Representation and Reasoning about General Solid Rectangles

Xiaoyu Ge and Jochen Renz
Research School of Computer Science
The Australian National University
Canberra, Australia
{xiaoyu.ge, jochen.renz}@anu.edu.au

Abstract
Entities in two-dimensional space are often approximated using rectangles that are parallel to the two axes that define the space, so-called minimum-bounding rectangles (MBRs). MBRs are popular in Computer Vision and other areas as they are easy to obtain and easy to represent. In the area of Qualitative Spatial Reasoning, many different spatial representations are based on MBRs.

Surprisingly, there has been no such representation proposed for general rectangles, i.e., rectangles that can have any angle, nor for general solid rectangles (GSRs) that cannot penetrate each other. GSRs are often used in computer graphics and computer games, such as Angry Birds, where they form the building blocks of more complicated structures. In order to represent and reason about these structures, we need a spatial representation that allows us to use GSRs as the basic spatial entities.

In this paper we develop and analyze a qualitative spatial representation for GSRs. We apply our representation and the corresponding reasoning methods to solve a very interesting practical problem: Assuming we want to detect GSRs in computer games, but computer vision can only detect MBRs. How can we infer the GSRs from the given MBRs? We evaluate our solution and test its usefulness in a real gaming scenario.

1 Introduction and Problem Description
In many real-world problems and tasks we are dealing with entities embedded in two- or three-dimensional space. Their shape, location, and other spatial properties are often important factors in finding a solution to these problems. Representing spatial information and methods and techniques for processing this information are therefore important components of many problems and their solutions. A typical way of representing spatial information is to have a coordinate system and to specify spatial properties of entities in terms of their coordinates. However, when the exact coordinates are not available or not important, we can also use a qualitative spatial representation that makes only those distinctions that are important for a problem [Cohn and Renz, 2008]. For example, when following navigation instructions it is usually sufficient to know which landmarks are to the left or right, in the front or behind the car, but not their exact location. A major advantage of making only the important distinctions is that it can reduce the search space when solving a problem. Instead of searching over all possible coordinates, we only need to search over the different qualitative distinctions.

This paper is motivated by a practical problem that benefits from a qualitative spatial representation in order to reduce the search space for follow-up problem solving. The problem is part of the Angry Birds AI challenge [AI12, 2012; AIBirds, 2013], where the task is to build an AI agent that can autonomously and successfully play the Angry Birds game [Rovio, 2013]. In the game pigs are sheltered by a structure consisting of blocks of different sizes, colours and shapes that roughly behave according to the laws of physics. The player needs to kill the pigs by shooting birds and hitting the pigs either with the birds directly or indirectly with falling blocks. Identifying good shots clearly benefits from an analysis of the sheltering structure.

The different objects in the game are detected using a provided computer vision system that places a minimal bounding rectangle (MBR) that is parallel to the game frame around each detected object and classifies the object type (see Figure 1). MBRs are used as they can be detected fast and in a robust way [Caldwell, 2005]. Some methods such as qualitative physics [Rajagopalan and Kuipers, 1994; Forbus et al., 1991], video analysis [Gupta et al., 2010; Siskind, 2003] require precise numerical calculations to solve similar problems. However, trying to detect the actual shapes of the objects is very unreliable and error-prone, as can be seen in this video [Robertson, 2012] where the exact shapes are used. As most of the objects in the sheltering structure are rectangular blocks of different angles called general solid rectangles (GSR), we would like to obtain the actual blocks rather than their MBRs. This should enable a much more accurate analysis of the sheltering structure than using MBRs.

We call the problem we need to solve QualGSR(Θ,R): given a set Θ of MBRs, identify for each MBR an actual contained GSR such that the GSRs do not overlap and form a stable structure under gravity. In order to reduce the search space for the analysis of the sheltering structure, we use a qualitative representation R of the GSRs that only distinguishes important features of GSRs.

