Team Competition*

Pingzhong Tang
Computer Science
Department
Hong Kong University of
Science and Technology
Clear Water Bay, Hong Kong
kenshin@cse.ust.hk

Yoav Shoham
Computer Science
Department
Stanford University, California,
94305
shoham@stanford.edu

Fangzhen Lin
Computer Science
Department
Hong Kong University of
Science and Technology
Clear Water Bay, Hong Kong
flin@cse.ust.hk

ABSTRACT

In a team competition, two participating teams have an equal number of players, and each team orders its players linearly based on their strengths. A mechanism then specifies how the players from the two teams are matched up and how to score them. There are two types of manipulations by a team: Misreporting the strength ordering and deliberately losing a match. To identify these strategically behaviors, we model the team competition problem in a game-theoretical framework, under which we prove necessary and sufficient conditions which ensure that truthful reporting and maximal effort in matches are equilibrium strategies, and which further ensure certain fairness conditions described by choice functions.

Categories and Subject Descriptors

J.4 [**Social and Behavior Sciences**]: Economics; I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent System

General Terms

Economics

Keywords

Team Competition, Mechanism Design, Truthfulness, Dominant Strategy Implementation

1. INTRODUCTION

Tens of centuries ago in ancient China, the emperor Qi challenged his minister Tian to a horse race. The rule was that each of them would put forward three horses, one at a time, and each time the two horses would race. Whoever won at least two of the races would be the winner. As the story goes, Tian learnt that his best horse was not as good as the Qi's best but better than Qi's second best one, and his second best one was not as good as Qi's second best but better than Qi's worst one. Knowing that Qi would put forward his best horse first, his second best second, the clever minister put forward his worst horse first, his best horse second and his second best last. As expected, while Tian's worst horse lost badly to Qi's best horse at the beginning of the competition, he won it

nonetheless because his best beat Qi's second best, and his second best beat Qi's worst horse.

More generally, we can consider the competition to be between two teams (e.g, the emperor's team and the minister's team in the story), and each team has an equal number of players (e.g, the horses in the story). Each team is asked to provide an order by which his players will compete, and the rule or the agenda of the competition specifies how the team's players will be matched up based on the orders given by the two teams, and who will win the competition based on the individual matches. In the story, the rule was simply that the players from the two teams raced against each other according to the orders given by the two teams, and whoever won more matches won the competition.

Similar examples are the current international team competitions for table tennis (aka Corbillon Cup and Swaything Cup). Two competing countries put forward three players in a certain order. The agenda is a modification of the horse racing one by adding two matches between the first player and the second player from each team.

In this paper, we consider how to design rules of a competition so that the outcome will reflect the "true strength" of the competing teams. In the case of horse race mentioned above, one is inclined to say that the emperor's team was stronger and should have won the competition. But in general, what counts as the "true strength" of a team and how to compare them is also an issue that we need to consider. To address these problems, we shall consider the team competition problem as a mechanism design problem.

Since first mathematically analyzed by Hurwicz [2], the theory of mechanism design has been extensively studied in economics (see [5] for an introduction and [6] for a survey), theoretical computer science [7] and artificial intelligence [10, 11, 9].

We formulate a team competition as a mechanism which takes the reported orderings as the input and returns an outcome of the competition. According to the reported orderings, the mechanism specifies a number of single matches and the score awarded to the winning player of each match. Each team then gets the sum of all the scores from its winning players and such a score profile from both teams reflects the outcome. Each team prefers an outcome with higher own score and lower opponent's score.

Based on the dependence between the matches specified, we categorize all the mechanisms into the static and the dynamic. In a static mechanism, each match is predetermined and takes place independently of the results of other matches while in a dynamic mechanism, each match may only take place if the results of previous matches satisfy certain condition.

We propose three major objectives for a desirable mechanism to achieve: truth revelation, fairness and conciseness. In other words, truth revelation requires that reporting the truthful ordering of its

---

players is a dominant strategy for each team in the game induced by the mechanism, that is, no matter what the opponent reports, a team can never benefit from misreporting. Fairness requires that when both team play truthfully, the resulting outcomes fairly reflect the *strength* of each team, which is characterized by a choice function. For each state of the world (described by an ordering over all the players), the choice function selects socially desirable outcome to be implemented by the mechanism. Fairness can then be described by a set of conditions imposed on the choice function such as player anonymity, team anonymity, monotonicity and pareto efficiency. We also propose four particular choice functions that satisfies all the conditions. Finally, upon the satisfaction of the previous two objectives, we sometimes require conciseness of the mechanism, that is, the mechanism keeping the number of matches as small as possible.

We then give sufficient and necessary conditions to allow a mechanism to implement these objectives. In particular, static mechanisms can truthfully implement two choice functions mentioned above but fail to implement the other two, while for dynamic mechanisms, even a subclass called "knock-in" suffices in truthfully implementing all the four choice functions. Generally speaking, dynamic mechanisms are also supposed to be more concise than the static ones.

Finally, we also deal with the moral hazard issue [4] where each team has another level of strategic behavior: a team can benefit from throwing a match in certain mechanisms. To prevent this from happening, we force honest play to be a equilibrium strategy for each team: given the other team play honestly, we would also prefer to play honestly. We give sufficient and necessary conditions for this part too. Interestingly, most of these conditions coincide with those of dominant strategy truthfulness.

