Eliciting Collusion Features

Jussi Laasonen
Department of Information Technology
FI-20014 University of Turku, Finland
jussi.laasonen@utu.fi

Timo Knuutila
Department of Information Technology
FI-20014 University of Turku, Finland
timo.knuutila@utu.fi

Jouni Smed
Department of Information Technology
FI-20014 University of Turku, Finland
jouni.smed@utu.fi

ABSTRACT

Collusion is covert co-operation between participants of a game. It poses technical, game design, and communal problems to multiplayer games that do not allow the players to share knowledge or resources with other players. In this paper, we analyse experimental data from a simple two-dimensional game using synthetic players to elicit features that indicate collusion.

Categories and Subject Descriptors

I.2.1 [**Applications and Expert Systems**]: Games; K.4.2 [**Social Issues**]: Abuse and crime involving computers; K.8 [**Personal Computing**]: Games

General Terms

Collusion, computer games, multiplaying, cheating prevention, online gaming

1. INTRODUCTION

When the rules of a game forbid the players to co-operate, any attempt of covert co-operation is called *collusion*. The players who are colluding (i.e., whose goal is to win together or to help one another to win) are called *colluders*. Collusion poses a major threat to games that assume that the players aim at individual goals individually, because many types of collusion are impossible to prevent in real time. Even detecting collusion can require discerning and understanding the player's motivation – which is often an impossible task for human beings, too. For this reason, collusion is usually detected only afterwards by studying the behaviour of the players and recognizing characteristic patterns that indicate forbidden co-operation.

When the players of a game decide to collude, they make an agreement on the terms of collusion [9]. This agreement has the following four components:

Consent How do the players agree on collusion?

- *Express collusion*: The colluders make an explicit hidden agreement on co-operation before or during the game.
- *Tacit collusion*: The colluders have made no agreement but act towards a mutually beneficial goal (e.g., try to force the weakest player out of the game).

Scope What areas of the game does the collusion affect?

- *Total collusion*: The colluders co-operate on all areas of the game.
- *Partial collusion*: The colluders co-operate only on certain areas and compete on others (e.g., sharing resource pools but competing elsewhere).

Duration When does the collusion begin and end?

- *Enduring*: Collusion agreement lasts for the duration of the game.
- *Opportunistic*: Collusion agreements are formed, disbanded, and altered continuously.

Content What is being exchanged, traded, or donated in the collusion?

- *Knowledge*: The colluders share expertise (e.g., inside information on the game mechanics), in-game information (e.g., the colluders inform one another the whereabouts of the non-colluding players) or stance (e.g., the colluders agree on playing "softly" against one another).
- *Resources*: The colluders share in-game resources (e.g., donating digital assets to one another) or extra-game resources (e.g., a sweatshop is playing a character which will be sold later for real-world money).

We must also discern the roles of the partakers – *players* and *participants* – of the game: A player in a game can be controlled by one or more participants, and a participant can control one or more players in a game [10]. There are two types of collusion: (i) collusion among the players which happens *inside* the game, and (ii) collusion among the participants which happens *outside* the game. To detect player collusion, we have to analyse whether the players' behaviour diverges from what is reasonably expectable. To detect participant collusion, we have to analyse the participants behind the players to detect whether they are colluding.

DISIO 2011, March 21, Barcelona, Spain

DOI 10.4108/icst.simutools.2011.245528

If the content of the collusion agreement is an in-game resource, it is possible to detect by analysing the game session logs [3]. Detecting shared knowledge, however, is more difficult, because we can only observe the decisions made by the players in order to discern the intention behind the decision-making.

To find out features indicating collusion we can collect real-world data – but it is hard to ascertain what has been the driving force behind the human players at a given time. Another approach is to use synthetic players [7] some of which have been programmed to collaborate. This way we can create a large amount of test data with known cooperative properties. Naturally, the results obtained for artificial data will be later evaluated and verified using real examples.

The idea behind our approach is [10]:

1. Generate game data with different number of players, colluders, game types, and collusion strategies.
2. Devise detection methods.
3. Run the detection method against the data to get results.
4. Compare accuracy: How many (if any) of the colluders got detected.
5. Compare swiftness: How quickly the colluders were detected.

