A Coalitional Algorithm for Recursive Delegation

108

No Institute Given

Abstract. Within multi-agent systems, some agents may delegate tasks to other agents for execution. Recursive delegation designates situations where delegated tasks may, in turn, be delegated onwards. In unconstrained environments, recursive delegation policies based on quitting games are known to outperform policies based on multi-armed bandits. In this work, we incorporate allocation rules and rewarding schemes when considering recursive delegation, and reinterpret the quitting-game approach in terms of coalitions, employing the Shapley and Myerson values to guide delegation decisions. We empirically evaluate our extensions and demonstrate that they outperform the traditional multi-armed bandit based approach, while offering a resource efficient alternative to the quitting-game heuristic.

1 Introduction

Delegation within multi-agent systems involves a *delegator* handing over a task to a *delegatee*. While a single delegation event is often considered in works dealing with trust, [4, 2], we address situations where agents are allowed to pass the task onwards until it is eventually executed —a process termed *recursive delegation*. In [1], it has be shown that existing trust mechanisms can be improved within such recursive settings through a game theoretic treatment of the problem. Here, we extend the basic recursive delegation scenario to include an explicit reward rule associated with successful delegation, subject to an equally explicit resource constraint.

To exemplify the applications our approach may capture, consider a distributed network composed of heterogeneous sensors with distinct capabilities [7, 5]. These sensors can repeatedly delegate a task across the network, but must do so mindful of their energy consumption (and timeliness of response), as well as the quality of the information returned (with the latter serving as a reward in this context). Upon receiving a task, a sensor must decide whether to delegate the task onwards or execute it (by sensing), attentive to the constraints and rewards attached to its decision.

Non-cooperative games in the form of quitting games have already been applied to the study of recursive delegation [1]. Compared to nested multi-armed bandits, the former display greater efficiency, producing higher probabilities of successful delegation with lower levels of regret [1]. These techniques, however, do not take explicit resource constraints and rewards into account, whereas in our work, we not only introduce such additional aspects, but also formulate a coalitional alternative to non-cooperative decision making in recursive delegation domains.

The remainder of this paper is structured as follows. In the next section we describe the non-cooperative approach to recursive delegation. In Section 3, we present

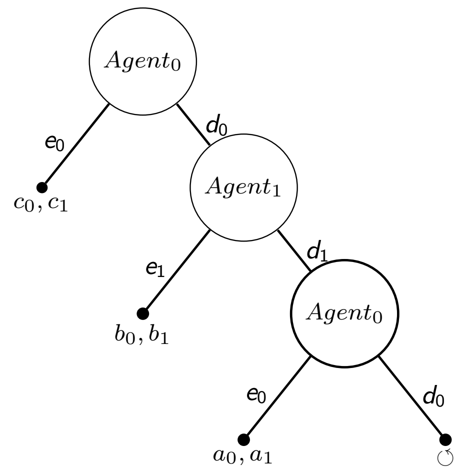


Fig. 1: Quitting Game in Extensive Form

our implementation of the Shapley and Myerson values as coalitional algorithms for recursive delegation. Section 4 empirically compares the different approaches, Section 5 discusses our results alongside directions for future work, and Section 6 gathers our main conclusions.

2 Recursive Delegation as a Quitting Game

Adversarial techniques to reason about recursive delegation are built on an adaptation of quitting games; a class of stochastic games [15]. Players of a quitting game have two available actions, either choosing to continue the game (action d), or quitting the game altogether (action e). The former action allows the game to repeat, while the other brings the game to an end. After each game, all players receive whatever rewards they have earned.

A two-player game in extensive form is illustrated in Figure 1. Here, either player selecting action e leads to the realisation of their respective rewards. Both players playing d leads to the continuation of the same game (denoted by $\circlearrowleft$).

In an n-player game, the *strategy* of player i is a probability measure $x_{i(t)} : \mathbb{R}_0^+ \to [0, 1]$ representing the likelihood of playing d at iteration t. A profile or vector of strategies $\mathbf{x}_t$, would then produce a stream of rewards r_{S_t}, contributed by the subset of players S who have chosen not to quit the game by iteration t. The expected reward of player i at iteration t thus becomes $w_{i(t)}(\mathbf{x}_t) := \mathbb{E}_{\mathbf{x}}[r_{S_t}]$. Let us note in passing that the subscript $i(t) := i \circ t : \{0, \ldots, T-1\} \to \{0, \ldots, n-1\}$ indicates the value of a variable associated with player i at iteration t, and that it is attached to said variable whenever the index's omission, or its simplification, seems ambiguous.

For $\epsilon \ll b_0/c_0$, $a_0 > 0$, $a_1 < c_1$, $c_0 < b_0$, $a_1 \geq b_1$ and $x_0 \ll 1$ the stationary profile $\mathbf{z} \equiv \langle x_0, d_1 \rangle$ —where $Agent_0$ delegates the task with very low probability, while $Agent_1$ systematically chooses to delegate— is produced [16]. That is, the expected reward of **z** plus an overhead $\epsilon > 0$, is at least that of any other strategy $y_{i(t)}$ for every player i, or equivalently $w_{i(t)}(\mathbf{z}) \geq w_{i(t)}(\mathbf{x}_{-i(t)}, y_{i(t)}) - \epsilon$. Thus, the profile **z** describes an ϵ-equilibrium [9].

Quitting games share many facets of recursive delegation, effectively capturing self-embedded instances of strategic interaction which resemble the replication of delegation requests along a *delegation chain*, i.e., the sequence of delegatees who receive delegation requests involving the same task. Unlike a standard quitting game, however, delegation requires distinct strategic scenarios, where players alternate between the delegator and delegatee roles. The adjustment to this scenario is conducive to the definition of a *Delegation Game* [1].