The rest of the paper is organized as follows: we introduce the mechanism design model for team competition problem and then describe desirable objectives to achieve. Next we characterize the implementation issues of two types of mechanisms by several theorems. We also deal with a variation to the problem where the players can strategically lose. Finally, we discuss certain interesting issues and then conclude the paper.

2. BASIC MODELS

In this section, we build up the mathematic model for analyzing team competition.

2.1 Team Competition Environments

We first define the so called team competition environment where the designer operates.

DEFINITION 1. *A team competition environment C is a tuple (A, B, Θ, O, P), where*

- $A = \{a_1, \ldots, a_n\}$ *is the set of players of team A.*

- $B = \{b_1, \ldots, b_n\}$ *is the set of players of team B.*

- Θ *is the set of possible states, where*

  - *Each state $\theta \in \Theta$ uniquely defines a linear order $>_\theta$ on $A \cup B$. If $a >_\theta b$, then a wins the match versus b in state θ.*

  - *We denote θ_A and θ_B as the orders on A and B that are derived respectively from θ, which can be seen as the private information of A and B that cannot be observed by others. We denote Θ_A and Θ_B as the set of all possible θ_A and θ_B.*

- $O = \{(s_A, s_B) | s_A, s_B \in \mathbb{R}\}$ *is the set of outcomes of the competition. s_A and s_B are the scores for team A and B respectively.*

- P *is a preference relation over O.*

In this paper, we consider P to be the one that team A weakly prefers (s_A, s_B) to (s'_A, s'_B) iff

$$s_A \geq s'_A \text{ and } s_B \leq s'_B,$$

and the preference becomes strict if it is weak and one of the equality does not hold. Team B has the opposite preference. It is worth pointing out that when $s_A > s'_A$ and $s_B > s'_B$, the preference between (s_A, s_B) and (s'_A, s'_B) is not defined. Indeed, it is sometimes unclear whether $2 : 3$ is better than $3 : 4$[1].

An easy way to avoid this is to choose a subset of O with $s_A + s_B = c$, where c is a constant. A mechanism is constant-sum if it is defined on such a subset. Our first type of mechanisms called static mechanisms are constant-sum.

2.2 Static Mechanisms

DEFINITION 2. *A static mechanism M on a team competition environment is a tuple (S_A, S_B, C) where*

- $S_A : \Theta_A \to L_A$ *is the set of A's pure strategies that map A's private information to a possible linear orders on A. We denote L_A as the set of all linear orders on A. We also allow randomization, which is a lottery over S_A. Similar for S_B.*

- C *is an $n \times n$ score matrix $C_{n \times n}$ where*

  - *Each entry $c_{i,j}$ denotes the score assigned to the match between a_i and b_j, where a_i is the i-th player reported by A and b_j the j-th player reported by B.*

  - *The winner of the match gets $c_{i,j}$ and the loser gets 0.*

  - *The total score that team A can get in state θ is $s_A = \Sigma_{(a_i >_\theta b_j) \in A \times B} c_{i,j}$. Similar for s_B.*

  - *Such a pair (s_A, s_B) creates an outcome in O.*

In fact, when there is no restriction on Θ, Θ_A and L_A both denote the set of all permutations on A. However, we make different notations to clarify that Θ_A is the set of private information based on which A chooses an order from L_A to report.

In comparison with standard mechanism definition ([8], Chapter 10), the matrix C plays the role of an outcome function: for each state, the matrix maps the reports from A and B to an outcome $(s_A, s_B) \in O$. Note that $s_A + s_B = c$, where $c = \Sigma_{1 \leq i,j \leq n} c_{i,j}$. Therefore, it is constant-sum and the preference relation P is well defined on O.

Note also that there are potentially n^2 matches since there are n^2 entries in the matrix. However, if $c_{i,j} = 0$, then a match between a_i and b_j is not necessary, because its outcome will not affect the score of either team. Similarly, if $c_{i,j} \neq 0$, it does not necessarily mean that there is only one match between a_i and b_j. It could possibly mean that there are several matches between a_i and b_j whose scores sum up to $c_{i,j}$ so that they are equivalent to a single match with score $c_{i,j}$.

---

[1]We will get back to another type of preference where each team is not so sensitive about the score but only cares about if it wins or not. In this case, $2 : 3$ is equally preferred to $3 : 4$

EXAMPLE 1. **(Horse Race)** *The aforementioned horse race is a static mechanism, whose score matrix is as follows:*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

EXAMPLE 2. **(Table Tennis)** *The international team competition for table tennis is also a static mechanism, whose score matrix is as follows:*

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.3 Dynamic Mechanisms

Consider a static mechanism where there are 4 matches: (a_1 vs b_1), (b_1 vs a_2), (a_2 vs b_2) and (a_1 vs b_2). Suppose the matches take place sequentially and always the first player wins, then after the first 3 matches, we already know that $a_1 >_\theta b_2$ by transitivity. Therefore, the 4th match is redundant.

The above intuition can be realized in designing more concise mechanisms called dynamic mechanism, where the next match is jointly determined by the reported lists as well as the results of previous matches.