Apart from games susceptible to collusion such as poker [3, 11, 12] and bridge [13], collusion have been addressed also in the context of tournaments [4], multiple choice examinations [1] and covert communication channels [14].

The plan of this paper is as follows: We begin by presenting the rules of the game Pakuhaku and the computer-controlled players, which are used in our experiments. After that, we look at how we can measure the game state using utility functions or evaluation functions. They are used when we analyse the experimental data from Pakuhaku. Finally, we conclude the paper with a discussion and line out steps for future work.

2. PAKUHAKU

The Pakuhaku game, introduced in [10], is used to generate test data for the collusion detection methods. The game is inspired by the classical computer game *Pac-Man* [5]. Some features of the original game have been omitted – such as the maze, ghosts, and power-up cherries – and we allow multiple players to take part in the game. The goal of the game is simple: eat as many pills scattered in the game world as possible.

At each turn, each player makes a decision on the direction where to go. This decision is based on knowledge about the game world, which is limited by a fog-of-war (i.e., the player's immediate surroundings). The system provides a communication channel, where the colluders can exchange one message in each turn. Communication can be used to assist, restrict and guide the co-colluders. We have also extended the game so that players can shoot at other players with a beam weapon that paralyses its target. The beam has an unlimited range and it hits all the players along its trajectory (i.e., there is no "friendly fire").

In our experiments, the game type can be one of the following:

- *Evenly distributed pills*: A given number of pills are positioned evenly into rows and columns into the game world. The game ends when all the pills have been eaten. In our test cases, the number of pills is 64.
- *Dispenser competition*: One pill is repositioned to the game world into a random position after it gets eaten. The game ends when the sum of pills eaten by all the players reaches a given number. In our test cases, the number of eaten pills is 64.

We use a game world of the size 800×800 units in our experiments. The players and pills are modelled as circles with the radii of 10 and 3 units, respectively. The players' fog-of-war has a radius of 75 units. The players can move freely 5 units in any direction at each turn, send one message to other players and shoot. The beam weapon has a loading time of 20 turns and it is not loaded when the game starts. All the players begin at the centre of the game world.

All the players use the same approach: If there are visible pills, move towards the nearest pill; otherwise, select randomly a target position that is not in the tabu list [2]. A position is considered to be tabu, if its distance to a position in the tabu list is less than the radius of the fog-of-war. If a pill gets eaten, add the position of the pill to a tabu list. Also, add the positions of all visible players the tabu list. If there are no visible pills, move towards the previously selected target position. When the target position is reached, select randomly a new target position that is not in the tabu list. To prevent possible lock-ups the tabu list is emptied prematurely, if no permitted position is found after 1,000 trials.

The players and colluders have four different implementations of this basic behaviour to guide their movements:

- *Default player*: The player keeps its own tabu list.
- *Tabu colluder*: The player informs all new additions to its tabu list to its co-colluders (i.e., the colluders have a shared tabu list).
- *Area-of-interest colluder*: When the game begins, the colluders divide the game world into non-overlapping areas-of-interest (AOI), which are either evenly sized rectangles or – if that is not possible – evenly sized horizontal slices. During the game the player acts like a tabu colluder but does not leave its designated AOI.
- *Blocking colluder*: When the game begins, one of the colluders is elected as a leader, who will play like a tabu colluder. The other colluders also play like the tabu colluder but try to keep near the non-colluding players so that they can prevent them from reaching pills by eating them first.

Non-colluding players shoot at a randomly selected visible player immediately when their weapon is loaded. Colluding players engage in soft play and do not shoot at their co-colluders – although they might get hit by collateral fire.

3. MEASURING THE GAME STATE

To detect collusion we need data that we can analyse by applying two kinds of measures:

- *Utility function* (or pay-off function) that gives the final numeric value for the game [6, p. 162], [8, p. 76].

- *Evaluation function* that correlates strongly in each game state to the actual chance of winning [6, pp. 171–173], [8, pp. 78–80].

In the following we present utility functions and candidate features for Pakuhaku.

