

Da teoria à prática, como transitar com amplos benefícios: um estudo de caso – Algoritmos Vivos.

Pedro Luis Kantek Garcia Navarro

Professor aposentado. Lecionou nas Universidade Federal do Paraná (UFPR), Universidade Tecnológica do Paraná (UTFPR), Pontifícia Universidade Católica do Paraná (PUC/Pr) e Universidade Positivo (UP) em Curitiba, Brasil. E-mail: pkantek@gmail.com.

Resumo da Proposta

Apresenta um método de passar de aulas teóricas para aulas práticas denominado ALGORITMOS VIVOS. Nele cada aluno em cada aula recebe um problema único no universo. Gerado, impresso (e resolvido) por um software, ele delega ao aluno estudar o tema e aplicar a teoria a esta instância em particular. Iniciado em 1993 o método evoluiu para 653 temas, todos vinculados à área de ciências exatas. O trabalho descreve algumas grandes áreas de aplicação. A plataforma de desenvolvimento é explicitada. O processo de correção semi-automatizado é também descrito. Alguns resultados reais de seu uso em quase 30 anos são apresentados. Termina o trabalho a mostragem de seis exemplos de Algoritmos Vivos: o problema do caixeiro viajante, uma proposta de linguagem para ser mostrada no primeiro dia de um curso de programação, o método PERT/CPM, uma compressão de dados usando Huffman, uma calculadora manual e a máquina ENIGMA. Ao final, há a disponibilização da plataforma.

Tipo de Trabalho

Pecha Kucha (apresentação curta)

Tema da Conferência

Outras práticas de conhecimento aberto (hardware e software livre, educação aberta)

Palavras-chave

Ferramenta pedagógica; algoritmo; programação; prática em sala de aula

Audiência

Gestores educacionais e professores de qualquer área científica de ciências exatas. Organizadores de currículos à distância. Interessados no desenvolvimento de ferramentas pedagógicas inovadoras.

Proposta

O ano é 1993. Precisando ensinar Estruturas de Dados e nelas a compressão de Huffman, disciplina e tema difíceis, complexos e abstratos, pensava em como o computador poderia ajudar. A resposta: gerando um exercício diferente para cada aluno. O que começa como experiência logo se torna método: as aulas ficam mais interessantes. Como todos têm o mesmo enunciado, naturalmente surge o trabalho em grupo. O gabarito – gerado no momento da criação do exercício – está guardado e facilita a posterior correção. Não há risco de *socialização de resultados*, lembremos que cada aluno tem uma instância distinta. As aulas migram da teoria à prática. O resultado ganha um nome: ALGORITMOS VIVOS. Ele é sempre uma folha A4, contendo a teoria subjacente ao tema e um problema com dados aleatoriamente gerados (e convenientemente resolvidos pelo programa gerador, para ter a resposta certa). O método rapidamente se expande para praticamente todos os temas das diversas disciplinas. O que começou com um exercício em 1993, chega a 2021 com uma coleção de 653 exercícios/temas. Eis alguns quantitativos: 105 de introdução à Ciência da Computação, 182 de programação, 47 de bancos de dados, 46 de inteligência artificial, 27 de matemática discreta, 19 quebra-cabeças e 224 de outros assuntos variados.

Uma consequência importante é diluir o processo de avaliação, diminuindo o terror dos alunos. Agora toda aula é uma prova. O exame final pode ter seu peso minimizado sem prejuízo. Depois de quase 30 anos de uso é possível avaliar algum resultado: diminuiu a evasão, as disciplinas perderam a pecha de *reprovadoras* e ao final do curso diversos TCCs (trabalho de conclusão de curso) tem como base algum Algoritmo Vivo feito ao longo do curso, uma bonita homenagem ao método.

A plataforma principal dos Algoritmos Vivos é a linguagem APL. Ela é insuperável para isso. Criou-se uma linguagem algorítmica e seu correspondente compilador, chamados CPC para construir e executar programas dentro do ambiente. Depois, há uma passagem pelo LATEX para garantir a elegância dos textos e o sistema de apoio ao uso está feito em Harbour. Há vários suportes em C++, Python, Lisp e Prolog conforme o caso.

Detalhe importante é que no meio do processo gera-se um arquivo PDF contendo as folhas individuais. Está aberta a porta para uma fácil publicação na WEB dos exercícios pedidos. Isto funciona como *back-up* da distribuição e como memória da disciplina. O método está comprovado: diferentes exercícios compartilham cerca de 60% do código gerador e hoje é possível e razoavelmente fácil expandir a idéia para qualquer tema da área de exatas: matemática, cálculo, finanças, contabilidade, estatística, modelagem,...