DEFINITION 3. *A dynamic mechanism D on a team competition environment is a tuple (S_A, S_B, H, f_n, R_s) where*

- *S_A and S_B are the sets of strategies the same as the ones in static mechanism*

- *$H = H_T \cup H_N$ is the set of histories, where H_T denotes the set of terminal histories and H_N are the set of nonterminal ones. They are defined inductively as follows:*

  - *$\emptyset \in H_N$*
  - *If $h \in H_N$, then $h : (a_i > b_j) \in H$ and $h : (b_j > a_i) \in H$. It says if h is a nonterminal history, then by concatenating it with the match where a_i wins b_j or b_j wins a_i, a new history is produced. The new history can be either terminal or nonterminal.*

- *$f_n : H_N \times P_A \times P_B \to A \times B$ the next function that maps each nonterminal history as well as the reported messages to a pair of players to compete in the next match.*

- *$R_s : H_T \to O$ the scoring rules that maps each terminal history to an outcome, i,e, a score profile.*

Dynamic mechanism is a more general concept than static mechanism in the sense that every static mechanism can be represented in the form of a dynamic one: the one with constant next function that arranges a list of independent matches sequentially. Quite often, it is more interesting to focus on certain specific classes of dynamic mechanisms. For instance, the following "knock-out" mechanism has been quite popular in the Go community.

EXAMPLE 3. **(Knock-out Competition)** *Upon receiving the reported lists of players $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$,*

- *a_1 vs b_1 will be initiated as the first match.*

- *In the following rounds, if the current match is a_i vs b_j, then the next function will assign a_{i+1} vs b_j if b_j beats a_i and assign a_i vs b_{j+1} otherwise.*

- *The set of terminal histories are those with every player in one team has lost. The scoring rule assign the winning team n points and the losing team the number of matches that it wins.*

It can be seen from the above example that there are at most $2n - 1$ matches in a knock-out competition: each match eliminates one player who will never show up in the future matches. This perfectly fits the context of a Go competition, where a match normally takes hours.

3. DESIRABLE PROPERTIES

The space of static as well as dynamic mechanisms is obviously infinite. For this sake, it is necessary to identify desirable properties for mechanisms to satisfy. For example, in the Horse Racing example, we might hope that a mechanism that forces both teams to report the truth. Further after receiving the truth, we might hope that the outcome of the match fairly reflects the true state. For example in the horse racing competition, we should expect the outcome to be at least some score profile in which the emperor wins. Last but not least, we might also hope that the number of matches as small as possible. We characterize some of the standards using the choice function and ask the question what mechanisms *truthfully* implement the choice function.

3.1 Dominant Strategy Truthfulness

DEFINITION 4. *We say that a mechanism is dominant-strategy truthful if for every state θ, a team cannot end up with a worse outcome by reporting its truthful order induced by $>_\theta$ than reporting any other order, no matter what the other team reports.*

If a mechanism is dominant-strategy truthful, then by being aware of the private information, each team will choose to report its truthful ordering because it is in its best interest to do so.

3.2 Choice Functions

As already mentioned, a choice function describes which outcome should occur for a given state.

DEFINITION 5. *A choice function $f : \Theta \to O$ maps a state to an outcome.*

3.2.1 Restrictions on Choice Functions

The first restriction on a choice function is what we call player anonymity, which says the players are indistinguishable inside a team.

DEFINITION 6. **(Player Anonymity)** *Suppose $p(A)$ $(p(B))$ is a permutation of $A(B)$, and if $f(\theta) = o$, then $f(\theta') = o$ where θ' is obtain from θ by replacing each $a \in A$ $(b \in B)$ by its permutation $p(a)$ $(p(b))$.*

For example,

$$a >_{\theta_1} b >_{\theta_1} b' >_{\theta_1} a',$$

should lead to the same outcome as

$$a' >_{\theta_2} b >_{\theta_2} b' >_{\theta_2} a,$$

as well as

$$a >_{\theta_3} b' >_{\theta_3} b >_{\theta_3} a'.$$

The second restriction is called team anonymity, which says the choice function f does not discriminate for or against one particular team.

DEFINITION 7. *(Team Anonymity) Suppose $p : A \to B$ is a bijection between A and B, and if $f(\theta) = (s_A, s_B)$, then $f(\theta') = (s_B, s_A)$ where θ' is obtained from θ by swapping each $a \in A$ and $p(a) \in B$.*

For example, again if $f(\theta_1) = (s_A, s_B)$ and θ_1 is as follows:

$$a >_{\theta_1} b >_{\theta_1} b' >_{\theta_1} a',$$

then $f(\theta_4) = (s_B, s_A)$, where θ_4 is as follows:

$$b >_{\theta_4} a >_{\theta_4} a' >_{\theta_4} b'.$$

.

The third restriction is the so-called monotonicity, which says no worse outcome will be brought about for a team if none of its players falls in the overall ranking.

DEFINITION 8. *(Monotonicity) For any two states θ and θ', if $f(\theta) = o$, $f(\theta') = o'$ and θ' is an improvement to θ for team $A(B)$, then $A(B)$ at least prefers o' the same as o. A state θ' is an improvement to another state θ for team $A(B)$, if $\forall a \in A$ ($b \in B$), the ranking of $a(b)$ in state θ' is improved or stays the same as in state θ.*

Finally, the last restriction pareto efficiency says that if one team has the better ith best player for all i, then it should get a higher score in the final outcome.

DEFINITION 9. *(Pareto Efficiency) If in any state θ satisfying $\forall 0 \leq i \leq n$, the i-th ranked player of team A is better ranked than team B, then $f(\theta) = (S_A, S_B)$ satisfies $S_A \geq S_B$.*

For example, if

$$a >_{\theta_5} b >_{\theta_5} a' >_{\theta_5} b',$$

then $s_A \geq s_B$. The Pareto efficiency is not independent since we can prove in the following that team anonymity plus monotonicity imply the pareto principle.

