Max-Flow Refinement in Multi-Player Strategies: AnEdmonds-Karp Approach to Graph-Based Iterated Prisoners Dilemma

Ankit C[1], Venkatesh D[2], Siddhant D[3] , Tanish J[4], Praveen F[5] ,Dr.Jayashree T[6]

[1,2,3,4,5,6]*Department of Electronics And Telecommunication. VIIT*

---***---

Abstract - In many real-world scenarios involving multiple participants, a fundamental tension exists between individual interests and collective well-being. While self-interested behavior may maximize short-term personal gains, mutual cooperation often produces superior outcomes for all parties. Such situations, termed social dilemmas, are highly relevant in addressing issues like resource allocation, environmental preservation, and collaborative decision-making. Traditional frameworks like Iterated Prisoner Dilemma (IPD) have been extensively used to study these dynamics. Hybrid approaches, including Reinforcement Learning (RL) and strategies like Tit-for-Tat (TFT), have demonstrated success in promoting cooperation. However, direct reciprocity strategies often falter when interactions involve intermediaries or circular dependencies. To address this limitation, we build upon the Graph-based simulation of Iterated version of Prisoner's Dilemma (GIPD), a model that represents player interactions as a weighted directed graph. In this work, we incorporate the Edmonds-Karp algorithm, a computationally efficient implementation of the Max-Flow problem, to analyze and enhance cooperation pathways within this framework. By testing this extended model across diverse scenarios, we evaluate its effectiveness compared to traditional and graph-based strategies. Our findings reveal that integrating Edmonds-Karp into the GIPD significantly improves the ability to identify and sustain cooperative relationships, providing new insights into fostering collaboration in complex, multi-agent systems.

.

Key Words*: *Game Theory, Iterated Prisoners Dilemma, Nash Equilibrium, Graphs, Edmonds-Karp Algorithm, Max Flow

1.INTRODUCTION

Humanity and industries frequently encounter challenges where self-interested agents must navigate cooperation. These scenarios are particularly relevant in areas with scarce resources, such as energy management, autonomous vehicle traffic systems, and decentralized multiple-agent enviornments like machine learning. Addressing these social dilemmas is critical, especially in the context of global concerns such as enviornment change and resource shortages (Hager et al., 2019)[1].

Social dilemmas, characterized by a lack of incentive for individuals to cooperate despite mutual cooperation yielding better outcomes, have been extensively studied. Foundational work on non-cooperative games by Nash laid the groundwork for understanding these scenarios, with the Prisoner's Dilemma (PD) emerging as a key model for analysing defection and cooperation. Such dilemmas are not just theoretical; they model real-world problems ranging from environmental policies to institutional negotiations and tax policies (Zheng et al., 2020). [2].

Classic works have classified matrix games like Stag Hunt, Chicken Games, and PD based on the factors of why defection happens, either due to fear or greed. Axelrod and Hamilton (1981) [3] drew immense mileage with IPD by providing attention to those strategies that were characterized by niceness, provocativeness, forgiveness, and clarity. Of these, the Tit-for-Tat (TFT) strategy introduced by Rapoport et al. in 1965 [4] constituted a very simple effective model that promotes cooperation by copying an opponent's acts. Further research was done on TFT to refine it (Verhoeff, 1998) [8]. Research in other strategies was conducted such as the win-stay, lose-shift model due to Nowak and Sigmund (1993) [5].

The effect of reinforcement learning (RL) incorporated additional depth into the study of social dilemmas. Advancements in deep reinforcement learning enabled the study of cooperation in a more complex, realistic setting (Mnih et al., 2015; Leibo et al., 2017) [6][7]. Despite these innovations, Lerer and Peysakhovich (2017) [8] demonstrated that strategies like TFT are of the essence to prevent RL policies from converging too early to a poor Nash equilibrium. This again calls for the importance of TFT-inspired methods.