This is a difficult problem for a number of reasons. Firstly, there is no existing qualitative representation of angled rect-
angles, only of rectangles parallel to the axes [Balbiani et al., 1998]. Secondly, the given MBRs are fixed and we know their actual coordinates, i.e., the contained GSR is restricted to a given box. This is different from any qualitative representation and any qualitative reasoning methods previously developed [Gerevini and Renz, 2002; Gottfried, 2003]. We need to combine quantitative with qualitative information in order to solve this. Thirdly, computing stability under gravity is impossible [Forbus et al., 1987]. Qualitative stability can therefore only be an approximation.

In the following we give a brief introduction to qualitative spatial representation and reasoning (QSR). We present GR-n, a qualitative representation of GSRs that can be adjusted according to the number of required distinctions. We then develop a method for solving the QualGSR(Θ, GSR-10) problem and evaluate it using real Angry Birds instances.

![Figure 1: (a) a typical Angry Birds scenario, (b) the corresponding set of MBRs](image)

2 Qualitative Spatial Representation and Reasoning (QSR) about Rectangles

A qualitative spatial representation typically takes a domain $\mathcal{D}$ of entities of a two- or three-dimensional space (for example points, line segments, or extended regions) and defines a set of binary relations $\mathcal{R} = \{R_1, \ldots, R_n\}$ where each $R_i \subseteq \mathcal{D} \times \mathcal{D}$ represents a meaningful distinctions of how these entities can be related to each other. It is common to define relations that are jointly exhaustive and pairwise disjoint (JEPD), as between any two entities exactly one such relation holds. JEPD relations are also called atomic or basic relations. Two well-known examples are RCC-8 [Randell et al., 1992] that distinguishes eight topological relations between extended regions, or the Cardinal Direction relations [Ligozat, 1998] that distinguish 8 different direction relations between points in a plane. Knowledge about the relation between two entities $x, y$ can be represented as a constraint $x R y$ where $R \in 2^{\mathcal{R}}$. This allows us to specify indefinite knowledge using a union of atomic relations that might hold.

Reasoning about these relations can be done by using weak composition ($\circ$) [Düntsch et al., 2001]. $R \circ S$, for any $R, S \in 2^{\mathcal{R}}$, is defined as the union of all atomic relations that can hold between two entities $x$ and $y$ given that $x R z$ and $z S y$ hold for some entity $z$. Weak composition between atomic relations is usually pre-computed and stored in a composition table. In combination with the operators intersection and converse of relations, weak composition forms a powerful reasoning method that allows us to infer unknown relations, remove impossible relations, or determine consistency of given knowledge. More details on QSR can be found in [Cohn and Renz, 2008].

Rectangles form an important domain in QSR research. Rectangles are easy to represent and easy to obtain, for example using computer vision [Stollnitz et al., 1996; Chaudhuri and Samal, 2007]. Minimum bounding boxes are commonly used to approximate objects [Nakagawa and Rosenfeld, 1979]. In all cases we are aware of, researchers were using rectangles or MBRs that are parallel to the axes defining the space. Probably the best known QSR approach for rectangles is the Rectangle Algebra (RA) [Balbiani et al., 1998; 1999]. It is a simple extension of the Interval Algebra (IA) [Allen, 1983] to two-dimensional space. The IA distinguishes thirteen different binary relations between one-piece intervals on a directed line, typically interpreted as the time line. These atomic relations are defined by the 13 possible configurations of the start and end points of the two intervals, leading to the relations before (b), meets (m), overlaps (o), starts (s), during (d), equal (eq), finishes (f) and their converse relations (bi), (mi), (oi), (si), (di), (fi). When taking two rectangles (or any two one-piece objects) $A, B$ in two-dimensional space, we can use the projection of the two objects to the $x$ and the $y$ axes and obtain two intervals on each axis, one for each object. We can now represent the relationship between the two objects as a pair $R = (R_x, R_y)$ where $R_x$ represents the IA relation on the $x$-axis and $R_y$ the IA relation on the $y$-axis. These $13 \times 13 = 169$ relations form the atomic relations of the Rectangle Algebra. These relations allow us to represent information about the topology and direction of rectangles. Other approaches using rectangles are for example the Cardinal Direction Calculus (CDC) [Goyal and Egenhofer, 2001] or CORE-9 [Cohn et al., 2012; Sadeghi Sokeh et al., 2013].

