



Modelagem urbana e desempenho de sistemas computacionais: estudo e aplicação em um simulador de crescimento urbano – SACI®.

SARAIVA, Marcus Vinicius Pereira; POLIDORI, Maurício Couto

*UFPeI, Faculdade de Arquitetura e Urbanismo
Rua Benjamim Constant, 1359 - Campus Porto - Pelotas/RS.
marcus.saraiva@gmail.com; mauricio.polidori@terra.com.br*

1. INTRODUÇÃO

Cidades em crescimento têm sido assumidas como fenômenos complexos, envolvendo grande quantidade de fatores urbanos, naturais e institucionais. Estes fatores interagem em diferentes escalas e mudam ao longo do tempo, desafiando a ciência a encontrar explicações, nexos e padrões. Modelos de computador têm sido utilizados com sucesso para entender o processo de crescimento das cidades, possibilitando a realização de simulações da dinâmica urbana e permitindo a elaboração e teste de novas regras e processos de crescimento. Este artigo trata do modelo de simulação de crescimento urbano denominado SACI® - Simulador do Ambiente da Cidade, desenvolvido por Polidori (2004) e implementado como uma extensão do software ArcView®, utilizando seus recursos nativos de Avenue e a linguagem de programação C++.

Este trabalho está proposto para revisar o modelo de simulação de crescimento urbano SACI, suas características conceituais e computacionais, identificando as atuais limitações e potencialidades do software, com o objetivo de desenvolver uma nova versão do sistema. Os principais problemas identificados pelos usuários do sistema são: 1) excessivo tempo de processamento necessário para efetuar as simulações; 2) dependência de software proprietário de SIG, no caso, o ArcView®, da ESRI. O estudo destes problemas tem quatro etapas de trabalho: a) estudar o modelo de simulação de crescimento urbano SACI® e sua implementação computacional; b) encontrar, no algoritmo geral do sistema, as rotinas responsáveis pelo desempenho atual do software; c) desenvolver algoritmos mais eficientes ou alternativos para as rotinas encontradas na etapa anterior; d) implementar esses algoritmos em um sistema independente, com suporte aos formatos de arquivos utilizados pelos softwares de SIG mais tradicionais.

2. O MODELO DE SIMULAÇÃO DE CRESCIMENTO URBANO

O modelo de simulação de crescimento urbano SACI® é dedicado a realizar estudos da dinâmica espacial urbana, considerando integradamente fatores naturais, urbanos e institucionais, implementando crescimento interno e externo à cidade preexistente. A cidade e o território são modelados em ambiente computacional, reedi-

tando os modelos de potencial e centralidade (Krafta, 1994), com apoio em teoria de grafos, autômato celular e geotecnologias.

Autômatos celulares foram desenvolvidos inicialmente por John Von Neumann e Stanislaw Ulam, na década de 40, estando ligados a estudos sobre computação, inteligência e vida artificiais (Torrens, 2000). Podem ser considerados como um espaço finito composto por células organizadas em um grid, que mudam de estado automaticamente seguindo certas regras de transição, em função dos estados das células vizinhas. A interação desses componentes simples (células) com sua vizinhança pode gerar padrões de comportamento complexo, assim como ocorre com as cidades.

A teoria de grafos provém da topologia (Sánchez, 1998), a qual se dedica a estudar relações entre pontos, linhas e superfícies, a partir de suas conexões. Grafos são conjuntos finitos de nodos conectados por arestas (Mariani, 2001) e são úteis para representar as relações de interação ou tensão espacial entre as células. Para isso, é necessário converter a estrutura do grid em um grafo, o que é feito considerando cada célula como um vértice e a vizinhança celular do tipo Moore (8 vizinhos) como as arestas de um grafo. Esta representação está ilustrada na Figura 1.

