5-1-2016

Expected Worth for 2 × 2 Matrix Games with Variable Grid Sizes

Michael R. Powers

Martin Shubik

Wen Wang

EXPECTED WORTH FOR 2 × 2 MATRIX GAMES WITH VARIABLE GRID SIZES

By

Michael R. Powers, Martin Shubik, and Wen Wang

May 2016
Revised October 2016

COWLES FOUNDATION DISCUSSION PAPER NO. 2039R

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS
YALE UNIVERSITY
Box 208281
New Haven, Connecticut 06520-8281

http://cowles.yale.edu/

Expected Worth for 2×2 Matrix Games with Variable Grid Sizes

Michael R. Powers,[*] Martin Shubik,[†] and Wen Wang[‡]

October 16, 2016

Abstract

We offer a detailed examination of a broad class of 2×2 matrix games as a first step toward considering measures of resource distribution and efficiency of outcomes. In the present essay, only noncooperative equilibria and entropic outcomes are considered, and a crude measure of efficiency employed. Other solution concepts and the formal construction of an efficiency index will be addressed in a companion paper.

JEL Classifications: C63, C72, D61
Keywords: 2×2 matrix games, efficiency, coordination, worth of coordination.

1 2×2 Matrix Games with Cardinal Payoffs

In the folklore and elementary pedagogy of game theory, the 2×2 matrix game plays a special role. Several of these games bear well-known names, such as the Prisoner's Dilemma, Stag Hunt, and Battle of the Sexes. Although there are only 144 strategically different 2×2 games with strictly ordinal preferences, one often is interested in considering related games with cardinal preferences, whose number is unbounded. The present paper is devoted to addressing applications in which it is desirable to examine a large but finite set of 2×2 games with cardinal preferences.

[*]Zurich Group Professor of Risk Mathematics, School of Business and Management, and Professor, Schwarzman Scholars Program, Tsinghua University.

[†]Seymour Knox Professor of Mathematical Institutional Economics (Emeritus), Yale University, and External Faculty, Santa Fe Institute.

[‡]International Business School, Nankai University.

1.1 Outcome Sets

A generic 2×2 game is described by the matrix shown in Table 1. Here, the row player has the two strategies, "Up" and "Down" (corresponding to rows $i = 1, 2$, respectively), whereas the column player has "Left" and "Right" (corresponding to columns $j = 1, 2$, respectively). This yields four possible payoff pairs (outcomes), $(a_{i,j}, b_{i,j})$, for $i = 1, 2$ and $j = 1, 2$.

Table 1: A Generic 2×2 Matrix Game

	Left ($j = 1$)	Right ($j = 2$)
Up ($i = 1$)	$a_{1,1}, b_{1,1}$	$a_{1,2}, b_{1,2}$
Down ($i = 2$)	$a_{2,1}, b_{2,1}$	$a_{2,2}, b_{2,2}$

The universe of all strictly ordinal games is easily denumerated by noting that each of the payoff vectors $[a_{11}, a_{12}, a_{21}, a_{22}]$ and $[b_{11}, b_{12}, b_{21}, b_{22}]$ must be permutations of the ordinal vector $[1, 2, 3, 4]$, yielding a total of $4! \times 4! = 576$ different outcomes. This number can be divided by 2 to remove duplications arising from interchanging rows, and by another 2 to account for interchanging columns, leaving the canonical 144 strategically distinct games shown in Appendix 1. Topologically, the outcome sets of these games may be characterized by a smaller set of 24 distinct shapes, 22 of which are two-dimensional (i.e., games of opposition) and the remaining 2 one-dimensional (i.e., games of coordination). These shapes are shown in detail in Appendix 1, where they are associated with the 144 games.

In considering cardinal games, we assume that the payoff pairs, $(a_{i,j}, b_{i,j})$, may be expressed in well-defined units of money or gold, with a fixed minimal level of fineness that can be perceived and/or traded.[1] Is there an upper bound on how large an individual payoff can be? Philosophically, one could argue in either direction; but for all practical purposes, one can impose a large enough upper bound that encompasses all possible observations for a given society. We therefore investigate a closed set of 2×2 matrix games with payoffs given by elements in the set $\{1/2^{k-1}, 2/2^{k-1}, \ldots, 4\}$, for $k \in \{1, 2, ...\}$, with a grid size of $\Delta = 1/2^{k-1}$. Equivalently, one might choose the payoff set $\{1, 2, \ldots, 2^{k-1} \times 4\}$, with a grid size of $\Delta = 1$. Although the latter approach offers the simplicity

[1] We do not concern ourselves with individual preferences directly, but allow for the possibility that each amount of money/gold is mapped onto individual preferences in some risk-averse manner.

of an easily comprehensible fixed grid size, the former provides both a bounded maximum payoff size and an intuitively straightforward limiting process to assess the impact of grid size on player behavior.

In the remainder of the paper, we study various properties of cardinal games arranged into 144 categories associated with their corresponding strictly ordinal games. Our investigation relies on both analytical and simulation methods. In the latter case, we employ a computer program that carries out the following steps for each of games $G = 1, 2, \ldots, 100,000$, for a given value $k \in \{1, 2, ...\}$:

1. For each of the four cells, $(i, j) = (1, 1), (1, 2), (2, 1), (2, 2)$, generate two independent random variables, $a_{i,j} \sim \text{Uniform}\left\{1/2^{k-1}, 2/2^{k-1}, \ldots, 4\right\}$ and $b_{i,j} \sim \text{Uniform}\left\{1/2^{k-1}, 2/2^{k-1}, \ldots, 4\right\}$, where the four pairs $(a_{i,j}, b_{i,j})$ are mutually independent.

2. If either $a_{i,j} = a_{i',j'}$ or $b_{i,j} = b_{i',j'}$ for any $(i, j) = (i', j')$, then reject the game and return to step (1).

3. Define the cardinal 2×2 game G by the four payoff pairs $(a_{i,j}, b_{i,j})$.

4. Separately order the four $a_{i,j}$ and four $b_{i,j}$ from lowest to highest, and let $\tilde{a}_{i,j} = \text{rank}(a_{i,j}) \in \{1, 2, 3, 4\}$ and $\tilde{b}_{i,j} = \text{rank}(b_{i,j}) \in \{1, 2, 3, 4\}$ for all (i, j).

5. Define the ordinal 2×2 game G by the four payoff pairs $\left(\tilde{a}_{i,j}, \tilde{b}_{i,j}\right)$, and match this game to one of the 144 canonical strictly ordinal games.

By symmetry, we know that the number of ordinal games generated for each of the 144 canonical forms will be approximately the same.

1.2 Mass Properties

The generation of a large number of distinct cardinal games, each associated with one of the 144 canonical ordinal games, provides the means to consider the mass properties of several approaches to game play. A solution is the outcome (or set of outcomes) derived by the selection of a strategy by each of the game's players, and may be based upon a wide array of individual player characteristics.

For the present, however, we will limit consideration to (1) noncooperative, individually optimizing players, and (2) entropy players selecting each row or column randomly, with probability 1/2.

2 Joint Maximum Payoffs

Given a set of cardinal games generated randomly, as above, it is natural to consider the distribution of the joint maximum payoff, $JM_k = \max_{i\in\{1,2\},j\in\{1,2\}} \{a_{i,j} + b_{i,j}\}$, for a given $k \in \{1, 2, ...\}$, and easy to see that the sample space of JM_k is given by the set of values $\{5/2^{k-1}, 6/2^{k-1}, \ldots, 8\}$.

2.1 Distribution for $k = 1$

For the case of $k = 1$,[2] this sample space is simply the set of integers $\{5, 6, 7, 8\}$, and it is useful to associate these joint maxima with each of three game categories: (1) games of coordination, for which $JM_1 = 8$; (2) mixed-motive games, for which $JM_1 \in \{7, 6\}$; and (3) games of opposition, for which $JM_1 = 5$. For this baseline case, one can work out the distribution of the joint maximum as in Table 2, from which it is clear that JM_1 is negatively skewed, with mean, median, and mode of 6.875, 7, and 7, respectively.

Table 2: Distribution of JM_1

Value	# of Games
8	36
7	60
6	42
5	6
Total	144

2.2 Distribution for $k > 1$

For $k > 1$, the distribution of JM_k is more complex, but much can be learned simply by considering the limiting case as $k \to \infty$. Letting $a_{i,j}$ and $b_{i,j}$ be independent and identically distributed (continuous) Uniform $(0, 4]$ random variables for all (i, j), one can define $JM_\infty = \max_{i\in\{1,2\},j\in\{1,2\}} \{a_{i,j} + b_{i,j}\}$,

[2] We note that for $k = 1$, the sample space of the random cardinal games is identical to the set of 144 canonical ordinal games.

and observe that $JM_\infty = \max\{X_1, X_2, X_3, X_4\}$, where $X_\ell \sim$ i.i.d. Triangular $[0, 8]$; i.e.,

$$F_X(x) = \begin{cases} x^2/32 & \text{for } x \in [0,4] \\ -x^2/32 + x/2 - 1 & \text{for } x \in (4,8] \end{cases}.$$

It then follows that

$$F_{JM_\infty}(y) = \begin{cases} \left(y^2/32\right)^4 & \text{for } y \in [0,4] \\ \left(y^2/32 - y/2 + 1\right)^4 & \text{for } y \in (4,8] \end{cases}$$

and

$$f_{JM_\infty}(y) = \begin{cases} \left(y^2/32\right)^3 (y/4) & \text{for } y \in [0,4] \\ \left(y^2/32 - y/2 + 1\right)^3 (y/4 - 2) & \text{for } y \in (4,8] \end{cases}.$$

This probability density function, plotted in Figure 1, shows that the distribution of JM_∞ is negatively skewed, with mean, median, and mode given approximately by 5.6968, 5.7436, and 5.8619, respectively.[3]

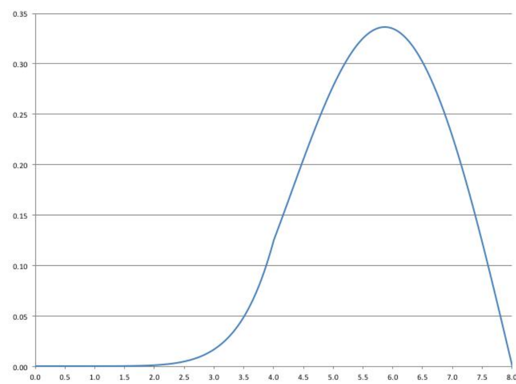


Figure 1: Probability Density Function of JM_∞

[3] These parameter values are calculated as follows: $Mean = \int_0^8 [1 - F_{JM_\infty}(y)]\,dy = \int_0^4 \left[1 - \left(y^2/32\right)^4\right] dy + \int_4^8 \left[1 - \left(y^2/32 - y/2 + 1\right)^4\right] dy = 3{,}589/630 \approx 5.6968$; $Median = F_{JM_\infty}^{-1}(1/2) = \left\{m : \left(m^2/32 - m/2 + 1\right)^4 = 1/2\right\} = 8 - 4\sqrt{2 - \sqrt[4]{8}} \approx 5.7436$; and $Mode = \arg\max_{y \in [0,8]} \{f_{JM_\infty}(y)\} = \operatorname{root}_{y \in (4,8]} \left\{\left(y^2/32 - y/2 + 1\right)^2 \left(7y^2/32 - 7y/2 + 13\right)\right\} = 8 - 8/\sqrt{14} \approx 5.8619$.

The above analysis reveals that the shape of the distribution of the joint maximum remains negatively skewed for large values of k, just as it is for $k = 1$. One noteworthy difference, however, is that games of coordination and games of opposition become less and less probable, approaching sets of measure zero as $k \to \infty$.

3 Noncooperative Equilibrium

The concept of noncooperative equilibrium has existed in the economic literature since the work of Augustin Cournot (1836), but was mathematically fully formalized and generalized by John Nash (1952). We define the outcome of a 2×2 matrix game as an ordered pair of strategies, $(s_{\mathrm{R}}, s_{\mathrm{C}})$, in which the first element denotes the row player's method of selecting a row ($i \in \{1, 2\}$), and the second element denotes the column player's method of selecting a column ($j \in \{1, 2\}$).[4] We further define a noncooperative equilibrium as an outcome in which each player has no motivation to change his or her strategy, given the indicated strategy of the other player. Restricting attention to pure strategies, in which each player's decision consists of a fixed (as opposed to random) choice of row or column, one can see that $(s_{\mathrm{R}}^*, s_{\mathrm{C}}^*) = (i^*, j^*)$ constitutes a pure-strategy noncooperative equilibrium (PSNE) if and only if

$$i^* = \underset{i \in \{1,2\}}{\arg\max} \{a_{i,j^*}\}$$

and

$$j^* = \underset{j \in \{1,2\}}{\arg\max} \{b_{i^*,j}\}\,.$$

A simple illustration of PSNE is given by the ordinal Prisoner's Dilemma of Table 3. If both prisoners remain silent, then each will be given only a minor penalty (of 3); however, if one confesses and the other does not, then the former receives a very light penalty (of 4), whereas the latter receives a more severe penalty (of 1) than the penalty if both had confessed (of 2). For this game, it is easy to confirm that the strategy pair (Down, Right) forms a PSNE with payoffs $(2, 2)$.

[4] In a matrix or normal-form game, any move is equivalent to a strategy because the players face no contingencies.

Table 3: Prisoner's Dilemma

	Left ("Remain Silent")	Right ("Confess")
Up ("Remain Silent")	3,3	1,4
Down ("Confess")	4,1	2,2

One of the most important contributions of Nash (1952) was the extension of noncooperative equilibrium from games with only pure strategies to games allowing each player to select a probability distribution over his or her possible choices. Thus, instead of just the pure-strategy pairs, $(s_{\mathrm{R}}, s_{\mathrm{C}}) = (i, j)$, we can consider random-strategy pairs, $(s_{\mathrm{R}}, s_{\mathrm{C}}) = (x, y)$, in which $x \in (0, 1)$ denotes the row player's (non-trivial) probability of selecting Up ($i = 1$), and $y \in (0, 1)$ denotes the column player's (likewise non-trivial) probability of selecting Left ($j = 1$). A noncooperative equilibrium that involves random strategies is referred to as a mixed-strategy noncooperative equilibrium (MSNE). One can solve for a game's MSNEs from the two conditions:

$$x^* = \underset{x \in (0,1)}{\arg\max} \left\{ a_{1,1} x y^* + a_{1,2} x \left(1 - y^*\right) + a_{2,1} \left(1 - x\right) y^* + a_{2,2} \left(1 - x\right) \left(1 - y^*\right) \right\} \tag{4.1}$$

$$y^* = \underset{y \in (0,1)}{\arg\max} \left\{ b_{1,1} x^* y + b_{1,2} x^* \left(1 - y\right) + b_{2,1} \left(1 - x^*\right) y + b_{2,2} \left(1 - x^*\right) \left(1 - y\right) \right\}. \tag{4.2}$$

The simple game of Matching Pennies, shown in Table 4, provides an intuitively reasonable use of mixed strategies. For the mixed-strategy pair $(x, y) = (1/2, 1/2)$, each player receives an expected payoff of 0. However, if either player selected a pure strategy, then the other player's best response would cause the first player to lose 1 unit with certainty.[5] Although coin tosses are commonly used by individuals in certain decision-making settings, the use of more complicated mixed strategies appears to depend heavily on player sophistication and problem context.

Table 4: Matching Pennies

	Heads	Tails
Heads	1,-1	-1,1
Tails	-1,1	1,-1

[5] Alternatively, one could consider an ordinal version of this game in which the payoffs -1 and 1 are replaced by the payoff ranks 1 and 2, respectively. In the ordinal game, using the mixed strategy $(x, y) = (1/2, 1/2)$ gives each player the same opportunity to receive 1 or 2.