Definition 1 (Delegation Game). *The tuple $\Gamma_d = \langle N, A, (u_i, r_i)_{i \in N}, \boldsymbol{x} \rangle$ encodes a delegation game among $|N|$ players, where every player has the following attributes:*

Actions: $A := \{d, e\}$ *and* $A_i = A, \forall i \in N$. $\Delta(A)$ *is the collection of all probability distributions over the set of available actions.*

Rewards: $r_i : \times_{j \in D \subset N} \Delta(A_j) \rightarrow \mathbb{R}, \forall i \in N$ *is a Lebesgue measurable function representing the gains of player i when a group of agents $D \subset N$ have been delegated to.*

Strategy: $x_i : A_i \rightarrow [0, 1], \forall i \in N$ *is the probability of player i playing action d.*

Profile: $\boldsymbol{x}_t := \langle x_{i(t)} \rangle_{i \in N}$. *Profiles induce a probability distribution $\boldsymbol{P_x} \in \Delta(A)$ over the set of actions, which permits the computation of the expected rewards $w_{i(t)}(\boldsymbol{x}_t) := \mathbb{E}_{\boldsymbol{x}}[r_{i(t)}]$.*

Updating Rule: $u_{i(t)} : \times_{j \in D_{t-1}} A_j \times \mathbb{R} \rightarrow \Delta(A)$ *is a measurable set-valued function that dictates the transition from one state of the system to a potentially different profile.*

When rewards are subject to a stochastic process, the selection of an action has to be expressed in terms of strategic profiles ($\mathbf{x}_t$). The probability distribution these profiles induce is then used to calculate the expected rewards ($w_{i(t)}$). By contrasting expected rewards in the manner of an ϵ-equilibrium, delegators and delegatees select their strategies, which once played provoke the respective information states to update ($u_{i(t)}$).

The entire delegation and learning process based on delegation games, is captured by the DIG algorithm presented in [1]. As may be apparent, neither quitting games nor the algorithm take explicit account of the costs associated with exploration or the rewarding mechanism motivating the decision to delegate; we introduce these considerations in the next section.

3 Recursive Delegation as a Coalitional Game

Our approach construes delegation as a recursive and collective process where delegatees form coalitions by playing a delegate action, in accordance with their individual capacity to generate and retrieve value amid restrictions and incentives that condition such capacity. We proceed to describe how coalitions are formed, and state the allocation and distribution rules devised to reflect the delegation structure contained in Definition 1. To illustrate our ideas, let us revisit the opening example on sensor networks.

For the efficient design of one such network, the main aspects typically considered are 1) the features of the sensors as mobile nodes; 2) the limitations these nodes may face in terms of energy consumption, memory size to buffer data, or wireless transmission capacity [17]; and 3) the metrics used to assess the impact of their individual contributions on the overall data-gathering performance of the system [12]. We account for 1) by introducing explicit value allocation rules stating each node's potential to generate sensing data. The idea being that in, e.g., event-driven applications, nodes near active locations may have higher sensing rates, thus inducing delegation.

Constraints in the form of fixed amounts of a productive resource enabling delegation —electrical energy, most notably— reflect 2). Although wireless charging allows nodes to transmit energy across the network, thereby internalising these budgetary restrictions into the functioning of the sensors themselves [8], we opt to deal with resource constraints as extrinsic to the system. The reason for this is that, in a single-task environment, indefinite delegation is undesirable, and self-sustainability in regards to the productive resource becomes subsidiary. In multi-objective applications, however, these considerations might be relevant, as multiple tasks may compete for the same productive resources involved in delegation.

The criteria used to model the selection of delegatees respond to 3). As presented in Section 2, mixed strategies serve as metrics to compute ϵ-equilibria for the quitting-game approach. Alternatively, as shown in [1], the largest Gittins Index can be used to select a suitable delegatee in multi-armed bandit (MAB) models. As will be introduced in Section 3.2, the Shapley and Myerson values serve the same purpose in our coalitional game. We now proceed to outline the design features associated with aspects 1) and 2).

3.1 Delegation and Allocation Rules

Given a set of allocation rules, resource constraints and the definition of a solution concept, we present a general framework for reasoning about delegation under these conditions. To do so, consider the tree in Figure 2 which describes a delegation network of agents $N \equiv \{a, b, c, f, g, \ldots, m\}$ whose decisions consume a limiting resource C.

To capture task execution, we introduce *dummy agents* into our representation of the network. These dummy agents appear as solid unlabeled nodes in Figure 2. A task reaching a dummy agent must be executed (as it cannot be further delegated), and is recorded as carried out by the agent who generates the delegation request.

Consider agent a, the originator of the delegation process, also termed the *root*. This agent can play e_a and perform the task itself, i.e., delegate to the dummy agent. It can also delegate the task to b, in which case b might accept the task by playing e_b, or reject it by playing d_b, thus returning the task to a and forcing a to perform the task itself (via action e_a). Alternatively, a could delegate to c, whence the task may reach f who could, in turn, proceed as b. The task may also be further delegated to g who, had decided not to play e_g, could pass on the task via d_f until a terminal node appears, some other node plays an execute action, or the constraints are no longer satisfied. In this context, *coalitions*, i.e., groups of players treated as strategic units, amount to delegation chains; we refer to this process of formation of coalitions as the *quitting structure* of the game.