PROPOSITION 1. *A choice function f satisfies the pareto principle if it is both team anonymous and monotonic.*

PROOF. Suppose otherwise, then there exists a state θ such that $\forall 0 \leq i \leq n$, the i-th ranked player a_i of team A is better ranked than that b_i of team B and $s_A < s_B$. Now we swap the role of a_i and b_i for all i in θ and we call the new state θ'. By team anonymity, we have $f(\theta') = f(s_B, s_A)$, which is a worse outcome than $f(\theta)$ for B. However, since θ' is an improvement to θ for team B, team B's outcome then should be no worse by monotonicity. This leads to a contradiction. □

3.2.2 Examples of Choice Functions

It is not difficult to see that the following four choice functions satisfy all the restrictions mentioned above.

- Borda Count: According to $>_\theta$ on $A \cup B$ with $|A \cup B| = 2n$, we assign the player that ranked first 2n-1 points, the second 2n-2 points,..., and the last 0 point. $s_A = \Sigma_{a_i \in A} point(a_i)$. s_B can be defined symmetrically. $f_{BC}(\theta) = (s_A, s_B)$.

- Pairwise comparison: Suppose θ_A and θ_B are $(a'_1 > a'_2 > , \ldots, > a'_n)$ and $(b'_1 > b'_2 >, \ldots, > b'_n)$. Then $s_A = |\{(a'_i, b'_i) | a'_i >_\theta b'_i\}|$. s_B can be defined symmetrically. $f_{PC}(\theta) = (s_A, s_B)$.

- Max: Suppose the best players of A and B by θ are a and b respectively, then $f_{Max}(\theta) = (1, 0)$ if $a >_\theta b$ and $f_{Max}(\theta) = (0, 1)$ otherwise.

- Min: Suppose the worst players of A and B by θ are a and b respectively, then $f_{Min}(\theta) = (1, 0)$ if $a >_\theta b$, otherwise $f_{Min}(\theta) = (0, 1)$.

In other words, f_{BC} sums the rankings of all the players in each team, f_{PC} sums all the winnings for the pairwise comparison between players in the "same level" while f_{Max} and f_{Min} care only about the best and worst players.

3.2.3 Dominant Strategy Truthful Implementation

DEFINITION 10. *A mechanism M dominant strategy truthfully implements a choice function f, if M is dominant strategy truthful and if both teams report truthfully, the resulting outcome coincides with the one prescribed by f.*

If a mechanism dominant strategy truthfully implements a choice function, then both teams will report truthfully. Moreover, the truthful reports will lead to the desirable outcome prescribed by f.

3.3 Conciseness

As mentioned before, for a static mechanism, if all entries of its scoring matrix are non-zero, there are potentially n^2 matches while for certain classes of dynamic mechanisms such as the knock-out, there are at most $2n - 1$ matches. In general, we hope that the number of matches is at most linear in the number of players.

DEFINITION 11. *(Linearity) A mechanism is linear if the number of matches is $O(n)$, where n is the number of players in each team.*

4. THE RESULTS

In this section, we present our answers to the question we asked earlier: what mechanisms implement the choice function with previously mentioned properties.

4.1 Implementation by Static Mechanisms

We say that a matrix $C_{n \times n}$ is *non-increasing* if $c_{i_1, j_1} \geq c_{i_2, j_2}$ whenever $i_1 \leq i_2$ and $j_1 \leq j_2$ hold simultaneously.

THEOREM 1.

1. *A static mechanism M is dominant strategy truthful iff its score matrix $C_{n \times n}$ is non-increasing.*

2. *If a static mechanism M dominant strategy truthfully implements a choice function f, then*

    - *f is player anonymous;*

    - *f is team anonymous iff the score matrix satisfies $C = C^T$, where C^T is the transposition of C.*

    - *f is monotonic iff the score matrix has no negative entry*

PROOF.

1. $\Rightarrow$: If M is dominant strategy truthful, without loss of generality, suppose there exist i, j such that $c_{i,j} < c_{i+1,j}$. Now consider such a state $\theta : b_1 > b_2 >, \ldots, b_{n-1} > a_1 > , \ldots, a_i, b_n, \ldots, a_n$. In other words, θ is a state where team A can win only i matches against the worst player b_n of B. Now if B reports b_n as its jth player, then if A reports honestly, he will get $c_{1,j}+, \ldots, +c_{i,j}$ while if A swaps a_i and a_{i+1}, A will get a better score $a_{1,j}+, \ldots, +c_{i-1,j} + c_{i+1,j}$, which contradicts the dominant strategy truthfulness of M.

$\Leftarrow$: If $C_{n \times n}$ is non-increasing, for any state θ and any $b \in B$ reported as jth player, suppose according to θ, we have $a_1 >, \ldots, a_i > b > a_{i+1} >, \ldots, > a_n$. If A reports honestly, it will get $c_{1,j}+, \ldots, +c_{i,j}$ from b, otherwise, it will get $c_{m_1,j}+, \ldots, +c_{m_i,j}$. Since C is non-increasing, we have $c_{1,j}, \ldots, c_{i,j}$ are the greatest i entries in column j of C, so $c_{1,j}+, \ldots, +c_{i,j} \geq c_{m_1,j}+, \ldots, +c_{m_i,j}$. Since j and θ are arbitrarily chosen, we have A is dominant strategy truthful.