3.1 Population of PSNEs

In the study of matrix games, it is useful to separate PSNEs and MSNEs because the former depend only on payoff ordinalities, whereas the latter are sensitive to cardinal differences. For the set of 144 strictly ordinal 2×2 games, the distribution of the number of PSNEs is given in Table 5.

Table 5: Distribution of PSNEs

Value	# of Games
0	18
1	108
2	18
Total	144

3.2 Population of MSNEs

From the list in Appendix 2, one can see that the set of 144 canonical games contains 36 games with exactly one MSNE, and no games with more than one MSNE. These 36 games match exactly with the set of games having either zero or two PSNEs, from which it follows that all 144 ordinal games contain at least one noncooperative equilibrium.

3.3 Some Easy Calculations

Most features and solutions of 2×2 matrix games – whether strictly ordinal or cardinal – are easy to calculate. In developing procedures for such calculations, dominant rows and/or columns play an important role.

A row [column] in a matrix game is said to dominate (strictly) another row [column] if and only if each payoff in the first row [column] is greater than the corresponding payoff in the second row [column]. In a 2×2 game, it is well known that:

- If a game possesses exactly one dominant row or column, then it must possess exactly one PSNE.
- If a game possesses zero dominant rows or columns, then it must possess zero PSNEs (and therefore exactly one MSNE).

- If a game possesses one dominant row and one dominating column, then it may possess either (a) exactly one PSNE or (b) two PSNEs and exactly one MSNE.

The above facts enable us to construct the following algorithm for computing all PSNEs and MSNEs:

1. Check each row and column for the dominance property, and let $d \in \{0, 1, 2\}$ denote the total number of dominant rows/columns.

2. If $d = 1$, then solve for the unique PSNE by identifying the cell in the dominant row [column] that has the greater payoff for the column [row] player.

3. If $d = 0$, then solve for the MSNE explicitly as

$$(x^*, y^*) = \left(\frac{a_{2,2} - a_{1,2}}{a_{1,1} - a_{1,2} - a_{2,1} + a_{2,2}}, \frac{b_{2,2} - b_{1,2}}{b_{1,1} - b_{1,2} - b_{2,1} + b_{2,2}} \right).$$

4. If $d = 2$, then solve for each of the two PSNEs using the method described in step (1) immediately above, and solve for the MSNE as in step (2) immediately above. (The two PSNEs also can be found as corner solutions of the system of equations in (4.1) and (4.2).)

4 Populations of Three Special Games

In Appendix 3, we provide the C++ program used to generate cardinal (and strictly ordinal) games as described in steps (1) through (5) of Subsection 1.1. Appendix 4 contains various sample means and variances associated with 100,000 cardinal games generated with a grid size of $\Delta = 1/256$ (i.e., for $k = 9$).

In the present section, we consider the populations of three special games discussed widely in the behavioral science literature:

- Prisoner's Dilemma (game 1 of Appendix 2);

- Stag Hunt (game 41 of Appendix 2); and

- Battle of the Sexes (game 10 of Appendix 2).

The common names of these games attach considerable context to the abstract payoff structure, which may or may not be justified. In particular, there is little indication that anyone who does not know these common names would associate them with the specific games involved (see, e.g., I. Powers and Shubik, 1991). Nevertheless, the structural features of these three settings allow us to illustrate several important aspects of 2×2 games.

4.1 Prisoner's Dilemma

Possibly the most studied of all games is the Prisoner's Dilemma, whose popularity arises at least in part because its ordinal form is the only game within the canonical 144 for which: (a) there is a unique PSNE, (Down, Right), that is strictly dominated by another feasible outcome, (Up, Left); and (b) all other outcomes are Pareto optimal. Figure 22 of Appendix 1 presents the payoff set for this well-known game.

In exploring the population of cardinal Prisoner's Dilemma games for a given choice of grid size, $\Delta = 1/2^{k-1}$, it is helpful to think of the entire domain of possible payoffs, from the game with the smallest payoff values, in Table 6, to that with the largest payoff values, in Table 7. Naturally, an ordinal treatment of preferences recognizes no difference between these two games.

Table 6: Prisoner's Dilemma, Smallest Payoffs

	Left	Right
Up	$\frac{3}{2^{k-1}}, \frac{3}{2^{k-1}}$	$\frac{1}{2^{k-1}}, \frac{4}{2^{k-1}}$
Down	$\frac{4}{2^{k-1}}, \frac{1}{2^{k-1}}$	$\frac{2}{2^{k-1}}, \frac{2}{2^{k-1}}$

Table 7: Prisoner's Dilemma, Largest Payoffs

	Left	Right
Up	$4 - \frac{1}{2^{k-1}}, 4 - \frac{1}{2^{k-1}}$	$4 - \frac{3}{2^{k-1}}, 4$
Down	$4, 4 - \frac{3}{2^{k-1}}$	$4 - \frac{2}{2^{k-1}}, 4 - \frac{2}{2^{k-1}}$

As k increases and the grid becomes finer, the population of cardinal Prisoner's Dilemma games covers more and more of the entire square interval $(0, 4]^2$. In all cases, the game has only the single

PSNE (Down, Right), whose payoff pair depends on the particular values of $a_{i,j}$ and $b_{i,j}$ generated in step (1) of Subsection 1.1.

As a baseline, we note that for $k = 1$, the PSNE payoff pair is always $(2, 2)$, whereas entropy players do better on average, obtaining an expected payoff of $(2.5, 2.5)$. Figures 2 and 3 provide scatter diagrams of payoff pairs from 100,000 games for $k = 9$ and $\Delta = 1/256$. Figure 2 shows the PSNEs, and Figure 3 shows the corresponding entropic outcomes. In each case, the sample-mean payoff pair is indicated by a red diamond near the center of the point cluster.

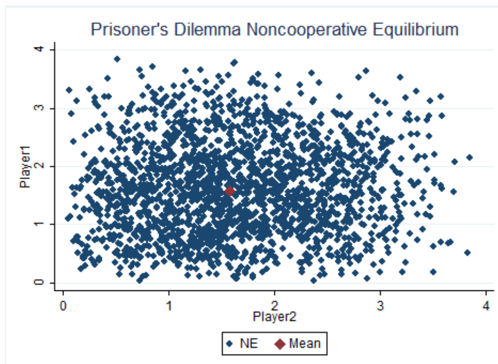


Figure 2: Prisoner's Dilemma, Noncooperative Equilibrium

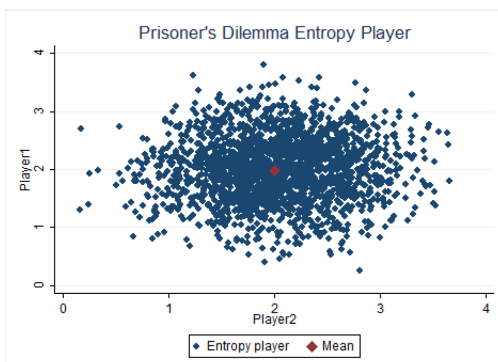


Figure 3: Prisoner's Dilemma, Entropy Players

4.2 Stag Hunt

The Stag Hunt (Table 8) possesses two PSNEs, one of which, (Up, Left), strictly dominates the other, (Down, Right). However, depending on underlying assumptions, it can be argued that the smaller payoff pair sometimes will be chosen by rational players. In particular, if the row [column] player believes that the column [row] player will choose Right [Down] with probability greater than 1/2, then he or she will be motivated to choose Down [Right].

Table 8: Stag Hunt

	Left	Right
Up	4,4	1,3
Down	3,1	2,2

For $k = 1$, the two PSNEs have payoff pairs $(4, 4)$ and $(2, 2)$, respectively, and the MSNE, $(x^*, y^*) = (1/2, 1/2)$, yields expected payoffs of $(2.5, 2.5)$. Given the value of (x^*, y^*), one can see that the MSNE yields identical strategies and payoffs as the game with entropy players.

Figures 4 and 5 provide scatter diagrams of payoff pairs from 100,000 games for $k = 9$ and $\Delta = 1/256$. Figure 4 includes the PSNEs and MSNE, each with equal frequency, and Figure 5 shows the corresponding entropic outcomes. As before, the sample-mean payoff pairs are indicated by red diamonds near the centers of the point clusters.

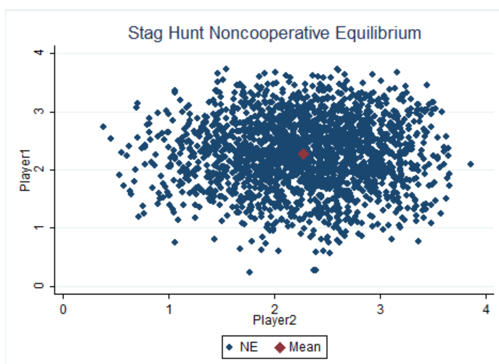


Figure 4: Stag Hunt, Noncooperative Equilibrium

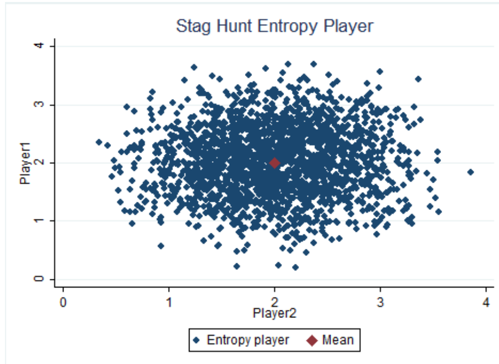


Figure 5: Stag Hunt, Entropy Players

4.3 Battle of the Sexes

Unlike the Prisoner's Dilemma and Stag Hunt, the Battle of the Sexes (Table 9) is not a symmetric game (i.e., $a_{i,j} \neq b_{j,i}$ for some (i,j)). Like the Stag Hunt, however, it possesses two PSNEs, (Up, Left) and $(\text{Down}, \text{Right})$, and 1 MSNE, $(x^*, y^*) = (1/2, 1/2)$.

Table 9: Battle of the Sexes

	Left	Right
Up	4, 3	2, 2
Down	1, 1	3, 4

For $k = 1$, the two PSNEs have payoff pairs $(4,3)$ and $(3,4)$, respectively, and the MSNE gives expected payoffs of $(2.5, 2.5)$. Thus, as in the case of the Stag Hunt, the game with entropy players yields identical strategies and payoffs as noncooperative players using the MSNE.

Figures 5 and 6 provide scatter diagrams of payoff pairs from 100,000 games for $k = 9$ and $\Delta = 1/256$. Figure 5 includes the PSNEs and MSNE, each with equal frequency, and Figure 6 shows the corresponding entropic outcomes. Once again, the sample-mean payoff pairs are indicated by red diamonds.

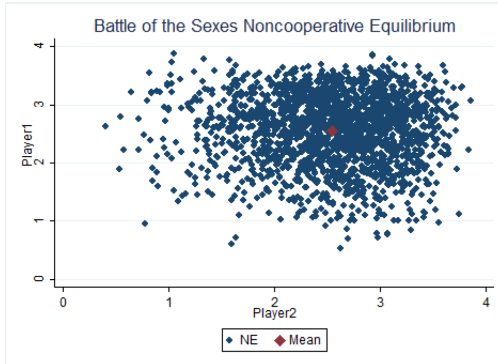


Figure 6: Battle of the Sexes, Noncooperative Equilibrium

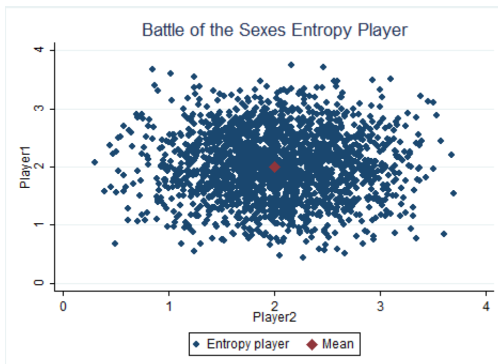


Figure 7: Battle of the Sexes, Entropy Players

4.4 Correlated Strategies and Efficiency

In examining the results of the Stag Hunt and Battle of the Sexes, we observe that the expected payoffs of the MSNE are strictly lower than at least one pair of PSNE payoffs in both games. This suggests a potential problem of ineffective coordination. In other words, the players may arrive at a noncooperative equilibrium for which the individual and/or joint payoffs (i.e., a_{i^*,j^*}, b_{i^*,j^*}, and/or $a_{i^*,j^*} + b_{i^*,j^*}$) are less than optimal. Therefore, the value of being able to coordinate strategies may be substantial.

Table 10 offers some quantification of the deficiencies attributable to ineffective coordination for the three games discussed above. For each game, the second column presents the average of the joint payoffs over all noncooperative equilibria, giving equal weight to each PSNE and MSNE. For the Prisoner's Dilemma, this is $2 + 2 = 4$; for the Stag Hunt, it is $[(4 + 4) + (2 + 2) + (2.5 + 2.5)]/3 \approx$

5.6667; and for the Battle of the Sexes, it is $[(4+3)+(3+4)+(2.5+2.5)]/3 \approx 6.3333$. In the third column, we construct a simple efficiency measure by dividing the average joint payoff by the maximum possible joint payoff that can be achieved by either a pure- or mixed-strategy pair imposed by an exogenous agency (custom, law, private intermediation, etc.). Since the agent is exogenous, it can expand the domain of mixed strategies to correlated strategies, for which the players' random selections of Up and Down are statistically dependent. For the Prisoner's Dilemma, this yields $3+3=6$; for the Stag Hunt, it yields $4+4=8$; and for the Battle of the Sexes, it yields either $4+3=3+4=7$ or $[(4+3)\,p+(3+4)\,(1-p)]=7$, where the latter value comes from any correlated mixed strategy that chooses (Up, Left) and $(\text{Down}, \text{Right})$ with probabilities p and $1-p$, respectively.[6] The fourth and fifth columns present corresponding calculations for games with entropy players.

Table 10: Joint Payoffs for $k = 1$

Game	Avg. of NE Payoffs	NE Efficiency	Entropy Payoffs	Entropy Efficiency
Prisoner's Dilemma	4.0000	0.6667	5.0000	0.8333
Stag Hunt	5.6667	0.7083	5.0000	0.6250
Battle of the Sexes	6.3333	0.9048	5.0000	0.7143

5 Discussion

5.1 Efficiency Analysis of All 2×2 Games

Table 11 provides efficiency measures – as defined in the previous subsection – for the entire population of cardinal games generated in steps (1) through (5) of Subsection 1.1 for $k = 1$ and $k = 9$. This table addresses the nature of the optimality of individual behavior within all possible 2×2-game structures, subdivided by the values of JM_1 in the associated canonical ordinal game. Thus, for clarity, we would note that: (a) Prisoner's Dilemma games are included in the category of $JM_1 = 6$; (b) Stag Hunt games are included in $JM_1 = 7$; and (c) Battle of the Sexes games are included in $JM_1 = 8$.

[6] Within the conventional Battle of the Sexes storyline, the man and woman who are trying to decide which movie to see could choose between "his" movie and "her" movie by tossing a coin (for which $p = 1/2$).

Table 11: Efficiencies of All Games

	$k=1$ NE	$k=1$ Entropy	$k=9$ NE	$k=9$ Entropy
$JM_1=8$	0.9410	0.6250	0.7532	0.5000
$JM_1=7$	0.9127	0.7143	0.7305	0.5716
$JM_1=6$	0.9352	0.8333	0.7491	0.6669
$JM_1=5$	NA	NA	NA	NA
Average	0.9300	0.7386	0.7445	0.5910

As is often true in game theory, behavioral paradoxes abound. We purposely indicate that efficiency measures are "not applicable" for the case of $JM_1=5$, because these games of opposition are qualitatively different from all others. Formally, the efficiency measures could be defined as 1.0, but such calculations would be misleading because these games do not reflect the characteristics of a society. Specifically, there is no room for cooperation, coordination, or any form of discourse, and whatever one individual gains the other loses. As noted before, such structural dystopias become increasingly rare in the "Flatland" (see Abbott, 1952 [1884]) of matrix games that arises for large values of k.