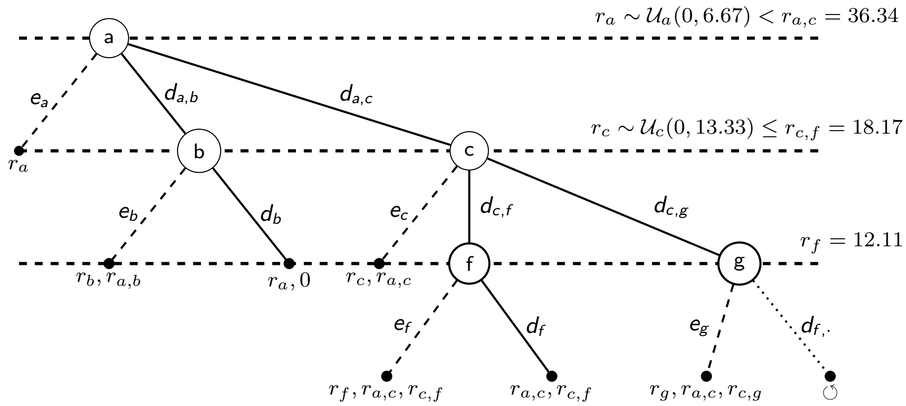


Fig. 2: Delegation Game in Extensive Form

Now let us turn to the nature of the rewards underpinning the assessment of a decision's profitability. The delegation game in [1] did not make direct reference to the way in which rewards were formed. Agents reached out to one another unconcerned with allocation and distribution rules, inasmuch as the only interactions affecting the calculation of their expected rewards were those with their immediate neighbours. Our proposal, on the contrary, is said to be coalitional because agents acknowledge the contributions of all delegatees in the same delegation chain. We, therefore, assign a (global) value V for playing the game, and introduce extrinsic rules for its allocation and the distribution of rewards emanating from it.

To see this, let V be the largest value delegatees are capable of achieving as terminal nodes of a delegation chain. Since globally known from the beginning, the value of the game is initially apportioned among delegators and potential delegatees following a directly proportional distribution rule. The further away from the root, the larger the value an agent can generate, thus incentivising delegation. In the sensors case, this accounts for flexible and diverse architectures of sparsely distributed nodes with high sensing rates, other nodes with lower rates but capable of picking up data from the former sensors while roaming the network, and yet another group of sensor nodes acting as data centres or base stations (c.f., [18, 12]).

In contrast, the distribution of the final outcome of delegation, that is, the actual set of rewards, obeys an inversely proportional rule. The closer to the root, the larger the share of the game's value, implying that the task is more profitable the sooner it is executed. In terms of the sensor network, this rule reflects problems of data latency and long delivery delay. If the time elapsed between data being buffered and uploaded to base stations is too long, it might be preferable to generate a greater number of delegation queries to proximate sensors [17].

To detail our approach, we now provide an example of the operation of these two rules which substantiates our approach *w.l.o.g.* Let us, first, designate the initial allocations of V over all $n \equiv |N|$ agents by $\{v_i\}_{i \in N}$. These values are realised by the terminal node of any delegation chain as *outcomes* $\{o_i\}_{i \in N}$ —bear in mind that delegation chains may contain a single player, or, equivalently, every agent is a potential task-executioner. The rewards $\{r_i\}_{i\in N}$ accrued to the members of any chain are obtained from the outcome associated with the chain's terminal node. That is, each player's potential to produce the value of the game is conveyed through their respective outcomes, which propagate across the delegation chain in the form of rewards. The following steps illustrate the calculation of rewards in the subgraph spanned by $\{a, b, c, f, g\}$, up to a hypothetical third level of delegation for a game with value $V = 100$:

1. Distribute V proportionately among all agents, depending on their position along the tree:
$$\frac{v_a}{1} = \frac{v_b}{2} = \cdots = \frac{v_g}{5} = \frac{V}{1+2+\cdots+5},$$
$$\text{i.e.,}\quad v_a = 6.67, v_b = 13.33, v_c = 20,$$
$$v_f = 26.67, v_g = 33.33$$
2. Sample the outcomes from a uniform distribution between v_i (computed above) and V:
$$o_i \sim \mathcal{U}_i(v_i, V), \forall i \in \{a, b, c, f, g\},$$
$$\text{e.g.,}\quad o_a = 57.89, o_b = 75.31, o_c = 68.22,$$
$$o_f = 66.63, o_g = 80.77$$
3. Once a potential coalition/delegation chain forms, e.g., $\{a, c, f\}$, distribute the outcome yielded by the agent executing the task in an inversely proportional manner:
$$\frac{r_{a,c}}{1} = \frac{r_{c,f}}{1/2} = \frac{r_f}{1/3} = \frac{o_f}{1+1/2+1/3},$$
$$\text{i.e.,}\quad r_a = 36.34, r_c = 18.17, r_f = 12.11$$

More generally, our proportional rule implies that the value of the game is allocated according to the relation $\frac{v_i}{i+1} = \frac{V}{T_n}$, where $T_k := \sum_{j=1}^{k} j = \frac{k(k+1)}{2}$. The inversely proportional rule requires individual rewards to satisfy $\frac{r_{i,j}}{i} = \frac{o_k}{H_k}$, where $H_k := \sum_{i=1}^{k} \frac{1}{i}$ and $r_i \equiv r_{i,j}$ for every $i = j \in N$. Insofar as these two rules depict the structure of incentives behind delegation, they will frame the evaluation of our coalitional algorithm against the corresponding benchmarks; namely, the original quitting-game based approach in [1], and the MAB model also presented in [1] which extends the numerical approximation to the Gittins Index introduced in [3].