2. This part follows immediately from the definitions.

$\square$

Obviously, the score matrix in neither horse racing nor table tennis competition is non-decreasing. According to theorem 1, they are not dominant strategy truthful. For instance in the table tennis example, if the state is as follows: $a_1 > b_1 > b_2 > a_2 > b_3 > a_3$. Note that if both A and B reported truthfully, B will lose the competition by 2:3. However, if B misreports his order as $b_1 >' b_3 >' b_2$ and A still reports truthfully , B will win the competition by 3:2.

THEOREM 2.

1. *The static mechanism dominant strategy truthfully implements f_{BC} if its score matrix $C_{n \times n} = 1_{n \times n}$, where $1_{n \times n}$ is the matrix with all the entries equal to 1;*[2]

2. *The static mechanism dominant strategy truthfully implements f_{Max} if its score matrix satisfies $c_{1,1} = 1$ and $c_{i,j} = 0$ otherwise;*

3. *There is no static mechanism dominant strategy truthfully implements either f_{PC} or f_{Min}.*

PROOF.

1. It is not hard to see that for $1_{n \times n}$, the mechanism simply counts the sum of the number of opponents that are weaker for each player. Moreover, it is non-decreasing. So it truthfully implements f_{BC} minus a constant $\frac{n(n-1)}{2}$ in dominant strategy. The constant stands for the sum of additional scores if they are allowed to play with their teammates.

2. This part follows directly from the definition

3. Suppose M with score matrix C truthfully implements f_{PC}, then suppose for $\theta : a_1 > b_1 > \ldots$, $f_{PC}(\theta) = (s_A, s_B)$, then we have for $\theta' : b_1 > a_1 > \ldots$, $f_{PC}(\theta') = (s_A - 1, s_B + 1)$. This can be achieved only if $c_{1,1} = 1$. Similarly, suppose for $\theta'' : \ldots, > a_n > b_n$, $f_{PC}(\theta'') = (s'_A, s'_B)$ then we have for $\theta''' : \ldots, > b_n > a_n$, $f_{PC}(\theta''') = (s'_A - 1, s'_B + 1)$. This can be achieved only of $c_{n,n} = 1$. Since M is dominant truthful, C is non-increasing, therefore C can only be $1_{n \times n}$. However, $1_{n \times n}$ obviously does not implement f_{PC}. A contradiction. Similar for f_{Min}.

$\square$

It is worth pointing out that the current international team competition for tennis (aka Davis Cup) uses the static mechanism whose score matrix is $1_{2 \times 2}$, therefore it implements Borda Count. However, the number of matches is n^2, although it does not make much difference with linear ones when $n = 2$.

[2]In fact, it implements Borda Count minus a constant $\frac{n(n-1)}{2}$.

4.2 Implementation by Dynamic Mechanisms

Unlike any static mechanism, whose preference is well-defined on a constant-sum set of outcomes, there are cases where the preferences on the set of outcomes are not well-defined for dynamic mechanisms. Therefore, it is meaningless to talk about the implementation issues for general dynamic mechanisms. Alternatively, we focus on a particular class of dynamic mechanism called "knock-in" in contrast to the "knock-out" mechanism mentioned in example 3. That is, after each match, the loser stays to compete with the winner's successor.

DEFINITION 12. *(Knock-in Competition) Upon receiving the reported lists of players $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$,*

- *a_1 vs b_1 is initiated as the first match;*

- *In the following rounds, if the current match is a_i vs b_j, then the next function will assign a_{i+1} vs b_j if a_i beats b_j and assign a_i vs b_{j+1} otherwise.*

- *The set of terminal histories are those where every player has won in one team, which we call the winning team. The other team is called the losing team.*

It follows from the definition immediately that

THEOREM 3.

1. *Every knock-in mechanism satisfies linearity.*

2. *By assigning 1 to the winning team and 0 to the losing team in each terminal history, the knock-in mechanism always leads to an outcome that coincides with the one predicated by f_{Min}, and therefore truthfully implements f_{Min} in dominant strategy.*

PROOF.

1. This part follows by observing that a winner is eliminated each match and never reappears. So a whole team is eliminated for at most $2n - 1$ matches.

2. Note that in a knock-in competition, the team with the better worst player is always the winning team, no matter what two team reports. So the knock-in mechanism trivially implements (no matter truthfully or not) f_{Min}.

$\square$

The flexibility of designing a knock-in mechanism lies in the choice of scoring rules. The scoring rules should be designed in a way such that, on one hand the preference for the set of terminal histories is well defined and on the other hand, the scores align with the incentives of reporting truth. We introduce in the following, two possible classes of such scoring rules.

4.2.1 Score By Play Order

With this type of scoring rule, we assign a constant score c_1 to the winner of the first match, a constant score c_2 to the second match, and so on. The score of the team and an outcome are defined as usual. Note that the number of matches in such a knock-in mechanism is a variable ranging from n to $2n - 1$. In order to make it a constant-sum competition, we uniformly assign $2n$ constants as follows:

DEFINITION 13. *In a score-by-play-order rule, we have a list of $2n$ constants $\{c_1, c_2, \ldots, c_{2n}\}$, we assign c_1 to the winner of the first match, c_2 to the winner of the second match and so on. When reaching a terminal history after n_0 matches, $n \leq n_0 \leq 2n - 1$, we assign the remaining constants $c_{n_0+1}, \ldots, c_{2n}$ to the remaining players (the order does not matter). Therefore, each player receives a score in a terminal history.*

In this way, we have $s_A + s_B = \Sigma_{1 \leq i \leq 2n} c_i$. The following theorem characterizes the dominant strategy truthfulness of knock-in mechanisms with this type of scoring rules.

