On the determinant and its derivatives of the rank-one corrected generator of a Markov chain on a graph

J. A. Filar · M. Haythorpe · W. Murray

Received: 28 December 2011 / Accepted: 24 January 2012 / Published online: 4 February 2012 © Springer Science+Business Media, LLC. 2012

Abstract We present an algorithm to find the determinant and its first and second derivatives of a rank-one corrected generator matrix of a doubly stochastic Markov chain. The motivation arises from the fact that the global minimiser of this determinant solves the Hamiltonian cycle problem. It is essential for algorithms that find global minimisers to evaluate both first and second derivatives at every iteration. Potentially the computation of these derivatives could require an overwhelming amount of work since for the Hessian $N^2$ cofactors are required. We show how the doubly stochastic structure and the properties of the objective may be exploited to calculate all cofactors from a single LU decomposition.

Keywords Markov chain · Generator matrix · Derivative · Determinant · Doubly stochastic · LU decomposition · Rank-one correction · Hamiltonian cycle problem · Cofactors

1 Introduction

The Hamiltonian cycle problem (HCP) is an important graph theory problem that features prominently in complexity theory because it is NP-complete [3]. HCP has also gained recognition because two special cases: the Knight’s tour and the Icosian game, were solved by
Euler and Hamilton, respectively. Finally, HCP is closely related to the well known Traveling Salesman problem.

The definition of HCP is the following: given a graph $\Gamma$ containing $N$ nodes, determine whether any simple cycles of length $N$ exist in the graph. These simple cycles of length $N$ are known as Hamiltonian cycles. If $\Gamma$ contains at least one Hamiltonian cycle (HC), we say that $\Gamma$ is a Hamiltonian graph. Otherwise, we say that $\Gamma$ is a non-Hamiltonian graph.

While there are many graph theory techniques that have been designed to solve HCP, another common approach is to associate a variable $x_{ij}$ with each arc $(i, j) \in \Gamma$, and solve an associated optimisation problem. A convenient method of representing these constraints is to use a matrix $P(x)$, where

$$p_{ij}(x) = \begin{cases} x_{ij}, & (i, j) \in \Gamma, \\ 0, & \text{otherwise.} \end{cases}$$

The discrete nature of HCP naturally lends itself to integer programming optimisation problems. However, arising from an embedding of HCP in a Markov decision process (a practice initiated by Filar et al [2]), continuous optimisation problems that are equivalent to HCP have been discovered in recent times. In particular, it was demonstrated in [1] that if we define

$$A(P(x)) = I - P(x) + \frac{1}{N}ee^T,$$

where $e$ is a column vector with unit entries, then HCP is equivalent to solving the following optimisation problem:

$$\min -\det(A(P(x)))$$

subject to

$$\sum_{j \in A(i)} x_{ij} = 1, \quad i = 1, \ldots, N,$$

$$\sum_{i \in A(j)} x_{ij} = 1, \quad j = 1, \ldots, N,$$

$$x_{ij} \geq 0, \quad \forall (i, j) \in \Gamma,$$

where $A(i)$ is the set of nodes reachable in one step from node $i$. Constraints (2)–(4) are called the doubly-stochastic constraints. For neatness, we refer to constraints (2)–(4) as the set $DS$, and we call the objective function $f(P(x))$. Then, the above problem can be represented as follows:

$$\min \{ f(P(x)) | x \in DS \}.$$

Note that we need to find a global minimiser and, typically, there are many of them. However, the number of global minimisers is typically extremely small compared to the number of local minimisers. One consequence of multiple global minimisers is that there is a similarly large number of stationary points. To distinguish such stationary points from minimisers, algorithms (see for example [4]) require the use of second derivatives. It is not hard to appreciate that evaluating such derivatives will be expensive even for moderately-sized problems unless some special structure is identified. We exploit the structure of the Hessian and the fact the points at which it is evaluated are in $DS$. We assume an algorithm to solve this problem starts at a feasible point and all iterates remain feasible. For problems with nonlinear objectives this is almost always the best approach. Finding a point in $DS$ is simple and does not involve evaluating the objective or its derivatives.