Thus, aspect 1) is encapsulated in the interplay of equations $\frac{v_i}{i+1} = \frac{V}{T_k}$ and $\frac{r_{i,j}}{i} = \frac{o_k}{H_k}$, i.e., the value allocation and reward generation rules, respectively. Aspect 2), for its part, is incorporated into our framework via the explicit recognition of the value of the game V, and the straightforward imposition of a numerical parameter $K \in \mathbb{R}_0^+$ constraining the generation of delegation requests and the production of rewards out of V. Having established the relational characteristics of the agents in our delegation networks, and the rules or conditions that mediate their interactions —as per design aspects 1) and 2) outlined at the outset of this section— we go on to present the criterion and computational procedures delegators use to select a delegatee among its neighbours —thus reflecting aspect 3).

3.2 Recursive Delegation as a Coalitional Game

The initial allocation of values and the definition of rewards may circumscribe the delegator's decision, but the guiding principle behind delegation is given by the solution concept used to select one or another delegatee. We employ the Shapely and Myerson values to this effect. Computing these values allows us to map potential rewards to groups of agents, so the advantages of forming a particular delegation chain can be assessed.

In a coalitional setting, potential delegation chains are treated as coalitions, i.e., groups of agents who evaluate the collective aspect of task completion/delegation. In spite of the individual nature of the rewarding scheme, the completion of the task is considered a common objective, and the network-wide impact of the resources expended in achieving the task is acknowledged by delegators. Hence, in the form of neighbouring conditions, preexisting valuations of available coalitions, and an internal mechanism for extracting individual contributions, these elements provide the basis of a delegation game of coalitions (DEC):

Definition 2 (Delegation Game of Coalitions). *A Delegation Game of Coalitions is a tuple $\varGamma_c = \langle N, V; \mathcal{B}, \nu \rangle$, characterised by the following elements:*

Value of the game: $V \in \mathbb{R}_0^+$, *gives the maximum value delegation can yield.*
Coalitional Structure: *A partition $\mathcal{B}$ of the set of agents $N = \{1, \ldots, n\}$ conforming to the quitting structure of delegation .*
Outcomes: $o_i : \varDelta(\{w_i\}_{i \in N}) \to \mathbb{R}_0^+$ *for every $i \in N$, are obtained from the stochastic process dictating the allocation of the value of the game. $\varDelta(\{w_i\}_{i \in N})$ is the collection of potential distributions over the set of admissible distributions of V.*
Characteristic Function: $\nu : 2^n \to \mathbb{R}^+$, *associates every coalition $D \subset \mathcal{B}$ with the expected value of its aggregated reward, i.e., $\nu(D) = \sum_{i \in D} \mathbb{E}[o_i]$.*

The coalitional structure of DEC encompasses those combinations of agents compatible with the quitting structure of delegation described in 3.1. The characteristic function links the expected rewards to the corresponding coalition(s) in the set of all permutations of agents, mapping invalid ones (e.g., those where a delegator comes last) to zero. The rewarding rule $\psi : \{o_i\}_{i \in D \subset \mathcal{B}} \to \mathbb{R}_0^+$, assigns rewards to the members of coalitions D belonging to the partition of the game $\mathcal{B}$. In the abstract, the solution concept is but a mapping $\phi : U \to \mathbb{R}^n$ with $U := \{\varGamma_c : n \subseteq \mathbb{R}^+\}$, while in our experiments it takes the form of the Shapley and the Myerson values.

Definition 3 (Shapley Value [13]). *The Shapley Value of a coalitional game $\varGamma = \langle N; \nu \rangle$ –such as DEC– is a solution concept that retrieves the individual contribution of any player, subject to the coalitional structure of the game given by all subsets $D \subseteq N$. It can be computed as follows for every player $i \in N$.*

$$Sh_i(N; \nu) := \sum_{D \subseteq N} g_D[\nu(D) - \nu(D \backslash \{i\})]; \; g_D := \frac{(|D| - 1)!(n - |D|)!}{n!}. \quad (1)$$

That is, players foreign to a coalition D can be arranged in as many as $(n - |D|)!$ ways. In turn, within D all those players different from player i can be sorted in

$(|D|-1)!$ ways. The contribution of player i to the coalition is given by the difference between the aggregated value of D and that of the subsets (coalitions) excluding player i. The total number of such subsets amounts to $(|D|-1)!(n-|D|)!$. To obtain the corresponding average contribution, the sum over all possible coalitions is divided by the number of all admissible combinations of players, i.e., $n!$.

The Myerson Value is a refinement of the Shapley Value. The Myerson Value exclusively targets graph-restricted games, i.e., coalitional games whose coalitions can only reflect specific subgraphs of the underlying general graph of interactions [10]. The idea being that coalitions are highly dependent on their context. This means that the characteristic function should only be defined over connected components $\mathcal{S}(N)$, as given by the topology of the network enabling delegation. Connectedness, in this sense, refers to the existence of a path connecting any pair of non-adjacent nodes, such as a and g in Figure 2.

Definition 4 (The Myerson Value [10]). *Let $\Gamma = \langle N, \nu \rangle$ be a coalitional game. The Myerson Value (My_i) of Γ, corresponds to the Shapley Value for the characteristic function defined over connected coalitions i.e., $My_i(N;\nu) = Sh_i(N;\nu_M)$ such that*

$$\nu_M(D) = \begin{cases} \nu(D) & if D \in \mathcal{S}(N) \\ \sum_{K_i \in K(D)} \nu(K_i) & otherwise \end{cases}$$

The Shapely Value provides a means of differentiating individual contributions to a delegation chain within multi-agent systems, while incorporating the quitting structure of delegation outlined in Section 3.1. The Myerson Value implements the same procedure over a subset of players which not only conform to the quitting structure, but also respond to a particular configuration of the system laid out before the first delegation request had been issued.

