira da significação musical e a promessa dos modelos conexionistas. Propõe-se ainda uma taxonomia de dois grandes grupos de algoritmos: um baseado na Ars Combinatoria, no acaso e nas regras de formação generalizadas de peças-modelo; e outro baseado em modelos genéticos que empregam recursividade. Estuda-se ainda a peça Metrópolis, composta algorítmicamente, que emprega uma mistura de princípios desses dois grupos.*

**Palavras-chave:** *criatividade musical, lógica musical, taxonomia de algoritmos, composição algorítmica*

**Abstract.** *This article deals with a variety of topics related to creativity in algorithmic composition, embracing the view that current algorithms are not able to display true creativity, although they may be very efficient auxiliary tools in musical composition. Among the approached topics are: the logic of the musical thought, the obstacle of musical meaning and the promises of connectionist models. It is proposed yet the taxonomy of two large groups of algorithms: one based in the Ars Combinatoria, chance and rule-based models after referential pieces; and another based in genetic models that use recursive procedures. An algorithmically composed piece named Metropolis that employs a mixture of principles belonging to the two groups is analyzed.*

**Key-words:** *musical creativity, musical logic, taxonomy of algorithms, algorithmic composition*

## 1. Primeira Reflexão

Mais de quarenta após a escrita de um primeiro programa de computador cujo propósito era compor música e após diversos outros projetos nesse campo permanece-se convencido de que o computador é uma extraordinária ferramenta para auxiliar o compositor a expandir sua capacidade criativa. Entretanto também reconhece-se suas limitações fundamentais quando se trata de simular o pensamento criativo humano e neste momento curvamo-nos à evidência de que nenhum programa feito para os computadores que temos hoje exibe de fato criatividade intrínseca. Toda criatividade

que eventualmente nos pareça emergir nos resultados do processamento de um programa para composição musical decorre inevitavelmente da criatividade da mente do programador.

Embora conscientes de que uma parte significativa da comunidade da computação musical diverge dessa avaliação, cremos, porém, que faríamos má ciência aderindo a um pensamento apenas porque ele é dominante ou simplesmente deixando-se levar pelo fascínio do potencial de suas implicações, sem atentar para os fatos objetivos.

## 2. Segunda Reflexão

Um dos mais destacados defensores da criatividade intrínseca dos programas de composição musical é o americano David Cope. Ele nos provoca argumentando que “se os humanos não forem capazes de criar programas de máquina que sejam criativos, então os humanos não seriam de fato muito criativos” [Cope 2005, p.6].

Este é obviamente um sofisma. Não se trata de afirmar que os humanos são incapazes de conferir criatividade a um programa, mas sim que a tecnologia hoje disponível ainda não nos permite simular as características fundamentais do pensamento criativo. Pode-se responder a Cope com outro tipo de evidência: se já somos capazes de conferir criatividade a programas de computador por que não estamos usando essa capacidade computacional para resolver os inúmeros problemas da humanidade que ninguém ainda foi capaz de solucionar por falta de uma “idéia luminosa”? Por que, por exemplo, nenhum programa jamais foi capaz de abordar algum problema do mundo físico *propondo* uma equação que representasse o fenômeno?

Por outro lado os computadores são fenomenais quando se trata de validar uma equação proposta pelo programador. Nenhum pesquisador dos dias de hoje dispensaria em sua consciência o auxílio de um computador. Entretanto nunca se viu um computador conceber uma nova teoria. Vemo-los, sim, ajudar a implementar as teorias que propomos. Aliás, nem a própria validação ou refutação de uma teoria os computadores são capazes de fazer. A interpretação dos dados processados depende sempre de decisões humanas *ex-machina*, ainda que elas possam já estar embutidas no algoritmo.

## 3. Terceira Reflexão

Há certos problemas que nos dificultam a simulação das características do pensamento criativo humano. O primeiro problema é que o pensamento criativo não é inteiramente lógico. O pensamento criativo tira partido tanto do acerto quanto do erro.

