1. Introduction

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

\begin{align*}
    x \in X := \left\{ x \in \mathbb{R}^{n} : \sum_{j=1}^{n} x_{j} = 1 \right\} \\
    y \in Y := \left\{ y \in \mathbb{R}^{m} : \sum_{i=1}^{m} y_{i} = 1 \right\}
\end{align*}

the following relation holds

\begin{align*}
    \max_{x \in X} \min_{y \in Y} y^{\prime}Ax &= \min_{y \in Y} \max_{x \in X} y^{\prime}Ax
\end{align*}

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

\begin{align*}
    y^{\prime}Ax^{\ast} \geq v^{\ast} = (y^{\ast})^{\prime}Ax^{\ast} \geq (y^{\ast})^{\prime}A x \quad \forall x \in X, \forall y \in Y
\end{align*}

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

We call a pure strategy any probability vector for which

\begin{align*}
    x_{j=k} = 1, \ x_{j\neq k} = 0, \ 1 \leq k \leq n \\
    y_{i=k} = 1, \ y_{i\neq k} = 0, \ 1 \leq k \leq m
\end{align*}

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

\begin{align*}
    \rho^{\ast} = \min_{0 \leq i \leq m} A x^{\ast} = v^{\ast}
\end{align*}

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

\begin{align*}
    \gamma^{\ast} = \max_{0 \leq j \leq n} (y^{\ast})^{\prime}A = v^{\ast}
\end{align*}

therefore

\begin{align*}
    \rho^{\ast} = \gamma^{\ast} = v^{\ast}
\end{align*}
It can be easily shown that the reverse statement is also true. If for any probability vectors \( \mathbf{x}, \mathbf{y} \)

\[
(1.10) \quad \rho = \min_{0 \leq i \leq m} \mathbf{Ax} = \max_{0 \leq j \geq n} \mathbf{y}'\mathbf{A} = \gamma
\]

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

\[
(1.11) \quad \rho = \min_{0 \geq i \geq m} \mathbf{Ax} \leq v^* \leq \max_{0 \geq j \geq n} \mathbf{y}'\mathbf{A} = \gamma
\]

with \( \gamma > \rho \). We call the positive difference

\[
(1.12) \quad \delta = \gamma - \rho \geq 0
\]

the \textit{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

\[
(2.1) \quad a_{i,j} = \frac{a_{i,j} - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}}
\]

where \( a_{\text{min}}, a_{\text{max}} \) 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

\[
(2.2) \quad \mathbf{B} = c_1 \cdot (\mathbf{A} + c_2 \cdot \mathbf{U})
\]

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 $A$ 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)

```plaintext
input : $m \times n$ matrix $A$, number of iterations $T$
output: mixed strategies $y^* \in \mathbb{R}^m, x^* \in \mathbb{R}^n$ , duality gap $d^*$

1 begin
2    $x_j \leftarrow 1/n \quad \forall 1 \leq j \leq n$
3    $y_i \leftarrow 1/m \quad \forall 1 \leq i \leq m$
4    $h \leftarrow Ax$
5    $g \leftarrow y'A$
6    $\rho \leftarrow \min h$
7    $\gamma \leftarrow \max g$
8    $\rho_{max} \leftarrow \rho$
9    $\gamma_{min} \leftarrow \gamma$
10   $v \leftarrow \gamma_{min} + \rho_{max}$
11 for $t = 1$ to $T$ do
12    $\Delta x_j \leftarrow (g_j - v) \cdot [g_j > v]$
13    $x \leftarrow (1 - \gamma + \rho) \cdot x + (\gamma - \rho) \cdot \frac{\Delta x}{\sum_{j=1}^n \Delta x_j}$
14    $h \leftarrow Ax$
15    $\rho \leftarrow \min h$
16    if $\rho > \rho_{max}$ then
17        $\rho_{max} \leftarrow \rho$
18        $x^* \leftarrow x$
19        $v \leftarrow \gamma_{min} + \rho_{max}$
20    end if
21    $\Delta y_i \leftarrow (v - h_i) \cdot [h_i < v]$
22    $y \leftarrow (1 - \gamma + \rho) \cdot y + (\gamma - \rho) \cdot \frac{\Delta y}{\sum_i \Delta y_i}$
23    $g \leftarrow y'A$
24    $\gamma \leftarrow \max g$
25    if $\gamma < \gamma_{min}$ then
26        $\gamma_{min} \leftarrow \gamma$
27        $y^* \leftarrow y$
28        $v \leftarrow \gamma_{min} + \rho_{max}$
29    end if
30 end for
31 $d^* = \gamma_{min} - \rho_{max}$
32 end
```
2.3. Description.

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

2.3.2. Lines 4-5. We create \( h \), an \( m \) dimensional vector as the result of the matrix-vector multiplication \( Ax \). Therefore each element of \( h \) 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 \( g \), a \( n \) dimensional vector being the result of the vector-matrix multiplication \( y' A \), 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 \( \rho \) to the minimum element of the vector \( h \) and \( \gamma \) to the maximum element of the vector \( g \). We also initialize \( \rho_{max} \) to \( \rho \) and \( \gamma_{min} \) to \( \gamma \).

2.3.4. Line 10. We define \( v \) as the middle point of \( \gamma_{min} \) and \( \rho_{max} \).

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 \( \Delta x \) as an update step for the vector \( x \). We set each \( \Delta x_j \) 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 \( \Delta x \) so that \( \sum_{j=1}^{n} \Delta x_j = 1 \) and we update \( x \) as
\[
x \leftarrow (1 - d) \cdot x + d \cdot \Delta x
\]
where \( d = \gamma - \rho \) is the current duality gap.

2.3.7. Lines 14-15. We compute the new value for \( h \) using the updated value of \( x \) and also we update the value of \( \rho \) as \( \min h \)

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

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

2.3.9. Lines 21-22. We define \( m \)-dimensional vector \( \Delta y \) as an update step for \( y \) with each \( \Delta y_i \) 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 \( \Delta y \) so that \( \sum_{i=1}^{m} \Delta y_i = 1 \) and we update \( y \) as
\[
y \leftarrow (1 - d) \cdot y + d \cdot \Delta y
\]
where \( d = \gamma - \rho \) is the current duality gap.
2.3.10. **Lines 23-24.** We compute the new value for $g$ using the updated value of $y$ and also we update the value of $\gamma$ as $\max g$.

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

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

### 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 ($\gamma_{\text{min}} - \rho_{\text{max}}$) 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.

![Duality Gap Graph](image-url)

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

**Trifon Triantafillidis**
26 Kofidou, 55236 Panorama, Thessaloniki, Greece

*E-mail address*, Trifon Triantafillidis: trifon@bluebit.gr

*URL*: [http://www.bluebit.gr](http://www.bluebit.gr)