THEOREM 4. *A knock-in mechanism M by a score-by-play-order rule*

- *is dominant strategy truthful iff $\{c_1, c_2, \ldots, c_{2n}\}$ is non-increasing.*

- *dominant strategy truthfully implements a choice function f, then*

    - *f is player anonymous, team anonymous, monotonic and pareto efficient.*

- *dominant strategy truthfully implements*

    - *f_{BC} with $\{2n - 1, 2n - 2, \ldots, 1, 0\}$.*

    - *f_{Max} with $\{1, 0, \ldots, 0\}$.*

PROOF. We prove the first claim and leave the rest to the readers.

$\Rightarrow$: If M is dominant strategy truthful and suppose $\{c_1, c_2, \ldots, c_{2n}\}$ is not non-increasing. Without loss of generality, let $c_m < c_{m+1}$. Suppose the mth match is between a_i and b_j. Now consider a state

$$\theta : a_i > b_j > a_{i+1} > b_{j+1}.$$

If A reports truthfully, the competition goes as follows:
a_i wins b_j, team A gets c_m;
a_{i+1} loses to b_j, team B gets c_{m+1};
a_{i+1} wins b_{j+1}, team A gets c_{m+1};
    If A swaps a_i and a_{i+1}, the competitions goes as follows:
a_{i+1} loses to b_j, team B gets c_m;
a_{i+1} wins b_{j+1}, team A gets c_{m+1};
a_i wins b_{j+1}, team A gets c_{m+1}.
Obviously by lying, team A gets a better outcome. A contradiction.
$\Leftarrow$: If $\{c_1, c_2, \ldots, c_{2n}\}$ is non-increasing, we prove by enumeration that if $a_i > a_{i+1}$, then it is always no worse to report $\ldots, a_i, a_{i+1}, \ldots$ than $\ldots, a_{i+1}, a_i, \ldots$. Suppose the current opponent is b_j followed by b_{j+1} and the score of the current match is c_m. There are the following cases:

- Case 1: $a_i > a_{i+1} > b_j > b_{j+1}$ or $a_i > a_{i+1} > b_{j+1} > b_j$. No matter reporting a_i, a_{i+1} or a_{i+1}, a_i will give team A the same outcome.

- Case 2: $a_i > b_j > a_{i+1} > b_{j+1}$. By reporting a_i, a_{i+1}, the competition goes as follows:
a_i wins b_j, team A gets c_m;
a_{i+1} loses to b_j, team B gets c_{m+1};
a_{i+1} wins b_{j+1}, team A gets c_{m+2};
while by reporting a_{i+1}, a_i, the competition goes as follows:
a_{i+1} loses to b_j, team B gets c_m;
a_{i+1} wins b_{j+1}, team A gets c_{m+1};
a_i wins b_{j+1}, team A gets c_{m+2};
Obviously truthful reporting is no worse for team A.

- Case 3: $a_i > b_{j+1} > a_{i+1} > b_j$. Now matter reporting a_i, a_{i+1} or a_{i+1}, a_i will give team A the same outcome.

- Case 4: $a_i > b_j > b_{j+1} > a_{i+1}$. By reporting a_i, a_{i+1}, the competition goes as follows:
a_i wins b_j, team A gets c_m;
a_{i+1} loses to b_j, team B gets c_{m+1};
a_{i+1} loses b_{j+1}, team B gets c_{m+2};
while by reporting a_{i+1}, a_i, the competition goes as follows:
a_{i+1} loses to b_j, team B gets c_m;
a_{i+1} loses b_{j+1}, team B gets c_{m+1};
Note that a_i is still in the game. Since $a_i > a_{i+1}$, a_i will win whoever a_{i+1} wins, getting a score c_{m+t}, $t > 0$. Since $c_m \geq c_{m+t}$. Truthful reporting is no worse for team A.

- Case 5: $a_i > b_{j+1} > b_j > a_{i+1}$. This case is equivalent to case 4.

- Case 6: $b_j > a_i > b_{j+1} > a_{i+1}$. By reporting a_i, a_{i+1}, the competition goes as follows:
a_i loses to b_j, team B gets c_m;
a_i wins b_{j+1}, team A gets c_{m+1};
a_{i+1} loses b_{j+1}, team B gets c_{m+2};
while by reporting a_{i+1}, a_i, the competition goes as follows:
a_{i+1} loses to b_j, team B gets c_m;
a_{i+1} loses b_{j+1}, team B gets c_{m+1};
a_i will win immediately after a_{i+1} wins and get $c_{m+t} \leq c_{m+1}$. Truthful reporting is no worse for team A.

- Case 7: $b_{j+1} > a_i > b_j > a_{i+1}$. By reporting a_i, a_{i+1}, the competition goes as follows:
a_i wins b_j, team A gets c_m;
a_{i+1} loses to b_j, team B gets c_{m+1};
a_{i+1} loses b_{j+1}, team B gets c_{m+2};
while by reporting a_{i+1}, a_i, the competition goes as follows:
a_{i+1} loses to b_j, team B gets c_m;
a_{i+1} loses b_{j+1}, team B gets c_{m+1};
a_i will win immediately after a_{i+1} wins and get $c_{m+t} \leq c_m$. Truthful report is no worse for team A.

- Case 8: $b_{j+1} > a_i > a_{i+1} > b_j$ or $b_j > a_i > a_{i+1} > b_{j+1}$. No matter reporting a_i, a_{i+1} or a_{i+1}, a_i will give team A the same outcome.