Um segundo ponto é que o pensamento criativo freqüentemente é não-discreto e analógico, ou seja, opera no mesmo âmbito dos processamentos cerebrais com os quais inventamos nossas metáforas e fazemos comparações. Pinker (2008, p.293) observa que “o poder da analogia não vem da observação de uma mera semelhança nos componentes. Ela vem da observação de *relações*<sup>1</sup> entre as partes, mesmo que as partes em si sejam bem diferentes.” Esse é o problema fundamental do limite do pensamento computacional possível com a tecnologia atualmente disponível: os computadores são,

---

<sup>1</sup> Itálico dos autores.

de fato, capazes de comparar dois componentes, entretanto não são capazes de *abstrair a relação de similaridade* entre eles, pois operam sobre amostras de uma realidade. A criatividade implícita nos algoritmos de composição musical esbarra nesse limite. A partir de um modelo de comparação definido a priori é possível determinar algoritmicamente o grau de similaridade entre dois eventos musicais (Wiggins e Allan, 2007). Fazendo a engenharia reversa podemos produzir eventos semelhantes a um evento dado, porém não sabemos (ainda) construir um algoritmo que possa definir o próprio modelo de similaridade a ser adotado na comparação. É isso que a mente humana faz quando é considerada criativa.

Por exemplo, sabemos que é possível programar um computador para identificar semelhanças entre duas impressões digitais. Entretanto é preciso informar ao computador o critério de semelhança necessário e suficiente para se identificar uma pessoa através da comparação de digitais. Em outras palavras, o computador *não resolve o problema* da identificação. Ele apenas repete, com maior velocidade e manipulando bancos de dados maiores, o algoritmo de busca que no passado era processado manualmente. A *solução do problema* depende da fixação do critério que determina qual a semelhança suficiente para que a identificação seja correta e, em outro nível, da própria formulação da hipótese de que uma mera impressão digital seja suficiente para identificar uma pessoa inteira. Nessa metáfora, ou mais precisamente, nessa sinédoque (da parte pelo todo) é que está o ato criativo da solução do problema da identificação.

Mas quando se trata da tarefa repetitiva de comparar uma determinada impressão com outras armazenadas num banco de dados, o computador de fato dá conta desse processo com incomparável competência.

#### **4. Quarta Reflexão**

Turner destaca outro ponto essencial do nosso problema: “todos reconhecemos que soluções criativas devam ser originais [...] mas as diferenças também devem ser *significativas*<sup>2</sup>. [...] A *novidade significativa* distingue uma solução criativa de outras que são apenas adaptações de soluções antigas” [Turner apud Cope 2005, p.9].

Essa colocação de Turner se projeta em nosso caso no questionamento do nível operacional da linguagem computacional. Sabemos que as linguagens de computador operam sobre duas classes de signos (segundo a classificação de C. S. Peirce) que são fundamentais para qualquer linguagem, seja ela verbal, imagética ou sonora: o ícone e o índice. Um programa de computador é capaz de comparar duas cadeias de dígitos binários, e assim simular qualquer operação cognitiva que reflita aspectos de iconicidade de uma linguagem, ou seja, questões de semelhança. A comparação de imagens de impressões digitais, como mencionado acima, é um exemplo disso. Mais idiomático ainda para o computador é associar um índice a uma cadeia de dígitos, permitindo assim simular operações cognitivas de natureza indicial. A identificação da pessoa portadora da impressão digital num banco de dados é um exemplo disso.

---

<sup>2</sup> Itálico dos autores.

Todavia o computador não é capaz de realizar operações genuinamente simbólicas, porque símbolos requerem capacidade não só de associação mas também de generalização. Uma pessoa poderia olhar um arquivo de impressões digitais e formular uma hipótese sobre como diferenciar as impressões de homens e mulheres. Talvez a resposta pudesse estar nas diferenças de tamanho das impressões ou numa alteração peculiar no formato da impressão causada pelo uso de unhas freqüentemente mais longas nas mulheres. As diferenças que apontamos podem se tornar *significativas* para a mente humana, mas não para um computador, que não formula hipóteses nem “pensa” simbolicamente.

