Design and Secure Evaluation of Side-Choosing Games

AHMED ABDELMEGED, Northeastern University
RUIYANG XU, Northeastern University
KARL LIEBERHERR, Northeastern University

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 programming language for human computation for formal problem solving and our interest in this paper is on how to evaluate an SCG tournament fairly and effectively. We observe that a specific kind of collusion, where players lie about their strength and sacrifice themselves, could bias the evaluation of SCG tournaments dramatically. Following the idea of Social Choice Theory in the sense of Arrow, we take an axiomatic approach to guarantee that a specific kind of collusion is impossible. We 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. Limit: 18 pages. July 24-28, '16 The Netherlands.

1. INTRODUCTION

A side-choosing game (SCG) $H = \langle G, GS, Q, p_x, p_y \rangle$ 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* has a winning strategy (*white* moves first).

The players p_x, p_y of H have their preferred, static side (*white* = *Proponent* or *black* = *Opponent*), depending on whether they believe Q or !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 dynamic sides which we implement by making (per game) at most one of them the devil's advocate (or forced). The dynamic assignment is made by a trusted third party fairly. The details of this assignment are not important to this paper as we focus on evaluating the game results abstracting from this assignment.

Use the second figure from Google Drive.

The side-choosing game $\langle G, GS, Q, p_x, p_y \rangle$ produces a game result row consisting of (1) the **winner** (p_x or p_y) (2) the **loser** (p_x or p_y) and (3) at most one **forced** player (p_x or p_y or 0, if none was forced). A set of game results produced by multiple binary SCGs is called an SCG-Table which we often interpret as a labeled multi-graph (see Table I and the corresponding graph in Fig. 2 and for more graphs Fig. 4 and 5).

[1] Extensive form games are widely used to model multi-agent sequential decision making. They are represented by game trees.

Author's addresses: A. Abdelmeged, R. Xu and K. Lieberherr, College of Computer and Information Science, Northeastern University; email: lieber@ccs.neu.edu

DOI: http://dx.doi.org/10.1145/0000000.0000000

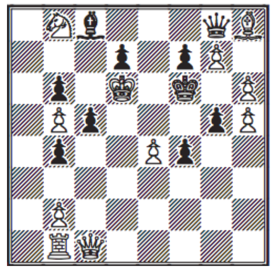

Fig. 1. An SCG example: white mates in two

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= (**winner**="Alice", **loser**="Bob", **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 subfamily are claims in formal sciences with side choice added. A second important subfamily are claims related to algorithm specifications for algorithm competitions (a la TopCoder, see topcoder.com or Kaggle, see kaggle.com).

1.2. Motivation

Why is the concept of SCG useful? A game state GS of an extensive form game G is a model of a claim. Claims are ubiquitous in human reasoning. What is surprising to some readers is that any formal claim can be brought to the extensive game form. The idea behind the translation is to express the claim as an interpreted predicate logic sentence. The quantifiers determine the game moves: There is a player (Proponent) corresponding to the existential quantifiers and a player (Opponent) corresponding to the universal quantifiers. SCG can be viewed as a plausibility check 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 show how to aggregate multiple such plausibility checks in a robust manner so that players cannot collude against other players [Abdelmeged 2014].

[2]Taken from the collection: SEVENTY-FIVE CHESS PROBLEMS by John Thursby, Trinity Coll., Cambridge, 1883.

We study the ranking of players, given a table of game results. We want to find the most meritorious players in a robust fashion so that collusion between players cannot prevent the most meritorious players to be recognized. For example, we want to prevent that a player lies about its strength and "sacrifices" himself by intentionally losing. This might rank undeserving players highly and will destroy the purpose of SCGs. For an explicit example of collusion, see section 4.9.

Finding the most meritorious player is an interesting problem because there can be a lot of noise in a game result table. For example, a player may correctly choose the white side but then not have the necessary skills to defend the correct side. Or a game state GS may be considered a winning state for white until a new player joins who knows how to win as black. To bring order into this complex situation, we propose an axiomatic approach.

The rest of the paper is organized as follows. Section 2 describes pertinent related work. Section 3 introduces the main concepts, including an interpretation in terms of graphs. It introduces the axioms NNEW, NPEL and CR. Section 4 provides a correspondence between our axioms and monotonicity constraints which provides the basis for many of our proofs. It introduces our main formal result, the Collusion-Resistance Theorem and several ramifications of it. It also introduces an equilibrium concept for SCGs. Section 5 discusses the broad applications of SCGs and section 6 mentions future work.

