BLUEBIT ASD ALGORITHM

T. TRIANTAFILLIDIS

1. Introduction

1.1. **Problem Definition.** The minimax theorem proved by John von Neumann in 1928 states that for every $m \times n$ matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and probability vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$

$$\mathbf{x} \in \mathcal{X} := \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{j=1}^{n} x_j = 1 \right\} \tag{1.1}$$

$$\mathbf{y} \in \mathcal{Y} := \left\{ \mathbf{y} \in \mathbb{R}^m : \sum_{i=1}^{m} y_i = 1 \right\} \tag{1.2}$$

the following relation holds

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{y}'\mathbf{A}\mathbf{x} = \min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} \mathbf{y}'\mathbf{A}\mathbf{x} \tag{1.3}$$

We call the vectors $\mathbf{x}^*, \mathbf{y}^*$ a minimax solution of $\mathbf{A}$ if they satisfy (1.3). The scalar $v^* = (\mathbf{y}^*)'\mathbf{A}\mathbf{x}^*$ is the value at the the equilibrium point and in a game theory context it is called the ***game value***. For any other vectors $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$ it will be

$$\mathbf{y}'\mathbf{A}\mathbf{x}^* \geq v^* = (\mathbf{y}^*)'\mathbf{A}\mathbf{x}^* \geq (\mathbf{y}^*)'\mathbf{A}\mathbf{x} \quad \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y} \tag{1.4}$$

Finding one (not necessarily unique) pair of vectors $\mathbf{x}^*, \mathbf{y}^*$ satisfying (1.4) solves the minimax problem.

We call a ***pure strategy*** any probability vector for which

$$x_{j=k} = 1,\ x_{j \neq k} = 0,\ 1 \leq k \leq n \tag{1.5}$$

$$y_{i=k} = 1,\ y_{i \neq k} = 0,\ 1 \leq k \leq m \tag{1.6}$$

A pure strategy for $\mathbf{y}$ can always be applied in (1.3), therefore we may conclude that $\mathbf{x}^*$ is not optimal unless

$$\rho^* = \min_{0 \leq i \leq m} \mathbf{A}\mathbf{x}^* = v^* \tag{1.7}$$

and also for the same reason $\mathbf{y}^*$ is not optimal unless

$$\gamma^* = \max_{0 \leq j \leq n} (\mathbf{y}^*)'\mathbf{A} = v^* \tag{1.8}$$

therefore

$$\rho^* = \gamma^* = v^* \tag{1.9}$$

Date: June 01, 2007.

It can be easily shown that the reverse statement is also true. If for any probability vectors $\mathbf{x}, \mathbf{y}$

$$\rho = \min_{0 \leq i \leq m} \mathbf{A}\mathbf{x} = \max_{0 \leq j \geq n} \mathbf{y}'\mathbf{A} = \gamma \tag{1.10}$$

then the vectors $\mathbf{x}, \mathbf{y}$ consist a minimax solution.

Obviously for any pair of non optimal vectors $\mathbf{x}, \mathbf{y}$ it will be

$$\rho = \min_{0 \geq i \geq m} \mathbf{A}\mathbf{x} \leq v^* \leq \max_{0 \geq j \geq n} \mathbf{y}'\mathbf{A} = \gamma \tag{1.11}$$

with $\gamma > \rho$. We call the positive difference

$$d = \gamma - \rho \geq 0 \tag{1.12}$$

the ***duality gap***. Any algorithm which gradually reduces the duality gap to zero, solves the minimax problem.

2. The new Algorithm

2.1. **Preliminaries.** We are given a $m \times n$ matrix $\mathbf{A}$ and we are asked to compute a minimax solution for this matrix. Without loss of generality we will assume that $\mathbf{A}$ contains elements within the range $[0, 1]$. If not, we may apply a transformation to all matrix elements so that

$$a_{i,j} = \frac{a_{i,j} - a_{min}}{a_{max} - a_{min}} \tag{2.1}$$

where a_{min}, a_{max} denote the minimum and the maximum of the matrix elements respectively. Let $\mathbf{U}$ be a $m \times n$ matrix with every elements equal to 1. It can be easily shown that any matrix $\mathbf{B}$ in the form

$$\mathbf{B} = c_1 \cdot (\mathbf{A} + c_2 \cdot \mathbf{U}) \tag{2.2}$$

shares the same minimax solutions as matrix $\mathbf{A}$. Selecting suitable constants c_1, c_2 can ensure that all matrix elements will fall within the range $[0, 1]$.