Ora a música, como qualquer outra linguagem, tem todos os três níveis operacionais acima mencionados. Por isso um computador consegue auxiliar o compositor nos níveis icônicos e indiciais da linguagem musical, mas não no raciocínio simbólico, a não ser sobre classes menores e limitadas que tenham sido reduzidas a regras binárias. Entretanto toda organização formal do discurso musical – motivos, fraseologia, morfologia – pertence ao campo do simbólico, assim como todos os processos de significação musical – referências, intertextualidade, tópicos, etc – e sem eles a música não alcança sua completude como linguagem. Mas isso, é claro, depende de estarmos adotando uma concepção quiçá conservadora sobre o que é música. Se aceitarmos uma definição aberta, ao estilo de John Cage, de que qualquer fenômeno sonoro pode ser entendido como música, então o computador seria sim capaz de realizar composições acabadas e qualquer programa poderia ser criativo. Mas aí também se esgarçarão todos os critérios de valoração das obras musicais e nosso problema se dissolve no ar: compor algoritmicamente porquê e para quê, se qualquer sucessão de sons ambientais for também considerada música?

Encontra-se em Gareth Loy uma reflexão acurada sobre esse problema. Diz ele:

Os argumentos expostos sugerem que é uma qualidade reflexiva baseada na significação que distingue um ato criativo de outro meramente gerativo. O significado comunicado por um sistema formal é, essencialmente, ele mesmo. O significado comunicado por um ato criativo situa-se para além do produto criado. Seu significado é uma “amarração” de idéias e experiências não relacionadas previamente na mente do ouvinte que se conectam com as idéias e experiências da mente do compositor que o levaram à criação” [Gareth Loy 1991, p.34].

## 5. Quinta Reflexão

Outro argumento que se costuma levantar a favor da viabilidade de uma composição integralmente algorítmica diz respeito à possibilidade de se avaliar o processo apenas pelo produto final. Diz esse argumento que se o produto for convincente quanto à sua criatividade, isso seria prova suficiente de que o algoritmo que o gerou tinha de fato potencial criativo. Esse tipo de argumento, que surge nos debates sobre inteligência artificial, é uma adaptação falaciosa do teste de Turing sobre computabilidade.

Para contestar esse argumento apresenta-se um exemplo hipotético que ilustra a objeção. Suponhamos que um pesquisador, inusitado *expert* tanto em musicologia histórica quanto composição algorítmica, tenha encontrado por acaso, numa igreja do interior de Minas Gerais, um cofre lacrado há séculos. Ao abri-lo teve a surpresa de encontrar manuscritos inéditos assinados por ninguém menos que J. S. Bach. Aquele

pesquisador havia tido, a vida toda, uma só obsessão: escrever pelo menos uma peça que fosse tão bela quanto uma de Bach, e ao seu estilo. Para isso ele estudou durante anos as características das técnicas do compositor e tentou gerar por lógica algorítmica composições que se parecessem com as dele. Até que finalmente suas peças realmente começaram a soar semelhantes às de Bach, pelo menos quanto ao estilo, mas, para sua decepção, nenhuma parecia alcançar a perfeição criativa das obras do mestre.

A descoberta mineira ofereceu-lhe a oportunidade de superar a frustração. Transcreveu para a memória de um computador, nota por nota, a longa lista de músicas desconhecidas que encontrara. Criou a seguir um algoritmo absolutamente elementar cuja única função era disparar aleatoriamente a execução das peças armazenadas no banco de dados do computador. Como a lista era infinitamente longa, durante horas infindáveis os usuários do programa podiam ouvir músicas que julgavam ser composições algorítmicas do pesquisador. Finalmente suas peças soavam tão perfeitas e criativas como as obras originais de Bach! Ninguém desconfiou da artimanha e todos se convenceram que um algoritmo computacional podia realmente ser tão criativo quanto a mente de um grande mestre como J. S. Bach. Só o autor da façanha sabia que era apenas a criatividade do próprio Bach que se manifestava no produto daquele algoritmo.

## 6. Sexta Reflexão

Outros obstáculos que são colocados para o reconhecimento da criatividade intrínseca de nossos programas de computação musical dizem respeito à natureza inconsciente do processo criativo e ao processamento paralelo de múltiplas informações em que nossa mente estaria envolvida quando compomos. É provável que os dois conceitos estejam intimamente relacionados. Os modelos psicológicos ou cognitivos que reconhecem a dicotomia consciente/inconsciente postulam invariavelmente a existência de processamentos paralelos e independentes dos pensamentos, mas ao mesmo tempo supõe que essas *camadas autônomas* são capazes de trocar informações entre si, em múltiplos níveis lógicos e semânticos, dialeticamente, influenciando-se mutuamente.

