

Algoritmos de Busca de Vizinhos em Diferentes Cenários de Distribuição Espacial

Bruno Normande Lins¹, Leonardo Pereira Viana²

¹Mestrado em Modelagem Computacional do Conhecimento (MMCC)
Universidade Federal Alagoas (UFAL)
Maceió – AL – Brasil

²Instituto de Computação – Universidade Federal Alagoas
Universidade Federal Alagoas (UFAL)
Maceió – AL – Brasil

normandelins@gmail.com, lpviana@lccv.com.br

Resumo. Neste trabalho são analisados os principais algoritmos de detecção de contatos em relação ao seu comportamento em diferentes cenários de distribuição espacial. Em busca do conhecimento de como tais mudanças de cenários afetam os diferentes algoritmos, cada algoritmo será testado em cada um dos cenários com números crescentes de partículas e seus tempo de execução medidos e comparados.

1. Introdução

Desde a formalização do Método dos Elementos Discretos (DEM) [Cundall 1988], algoritmos de detecção de contatos têm sido um problema em aberto: podendo chegar ocupar até 80% do tempo total de processamento [Munjiza 2004, Ericson 2005] tal classe de algoritmo se torna um verdadeiro gargalo nos sistemas em que precisam ser utilizados.

A paralelização de tais algoritmos não conseguiu baixar a proporção de tempo gasta nessa etapa, mas conseguiu possível a simulação com DEM para milhões [Markauskas and Kačeniauskas 2015] e até centenas de milhões [Fattebert et al. 2012] de partículas simultâneas.

O esforço para otimização dos sistemas que usam DEM está agora focado no balanceamento dinâmico do espaço de busca [Fattebert et al. 2012]. Tal abordagem permite a utilização de diferentes tipos de algoritmos em cada situação. Os algoritmos tradicionais de *grid* já foram comparados com algoritmos de árvore [Han et al. 2007] e já existem resoluções, para a queda no desempenho desses algoritmos quando a diferença dos tamanhos das partículas são muito grandes [Perkins and Williams 2001, Araújo et al. 2007].

Esse trabalho busca responder a pergunta de quais algoritmos são mais recomendados para os diferentes tipos de distribuições de partículas em uma simulação. Para tal os principais algoritmos de *grid* foram testados em diferentes tipos de distribuições, com número crescente de partículas. Algoritmos de árvore não foram incluídos pois seu desempenho já foi mostrado inferior em relação aos de *grid* em cenários com partículas de tamanho homogêneo.

2. Algoritmos de Detecção de Contato

O algoritmo mais básico de detecção de contato é a checagem direta. Nesse algoritmo, dado N elementos de entrada, para cada elemento N_i é feito o cálculo de verificação de

contatos com os outros $N - 1$ elementos. A complexidade assintótica desse algoritmo é $O(N^2)$, o que o torna impraticável para sistemas em tempo real ou que possuam mais do que algumas centenas de corpos interagindo.

A solução encontrada é realizar uma etapa conhecida como **Busca por Vizinhos**. Nessa etapa primeiro é feito um mapeamento de todos os elementos do sistema para em seguida ser verificado o contato apenas entre os elementos mais próximos, seus vizinhos, eliminando assim a maioria dos testes do sistema [Munjiza 2004].

Os algoritmos testados foram escolhidos por melhor representarem os diferentes tipos de algoritmos de *grid*, sendo um de mapeamento direto e outros dois usando listas, com e sem ordenação. Além disso esses algoritmos são apontados como os mais importante na literatura [Munjiza 2004, Ericson 2005]. São eles: Mapeamento Direto; Detecção de Contato por Ordenação; e NBS-Munjiza.

3. Metodologia

Os testes foram realizados usando o *software* DEMOOP, implementado no Laboratório de Computação Científica e Visualização (LCCV) na UFAL e executados no *cluster* computacional GradeBR.

Os casos de teste representam três cenários distintos de distribuição de partículas: empacotamento denso, empacotamento esparsos e empacotamento misto. Para cada tipo de empacotamento os algoritmos foram executados com os mesmos casos de entrada, com o número de partículas N variando de 5×10^5 até 5×10^6 .

Os resultados encontrados foram obtidos a partir da média aritmética do tempo de execução de cada algoritmo para cada valor de N .