While such approaches are useful for representing spatial relations between MBRs, they are only an approximation of GSRs and do not help us in representing or reasoning about GSRs. Moreover, in the problems we consider, all MBRs are given, i.e., all atomic relations between MBRs are known and the information is necessarily consistent, so no further reasoning is possible using typical QSR approaches. In the next section we define useful qualitative relations for GSR, both unary relations and binary relations.

3 Representing General Solid Rectangles

Dealing with general rectangles (GR), i.e., rectangles with any angle, is much harder than dealing with standard MBRs. This becomes obvious when looking at the domain of GRs and how they relate to the domain of MBRs: For each MBR there are infinitely many GRs. Given an MBR $M$, we can obtain a GR $m$ that is bounded by $M$, i.e., $M$ is the MBR of $m$ in the following way: 1. Take the centre point $p$ of $M$ and draw a circle $c$ around $p$ that either intersects all four edges or all four corners of $M$. We call $c$ the auxiliary circle. Clearly there are infinitely many auxiliary circles for any given $M$. 2. We can now pick one of the two intersecting points at each edge and have to pick the diagonally opposite point of the opposite edge. Connecting these four points gives us a general rectangle, that is there are four different GRs for any $M$ and for any $c$ (see Fig.2(a)). If $c$ intersects the four corners, there is only one GR, the MBR itself. Throughout the paper we use the lower case $m$ to denote an arbitrary GSR and upper case $M$ to represent the MBR of $m$. We call a GR that is equiva-
lent to its MBR a regular rectangle and one that is different from its MBR an angular rectangle.

### 3.1 Unary Relations of General Rectangles

Our first task is to distinguish meaningful classes of general rectangles, that is, we decide which GRs can be clustered together and form an equivalence class with respect to these distinctions. Since this depends on the particular application for which we use GRs, we will define adjustable distinctions.

Some natural distinctions for GRs are whether they are leaning to the left or leaning to the right. This is particularly important for the Angry Birds case where we need to infer leaning to the left or leaning to the right. This is particularly important for the Angry Birds case where we need to infer leaning to the left or leaning to the right. This is particularly important for the Angry Birds case where we need to infer leaning to the left or leaning to the right. This is particularly important for the Angry Birds case where we need to infer leaning to the left or leaning to the right. This is particularly important for the Angry Birds case where we need to infer leaning to the left or leaning to the right.

Some other interesting distinctions are whether a rectangle is “slim” or “fat”, that is, we decide which GRs can be clustered together and form an equivalence class with respect to these distinctions. Since this depends on the particular application for which we use GRs, we will define adjustable distinctions.

Our first task is to distinguish meaningful classes of general rectangles, that is, we decide which GRs can be clustered together and form an equivalence class with respect to these distinctions. Since this depends on the particular application for which we use GRs, we will define adjustable distinctions. Some natural distinctions for GRs are whether they are

---

**Proposition 1.** $(A_1, A_1)$ is not a valid contact relation.

**Proof.** Assume the relation $m_1(A_1, A_1) m_2$ holds, i.e., $m_1 A_1$ and $m_2 A_1$ share the same coordinate. $m_1 A_2$ and $m_1 A_3$ form a 90 degrees angle and so do $m_2 A_2$ and $m_2 A_3$. Since $m_1$ and $m_2$ do not overlap, but share their top-most
point \( A_1 \), the angle between \( m_1, A_3 \) and \( m_2, A_2 \) and the angle between \( m_2, A_8 \) and \( m_1, A_2 \) must add up to 180 degrees, and therefore each angle is less than 180 degrees. This means that \( m_1, A_1 \) or \( m_3, A_1 \) cannot be the top-most point of its rectangle, which contradicts the definition of \( A_1 \).