Ora, em torno de 1990, esse obstáculo prometia ser superado com a adoção de modelos conexionistas de programação que simulariam digitalmente o funcionamento de redes neurais. Esses algoritmos teriam a habilidade de aprender e evoluir por si mesmo. E ainda emular o funcionamento de processamentos paralelos, ainda que em última instância o processamento de dados em nossas máquinas acabe sempre constrangido por gargalos do tipo *pipe-line*.

Vinte anos se passaram e nenhuma obra prima de composição algorítmica emergiu como resultado desse modelo alternativo. Ou a programação desses algoritmos é difícil demais, ou a compreensão de nossos mecanismos de pensamento durante o processo criativo permanece inescrutável e, portanto, não passível de formalização e conseqüentemente de programabilidade. Ou mais provavelmente, ambos os casos.

A despeito disso cremos que esse é o horizonte que restou em nosso imaginário como possibilidade vislumbrada de que algum dia venha realmente a haver algoritmos com faculdades criativas inerentes. Para que isso aconteça me parece existir ainda outro requisito. A criação artística que mais nos impressiona envolve sempre um componente reflexivo sobre o próprio ato criativo. De outra maneira, é considerada fenômeno

epigonal. Esse é o *karma* que pesa sobre inúmeras das tentativas de música algorítmica por David Cope e outros em sua linha de pesquisa. A análise de um estilo existente como fator determinante de uma nova composição resultará sempre em peças “à maneira de”. Arte que copia o estilo de outros é considerada uma arte menor, não criativa. Uma criação musical autêntica implica necessariamente na invenção de um estilo novo ou num distanciamento significativo de estilos anteriores.

O modelo conexionista descrito por Gareth Loy (1991) prometia a possibilidade de uma simbiose entre o pensamento algorítmico e a mente do compositor, o que poderia finalmente resultar numa criatividade auxiliada por computador que resolvesse o problema estilístico. Todavia, no limite, isso implicaria que a criatividade do algoritmo ainda não seria uma propriedade exclusivamente inerente ao sistema, mas uma projeção da criatividade do programador. Assim, nossa ambição de dotar o sistema de autonomia criativa voltaria à estaca zero. Aliás, como poderia um algoritmo conter uma propriedade de criatividade que já não estivesse na mente do programador, se o programa é apenas uma projeção de sua mente? E ainda: somente *um compositor criativo* seria capaz de entender a natureza do próprio processo criativo para ser capaz projetá-lo abstratamente em um algoritmo. Acho que temos que nos contentar com isso, mas, ainda assim, esse *isso* não é nada desprezível!

## 7. Sétima Reflexão

Parece incorreto falar em algoritmos de composição musical como se eles fossem uma classe uniforme. É possível reconhecer pelo menos dois grandes grupos com abordagens diferentes do problema, com implicações diferentes.

O grupo mais tradicional baseia-se na mistura de postulações de regras de formação do discurso, com princípios de *Ars Combinatoria* associados a processos de seleção aleatória. Essa abordagem era a essência das “brincadeiras” propostas por Mozart e outros tantos vinte autores do século dezoito, que prometiam ao leigo, gerar ele mesmo, infinitas e diversificadas composições, usando um mero jogo de dados. Esses algoritmos prescreviam *regras* estilísticas de danças (a métrica, o tamanho das frases e períodos, a forma, etc) e ofereciam um banco de dados de *combinações* possíveis a partir do qual o autor-usuário devia selecionar *aleatoriamente*, pelo jogo de dados, os eventos que preencheriam o modelo formal.

Uma parte importante dos algoritmos de composição que utilizamos ainda hoje depende desses mesmos procedimentos clássicos: possibilidades combinatórias, acaso, regras de formação generalizadas a partir de peças-modelo pré-existentes. Lembremos que, já adentrando o século XX, Poincaré (1854-1912) ainda defendia que a criatividade era apenas um efeito de superfície da *Ars Combinatoria*.