2. RELATED WORK

Our concept of a side-choosing game is very broad but has not been formally studied before. We were influenced by semantic games which have a long history in logic [Kulas and Hintikka 1983]. Falsifiability as promoted by Karl Popper and many others was another strong influence. A claim is falsifiable if there is an argument which proves the claim to be false. We use a weaker form of falsifiability which we call plausibility checking. A claim C for which player p_x is a proponent is plausibility checkable by a player p_y if there is an argument involving p_x and p_y that brings p_x into a contradiction with respect to C. The argument is an interactive "debate" (p_y winning a game against p_x) but it does not prove that the claim is false, in general.

In [Rubinstein 1980], Rubinstein provides an axiomatic treatment of tournament ranking functions that bears some resemblance to ours. Rubinstein's treatment was developed in a primitive framework where "beating functions" are restricted to complete, asymmetric relations. Rubinstein showed that the points system, in which only the winner is rewarded with a single point is *completely* characterized by the following three *natural* axioms:

— anonymity which means that the ranks are independent of the names of players,
— positive responsiveness to the winning relation which means that changing the results of a player p from a loss to a win, guarantees that p would have a better rank than all other players that used to have the same rank as p, and
— Independence of Irrelevant Matches (IIM) which means that the relative ranking of two players is independent of those matches in which neither is involved.

Our important CR axiom is at least as strong as Rubinstein's IIM.

Our work also falls into the field of axiomatic approaches to measures on graphs, an area active since the 1980s and inspired by earlier Social Choice theory [Rubinstein 1980; Gupte and Eliassi-Rad 2012]. Graphs are mapped to rankings of nodes or edges while axioms constrain the space of those mappings. The measures which satisfy all axioms are of interest. [Rubinstein 1980] is one of the first papers in this space. We compress edge information of the graph into a total ordering (ranking) of the graph nodes under the control of axioms that give us desirable properties. Starting from a list

Table I. $Legal(P)$ for a two players set

winner	loser	forced
1	2	1
1	2	2
1	2	0
2	1	1
2	1	2
2	1	0

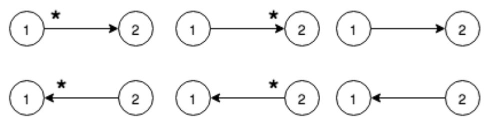

Fig. 2. Graph interpretation of Table I
(* = forced)

of axioms, which such a ranking must satisfy, we characterize functions that satisfy all the axioms (see section 4.4). We then show that there is a range of ranking functions that satisfy this characterization (see section 4.7).

[Simpson 2014] provides a comprehensive overview of techniques to "Combined Decision Making with Multiple Agents".Our work differs by working with multiple arguing or debating agents who have to defend their decisions. The concept of collusion is not mentioned in [Simpson 2014] while it is central to our analysis.

This paper is based on Ahmed Abdelmeged's dissertation [Abdelmeged 2014]. The dissertation is based on semantic games to which side-choice was added, and does not explicitly define side-choosing games. However, the proof of the Collusion-Resistance Theorem does not rely on the details of semantic games. Therefore, we introduced side-choosing games in this paper to have an appropriate context for formulating and proving the Collusion-Resistance Theorem. The proofs in this paper have been simplified through the systematic use of monotonicity constraints.

3. MAIN THEORY

We discuss ranking the players based on an SCG-table T under the axiom of collusion-resistance. When collusion-resistance does not hold, there are SCG-tables T for which a meritorious player is not top-ranked. This will frustrate meritorious players and therefore we enforce the axiom of collusion-resistance which is based on the concept of control. A player is not in control in a game if she does not participate or she loses while forced. Note that if a player is forced we cannot blame her when she loses. Collusion-resistance is formalized by expressing that adding a row where player p_x is not in control, will keep the ranking of p_x with respect to other players p_y invariant. It turns out that collusion-resistance is linked to the concept of fault: a player makes a fault if she loses while not forced.

Our ranking approach prevents sybil attacks. In an online competition, several sybils might enter and help others to win thereby preventing the strong players to win. In the presence of collusion-resistance sybils have no effect on overpowering perfect players.

Let P be the set of all players. $Legal(P)$ is the set of legal game results for P. For example, if $P = \{1, 2\}$, then Table I gives the table of all possible game results for 2 players. $Legal(P)$ contains $n \cdot (n-1) \cdot 3$ rows where n is the number of players in P. Fig. 2 gives the same information as a labeled multi-graph. 0 is used to indicate that no one was forced. In the graph notation, if no one is forced, no * is used on the edge.

$R(P)$ is the multiset (or bag) of possible game results for P allowing for repetition in the game results: $R(P)$ is a pair $(Legal(P), m)$, where m is the multiplicity function $m : Legal(P) \to \mathbb{N}_{\geq 1}$.