By definition, the category of $JM_1=8$ comprises games of coordination. These games always include an outcome with payoff pair $(4,4)$, and such outcomes must be PSNEs. In this case, the reason why efficiency is not exactly 1.0 is that some games have two PSNEs and 1 MSNE, and these lower the average.

Somewhat surprisingly, the fall-off in the efficiency of noncooperative equilibria as k increases from 1 to 9 is rather large for all game categories. For $k=1$, the efficiency loss in games of coordination ($JM_1=8$) and mixed-motive games ($JM_1=7,6$) is between 6 and 9 percent. For $k=9$, this grows to between 25 and 27 percent, and results for $k=10$ indicate that $k=9$ is very close to the limit, with differences in efficiency of less than 0.01 percent. (See Table 12.) Part of the decrease in efficiency is attributable to the fact that $E\left[JM_k\right]\to E\left[JM_\infty\right]\approx 5.6968$ for large k , which is substantially less than $E\left[JM_1\right]=6.875$.

Table 12: Efficiencies for $k = 9$ and $k = 10$

	$k = 9$ NE	$k = 9$ Entropy	$k = 10$ NE	$k = 10$ Entropy
$JM_1 = 8$	0.7532	0.5000	0.7528	0.5004
$JM_1 = 7$	0.7305	0.5716	0.7310	0.5717
$JM_1 = 6$	0.7491	0.6669	0.7489	0.6672
$JM_1 = 5$	NA	NA	NA	NA
Average	0.7445	0.5910	0.7447	0.5914

We also consider the completely different behavior of entropy players, who may be viewed as "know-nothing" or "zero-intelligence" decision makers. Although it is easy to construct games (e.g., the Prisoner's Dilemma) in which the entropy players' expected payoffs of $(2.5, 2.5)$ are greater than those of noncooperative players, Table 11 shows that for both $k = 1$ and $k = 9$, entropy players perform worse on average than noncooperative players. This is because they are unable to take advantage of certain game structures, such as row and/or column dominance, that assist coordination. Even in this highly constrained environment, differences in the cardinal measures of payoffs yield far greater variability and inequality when k is large than when $k = 1$. The meaning of this change is that as the variety of outcomes grows, the worth of coordination or collaboration grows as well.

5.2 Why the 2×2 Case Is So Important

There are many reasons why 2×2 games are crucial both to the study of game theory specifically, and behavioral science more generally. These include:

1. They offer a highly useful starting point for illustrating and contrasting many problems and paradoxes in strategic analysis.

2. They are widely used by introductory instructors of game theory in the behavioral sciences. (Is this pedagogical use justified? We would argue that it is, with appropriate qualifications.)

3. They greatly facilitate analogy generation and storytelling in connecting specific real-world problems to abstract models. Hence, they provide valuable exercises in tying the physical world to mathematics.

4. They offer minimal repeated-game models for the dynamics of learning, signaling, and other complex human behaviors.

5. Through a handful of special cases (e.g., the Prisoner's Dilemma, Stag Hunt, and Battle of the Sexes), they successfully illustrate fundamental problems in strategy and society.

Items (1), (2), and (4) are highly related for both teaching and research, especially if we believe that dyadic relations are of considerable importance in describing human and other animal behavior. It is thus for good reason that elementary textbooks in the behavioral sciences abound with 2×2-game examples.

6 For $n > 3$, a Basic Change in the Paradigm

On the whole, we would argue that the study of 2×2 games is important because so much of human activity is well modeled by interactions between two individuals or between one individual and an institution, with relatively few choices for each party in the short term. This perspective, however, can blind us against the enormous complexities that arise when increasing matrix size. For example, in the case of a 3×3 matrix, the number of distinct strategic cases rises to $(9!)^2 \,/\, (3!)^2 \approx 3.6578 \times 10^9$; and in the case of 3 players with 2 strategies each, we have a $2 \times 2 \times 2$ matrix with $(8!)^3 \,/\, (2!)^3 \approx 8.1935 \times 10^{12}$ strategically different possibilities.

In the case of 3×3 games, the simple example of Table 13 is sufficient to destroy the hopes of those interested in developing plausible dynamic strategic solutions. One natural candidate for simple dynamics is optimal response; that is, players consider where they have been in a previous play of the game, and use that as the basis for their current optimization. However, a brief glance at Table 13 shows that if the row and column players begin with strategies $(s_{\mathrm{C}}, s_{\mathrm{R}}) = (1, 1)$, then the column player will move to $s_{\mathrm{C}} = 3$, and a 4-cycle will emerge that never converges to the joint maximum PSNE at $(s_{\mathrm{C}}, s_{\mathrm{R}}) = (2, 2)$. (Note that if the payoff at $(s_{\mathrm{C}}, s_{\mathrm{R}}) = (2, 2)$ were $(1, 1)$ instead of $(9, 9)$, that particular outcome would still be a PSNE.) Quint, Shubik, and Yan (1995) demonstrated the extensive potential for cycling in a large class of $n \times n$ games involving both sequential and simultaneous moves.

Table 13: A Simple 3×3 Matrix Game

	$s_C = 1$	$s_C = 2$	$s_C = 3$
$s_R = 1$	4, 1	0, 0	1, 4
$s_R = 2$	0, 0	9, 9	0, 0
$s_R = 3$	1, 4	0, 0	4, 1

6.1 Smooth and Rough Games

In any 2×2 game, it is possible to compute 4 first differences between cell payoffs. In the 3×3 case, one can compute 12 first differences and 4 second differences, and the payoff sets still have only few hills and valleys. However, for matrices of 4×4 and above, the potential roughness of the payoff structure increases rapidly. Although the mathematical structure is clearly defined, the analysis of applied decision problems becomes extremely difficult unless some appropriate smoothing mechanism is imposed on the payoff surfaces. In corporate, military, and political planning, the adjustment, evaluation, refinement, and discarding of many features of a strategic process tend to narrow the final choices to a set that includes certainly less than ten, and often not more than two or three, alternatives.

These somewhat terse remarks will be enlarged in further work.

6.2 Coordination, Opposition, and Noncooperative Equilibria

As we increase the number of strategies, or players, or both, the relative numbers of games of coordination and opposition become vanishingly small.

If we consider $m \times n$ matrix games with $m, n \geq 3$, the relative number of PSNEs drops, and MSNEs proliferate. There is a literature illustrating this, which includes limiting formulas for the probability of encountering a PSNE in a large, random matrix game. (See Goldberg, Goldman, and Newman, 1968, Dresher, 1970, and I. Powers, 1990.)

6.3 Life Is a Set of Measure Zero

A guiding principle for exploring the enormous universe of matrix games is to select appropriate limiting processes to obtain robust sets of games addressing important questions of interest. We do that here in taking the limit of all cardinal 2×2 games with a minimal grid size.

In general, many problems of interest require such specification that they can be regarded as sets of measure zero with respect to far larger abstract categories to which they belong. An important example (to which we will return in a subsequent paper) is the presence of ties. In many contexts of human activity, ties in perceived valuation are present. Furthermore, when one individual has finer perceptions than another, the increased precision almost always works to his or her advantage. In the present essay, we have ruled out ties for simplicity and manageability. If we had permitted them, the number of canonical ordinal games would have increased from 144 to 726 (see Kilgour and Frazer, 1988).

7 Concluding Remarks

The principal purpose of this paper was to find a reasonably natural way to consider all cardinal 2×2 games within a given finite grid. The essentially combinatoric aspects of the investigation called for simulation to evaluate structural aspects that are difficult to see using only analytic methods. Our study of this structure provided a sufficiently rich background for examining in detail both noncooperative equilibria and entropy-player solutions. In a subsequent paper, we will consider other solution concepts that enable one to investigate the influence of structure on behavior with various intents.

The amount of "fat left in the system" depends on the solution used. In many cases, a referee, government, or other outside agency could be used to guide the system to a superior outcome, while consuming fewer resources than it adds. The gap between the current solution and the joint maximum is the maximum *worth* of the coordinator. Given the basic analysis of the present paper, we now are in a position to consider developing improved measures of efficiency and symmetry for any outcome in a matrix game. We plan to discuss this topic as well in a future paper.

In short, this first essay was aimed at providing a simple idea of the worth of government, with a quick and crude estimate. A second essay will be devoted to the many problems of structure and behavior arising from noncooperative equilibria in 2×2 games. Finally, a third essay will address the development of more sophisticated efficiency measures based upon the analyses of the prior work.

References

[1] Abbott, E. A., 1952 [1884], *Flatland: A Romance of Many Dimensions* (6th ed.), New York: Dover.

[2] Dresher, M., 1970, "Probability of a Pure Equilibrium Point in n-Person Games", *Journal of Combinatorial Theory*, 8, 134-145.

[3] Goldberg, K., Goldman, A. J., and Newman, M., 1968, "The Probability of an Equilibrium Point", *Journal of Research of National Bureau of Standards U.S.A.*, 72, B, 93-101.

[4] Kilgour, D. M. and Frazer, N. M., 1988, "A Taxonomy of All Ordinal 2×2 Games", *Theory and Decision*, 24, 99-117.

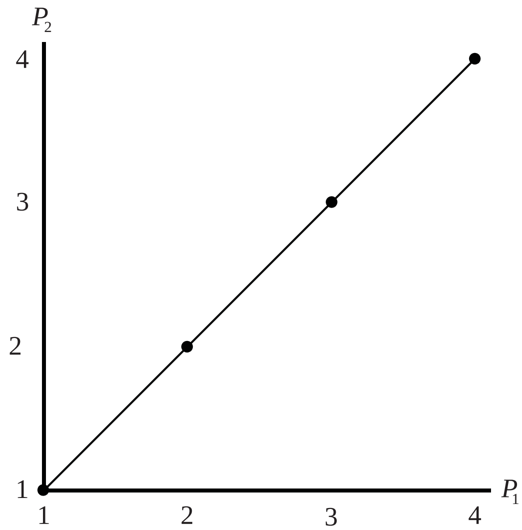
[5] Powers, I. Y., 1990, "Limiting Distributions of the Number of Pure Strategy Nash Equilibria in n-Person Games", *International Journal of Game Theory*, 19, 3, 277-286.

[6] Powers, I. Y. and Shubik, M., 1991, "The Names of the Games", *International Security and Arms Control Discussion Paper*, No. 7, Yale University.

[7] Quint, T., Shubik, M., and Yan, D., 1996, "Dumb Bugs and Bright Noncooperative Players: Games, Context, and Behavior", in *Understanding Strategic Interaction: Essays in Honor of Reinhard Selten*, (W. Albers, et al., eds.), Berlin: Springer-Verlag, 185-197.

Appendix 1

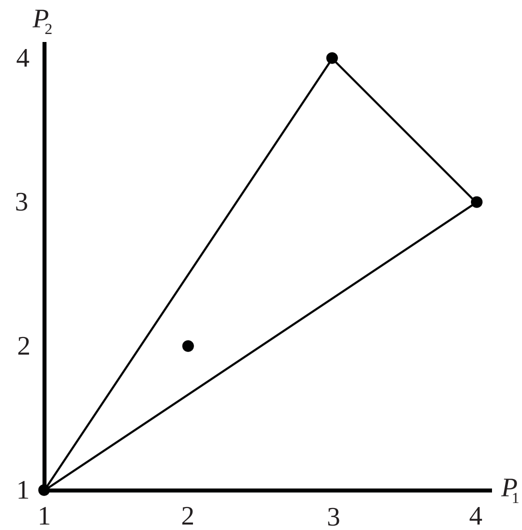
$$\begin{bmatrix} 4,4 & 3,3 \\ 2,2 & 1,1 \end{bmatrix}\begin{bmatrix} 4,4 & 3,3 \\ 1,1 & 2,2 \end{bmatrix}\begin{bmatrix} 4,4 & 2,2 \\ 1,1 & 3,3 \end{bmatrix}$$



20, 92, 124

FIGURE 1

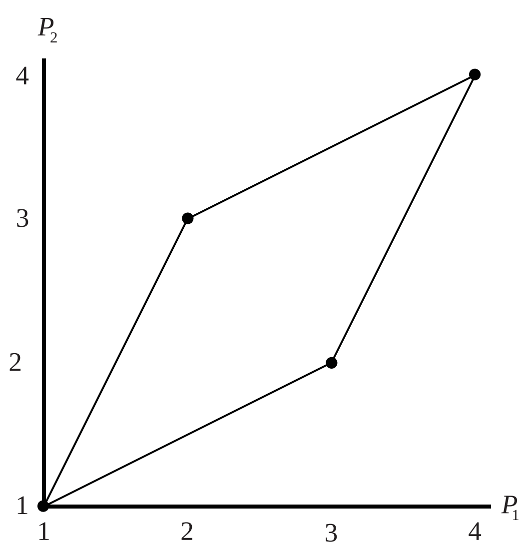
$$\begin{bmatrix} 3,4 & 4,3 \\ 2,2 & 1,1 \end{bmatrix}\begin{bmatrix} 3,4 & 4,3 \\ 1,1 & 2,2 \end{bmatrix}\begin{bmatrix} 2,2 & 3,4 \\ 4,3 & 1,1 \end{bmatrix}\begin{bmatrix} 2,2 & 4,3 \\ 3,4 & 1,1 \end{bmatrix}$$



34,79,74,10

FIGURE 2

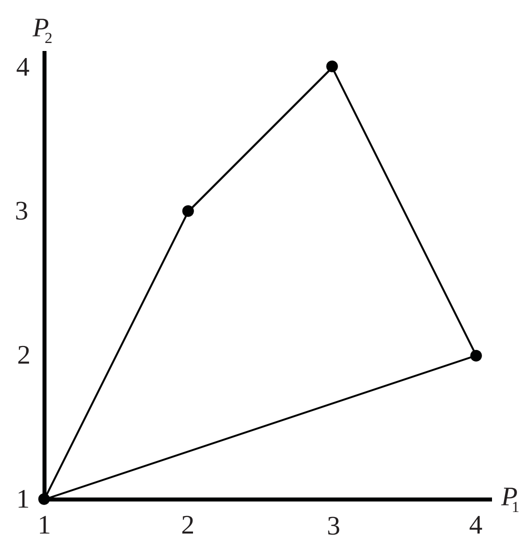
$$\begin{bmatrix} 4,4 & 3,2 \\ 2,3 & 1,1 \end{bmatrix}\begin{bmatrix} 4,4 & 2,3 \\ 3,2 & 1,1 \end{bmatrix}\begin{bmatrix} 4,4 & 3,2 \\ 1,1 & 2,3 \end{bmatrix}\begin{bmatrix} 4,4 & 2,3 \\ 1,1 & 3,2 \end{bmatrix}$$



19,3,93,122

FIGURE 3

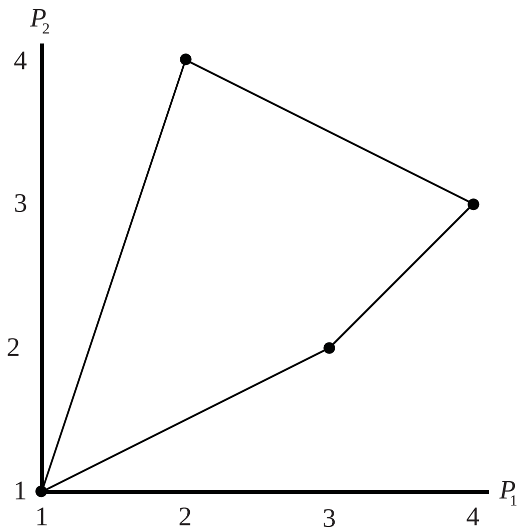
$$\begin{bmatrix} 3,4 & 4,2 \\ 2,3 & 1,1 \end{bmatrix}\begin{bmatrix} 3,4 & 4,2 \\ 1,1 & 2,3 \end{bmatrix}\begin{bmatrix} 2,3 & 4,2 \\ 1,1 & 3,4 \end{bmatrix}\begin{bmatrix} 2,3 & 3,4 \\ 4,2 & 1,1 \end{bmatrix}\begin{bmatrix} 3,4 & 1,1 \\ 2,3 & 4,2 \end{bmatrix}$$



