

ESCALONAMENTO DE EQUIPES DE CAMPO DA ESCELSA ATUALIZAÇÃO BIBLIOGRÁFICA

RAUL H.C. LOPES

1. INTRODUÇÃO

Este projeto visa o desenvolvimento de um sistema de computação para alocação de equipes para atendimentos de ocorrências emergenciais. Para tal o sistema deverá possuir ao menos duas funcionalidades:

- definir a localização de equipes a partir de dados históricos de atendimento;
- realizar simulações a partir de localizações de equipes e dados históricos de ocorrências emergenciais previamente fornecidos.

Este documento, dentro do cronograma acordado para desenvolvimento do projeto, apresenta os resultados iniciais de levantamento bibliográfico realizados neste primeiro mês de desenvolvimento do projeto.

2. MODELOS FORMAIS PARA O PROBLEMA

Como estabelecido anteriormente, o sistema a ser desenvolvido apresenta ao menos duas funcionalidades distintas: a definição dos locais para estacionamento de equipes e a simulação de alocação de equipes para atendimentos a ocorrências emergenciais.

2.1. A definição de localização de equipes. Vários modelos formais podem ser encontrados na literatura para o problema da definição dos locais de estacionamento de equipes. O primeiro desses modelos é conhecido na literatura como *post office problem*: um entregador hipotético de encomendas deseja abrir novo posto de entregas em uma região e precisa decidir sua melhor localização. Esse mesmo problema apresenta-se quando uma rede de supermercados busca abrir nova loja em uma vizinhança. Em qualquer dos casos, serão levados em conta no processo de decisão custos de transporte dos diversos consumidores até à nova loja. Uma solução da geometria combinatorial para esse problema é dada via diagramas de Voronoi [6].

No entanto, outras soluções da literatura podem servir para modelar este problema. Uma solução clássica da literatura sobre problema do caixeiro viajante, conhecido com *TSP*, do inglês *Travelling Salesman*

Problem [15], consiste na de definição de *clusters* de cidades, neste projeto *clusters* de chaves, que definiriam regiões que poderiam ser preferencialmente percorridas pelo viajante. É importante ressaltar que analogia com o **TSP** pode ser útil também para o problema de despacho de equipes de atendimento. Técnicas de clusterização têm sido estudadas desde a década de 70, principalmente a partir do trabalho fundamental de Hartigan [16]. Fasulo [10] revisa o estado da arte e aplicações de algoritmos de clusterização, enquanto Eppstein[8] estuda clusterização com aplicações em computação geométrica e **TSP**. Edelsbrunner [7] e Preparata [23] serão referências fundamentais em termos de computação geométrica, bem como Overmars [22] que estabelece os fundamentos de construção de estruturas dinâmicas que certamente ocupam o cerne da construção dos algoritmos a desenvolver.

2.2. Alocação de equipes para atendimento. A alocação de equipes para atendimento a ocorrências emergenciais pode ser modelada essencialmente como um problema geral de k servidores. É um modelo teórico [3, 2, 21] em que se admite um espaço métrico (\mathcal{M}, d) , onde \mathcal{M} é um conjunto de pontos, e $k \geq |\mathcal{M}|$ e d definem uma métrica sobre \mathcal{M} . Dada uma seqüência de requisições $a = [r_0, r_1, \dots, r_n]$, cada r_i define um ponto de \mathcal{M} em que um serviço é requisitado. Uma requisição r_j é atendida no instante em que ela se apresenta se existir um servidor no ponto r_j , no momento em que a requisição é apresentada. A não existência de servidor no ponto r_j demanda o deslocamento de algum servidor de outro ponto $m_i \in \mathcal{M}$ para r_j , (m_j). O custo de tal deslocamento, denotado por d_{ij} , é dado pela função d , que codifica uma função de distância ($d : m_i \times m_j \in \mathcal{M} \longrightarrow \mathcal{R}$). É importante observar, no entanto, que essa função de distância não precisa e, no contexto colocado, não deve ser Euclidiana: o custo de deslocamento de m_i a m_j não é necessariamente igual ao custo de deslocamento de m_j para m_i e não tem, também, relação apenas com distância no sentido geográfico.

O problema de alocação de equipes pode ser visto de duas formas: a primeira seria quando se tem conhecimento de todas (ou parte) das requisições de serviços a serem realizados em um período de tempo dado. A segunda, quando as requisições são apresentadas ao programa gradualmente [3]. No primeiro caso, o algoritmo poderá avaliar **todas** as requisições, e sobre esta visão global das requisições, tomar uma decisão mais apropriada no sentido de melhor escalonar a seqüência de execução de cada requisição. Já na segunda situação, o algoritmo é *míope* e tem que tomar decisões a cada chegada de novo pedido de requisição. Desta forma, uma decisão para se atender uma requisição r_i e mais adiante outra r_j , por terem chegado nesta ordem, pode não ser a melhor ordem de decisão.