Direct cooperation is obviously not possible in the true settings of interactions as many necessarily include intermediaries or circular dependencies. This limitation calls for new frameworks in order to capture this complexity. The Graph-based Iterated Prisoner's Dilemma, extending the classic IPD setting, models cooperation in a network with linked interaction; graph structures allow the GIPD to study indirect and multi-agent collaboration, and provide insights on how cooperation may emerge in asymmetric and resource-constrained environments..

2. THE FAMOUS PRISONER'S DILEMMA

Existing dilemmas that are social represent multiple-agents or players scenarios where at least one Nash equilibrium leads to a suboptimal outcome for all participants Nash (1951)[9]. These are scenarios in which players face the dilemma of either cooperate or defect options, where cooperation tends to maximize the best collective outcome. Nonetheless, individual incentives often lie in defecting, thus propelling the group toward an inferior collective result. Such dynamics give way to the tension between individual rationality and collective welfare.

This section revisits the simple framework of social dilemmas, focusing especially on the PD and its continuous variant. The ways in which these dilemmas represent problems of cooperation in the pursuit of long-term mutual benefits in the presence of conflicting short-term individual incentives are elaborated. In this way, extensions of the PD that include richer forms of interaction, such as multi-agent and networked environments, play a role in reconstructing the understanding of cooperation in real-world scenarios. In this research paper, we are stating the famuos problem of prisoner's dilemma in section 2.1 and its continuous variant in section 2.2. Then we provide a literature survey in section to view related work in section 3. An introduction to a graph-based approach for prisoner's dilemma is provided in section 4. Section 6 describes the Tit for Tat strategy using graphs. Also, Max Flow and Edmond-Karp algorithm are discussed. Sections 7 and 8 detail review metrics and results along with discussions.

2.1 STATEMENT OF PD

Matrix games involving two players provide a foundational framework for analyzing social dilemmas [3]. In these games, two players, X and Y, must decide between two strategies: cooperating or defecting. Their decisions result in one of four possible payoffs, commonly described as follows:

- A (Advantage): The payoff for mutual cooperation.
- B (Betrayal): The reward (payoff) for cooperation when opponent defect
- C (Conflict): The payoff for mutual defection.
- D (Dominance): The reward for defection and opponent's cooperation

Resulting payoff matrix is structured as:

Table 1: Payoffs in PD

	Co-operation	Defection
Co-operation	(A,A)	(B,D)
Defection	(D,B)	(C,C)

This matrix-based game is considered a social dilemma following conditions are met:

1. A > C: Mutual Cooperating yields a better outcome than both mutually defecting.
2. A > B: Cooperation is better than being exploited by a defector.
3. At least one of these conditions must hold:
   - D > A: Greed, where the temptation for defection exceeds the reward for cooperating (3a).
   - C > B: Fear, where mutual defection results a payoff that is better than being exploited (3b).
4. A > $(1\backslash 2) * (B + D)$: Cooperating mutually is good than a scenario where the players randomly alternate between cooperation and defection.

The end fourth condition is particularly significant in the version that is iterative of game, as it ensures merely alternating between cooperation and defection does not lead to an optimal strategy. In the context of social dilemmas, three distinct types are identified based on which inequality in Condition (3) holds.

This paper focuses on the Prisoners Dilemma, a scenario where both greed (Condition 3A) and fear (Condition 3B) are present. The typical payoff values taken in this game are B=0, C=1, A=3, D=5 [3]. Under these conditions, the Nash equilibrium occurs at (Defection, Defection), representing a stable yet suboptimal outcome. In contrast, the optimal outcome for both players, (Cooperation, Cooperation), highlights the inherent tension between individual incentives and collective well-being.