33,80,105,76,125

FIGURE 4

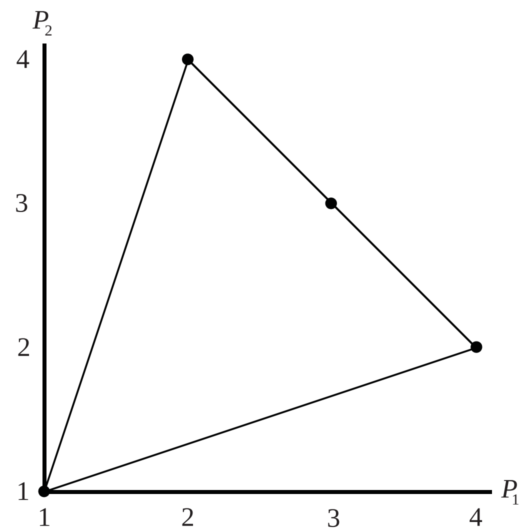
$$\begin{bmatrix} 2,4 & 4,3 \\ 1,1 & 3,2 \end{bmatrix}$$



101

FIGURE 5

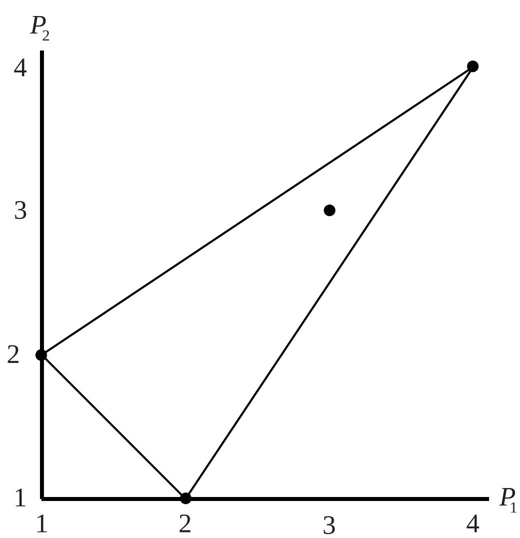
$$\begin{bmatrix} 3,3 & 4,2 \\ 2,4 & 1,1 \end{bmatrix}\begin{bmatrix} 3,3 & 2,4 \\ 4,2 & 1,1 \end{bmatrix}\begin{bmatrix} 3,3 & 4,2 \\ 1,1 & 2,4 \end{bmatrix}\begin{bmatrix} 2,4 & 4,2 \\ 1,1 & 3,3 \end{bmatrix}$$



35,53,21,103

FIGURE 6

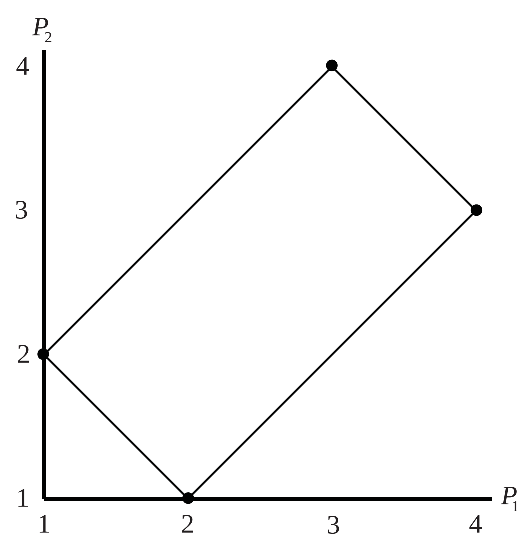
$$\begin{bmatrix} 4,4 & 3,3 \\ 1,2 & 2,1 \end{bmatrix}\begin{bmatrix} 4,4 & 3,3 \\ 2,1 & 1,2 \end{bmatrix}\begin{bmatrix} 4,4 & 1,2 \\ 2,1 & 3,3 \end{bmatrix}\begin{bmatrix} 4,4 & 1,2 \\ 3,3 & 2,1 \end{bmatrix}$$



32,82,69,48

FIGURE 7

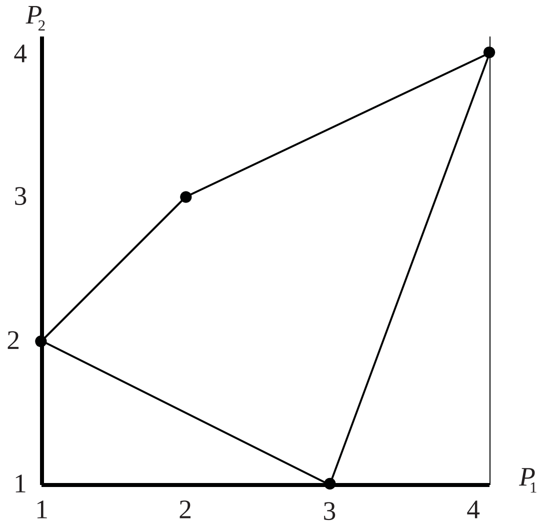
$$\begin{bmatrix} 3,4 & 4,3 \\ 1,2 & 2,1 \end{bmatrix}\begin{bmatrix} 3,4 & 4,3 \\ 2,1 & 1,2 \end{bmatrix}\begin{bmatrix} 3,4 & 2,1 \\ 1,2 & 4,3 \end{bmatrix}$$



18,94,67

FIGURE 8

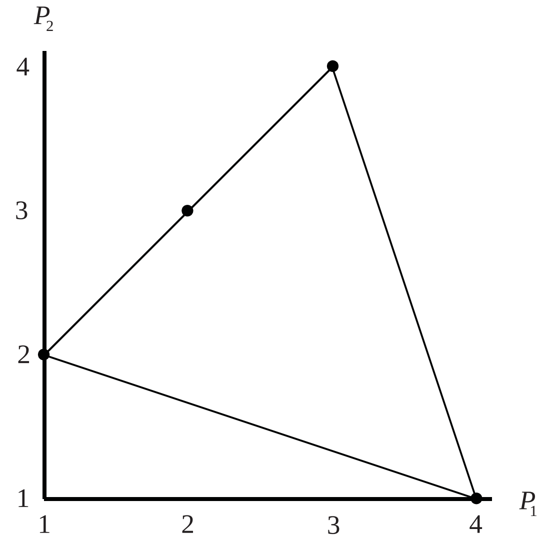
$$\begin{bmatrix} 4,4 & 1,2 \\ 3,1 & 2,3 \end{bmatrix}\begin{bmatrix} 4,4 & 3,1 \\ 1,2 & 2,3 \end{bmatrix}\begin{bmatrix} 4,4 & 2,3 \\ 3,1 & 1,2 \end{bmatrix}$$



140,29,106

FIGURE 9

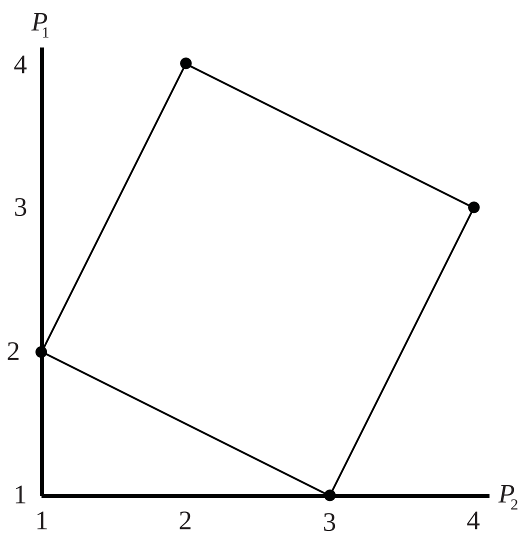
$$\begin{bmatrix} 3,4 & 4,1 \\ 2,3 & 1,2 \end{bmatrix}\begin{bmatrix} 3,4 & 1,2 \\ 2,3 & 4,1 \end{bmatrix}\begin{bmatrix} 3,4 & 4,1 \\ 1,2 & 2,3 \end{bmatrix}\begin{bmatrix} 3,4 & 2,3 \\ 1,2 & 4,1 \end{bmatrix}\begin{bmatrix} 2,3 & 4,1 \\ 1,2 & 3,4 \end{bmatrix}$$



30,128,16,71,110

FIGURE 10

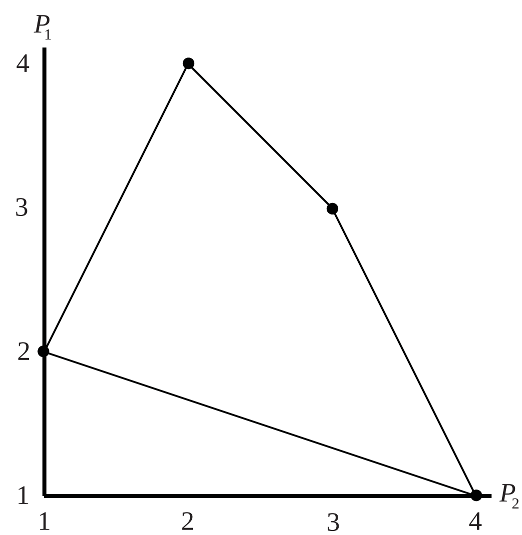
$$\begin{bmatrix} 2,4 & 4,3 \\ 1,2 & 3,1 \end{bmatrix}\begin{bmatrix} 2,4 & 3,1 \\ 1,2 & 4,3 \end{bmatrix}$$



2,46

FIGURE 11

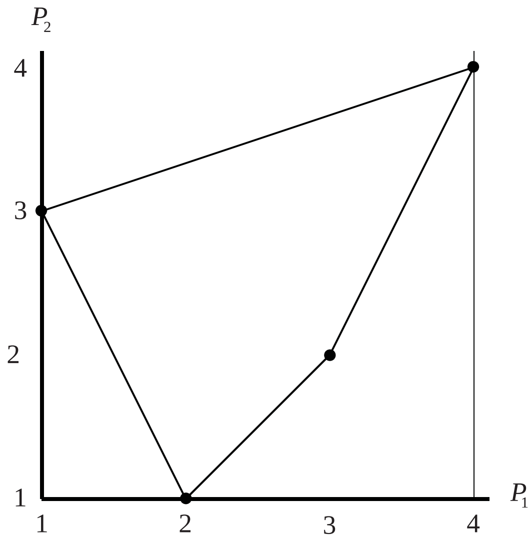
$$\begin{bmatrix} 3,3 & 4,1 \\ 1,2 & 2,4 \end{bmatrix}\begin{bmatrix} 2,4 & 4,1 \\ 1,2 & 3,3 \end{bmatrix}\begin{bmatrix} 2,4 & 3,3 \\ 1,2 & 4,1 \end{bmatrix}$$



84,108,41

FIGURE 12

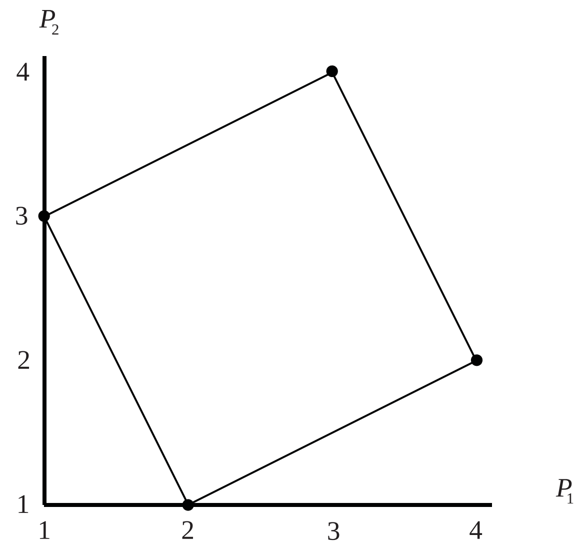
$$\begin{bmatrix} 4,4 & 3,2 \\ 1,3 & 2,1 \end{bmatrix}\begin{bmatrix} 4,4 & 3,2 \\ 2,1 & 1,3 \end{bmatrix}\begin{bmatrix} 4,4 & 2,1 \\ 3,2 & 1,3 \end{bmatrix}\begin{bmatrix} 4,4 & 2,1 \\ 1,3 & 3,2 \end{bmatrix}$$



28,85,113,132

FIGURE 13

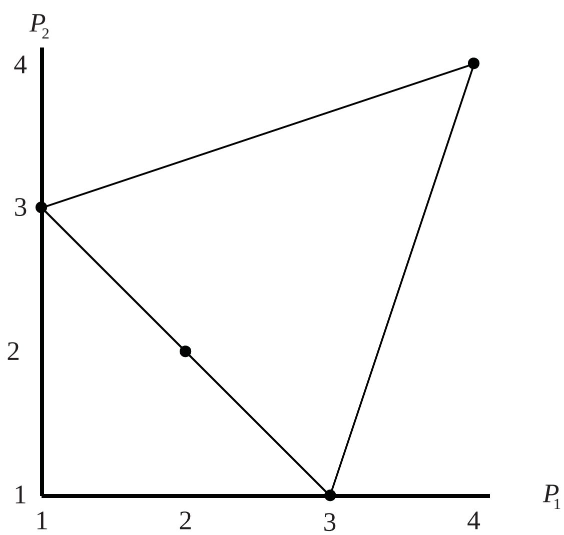
$$\begin{bmatrix} 3,4 & 4,2 \\ 1,3 & 2,1 \end{bmatrix}\begin{bmatrix} 3,4 & 2,1 \\ 1,3 & 4,2 \end{bmatrix}\begin{bmatrix} 3,4 & 4,2 \\ 2,1 & 1,3 \end{bmatrix}\begin{bmatrix} 3,4 & 2,1 \\ 4,2 & 1,3 \end{bmatrix}$$



86,63,96,44

FIGURE 14

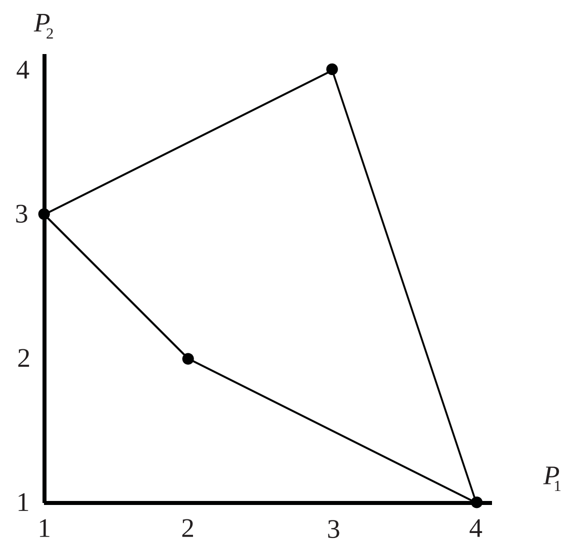
$$\begin{bmatrix} 4,4 & 3,1 \\ 1,3 & 2,2 \end{bmatrix}\begin{bmatrix} 4,4 & 3,1 \\ 2,2 & 1,3 \end{bmatrix}\begin{bmatrix} 4,4 & 1,3 \\ 3,1 & 2,2 \end{bmatrix}$$