Já um segundo grande grupo de algoritmos postula algum tipo de modelo evolutivo que parte de um estado inicial e calcula o estado seguinte a partir do anterior. A geração transformacional recursiva é o elemento diferencial desse modelo. Todavia parece não haver nunca “nada de novo sob o sol”. O modelo de pensamento sobre o qual se baseia esse tipo de algoritmo é tão “clássico” quanto o primeiro. Schoenberg identifica já em Mozart o modelo da *developing variation*, ou seja, “variação em desenvolvimento” ou “variação progressiva,” conforme a preferência do tradutor, que

realiza justamente um tipo de procedimento musical recursivo: o material subsequente é derivado do material anterior por algum procedimento de variação.

A música do século XIX nos permite identificar muitas outras maneiras e estilos de implementação desse princípio, bastando reconhecê-lo, pela análise, nas obras de Beethoven, Brahms e Liszt. Schoenberg propôs a continuidade desse modelo na música século vinte, reconhecendo que a variação em desenvolvimento continuou sendo o motor fundamental de seu processo criativo na fase da chamada atonalidade livre.

Não é de se estranhar, portanto, que a composição algorítmica, a partir das últimas décadas do século vinte, tenha encontrado seus próprios e férteis caminhos nesse segundo grande grupo de algoritmos. As cadeias de Markov, os autômatos celulares de von Neumann, os algoritmos genéticos em geral e os modelos de vida artificial [Todd e Miranda 2006, p.376] representam modelagens algorítmicas que expandem para a música gerada por computador o princípio composicional da *developing variation*. Afinal Schoenberg nunca definiu os procedimentos passo a passo que caracterizariam essas transformações do material original. Portanto qualquer método que realize esse princípio encaixa-se na definição em aberto que ele nos legou.

As redes transformacionais de Lewin (2007), embora concebidas como ferramentas analíticas, também podem ser usadas para modelar composições algorítmicas desse segundo grande grupo. Por se tratar de um modelo altamente formalizado matematicamente, as redes de Lewin se oferecem naturalmente à implementação algorítmica. Algumas peças de Coelho de Souza escritas com o auxílio de computador no final dos anos 80, como *Estação Paraíso* e *Diálogos*, usaram consistentemente o princípio das redes transformacionais de Lewin, ainda que de modo implícito, porque na época a teoria de Lewin era desconhecida no Brasil.

O ganho sobre modelos como Markov, Von Neumann e outros de cunho genéticos, é que o princípio das redes transformacionais de Lewin é derivado da observação analítica da própria linguagem musical e sua aplicabilidade, desde sua concepção, não está restrita a um determinado estilo musical. A prova disso é que a teoria de Lewin foi usada, por ele e por outros, tanto para a análise da música baseada em conjuntos atonais, como a música tonal clássica e até mesmo a ultra-cromática, um terreno que por décadas resistiu à especulação estrutural e formalização teórica.

## 8. Oitava Reflexão

Para exemplificar alguns pontos levantados anteriormente comentaremos problemas relativos à criatividade em uma peça algorítmica composta por Coelho de Souza em 1990. Trata-se de *Metrópolis*, que é parte do ciclo de peças *Tristes Trópicos*. Voltar a uma peça antiga permite um exercício de reflexão analítica que muitas vezes não é possível quando estamos próximos demais do objeto estudado. Esta foi a primeira composição de Coelho de Souza inteiramente algorítmica. Na época a tecnologia que lhe era disponível permitia elaborar o pensamento composicional com algoritmos relativamente complexos, mas a execução sonora estava limitada à tecnologia Midi. A composição, por isso, foi inteiramente baseada em conceitos de notas Midi, como ritmos proporcionais representados pelos pontos de ataque e relaxamento das teclas de um teclado e dinâmicas representadas por *velocidades*. O sintetizador Korg usado na primeira versão era monofônico, mas sensível a *velocidades*, de modo que não só ele

respondia a dinâmicas como se podia obter alguma variação do timbre em função do ataque e das oitavas, variações certamente de amplitude limitada, mas importantes para se conferir alguma diversidade de timbre ao produto final.

