Abstract

We present an important, general class of new games, called side-choosing games (SCGs), for "gamifying" problem solving in formal sciences. Applications of SCGs include (1) peer-grading in teaching to (2) studying the evolution of knowledge in formal sciences to (3) organizing algorithm competitions. We view SCGs as a new and general model for formulating formal problems that need to be solved using human computation. Our interest in this paper is on how to evaluate an SCG tournament fairly and effectively. The Collusion-Resistance Theorem is surprising: it tells us to be indifferent for SCG tournaments. The Collusion-Resistance Theorem for designing collusion-resistant evaluations prove the Collusion-Resistance Theorem as a general principle for designing collusion-resistant evaluations for SCG tournaments. The Collusion-Resistance Theorem is surprising: it tells us to be indifferent to wins but to count certain kinds of losses for scoring players and ranking them. If collusion is not an issue, we offer a family of useful ranking functions which are not collusion-resistant.

1 Introduction

A side-choosing game (SCG) \( H = (G, GS, Q, p_x, p_y) \) is based on an extensive form two-player game \( G \) between players \( p_x \) and \( p_y \) with perfect information and without ties, i.e., there is always a winner and a loser\(^1\). \( G \) is a game between two players, white and black. \( GS \) is a game state of \( G \) (i.e., a node of the game tree of \( G \)). \( Q \) is a proposition on the game state \( GS \) of the form: white or black has a winning strategy when white or black moves first.

The players \( p_x, p_y \) of an SCG have their preferred, static side (white or black), depending on whether they believe \( Q \) or \( \neg Q \) to be true. The players are free to choose their static side before the game. But during the game the players must have opposite "run-time" sides which we implement by making (per game) at most one of them the devil’s advocate (or forced).

The side-choosing game \( (G, GS, Q, p_x, p_y) \) produces a game result row consisting of (1) the winner \((p_x \text{ or } p_y)\) (2) the loser \((p_x \text{ or } p_y)\) and (3) at most one forced player \((p_x \text{ or } p_y \text{ or } 0, \text{ if none was forced})\). A set of game results produced by multiple binary SCGs is called an SCG-Table.

1.1 Examples

Consider the chess position \( GS \) in Fig. 1 as a side-choosing game\(^2\). The game \( G \) is chess, modified so that winning for white means to mate the black king in 2 moves. The proposition \( Q \) says that white starts in \( GS \) and wins. We have two players, Alice and Bob, who study \( G \) and \( GS \) and make their side choices. Alice believes she can win as white (she "sees" the mate in two) and therefore her side choice is white. Bob does not see the mate in two and therefore he wants to be black. The game is played and Alice wins (how is left as an exercise to the reader; there is only one optimal move for white.). Game result row= \( (\text{winner}="\text{Alice}" \text{, loser}="\text{Bob}" \text{, forced}=0) \). If white plays b3 and black f3 then white mates with Qb2. White wins but only because black made a mistake. The correct move for black is c4 (not f3) and white cannot mate in the next move. This example gives the wrong impression that playing one perfect game reveals the winning strategy (solution). But this is not the case most of the time.

Next we consider a large family of examples of side-choosing games: the family consists of semantic games (Kulas and Hintikka 1983) for claims with side choice added. The game \( G \) is defined by an interpreted logical sentence between white (proponent, existential quantifier) and black (opponent, universal quantifier). The outermost quantifier of the sentence determines who moves first. For white, the game is about making the sentence true by assigning values to variables. Black tries to prevent this. Side-choosing games exist for many different logics such as first-order predicate logic, higher-order logics and independence-friendly logic.

A first important subclass are claims in formal sciences to side choice added. The correct move for white means to mate the black king in 2 moves. The proposition \( Q \) says that white starts in \( GS \) and wins. We have two players, Alice and Bob, who study \( G \) and \( GS \) and make their side choices. Alice believes she can win as white (she "sees" the mate in two) and therefore her side choice is white. Bob does not see the mate in two and therefore he wants to be black. The game is played and Alice wins (how is left as an exercise to the reader; there is only one optimal move for white.). Game result row= \( (\text{winner}="\text{Alice}" \text{, loser}="\text{Bob}" \text{, forced}=0) \). If white plays b3 and black f3 then white mates with Qb2. White wins but only because black made a mistake. The correct move for black is c4 (not f3) and white cannot mate in the next move. This example gives the wrong impression that playing one perfect game reveals the winning strategy (solution). But this is not the case most of the time.

