



FABIO CHALUB  
Universidade  
Nova de Lisboa  
chalub@fct.unl.pt

## MATEMÁTICA EM DUAS RODAS

Pedalar é bom para o ambiente, para o bolso, exercita os músculos e... gera problemas interessantes de matemática. Numa das cidades com mais mobilidade alternativa do mundo, um grupo de matemáticos está a tentar tornar mais eficiente o sistema de aluguer por hora de bicicletas. Assim torna-se mais simples ir de um lado para o outro na musical cidade de Viena, capital da Áustria.

Com mais de uma centena de postos e 2500 bicicletas de aluguer, a Citybike Wien é uma das maiores e mais antigas empresas de serviços de transporte alternativo do mundo. Com um simples registo e com o fornecimento do número do cartão bancário ou de crédito, é possível a qualquer um utilizar uma bicicleta gratuitamente por uma hora, com um preço crescente no tempo a partir da segunda hora. Estas podem ser recolhidas e devolvidas em qualquer um das centenas de postos de serviço espalhados pela capital austríaca. Veja a figura 1.

Fazer o serviço funcionar, no entanto, exige algum esforço dos responsáveis. As deslocações tendem a ser da periferia para o centro na parte da manhã e na direção

contrária no final do dia. Outras características, como a topografia e as condições meteorológicas, induzem assimetria nas rotas. Desta forma, é necessário haver um sistema de transporte das bicicletas das estações cheias para as estações vazias. Para isto, há um serviço de carrinhas que circula durante todo o dia, carregando as bicicletas ociosas no contra fluxo dos utentes.

Ora, se há um serviço a ser feito, há um matemático a pensar em como este pode ser otimizado. E sendo Viena não só uma cidade amiga das duas rodas, mas também um dos locais com maior tradição científica no mundo, é natural que um grupo de cinco investigadores do Instituto of Computer Graphics and Algorithms da Universidade Técnica de Viena se debruçasse sobre o problema [1].

O primeiro passo é o de definir um objetivo para cada uma das diversas estações de serviço, mais especificamente, a quantidade de bicicletas que deve estar disponível em cada ponto. Ao contrário do que pode parecer à primeira vista, o ideal não é tê-lo cheio, pois ele serve também para devoluções. Ao não haver espaço para deixar a bicicleta, é necessário procurar outro – pagando o tempo necessário para isto. O valor ideal é fornecido pela própria administração da Citybike. A ocupação das diversas estações de serviço num certo momento é um dos dados iniciais do problema. Desta forma, define-se um grafo, onde cada ponto de apoio  $v$  é um nodo, ao qual associamos uma *capacidade máxima*  $C_v$ , um *valor corrente*

Figura 1. Um posto de serviço da Citybike Wien, temporariamente cheio. (Fonte:Wikimedia Commons).



$p_v$  e valor ideal  $q_v$ . Cada arco que liga dois nodos  $u$  e  $v$  é associado ao tempo  $t_{u,v}$  que uma carrinha demora a ir de um ponto ao outro. Há também um último nodo, representando o depósito, onde as carrinhas passam a noite.

As carrinhas têm de percorrer o grafo acima, carregando e descarregando bicicletas pelos diversos pontos de apoio de forma a rebalancear o sistema da melhor maneira possível e no mínimo tempo. Evidentemente, neste ínterim já há um novo desbalanceamento e o serviço deve recomeçar. Uma vez definido, o serviço de carrinhas não alterará as suas instruções até que o mesmo esteja completo. Sobre isto, dizemos que o processo de otimização é "estático". (A quantidade instantânea de bicicletas disponíveis pode ser consultada no <http://www.citybikewien.at>).

Evidentemente, há várias restrições ao serviço. As carrinhas têm uma capacidade máxima e também mínima (zero!). As suas rotas começam e terminam no depósito, e as viagens começam e terminam vazias, pois as bicicletas não podem passar a noite nas carrinhas estacionadas (o depósito de carrinhas é aberto ao público e não há forma prática de prender as bicicletas nas carrinhas). Algumas soluções podem ser particularmente curiosas, como uma carrinha deixar um certo número de bicicletas num ponto de apoio que logo a seguir será visitado por outra carrinha, que as recolherá. Além de haver certos inconvenientes práticos desta solução (como, por exemplo, um único atraso resultar num efeito em cascata), matematicamente é necessário impor que a cada momento o número de bicicletas disponíveis em cada posto seja um número maior ou igual a zero. Além disto, os estudos numéricos mostram que se considerarmos que numa volta completa de rebalanceamento, cada estação será apenas *carregada* ou *descarregada*, a solução obtida não é significativamente pior do que aquelas que usam as estações como local de armazenamento temporário de bicicletas. Dizemos que consideraremos apenas as soluções em que a disponibilidade de bicicletas em cada estação é uma função monótona do tempo.