For our theory we define a few basic predicates: $\forall p_x \in P, \forall r \in R(P)$

$$participant(p_x, r) \Leftrightarrow p_x \text{ is a participant in } r \tag{1}$$
$$win(p_x, r) \Leftrightarrow p_x \text{ won in } r \tag{2}$$
$$loss(p_x, r) \Leftrightarrow p_x \text{ lost in } r \tag{3}$$
$$forced(p_x, r) \Leftrightarrow p_x \text{ is forced to choose a side in } r \tag{4}$$
$$\neg control(p_x, r) \Leftrightarrow \neg participant(p_x, r) \vee (loss(p_x, r) \wedge forced(p_x, r)) \tag{5}$$
$$control(p_x, r) \Leftrightarrow participant(p_x, r) \wedge (\neg loss(p_x, r) \vee \neg forced(p_x, r)) \tag{6}$$
$$control2(p_x, r) \Leftrightarrow participant(p_x, r) \wedge \neg forced(p_x, r) \tag{7}$$
$$fault(p_x, r) \Leftrightarrow loss(p_x, r) \wedge \neg forced(p_x, r) \tag{8}$$

3.1. Graph Interpretation

Our theory can be visualized in terms of directed, labeled multi-graphs. The nodes are players and the labeled edges are game results. An edge points from winner to loser. The labels on the edges indicate who is forced. We call those graphs *SCG-graphs*. Our axiomatic approach to evaluation is based on counting different edge kinds incident with a node. It is based on local counts and ignores the structure of the SCG-graph. In Fig. 2 we give the graph of game results in Table I. In the future we plan to study such graphs using more holistic techniques that take paths in the graphs into account.

3.2. Scoring Functions

We also define counting functions for scoring players:

$$wf_{p_x}(T) = \text{the win count of } p_x \text{ in } T \text{ in a forced position}$$
$$wu_{P_x}(T) = \text{the win count of } p_x \text{ in } T \text{ in an unforced position}$$
$$lf_{p_x}(T) = \text{the loss count of } p_x \text{ in } T \text{ in a forced position}$$
$$lu_{p_x}(T) = \text{the loss count of } p_x \text{ in } T \text{ in an unforced position}$$

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

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

3.3. Ranking Axioms

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

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,

$$\forall p_x, p_y \in P, \forall T \subseteq R(P)[\, p_x \preceq^T p_y \Leftrightarrow U(wf_{p_x}(T), wu_{p_x}(T), wf_{p_x}(T), lu_{p_x}(T)) \leq U(wf_{p_y}(T), wu_{p_y}(T), wu_{p_y}(T), lu_{p_y}(T))\,] \tag{9}$$

We want the ranking relation to satisfy the following axioms defined in terms of table extensions. A Venn diagram of the axioms is in Fig. 3.

— **NNEW**: Winning cannot lower your rank:

$$\forall p_x, p_y \in P, \forall T \subseteq R(P), \quad \forall r \in \{r \,|\, r \in R(P) \setminus T \wedge win(p_x, r)\} \quad [\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y \,] \tag{10}$$

— **NPEL**: Losing cannot increase your rank:

$$\forall p_x, p_y \in P, \forall T \subseteq R(P), \quad \forall r \in \{r \,|\, r \in R(P) \setminus T \wedge loss(p_y, r)\} \quad [\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y \,] \tag{11}$$

— **CR**: Games you don't control don't lower your rank.

$$\forall p_x, p_y \in P, \forall T \subseteq R(P), \quad \forall r \in \{r \,|\, r \in R(P) \setminus T \wedge \neg control(p_x, r)\} \quad [\, p_x \preceq^T p_y \Rightarrow p_x \preceq^{T \cup \{r\}} p_y \,] \tag{12}$$

This axiom allows that games you don't control might improve your rank. Indeed, this flexibility is needed for interesting ranking functions. If we don't allow this flexibility, there are no interesting ranking functions in the class we study. See subsections 4.5 and 4.6. For an explicit example of collusion and a ranking function which is not CR, see section 4.9.

3.4. Universal Domain

From equation 9, 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.

3.5. Anonymity

From equation 9 it is clear that the scoring function ignores the identity of the player in calculating the score. Hence, the ranking relation $\preceq^T$ is unaffected by changing labels and therefore anonymous.

3.6. Monotonicity of U

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:

$U \uparrow_x$: U is monotonically non-decreasing on parameter x

$U \downarrow_x$: U is monotonically non-increasing on parameter x

$U \wr_x$: U is indifferent on the parameter x

4. PROPERTIES OF RANKING RELATIONS

In this section, we reformulate the axioms as equivalent monotonicity constraints.