**Proposition 2.** \((A_1, A_2)\) is not a valid contact relation.

**Proof.** We can reduce this to the \((A_1, A_1)\) case. Assume \( m_1, A_1 \) touches \( m_2, A_2 \) at point \( p \). We can now generate a GSR \( m_1 \) with the three corner points \( p, m_2, A_3, m_2, A_5 \) which is a sub-rectangle of \( m_2 \). However, \( m_3 \) and \( m_1 \) are in relation \((A_1, A_1)\) which is invalid and contradicts our assumption.

We can use similar proofs for the remaining invalid cases, while we can give example GSRs for the valid contact relations. In total there are 73 valid contact relations: \( |(A_1, A_1)| + |(A_1, R_6)| \times 2 + |(R_6, R_6)| + |(∅, 0)| = 32 + 16 \times 2 + 8 + 1 = 73 \).

**Theorem 3.** There are 73 atomic contact relations for GSRs.

### 3.3 The GSR-n Algebra

We can finally define our new algebra for general solid rectangles. It consists of 73 atomic contact relations, plus \( n = (4k - 1)^2 + 1 \) unary relations. Since only the number of unary relation varies, this is what determines the granularity of the algebra. For \( k = 1 \) we get 10 different categories of GSRs and hence call the algebra GSR-10.

### 4 Solving the QualGSR problem

We now use the algebra defined above to solve the QualGSR(\( Θ, \text{GSR-10} \)) problem, for a given set of MBRs \( Θ \). This problem occurs, for example, when developing an Angry Birds AI agent. The computer vision system provided for the Angry Birds AI competition [AI12, 2012; ABirds, 2013] can only detect MBRs, while the actual objects are GSRs. Our task is to find for each MBR \( M \in Θ \) a contained GSR \( m \) such that (1) the GSRs do not overlap, and (2) the GSRs are stable under downward gravity. Strictly speaking, \( M \) must be an eMBR, which is a fixed MBR with known coordinates (see Section 3.1). Stability depends on a number of factors such as friction or mass [Blum et al., 1970], and we use the stability conditions that we observed in the Angry Birds game (which are slightly different from real physics). We observed four cases when a GSR is stable and remains static, which depend on how it is supported. The four kinds of support are **Edge-Corner**, **Double-Edge**, **Double-Corner**, **One-Edge**. Figure 4 shows some examples for each of the supports. They are named by the type of contact sectors which support the GSR. E.g. **Edge-Corner** means the GSR is supported through one of its edges and one corner.

![Figure 4: Different stable support configurations](image)

### 4.1 Approximating Stability using GSR-10

We now illustrate how the stable configurations can be expressed qualitatively using GR-10 relations. We use **Edge-Corner** support for GSRs as an example. Under downwards gravity, eMBRs can be classified into three groups.

1. **\( M^{AL} \)** contains all the eMBRs whose GSRs will fall to the left if there is only one support at \( A_5 \). Specifically, \( M^{AL} = \{M^{(3,3)}, M^{(1,3)}, M^{(2,3)}, M^{(3,2)}\} \).
2. **\( M^{AR} \)** contains all the eMBRs whose GSRs will fall to the right if there is only one support at \( A_5 \). Specifically, \( M^{AR} = \{M^{(1,1)}, M^{(3,1)}, M^{(2,1)}, M^{(1,2)}\} \).
3. **\( M^{AN} \)** contains all the eMBRs whose GSRs is stable if there is only one support at \( A_5 \). \( M^{AN} = M^{(2,2)} \)

There are several sub-configurations of the Edge-Corner support. For a binary relation written in the form of \((A_i, ∗)\), the asterisk refers to any sector. The following are five conditions, satisfying any of them implies the existence of the Edge-Corner support:

1. \( ∃ M^{1AL}_1 ∃ M^{2}_2 : M^{1AL}_1(A_4, ∗)M^{2}_2 ∨ ∃ M^{3}_3 \) \( M^{1AL}_1(A_5, ∗)M^{3}_3 \)
2. \( ∃ M^{1AL}_1 ∃ M^{2}_2 : M^{1AL}_1(A_3, ∗)M^{2}_2 ∨ ∃ M^{3}_3 \) \( M^{1AL}_1(A_6, ∗)M^{3}_3 \)
3. \( ∃ M^{1AR}_1 ∃ M^{2}_2 : M^{1AR}_1(A_5, ∗)M^{2}_2 ∨ ∃ M^{3}_3 \) \( M^{1AR}_1(A_6, ∗)M^{3}_3 \)
4. \( ∃ M^{1AR}_1 ∃ M^{2}_2 : M^{1AR}_1(A_4, ∗)M^{2}_2 ∨ ∃ M^{3}_3 \) \( M^{1AR}_1(A_7, ∗)M^{3}_3 \)
5. \( ∃ M^{1AN}_1, ∃ M^{2}_2 : M^{1AN}_1(A_5, ∗)M^{2}_2 \)

Rule 1 and 2 (Figure 5.a) state that an angular rectangle that tends to fall to the left has **Edge-Corner** support if it has at least one contact in \( A_4 \) and one contact in \( A_5 \), or \( A_3 \) and \( A_6 \). Rule 3 and 4 (Figure 5.b) state that an angular rectangle that tends to fall to the right has **Edge-Corner** support if it has at least one contact in \( A_5 \) and one in \( A_6 \), or \( A_4 \) and \( A_7 \).

![Figure 5: (a.1-2) Edge-Corner Support Cases for \( M^{1AL}_1 \), (b.1-2) Edge-Corner Support Cases for \( M^{1AR}_1 \)](image)

### 4.2 Locally Consistent Instantiations

In the previous section, some rules were given to verify the stability of GSRs under gravity. The rules are binary constraints between eMBRs. We now show an approach of evaluating such binary constraints. Specifically, given two eMBR instances and a binary relation, we need to determine whether the two instances can satisfy the binary relation. We introduce the Min-Max Procedure to determine the possible binary relations between an arbitrary pair of eMBR instances.

**Critical Instantiations**

Given an eMBR instance, there is an infinite number of possible GSRs. Among those are two critical instantiations:
The **maximum instantiation** of an eMBR is the GSR with maximum size. The **minimum instantiation** of an eMBR is the GSR with minimum size. When the GSR is a regular rectangle, both critical instantiations are the same.

The two critical instantiations can be obtained with the help of the auxiliary circle \( c \) (see section 3). Firstly, we need to find out the auxiliary circle with the minimum radius that intersects the four edges of the eMBR. Now every corner of a valid GSR must be located between a corner of the eMBR and the point where \( c \) intersects. This not only limits the possible unary relations of a GSR, but also limits the critical instantiations. The maximum instantiation is an area connecting either four intersections according the QCI1 of the eMBR or the four corners. For example, if the eMBR is \( M^{(1,1)} \), then the maximum instantiation can be outlined by the two bottom left and the two up right intersections (see Figure 6). The minimum instantiation is a diagonal from the bottom left to the top right.

![Figure 6: Possible instantiations of eMBR \( M^{(1,1)} \): (left) intermediate, (middle) maximum, (right) minimum](image)

**Maximal and Minimal Effort**

Because the instantiations of an eMBR instance can vary from a line segment to a large rectangle and the contact between two arbitrary eMBRs depends on the actual instantiations, it can be observed that whether the two eMBRs can contact depends on the efforts made by the eMBRs on their corresponding instantiations. If both eMBRs make minimum efforts, then the instantiations are likely to be two general rectangles that are disjoint from each other while if both try their best effort to contact each other, then the instantiations might be two general rectangles that are intersecting.

Let \( M^{(i,j)} \), \( i, j \in \{1..3\} \) be an arbitrary eMBR instance and \( CS_1 \in \{\emptyset, R_1 \ldots R_8, A_1 \ldots A_8\} \) be a contact sector of \( M^{(i,j)} \). \( E^{\max}(M^{(i,j)}, CS_1) \) denotes an area where \( M^{(i,j)} \) makes maximal effort to contact any other GSRs with \( CS_1 \). If the \( E^{\max} \) cannot contact any other GSRs, then there will be no instantiations of the \( M^{(i,j)} \) can achieve the contact. In the sequel, \( E^{\min}(M^{(i,j)}, CS_1) \) refers to the corresponding area where the minimum effort is made. If the \( E^{\min} \) can contact an arbitrary GSR, then all instantiations of the \( M^{(i,j)} \) can achieve the contact via the specified sector. The maximal and minimal effort of an eMBR instance depends on the critical instantiations of the eMBR.