Embora gostássemos de ter no final do serviço cada estação com o número ideal de bicicletas, isto frequentemente não é possível. Portanto, o objetivo é o de *minimizar* esta diferença. Leva-se em conta, também, mas como pesos significativamente inferiores (da ordem de uma centena de milésimo), o tempo necessário para o serviço, incluindo não só o tempo do transporte mas também o de carga e de descarga. Os pesos relativos foram definidos pelos serviços da empresa, que se mostrou muito

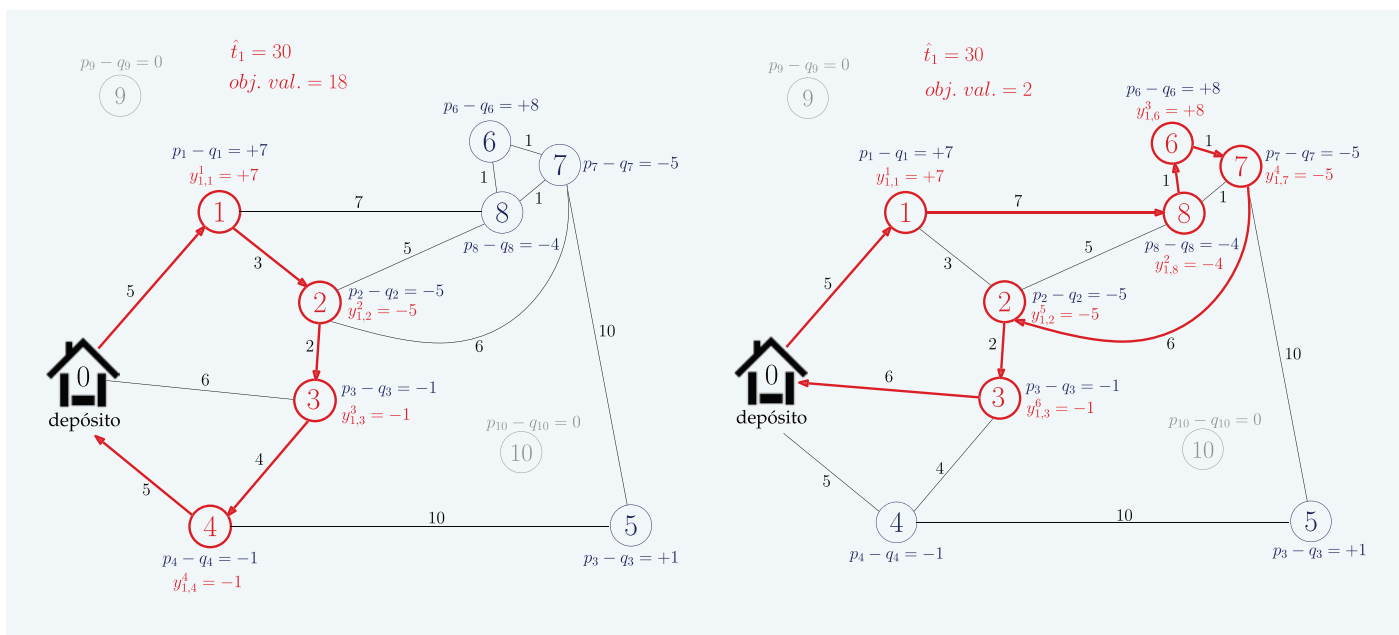
mais interessado em ter o número correto de veículos disponíveis aos utentes do que em minimizar os custos do rebalanceamento.