Comentando sobre a correção: na modalidade completa de uso, há que se ter um banco de dados de alunos, de disciplinas, de temas e de exercícios. Na folha contendo um Algoritmo Vivo é impresso um código de barras de identificação. A correção é simples: lido o código de barras o sistema apresenta na tela ao corretor humano o resultado esperado e aguarda que este lance o grau obtido por aquele aluno naquele exercício. Menos de 1 minuto por folha.

A seguir seis exemplos de Algoritmos Vivos. O primeiro é sobre o problema do caixeiro viajante. Trata-se de algo fundamental na Ciência da Computação. Há várias cidades com posições conhecidas e alguém deve visitar todas uma única vez retornando à origem. Busca-se qual o trajeto que minimiza a distância percorrida. O problema cresce muito rápido: para 100 cidades qualquer computador demora semanas. A folha pede que o aluno calcule o resultado para um conjunto de 10 cidades reais do Estado do Paraná.

O segundo é destinado ao primeiro dia de aula de um curso de programação. Descreve-se o trabalho de Alan Turing e o de Von Neumann. Mostra-se uma mini linguagem de programação bem simples. Explicitam-se 4 algoritmos (aleatórios) com seus dados de entrada e pede-se que o aluno descubra o que cada algoritmo fará.

O terceiro é sobre o método de planejamento PERT/CPM. Introduce a Guerra Fria e nela o projeto do míssil POLARIS, com suas exigências de planejamento com custos terríveis em caso de não cumprimento. Mostra o método PERT/CPM, e o Algoritmo Vivo gera um projeto (aleatório) com cerca de 30 atividades. O aluno precisa achar o caminho crítico.

O quarto homenageia a origem de tudo e pede que o aluno comprima e descomprima dois versos de Os Lusíadas usando o algoritmo de Huffman com toda a sua complexidade.

O quinto pede que o aluno construa uma calculadora manual. Descreve-se o trabalho de Napier envolvendo a correlação entre números e logaritmos mostrando como se pode economizar esforço ao fazer isto. Citam-se as antigas régua de cálculo e a tarefa final é construir uma *régua de cálculo de papel* e nela fazer 3 operações aritméticas.

Finalmente o sexto exemplo descreve a máquina de criptografia ENIGMA em uso na II Grande Guerra e sua importância na Batalha do Atlântico. Avança-se pelo trabalho de Turing na sua decifração, tema coberto pelo filme *The Imitation Game*, vencedor de diversos prêmios. O exercício entrega uma ENIGMA, sua regulação e uma mensagem criptografada. O aluno precisa descobrir o que está lá que a propósito é um pequeno trecho de Shakespeare.

Em consonância com o tema desta Conferência (“Ciência Aberta”) o acervo dos algoritmos vivos está descrito em <http://algoritmovivo.com> e lá há também alguns exemplos de seu uso na Universidade Federal do Paraná. Tudo está à disposição de universidades e professores. Há um projeto de torná-lo mais amigável, mas já está à disposição. O autor pode ser contactado em pkantek@gmail.com e será um prazer atender e responder.

Referências Bibliográficas

BROWN M. e SEDGEWICK R. "A system for algorithm Animation", Proceedings of ACM SIGGRAPH 1984, Minneapolis, July 1984, pp 177-186.

BYRNE M, CATRAMBONE R. e STASKO J. "Do algorithms animations aid learning?" Technical Report GIT-GVU-96-18, GVU Center, Georgia Institute of Technology, Atlanta, august 1996.

De MARCO, Tom e LISTER, Timothy. Peopleware. Ed McGraw Hill. 1990.

DECRETO LEI n. 2306, de 19 de agosto de 1997, que regulamenta as Instituições de Ensino Superior no Brasil. Disponível na íntegra em www.mec.gov.br/Sesu/ftp/Decreto2306.doc.

DEWDNEY, A. K. The Turing Omnibus. Computer Science Press. 1989.

HAAJANEM, M. et alli: Animation of user algorithms on the Web. In: Proc. VL '97, IEEE. Symposium on Visual Languages, IEEE 1997, 360-367.
<http://www.cs.helsinki.fi/research/aaps/Jeliot/>

HILL T. "Assessing the instructional value of students predictions in tree animations", SIGCSE Doctoral Consortium Application, University of Mississippi, 2002.

KANTEK, Pedro. Uma Introdução ao APL. CELEPAR. 1982.

LIMA, Lauro de Oliveira. Treinamento em Dinâmica de Grupo. Vozes. 1973.

STASKO J. "TANGO: A framework and system for Algorithm Animation", Computer, vol 23, no.9, September 1990, pp-27-39.

WIRTH, Niklaus. Algorithms and Data Structures. Prentice Hall. 1981