**Min-Max Evaluation**

Given an arbitrary pair of eMBRs, to determine whether a specific binary relation holds between them, the QCI1 information together with their bounding rectangles should be taken into considerations. The Min-Max evaluation can perform this task effectively. For a binary constraint to hold, the following requirements should be satisfied:

1. the maximum efforts of the two eMBRs in the constraint must be intersecting. We call it **maximum-maximum case**, otherwise **maximum-minimum failure**
2. the minimum efforts of the two eMBRs in the constraint must be disjoint, it is the **minimum-minimum case**, otherwise **minimum-minimum failure**

The minimal-maximum failures indicates that no matter how the GSRs are instantiated over the two eMBRs, they will be always penetrating each other. In the sequel, the maximum-maximum failures indicates that the contact specified by the binary relation can never be achieved between the two eMBRs. The only way that may solve these failure is to change the unary relations of the eMBRs.

The following are two examples of the min-max evaluation on the constraint \( M^{(1,1)}_1(A_6, A_2)M^{(1,1)}_2 \) (see Figure 7) and the constraint \( M^{(1,1)}_1(A_6, R_3)M^{(1,1)}_2(\text{see Figure 8})

![Figure 7: left: minimum-minimum, middle: maximum-maximum, right: instantiations that satisfy the constraint](image)

![Figure 8: left: minimum-minimum, middle: maximum-maximum, right: instantiations that satisfy the constraint](image)

**4.3 Consistent Instantiations**

In the previous section, we have shown a local procedure, min-max, for inferring possible pair-wise contacts. To approximate consistency for the whole configuration, we can compute path-consistency, which is a standard method in qualitative reasoning. Figure 9.a shows an example where path-consistency can be checked by weak composition of GR-10 relations. The min-max algorithm returns \( M^{(1,1)}_1(A_6, A_2)M^{(1,1)}_2, M^{(1,1)}_1(A_6, A_2)M^{(1,1)}_3 \), and \( M^{(1,1)}_1(A_6, A_2)M^{(1,1)}_2 \) as one possibility. But this is clearly identified as inconsistent by the path-consistency algorithm because \( (A_6, A_2) \cap ((A_6, A_2) \circ (A_6, A_2)) = \emptyset \). Another possibility returned by min-max is path-consistent and also consistent, as can been seen in red in Figure 9.a. Figure 9.b shows a case where verifying path-consistency requires actual coordinates. In a pair-wise consistent configuration, the boxes with IDs 2, 3, 4 are instantiated as regular rectangles and the big rectangle is instantiated as a GSR that leans to the left, written as \( M^{(1,3)} \). To achieve a consistent scenario, we need to evaluate the heights of the MBRs to determine which subset of the regular rectangles can touch the angular rectangle at the same time. A detection of inconsistency indicates that the vision software makes mistakes because static Angry Birds scenarios should always be consistent.
4.4 A Method for Solving QualGSR

We propose a method that utilizes GR-10 and all the techniques we mentioned above to identify a stable set of GSRs for a given set of MBRs. Our method first randomly assigns a unique ID for each input MBR. Each MBR maintains a list of its neighbors. The neighbors of a MBR are all other MBRs that intersect or boundary touch the MBR. A MBR is initialized when it has been given a GSR-10 unary relation, which creates an eMBR instance (see Alg.1 line 22). Our method will backtrack by picking up the MBR with the lowest ID from the remaining uninitialized MBRs (see Alg.1 line 26), create an eMBR instance on it, and get possible contacts by performing the min-max check (see Alg.1 line 23) between the MBR and its initialized neighbors pair-wisely. Branch pruning (see Alg.1 line 26) happens when the algorithm detects that an eMBR cannot be stable when all its neighbors have been initialized. The algorithm will terminate when it finishes a branch where all the MBRs are initialized and stable under valid contacts. To approximate consistency, we used the aforementioned height evaluation and similar approximations (see Section 4.3). This method is exponential in the number of GSRs. But different heuristics can be used to improve the efficiency of our method. For example we can identify cases where a GSR must be an angular rectangle.