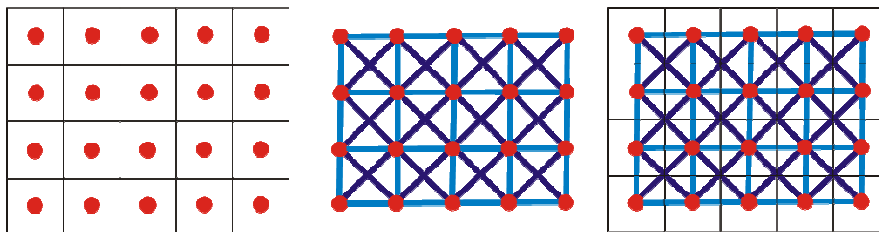


Figura 1: a) representação celular; b) representação em grafos; c) células + grafos.

Enquanto autômatos celulares tratam de relações em nível de vizinhança imediata, grafos permitem representar estruturas espaciais a partir de relações globais ou vizinhanças remotas. Deste modo, a utilização de recursos de grafos com recursos de autômatos celulares permite tratar integradamente relações locais e relações globais.

As informações disponíveis para a modelagem urbana são armazenadas em um ambiente SIG, responsável pela leitura, organização e visualização dos dados. Softwares de SIG permitem a execução de operações de elevada complexidade envolvendo dados espaciais com relativa facilidade, aliada à possibilidade de desenvolvimento de novas ferramentas geoespaciais, através de geocomputação, destinadas à solução de diversos problemas específicos. No SACI[®], a integração com SIG é feita utilizando o software ArcView[®] 3.3, desenvolvido pela ESRI, ao qual o modelo de crescimento é instalado na forma de uma extensão.

Após quatro anos do seu lançamento, é possível fazer uma análise da situação atual do SACI[®] e projetar novos rumos para o sistema. O objetivo deste trabalho trata da superação das limitações e da implementação de novas funcionalidades ao sistema.

3. RESULTADOS PARCIAIS

Analisando o algoritmo geral do sistema foram encontrados três pontos críticos, responsáveis pela maior parte do tempo de processamento demandado pelas simu-

lações: a implementação do algoritmo de caminhos mínimos A*, a ausência de processamento paralelo e a interface entre o ArcView[®] e o SACI[®].

No modelo de crescimento, em cada iteração o sistema calcula as tensões espaciais e os caminhos mínimos entre todos os pares possíveis de células. Os caminhos mínimos entre duas células são calculado utilizando o algoritmo de busca heurística A*, levando em conta atrações e resistências existentes nas células do grid. A quantidade de vezes em que esta busca é executada, aliada à complexidade do algoritmo, torna este um ponto crítico no desempenho geral do sistema.

De modo a minimizar o impacto da busca de caminhos no desempenho da simulação, serão implementadas as seguintes técnicas para ganho de velocidade, descritas por Lester (2003): a) utilização de vetores unidimensionais para armazenamento de listas, ao invés de estruturas de dados dinâmicas e orientadas a objetos, que podem consumir uma parcela de tempo maior necessária para a criação e manutenção de tais objetos; b) manutenção da lista aberta utilizando uma binary heap, o que aumenta o desempenho exponencialmente nos casos de grids maiores.

Outra técnica que pode ser utilizada neste caso, com possibilidades de ganhos significativos de performance, é o armazenamento de caminhos já calculados. Neste caso o sistema inicia a busca de caminhos entre células mais distantes espacialmente, ao invés de começar pelas células mais próximas. Uma vez encontrado um dos caminhos mínimos entre estas células, o caminho entre a célula de origem e todas as células intermediárias estará automaticamente calculado, bastando ser armazenado em memória para ser utilizado novamente quando necessário. Este processo está ilustrado na Figura 2, adiante.

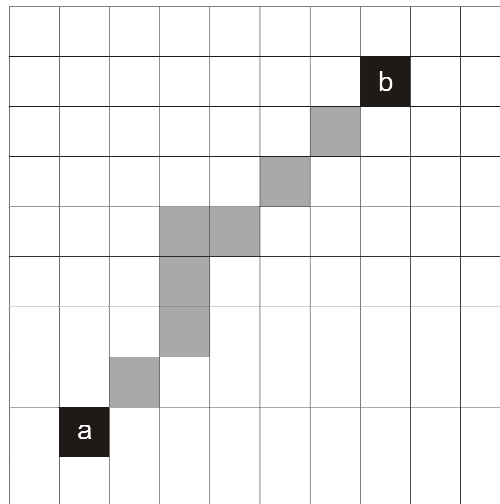


Figura 2: a – célula de origem; b – célula de destino; pintadas em cinza, células pertencentes a um dos caminhos mínimos possíveis entre a e b, que deve ser armazenado em memória para uso posterior.