Let A be a set of all players, $P \subseteq A$ be the set of non-colluding players and $Q \subseteq A$ be the set of colluding players such that $P \cap Q = \varnothing$ and $P \cup Q = A$. Let $p \in P$ be a non-colluding player and $q \in Q$ be a colluding player. Moreover, let

- $r_{\max}$: the number of rounds in a single game
- w, h: the width and height of the game world
- $\text{wins}(p)$: the number of games the player p has won
- $\text{pills}(p)$: the number of pills the player p has eaten
- $\text{hits}(p)$: the number of hits the player p has given
- $\text{hits}(p, q)$: the number of hits the player p has given to the player q
- $\text{deaths}(p)$: the number of hits the player p has taken
- $\ell_{p,r} = (x_{p,r}, y_{p,r})$: the location of player p at the round r $(1 \leq r \leq r_{\max})$
- $\bar{v}_p$: the velocity vector (from the previous position to the current position) of the player p
- $\text{box}(p)$: the bounding box of the player p (i.e., a square centered on the player's location with the width and height of two times the radius of the fog-of-war)
- $\text{box}(L)$: the bounding box of the set of locations L
- $\text{area}(b)$: the area of the bounding box b

3.1 Utility Functions

In our earlier study we showed that the utility (or pay-off) increases for the colluding players when more players engage in collusion until an optimal number of colluders is reached and, after that, it approaches asymptotically a fair-play pay-off [10]. To analyse how collusion affects the Pakuhaku game, we calculate utility functions for different numbers of colluders in different game types.

Collusion pay-off is measured by two utility functions:

$$u_p = \sum_{q \in Q} \text{pills}(q)/|Q| - \sum_{p \in P} \text{pills}(p)/|P|,$$

which measures how efficient the players are at accumulating score, and

$$u_w = \sum_{q \in Q} \text{wins}(q)/|Q| - \sum_{p \in P} \text{wins}(p)/|P|,$$

which measures how efficient the players are at winning games. Even if u_p has a high value, it does not necessarily entail that u_w also has a high value.

In our analysis, we also use a variant of u_p, where the denominator assumes that the pills are distributed evenly to all players:

$$u'_p = \sum_{Q} \text{pills}(q)/|Q| - e/|A|,$$

where e is the total number of pills. In our test cases, $e = 64$.

The utility function u_w can be used to calculate a pay-off when joining to the game has an associated cost c:

$$u_c = z u_w - c|Q|$$

where z is the prize for winning. We can let $z = |A|c = (|P| + |Q|)c$, which is a quite commonly used "banker's fee":

$$\begin{aligned} u_c &= (|P| + |Q|)cu_w - c|Q| \\ &= |P|cu_w + |Q|cu_w - c|Q| \\ &= c(|P|u_w + |Q|(u_w - 1)). \end{aligned}$$

Now, if $c = 1$, we get $u_c = |P|u_w + |Q|(u_w - 1)$. As we can see, u_c is a linearly growing function of u_w.

3.2 Candidate features

We use the following candidate features, which are calculated at the end of game over the players in subset S:

- $\text{scoreTotal}(S) = \sum_{s \in S} \text{pills}(s)/|S|$
- $\text{scoreDelta}(S) = \text{stddev}_{s \in S}\, \text{pills}(s)$
- $\text{deathsTotal}(S) = \sum_{s \in S} \text{deaths}(s)/|S|$
- $\text{deathsDelta}(S) = \text{stddev}_{s \in S}\, \text{deaths}(s)$
- $\text{totalUnion}(S) = \text{area}\left(\bigcup_{s \in S} \text{box}(\{\ell_{s,r} | 1 \leq r \leq r_{\max}\})\right) / wh$ (measures how much of the playing area subset S as a whole has explored)
- $\text{totalIntersection}(S) = \text{area}(\bigcap_{s \in S} \text{box}(\{\ell_{s,r} | 1 \leq r \leq r_{\max}\})) / wh$ (measures how much of the playing area is explored by all the player in subset S; if this value is small, the players may have been co-ordinating to keep distance between each other)

The values needed in the metric calculations are calculated once per turn r and aggregated at end of the game. The aggregation includes mean, standard deviation, minimum and maximum (e.g., for "union" the metrics are called "union", "unionStdDev", "unionMin" and "unionMax", respectively). The metrics hitsMutual and hitMutualDelta are aggregated using mean only.