Algorithm 1 Coalition Formation

Input: i: Index of the agent seeking coalitions, $path$: Length of the last delegation chain.
Output: $coalition$: Sequence of agents receiving a delegation request.

function CFORM(i)
  $k \leftarrow i.delegatee$
  $coalition \leftarrow \{j, k\}$
  $max\_length \leftarrow \mathcal{U}(2, 3)$
  $path\_length \leftarrow len(coalition)$
  while $path\_length < max\_length$ **do**
    if $k.out\_neighbours \neq \emptyset$ **then**
      $m \leftarrow sample(k.out\_neighbours)$
      $coalition \leftarrow coalition \cup \{m\}$
      $k \leftarrow m$
      $path\_length \leftarrow len(coalition)$
  return $coalition$

Our implementation of the Shapley and Myerson values requires a procedure to obtain the quitting structure of the game. Such procedure is given by Algorithm 1. It stipulates the formation of coalitions as a retrospective endeavour which looks into past delegation chains, permitting agents to recursively select new coalition members among their neighbours' neighbours (lines 6-11). Every agent foresees a coalition/delegation chain of length at most three (line 4); that is, itself, its immediate neighbour, and its neighbour's neighbour, intending to reflect myopic behaviour on the part of delegators.

Algorithm 2 implements DEC with the Myerson Value. Its Shapley version would only see the solution concept changed to Equation (1) (in line 4 of the function *DEL*). The computation of both the Myerson and Shapley values follow the divide-and-conquer approach of [14], which performs a recursive backtrack in a depth-first search for the delegation chains rooted at delegator j.

Our algorithm strives to find the largest contributions among all the delegation chains allowed by the quitting structure, subject to a resource constraint (line 3) and the allocation rules introduced before. Its inputs correspond to said resource constraint (K), the value of the game (V), and the set of probabilities of successful execution describing each delegatee's ability to perform the delegated task ($\{s_i\}_{i\in[n]}$).

As our algorithm requires the initialisation of individual outcomes (o_i), rewards (r_i) and neighbourhoods ($P_i \equiv \{a_i, ad_i\}$, where ad_i represents the neighbours of agent a_i), we have grouped those procedures under *Init_DEC*. After intialising counters of successful and failed execution (line 5), as well as the sets containing potential coalitions and actual delegatees (line 6), we apply the value allocation rule in line 7 to every agent in the system, followed by the sampling of outcomes as indicated in our opening example (line 8), so the distribution rule in line 9 enables the initialisation of the rewards on the basis of each agent's outcome.

We enter the main procedure *DEC* through a "while" statement at line 3. This statement guarantees that the game is played for as long as there is available productive resource K to effect a delegation request. Delegators employ the function *DEL* to allocate the delegation request. First, they seek a fitting coalition of three players at the most, by invoking the function *CForm* in line 2. Then, delegators compute the Myerson value of the resulting coalition (line 4), and proceed to select the delegatee who makes the largest contribution (line 5). If the selected delegatee is not its dummy agent, the delegation request is replicated (line 8) and the rewards obtained via our distribution rule in line 9.

We leave our core function at line 6, where the probability of successful execution of the selected delegatee (a_m) is contrasted against the state of nature as given by the probability $1 - \delta$. Not unlike the Delegation Game of [1], in our algorithm a favourable state of nature secures the execution of the task by the appointed delegatee, otherwise defaulting to the delegator itself; triggering the α and β counters as well as those keeping track of the fraction of the productive resource consumed throughout delegation, which is equal to the ratio between the number of successful interactions and the total number of visits to the chosen delegatee (lines 8 and 11 resp.). Past this stage, the outcomes are once again sampled (line 12), and the characteristic function of the game is learned (line 14). This process repeats until the limiting resource is depleted.

Algorithm 2 Delegation Game of Coalitions Under Myerson

Input: V: Value of the game, K: A real number denoting the resource constraint, s_i: Probability of successful execution of agent i.

Output: S: Sequence of agents receiving a delegation request. ν: Set of values of the characteristic function.

function INIT_DEC(K, V)
  $\nu \leftarrow \emptyset$
  $Constraint \leftarrow K$, $Consumption \leftarrow 0$
  for $j = 1 \rightarrow n$ **do**
    $\alpha_j \leftarrow 0, \beta_j \leftarrow 0$
    $D_j \leftarrow \emptyset, S_j \leftarrow \emptyset$
    $v_j \leftarrow (j+1)V/\sum_{i\in[n]} i$
    $o_j \leftarrow \mathcal{U}(0, v_j)$
    $r_j \leftarrow jo_j/\sum_{i\in[n]} 1/i$
    $P_j \leftarrow \{a_j, ad_j\}$

---

function DEL(P_k)
  $coalition \leftarrow$ CFORM(k)
  $D_k \leftarrow D_k \cup \{coalition\}$
  $my_k \leftarrow My_k(|D_k|; \sum_{i\in D_k} r_i)$
  $m \leftarrow \text{argmax}_{i\in ad_k}(my_i)$
  $S_k \leftarrow S_k \cup \{a_m\}$
  if $m \neq k$ **then**
    return DEL(P_m,s_m)
    $r_m \leftarrow ko_m/\sum_{i\in coalition} 1/i$
  else
    $r_m \leftarrow o_m$
  return (m, r_m, S_k)