Next we consider a large family of examples of side-choosing games: the family consists of semantic games (Kulas and Hintikka 1983) for claims with side choice added. The game \( G \) is defined by an interpreted logical sentence between white (proponent, existential quantifier) and black (opponent, universal quantifier). The outermost quantifier of the sentence determines who moves first. For white, the game is about making the sentence true by assigning values to variables. Black tries to prevent this. Side-choosing games exist for many different logics such as first-order predicate logic, higher-order logics and independence-friendly logic.

A first important subclass are claims in formal sciences to side choice added. The correct move for white means to mate the black king in 2 moves. The proposition \( Q \) says that white starts in \( GS \) and wins. We have two players, Alice and Bob, who study \( G \) and \( GS \) and make their side choices. Alice believes she can win as white (she "sees" the mate in two) and therefore her side choice is white. Bob does not see the mate in two and therefore he wants to be black. The game is played and Alice wins (how is left as an exercise to the reader; there is only one optimal move for white.). Game result row= \( (\text{winner}="\text{Alice}" \text{, loser}="\text{Bob}" \text{, forced}=0) \). If white plays b3 and black f3 then white mates with Qb2. White wins but only because black made a mistake. The correct move for black is c4 (not f3) and white cannot mate in the next move. This example gives the wrong impression that playing one perfect game reveals the winning strategy (solution). But this is not the case most of the time.
petitions (a la TopCoder, see topcoder.com or Kaggle, see kaggle.com).

1.2 Motivation
Claims are ubiquitous in human reasoning. An SCG can be viewed as a plausibility check of a claim. A game state GS of an extensive form game G is a model of a claim. When I argue that a claim is true but then I cannot defend it using a plausibility check, there must be something wrong with my argument. We are interested in how to aggregate multiple such plausibility checks in a robust ranking manner, considering incentive and trust.

There are two kinds of incentives in SCG: the incentive (1) to be top-ranked which brings money or fame and (2) to get feedback during game play which builds skills and provides opportunity for learning. Incentive (2) suggests productive applications of SCGs in education.

Trust in the SCG approach is related to the belief that good work as a player will get rewarded and that it is not possible to be top-ranked without doing good work. There should be no sneaky ways to game the system: money or fame must be well deserved. Trust can be broken in at least two ways: (1) by defining games but not checking that all game rules are perfectly followed and (2) by having tournaments and evaluations where you can succeed without hard work as a player will get rewarded and that it is not control don’t affect your rank”. You are in control if you participate and are not forced.

The axioms are formalized by expressing that adding a row where player \( p_x \) satisfies a property, will keep the ranking of \( p_x \) with respect to other players \( p_y \), invariant.

Let \( P \) be the set of all players. \( R(P) \) is the set of possible game results for \( P \) (for formal definition see (Abdelmeged, Xu, and Lieberherr 2015)). For our theory we define a few basic predicates: \( \forall p_x \in P, \forall r \in R(P) \)

\[
\begin{align*}
\text{participant}(p_x, r) &\iff p_x \text{ is a participant in the game } r \\
\text{win}(p_x, r) &\iff p_x \text{ won the game } r \\
\text{loss}(p_x, r) &\iff p_x \text{ lost the game } r \\
\text{forced}(p_x, r) &\iff p_x \text{ is forced to choose a side in the game } r \\
\text{control}(p_x, r) &\iff \text{participant}(p_x, r) \land \neg \text{forced}(p_x, r) \\
\text{fault}(p_x, r) &\iff \text{loss}(p_x, r) \land \neg \text{forced}(p_x, r)
\end{align*}
\]

We also define counting functions for scoring players:

\[
\begin{align*}
wf_p(T) &= \text{the win count of } p_x \text{ in } T \text{ in a forced position} \\
wu_p(T) &= \text{the win count of } p_x \text{ in } T \text{ in an unforced position} \\
lp(T) &= \text{the loss count of } p_x \text{ in } T \text{ in a forced position} \\
lT(T) &= \text{the loss count of } p_x \text{ in } T \text{ in an unforced position}
\end{align*}
\]

It’s obvious that given table \( T = T \cup \{r\} \) and \( X \in \{wf, wu, lf, lu\} \) the following transitional relations hold:

\[
X_{p_x}(T') = \begin{cases} 
X_{p_x}(T) + 1 & \text{if } X \text{ happens in } \{r\} \\
X_{p_x}(T) & \text{otherwise}
\end{cases}
\]

The equation above is critical to transform predicate logic axioms into algebraic formulas presented later in this paper.