4.1. Collusion Resistance (CR)

Given $T' = T \cup \{r\}$, we reformulate CR as follows:

$$U(wf_{p_x}(T), wu_{P_x}(T), lf_{p_x}(T), lu_{p_x}(T)) \leq \\ U(wf_{p_y}(T), wu_{p_y}(T), lf_{P_y}(T), lu_{p_y}(T)) \\ \Rightarrow \\ U(wf_{p_x}(T'), wu_{P_x}(T'), lf_{p_x}(T'), lu_{p_x}(T')) \leq \\ U(wf_{p_y}(T'), wu_{p_y}(T'), lf_{P_y}(T'), lu_{p_y}(T')) \quad (13)$$

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,

$$U(wf_{p_x}(T'), wu_{P_x}(T'), lf_{p_x}(T'), lu_{p_x}(T')) = \\ U(wf_{p_x}(T), wu_{P_x}(T), lf_{p_x}(T), lu_{p_x}(T))$$

$$U(wf_{p_y}(T'), wu_{p_y}(T'), lf_{P_y}(T'), lu_{p_y}(T')) = \\ U(wf_{p_y}(T) + 1, wu_{p_y}(T), lf_{P_y}(T), lu_{p_y}(T))$$

From the CR constraint above, we have:

$$U(wf_{p_x}(T), wu_{P_x}(T), lf_{p_x}(T), lu_{p_x}(T)) \leq \\ U(wf_{p_y}(T) + 1, wu_{p_y}(T), lf_{P_y}(T), lu_{p_y}(T)) \quad (14)$$

From equations 9 and 14, we get the monotonicity constraint,

$$U \uparrow_{wf} \quad (15)$$

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,

$$U(wf_{p_x}(T), wu_{p_x}(T), lf_{p_x}(T), lu_{p_x}(T)) \leq \\ U(wf_{p_y}(T), wu_{p_y}(T) + 1, lf_{p_y}(T), lu_{p_y}(T)) \quad (16)$$

From equations 9 and 16, we get the monotonicity constraint,

$$U \uparrow_{wu} \quad (17)$$

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

$$U \uparrow_{lf} \quad (18)$$

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

$$U \uparrow_{lu} \quad (19)$$

II. Game results where p_x lost but forced. Let us consider game results $\{r\}$ where p_x was forced to lose against some third player p_z, given $T' = T \cup \{r\}$ we have,

$$U(wf_{p_x}(T'), wu_{P_x}(T'), lf_{p_x}(T'), lu_{p_x}(T')) = \\ U(wf_{p_x}(T), wu_{P_x}(T), lf_{p_x}(T) + 1, lu_{p_x}(T))$$

$$U(wf_{p_y}(T'), wu_{p_y}(T'), lf_{P_y}(T'), lu_{p_y}(T')) = U(wf_{p_y}(T), wu_{p_y}(T), lf_{P_y}(T), lu_{p_y}(T))$$

Following the CR constraint above, we have:

$$U(wf_{p_x}(T), wu_{p_x}(T), lf_{p_x}(T) + 1, lu_{p_x}(T)) \leq U(wf_{p_y}(T), wu_{p_y}(T), lf_{p_y}(T), lu_{p_y}(T)) \quad (20)$$

From equations 9 and 20, we get the monotonicity constraint,

$$U \downarrow_{lf} \quad (21)$$

Now, CR can be summarized in terms of monotonicity constraints as,

$$U \uparrow_{wf} \wedge U \uparrow_{wu} \wedge U \wr_{lf} \wedge U \uparrow_{lu} \quad (22)$$

RUIYANG: reviewer 3 says that the logical step from the initial formulation to the reformulation is rather large. Can you make the step appear smaller by giving an informal argument why (22) is the same as CR? RUIYANG: A helpful addition would be to describe the general translation of monotonicity constraints to predicate logic. A helpful addition would be to give the general translation of a family of predicate logic sentences to monotonicity constraints.

4.2. 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,

$$U(wf_{p_x}(T) + 1, wu_{p_x}(T), lf_{p_x}(T), lu_{p_x}(T)) \leq U(wf_{p_y}(T), wu_{p_y}(T), lf_{p_y}(T), lu_{p_y}(T)) \quad (23)$$

From equations 9 and 23, we get the monotonicity constraint,

$$U \downarrow_{wf} \quad (24)$$

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

$$U \downarrow_{wu} \quad (25)$$

Summarizing the monotonicity constraints, we have,

$$U \downarrow_{wf} \wedge U \downarrow_{wu} \quad (26)$$

4.3. 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,

$$U(wf_{p_x}(T), wu_{p_x}(T), lf_{p_x}(T), lu_{p_x}(T)) \leq U(wf_{p_y}(T), wu_{p_y}(T), lf_{p_y}(T) + 1, lu_{p_y}(T)) \quad (27)$$

From equations 9 and 27, 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,

$$U \uparrow_{lf} \wedge U \uparrow_{lu} \quad (28)$$

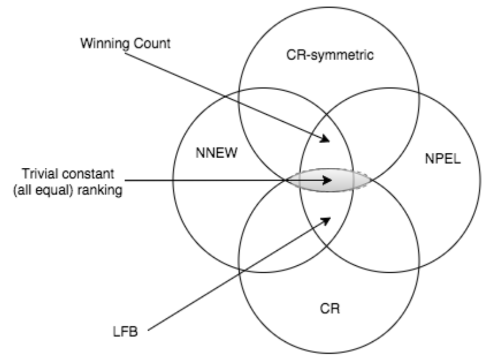

Fig. 3. Relations among NNEW, NPEL, CR and CR-symmetric which touch at the all-equal ranking.