26,15,41

FIGURE 15

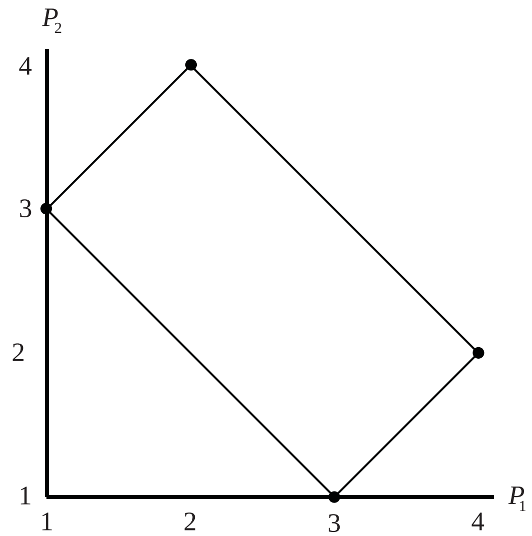
$$\begin{bmatrix} 3,4 & 4,1 \\ 1,3 & 2,2 \end{bmatrix}\begin{bmatrix} 3,4 & 2,2 \\ 1,3 & 4,1 \end{bmatrix}\begin{bmatrix} 3,4 & 1,3 \\ 2,2 & 4,1 \end{bmatrix}\begin{bmatrix} 3,4 & 4,1 \\ 2,2 & 1,3 \end{bmatrix}\begin{bmatrix} 2,2 & 4,1 \\ 1,3 & 3,4 \end{bmatrix}$$



87,66,5,27,116

FIGURE 16

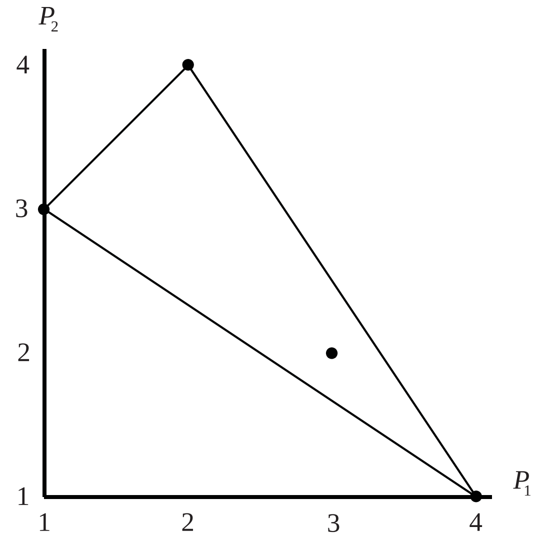
$$\begin{array}{c} M \\ \begin{bmatrix} 2,4 & 3,1 \\ 1,3 & 4,2 \end{bmatrix} \begin{bmatrix} 2,4 & 3,1 \\ 4,2 & 1,3 \end{bmatrix} \begin{bmatrix} 2,4 & 4,2 \\ 1,3 & 3,1 \end{bmatrix} \end{array}$$



43,64,112

FIGURE 17

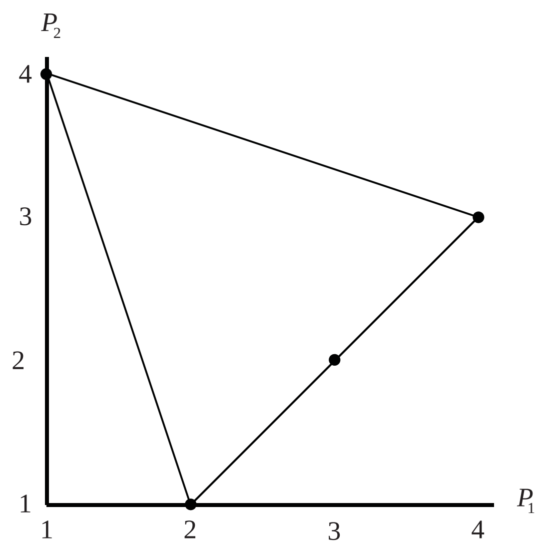
$$\begin{array}{c} M \\ \begin{bmatrix} 2,4 & 4,1 \\ 1,3 & 3,2 \end{bmatrix} \begin{bmatrix} 2,4 & 3,2 \\ 1,3 & 4,1 \end{bmatrix} \begin{bmatrix} 2,4 & 4,1 \\ 3,2 & 1,3 \end{bmatrix} \begin{bmatrix} 3,2 & 4,1 \\ 1,3 & 2,4 \end{bmatrix} \end{array}$$



114,45,6,88

FIGURE 18

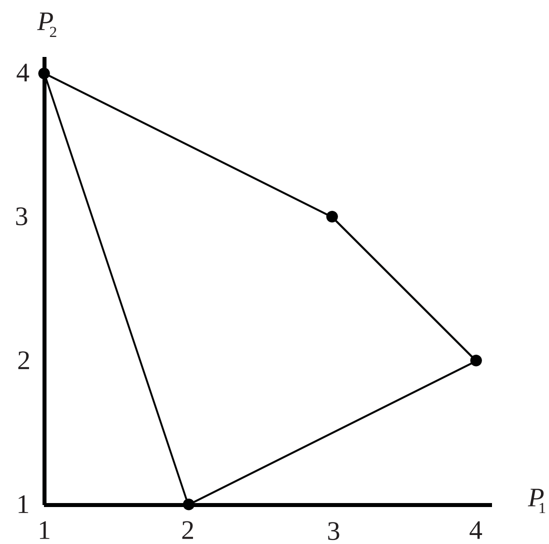
$$\begin{array}{c} M \\ \begin{bmatrix} 3,2 & 2,1 \\ 4,3 & 1,4 \end{bmatrix} \end{array}$$



143

FIGURE 19

$$\begin{array}{c} M \\ \begin{bmatrix} 3,3 & 4,2 \\ 1,4 & 2,1 \end{bmatrix} \begin{bmatrix} 3,3 & 4,2 \\ 2,1 & 1,4 \end{bmatrix} \begin{bmatrix} 3,3 & 2,1 \\ 4,2 & 1,4 \end{bmatrix} \end{array}$$

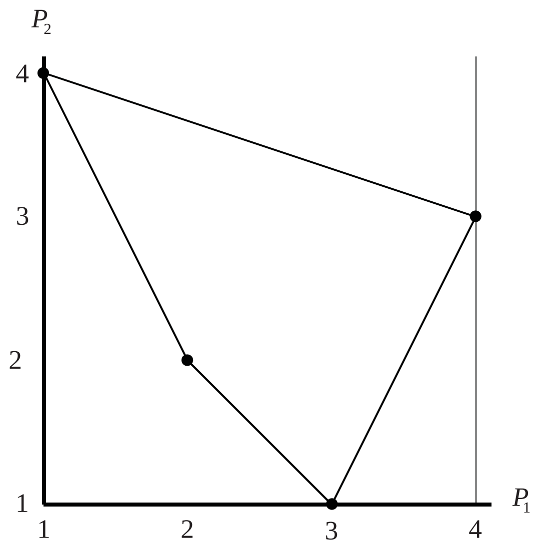


89,24,142

FIGURE 20

M

$$\begin{bmatrix} 2,2 & 3,1 \\ 4,3 & 1,4 \end{bmatrix}$$

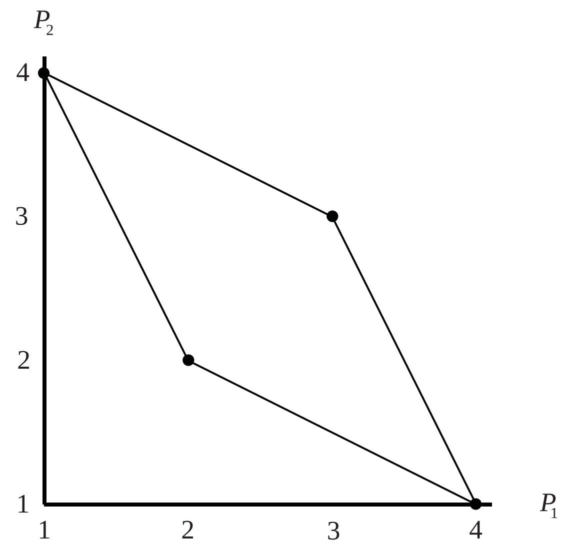


57

FIGURE 21

M

$$\begin{bmatrix} 3,3 & 4,1 \\ 1,4 & 2,2 \end{bmatrix}\begin{bmatrix} 3,3 & 4,1 \\ 2,2 & 1,4 \end{bmatrix}\begin{bmatrix} 2,2 & 4,1 \\ 3,3 & 1,4 \end{bmatrix}\begin{bmatrix} 2,2 & 4,1 \\ 1,4 & 3,3 \end{bmatrix}$$

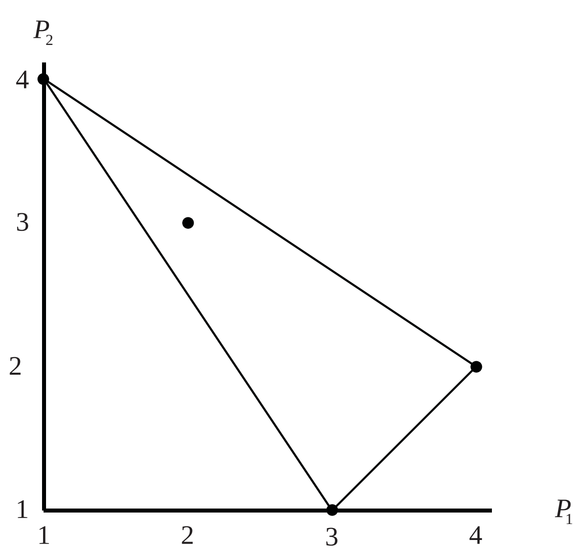


12,99,138,1

FIGURE 22

M

$$\begin{bmatrix} 2,3 & 4,2 \\ 1,4 & 3,1 \end{bmatrix}\begin{bmatrix} 2,3 & 3,1 \\ 4,2 & 1,4 \end{bmatrix}$$

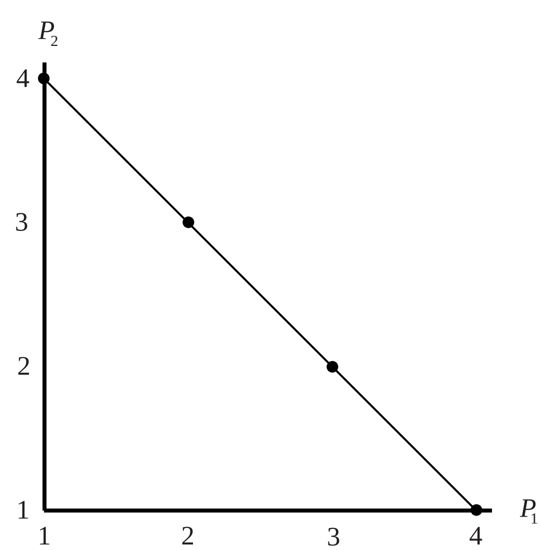


118,144

FIGURE 23

M

$$\begin{bmatrix} 2,3 & 4,1 \\ 1,4 & 3,2 \end{bmatrix}\begin{bmatrix} 3,2 & 4,1 \\ 2,3 & 1,4 \end{bmatrix}\begin{bmatrix} 2,3 & 4,1 \\ 3,2 & 1,4 \end{bmatrix}$$



120,100,136

FIGURE 24

Appendix 2

Game #	Payoff Matrix	Shape	Joint Max	PSNEs	Symmetric	Nash Payoff Row	Nash Payoff Col.	Dom.	Pareto Optima	Transpose
1	(1,4) (3,3) (2,2) (4,1)	22	6	1	Sym	2	2	2	3	NA
2	(1,2) (3,1) (2,4) (4,3)	11	7	1		2	4	2	2	115
3	(1,1) (3,2) (2,3) (4,4)	3	8	1	Sym	4	4	2	1	NA
4	(1,4) (3,3) (4,2) (2,1)	20	6	1		4	2	1	3	108
5	(1,3) (3,4) (4,1) (2,2)	16	7	1		3	4	1	2	117
6	(1,3) (3,2) (4,1) (2,4)	18	6	0		2.5	2.5	0	3	134
7	(1,2) (3,3) (4,4) (2,1)	7	8	2	Sym	4 3	4 3	0	1	NA
8	(1,2) (3,1) (4,4) (2,3)	9	8	1		4	4	1	1	113
9	(1,1) (3,2) (4,3) (2,4)	5	7	1		3	2	1	2	105
10	(1,1) (3,4) (4,3) (2,2)	2	7	2	Sym	3 4	4 3	0	2	NA
11	(1,4) (2,3) (3,2) (4,1)	24	5	1		3	2	2	4	120
12	(1,4) (2,2) (3,3) (4,1)	22	6	1	Sym	3	3	2	3	NA
13	(1,4) (2,1) (3,2) (4,3)	19	7	1		4	3	1	2	128
14	(1,3) (2,4) (3,1) (4,2)	17	6	1		4	2	2	2	112
15	(1,3) (2,2) (3,1) (4,4)	15	8	1		4	4	1	1	131
16	(1,2) (2,3) (3,4) (4,1)	10	7	1		3	4	1	2	137
17	(1,2) (2,4) (3,1) (4,3)	11	7	1		4	3	2	2	86
18	(1,2) (2,1) (3,4) (4,3)	8	7	1		3	4	2	2	109
19	(1,1) (2,3) (3,2) (4,4)	3	8	1	Sym	4	4	2	1	NA
20	(1,1) (2,2) (3,3) (4,4)	1	8	1		4	4	2	1	102
21	(1,1) (2,4) (3,3) (4,2)	6	6	1		3	3	1	3	123
22	(1,4) (2,3) (4,2) (3,1)	23	6	1		4	2	2	3	114

Game #	Payoff Matrix	Shape	Joint Max	PSNEs	Symmetric	Nash Payoff		Dom.	Pareto Optima	Transpose
						Row	Col.			
23	(1,4) (2,2) (4,3) (3,1)	21	7	1		4	3	2	2	87
24	(1,4) (2,1) (4,2) (3,3)	20	6	1		3	3	1	3	127
25	(1,3) (2,4) (4,1) (3,2)	18	6	1		3	2	2	3	118
26	(1,3) (2,2) (4,4) (3,1)	15	8	1	Sym	4	4	2	1	NA
27	(1,3) (2,2) (4,1) (3,4)	16	7	1		3	4	1	2	135
28	(1,3) (2,1) (4,4) (3,2)	13	8	1		4	4	2	1	83
29	(1,2) (2,3) (4,4) (3,1)	9	8	1		4	4	1	1	132
30	(1,2) (2,3) (4,1) (3,4)	10	7	1		3	4	2	2	98
31	(1,2) (2,4) (4,1) (3,3)	12	6	1		3	3	2	3	89
32	(1,2) (2,1) (4,4) (3,3)	7	8	1		4	4	2	1	107
33	(1,1) (2,3) (4,2) (3,4)	3	7	1		3	4	2	2	81
34	(1,1) (2,2) (4,3) (3,4)	2	7	1		3	4	2	2	104
35	(1,1) (2,4) (4,2) (3,3)	6	6	1	Sym	3	3	2	3	NA
36	(1,1) (2,4) (4,3) (3,2)	5	7	1		4	3	1	2	125
37	(1,4) (4,3) (2,2) (3,1)	21	7	1		2	2	1	2	116
38	(1,4) (4,2) (2,3) (3,1)	23	6	1		2	3	1	3	88
39	(1,4) (4,1) (2,2) (3,3)	22	6	0		2.5	2.5	0	3	138
40	(1,4) (4,1) (2,3) (3,2)	24	5	1		2	3	1	4	100
41	(1,3) (4,4) (2,2) (3,1)	15	8	2	Sym	4 2	4 2	0	1	NA
42	(1,3) (4,4) (2,1) (3,2)	13	8	1		4	4	1	1	106
43	(1,3) (4,2) (2,4) (3,1)	17	6	1		2	4	1	2	97
44	(1,3) (4,2) (2,1) (3,4)	14	7	0		2.5	2.5	0	2	126