2.1 Ranking Axioms
We define a pre-order \( \preceq_U \) called the weakly better relation \( \forall T \subseteq G \) based on the scoring function \( U : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R} \). Lower \( U \) means better player. For convenience, we drop the subscript and refer to it simply as \( \preceq \).

We want to assign to each player a score solely based on the players’ demonstration of ability. We use the above four counting functions, based on wins and losses and whether a player was forced, to calculate a player’s score. We formally define the ranking relation as,

\[
\begin{align*}
\forall p_x, p_y \in P, \forall T \subseteq R(P), &\forall p_x \preceq_T p_y \iff \\
U(wf_p(T), wu_p(T), lu_p(T)) - U(wf_p(T), wu_p(T), lu_p(T)) \leq U(wf_p(T), wu_p(T), lu_p(T)) &\iff U(wf_p(T), wu_p(T), lu_p(T)) \leq U(wf_p(T), wu_p(T), lu_p(T)) \iff
\end{align*}
\]

We want the ranking relation to have the following properties defined in terms of table extensions:

- **NEW**: Winning cannot lower your rank:

\[
\forall p_x, p_y \in P, \forall T \subseteq R(P), \forall r \in \{r \mid r \in R(P) \land T \land win(p_x, r)\}
\[
[p_x \preceq_T p_y \Rightarrow p_x \preceq_T U(r) \preceq_T p_y] \tag{2}
\]

2. Main Theory
We discuss the ranking of players based on an SCG-table \( T \) under the axioms “winning cannot lower your rank” and “losing cannot increase your rank” and “games you don’t
• NPEL: Losing cannot increase your rank:
\[
\forall x, y \in P, \forall T \subseteq R(P), \forall r \in \{r | r \in R(P) \setminus T \land \text{loss}(y, r)\}, \quad [p_x \leq^T p_y \Rightarrow p_x \leq^{T \cup \{r\}} p_y] \tag{3}
\]

• CR: Games you don’t control don’t lower your rank.
\[
\forall x, y \in P, \forall T \subseteq R(P), \forall r \in \{r | r \in R(P) \setminus T \land \neg \text{control}(x, r)\}, \quad [p_x \leq^T p_y \Rightarrow p_x \leq^{T \cup \{r\}} p_y] \tag{4}
\]

2.2 Universal Domain
From equation 1, it is clear that for every logically possible game result table \(T\), we have a valid preorder. This implies that our ranking relation satisfies the Universal Domain property.

2.3 Anonymity
From equation 1 it is clear that the scoring function ignores the identity of the player in calculating the score. Hence, the ranking relation is indifferent on the parameter \(x\).

2.4 Monotonicity of \(U\) and Notations
As we score a player solely based on the player’s wins and losses, NNEW and NPEL imply that the function \(U\) is monotonic. One interesting property of the parameters of \(U\) for a particular player is that when we add a new game to the existing game result table \(T\), at most one parameter increments. This allows us to define the following notations:

\(\uparrow_x\): \(U\) is monotonically non-decreasing on the parameter \(x\)

\(\downarrow_x\): \(U\) is monotonically non-increasing on the parameter \(x\)

\(\uparrow\): \(U\) is indifferent on the parameter \(x\)

3 Properties of Ranking Relations
In this section, we reformulate the axioms as equivalent monotonicity constraints. It is easier to reason in the space of monotonicity constraints than in the space of predicate logic.