4.4. Local Fault Based (LFB)

As we want the ranking relation to satisfy all the three properties NNEW, NPEL and CR, from equations 22, 26 and 28, we get the monotonicity constraints,

$$U \wr_{wf} \wedge U \wr_{wu} \wedge U \wr_{lf} \wedge U \uparrow_{lu} \tag{29}$$

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 29 as Local Fault Based (LFB). The monotonicity constraints in equation 29 can be easily reformulated in predicate logic.

— LFB: Games in which you don't make faults don't affect your rank.

$$\forall p_x, p_y \in P, \forall T \subseteq R(P), \quad \forall r \in \{r \,|\, r \in R(P) \setminus T \wedge \neg fault(p_x, r)\} \quad [\, p_x \preceq^T p_y \Leftrightarrow p_x \preceq^{T \cup \{r\}} p_y \,] \tag{30}$$

— **Collusion-Resistance Theorem** We just proved:

$$(\text{NNEW} \wedge \text{NPEL} \wedge \text{CR}) \Leftrightarrow \text{LFB}$$

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 (see section 4.7). 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.

Fig. 3 gives a graphical rendering of the LFB property.

4.5. CR in retrospect

CR says that game results where you are not in control cannot lower your rank. We consider CR-symmetric: game results where you are not in control cannot increase your rank. We formalize CR-symmetric by using the worse-than relation ($\succ$) defined by

$$p_x \succ^T p_y \Leftrightarrow \neg(p_x \preceq^T p_y).$$

CR-symmetric:

$$\forall p_x, p_y \in P, \forall T \subseteq R(P), \\ \forall r \in \{r \,|\, r \in R(P) \setminus T \wedge \neg control(p_x, r)\} \\ [\, p_x \succ^{T} p_y \Rightarrow p_x \succ^{T \cup \{r\}} p_y \,] \quad (31)$$

It turns out that NNEW and NPEL and CR and CR-symmetric lead to trivial scoring functions. Proof: Following the proof approach introduced earlier, we get

$$U \wr_{wf} \wedge U \wr_{wu} \wedge U \wr_{lf} \wedge U \wr_{lu} \quad (32)$$

CR-symmetric leads to monotonicity constraints like in (22) but with the arrows pointing in the opposite direction.

We illustrate the *trivial scoring function fact* with an example. Consider fault counting which is CR and clearly not constant. We show that CR-symmetric does not hold using the following construction: Choose p_x and p_y so that $U(p_x) = U(p_y) - 1$. Then consider a game result r where p_y makes a fault in a game where p_x does not participate. Now p_x is not in control. When we add r to the table we have $U(p_x) = U(p_y)$ and CR-symmetric is violated.

4.6. Replacing CR?

What happens when we use CR-symmetric instead of CR? NNEW and NPEL and CR-symmetric are equivalent to the monotonicity constraints

$$U \downarrow_{wf} \wedge U \downarrow_{wu} \wedge U \wr_{lf} \wedge U \downarrow_{lu} \quad (33)$$

union the constraints in (26) and (28). This simplifies to

$$U \downarrow_{wf} \wedge U \downarrow_{wu} \wedge U \wr_{lf} \wedge U \wr_{lu} \quad (34)$$

which is win counting (which we know is not CR). CR-symmetric is susceptible to collusion. That is why we don't use CR-symmetric. CR and CR-symmetric in the presence of NNEW and NPEL are "contradictory" in the sense that they allow only uninteresting scoring functions.

In Fig. 3 we show graphically how the four properties NNEW, NPEL, CR and CR-symmetric intersect.

4.7. A Family of Collusion-Resistant Rankings

When designing a techno-social system for solving precisely formulated problems there are are many concerns to be addressed. Besides just using simple fault-counting, there are other weighted fault counting functions of interest. In the graph representation of an SCG-table T, we consider two kinds of edges: α edges going into p_x are edges where no one was forced. β edges going into p_x are edges where the winner was forced. In both cases p_x made a fault but we are counting the two kinds of edges differently. α (β) edges have weight $\alpha > 0$ ($\beta > 0$), respectively. The resulting scoring function U has the property that a high α (compared to β) encourages non-forced players to win. Other families of collusion-resistant rankings can be defined by considering finer-grained properties of game results.