Game #	Payoff Matrix	Shape	Joint Max	PSNEs	Symmetric	Nash Payoff Row	Nash Payoff Col.	Dom.	Pareto Optima	Transpose
45	(1,3) (4,1) (2,4) (3,2)	18	6	1		2	4	1	3	91
46	(1,2) (4,3) (2,4) (3,1)	11	7	2		4 2	3 4	0	2	133
47	(1,2) (4,3) (2,1) (3,4)	8	7	1		4	3	1	2	94
48	(1,2) (4,4) (2,1) (3,3)	7	8	1		4	4	1	1	82
49	(1,2) (4,1) (2,4) (3,3)	12	6	1		2	4	1	3	121
50	(1,2) (4,1) (2,3) (3,4)	10	7	0		2.5	2.5	0	2	143
51	(1,1) (4,3) (2,2) (3,4)	2	7	1		4	3	1	2	79
52	(1,1) (4,2) (2,3) (3,4)	4	7	1		4	2	1	2	101
53	(1,1) (4,2) (2,4) (3,3)	6	6	2	Sym	4 2	2 4	0	3	NA
54	(1,1) (4,4) (2,2) (3,3)	1	8	1		4	4	1	1	92
55	(1,1) (4,4) (2,3) (3,2)	3	8	2		4 2	4 3	0	1	122
56	(1,4) (4,3) (3,2) (2,1)	19	7	1		3	2	1	2	110
57	(1,4) (4,3) (3,1) (2,2)	21	7	0		2.5	2.5	0	2	141
58	(1,4) (4,2) (3,3) (2,1)	20	6	1		3	3	1	3	84
59	(1,4) (4,1) (3,2) (2,3)	24	5	0		2.5	2.5	0	4	136
60	(1,4) (4,1) (3,3) (2,2)	22	6	1		3	3	1	3	99
61	(1,3) (4,4) (3,2) (2,1)	13	8	2		4 3	4 2	0	1	140
62	(1,3) (4,4) (3,1) (2,2)	15	8	1		4	4	1	1	111
63	(1,3) (4,2) (3,4) (2,1)	14	7	1		3	4	1	2	95
64	(1,3) (4,2) (3,1) (2,4)	17	6	0		2.5	2.5	0	2	130
65	(1,3) (4,1) (3,2) (2,4)	18	6	0		2.5	2.5	0	3	144
66	(1,3) (4,1) (3,4) (2,2)	16	7	1		3	4	1	2	90

Game #	Payoff Matrix	Shape	Joint Max	PSNEs	Symmetric	Nash Payoff Row	Nash Payoff Col.	Dom.	Pareto Optima	Transpose
67	(1,2) (4,3) (3,4) (2,1)	8	7	2		4 3	3 4	0	2	129
68	(1,2) (4,3) (3,1) (2,4)	11	7	1		4	3	1	2	96
69	(1,2) (4,4) (3,3) (2,1)	7	8	2	Sym	4 3	4 3	0	1	NA
70	(1,2) (4,4) (3,1) (2,3)	9	8	1		4	4	1	1	85
71	(1,2) (4,1) (3,4) (2,3)	10	7	1		3	4	1	2	119
72	(1,2) (4,1) (3,3) (2,4)	12	6	0		2.5	2.5	0	3	142
73	(1,1) (4,3) (3,2) (2,4)	5	7	1		4	3	1	2	80
74	(1,1) (4,3) (3,4) (2,2)	2	7	2	Sym	4 3	3 4	0	2	NA
75	(1,1) (4,2) (3,3) (2,4)	6	6	1		4	2	1	3	103
76	(1,1) (4,2) (3,4) (2,3)	4	7	2		3 4	4 2	0	2	139
77	(1,1) (4,4) (3,2) (2,3)	3	8	1		4	4	1	1	93
78	(1,1) (4,4) (3,3) (2,2)	1	8	2		4 3	4 3	0	1	124
79	(1,1) (2,2) (3,4) (4,3)	2	7	1		3	4	1	2	51
80	(1,1) (2,3) (3,4) (4,2)	4	7	1		3	4	1	2	73
81	(1,1) (2,4) (3,2) (4,3)	5	7	1		4	3	2	2	33
82	(1,2) (2,1) (3,3) (4,4)	7	8	1		4	4	1	1	48
83	(1,2) (2,3) (3,1) (4,4)	9	8	1		4	4	2	1	28
84	(1,2) (2,4) (3,3) (4,1)	12	6	1		3	3	1	3	58
85	(1,3) (2,1) (3,2) (4,4)	13	8	1		4	4	1	1	70
86	(1,3) (2,1) (3,4) (4,2)	14	7	1		3	4	2	2	17
87	(1,3) (2,2) (3,4) (4,1)	16	7	1		3	4	2	2	23
88	(1,3) (2,4) (3,2) (4,1)	18	6	1		3	2	1	3	38

Game #	Payoff Matrix	Shape	Joint Max	PSNEs	Symmetric	Nash Payoff Row	Nash Payoff Col.	Dom.	Pareto Optima	Transpose
89	(1,4) (2,1) / (3,3) (4,2)	20	6	1		3	3	2	3	31
90	(1,4) (2,2) / (3,1) (4,3)	21	7	1		4	3	1	2	66
91	(1,4) (2,3) / (3,1) (4,2)	23	6	1		4	2	1	3	45
92	(1,1) (2,2) / (4,4) (3,3)	1	8	1		4	4	1	1	54
93	(1,1) (2,3) / (4,4) (3,2)	3	8	1		4	4	1	1	77
94	(1,2) (2,1) / (4,3) (3,4)	6	7	1		3	4	1	2	47
95	(1,2) (2,4) / (4,3) (3,1)	11	7	1		4	3	1	2	63
96	(1,3) (2,1) / (4,2) (3,4)	14	7	1		3	4	1	2	68
97	(1,3) (2,4) / (4,2) (3,1)	17	6	1		4	2	1	2	43
98	(1,4) (2,1) / (4,3) (3,2)	19	7	1		4	3	2	2	30
99	(1,4) (2,2) / (4,1) (3,3)	22	6	1		3	3	1	3	60
100	(1,4) (2,3) / (4,1) (3,2)	24	5	1		3	2	1	4	40
101	(1,1) (3,2) / (2,4) (4,3)	5	7	1		2	4	1	2	52
102	(1,1) (3,3) / (2,2) (4,4)	1	8	1		4	4	2	1	20
103	(1,1) (3,3) / (2,4) (4,2)	6	6	1		2	4	1	3	75
104	(1,1) (3,4) / (2,2) (4,3)	2	7	1		4	3	2	2	34
105	(1,1) (3,4) / (2,3) (4,2)	4	7	1		2	3	1	2	9
106	(1,2) (3,1) / (2,3) (4,4)	9	8	1		4	4	1	1	42
107	(1,2) (3,3) / (2,1) (4,4)	7	8	1		4	4	2	1	32
108	(1,2) (3,3) / (2,4) (4,1)	12	6	1		2	4	1	3	4
109	(1,2) (3,4) / (2,1) (4,3)	8	7	1		4	3	2	2	18
110	(1,2) (3,4) / (2,3) (4,1)	10	7	1		2	3	1	2	56

Game #	Payoff Matrix	Shape	Joint Max	PSNEs	Symmetric	Nash Payoff Row	Nash Payoff Col.	Dom.	Pareto Optima	Transpose
111	(1,3) (3,1) (2,2) (4,4)	15	8	1		4	4	1	1	62
112	(1,3) (3,1) (2,4) (4,2)	17	6	1		2	4	2	2	14
113	(1,3) (3,2) (2,1) (4,4)	13	8	1		4	4	1	1	8
114	(1,3) (3,2) (2,4) (4,1)	18	6	1		2	4	2	3	22
115	(1,3) (3,4) (2,1) (4,2)	14	7	1		4	2	2	2	2
116	(1,3) (3,4) (2,2) (4,1)	16	7	1		2	2	1	2	37
117	(1,4) (3,1) (2,2) (4,3)	21	7	1		4	3	1	2	5
118	(1,4) (3,1) (2,3) (4,2)	23	6	1		2	3	2	3	25
119	(1,4) (3,2) (2,1) (4,3)	19	7	1		4	3	1	2	71
120	(1,4) (3,2) (2,3) (4,1)	24	5	1		2	3	2	4	11
121	(1,4) (3,3) (2,1) (4,2)	20	6	1		4	2	1	3	49
122	(1,1) (3,2) (4,4) (2,3)	3	8	2		4 3	4 2	0	1	55
123	(1,1) (3,3) (4,2) (2,4)	6	6	1		3	3	1	3	21
124	(1,1) (3,3) (4,4) (2,2)	1	8	2		4 3	4 3	0	1	78
125	(1,1) (3,4) (4,2) (2,3)	4	7	1		3	4	1	2	36
126	(1,2) (3,1) (4,3) (2,4)	11	7	0		2.5	2.5	0	2	44
127	(1,2) (3,3) (4,1) (2,4)	12	6	1		3	3	1	3	24
128	(1,2) (3,4) (4,1) (2,3)	10	7	1		3	4	1	2	13
129	(1,2) (3,4) (4,3) (2,1)	8	7	2		4 3	3 4	0	2	67
130	(1,3) (3,1) (4,2) (2,4)	17	6	0		2.5	2.5	0	2	64
131	(1,3) (3,1) (4,4) (2,2)	15	8	1		4	4	1	1	15
132	(1,3) (3,2) (4,4) (2,1)	13	8	1		4	4	1	1	29

Game #	Payoff Matrix		Shape	Joint Max	PSNEs	Symmetric	Nash Payoff Row	Nash Payoff Col.	Dom.	Pareto Optima	Transpose
133	(1,3) (4,2)	(3,4) (2,1)	14	7	2		3 4	4 2	0	2	46
134	(1,4) (4,2)	(3,1) (2,3)	23	6	0		2.5	2.5	0	3	6
135	(1,4) (4,3)	(3,1) (2,2)	21	7	1		4	3	1	2	27
136	(1,4) (4,1)	(3,2) (2,3)	24	5	0		2.5	2.5	0	4	59
137	(1,4) (4,3)	(3,2) (2,1)	19	7	1		4	3	1	2	16
138	(1,4) (4,1)	(3,3) (2,2)	22	6	0		2.5	2.5	0	3	39
139	(1,1) (2,4)	(4,3) (3,2)	5	7	2		4 2	3 4	0	2	76
140	(1,2) (2,3)	(4,4) (3,1)	9	8	2		4 2	4 3	0	1	61
141	(1,3) (2,2)	(4,1) (3,4)	16	7	0		2.5	2.5	0	2	57
142	(1,4) (2,1)	(4,2) (3,3)	20	6	0		2.5	2.5	0	3	72
143	(1,4) (2,1)	(4,3) (3,2)	19	7	0		2.5	2.5	0	2	50
144	(1,4) (3,1)	(4,2) (2,3)	23	6	0		2.5	2.5	0	3	65

Game # corresponds to the numbering system established in the companion paper
Payoff Matrix gives the normal form of each game with payoffs listed as (row payoff, column payoff)
Shape corresponds to the shape of the payoff set's convex hull as shown in Appendix 1
Joint Max gives the highest possible combined payoff for the two players
Symmetric is marked "Sym" if the game is symmetric, otherwise it is left blank
Nash Payoff lists the payoffs of the noncooperative equilibrium
if there are two equilibria with different payoff sums, the one with the highest sum is listed first
Dom. specifies the number of row and column strategies that are strictly dominated
Pareto Optima gives the number of payoff pairs that are Pareto optimal
Transpose lists the game number corresponding to the transpose of the game shown