3.1 Collusion Resistance (CR)
Given \(T' = T \cup \{r\}\), we reformulate CR as follows:

\[
U(wf_x(T), wu_x(T), l_f_x(T), l_u_x(T)) \leq U(wf_x(T), wu_x(T), l_f_x(T), l_u_x(T)) \\
\Rightarrow U(wf_x(T'), wu_x(T'), l_f_x(T'), l_u_x(T')) \leq U(wf_x(T'), wu_x(T'), l_f_x(T'), l_u_x(T')) \tag{5}
\]

Considering the definition of "not in control", there are 2 cases to treat:

I. Game results where \(p_x \) did not participate. Then \(p_y\) may have won or lost in a forced or unforced position against some third player \(p_z\).

Let us consider the row \(r\) where \(p_y\) wins over \(p_z\) in a forced position, given \(T' = T \cup \{r\}\) we have,

\[
U(wf_x(T'), wu_x(T'), l_f_x(T'), l_u_x(T')) = U(wf_x(T), wu_x(T), l_f_x(T), l_u_x(T))
\]

From the CR constraint above, we have:
\[
U(wf_x(T'), wu_x(T'), l_f_x(T'), l_u_x(T')) \leq U(wf_x(T), wu_x(T), l_f_x(T), l_u_x(T)) \tag{6}
\]

From equations 1 and 6, we get the monotonicity constraint,

\[
U \uparrow_{wf} \tag{7}
\]

Similarly, let us consider the case \(r\) where \(p_y\) wins over \(p_z\) in an unforced position, given \(T' = T \cup \{r\}\) we have,

\[
U(wf_x(T), wu_x(T), l_f_x(T), l_u_x(T)) \leq U(wf_x(T), wu_x(T) + 1, l_f_x(T), l_u_x(T)) \tag{8}
\]

From equations 1 and 8, we get the monotonicity constraint,

\[
U \uparrow_{wu} \tag{9}
\]

Using a similar argument, for the case where \(p_y\) loses over \(p_z\) in a forced position, we have

\[
U \uparrow_{lf} \tag{10}
\]

Also, for the case where \(p_y\) loses over \(p_z\) in an unforced position, we have

\[
U \uparrow_{lu} \tag{11}
\]

II. Game results where \(p_x\) is forced. In this case we have the following results:

\[
U \uparrow_{fuw} \land U \uparrow_{lu} \tag{12}
\]

The full proof for (12) is in the technical report (Abdelmeged, Xu, and Lieberherr 2015). Now, CR can be summarized in terms of monotonicity constraints as,

\[
U \uparrow_{wf} \land U \uparrow_{wu} \land U \uparrow_{lf} \land U \uparrow_{lu} \tag{13}
\]

3.2 Non Negative Effect of Winning (NNEW)
Let us consider a game result \(\{r\}\) where \(p_x\) won against a third player \(p_z\), \(p_x\) could have won either in a forced or unforced position.

First, considering the case where \(p_x\) wins over \(p_z\) in a forced position, we have,

\[
U(wf_x(T) + 1, wu_x(T), l_f_x(T), l_u_x(T)) \leq U(wf_x(T), wu_x(T), l_f_x(T), l_u_x(T)) \tag{14}
\]

From equations 1 and 14, we get the monotonicity constraint,

\[
U \downarrow_{wu} \tag{15}
\]

Similarly, for the case where \(p_x\) wins over \(p_z\) in an unforced position, we have

\[
U \downarrow_{wu} \tag{16}
\]

Summarizing the monotonicity constraints, we have,

\[
U \downarrow_{wu} \land U \downarrow_{wu} \tag{17}
\]

\[
\forall p_x, p_y \in P, \forall T \subseteq R(P), \forall r \in \{r | r \in R(P) \setminus T \land \text{control}(p_x, r)\}, \quad [p_x \leq^T p_y \Rightarrow p_x \leq^{T \cup \{r\}} p_y] \tag{18}
\]

\[
\forall p_x, p_y \in P, \forall T \subseteq R(P), \forall r \in \{r | r \in R(P) \setminus T \land \neg \text{control}(p_x, r)\}, \quad [p_x \leq^T p_y \Rightarrow p_x \leq^{T \cup \{r\}} p_y] \tag{19}
\]
3.3 Non Positive Effect of Losing (NPEL)
Let us consider a game result \{r\} where \(p_y\) lost against a third player \(p_z\).