2.2. **The Algorithm.** With the assumption that $\mathbf{A}$ contains elements in the range $[0, 1]$ the following algorithm minimizes the duality gap.

Algorithm 1: Bluebit (US Patent 7,991,713 B2 - international patents pending)

---

input : $m \times n$ matrix $\mathbf{A}$, number of iterations T

output: mixed strategies $\mathbf{y}^* \in \mathbf{R}^m, \mathbf{x}^* \in \mathbf{R}^n$, duality gap d^*

begin

  $x_j \leftarrow 1/n \quad \forall\, 1 \leq j \leq n$

  $y_i \leftarrow 1/m \quad \forall\, 1 \leq i \leq m$

  $\mathbf{h} \leftarrow \mathbf{Ax}$

  $\mathbf{g} \leftarrow \mathbf{y}'\mathbf{A}$

  $\rho \leftarrow \min \mathbf{h}$

  $\gamma \leftarrow \max \mathbf{g}$

  $\rho_{max} \leftarrow \rho$

  $\gamma_{min} \leftarrow \gamma$

  $v \leftarrow \dfrac{\gamma_{min} + \rho_{max}}{2}$

  for $t = 1$ *to* T **do**

    $\Delta x_j \leftarrow (g_j - v) \cdot [g_j > v]$

    $\mathbf{x} \leftarrow (1 - \gamma + \rho) \cdot \mathbf{x} + (\gamma - \rho) \cdot \dfrac{\Delta \mathbf{x}}{\sum_{j=1}^{n} \Delta x_j}$

    $\mathbf{h} \leftarrow \mathbf{Ax}$

    $\rho \leftarrow \min \mathbf{h}$

    if $\rho > \rho_{max}$ **then**

      $\rho_{max} \leftarrow \rho$

      $\mathbf{x}^* \leftarrow \mathbf{x}$

      $v \leftarrow \dfrac{\gamma_{min} + \rho_{max}}{2}$

    end if

    $\Delta y_i \leftarrow (v - h_i) \cdot [h_i < v]$

    $\mathbf{y} \leftarrow (1 - \gamma + \rho) \cdot \mathbf{y} + (\gamma - \rho) \cdot \dfrac{\Delta \mathbf{y}}{\sum_i \Delta y_i}$

    $\mathbf{g} \leftarrow \mathbf{y}'\mathbf{A}$

    $\gamma \leftarrow \max \mathbf{g}$

    if $\gamma < \gamma_{min}$ **then**

      $\gamma_{min} \leftarrow \gamma$

      $\mathbf{y}^* \leftarrow \mathbf{y}$

      $v \leftarrow \dfrac{\gamma_{min} + \rho_{max}}{2}$

    end if

  end for

  $d^* = \gamma_{min} - \rho_{max}$

end

2.3. **Description.**

2.3.1. *Lines 2-3.* In the initialization part of the algorithm we initialize all elements of $\mathbf{x}$ to $1/n$ and all elements of $\mathbf{y}$ to $1/m$. Any other probability distribution can be used to initialize the vectors $\mathbf{x}, \mathbf{y}$.

2.3.2. *Lines 4-5.* We create $\mathbf{h}$, a m dimensional vector as the result of the matrix-vector multiplication $\mathbf{Ax}$. Therefore each element of $\mathbf{h}$ will be equal to

$$h_i = \sum_{j=1}^{n} a_{i,j} x_j \quad \forall\, 0 \leq i \leq m$$

In the same way we create $\mathbf{g}$, a n dimensional vector being the result of the vector-matrix multiplication $\mathbf{y'A}$, having each of its elements equal to

$$g_j = \sum_{i=1}^{m} a_{i,j} y_i \quad \forall\, 0 \leq j \leq n$$

2.3.3. *Lines 6-9.* We set ρ to the minimum element of the vector $\mathbf{h}$ and γ to the maximum element of the vector $\mathbf{g}$. We also initialize ρ_{max} to ρ and γ_{min} to γ.

2.3.4. *Line 10.* We define v as the middle point of γ_{min} and ρ_{max}.

2.3.5. *Line 11-30.* We repeat for a number of T iterations.

2.3.6. *Lines 12-13.* We define n-dimensional vector $\Delta\mathbf{x}$ as an update step for the vector $\mathbf{x}$. We set each Δx_j equal to

$$\Delta x_j = \begin{cases} g_j - v & \text{if } g_j > v \\ 0 & \text{if } g_j \leq v \end{cases}$$

We then normalize $\Delta\mathbf{x}$ so that $\sum_{j=1}^{n} \Delta x_j = 1$ and we update $\mathbf{x}$ as

$$\mathbf{x} \leftarrow (1-d) \cdot \mathbf{x} + d \cdot \Delta\mathbf{x}$$

where $d = \gamma - \rho$ is the current duality gap.