Estava em moda na época a pesquisa sobre fractais. Um encontro em Nova York com o pesquisador e compositor Charles Dodge e o subsequente convite para que ele participasse da Bienal de Artes de São Paulo de 1989 despertou o interesse de Coelho de Souza pelo tema da música fractal a partir de um artigo escrito por Dodge [1988]. *Metrópolis* de fato não leva até as últimas conseqüências as implicações da matemática fractal, mas se inspira em algumas características formais desse tipo de número. Algumas experiências iniciais logo comprovaram que as dificuldades de implementação de uma peça integralmente fractal podiam ser insuperáveis com a tecnologia disponível na época, mas mesmo assim valia a pena um compromisso parcial com o pensamento fractal, principalmente quando ele sugeria a possibilidade de relações entre números irracionais, projetados sobre as proporções rítmicas, e quanto ao uso do princípio da auto-similaridade, que tem na técnica clássica da imitação um precursor bem sucedido. Na época encontrava-se o princípio da auto-similaridade fractal claramente exposto em Stevens [1989, p.26]: “a maioria dos fractais é auto-similar, de modo que a forma que identificamos no desenho de uma curva fractal repete a si mesmo em uma escala cada vez menor à medida que alargamos a imagem mais e mais.”

A figura 1 ilustra como uma forma aproximada desse princípio fractal foi aplicada ao nível rítmico de *Metrópolis*. A figura apresenta a tradução em notação musical de quatro momentos da lista de dados Midi produzidos pelo algoritmo. Esses momentos aparecem em seções sucessivas da peça e representam um mergulho progressivo em texturas cada vez mais densas. O projeto da peça propunha que ela se auto-reciclasse em sucessivas variações dos mesmos materiais, mas, para cada novo ciclo, o algoritmo de geração da peça calculava a inserção de uma imagem proporcional da cadeia original, ritmicamente diminuída, sobre as notas de maior duração. Note-se que, embora as classes de alturas das notas inseridas permaneçam as mesmas, sua ordem, suas dinâmicas e seus registros eram alterados por um cálculo permutacional.



Figura 1. Progressão da diminuição fractal em quatro seções de *Metrópolis*

O algoritmo de geração da peça, escrito em C, beneficiou-se de modelos desenvolvidos por Schildt [1987]. A necessidade poética de um cálculo algorítmico para a composição dessa peça deveu-se principalmente ao grande volume de dados que necessitava ser gerado para produzir o efeito almejado. A figura 1 dá conta apenas de um pequeno fragmento de uma das vozes da polifonia da peça. O acúmulo sucessivo de eventos acaba produzindo uma massa de dados que só podia ser manipulada computacionalmente, decorrendo daí a característica idiomática da peça que justificava o emprego tanto do algoritmo composicional quanto da execução controlada por computador (através de um programa seqüenciador).

O efeito poético almejado era uma metáfora do adensamento progressivo e incontrollável das grandes cidades. A técnica empregada nada mais é do que uma atualização computacional do princípio barroco da diminuição rítmica. A técnica barroca da diminuição foi exemplarmente utilizada em peças como as *Variações do Ferreiro Harmonioso* de G. F. Handel, e nas *Variações Goldberg* de J. S. Bach. Concebido como um dispositivo para a exibição da virtuosidade instrumental, sua transposição para um meio computacional elimina a aura de excepcionalidade performática, mas permite uma atualização da significação como representação do mundo contemporâneo.

Reconhece-se que o procedimento pseudo-fractal empregado é gerado segundo características do segundo grupo de algoritmos descrito acima, isto é, aqueles que utilizam o princípio da recursividade transformacional. Por outro lado a geração de notas e dinâmicas pertence ao primeiro grupo de algoritmos, pois emprega o princípio de permutação de intervalos para a geração das alturas, combinado a processos aleatórios de decisão sobre qual permutação seria utilizada num ponto específico da composição. Como predomina na obra uma percepção global da textura musical, de fato era pouco relevante qual permutação estivesse sendo escolhida, desde que ela fosse sempre mutável para conferir variedade e alguma imprevisibilidade à textura resultante.