4.8. A Simple Property of Fault Counting

We consider the ranking we get from the scoring function U which counts faults in a table T ($lu_{p_x}(T)$). A *quasi-perfect player* p_x is a player with zero fault counts ($lu_{p_x}(T) = 0$). (A perfect player is quasi-perfect but the converse does not necessarily hold because a quasi-perfect player may make faults in the future.) A *top-ranked* player is a player for which there exists no stronger player in the ranking. We have the simple but desirable

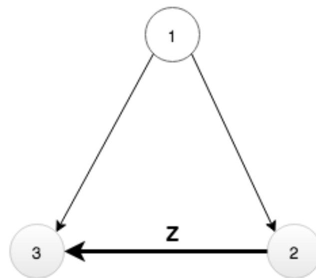

Fig. 4. Simplest Counterexample: Players 2 and 3 collude

Table II. Game results for Fig. 5

player	win	loss	fault
1	2	2	0
2	z+1	1	0
3	1	z+1	z

meritocracy property: for all SCG-Tables quasi-perfect implies top-ranked under fault-counting. This easily generalizes to: When a ranking is LFB then for all SCG-tables, quasi-perfect players are top ranked. Next we give an explicit counterexample for win counting.

4.9. Counterexamples for Win Counting

We assume that players can recognize each other and use that knowledge to alter their play. This assumption is satisfied in most applications even when the players are implemented in software.

Under win counting, quasi-perfect players are not necessarily top-ranked. Win counting is defined by

$$U(p_x, T) = -(wf_{p_x}(T) + wu_{p_x}(T)) \tag{35}$$

The corresponding ranking function is NNEW and NPEL but ¬ CR. Fig. 4 gives the smallest counterexample (both in terms of number of players and number of game results).

We set $z = 3$. Player 2 is top ranked with the 3 wins and 1 fault. But, player 1, the quasi perfect player with no faults is not top ranked. The reason is that there is collusion: player 3 helped player 2 accumulate wins which helped to overrule the quasi-perfect player.

4.9.1. All Perfect with Liars. How many colluding players are needed to prevent a perfect player from winning under win counting? We show with an example that 2 out of n are enough provided the 2 players play enough games.

We have $n-2$ perfect players and a total of n players. The tournament is basically a full round-robin tournament where the non-forced player always wins, except for the pair of colluding players. One player (the liar) helps the other player to accumulate wins. The helping games have multiplicity z. Fig. 5 shows the graph of game results for $n = 3$ with players numbered 1,2,3. Player 1 is perfect, 3 is the liar (lying about its strength) and 2 is being helped. Players 2 and 3 collude: player 3 (the liar) helps player 2 win points. Although player 3 is also perfect, it lies about its strength when it plays against player 2. Table II shows the game statistics: For $z \geq 2$ only fault-counting is collusion-resistant and player 1 is top-ranked. For each scoring function, the cells corresponding to the best scores have been shaded (currently * means shading).

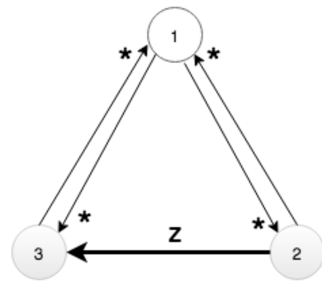

Fig. 5. Player 3 lies about its strength

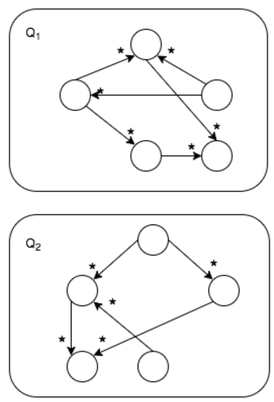

Fig. 6. Equilibrium of two disjoint subsets of P

4.10. Equilibria

Let Q be a subset of a set of players P. Q is in an equilibrium if games between players in Q are fault-free. This means that players in P agree on a construction for defending the claim. They always win when they are not forced and they always lose when they are forced. There might be several islands of agreement represented by disjoint subsets of P. Let Q_1 and Q_2 be two disjoint subsets of P, each being in an equilibrium (see Fig. 6). What happens when a player in Q_1 plays against a player in Q_2? If the construction used in Q_2 is of "higher quality," the player in Q_2 will win when not forced. The quality of construction used by a player p_x might increase over time. p_x might have an insight which leads to a better solution shaking up an equilibrium. There will be again games leading to faults until the island of player p_x has learned about the new construction.

5. SCG APPLICATIONS

After the introductory examples in section 1.1 and the theory, we outline now the breadth of applicability of SCGs. We motivate the importance of SCGs by describing applications, users and owners.