4.5 Evaluation in the Angry Birds Context

We implemented our method and applied it to the Angry Birds game where the current computer vision software used for the Angry Birds AI competition can only detect MBRs. Figure 1.a shows a typical Angry Birds level. Clearly, many objects in the level are GSRs. Figure 1.b illustrates a set of corresponding MBRs. In this example, it takes 0.854s (on a CORE i7 with 8GB RAM) for the algorithm to compute a stable scenario using GSRs and the result matches the real scenario. We obtained similar results (see Table.1) for other levels, that is, our method can be used to obtain an accurate representation of a game level that can then be used to analyse its structural properties.

5 Conclusion

In this paper we presented the first qualitative representation of angled rectangles that cannot penetrate each other, so called general solid rectangles (GSR). GSRs are frequently used in computer-generated images such as computer games. We used this representation to solve a practical problem that occurs when we use computer vision to extract GSRs from computer-generated images, but vision can only detect MBRs. We presented a method that can compute GSRs even under the condition that all obtained GSRs must be stable under gravity. This problem is particularly challenging as it requires us to combine qualitative and quantitative spatial information in a way that has not been done before. We tested our method for the popular Angry Birds game and showed that we can extract stable GSRs accurately and reasonably fast.

<table>
<thead>
<tr>
<th>Level</th>
<th>Total GSRs</th>
<th>Total Angular GSRs</th>
<th>Time(secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12</td>
<td>2</td>
<td>0.249</td>
</tr>
<tr>
<td>5*</td>
<td>22</td>
<td>16</td>
<td>0.854</td>
</tr>
<tr>
<td>6*</td>
<td>15</td>
<td>3</td>
<td>0.278</td>
</tr>
<tr>
<td>11*</td>
<td>17</td>
<td>4</td>
<td>0.317</td>
</tr>
<tr>
<td>13*</td>
<td>24</td>
<td>9</td>
<td>0.444</td>
</tr>
<tr>
<td>15*</td>
<td>27</td>
<td>6</td>
<td>0.404</td>
</tr>
<tr>
<td>16*</td>
<td>17</td>
<td>6</td>
<td>0.286</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>10</td>
<td>0.522</td>
</tr>
</tbody>
</table>

Table 1: Results on Poached Eggs levels [Rovio, 2013] (The * indicates intermediate scenarios after some shots)

Algorithm 1 The Stability Approximation algorithm

```
1: procedure STABILITY_APPROXIMATION(mbrs)
2:   candidates ← {}, id ← 0
3:   for mbr ∈ mbrs do
4:     id ← id + 1, add (mbr, id) to candidates
5:   end for
6:   SEARCH(candidates)
7:   end procedure
8:   procedure SEARCH(candidates)
9:     if all mbrs in candidates are initialized and stable
10:        then print the result, return true
11:        allCandidates ← REFINEN(candidates)
12:        for candidates ∈ allCandidates do
13:          if SEARCH(candidates) then return true
14:        end if
15:     end for
16:   end if
17:   end procedure
18:   procedure REFINEN(candidates)
19:     Set allCandidates, (mbr, id) ← candidates.pop()
20:     for unary in GR unary relations do
21:       contacts ← (neighbors, {})  embr ← (mbr, id, unary, contacts)
22:     end for
23:     for each neighbor, perform the min-max check with the embr to get the set of possible contacts. Perform consistency approximations. Generate all combinations of contacts by picking up one contact from each contact set. for contacts ∈ all combinations do
24:       apply contacts on embr and its neighbors
25:     if branch pruning happens then continue
26:     else copy the candidates list, in the new list update embr’s contacts and its neighbors’ accordingly, add the new list to allCandidates.
27:     end if
28:   end for
29:   end for
30: return allCandidates
31: end procedure
```
References


