Trust is Risk: Generalized Max Flow for Strategies between Idle and Conservative

Orfeas Stefanos Thyfronitis Litos

University of Edinburgh
`o.thyfronitis@ed.ac.uk`

Abstract. Previous versions of Trust is Risk present the Conservative and the Idle strategy as distinct and unrelated. This work is an attempt to generalize this idea into a continuous spectrum of strategies, the two ends of which correspond to the two previously defined strategies. Prior to querying the system for an indirect trust towards Bob, Alice can attribute a specific expected strategy to each one of the participating players, or even fine-tune the response of each player to a steal action from each one of the players she directly trusts. The system then executes the generalized MaxFlow [1] to determine the indirect trust from Alice to Bob, given the specified strategies of the rest of the players.

The generalized MaxFlow algorithm expects as input a graph with capacities as well as a *gain factor* for each edge, which in our case will be a number in $[0, 1]$. Intuitively, the gain factor of an edge (v, w) represents the ratio of "leakage" this edge causes. In our case, it represents the percentage of funds that v will try to replenish when w steals from her. For the sake of example, consider the following graph:

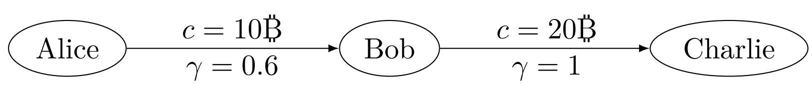

Fig. 1: Alice trusts Charlie 6₿

A gain factor of 0 means that v will tolerate any amount of stolen funds by w without trying to replenish them by stealing others that directly trust her, whereas a gain factor of 1 means that v will try to replenish any amount of stolen funds by w. If the gain factor is 0 on edges (v, w) for all $w \in \mathcal{V}$, then v is following the Idle strategy, whereas if the gain factor is 1 on edges (v, w) for all $w \in \mathcal{V}$, then v is following the Conservative strategy. The incoming direct trusts to the Evil player should all have a gain factor equal to 1, since we consider that she steals all her incoming direct trust.

References

1. Wayne K. D., Tardos E.: Generalized Maximum Flow Algorithms. Ph.D. thesis: Cornell University (1999)