- $\text{union} = (\max_{s \in S}(x_{s,r}) - \min_{s \in S}(x_{s,r}))(\max_{s \in S}(y_{s,r}) - \min_{s \in S}(y_{s,r}))/wh$
- $\text{intersection} = \text{area}\left(\bigcap_{s \in S} \text{box}(s)\right)/wh$
- $\text{distance} = \max_{s,t \in S}(|\bar{v}_s - \bar{v}_t|)$
- $\text{directions} = \left|\sum_{s \in S} \bar{v}_s\right|$
- $\text{angle} = \text{median}_{\phi_i \in \Phi, \phi_i \leq \phi_{u+1}}(\phi_2 - \phi_1, \cdots, \phi_{|S|} - \phi_{|S|-1}, 2\pi - \phi_{|S|} + \phi_1)$, where $\Phi = \left\{\phi_s = \cos^{-1}\left(\frac{\bar{v}_s \cdot \bar{v}_s}{|\bar{v}_s|^2}\right) | s \in S\right\}$.
- $\text{hitsMutual} = \sum_{p \in S} \sum_{q \in S, p \neq q} \text{hits}(p, q)$
- $\text{hitsMutualDelta} = \text{stddev}(\text{hits}(p, q) \,|\, p, q \in S, p \neq q)$
- $\text{hitsAll} = \sum_{s \in S} \text{hits}(s)$
- $\text{hitsAllDelta} = \text{stddev}_{s \in S}(\text{hits}(s))$
- $\text{meanX} = \text{mean}(x_{s,r}), s \in S$
- $\text{meanY} = \text{mean}(y_{s,r}), s \in S$

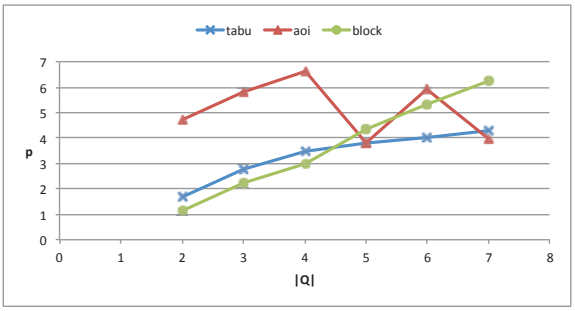

Figure 1: Utility function u_p of the $|Q|$ colluders in an evenly distributed game world.

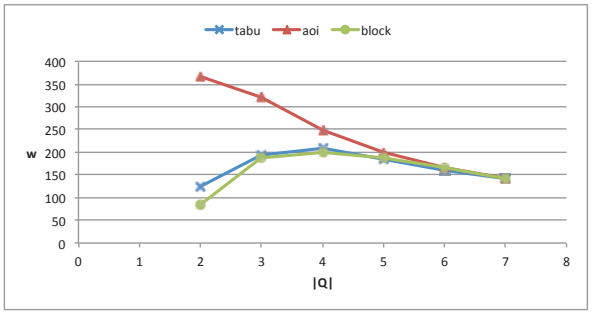

Figure 2: Utility function u_w of the $|Q|$ colluders in an evenly distributed game world.

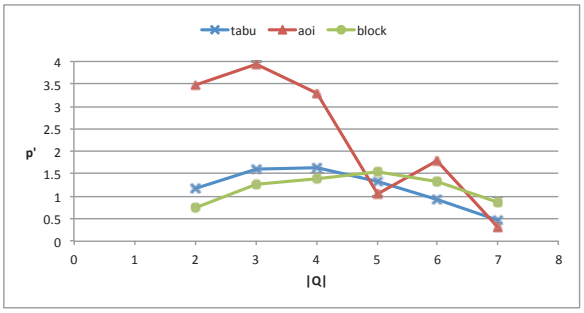

Figure 3: Utility function u'_p of the $|Q|$ colluders in an evenly distributed game world.