- **SCG Applications.**
  - **Formal Sciences** Formal sciences are disciplines concerned with formal systems, such as logic, mathematics, statistics, theoretical computer science, information theory, game theory, systems theory, and decision theory. A claim is defined using an *interpreted* predicate logic sentence. This is *not* an exercise in logic as the quantifiers are only used to define the tasks that the users must perform. The sen-

tence is interpreted in a structure which might be defined by a complex program encoding the functionality best executed by computers.

  - **Formal Claims based on Simulation Environments** Robotics and biological sciences, etc. fall into this category. The structure in which the claim is interpreted is the simulation environment.

- **SCG Users.** Users are problem solvers or learners and they operate directly or indirectly. In **direct mode**, the users perform the moves themselves, maybe using software. In **indirect mode**, the users produce software that plays the SCG on their behalf. There is a simple SCG-interface that the software has to follow. Of course, indirect users must have software development skills.
  The indirect mode is of central interest to us because it is a novel approach to develop software for computational problems using a group of people. The quality control of the software is automated by running an online or offline tournament to determine the top-ranked software. The claim under consideration determines what quality means. Note that the SCG-interface implies that testing is an integral part of the solution.
  Users of SCGs include:
  - **Students in high schools and universities.** They must understand the concept of a claim. Focus is on dissemination of knowledge through peer teaching and peer evaluation.
  - **Researchers.** Focus is on creation of new knowledge and its peer evaluation. Researchers propose claim variations.
  - **Citizen Scientists.** They might find innovative constructions that are imperfect. Experts might benefit from those ideas and correct them.
- **SCG Owners.**
  Owners define claims. Some users also play the role of owners. Owners don't need expertise how to solve the problems.
  Owners include: (1) Teachers and Professors. (2) Research Directors, Heads of Research Programs, Organizations like NSF, DARPA, ONR etc. (3) Program Chairs of conferences and Journal Editors. (4) Companies who need a specific computational problem solved for which no off-the-shelve solution is available. (5) Companies who are looking for employees with skills in a specific domain. E.g., Facebook organized a competition on kaggle.com and the winner got a Facebook job.

5.1. Applications of Side-Choosing Games to Existing Systems

Our study of side-choosing games is motivated by their potential to organize problem-solving competitions and by their successful use in CS education at Northeastern University. We believe SCGs are a foundation for platforms like TopCoder or Kaggle or for scientific human-computation tools like Foldit [Cooper et al. 2010].

- **Education in Formal Sciences.** Our favorite way of summarizing learning objectives for a formal science domain is to say that learners must demonstrate the skill of judging claims in the domain, choosing their side on the claim and then defending their side choice through game play against other students. The resulting peer-teaching and peer-grading is very attractive. A claim is representing a lab in which students learn and is chosen in such a way that solving the problem requires skills that students should have.
  - **Using piazza.com.** To post claims and to organize the playing of games related to those claims we used piazza.com. This worked very well, especially when we divided the Algorithms class into small groups of three students and kept the games in those small groups. The undergraduate students solved challenging problems like finding the worst-case input for the Gale-Shapley algorithm or optimally solving a product stress testing problem.

— **Using our own software.** In software development classes we had the students develop "avatars" to play the game and we did a full-round-robin tournament evaluation of the avatars. The problem to be solved was a maximum constraint satisfaction problem.

— **Improving Evaluation in Problem-Solving Competitions for Computational Problems.** A significant advantage of our approach is that the evaluation of solutions is done by peers and not the competition organizer. This is relevant to systems like topcoder.com and various competitions like SAT-solver competitions. The competition organizer only acts in a role as referee. Instead of static benchmarks, dynamic benchmarks are developed through game play.
The quality of the solutions produced depends on the skills of the players who might not be motivated or not have the knowledge necessary to solve the problem. To attract strong players either money or fame has to be given; a common theme in human computation.

5.1.1. Gale-Shapley Lab. We present an example from our Algorithms class. The students have studied the Gale-Shapley algorithm for producing a stable matching of n women with n men given their preferences. To get a better understanding of how the algorithm works (it is a loop), the students have to find for a given n a set of preferences which create the most number q of iterations of the algorithm. The claim $GSW = GaleShapleyWorstCaseClaim(n = 10, q = 30)$ says that for 10 women and men there is a set of preferences generating 30 iterations of the outer loop of the Gale-Shapley algorithm. And the claim is also that it is not possible to have more iterations with other preferences. The predicate logic representation of GSW automatically produces the following game between a P(roponent) and O(pponent): P produces an input $i(n)$ of preferences for n women and men. The algorithm is run on $i(n)$ and produces $q(n)$ iterations. If $q(n) < 30$, P has made a fault. If $q(n)$ is too small, O produces input $i_1(n)$ which is run and produces $q_1(n)$ iterations. If $q_1(n) > 30$, P has also made a fault. This is the essence of the semantic game behind the predicate logic formula specifying the problem.