---

procedure DEC($K, V; \{s_k\}_{k\in[n]}$)
  INIT_DEC(K, V)
  while $Constraint \geq Consumption$ **do**
    for $j = 1 \rightarrow n$ **do**
      $(m, r_m, S_j) \leftarrow$ DEL(P_j,s_j)
      if $s_m > 1 - \delta$ **then**
        $\alpha_j \leftarrow \alpha_j + 1$
        $Consumption \leftarrow Consumption + \frac{1}{\alpha_m+\beta_m}$
      else
        $\beta_j \leftarrow \beta_j + 1$
        $Consumption \leftarrow Consumption + \frac{1}{\alpha_j+\beta_j}$
      Update outcomes
      $S \leftarrow S_j \cup \{S_j\}$
      $\nu \leftarrow \nu \cup \{\sum_{k\in D_j} r_k\}$
    $Constraint \leftarrow Constraint - Consumption$
  return (S, ν)

4 Evaluation

4.1 Experimental Set-up

Our objective remains establishing whether the coalitional approach of Algorithm 2 can outperform DIG given the new constraints and rules. The evaluation of Algorithm 2 was carried out over Random Networks and Directed Trees extending up to 4 levels of delegation, with a branching factor of 5 neighbours per delegator among a population of 156 agents; as such is the number of nodes in a tree-like layout including its root. The levels of the limiting resource were allowed to range within 500 and 800 units, whereas the value of the game varied from 800 to 1000 units. Our algorithm and contrasting benchmarks were tested for the span of 100 runs, elapsing 1000 trials.

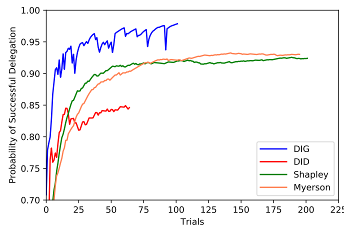
The systems under consideration are made up of agents arranged in either 4-level trees rooted at the first delegator in the network, or ad-hoc graphs whose edges are generated as delegation progress; their respective dynamics are dictated by the algorithms used to make delegation decisions. Directed Trees offer a structured environment for accommodating agents who establish a relation of precedence upon delegating. Random Networks, instead, are discovered as agents delegate —the probability of delegating arising form each algorithm simultaneously dictates the probability of spanning an edge from a delegator to a delegatee. The benchmarks used to compare our approach include the DIG algorithm in [1] and the adaptation of the Gittins Index also proposed in [1], but originally formulated in [3]. This selection circumscribes multi-armed bandits and non-coalitional game theory models whenever recursive delegation takes place in constrained environments.

4.2 Results

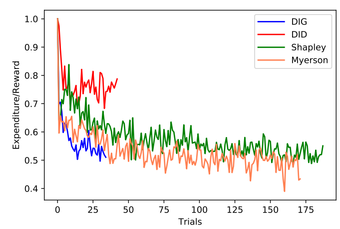
Figure 3 depicts the behaviour of the probabilities of successful delegation (PSD) and the ratio between the amount of productive resource expended in delegating and the value of the game generated through delegation (E/R). These two variables define our criteria of performance. The curves they describe stop at different trials due to the resource constraints faced by all agents and the ways in which the algorithms make use of it. Every delegate action consumes a productive resource; when this budget is depleted, delegators cannot delegate the task onwards. That is, the delegation process effectively comes to an end; a situation which coalitional games had to face at a much later point in time than its benchmarking algorithms.

In Random Networks and Directed Trees, DIG displays superior performance compared to the MAB approach (DID) and the coalitional alternatives. It attains larger rewards and higher probabilities of successful delegation. The great variability of this result, however, casts doubts on the efficiency of DIG. Directed Trees provide a structured environment for all algorithms to explore. In situations like this, previous knowledge of their neighbours' connectivity allows agents to expend less resources while exploring potential delegation chains. We find that the limiting resource not only lasts longer but leads to more stable delegation chains where tasks are more likely to be successfully executed (Figure 3 a).

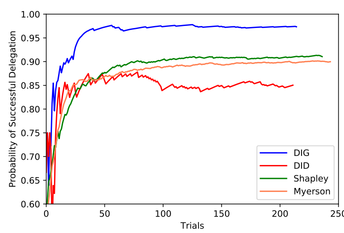
Figures 3a and 3b indicate that Myerson does not perform remarkably well on tree-like structures, despite being designed to better cope with fixed delegation patterns. It appears that in the early stages of delegation ($T < 50$ for Figure 3a) productive but costly coalitions were formed, which on account of the functioning of the algorithm would stifle exploration and trap delegators in chains with relatively poor capacity to adapt to delegation under tightening resource constraints.



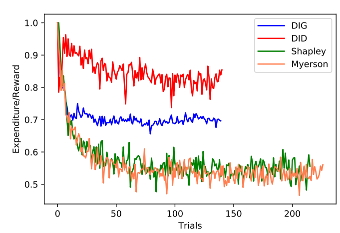
a: Probability of Successful Delegation over Random Networks



b: Expenditure-Reward Ratio over Random Networks



c: Probability of Successful Delegation over Directed Trees



d: Expenditure-Reward Ratio over Directed Trees

Fig. 3: Comparative Performance Between Topologies

Myerson, Shapley and DIG make use of the limiting resource until roughly the same trial. The difference being that DIG succeeds in generating at least one extra quarter of the value attained by the best performing coalitional algorithm (Shapley). A difference further reflected in the levels of regret associated with these results (Table 1).