Appendix 3

```
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <iomanip>
using namespace std;

const int maxnum=1<<11;
const double di=1<<9;
const int loop=300000;
class cmatr
{
	public: int aul,bul,aur,bur,adl,bdl,adr,bdr;
	public: int d1,d2,d3,d4,d5,d6,d7,d8;
	public: int nash;
	public: double na1,na2,na3,na4;
	public: void setnum(int a1,int a2,int a3,int a4,int a5,int a6,int a7,int a8)
	{
		aul=a1;bul=a2;aur=a3;bur=a4;
		adl=a5;bdl=a6;adr=a7;bdr=a8;
		d1=a1;d2=a2;d3=a3;d4=a4;
		d5=a5;d6=a6;d7=a7;d8=a8;
	}
	public: void print()
	{
		cout<<"cm"<<endl<<"aul "<<aul<<" bul "<<bul<<" aur "<<aur<<" bur "<<bur<<endl;
		cout<<"adl "<<adl<<" bdl "<<bdl<<" adr "<<adr<<" bdr "<<bdr<<endl;
		cout<<d1<<' '<<d2<<' '<<d3<<' '<<d4<<endl<<d5<<' '<<d6<<' '<<d7<<' '<<d8<<endl;
	}
	public: bool eql(cmatr a)
	{
		for (int i=1;i<=4;i++)
		{
			if ((aul==a.aul)&&(bul==a.bul)&&(aur==a.aur)&&(bur==a.bur)
				&&(adl==a.adl)&&(bdl==a.bdl)&&(adr==a.adr)&&(bdr==a.bdr)) return true;
			a.setnum(a.adr,a.bdr,a.aul,a.bul,a.aur,a.bur,a.adl,a.bdl);
		}
		return false;
	}
	public: void rep()
	{
		aul=1;aur=1;bul=1;bur=1;adl=1;adr=1;bdl=1;bdr=1;
		if (d1>d3) aul++; if (d1>d5) aul++; if (d1>d7) aul++;
```

```
            if (d3>d1) aur++; if (d3>d5) aur++; if (d3>d7) aur++;
            if (d5>d1) adl++; if (d5>d3) adl++; if (d5>d7) adl++;
            if (d7>d1) adr++; if (d7>d3) adr++; if (d7>d5) adr++;
            if (d2>d4) bul++; if (d2>d6) bul++; if (d2>d8) bul++;
            if (d4>d2) bur++; if (d4>d6) bur++; if (d4>d8) bur++;
            if (d6>d2) bdl++; if (d6>d4) bdl++; if (d6>d8) bdl++;
            if (d8>d2) bdr++; if (d8>d4) bdr++; if (d8>d6) bdr++;
        }
};
class matr
{
        public: int ul,ur,dl,dr;
        public: int d1,d2,d3,d4;
        public: void randnum()
        {
            ul=rand()%maxnum+1;d1=ul;
            ur=rand()%maxnum+1;d2=ur;
            dl=rand()%maxnum+1;d3=dl;
            dr=rand()%maxnum+1;d4=dr;
        }
        public: bool eql()
        {
            if ((ul==ur)||(ul==dl)||(ul==dr)) return true;
            if ((ur==dl)||(ur==dr)) return true;
            if (dl==dr) return true;
            return false;
        }
        public: void rep()
        {
            int a1=1,a2=1,a3=1,a4=1;
            if (ul>ur) a1++; if (ul>dl) a1++; if (ul>dr) a1++;
            if (ur>ul) a2++; if (ur>dl) a2++; if (ur>dr) a2++;
            if (dl>ul) a3++; if (dl>ur) a3++; if (dl>dr) a3++;
            if (dr>ul) a4++; if (dr>ur) a4++; if (dr>dl) a4++;
            ul=a1;ur=a2;dl=a3;dr=a4;
        }
        public: cmatr combine(matr b)
        {
            cmatr c;
            c.setnum(ul,b.ul,ur,b.ur,dl,b.dl,dr,b.dr);
            return c;
        }
        public: void print()
        {
```

```
		cout<<ul<<' '<<ur<<' '<<dl<<' '<<dr<<endl;
	}
};
	double a[500000],b[500000],a1[500000],b1[500000];
	double oa[150][4000],ob[150][4000],oa1[150][4000],ob1[150][4000];
	int soa[150],sob[150],soa1[150],sob1[150];
	int ty[500000];
	char nu[4]="";
	string    s="out";
	string s2="solution";
	string s1;
	int m=0;
int main()
{
	srand((int)time(0));
	int aul,bul,aur,bur,adl,bdl,adr,bdr;
	int i,j,k,t,tp;
	int res[150];
	double mr[150],mc[150],vr[150],vc[150],sr[150],sc[150],ms[150],vs[150],ss[150];
	double mr1[150],mc1[150],vr1[150],vc1[150],sr1[150],sc1[150],ms1[150],vs1[150],ss1[150];
	double mmr,mmc,vvr,vvc,ssr,ssc,mms,vvs,sss;
	double mmr1,mmc1,vvr1,vvc1,ssr1,ssc1,mms1,vvs1,sss1;
	int rc[150],rc1[150];
	cmatr samp[150];
	cmatr c,d;
	cmatr *p;
	matr pa,pb;

	FILE * fp=NULL;
	fp=fopen("www.txt","r");
	for (i=1;i<=144;i++)
	{
		fscanf(fp,"%d%d%d%d%d%d%d%d",&aul,&bul,&aur,&bur,&adl,&bdl,&adr,&bdr);
		samp[i].setnum(aul,bul,aur,bur,adl,bdl,adr,bdr);
		res[i]=0;
	}
	for (i=1;i<=144;i++)
	{
		fscanf(fp,"%d",&samp[i].nash);
		if ((samp[i].nash==1)||(samp[i].nash==0))
			fscanf(fp,"%lf%lf",&samp[i].na1,&samp[i].na2);
		if (samp[i].nash==2)
			fscanf(fp,"%lf%lf%lf%lf",&samp[i].na1,&samp[i].na2,&samp[i].na3,&samp[i].na4);
```

```
}
fclose(fp);
fp=NULL;
for (i=1;i<=loop;i++)
{
	pa.randnum();
	while (pa.eql()) pa.randnum();
	pb.randnum();
	while (pb.eql()) pb.randnum();
	c=pa.combine(pb);
	c.rep();
	a1[i]=(double) (c.d1+c.d3+c.d5+c.d7)/4;
	b1[i]=(double) (c.d2+c.d4+c.d6+c.d8)/4;
	for (k=1;k<=144;k++)
	{
		if (samp[k].eql(c))
		{
			res[k]++;
			ty[i]=k;
			a[i]=0;
			b[i]=0;
			if (samp[k].nash==0)
			{
				a[i]=c.d1+c.d3+c.d5+c.d7;
				a[i]=a[i]/4;
				b[i]=c.d2+c.d4+c.d6+c.d8;
				b[i]=b[i]/4;
			}else if (samp[k].nash==1)
			{
				if ((c.aul==samp[k].na1)&&(c.bul==samp[k].na2))
				{
					a[i]=c.d1;b[i]=c.d2;
				};
				if ((c.aur==samp[k].na1)&&(c.bur==samp[k].na2))
				{
					a[i]=c.d3;b[i]=c.d4;
				};
				if ((c.adl==samp[k].na1)&&(c.bdl==samp[k].na2))
				{
					a[i]=c.d5;b[i]=c.d6;
				};
				if ((c.adr==samp[k].na1)&&(c.bdr==samp[k].na2))
				{
					a[i]=c.d7;b[i]=c.d8;
```

```
            };
        }else if (samp[k].nash==2)
        {
            if ((c.aul==samp[k].na1)&&(c.bul==samp[k].na2))
            {
                a[i]=c.d1;b[i]=c.d2;
            };
            if ((c.aur==samp[k].na1)&&(c.bur==samp[k].na2))
            {
                a[i]=c.d3;b[i]=c.d4;
            };
            if ((c.adl==samp[k].na1)&&(c.bdl==samp[k].na2))
            {
                a[i]=c.d5;b[i]=c.d6;
            };
            if ((c.adr==samp[k].na1)&&(c.bdr==samp[k].na2))
            {
                a[i]=c.d7;b[i]=c.d8;
            };
            if ((c.aul==samp[k].na3)&&(c.bul==samp[k].na4))
            {
                a[i]+=c.d1;b[i]+=c.d2;
            };
            if ((c.aur==samp[k].na3)&&(c.bur==samp[k].na4))
            {
                a[i]+=c.d3;b[i]+=c.d4;
            };
            if ((c.adl==samp[k].na3)&&(c.bdl==samp[k].na4))
            {
                a[i]+=c.d5;b[i]+=c.d6;
            };
            if ((c.adr==samp[k].na3)&&(c.bdr==samp[k].na4))
            {
                a[i]+=c.d7;b[i]+=c.d8;
            };
            a[i]+=(double) (c.d1+c.d3+c.d5+c.d7)/4;
            b[i]+=(double) (c.d2+c.d4+c.d6+c.d8)/4;
            a[i]=(double)a[i]/3;
            b[i]=(double)b[i]/3;
        };
        break;
    }
  }
}
```

```
for (i=1;i<=144;i++)
{
        mr[i]=0;mc[i]=0;vr[i]=0;vc[i]=0;sr[i]=0;sc[i]=0;
        rc[i]=0;
        mr1[i]=0;mc1[i]=0;vr1[i]=0;vc1[i]=0;sr1[i]=0;sc1[i]=0;
        rc1[i]=0;
        soa[i]=0;sob[i]=0;
        soa1[i]=0;sob1[i]=0;
}
mmr=0;mmc=0;vvr=0;vvc=0;ssr=0;ssc=0;mms=0;vvs=0;sss=0;
mmr1=0;mmc1=0;vvr1=0;vvc1=0;ssr1=0;ssc1=0;mms1=0;vvs1=0;sss1=0;

for (i=1;i<=loop;i++)
{
        a[i]=a[i]/di;
        b[i]=b[i]/di;
        a1[i]=a1[i]/di;
        b1[i]=b1[i]/di;
}
for (i=1;i<=loop;i++)
{
        tp=ty[i];
        rc[tp]++;
        rc1[tp]++;
        sr[tp]+=a[i];sr1[tp]+=a1[i];
        sc[tp]+=b[i];sc1[tp]+=b1[i];
        ss[tp]+=a[i]+b[i];ss1[tp]+=a1[i]+b1[i];
        ssr+=a[i];ssc+=b[i];sss+=a[i]+b[i];
        ssr1+=a1[i];ssc1+=b1[i];sss1+=a1[i]+b1[i];
        soa[tp]++;
        oa[tp][soa[tp]]=a[i];
        sob[tp]++;
        ob[tp][sob[tp]]=b[i];
        soa1[tp]++;
        oa1[tp][soa1[tp]]=a1[i];
        sob1[tp]++;
        ob1[tp][sob1[tp]]=b1[i];
}
for (i=1;i<=144;i++)
{
        mr[i]=sr[i]/rc[i];mr1[i]=sr1[i]/rc1[i];
        mc[i]=sc[i]/rc[i];mc1[i]=sc1[i]/rc1[i];
        ms[i]=ss[i]/rc[i];ms1[i]=ss1[i]/rc1[i];
        sr[i]=0;sc[i]=0;ss[i]=0;sr1[i]=0;sc1[i]=0;ss1[i]=0;
```

```
	}
	mmr=ssr/loop;mmc=ssc/loop;mms=sss/loop;
	mmr1=ssr1/loop;mmc1=ssc1/loop;mms1=sss1/loop;
	ssr=0;ssc=0;sss=0;
	ssr1=0;ssc1=0;sss1=0;
	for (i=1;i<=loop;i++)
	{
		tp=ty[i];
		sr[tp]+=(a[i]-mr[tp])*(a[i]-mr[tp]);sr1[tp]+=(a1[i]-mr1[tp])*(a1[i]-mr1[tp]);
		sc[tp]+=(b[i]-mc[tp])*(b[i]-mc[tp]);sc1[tp]+=(b1[i]-mc1[tp])*(b1[i]-mc1[tp]);

ss[tp]+=(a[i]+b[i]-ms[tp])*(a[i]+b[i]-ms[tp]);ss1[tp]+=(a1[i]+b1[i]-ms1[tp])*(a1[i]+b1[i]-ms1[tp]);
		ssr+=(a[i]-mmr)*(a[i]-mmr);ssr1+=(a1[i]-mmr1)*(a1[i]-mmr1);
		ssc+=(b[i]-mmc)*(b[i]-mmc);ssc1+=(b1[i]-mmc1)*(b1[i]-mmc1);
		sss+=(a[i]+b[i]-mms)*(a[i]+b[i]-mms);sss1+=(a1[i]+b1[i]-mms1)*(a1[i]+b1[i]-mms1);
	}
	vvr=ssr/loop;vvc=ssc/loop;vvs=sss/loop;
	vvr1=ssr1/loop;vvc1=ssc1/loop;vvs1=sss1/loop;
	for (i=1;i<=144;i++)
	{
		vr[i]=sr[i]/rc[i];vr1[i]=sr1[i]/rc1[i];
		vc[i]=sc[i]/rc[i];vc1[i]=sc1[i]/rc1[i];
		vs[i]=ss[i]/rc[i];vs1[i]=ss1[i]/rc1[i];
	}
	cout.setf(ios::fixed);
	fp=fopen("out.txt","w");
	fprintf(fp,"r mean %.3lf c mean %.3lf sum mean %.3lf r var %.3lf c var %.3lf sum
var %.3lf\n",mmr,mmc,mms,vvr,vvc,vvs);
	fprintf(fp,"r mean %.3lf c mean %.3lf sum mean %.3lf r var %.3lf c var %.3lf sum
var %.3lf\n",mmr1,mmc1,mms1,vvr1,vvc1,vvs1);
	for (i=1;i<=144;i++)
	{

		//cout<<setprecision(3)<<"BOX "<<i<<":"<<"r mean:"<<mr[i]<<"    c mean"<<mc[i]<<"
sum mean:"<<ms[i]<<"    r var"<<vr[i]<<"    c var"<<vc[i]<<"    sum var:"<<vs[i]<<endl;
		fprintf(fp,"BOX %d:r mean %.3lf c mean %.3lf sum mean %.3lf r var %.3lf c var %.3lf
sum var %.3lf\n",i,mr[i],mc[i],ms[i],vr[i],vc[i],vs[i]);
		fprintf(fp,"BOX %d:r mean %.3lf c mean %.3lf sum mean %.3lf r var %.3lf c var %.3lf
sum var %.3lf\n",i,mr1[i],mc1[i],ms1[i],vr1[i],vc1[i],vs1[i]);
	}
	fclose(fp);
	for (i=1;i<=144;i++)
	{
		m++;
```

```
        itoa(m,nu,10);
        s1=s+nu+".txt";
        fp=fopen(s1.c_str(),"w");
        fprintf(fp,"BOX %d\n",i);
        for (k=1;k<=soa[i];k++)
            fprintf(fp,"%.3lf    %.3lf\n",oa[i][k],ob[i][k]);
        fclose(fp);
    }
    m=0;
    for (i=1;i<=144;i++)
    {
        m++;
        itoa(m,nu,10);
        s1=s2+nu+".txt";
        fp=fopen(s1.c_str(),"w");
        fprintf(fp,"BOX %d\n",i);
        for (k=1;k<=soa1[i];k++)
            fprintf(fp,"%.3lf    %.3lf\n",oa1[i][k],ob1[i][k]);
        fclose(fp);
    }
    return 0;
}
```

Appendix 4

Table 14: The output for k=9 for the 144 games

Game#	Type	Row mean	Column mean	Sum mean	Row var	Column var	Sum var
1	NE	1.579	1.577	3.156	0.658	0.618	1.297
	Entropy	1.989	1.995	3.984	0.350	0.315	0.684
2	NE	1.609	3.202	4.811	0.623	0.428	1.062
	Entropy	1.996	1.994	3.990	0.324	0.324	0.662
3	NE	3.197	3.228	6.425	0.429	0.403	0.820
	Entropy	1.994	2.011	4.005	0.330	0.325	0.667
4	NE	3.198	1.585	4.784	0.415	0.629	1.044
	Entropy	1.992	1.996	3.987	0.319	0.329	0.647
5	NE	2.377	3.184	5.561	0.639	0.437	1.039
	Entropy	1.986	2.002	3.987	0.321	0.341	0.654
6	NE	1.995	2.001	3.996	0.336	0.347	0.669
	Entropy	1.994	1.999	3.993	0.336	0.347	0.669
7	NE	2.518	2.536	5.053	0.370	0.373	0.741
	Entropy	1.983	1.993	3.976	0.327	0.318	0.645
8	NE	3.217	3.224	6.441	0.422	0.405	0.848
	Entropy	2.023	2.011	4.034	0.333	0.332	0.678
9	NE	2.391	1.603	3.994	0.658	0.613	1.244
	Entropy	1.998	2.011	4.009	0.347	0.314	0.641
10	NE	2.543	2.538	5.081	0.377	0.366	0.732
	Entropy	1.999	2.001	4.001	0.327	0.320	0.630
11	NE	2.398	1.605	4.003	0.631	0.644	1.260
	Entropy	2.004	2.000	4.004	0.335	0.328	0.666
12	NE	2.409	2.399	4.808	0.647	0.639	1.293
	Entropy	2.004	1.994	3.999	0.340	0.338	0.673
13	NE	3.192	2.410	5.601	0.415	0.634	1.051
	Entropy	2.007	2.012	4.019	0.324	0.336	0.669
14	NE	3.200	1.606	4.807	0.426	0.648	1.058
	Entropy	1.998	1.995	3.993	0.329	0.328	0.668
15	NE	3.192	3.201	6.393	0.430	0.420	0.866
	Entropy	2.010	1.986	3.996	0.342	0.319	0.666
16	NE	2.390	3.234	5.624	0.655	0.414	1.034
	Entropy	1.989	2.014	4.003	0.335	0.331	0.664
17	NE	3.206	2.420	5.625	0.435	0.607	1.048
	Entropy	2.014	2.015	4.029	0.341	0.320	0.660
18	NE	2.378	3.217	5.595	0.641	0.424	1.036
	Entropy	1.986	2.013	3.998	0.341	0.334	0.663
19	NE	3.170	3.199	6.369	0.440	0.432	0.868
	Entropy	1.970	1.999	3.969	0.325	0.327	0.643
20	NE	3.209	3.196	6.405	0.435	0.416	0.841
	Entropy	2.005	2.008	4.013	0.332	0.312	0.638
21	NE	2.404	2.392	4.796	0.620	0.641	1.303
	Entropy	2.004	2.004	4.008	0.337	0.341	0.695
22	NE	3.199	1.599	4.798	0.419	0.645	1.050
	Entropy	1.990	1.995	3.985	0.326	0.342	0.652
23	NE	3.236	2.389	5.625	0.434	0.649	1.090
	Entropy	2.025	1.985	4.010	0.329	0.328	0.653