5.1.2. Approximate MaxCSP Lab. We present a simple example of an algorithm development lab. We are interested in algorithms for approximately solving MaxCSP instances with guaranteed performance. We are considering Boolean constraint satisfaction problems of the following form: Each constraint is of the form $R(x_1, x_2, x_3)$ which is true when exactly one of the three Boolean variables is true. Given a CSP formula consisting of n variables we are interested in finding an assignment that satisfies the fraction τ_R of the constraints and we want to maximize τ_R. It turns out that $\tau_R = 4/9$. The SCG behind this problem has to deliver counterexamples (where the fraction t cannot be satisfied) if $t > \tau_R$ and to produce an assignment where the fraction t is satisfied, if $t \leq \tau_R$. Notice that in this context the algorithm designer needs not only to provide an algorithm which satisfies the required fraction of constraints but she also needs an algorithm that can produce "hard" inputs.

5.2. 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. 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. This paper is addressing point (2). Point (1) is addressed by having reliable software to check the game rules related to the claims.

6. FUTURE WORK

The work in this paper abstracts away from who is proponent and who is opponent of a claim in a game. When the proponent/opponent information is considered we have a richer labeling structure on the edges of the SCG-graphs. Each edge gets a pair of static and dynamic labels (**What do you mean static and dynamic here?**) where the dynamic labels are determined by the static labels plus the forcing information. Recall from the introduction that a static labels provide the side-choices $\{P(roponent), O(pponent)\}$ and the dynamic labels provide the roles used when the game is played. We call those graphs *extended* SCG-graphs.

A player is called *consistent*, if it always uses the same static side-choice across all games. We plan to prove the following Collusion-Inconsistency Conjecture: For all ranking functions R (which are not LFB) and for all extended SCG graphs where there is a quasi-perfect player that is not top-ranked under R, there exists a player that is not consistent. This conjecture would prove that non-collusion-resistance implies inconsistency (**This is not true, see Figure 4**).

We call an SCG-graph *consistent* if it has a completion to an extended SCG-graph where all players are consistent. The SCG-graph in Fig. 4 is inconsistent because of the odd cycle and the fact that none of the players is forced. The SCG-graph in Fig. 5 is inconsistent too. We conjecture that the SCG-graph consistency problem is solvable in polynomial time. Note that a mapping from nodes to $\{P(roponent), O(pponent)\}$ that is compatible with the SCG-graph, serves as a witness for SCG-graph consistency. *Compatibility* of a node mapping is defined in terms of the forced labels: when the two nodes incident with an edge have the same value under the map then exactly one of the two nodes must be forced and if they have different values then none of the two nodes must be forced.

We want to study SCGs with imperfect information and with random moves. Independence-friendly logic and the corresponding semantic games are a good starting point.

An interesting question is what can be said about the truth value of a claim given an SCG-table of game results and information about the strength of the players.

Another avenue is to consider more general scoring functions U. For example, U could take also the function $np^T(p_x)$ into account. $np^T(p_x)$ is the number of game results in table T where p_x does not participate. The higher np, the weaker a player.

Collusion is linked to trust in an SCG-Table T to find the best players. Collusion-resistance eliminates some collusion but there is still other collusion possible. To explore the link between trust and collusion is interesting future work. Trust can be improved by controlling the game scheduler to enforce Swiss-style scheduling, for example.

7. CONCLUSION

We propose the concept of side-choosing Game (SCG) as 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 players 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 players can propose new claims. A modular approach to solving claims is needed. For example, a complex claim C_1might 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.

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.

Acknowledgments We would like to thank Thomas Wahl for his influential feedback on an earlier version of the paper.

REFERENCES

ABDELMEGED, A. 2014. Organizing computational problem solving communities via collusion-resistant semantic game tournaments. Ph.D. thesis, Northeastern University, Boston, MA, USA. Advisor-Karl Lieberherr.

COOPER, S., TREUILLE, A., BARBERO, J., LEAVER-FAY, A., TUITE, K., KHATIB, F., SNYDER, A. C., BEENEN, M., SALESIN, D., BAKER, D., AND POPOVIĆ, Z. 2010. The challenge of designing scientific discovery games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. FDG '10. ACM, New York, NY, USA, 40–47.

GUPTE, M. AND ELIASSI-RAD, T. 2012. Measuring tie strength in implicit social networks. *Proceedings of the 4th Annual ACM Web Science Conference*, pp. 109–118.

KULAS, J. AND HINTIKKA, J. 1983. *The Game of Language: Studies in Game-Theoretical Semantics and Its Applications*. Synthese Language Library. Springer.

RUBINSTEIN, A. 1980. Ranking the participants in a tournament. *SIAM Journal on Applied Mathematics 38,* 1, pp. 108–111.

SIMPSON, E. 2014. Combined decision making with multiple agents. Ph.D. thesis, University of Oxford.