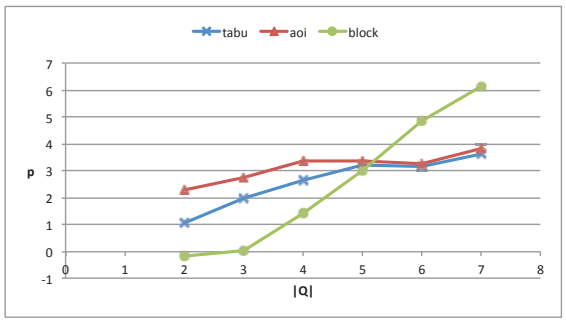

Figure 4: Utility function u_p of the $|Q|$ colluders in a dispenser competition.

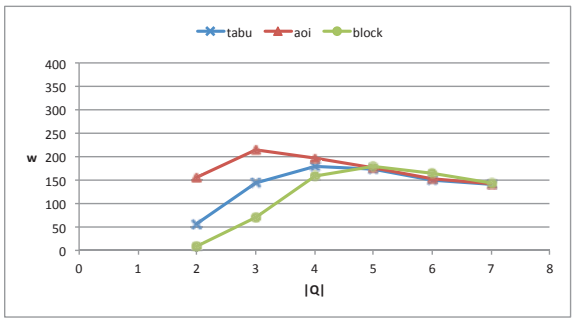

Figure 5: Utility function u_w of the $|Q|$ colluders in a dispenser competition.

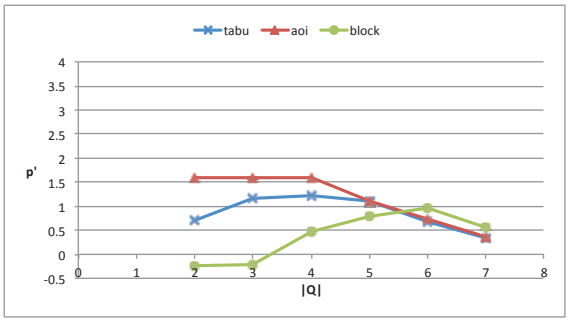

Figure 6: Utility function u'_p of the $|Q|$ colluders in a dispenser competition.

- correlation = $\left|\text{pearsonCorrelation}_{s \in S}(\ell_{s,r})\right|$

The metrics union, intersection and distance measure how dense or sparse group the players form. For example, if the mean of the intersection of the player positions is small, it can indicate that the players are moving together. The metrics direction and angle measure how uniformly the players move (e.g., they are following the same targets). The metrics meanX and meanY measure whether the players prefer staying in a certain area of the game world. The metric correlation measures whether the players are moving in a straight line in relation to one another.

4. RESULTS AND ANALYSIS

We have analysed two different game types with 4 colluders: evenly distributed pills and dispenser competition. First, we computed the values of the utility functions u_p, u_w, and u'_p in the both settings. Graphs for the utility functions can be found in Figures 1–6.

There is a notable fluctuation in the values of u_p and u'_p when we have AOI colluders in an evenly distributed setting (see Figure 1 and Figure 3). This is due to the suboptimal partitioning of the playing area (i.e., depending on the number of colluders the game world is divided into rectangles or horizontal slices). In the dispenser competition this effect is not visible.

The optimal number of colluders is influenced by the selected utility function. Functions u_w and u'_p have shapes like the theoretical pay-off function introduced in [10]. Also the maximum value is reached with a lower number of colluders

Table 1: Information gains in dispenser setting with 4 colluders.