- Case 9: $b_{j+1} > b_j > a_i > a_{i+1}$ or $b_j > b_{j+1} > a_i > a_{i+1}$. No matter reporting a_i, a_{i+1} or a_{i+1}, a_i will give team A the same outcome.

Therefore, truthful reporting is always no worse than misreporting. $\square$

For example, to truthfully implement f_{BC}, we first let a_1 vs b_1 and winner gets $2n - 1$, the loser stays to compete with the next player of the other team, and so on. Since the earlier matches have higher scores and each player would get a score anyway, each team then would like to win as early as possible, so truthful report is no worse than misreport.

4.2.2 Score by Position

With this type of scoring rules, we assign the score to each match similarly to what we did in the static mechanism, except that each match is asymmetric: the score that a_i gets from winning a_i vs b_j may not be the same as b_j gets from winning the same match. Further, for the sake of symmetry, we require that the score that a_i gets from winning a_i vs b_j is the same as the score b_i gets from winning b_i vs a_j. Therefore, only one score matrix is needed for such a scoring rule.

DEFINITION 14. *In a score-by-position rule with a score matrix $C_{n \times n}$, for any match a_i vs b_j, a_i gets $c_{i,j}$ if he wins and b_j gets $c_{j,i}$ otherwise.*

Unfortunately, in general this type of mechanism is not constant-sum. However it can truthfully implements f_{PC} in dominant strategy with certain restrictions on C.

THEOREM 5.

- *A knock-in mechanism M with score-by-position rule is dominant strategy truthful if its score matrix satisfies $\forall 1 \leq i, j, i+1, j+1 \leq n$*

    1. $c_{i,j} \geq c_{i,j+1}$
    2. $c_{i,j} \geq c_{i+1,j+1}$

- *A knock-in mechanism M with score-by-position rule dominant strategy truthfully implements f_{PC} with $c_{i,j} = 1$ if $i \geq j$ and $c_{i,j} = 0$ otherwise.*

PROOF.

- The proof of the first claim is similar to that of theorem 4. We enumerate all the possible cases of every neighboring pair and show that truthful reporting is always no worse than misreport.

- First, if a mechanism with the score matrix C such that $c_{i,j} = 1$ if $i \geq j$ and $c_{i,j} = 0$ otherwise, then it satisfies

    1. $c_{i,j} \geq c_{i,j+1}$
    2. $c_{i,j} \geq c_{i+1,j+1}$

    So, according to the first claim, it is dominant truthful. The implementation of f_{PC} then follows from the fact that according to the matrix, each player can get 1 point iff it beats certain higher or equally ranked player.

□

In other words, to implement f_{PC}, we assign each player 1 point if he wins some higher ranked opponents while assign each player 0 point if he loses or win some lower ranked opponents. For example, if a_3 wins b_2, then a_3 gets 1 and b_2 gets 0, otherwise both a_3 and b_2 get 0. This indicates a higher ranked player can never score by competing with a lower ranked player, but he still has to win to leave the competition so that his lower ranked team mates can have a chance to score.

4.3 Summary

To sum up, we generalize dominant strategy truthfulness and truthful implementation by certain conditions for both static and dynamic mechanisms. We also prove that static mechanisms are able to implement f_{BC} and f_{Max} but fail to implement f_{PC} and f_{Min} while even a small amount of dynamic mechanisms called knock-in is sufficient to implement all the four choice functions. Furthermore, the static mechanisms are not guaranteed to be linear while the knock-in mechanisms, by eliminating one player each match, can never exceed $2n - 1$ matches.

5. STRATEGIC LOSING

The assumption so far is that once the state θ induces $a >_\theta b$, a will always beat b. This makes sense when the players are non-strategic individuals. For instance, each player is a number in a sealed envelope and the winner is the greater one.

However, sometimes the players can be strategic in the sense that the stronger one can lose a match deliberately if he wants to. Furthermore, the weaker one may also choose to lose if he predicts that the stronger one may do so. For example, if we assign a sufficiently large score to the second match in a knock-in mechanism, then each team's best player will play the first match as badly as he can in order to compete in the second match. Similar to the case where there are negative entries in the score matrix of a static mechanism.

The best way to prevent this from happening is to force "playing one's best" as a dominant strategy of a team. In other words, no matter what other team does, it is always no worse to play our best. However, we make no attempt to study this formally, mainly because when both players try to play badly, the result of the match is unclear.

As a compromise, we can force "playing one's best" as a Nash equilibrium strategy of each team. In other words, if the opponents always play their best, it is never any worse to do the same.

DEFINITION 15.

- *A strategy $S_p : \Theta_{A(B)} \rightarrow \{P, L\}$ of a player in a static mechanism is function that maps the private information to a action of either Play the best one's best or Lose deliberately.*

- *A strategy $S_p : \Theta_{A(B)} \times H_N \rightarrow \{P, L\}$ of a player in a dynamic mechanism is function that maps each state and a nonterminal history to a action.*

Suppose we have $a >_\theta b$, then if a plays P, a wins no matter what b plays; if b plays P, a wins if a plays P and b wins if a plays L.[3]

DEFINITION 16. *We say a mechanism is honest, if in each state, given all the players in the other team play P, it is also no worse for each player to play P.*

The following theorem generalizes the result of honesty in both static and dynamic mechanisms.

THEOREM 6.