Algoritmos para problemas que recebem todas as requisições como parte de seus dados são denominados *offline*. Algoritmos que tomam decisões à medida em que as requisições forem chegando são denominados algoritmos *on-line* [3, 17]. Algoritmos *off-line* são usualmente usados como padrão de avaliação de competitividade das soluções para os problemas *online*. Dos algoritmos para solucionar problemas de *k-servers online*, dois parecem ter uma aplicação mais direta:

- **Algoritmo DC-tree** Desenvolvido em [4], esse algoritmo considera que os servidores estão dispostos em um grafo e que relação de vizinhança e custo de deslocamento são dados pelas arestas e pesos respectivos.
- **Algoritmo de k-taxicab** Apresentado em [11], aborda o problema de se apresentar solução para a situação em que dados dois pontos s, t no espaço, um servidor deve ser deslocado para s e, logo em seguida, para t .

Simulações sobre dados históricos de atendimento deverão ser realizadas. No processo de avaliação de resultados das simulações poderá ser conveniente utilizar a teoria clássica de avaliação de desempenho [18, 19, 20, 9, 12].

Uma linha completamente diferente para abordagem deste problema passa pelo uso de modelos baseados no conhecimento clássico de roteamento de veículos e **TSP** [5, 14, 15]. Neste caso, objetiva-se o menor caminho, ou tempo de atendimento, de m caixeiros viajantes ou veículos para atendimento a uma sequência de requisições.

A complexidade evidente da tarefa computacional envolvida leva-nos a buscar como alternativa de solução o uso de meta-heurísticas [1] e em especial de algoritmos baseados em *tabu search* [13], certamente a meta-heurística de maior sucesso no campo de soluções para **TSP**.

3. PRÓXIMOS PASSOS

A primeira fase da atualização bibliográfica está completa. Foram selecionados algoritmos que estão na fase atual em estudo detalhado. Além disso, assim que a assinatura do contrato entre ESCELSA e FEST o permitir, uma equipe de seis estudantes de graduação, um mestre e um doutor será contratada para começar de imediato a implementar e testar os algoritmos. A fase atual inclui as seguintes tarefas:

- estudo detalhado dos algoritmos indicados neste relatório;
- definição de uma arquitetura inicial de protótipo para implementação e teste dos algoritmos;
- nova rodada de atualização bibliográfica.

REFERÊNCIAS

1. J. Beasley, K. Dowsland, F. Glover, M. Laguna, C. Peterson, C.R. Reeves, and B. Söderberg, *Modern heuristic techniques for combinatorial problems*, Colin R. Reeves, 1992.
2. A. Borodin, N. Linial, and M. Saks, *An optimal online algorithm for metrical task systems*, Journal of the ACM **39** (1992), 745–763.
3. Alan Borodin and Ran El-Yaniv, *Online computation and competitive analysis*, Cambridge University Press, 1998.
4. M. Chrobak and L.L. Larmore, *An optimal on-line algorithm for k-servers on trees*, SIAM Journal on Computing **20** (1991), no. 1, 144–148.
5. William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver, *Combinatorial optimization*, John Wiley & Sons, Inc., 1998.
6. Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf, *Computational geometry*, Kluwer Academic Publishers, 1998.
7. Herbert Edelsbrunner, *Algorithms in combinatorial geometry*, Springer-Verlag, 1987.
8. David Eppstein, *Fast hierarchical clustering and other applications of dynamic closest pair*, Proceedings of SODA, 1998.
9. L. Breslau et al., *Advances in network simulation*, IEEE Computer **33** (2000), no. 5, 59–67.
10. Daniel Fasulo, *Ana analysis of recent work on clustering algorithms*, Tech. Report 01-03-02, Department of Computer Science, University of Washington, 1999.
11. A. Fiat, Y. Rabani, and Y. Ravid, *Competitive k-server algorithms*, IEEE, 1990, pp. 454–463.
12. S. Floyd and V. Paxson, *Difficulties in simulating the internet*, IEEE/ACM transactions on Networking **9** (2001), no. 4, 392–403.
13. Fred W. Glover and Manuel Laguna, *Tabu search*, Kluwer Academic Publishers, 1998.
14. Dan Gusfield and Robert W. Irving, *The stable marriage problem: structure and algorithms*, The MIT Press, 1989.
15. Gregory Gutin and Abraham P. Punen, *Traveling salesman problem and its variations*, Kluwer Academic Publishers, 2002.
16. John Hartigan, *Clustering algorithms*, JOHN WILEY & SONS, 1975.
17. Dietrich Hauptmeier, Sven O. Krumke, and Jörg Rambau, *The online dial-a-ride problem under reasonable load*, Tech. Report ZIB-Report 99-08, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1999.
18. Leonard Kleinrock, *Queueing systems, volume i: Theory*, JOHN WILEY & SONS, 1975.
19. ———, *Queueing systems, volume ii: Computer applications*, JOHN WILEY & SONS, 1976.
20. A. Law and W. Kelton, *Simulation modeling and analysis*, McGraw-Hill, 2000.
21. Rajeev Motwani and Prabhakar Raghavan, *Randomized algorithms*, Cambridge University Press, 1995.
22. Mark H. Overmars, *The design of dynamic data structures*, Springer-Verlag, 1983.
23. Franco P. Preparata and Michael Ian Shamos, *Computational geometry: An introduction*, Springer-Verlag, 1985.