Na verdade seria possível representar a forma da peça através de uma gigantesca rede transformacional de Lewin, uma vez que não só todas as notas, mas todos os parâmetros são gerados como transformações de uma semente inicial, dada pela progressão 1-2-3-4 que se projeta sobre os intervalos de alturas, ritmos e dinâmicas. Esses números inteiros são a seguir transformados em proporções irracionais. Essa seria uma razão suficiente para se pleitear a necessidade de um tratamento algorítmico para a peça. Ainda assim não é possível defender que os algoritmos em C desenhados para essa composição tenham uma criatividade intrínseca. Todas as decisões fundamentais para a poética da composição da peça foram tomadas antecipadamente pelo programador. O programa apenas cumpriu um processo que lhe fora pré-determinado. Outros *outputs* do mesmo algoritmo gerariam peças diferentes nos detalhes, mas essencialmente equivalentes na significação musical.

Relatemos ainda uma etapa do processo composicional de *Metrópolis* que se revelou bastante instrutiva. A primeira versão do algoritmo havia produzido resultados que soavam muito insatisfatórios do ponto de vista musical, não importando quantos *outputs* diferentes fossem gerados. Mesmo não se encontrando nenhuma falha na lógica estrutural da composição, aquelas sucessões vertiginosas de notas pareciam meramente caóticas. A solução veio de Debussy, através de um artigo de Eimert [1961]. Em *Jeux*,

Debussy resolveu um problema similar criando um discurso não-tonal que nunca repete o que foi dito antes, em larga e média escala, mas impõe que cada evento seja sempre repetido imediatamente duas vezes. A repetição, que provoca uma simetria perceptual, torna-se condição *sine qua non* para uma linguagem musical que não pode contar com um vocabulário a priori de significações codificadas. Em outras palavras, uma das ilusões frequentes na composição algorítmica é pleitear a auto-suficiência da estruturação formal, esquecendo-se das condicionantes perceptuais da linguagem musical. Afinal a música é para ser ouvida e entendida pelo homem, não pela máquina.

## Conclusões

As técnicas tradicionais de IA baseiam-se principalmente na representação simbólica de processos cognitivos, uma representação explícita de um domínio específico do conhecimento. As redes neurais artificiais, entretanto, não requerem uma representação simbólica tão explícita, podendo ser treinadas e ajustadas para fazer reconhecimentos de padrões, até que se mapeiem satisfatoriamente conjuntos de entrada em conjuntos de saída.

Nos anos 1970 e 1980 notou-se um esforço concentrado das comunidades de IA em explicar o funcionamento lógico do intelecto exclusivamente em termos de manipulações simbólicas do conhecimento. Ao final dos anos 1980, ferramentas como as redes neurais artificiais e modelos conexionistas trouxeram frameworks mais simples e mecanicamente robustos para aplicações de IA. Em música, o interesse voltou-se para os sistemas inteligentes que poderiam aprender a resolver problemas musicais, como, por exemplo, a harmonização, abordando questões fundamentais sobre o que é o conhecimento musical e como um computador pode raciocinar sobre ele.

Não obstante, para que os problemas fossem computacionalmente tratáveis lançou-se mão de modelos reducionistas, embora com o desafio de aliar à abordagem, ao máximo possível, um "conhecimento natural", que seja psicologicamente plausível e compatível com as intuições musicais humanas. Widmer [1992] trabalhou neste sentido buscando formalizar modelos qualitativos para percepção musical por meio de uma hierarquia multidimensional de raciocínios lógicos, em que progressivamente cada nível algorítmico se ocupasse da qualificação de um aspecto musical (como métrica, a consonância, a coerência, a regularidade, etc). Em síntese a indução e a recursividade de regras previamente estabelecidas ou definidas em tempo de execução constituíam a base lógica processual, e embora os procedimentos parecessem dedutivos, deve-se ter em mente que, de fato, o sistema somente fornecia "qualificações musicais plausíveis", com influências pautadas pelas regras fornecidas pelo compositor.