- *A static mechanism is honest iff its score matrix C has no negative entry.*

- *A knock-in mechanism with score-by-play-order rule is honest iff $\{c_1, c_2, \ldots, c_{2n}\}$ is non-increasing.*

- *A knock-in mechanism M with score-by-position rule is honest if its score matrix satisfies $\forall 1 \leq i, j, i+1, j+1 \leq n$*

    1. $c_{i,j} \geq c_{i,j+1}$
    2. $c_{i,j} \geq c_{i+1,j+1}$

It is interesting that for knock-in mechanisms, the conditions that characterize honesty coincide with that of dominant strategy truthfulness. In other words, as long as these conditions are satisfied, the resulting mechanism is both truthful and honest.

6. OTHER ISSUES

In this section, we discuss other interesting issues that we do not cover in previous sections.

---

[3]Our result still holds when randomization is allowed, but we omit this here the definition of the possible outcomes when someone plays a mixed strategy.

6.1 Win-Lose-Tie

We have considered an outcome to be a pair of real numbers representing the scores that each team will receive at the end of the competition. One could argue that in many cases, what really matters is who won the game. In the two-team competition setting considered in this paper, this can be done by assuming three possible outcomes 1 (team A won), 0 (tie), and -1 (team A lost).

An interesting question then is how this will affect the results of this paper. First of all, we notice that instead of changing the set of outcomes, the same effect can be acheived by changing the preference relation P into the following ordering: A strictly prefers (s_A, s_B) over (s'_A, s'_B) iff either

$$s_A > s_B \text{ and } s'_A \leq s'_B$$

or

$$s_A = s_B \text{ and } s'_A < s'_B.$$

and is indifferent to (s_A, s_B) and $(s'_A, s'_B)^4$ iff either

$$s_A > s_B \text{ and } s'_A > s'_B$$

or

$$s_A = s_B \text{ and } s'_A = s'_B$$

or

$$s_A < s_B \text{ and } s'_A < s'_B$$

. It is similar for team B's preference.

One can verify that, if a condition, which is sufficient for dominant strategy truthfulness (or implementation of a choice function) previously, is still sufficient because the weak preference persists (o_1 is weakly preferred to o_2 before implies o_1 is weakly preferred to o_2 now). However, a previously necessary condition may not hold now. For example, one can verify that the static mechanism with the following score matrix $C_{2\times 2}$:

$$\begin{bmatrix} 9 & 10 \\ 10 & 0 \end{bmatrix}$$

is dominant strategy truthful although it is not non-increasing.

6.2 Uncertainty in a Match

It is sometimes reasonable to assume uncertainty in a match even if one player is better than the other. In such a model, the state is described by a matrix $P_{n\times n}$ where $p_{i,j}$ is the probability that a_i wins b_j. There are several variations of this model. Knuth [3] introduced a simplification of the model called knockout tournament by assuming a linear ordering $x_1, x_2, \ldots$ among players where x_i always beats x_j when $j \geq i+2$ and x_i beats x_j only at probability p when $j = i + 1$. Graham, et al. [1] introduced an alternative model by assuming x_i beats x_j with probability p for any i, j such that $i < j$. Our model can be seen as a special case of Graham's model where $p = 1$. It would be an interesting future work to generalize our state description to a random given matrix $P_{n\times n}$.

6.3 Profit Maximization

Note that we only require our mechanisms to satisfy certain basic criteria, based on which we can do certain optimizations. Suppose the organizer is not so interested in the conciseness of the mechanism as the profit that he can make from selling tickets. Further, his profit might be determined by the following factors:

- The number of matches;

---

[4] A is indifferent to o_1 and o_2 if A weakly prefers o_1 to o_2 and weakly prefers o_2 to o_1 at the same time.

- The level of matches: a match between a_1 and b_1 can sell more tickets than that of a_n and b_n;

- The truthfulness and honesty, so that the participants want to play and the audience want to watch.

The organizer's objective is to design a mechanism that (a) maximizes the total profit while (b) satisfies the fairness and honesty conditions. As long as the ticket information is given, it is not hard to see that the designing problem becomes a search problem over the space where truthfulness and honesty are satisfied.

7. CONCLUDING REMARKS

We now summarize our contributions in this paper as follows:

- We have formulated the team competition problem in the framework of mechanism design and proposed two types of mechanisms called static and dynamic mechanisms that characterize general forms of team competitions.

- We have put forward certain criteria such mechanisms should satisfy.

- We have come up with certain theorems that describe how these criteria can be satisfied for both types of mechanisms.

In particular, the knock-in mechanisms turn out to be extremely rich in implementing all these criteria,. However, there are still a large amount of dynamic mechanisms unexplored. Besides the issues mentioned in section 6, it will be worthwhile to probe into the unexplored part of dynamic mechanisms in the future.

8. REFERENCES

[1] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.

[2] L. Hurwicz. Optimality and informational efficiency in resource allocation processes. In K. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*, pages 27–46. Stanford University Press, Stanford, CA, 1960.

[3] D. E. Knuth. Problem 86-2, a random knockout tournament (with solution). 93:127–129, 1987.

[4] J.-J. Laffont and D. Martimort. *The Theory of Incentives : The Principal-Agent Model*. Princeton University Press, December 2001.

[5] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, June 1995.

[6] E. Maskin and T. Sjostrom. Implementation theory. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*.

[7] N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[8] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[9] T. Sandholm. Computing in mechanism design. The new palgrave dictionary of economics, second edition, 2008.

[10] Y. Shoham and K. Leyton-Brown. *Multiagent Systems. Algorithmic, Game theoretic and Logical Fundations*. Cambridge University Press, 2008.

[11] M. P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 28(4es):14–15, 1996.