4. Resultados

A Figura 1 mostra os resultados obtidos no cenário de empacotamento denso. Podemos ver que nesse cenário o NBS-Munjiza obteve um desempenho muito inferior em relação aos outros algoritmos, chegando a um comportamento próximo ao quadrático. Por outro lado o algoritmo de Detecção de Contato por Ordenação, mesmo com as etapas de ordenação e busca binária, foi capaz de obter um desempenho superior até mesmo ao mapeamento direto.

Em média, o algoritmo NBS-Munjiza teve um desempenho de 194% sobre o algoritmo de Mapeamento direto. Já o algoritmo de Detecção por Ordenação obteve uma melhora de 27% no tempo obtido pelo Mapeamento Direto.

5. Conclusão

Os resultados obtidos até o momento se diferenciam bastante do esperado inicialmente. Em média o algoritmo NBS-Munjiza obteve tempo de execução próximo ao dobro do Mapeamento Direto, se mostrando muito abaixo da expectativa no tipo de cenário em que ele deveria possuir pelo menos uma eficiência similar ao Mapeamento Direto. Isso pode se dar devido ao custo de limpar e reconstruir as listas usadas para mapeamento durante toda a etapa de busca.

Já o algoritmo de Detecção por Ordenação se mostra superior, precisando em média de apenas 72% do tempo de execução necessário ao Mapeamento Direto. O

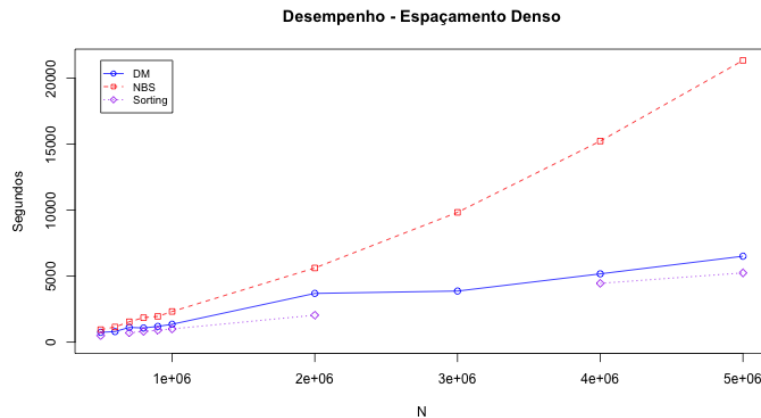


Figura 1. Tempo de execução dos algoritmos com empacotamento denso

provável motivo de tal melhora em desempenho é a simplicidade de suas estruturas usadas para o mapeamento, frente às listas encadeadas necessárias ao outro algoritmo.

Para solidificar o resultado é necessário finalizar os testes nos outros cenários de distribuição de partículas para obter mais conhecimento sobre o comportamento desses algoritmos.

Referências

- Araújo, J. P. N., Carvalho Júnior, H., Lira, W. W. M., and Ramos Junior, A. S. (2007). Estudo e implementação de um algoritmo de busca de contatos para problemas com elementos discretos planos de formas e dimensões arbitrárias. In *28st Iberian Latin American Congress on Computational Methods in Engineering*, pages 1–14, Porto, Portugal.
- Cundall, P. (1988). Formulation of a three-dimensional distinct element model—Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks. *International Journal of Rock Mechanics and Mining ...*, 25(3):107–116.
- Ericson, C. (2005). *Real-time collision detection*. Morgan Kaufmann Publishers, elsevier edition.
- Fattebert, J.-L., Richards, D., and Glosli, J. (2012). Dynamic load balancing algorithm for molecular dynamics based on Voronoi cells domain decompositions. *Computer Physics Communications*, 183(12):2608–2615.
- Han, K., Feng, Y. T., and Owen, D. R. J. (2007). Performance comparisons of tree-based and cell-based contact detection algorithms. *Engineering Computations*, 24(1-2):165–181.
- Markauskas, D. and Kačeniauskas, A. (2015). The comparison of two domain repartitioning methods used for parallel discrete element computations of the hopper discharge. *Advances in Engineering Software*, 84:68–76.
- Munjiza, A. (2004). *The Combined Finite-Discrete Element Method*. John Wiley & Sons.
- Perkins, E. and Williams, J. R. (2001). A fast contact detection algorithm insensitive to object sizes. *Engineering Computations*, 18(1-2):48–61.