2.3.7. *Lines 14-15.* We compute the new value for $\mathbf{h}$ using the updated value of $\mathbf{x}$ and also we update the value of ρ as $\min \mathbf{h}$

2.3.8. *Lines 16-20.* If the previous update of $\mathbf{x}$ has achieved a better (bigger) ρ, then we update the value of ρ_{max}, we use this new value of ρ_{max} to update v and we record $\mathbf{x}^*$ as the best up to now value for $\mathbf{x}$.

In the second part of the iteration we repeat the same actions for $\mathbf{y}$ in an symmetric way except that the inequalities and signs are reversed.

2.3.9. *Lines 21-22.* We define a m-dimensional vector $\Delta\mathbf{y}$ as an update step for $\mathbf{y}$ with each Δy_i equal to

$$\Delta y_i = \begin{cases} v - h_i & \text{if } h_i < v \\ 0 & \text{if } h_i \geq v \end{cases}$$

We then normalize $\Delta\mathbf{y}$ so that $\sum_{i=1}^{m} \Delta y_i = 1$ and we update $\mathbf{y}$ as

$$\mathbf{y} \leftarrow (1-d) \cdot \mathbf{y} + d \cdot \Delta\mathbf{y}$$

where $d = \gamma - \rho$ is the current duality gap.

2.3.10. *Lines 23-24.* We compute the new value for $\mathbf{g}$ using the updated value of $\mathbf{y}$ and also we update the value of γ as $\max \mathbf{g}$

2.3.11. *Lines 25-29.* If the previous update of $\mathbf{y}$ has achieved a better (smaller) γ, then we update the value of γ_{min}, we use this new value of γ_{min} to update v and we record $\mathbf{y}^*$ as the best up to now value for $\mathbf{y}$.

2.3.12. *Line 30.* The duality gap achieved is $\gamma_{min} - \rho_{max}$

3. Upper Bound for the Duality Gap

Numerical experiments on a big number of random matrices have shown that for square matrices ($m = n$) the duality gap achieved by the algorithm ($\gamma_{min} - \rho_{max}$) is upper bounded by $1/T$ where T denotes the number of iterations. For non-square matrices this also holds when $T > \max\{m, n\}$. Figure 1 displays a graph of the duality gap together with this upper limit versus the number of iterations.

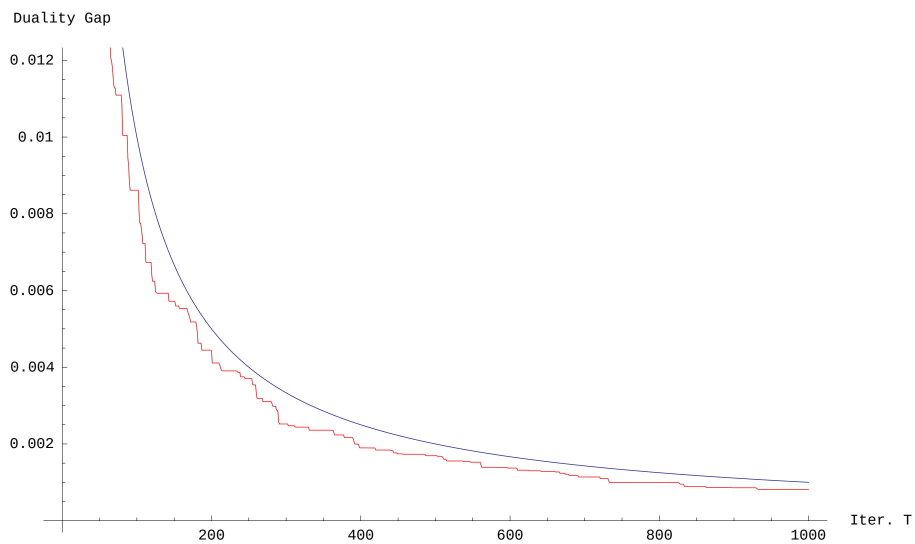

Figure 1. The duality gap $d = \gamma_{min} - \rho_{max}$ (red line) and its upper bound $1/T$ (blue line) versus the number of iterations for a random 100×100 matrix.

Trifon Triantafillidis
26 Kofidou,
55236 Panorama,
Thessaloniki, Greece
E-mail address, Trifon Triantafillidis: trifon@bluebit.gr
URL: http://www.bluebit.gr