Game#	Type	Row mean	Column mean	Sum mean	Row var	Column var	Sum var
26	NE	3.192	3.184	6.375	0.443	0.430	0.848
	Entropy	1.995	1.989	3.984	0.335	0.340	0.675
27	NE	2.395	3.204	5.599	0.648	0.443	1.101
	Entropy	1.988	2.000	3.988	0.329	0.356	0.673
28	NE	3.208	3.223	6.431	0.425	0.424	0.840
	Entropy	1.999	2.017	4.016	0.324	0.343	0.692
29	NE	3.215	3.203	6.417	0.418	0.420	0.836
	Entropy	2.010	2.002	4.012	0.331	0.328	0.634
30	NE	2.379	3.207	5.586	0.644	0.407	1.055
	Entropy	1.987	2.006	3.993	0.337	0.333	0.666
31	NE	2.412	2.376	4.788	0.643	0.630	1.302
	Entropy	2.010	1.988	3.997	0.331	0.326	0.674
32	NE	3.192	3.198	6.390	0.420	0.417	0.822
	Entropy	1.984	1.991	3.975	0.319	0.332	0.637
33	NE	2.386	3.206	5.592	0.629	0.420	1.112
	Entropy	1.991	1.995	3.987	0.326	0.303	0.636
34	NE	2.413	3.209	5.621	0.628	0.405	1.042
	Entropy	2.015	2.010	4.024	0.333	0.321	0.630
35	NE	2.402	2.430	4.832	0.618	0.648	1.283
	Entropy	1.989	2.011	4.000	0.320	0.324	0.642
36	NE	3.193	2.418	5.611	0.439	0.644	1.085
	Entropy	1.993	2.003	3.996	0.351	0.331	0.701
37	NE	1.593	1.616	3.209	0.620	0.646	1.262
	Entropy	1.997	2.017	4.015	0.326	0.323	0.661
38	NE	1.580	2.406	3.986	0.623	0.639	1.218
	Entropy	1.988	2.004	3.992	0.318	0.325	0.619
39	NE	1.999	2.020	4.019	0.326	0.334	0.663
	Entropy	1.997	2.019	4.016	0.326	0.334	0.663
40	NE	1.604	2.412	4.016	0.663	0.647	1.309
	Entropy	2.000	2.006	4.005	0.347	0.336	0.672
41	NE	2.278	2.268	4.546	0.363	0.332	0.695
	Entropy	2.008	1.998	4.005	0.346	0.315	0.655
42	NE	3.215	3.213	6.428	0.413	0.414	0.853
	Entropy	2.011	1.988	4.000	0.326	0.319	0.668
43	NE	1.620	3.204	4.824	0.612	0.429	1.063
	Entropy	2.005	1.994	3.999	0.334	0.333	0.670
44	NE	2.029	2.017	4.046	0.327	0.331	0.645
	Entropy	2.028	2.016	4.044	0.327	0.331	0.645
45	NE	1.587	3.218	4.805	0.634	0.406	1.045
	Entropy	1.995	2.013	4.008	0.328	0.314	0.645
46	NE	2.269	2.556	4.825	0.349	0.372	0.715
	Entropy	2.003	2.023	4.025	0.331	0.339	0.669
47	NE	3.202	2.416	5.618	0.445	0.605	1.040
	Entropy	1.998	1.991	3.989	0.339	0.313	0.624
48	NE	3.196	3.193	6.389	0.417	0.418	0.809
	Entropy	1.983	2.007	3.989	0.327	0.325	0.618
49	NE	1.592	3.193	4.785	0.638	0.440	1.070
	Entropy	1.992	1.979	3.971	0.338	0.331	0.663
50	NE	1.989	2.004	3.992	0.332	0.335	0.672
	Entropy	1.987	2.002	3.990	0.332	0.335	0.672

Game#	Type	Row mean	Column mean	Sum mean	Row var	Column var	Sum var
51	NE	3.207	2.373	5.579	0.406	0.626	1.082
	Entropy	2.000	1.991	3.991	0.317	0.328	0.657
52	NE	3.201	1.580	4.781	0.419	0.638	1.090
	Entropy	2.011	1.995	4.005	0.333	0.332	0.674
53	NE	2.278	2.250	4.528	0.340	0.344	0.663
	Entropy	2.009	1.982	3.991	0.325	0.323	0.635
54	NE	3.202	3.177	6.380	0.437	0.439	0.889
	Entropy	2.007	2.006	4.013	0.331	0.339	0.664
55	NE	2.253	2.521	4.774	0.365	0.397	0.739
	Entropy	1.985	1.994	3.980	0.345	0.347	0.670
56	NE	2.433	1.600	4.034	0.623	0.633	1.308
	Entropy	2.024	2.002	4.026	0.342	0.325	0.696
57	NE	1.989	2.009	3.999	0.363	0.330	0.681
	Entropy	1.988	2.008	3.996	0.363	0.330	0.681
58	NE	2.411	2.431	4.841	0.620	0.655	1.315
	Entropy	2.002	2.024	4.026	0.310	0.346	0.668
59	NE	2.008	2.015	4.023	0.313	0.329	0.656
	Entropy	2.006	2.014	4.020	0.313	0.329	0.656
60	NE	2.378	2.410	4.788	0.639	0.661	1.277
	Entropy	1.992	1.995	3.988	0.328	0.344	0.677
61	NE	2.527	2.279	4.806	0.369	0.363	0.724
	Entropy	1.991	2.010	4.001	0.330	0.341	0.659
62	NE	3.187	3.210	6.397	0.432	0.414	0.832
	Entropy	1.981	1.992	3.973	0.324	0.323	0.646
63	NE	2.359	3.210	5.569	0.633	0.405	1.028
	Entropy	1.972	2.006	3.978	0.328	0.330	0.660
64	NE	2.022	2.007	4.028	0.340	0.344	0.669
	Entropy	2.020	2.005	4.025	0.340	0.344	0.669
65	NE	1.994	1.986	3.980	0.315	0.354	0.643
	Entropy	1.992	1.985	3.977	0.315	0.354	0.643
66	NE	2.415	3.178	5.594	0.644	0.435	1.073
	Entropy	2.016	1.982	3.998	0.340	0.338	0.692
67	NE	2.557	2.529	5.086	0.371	0.379	0.763
	Entropy	2.015	1.999	4.014	0.331	0.340	0.685
68	NE	3.193	2.420	5.614	0.434	0.604	1.057
	Entropy	1.994	2.008	4.001	0.335	0.316	0.655
69	NE	2.522	2.562	5.084	0.379	0.356	0.734
	Entropy	1.984	2.021	4.005	0.329	0.317	0.657
70	NE	3.195	3.214	6.409	0.435	0.425	0.885
	Entropy	1.989	2.000	3.989	0.331	0.324	0.658
71	NE	2.427	3.204	5.631	0.633	0.400	1.079
	Entropy	2.018	2.000	4.018	0.336	0.313	0.656
72	NE	2.007	1.997	4.004	0.340	0.323	0.638
	Entropy	2.006	1.996	4.002	0.340	0.323	0.639
73	NE	3.196	2.382	5.578	0.424	0.659	1.057
	Entropy	1.996	1.989	3.984	0.330	0.342	0.670
74	NE	2.536	2.550	5.086	0.373	0.361	0.731
	Entropy	2.003	2.015	4.017	0.329	0.328	0.666
75	NE	3.189	1.618	4.807	0.439	0.625	1.036
	Entropy	1.983	2.007	3.990	0.337	0.316	0.640

Game#	Type	Row mean	Column mean	Sum mean	Row var	Column var	Sum var
76	NE	2.532	2.277	4.809	0.380	0.340	0.723
	Entropy	2.001	2.005	4.006	0.340	0.321	0.658
77	NE	3.212	3.197	6.409	0.429	0.426	0.869
	Entropy	2.002	1.995	3.996	0.332	0.332	0.657
78	NE	2.504	2.537	5.041	0.379	0.378	0.756
	Entropy	1.968	2.001	3.969	0.334	0.332	0.669
79	NE	2.427	3.198	5.625	0.637	0.413	1.054
	Entropy	2.011	2.003	4.014	0.325	0.319	0.649
80	NE	2.412	3.201	5.613	0.633	0.422	1.050
	Entropy	2.025	1.999	4.024	0.328	0.333	0.651
81	NE	3.196	2.408	5.605	0.433	0.637	1.074
	Entropy	1.986	2.006	3.992	0.332	0.337	0.676
82	NE	3.190	3.178	6.368	0.433	0.454	0.877
	Entropy	2.009	1.990	3.999	0.345	0.344	0.684
83	NE	3.213	3.205	6.419	0.449	0.426	0.867
	Entropy	2.021	1.992	4.013	0.349	0.326	0.678
84	NE	2.398	2.383	4.780	0.633	0.642	1.307
	Entropy	2.001	1.988	3.988	0.329	0.339	0.686
85	NE	3.188	3.198	6.386	0.430	0.433	0.885
	Entropy	1.994	2.000	3.994	0.326	0.327	0.653
86	NE	2.437	3.180	5.617	0.649	0.450	1.057
	Entropy	2.022	1.974	3.997	0.336	0.337	0.664
87	NE	2.399	3.202	5.602	0.610	0.424	1.015
	Entropy	2.003	2.002	4.005	0.327	0.333	0.646
88	NE	2.387	1.621	4.008	0.632	0.636	1.257
	Entropy	1.994	2.012	4.006	0.324	0.333	0.661
89	NE	2.417	2.420	4.837	0.630	0.649	1.335
	Entropy	2.011	2.007	4.019	0.325	0.340	0.688
90	NE	3.194	2.388	5.582	0.443	0.633	1.120
	Entropy	1.988	2.000	3.988	0.335	0.338	0.696
91	NE	3.211	1.624	4.835	0.424	0.644	1.092
	Entropy	2.003	2.020	4.023	0.321	0.333	0.666
92	NE	3.178	3.194	6.372	0.460	0.419	0.905
	Entropy	1.979	1.990	3.969	0.339	0.322	0.664
93	NE	3.211	3.217	6.428	0.400	0.427	0.832
	Entropy	2.000	2.028	4.028	0.318	0.335	0.660
94	NE	2.400	3.198	5.598	0.647	0.425	1.077
	Entropy	1.993	2.004	3.997	0.335	0.341	0.683
95	NE	3.186	2.417	5.602	0.422	0.651	1.058
	Entropy	1.980	2.010	3.991	0.329	0.342	0.654
96	NE	2.393	3.168	5.561	0.624	0.443	1.067
	Entropy	1.985	1.970	3.956	0.330	0.341	0.646
97	NE	3.209	1.581	4.790	0.416	0.622	1.052
	Entropy	2.007	1.980	3.988	0.328	0.322	0.656
98	NE	3.193	2.404	5.597	0.422	0.612	1.024
	Entropy	2.002	2.007	4.009	0.336	0.324	0.643
99	NE	2.391	2.423	4.814	0.651	0.623	1.260
	Entropy	1.994	2.018	4.012	0.337	0.328	0.660
100	NE	2.372	1.602	3.974	0.663	0.651	1.308
	Entropy	1.984	1.993	3.977	0.344	0.337	0.677

Game#	Type	Row mean	Column mean	Sum mean	Row var	Column var	Sum var
101	NE	1.594	3.206	4.800	0.633	0.422	1.038
	Entropy	1.997	1.995	3.992	0.331	0.330	0.676
102	NE	3.209	3.201	6.411	0.451	0.437	0.896
	Entropy	2.025	2.011	4.036	0.350	0.344	0.685
103	NE	1.603	3.176	4.779	0.658	0.419	1.084
	Entropy	2.002	1.998	4.000	0.346	0.323	0.661
104	NE	3.206	2.372	5.578	0.425	0.660	1.116
	Entropy	2.010	1.989	3.999	0.327	0.345	0.677
105	NE	1.568	2.401	3.969	0.617	0.628	1.285
	Entropy	1.974	2.022	3.997	0.320	0.329	0.660
106	NE	3.214	3.215	6.430	0.422	0.431	0.853
	Entropy	2.015	2.009	4.024	0.338	0.337	0.681
107	NE	3.186	3.228	6.414	0.431	0.426	0.860
	Entropy	1.990	2.019	4.009	0.329	0.335	0.696
108	NE	1.618	3.206	4.824	0.636	0.411	1.031
	Entropy	2.013	1.993	4.006	0.338	0.316	0.665
109	NE	3.205	2.405	5.610	0.427	0.651	1.053
	Entropy	2.003	2.012	4.016	0.332	0.339	0.649
110	NE	1.571	2.403	3.974	0.671	0.645	1.268
	Entropy	1.989	1.991	3.980	0.351	0.327	0.652
111	NE	3.226	3.205	6.431	0.410	0.430	0.834
	Entropy	2.019	1.987	4.006	0.325	0.333	0.656
112	NE	1.612	3.215	4.827	0.633	0.420	0.995
	Entropy	2.005	2.021	4.026	0.333	0.333	0.637
113	NE	3.192	3.206	6.398	0.436	0.423	0.827
	Entropy	1.990	1.994	3.984	0.335	0.335	0.664
114	NE	1.611	3.209	4.820	0.641	0.411	1.040
	Entropy	1.996	2.006	4.002	0.349	0.334	0.677
115	NE	3.209	1.630	4.839	0.420	0.623	1.041
	Entropy	2.001	2.027	4.028	0.325	0.332	0.654
116	NE	1.600	1.597	3.197	0.651	0.637	1.252
	Entropy	1.998	1.991	3.989	0.336	0.340	0.664
117	NE	3.212	2.388	5.600	0.400	0.652	1.089
	Entropy	1.999	2.000	3.998	0.316	0.342	0.676
118	NE	1.628	2.409	4.037	0.639	0.640	1.278
	Entropy	2.017	2.002	4.018	0.344	0.338	0.700
119	NE	3.181	2.389	5.570	0.452	0.649	1.115
	Entropy	1.975	1.996	3.971	0.344	0.345	0.688
120	NE	1.617	2.405	4.022	0.673	0.622	1.301
	Entropy	2.004	2.007	4.012	0.341	0.325	0.664
121	NE	3.195	1.553	4.748	0.434	0.599	1.078
	Entropy	1.993	1.968	3.961	0.323	0.327	0.662
122	NE	2.552	2.254	4.806	0.382	0.350	0.719
	Entropy	2.014	1.990	4.004	0.344	0.331	0.658
123	NE	2.359	2.395	4.754	0.631	0.649	1.291
	Entropy	1.969	1.996	3.965	0.325	0.336	0.645
124	NE	2.532	2.539	5.071	0.369	0.378	0.754
	Entropy	2.002	2.009	4.011	0.332	0.335	0.680
125	NE	2.384	3.195	5.579	0.624	0.422	1.032
	Entropy	1.987	2.007	3.994	0.320	0.335	0.634

Game#	Type	Row mean	Column mean	Sum mean	Row var	Column var	Sum var
126	NE	2.003	2.000	4.003	0.339	0.335	0.688
	Entropy	2.002	1.999	4.000	0.339	0.335	0.688
127	NE	2.408	2.426	4.834	0.654	0.641	1.284
	Entropy	2.011	2.016	4.027	0.342	0.332	0.666
128	NE	2.383	3.202	5.585	0.672	0.410	1.032
	Entropy	1.970	2.008	3.978	0.342	0.330	0.656
129	NE	2.537	2.551	5.088	0.382	0.375	0.774
	Entropy	2.001	2.022	4.023	0.345	0.335	0.695
130	NE	2.002	2.020	4.022	0.333	0.337	0.672
	Entropy	2.000	2.019	4.019	0.333	0.337	0.672
131	NE	3.218	3.205	6.423	0.414	0.426	0.817
	Entropy	2.017	1.991	4.008	0.335	0.325	0.671
132	NE	3.189	3.200	6.389	0.435	0.428	0.870
	Entropy	1.999	1.997	3.996	0.338	0.332	0.679
133	NE	2.545	2.287	4.832	0.354	0.348	0.713
	Entropy	2.007	2.020	4.027	0.322	0.327	0.658
134	NE	2.021	1.996	4.018	0.333	0.328	0.648
	Entropy	2.020	1.995	4.015	0.333	0.328	0.649
135	NE	3.225	2.396	5.621	0.419	0.642	1.034
	Entropy	2.014	1.997	4.011	0.327	0.337	0.634
136	NE	1.999	1.981	3.980	0.346	0.321	0.679
	Entropy	1.998	1.979	3.977	0.346	0.321	0.679
137	NE	3.212	2.384	5.596	0.416	0.645	1.070
	Entropy	2.002	1.994	3.995	0.320	0.343	0.657
138	NE	2.004	1.999	4.002	0.340	0.334	0.703
	Entropy	2.002	1.997	3.999	0.340	0.334	0.703
139	NE	2.271	2.543	4.814	0.350	0.376	0.737
	Entropy	2.003	1.998	4.001	0.327	0.333	0.667
140	NE	2.285	2.521	4.805	0.345	0.372	0.710
	Entropy	2.017	1.998	4.015	0.331	0.330	0.651
141	NE	1.977	1.991	3.968	0.320	0.344	0.674
	Entropy	1.975	1.990	3.965	0.320	0.344	0.674
142	NE	2.007	2.002	4.009	0.348	0.323	0.679
	Entropy	2.005	2.001	4.006	0.348	0.323	0.679
143	NE	2.011	2.009	4.020	0.328	0.327	0.645
	Entropy	2.009	2.008	4.017	0.328	0.327	0.645
144	NE	2.014	2.017	4.031	0.319	0.329	0.641
	Entropy	2.013	2.016	4.028	0.319	0.329	0.642