Tabu colluder		AOI colluder		Blocking colluder	
gain	metric	gain	metric	gain	metric
0.10802	hitsMutualDelta	0.10802	hitsMutual	0.10802	hitsAllDelta
0.10802	hitsAllDelta	0.10802	hitsAllDelta	0.10802	hitsMutual
0.10802	hitsMutual	0.10802	hitsMutualDelta	0.10802	hitsMutualDelta
0.07165	deathsTotalDelta	0.10802	correlationStdDev	0.03008	union
0.05496	hitsAll	0.10802	correlation	0.02992	intersection
0.04775	killsTotalDelta	0.10453	intersection	0.02801	distance
0.04235	deathsTotal	0.10802	union	0.01271	killsTotalDelta
0.02393	distance	0.10415	distance	0.01161	deathsTotal
0.02358	angle	0.10298	meanYStdDev	0.01033	intersectionStdDev
0.02201	directions	0.10261	meanXStdDev	0.00987	unionStdDev
0.02139	intersection	0.10261	hitsAll	0.00944	correlation
0.02116	union	0.08038	correlationMax	0.00833	directions
0.0194	killsTotal	0.06647	killsTotalDelta	0.00735	angleStdDev
0.01639	scoreTotal	0.06508	deathsTotalDelta	0.00691	totalIntersetion
0.01297	angleStdDev	0.03991	distanceStdDev	0.00612	deathsTotalDelta
0.00879	correlation	0.03411	deathsTotal	0.00501	hitsAll
0.00687	directionsStdDev	0.02992	meanY	0.00499	directionsStdDev
0.00647	intersectionStdDev	0.02827	killsTotal	0.00406	intersectionMax
0.00605	unionStdDev	0.02802	meanYMax	0.00405	unionMax
0.00508	correlationStdDev	0.02766	meanX	0.00386	scoreTotal
0.00343	meanXMax	0.02536	meanXMax	0.00379	angle
0.00337	totalIntersetion	0.02479	meanXMin	0.00313	correlationStdDev
0.0027	meanYMax	0.0233	meanYMin	0.00252	distanceMax
0.00255	intersectionMax	0.01842	scoreTotal	0	scoreTotalDelta
0.00249	unionMax	0.01414	unionMax	0	distanceMin
0.00245	meanYStdDev	0.0141	intersectionMax	0	distanceStdDev
0.00237	directionsMin	0.01117	totalIntersetion	0	angleMax
0	distanceStdDev	0.01003	distanceMax	0	correlationMax
0	directionsMax	0.00392	intersectionMin	0	angleMin
0	angleMax	0.00389	correlationMin	0	directionsMax
0	angleMin	0.00381	unionStdDev	0	correlationMin
0	meanYMin	0.003	intersectionStdDev	0	directionsMin
0	correlationMin	0.00158	totalUnion	0	meanXMin
0	correlationMax	0	directions	0	intersectionMin
0	meanXStdDev	0	angleMax	0	meanXMax
0	totalUnion	0	angleMin	0	killsTotal
0	scoreTotalDelta	0	angle	0	meanXStdDev
0	meanX	0	angleStdDev	0	meanX
0	intersectionMin	0	scoreTotalDelta	0	unionMin
0	distanceMax	0	unionMin	0	meanYMax
0	distanceMin	0	distanceMin	0	meanYMin
0	unionMin	0	directionsStdDev	0	meanY
0	meanY	0	directionsMax	0	totalUnion
0	meanXMin	0	directionsMin	0	meanYStdDev
0.75422	total	1.72233	total	0.52615	total

Table 2: Information gains in even distribution setting with 4 colluders.

