

Coalition Control Through Forced Betrayal

Cameron Browne, Queensland University of Technology (QUT)

Multiplayer games often suffer from the problem of non-strategic coalitions between players. This short note explores the use of an explicit revenge rule to counteract this problem, and finds it wanting, whereas an explicit betrayal rule appears to offer more potential for reducing coalition effects. These results are demonstrated on two hypothetical games.

1 Introduction

GAMES with more than two players can have inherent problems due to *non-strategic coalitions*¹ that might exist between players outside the context of the game, even pure strategy games with no hidden information and no chance elements. Three-player games tend to be especially prone to such problems, as there is not an even number of players to form stable teams.

Two common problems are the *kingmaker effect* [1], in which a player with no hope of winning is able to decide which opponent will be the eventual winner, and the *petty diplomacy* problem [2], in which two players tend to form an *alliance* at the expense of the third player or *victim*. Strategic coalitions can be beneficial for some games, as a balancing mechanism in which trailing players cooperate against the leader to prolong the contest, but non-strategic coalitions are generally detrimental to three-player games.

1.1 McCarthy's Revenge Rule

McCarthy's revenge rule is a metarule intended to reduce the kingmaker effect in what Straffin calls *three-person winner-take-all games* [3]. It can be expressed as:

If you are prevented from winning by a double-crosser, try to take the double-crosser with you. [4, p. 159]

or:

If I find myself in a situation where I can no longer win but must choose which other player will win, I will look back to see who has put me in this undesirable situation and choose to make that player lose. [3, p. 390]

This rule was proposed by computer scientist John McCarthy while playing *So Long Sucker*,

'a game of negotiation, alliances and backstabbing' [5], with its co-inventors including John Nash. This rule makes intuitive sense, and has an impressive heritage – McCarthy was one of the founders of artificial intelligence and Nash famously one of the founders of game theory and modern economics – so why does this rule not feature in the rule sets of more multiplayer games?

I believe that this is due to two main reasons. First, it was posed as a principle for rational players with no external agenda beyond the game, and this ideal situation rarely occurs in practice. Straffin's subsequent analysis even states that it does not take coalitions between players into account [3]. Second, it could simply be too difficult to enforce, as it may not always be clear *which* opponent is most to blame for an unfavourable position.

This paper proposes a simple development of this idea, to encode the revenge principle explicitly in the rules, so that players are *forced* to follow this principle regardless of their personal agendas, as a way of reducing the effect of non-strategic coalitions in multiplayer games.

2 Forced Revenge

There is an inherent logical problem with applying McCarthy's revenge rule directly to a rule set: *if two players have formed a non-strategic coalition, then neither will want to hurt the other the most*. This could actually make the coalition problem *worse*, as such a revenge rule would force the allies to act even more explicitly against the victim, who is fighting them both for dear life.

For example, consider the following hypothetical game called *Revenge*, invented for this exercise. The degree to which players hurt each other is indicated by the number of pieces captured. In order to test this game's robustness to coalition effects, let us assume that White and Black are *allies* who have prearranged a non-strategic coalition against the *victim* Red.²

¹Coalitions between players that satisfy some personal agenda rather than being of strategic benefit to either.

²For black-and-white readers: White = light, Red = medium and Black = dark.