Apesar do sucesso parcial de técnicas propostas nas últimas décadas, em especial para modelagem qualitativa, em que os sistemas não podem ser descritos e simulados em detalhe por serem muito complexos ou não muito bem conhecidos, frequentemente, entretanto, os autores reconhecem limitações nos modelos que não oferecem uma ampla formalização das inúmeras dimensões musicais, bem como a integração com aspectos psicoacústicos e tecnológicos.

Parece-nos que os desafios para a criatividade aflorar em sistemas computacionais apontam para o uso de modelos multidimensionais e suportados por

hierarquias de raciocínio paralelos. É possível que as limitações estejam na incapacidade atual de implementar modelos cuja complexidade (e previsibilidade) se aproximem das do cérebro humano, no limite portanto de nossa capacidade para racionalizá-los. Em outras palavras, a modelagem torna-se complexa demais via regras convencionais de lógica e programação.

Embora os recursos computacionais, particularmente os sistemas multiprocessadores, abram campo para ampliar as fronteiras neste sentido, não se pode esquecer que a criatividade para humanos deve ser entendida por humanos. Neste universo – consequência das metáforas humanas – não raro um aspecto controlador sempre predomina sobre os demais na unidade de tempo em cada projeto, e assim a noção da hierarquia se impõe ao processo criativo modelado.

Finalmente, entre as alternativas que oferecem caminhos férteis para a composição algorítmica há uma que não depende de novos modelos de inteligência artificial. A maioria das pesquisas em composição algorítmica ainda está presa à idéia de uma composição baseada em notas. Entretanto, a música eletroacústica há muito se descolou desse princípio, concentrando seu interesse na transformação do próprio som e não no encadeamento de suas “fundamentais” como representantes desses sons. Na maioria das vezes tudo o que um algoritmo de composição com notas faz, poderia ser feito manualmente com eficiência equivalente, pois a quantidade de dados manipulados é pequena. Entretanto quando se trata de manipular digitalmente a própria onda sonora, a massa de dados é de uma magnitude que só pode ser tratada computacionalmente. Nesse sentido é que uma composição algorítmica, que trabalhe diretamente sobre a forma da onda sonora, poderia tirar grande partido do auxílio computacional gerando digitalmente texturas percebidas criativamente em sua superfície. Se nesse caso talvez ainda hesitássemos em reconhecer que o algoritmo tivesse se tornado um colaborador de fato criativo, não restaria dúvida de que ele seria, pelo menos, um servo de extraordinárias virtudes.

## Referências

- Cope, D., *Computer Models of Musical Creativity*, The MIT Press, 2005.
- Dodge, C., (1988) “*Profile: A Musical Fractal*”, *Computer Music Journal*, Vol.12, No.3, Fall 1988, Massachusetts Institute of Technology, pp.10-14.
- Eimert, H., (1961) “*Debussy’s Jeux*”, In: *Die Reihe* 5, Theodore Presser, pp.3-20.
- Lewin, D., *Generalized Musical Intervals and Transformations*, Oxford University Press, 2007.
- Loy, D. G., (1991) “*Connectionism and Musiconomy*”, In: *Music and Connectionism*, Editado por Peter M. Todd e D. Gareth Loy, The MIT Press, pp.20-36.
- Pinker, S., *Do que é feito o pensamento: a língua como janela para a natureza humana*, Companhia das Letras, 2008.
- Schildt, H., *Artificial Intelligence Using C*, Osborne McGraw-Hill, 1987.
- Stevens, R., *Fractal Programming in C*, M&T Books, 1989.

Todd, P. e Miranda, E. (2006) "Putting some (artificial) life into models of musical creativity", In: *Musical Creativity: Multidisciplinary Research in Theory and Practice*, Editado por Irène Deliège e Geraint Wiggins, Psychology Press, pp.376-395.

Turner, S., *The Creative Process: A Computer Model of Storytelling and Creativity*, Lawrence Erlbaum, 1994.

Widmer, G., (1992) "Qualitative Perception Modeling and Intelligent Musical Learning", *Computer Music Journal*, Vol.16, No.2, Summer 1992, Massachusetts Institute of Technology, pp.51-68.

Wiggins, G., Allan, H. e Müllensiefen, D., (2007) "Methodological Considerations in Studies of Musical Similarity", Viena, SMIR'07.