First, considering the case where \(p_y\) loses over \(p_z\) in a forced position, we have,
\[
U(w_{f_p}(T), w_{u_{p_z}}(T), l_{f_p}(T), l_{u_{p_z}}(T)) \leq U(w_{f_p}(T), w_{u_{p_z}}(T), l_{f_p}(T) + 1, l_{u_{p_z}}(T))
\]  
(18)
From equations 1 and 18, we get the monotonicity constraint, \(U \uparrow_{lf}\). Similarly, for the case where \(p_y\) loses over \(p_z\) in an unforced position, we have \(U \uparrow_{lu}\). Summarizing the monotonicity constraints, we have,
\[
U \uparrow_{lf} \land U \uparrow_{lu}\quad (19)
\]

3.4 Local Fault Based (LFB)
As we want the ranking relation to satisfy all the three properties NNEW, NPEL and CR, from equations 13, 17 and 19, we get the monotonicity constraints,
\[
U \downarrow_{lu} \land U \downarrow_{wu} \land U \downarrow_{lf} \land U \uparrow_{lu}
\]  
(20)
This tells us that the scoring function should be monotonically non-decreasing on faults and indifferent on other parameters. We call the ranking relation that uses a scoring function that satisfies equation 20 as Local Fault Based (LFB). The monotonicity constraints in equation 20 can be easily reformulated in predicate logic as follows. LFB: Games in which you don’t make faults don’t affect your rank.
\[
\forall p_x, \forall p_y \in P, \forall T \subseteq R(P),
\forall r \in \{r\} r \in R(P) \setminus T \land \neg fault(p_x, r)
\]
\[
[p_x \preceq_T p_y \iff p_x \preceq \{r\} \cup \{r\}\]  
(21)

3.5 Collusion-Resistance Theorem
We just proved the Collusion-Resistance Theorem:

\[
(NNEW \land NPEL \land CR) \iff \text{LFB}
\]
This theorem tells us that collusion-resistant ranking functions have a simple form based on fault counting. There is an infinite family of such functions that can be used in the design of techno-social systems with guaranteed collusion resistance. The Collusion-Resistance Theorem is surprising: One would expect that counting wins against non-forced players would also be a good scoring function but it is not collusion resistant.

4 Conclusion
We propose the concept of side-choosing Game (SCG) as a model for plausibility checking of claims using a generalization of extensive form games. SCGs are useful for organizing techno-social systems for creating knowledge in Formal Sciences. Considering that a specific kind of collusion might compromise the truth, we modeled the ranking of participants functionally via three axioms or postulates: NNEW (Non-Negative Effect for Winning), NPEL (Non-Positive Effect for Losing) and the crucial axiom CR (Collusion-resistance, which says that games where one is not in control cannot affect ones ranking, hence preventing gaming the game). We prove the Collusion-Resistance Theorem which states that ranking has to be based on fault counting.

What comes next? Our plan is to deploy SCG-based applications on the web and gather the benefits of collective intelligence. So far, we have already applied SCG-based ideas and tools in designing courses at Northeastern University from algorithm and software development courses to basic courses on spreadsheets and databases. And we were planning to build a tool that can be used in MOOCs or algorithm competitions. An implementation of a domain-specific language for human computation in formal sciences is a challenge that requires several algorithms to be developed. Why not develop those algorithms with SCG-based human computation effectively bootstrapping the system based on user feedback. We view SCG as the programming language for human computation to solve complex problems.

Another important area that needs further work is where participants can propose new claims. A modular approach to solving claims is needed. For example, a complex claim \(C_1\) might be reducible to a simpler claim \(C_2\) so that a solution for \(C_2\) implies a solution for \(C_1\). We propose a formal study of claim relations which can themselves be captured as claims and approached with side-choosing games.

Collusion is linked to trust in a tournament to find the best players. Collusion-resistance eliminates some collusion but there is still other collusion possible. We will report on this at the workshop.

Acknowledgments: This work was supported by the Northeastern University Lieberherr Fund supported by Pamela and Ian Holland and other alumni. We would like to thank Thomas Wahl for his feedback.

References: For space reasons we give only a few references. A complete set is in Ahmed Abdelmeged’s dissertation (Abdelmeged 2014) on which this paper is based. However, side-choosing games are a contribution of this paper. Abdelmeged used semantic games with side-choice to formulate his results.

References