Tabu colluder		AOI colluder		Blocking colluder	
gain	metric	gain	metric	gain	metric
0.07742	hitsMutual	0.0833	correlation	0.05734	hitsMutual
0.07316	hitsMutualDelta	0.07263	scoreTotal	0.05342	hitsAllDelta
0.07316	hitsAllDelta	0.06956	hitsMutual	0.05342	hitsMutualDelta
0.02384	scoreTotal	0.06458	hitsMutualDelta	0.01379	scoreTotal
0.02017	deathsTotal	0.06458	hitsAllDelta	0.00499	union
0.0147	hitsAll	0.06111	correlationStdDev	0.00496	distance
0.01152	deathsTotalDelta	0.05669	union	0.00483	intersection
0.00774	angle	0.05662	intersection	0.0036	intersectionMax
0.0075	killsTotal	0.04769	distance	0.0036	unionMax
0.00647	directions	0.04449	scoreTotalDelta	0.00314	deathsTotal
0.00372	angleStdDev	0.02834	correlationMax	0.00279	intersectionStdDev
0.00361	meanXStdDev	0.02509	meanXStdDev	0.00258	unionStdDev
0.0035	unionStdDev	0.02507	meanYStdDev	0.00258	distanceMax
0.00334	meanYStdDev	0.02504	hitsAll	0	angleStdDev
0.00331	intersectionStdDev	0.02489	meanX	0	totalUnion
0.00323	directionsStdDev	0.02484	meanY	0	directionsStdDev
0.00306	meanYMax	0.01765	meanXMax	0	angle
0.00283	totalIntersetion	0.01741	meanYMin	0	directionsMin
0.00246	meanXMin	0.01741	meanYMax	0	unionMin
0.00244	union	0.01525	unionMax	0	distanceMin
0.00244	intersection	0.01524	intersectionMax	0	directions
0.00155	killsTotalDelta	0.01493	meanXMin	0	angleMax
0	directionsMax	0.01353	deathsTotal	0	angleMin
0	directionsMin	0.01319	totalIntersetion	0	distanceStdDev
0	distanceMin	0.0123	killsTotal	0	directionsMax
0	distance	0.00913	deathsTotalDelta	0	meanYStdDev
0	distanceStdDev	0.00877	distanceMax	0	meanYMin
0	angleMin	0.00801	intersectionStdDev	0	meanXMax
0	correlationMax	0.00776	unionStdDev	0	meanY
0	correlationMin	0.00645	correlationMin	0	correlationMin
0	correlation	0.00344	totalUnion	0	correlationMax
0	angleMax	0.00319	killsTotalDelta	0	meanYMax
0	correlationStdDev	0.00234	distanceStdDev	0	correlationStdDev
0	intersectionMin	0	directions	0	correlation
0	meanY	0	angleMax	0	scoreTotalDelta
0	meanX	0	angleStdDev	0	deathsTotalDelta
0	intersectionMax	0	angleMin	0	intersectionMin
0	meanXMax	0	angle	0	hitsAll
0	meanYMin	0	distanceMin	0	totalIntersetion
0	distanceMax	0	intersectionMin	0	meanXStdDev
0	scoreTotalDelta	0	unionMin	0	meanXMin
0	totalUnion	0	directionsStdDev	0	killsTotal
0	unionMin	0	directionsMax	0	meanX
0	unionMax	0	directionsMin	0	killsTotalDelta
0.35117	total	0.96052	total	0.21104	total

in u_w than in u'_p.

All utility functions have higher values in the evenly distributed setting than in the dispenser competition. All of the implemented collusion methods work better in such a preset game world. One reason for this is that the colluders share a common tabu list of visited locations. In a preset game world, the same list is useful through the entire game, but in a regenerating world the list needs to be reset whenever a new pill appears and the benefits of sharing the list have a shorter term.

Information gain means entropy before a change subtracted by the entropy after the change (i.e., how much entropy has reduced because of the change). It can be used to rank the attributes in the data sets: the larger the gain, the more information the attribute has about collusion. The information gains for different attributes can be found in Table 1 and Table 2. The information of the class attribute is 0.10820, which depends on the ratio of the colluders and non-colluders. From this follows that if an attribute has a gain of 0.10820, we can recognize collusion from that attribute alone.

In the dispenser setting, all colluders can be identified by mutual hits. Additionally, AOI colluders can be identified by the correlation of the players' locations. This is expected because the colluders engage in a very clear soft play by deliberately avoiding shooting at each other. Other shooting-related attributes also have a relatively high information gain.

In the even distribution setting, no single attribute can be used to detect collusion directly. However, the colluders can be identified based on a combination of attributes, but if players do not engage in soft play, they can no longer be identified reliably. Interestingly, collusion is more difficult to detect in the even distribution setting than in the dispenser setting, because of the higher collusion pay-off.

Apart from the hit-related functions, the metrics related to movement (e.g., angle and directions) have relatively high gains for tabu colluders. For AOI and blocking colluders, the location-based functions (e.g., union, intersection and distance) have higher gains. This is because tabu colluders move towards a common undiscovered area of the game world, while the non-colluding players move more randomly. AOI colluders try to stay evenly distributed in the game world, which we can observe in the functions. For the blocking colluder, the AOI-like behaviour can be caused when blockers follow non-colluders, which reside all around the game world, instead of getting towards the remaining pills. That also explains why blocking colluders have no gain with meanX and meanY functions.

Even if a player performs better than expected, it is not necessarily the best indicator of collusion. A player can be – fairly – better than the others. This is reflected in the scoreTotal metric, which does not provide a significant gain in the dispenser setting.