Random Networks, on the other hand, allow agents to select their own neighbours, and potential delegatees, based on an intrinsic property, i.e., the strategies and distribution of the Gittins Index for DIG and DID, respectively; or an external one as in the cases of Shapley and Myerson. Under these conditions agents rely more heavily on exploration, often incurring in greater costs, particularly for DIG. Delegators employing DIG guarantee a higher PSD at the expense of lesser rewards, which also implies a lower regret (Table 1).

Only the coalitional algorithms maintain the behaviour displayed over Directed Trees. There is a considerable improvement in their levels of (cumulative) regret which

Group	Network Structure	PSD	Rewards	Regret
Directed Trees	DIG	0.92 ± 0.008	435 ± 0.19	$4.398 \pm 5e^{-5}$
	DID	0.827 ± 0.012	377 ± 0.62	$12.996 \pm 2e^{-5}$
	Shapley	0.889 ± 0.010	355 ± 0.33	$10.361 \pm 3e^{-5}$
	Myerson	0.858 ± 0.011	358 ± 0.04	$10.27 \pm 3e^{-5}$
Random Networks	DIG	0.966 ± 0.006	387 ± 0.42	$2.971 \pm 2e^{-4}$
	DID	0.794 ± 0.014	219 ± 0.69	$8.810 \pm 1e^{-4}$
	Shapley	0.890 ± 0.010	330 ± 0.41	$8.975 \pm 3e^{-4}$
	Myerson	0.889 ± 0.010	329 ± 0.31	$9.702 \pm 2e^{-4}$

Table 1: Minimum Credible Intervals of the Mean Posterior PSD, Reward and Regret

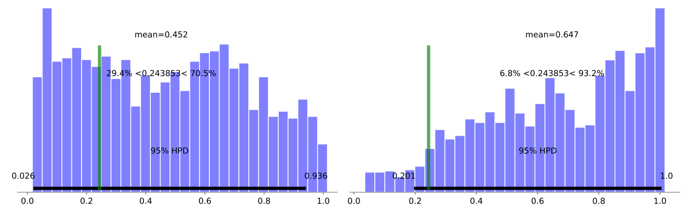
does not significantly reduce the reward obtained. Coalition formation as a criterion of delegation seems to traverse in an equally exhaustive manner both types of topologies.

As DID operates exclusively on a learning-by-observing mechanism, contrary to DIG agents who interact strategically, it struggles to traverse the delegation network when subject to resource constraints, often being confined to local maxima. We believe this is also the reason behind the high levels of the Expenditure-Reward ratio (E/R) encountered in Figures 3 b and 3 c, as well as the insufficient performance of the MAB heuristic compared to the levels of PSD reported in [1].

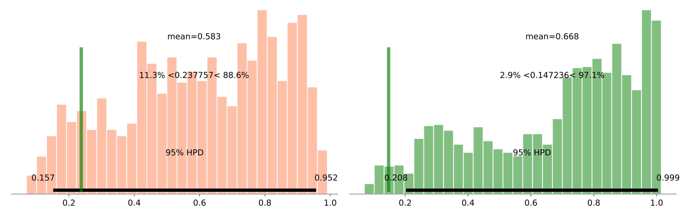
Despite DIG's appropriateness for use in recursive delegation, the relative variability of PSD noted at the beginning of this section and the decline in the levels of rewards, motivate further analysis when transitioning from trees to unstructured environments. For this reason, we opted to conduct a test of correlation between PSD and E/R.

Our test consists of a Bayesian reformulation of Pearson's [11] for a Gaussian mixture of the prior of the correlation coefficient, centered in accordance with the corresponding distribution of the observations plotted in Figure 3. PSD and E/R were fitted to a bivariate t-distribution with uninformative normal, uniform, and exponential priors for their respective means, variances and normality parameters, as per the BEST model put forward in [6]. All hyperparameters were obtained from the outputs of our original simulations (unreported). Figure 4 provides direct access to the posterior distribution of the correlation coefficient, in terms of the coefficient's 95% credible intervals. The results of the No-U-Turn sampler (unreported) guarantee the convergence of distributions, allowing for a direct interpretation of the mean posterior.

There exists a stronger correlation between efficient resource expenditure and increments in the likelihood of a successful delegation, when coalitional algorithms are used on Random Networks. Nonetheless, with a posterior probability of 86%, higher correlation values ($0.24 > 0.15$) are likely to be encountered in the same structures when agents use DIG (Figure 4).

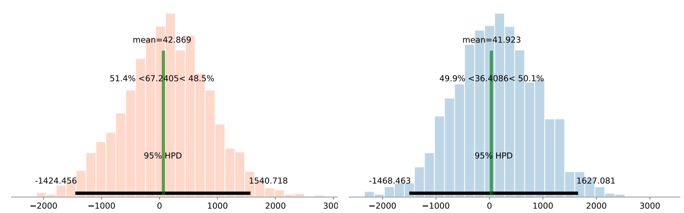


a: DIG over Directed Trees   b: DIG over Random Networks

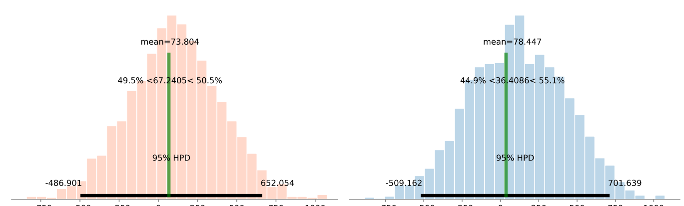


c: Myerson over Directed Trees   d: Shapley over Random Networks

Fig. 4: Posterior Distributions of the Correlation Statistic for PSD and R/E



a: Myerson over Random Networks   b: Shapley over Random Networks



c: Myerson over Directed Trees   d: Shapley over Directed Trees

Fig. 5: Posterior Distributions of the Differences of Means against DID

With respect to the same criterion, Shapley and Myerson can be considered more efficient in the use of the limiting resource. DIG, however, secures desirable levels of PSD while employing relatively concurrent levels of the resource at a rapid pace.