No entanto, não é possível fazer aquilo a que se chama *busca exaustiva*, ou seja, olhar todas as possibilidades para escolher a mais adequada. O espaço de soluções a ser investigado é tão amplo que é necessário definir um algoritmo de procura. Inicialmente, desenvolve-se um algoritmo ineficiente para desenhar uma primeira solução possível, e depois trata-se de melhorar esta solução introduzindo algumas perturbações no sistema. Veja a figura 2.

No primeiro caso, utiliza-se o algoritmo *glutão* (*greedy*): numa carrinha inicialmente vazia, vamos primeiro para as estações cheias, onde as bicicletas devem ser recolhidas. De cada estação segue-se para aquela que maximiza a quantidade a ser recolhida em relação ao tempo para lá chegar. Repete-se o processo até chegarmos à capacidade máxima do transporte e a seguir é iniciado o processo de entregas. De cada estação vai-se àquela onde a razão entre bicicletas entregues e tempo gasto é maximizado, até a carrinha estar esvaziada. Este algoritmo demora um longo tempo para gerar uma solução ineficiente, mas que poderia bem servir como ponto de partida para a segunda fase. No entanto, optou-se por usar um algoritmo (conhecido como *PILOT*) distinto para gerar uma solução melhor do que a do caso anterior (ainda que também ineficiente), e que é gerada mais rapidamente.

Para a segunda fase, na qual a solução inicialmente obtida é sucessivamente melhorada, usa-se o que é conhecido como *Variable Neighborhood Descent*, que procura soluções melhores, próximas da que já foi obtida, até atingir um ótimo local (ou seja, tal que pequenas modificações produzam sempre soluções piores). Em seguida, produz-se uma perturbação maior, de forma a escapar deste ótimo local e a procurar algum outro, que seja possivelmente ainda melhor. Assim se segue até se encontrar algo que seja satisfatório (mesmo que não seja, já que isto não é possível saber, o ótimo global – a melhor de todas as soluções).

O algoritmo foi testado com os dados reais, numa rede que à época tinha 92 pontos, com bons resultados. No entanto, para uma rede tão pequena, os primeiros algoritmos já forneciam uma resposta razoável e com instruções simples, e, de facto, próximo do que já era implementado de forma intuitiva pela Citybike Wien. As verdadeiras diferenças ocorriam com o aumento artificial da rede (até 700 nodos) mostrando também quais os algoritmos mais eficientes aquando do aumento da rede e em função das



restrições a serem implementadas (número de carrinhas, por exemplo). Assim, os resultados do trabalho são, de certa forma, modestos, mas mostram como, em problemas ainda mais complexos do que aquele com que estamos a lidar, a correta especificação do problema e dos seus algoritmos de solução pode ser crítica.

## REFERÊNCIAS

[1] Marian Rainer-Harbach, Petrina Papazek, Günther R. Raidl, Bin Hu, Christian Kloimüller. "PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems", *Journal of Global Optimization*, publicação online: 02 abril 2014.

## ERRATUM

Na figura 1 do último artigo ("Da Terceira Para A Quarta Dimensão", *Gazeta de Matemática* 174) há uma afirmação errada: a de que os únicos dados honestos são aqueles obtidos a partir dos sólidos platónicos. Há outras estruturas matemáticas que produzem dados onde a probabilidade de cair uma dada face para cima é independente da face, como por exemplo, a colagem pela base de duas pirâmides cujas faces (fora a base) são triângulos isósceles. Ser sólido regular é um pouco mais exigente, pois não só as faces devem ser idênticas, mas também os vértices e arestas. Agradeço ao leitor e colega Arala Chaves por me ter apontado o engano.

Figura 2. Dois exemplos de solução gerada inicialmente. No primeiro caso (acima), a carrinha segue o percurso ditado pelo algoritmo *glutão*, recolhendo sete bicicletas na primeira paragem, e deixando cinco, uma e um nas três paragens seguintes, antes de voltar ao depósito, menos de meia hora após iniciar o percurso. No segundo caso (abaixo), gerado pelo PILOT o funcionário da Citybike Wien recolhe sete, cinco, deixa quatro, volta a recolher oito e deixa cinco, cinco e uma, antes de voltar ao depósito em 28 minutos. Nos dois casos, a procura de soluções foi limitada àquelas em que a carrinha voltava ao depósito em, no máximo 30 minutos. Figuras gentilmente cedidas pelos autores do artigo [1] em particular G. Raidl.