Our preliminary experiments indicate that if the gain falls below 0.03, we can omit the attribute value. That would leave us a manageable set of attributes to observe: In the dispenser setting, we can use 7 attribute values for tabu colluders, 16 for AOI colluders, and 4 for blocking colluders. In the even distribution setting, we can use 17 attribute values for tabu colluders, 10 for AOI colluders, and 3 for blocking colluders. As further study, we will analyse whether the limit of 0.03 could be even higher.

The following functions do not have any gain in any setting: angleMin, angleMax, unionMin, distanceMin and directionsMax. In general minimum and maximum aggregations have no (or very low) gain. Most of the functions in the test data have natural upper or lower bounds, which are achieved during the game regardless of the playing style.

5. CONCLUSION

Our experiments indicate that it is possible to elicit features that indicate collusion. Even if the detection is not totally reliable, we can use it in multi-phase detection, where suspiciously acting players are first recognized by a lightweight method. The set of suspicious players is then scrutinized by more rigorous (and time-consuming) methods such as game theory. For example, when we are observing behaviour during the game, we could abstract it to see whether a group of players follows the rational way to play (e.g., if a subset of players reaches a Nash equilibrium, the reason could be collusion).

In this paper, we used a simple two-dimensional game world. Our intention is to validate the results by conducting experiments with human players in similar settings. In the future work, we want to also generalize the results to other game types that offer more freedom of choice to the players. Also, we will improve the decision-making of the players and – especially – the colluders to make them even craftier to win the game.

6. REFERENCES

[1] A. Ercole, K. D. Whittlestone, D. G. Melvin, and J. Rashbass. Collusion detection in multiple choice examinations. *Medical Education*, 36(2):166–172, 2002.

[2] F. Glover. Tabu search – part I. *ORSA Journal of Computing*, 1(3):190–206, 1989.

[3] U. Johansson, C. Sönströd, and R. König. Cheating by sharing information—the doom of online poker? In L. W. Sing, W. H. Man, and W. Wai, editors, *Proceedings of the 2nd International Conference on Application and Development of Computer Games*, pages 16–22, Hong Kong SAR, China, Jan. 2003.

[4] S. J. Murdoch and P. Zieliński. Covert channels for collusion in online computer games. In J. Fridrich, editor, *Information Hiding: 6th International Workshop*, volume 3200 of *Lecture Notes in Computer Science*, pages 355–369, Toronto, Canada, May 2004. Springer-Verlag.

[5] Namco. *Pac-Man*. Namco, 1979.

[6] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, third edition, 2009.

[7] J. Smed and H. Hakonen. Synthetic players: A quest for artificial intelligence in computer games. *Human IT*, 7(3):57–77, 2005.

[8] J. Smed and H. Hakonen. *Algorithms and Networking for Computer Games*. John Wiley & Sons, Chichester, UK, 2006.

[9] J. Smed, T. Knuutila, and H. Hakonen. Can we prevent collusion in multiplayer online games? In T. Honkela, T. Raiko, J. Kortela, and H. Valpola, editors, *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)*, pages 168–175, Espoo, Finland, Oct. 2006.

[10] J. Smed, T. Knuutila, and H. Hakonen. Towards swift and accurate collusion detection. In M. Roccetti, editor, *8th International Conference on Intelligent Games and Simulation (Game-On 2007)*, pages 103–107, Bologna, Italy, Nov. 2007.

[11] C. Vallvè-Guionnet. Finding colluders in card games. In H. Selvaraj and P. K. Srimani, editors, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume II, pages 774–775, Las Vegas, NV, USA, Apr. 2005.

[12] R. V. Yampolskiy. Online poker security: Problems and solutions. In P. Fishwick and B. Lok, editors, *GAME-ON-NA 2007: 3rd International North American Conference on Intelligent Games and Simulation*, Gainesville, FL, USA, Sept. 2007.

[13] J. Yan. Security design in online games. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03)*, pages 286–297, Las Vegas, NV, USA, Dec. 2003.

[14] S. Zander, G. Armitage, and P. Branch. Covert channels in multiplayer first person shooter online games. In *33rd Annual IEEE Conference on Local Computer Networks – LCN 2008*, pages 215–222, Montreal, Canada, Oct. 2008.