Uma característica dos autômatos celulares é a de que se tratam de processadores paralelos, e não seriais, o que permite que o processamento de várias células seja feito simultaneamente. A versão atual do sistema não utiliza esta característica, comportando-se como um processador serial. A implementação de processamento paralelo pode ser feita, neste caso, em dois níveis: 1) processamento paralelo em um mesmo computador, através do uso de threads; 2) a distribuição do processamento através de um cluster de computadores. No segundo caso, o grid poderia ser dividido em pedaços menores e distribuído entre vários computadores via rede, que executariam o processamento e devolveriam os resultados para um servidor, que se

encarregaria de consolidar os dados recebidos e os redistribuir aos computadores clientes a cada iteração.

Por fim, a integração com o software ArcView[®] também gera uma perda de desempenho considerável no processamento. Por tratar-se de um software relativamente antigo, o ArcView[®] tem problemas de compatibilidade com sistemas operacionais e processadores mais novos, como as versões de 64 bits do Microsoft Windows[®]. Esta limitação será superada com o lançamento de um sistema SACI independente de plataformas proprietárias, facilitando a manutenção futura do software e ganhando em desempenho a medida que os processadores, sistemas operacionais e compiladores evoluem.

A segunda versão do SACI[®] está sendo implementada com a utilização do ambiente de desenvolvimento CodeGear Delphi[®], escolhida por permitir a criação de softwares de alta performance para o ambiente Windows[®]. Além disso, o Delphi[®] traz os recursos da VCL (Visual Component Library), um framework baseado em componentes que permite a criação de avançadas interfaces de usuário com facilidade e rapidez. A VCL é plenamente expansível, permitindo a instalação de componentes de terceiros bem como a criação de novos componentes.

A principal inovação desta segunda versão do sistema é a substituição do ArcView[®] por um pacote de componentes de SIG próprio, integrado ao software. Este módulo é responsável por toda a interpretação, edição e visualização de dados raster necessários às simulações de crescimento urbano. Este conjunto de componentes foi desenvolvido como parte da VCL, portanto pode ser distribuído e utilizado em outros softwares que utilizem a linguagem Delphi[®].

4. CONCLUSÃO

Analisando o modelo de crescimento SACI[®] foi possível entender seu funcionamento e identificar a possibilidade de ganhos de performance significativos. A implementação dos algoritmos sugeridos ao longo deste texto, bem como de outros que serão encontrados na continuidade dos trabalhos, trarão um ganho significativo de produtividade aos pesquisadores que se utilizarem do SACI[®] para suas simulações de crescimento urbano. Outro ganho significativo será a retirada do vínculo com o ArcView[®] e a inclusão de novas funcionalidades ao sistema, tornando o SACI[®] um software independente de plataformas proprietárias e aumentando a quantidade de usuários do sistema.

5. REFERÊNCIAS BIBLIOGRÁFICAS

LESTER, Patrick (2003). **Using Binary Heaps in A* Pathfinding**. [disponível em 24 de julho de 2008 em www.policyalmanac.org/games/binaryHeaps.htm]

MARIANI, Antonio Carlos (2001). **Teoria dos Grafos**. UFSC/CTC/INE. [disponível em 20 de agosto de 2008 em www.inf.ufsc.br/grafos/livro.html]

POLIDORI, Maurício Couto (2004). **Crescimento urbano e ambiente: um estudo exploratório sobre as transformações e o futuro da cidade**. UFRGS, PPGECCO.

TORRENS, Paul (2000). **How cellular models of urban systems work**. London: Casa, UCL. 75 p. [disponível em 20 de agosto de 2008 em http://www.casa.ucl.ac.uk/working_papers.htm]

KRAFTA, Romulo (1994). **Modelling Intraurban configurational development**. Environment and Planning B: Planning and Design, v. 21. London: Pion. p. 67-82.