Were the limiting resource apt for alternative uses, a coalitional approach to delegation would be more appropriate than a non-cooperative one, but in any case more pertinent than a MAB-based procedure. In this sense, the Shapley and Myerson algorithms are considered approximate solutions to the (recursive) delegation problem .

Finally, to elaborate on our last claim let us examine Figure 5. It reports the difference between the mean rewards produced by DIG and those produced by DEC, using the same statistical model of the modified Pearson's test. Our results indicate that the group means are not credibly different. Over both Directed Trees and Random Networks, approximately 50% of the posterior probability is greater than zero, suggesting that the gap between the root's mean reward under DIG and the coalitional alternatives is not significantly different from zero. Furthermore, the means of the group distributions range between 41 and 79 units of value, which is less than a third of the average reward earned by the root per trial. So, on grounds of efficiency and value generation capacity, both implementations of DEC are on a par with DIG.

5 Discussion and Future Work

So far we have provided empirical evidence demonstrating that the quitting-game approach to recursive delegation retains all the desirable properties reported in [1], though mediated by the intensive use of the limiting resource. Our algorithm, on the other hand, guarantees the delegated task is carried out with a probability within reasonable limits ($PSD \approx 0.9$), while interactions can be sustained for longer periods of time ($T > 200$).

The resource-use efficiency of the Shapley and Myerson values is upheld by the mechanism dictating the formation of coalitions. The time complexity of this sampling process is quasilinear on restricted graphs and polynomial on random networks, due to the linear structure of the coalitions formed by DEC, thus conforming to the neighbour sampling complexity of DIG and DID [1]. The impact of more intricate coalitions on the levels of PSD, within complex systems where agents not only delegate but engage in multiple interactions dependent on the same productive resource, remain to be determined in future work.

6 Conclusions

In this paper we introduce resource constraints, alongside allocation and rewarding rules to recursive delegation. We further present a conceptual framework to cater for collective responses to these conditions. Quitting-game and multi-armed bandit based approaches are used as benchmarks for evaluating the performance of adaptations of the Shapley and Myerson values to recursive delegation.

Our results indicate that over predefined networks of agents (Directed Trees) and unstructured environments (Random Networks), the quitting game approach attains greater rewards and higher probabilities of successful delegation. This is possible, however, only with the intensive use of the productive resource limiting delegation. In scenarios where constraints are decisive for the operation of multi-agent systems, coalitional games provide a second-best yet more resource-efficient alternative.

References

1. Afanador, J., Baptista, M., Oren, N.: An adversarial algorithm for delegation. In: International Conference on Agreement Technologies. pp. 130–145. Springer (2018)
2. Alonso, R., Matouschek, N.: Optimal delegation. The Review of Economic Studies **75**(1), 259–293 (2008)
3. Brezzi, M., Lai, T.L.: Optimal learning and experimentation in bandit problems. Journal of Economic Dynamics and Control **27**(1), 87–108 (2002)
4. Castelfranchi, C., Falcone, R.: Towards a theory of delegation for agent-based systems. Robotics and Autonomous Systems **24**(3-4), 141–157 (1998). https://doi.org/10.1016/S0921-8890(98)00028-1
5. Etuk, A., Norman, T.J., Oren, N., Sensoy, M.: Strategies for truth discovery under resource constraints. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. pp. 1807–1808. International Foundation for Autonomous Agents and Multiagent Systems (2015)
6. Kruschke, J.K.: Bayesian estimation supersedes the t test. Journal of Experimental Psychology: General **142**(2), 573 (2013)
7. Li, J., Mohapatra, P.: Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. Pervasive and Mobile Computing **3**(3), 233–254 (2007)
8. Li, Z., Peng, Y., Zhang, W., Qiao, D.: Study of joint routing and wireless charging strategies in sensor networks. In: International Conference on Wireless Algorithms, Systems, and Applications. pp. 125–135. Springer (2010)
9. Maschler, M., Solan, E., Zamir, S.: Game Theory. Cambridge University Press (2013). https://doi.org/10.1017/CBO9780511794216
10. Myerson, R.B.: Graphs and cooperation in games. Mathematics of operations research **2**(3), 225–229 (1977)
11. Sedgwick, P.: Pearsons correlation coefficient. Bmj **345**, e4483 (2012)
12. Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. Ad Hoc Networks **1**(2-3), 215–233 (2003)
13. Shapley, L.S.: A value for n-person games. Contributions to the Theory of Games **2**(28), 307–317 (1953)
14. Skibski, O., Michalak, T.P., Rahwan, T., Wooldridge, M.: Algorithms for the shapley and myerson values in graph-restricted games. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. pp. 197–204. International Foundation for Autonomous Agents and Multiagent Systems (2014)
15. Solan, E., Vieille, N.: Quitting games. Mathematics of Operations Research **26**(2), 265–285 (2001)
16. Solan, E., Vieille, N.: Quitting games–an example. International Journal of Game Theory **31**(3), 365–381 (2003)
17. Vecchio, M., Viana, A.C., Ziviani, A., Friedman, R.: Deep: Density-based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility. Computer Communications **33**(8), 929–939 (2010)
18. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A two-tier data dissemination model for large-scale wireless sensor networks. In: Proceedings of the 8th annual international conference on Mobile computing and networking. pp. 148–159. ACM (2002)