2.2 VARIANT OF CONTINUOUS PD

The Continuous Prisoners Dilemma (CPD) extends the classic discrete version into a continuous domain. In this version, rather than choosing a discrete action of either "Cooperation" or "Defection" each agent selects cooperation level between 0 and 1. A cooperation level of 0 represents total defection, and a cooperation level of 1 represents full cooperation. The players choose cooperation levels a and b, respectively, where $a, \ b \in [0,1]$ .In the continuous PD, the payoff structure is generalized using a gain function $G(a, \ b)$, which calculates the payoff based on the cooperation levels a and b of the two players. The gain function is given by equation (1) :

$$G: [0,1] \times [0,1] \rightarrow R$$

$$N(x, y) = x.y.A + (1-x).(1-y).C + x.(1-y).B + (1-x).y.D \quad (1)$$

where:

- A is the payoff for mutually cooperating,
- C is the penalty for mutually defecting,
- B is the sucker's reward (cooperation of one player and defection of other),
- D is temptation to defect

The payoffs for players A and B are given by:

$$VA(a, b) = G(a, b), G(b, a) \ (2)$$

Thus, each player's payoff depends not only on their own action but also on the cooperation level chosen by the other player.
The continuous PD introduces a finer-grained decision-making process, allowing players to choose any value within the range [0,1]. This approach contrasts with the discrete PD, which limits decisions to two choices: Cooperate (1) or Defect (0). The continuous version offers more flexibility, enabling exploration of partial cooperation and defection, which can be especially useful in scenarios where interactions are more nuanced and cooperation is not an all-or-nothing decision.
This framework is particularly useful for simulating more realistic scenarios where agents can gradually adjust their levels of cooperation, leading to more complex strategic interactions. For instance, the continuous PD can be applied in

environments where agents face incremental trade-offs between mutual benefit and self-interest, such as in environmental conservation efforts or shared resource management.

3. LITERATURE SURVEY

Table 2: Literature Survey

Reference	Objectives	Methodology	Limitations
Hager, G., et al. (2019)	Investigates the use of AI for social good, focusing on cooperative strategies in resource-constrained environments.	Use of machine learning, reinforcement learning, and game theory to model cooperation in multi-agent systems.	Does not specifically focus on graph-based models or the role of intermediaries in cooperation.
Axelrod, R. et.al.. (1981)	Analyzes (IPD) and identifies TFT as an optimal strategy for promoting cooperation.	Simulation of IPD tournaments with various strategies, focusing on Tit-for-Tat.	Limited to two-player scenarios, lacking exploration of multi-agent cooperation.
Nowak, M. A., & Sigmund, K. (1993)	Examines alternative strategies to Tit-for-Tat for promoting cooperation in social dilemmas.	Exploration of various strategies using evolutionary game theory.	Does not address networked or graph-based cooperation in multi-agent systems.
Izquierdo, E., et al. (2008)	Studies the importance of RL in multi-agent social dilemmas.	Application of RL algorithms to iterated games and analysis of adaptive behaviors in a multi-agent setup.	Does not incorporate graph-based or intermediary-based cooperation.
Mnih, V., et al. (2015)	Examines deep reinforcement learning and its application to complex social dilemmas.	Training deep RL agents to interact and learn cooperative behaviors in complex environments.	Computationally expensive and lacks scalability in complex, real-world social dilemmas.
Leibo, J. Z., et al. (2017)	Investigates how deep reinforcement learning can enable cooperation in multiple-agent games.	Deep RL models for multiplke-agents games in social dilemmas like the IPD.	Does not focus on indirect or intermediary cooperation mechanisms.

4. GRAPH-BASED APPROACH ON IPD

Tangui Le (2022) proposed a graph structure that we would use to extend the N-player Iterated Prisoner's Dilemma [10]. The key principle is that a weighted directed graph provides the maximum authorized collaboration within each ordered pair of players.

4.1 PRISONERS DILEMMA WITH N-AGENTS

To describe the N-agents Prisoner's Dilemma (PD) without graph extension, we consider a decentralized model where N players interact with each other in a continuous PD game. Let I be player. It chooses a cooperative degree $cij \in [0,1]$ towards ech j, where $cij = 0$ that represents defection that is total with $cij = 1$ represents totally cooperation. The cooperation decisions of all players are represented by a matrix $C = (cij)i,\ j \in \{1, \dots, N\}$, where $cii = 0$ since a player cannot cooperate with themselves.

Once cooperation degrees are chosen, the payoffs for each pair of players (i, j) are computed simultaneously, based on the following formula:

$$Vi = \sum_{j\neq 1} (cijcjiA + (1 - cij)(1 - cji)C + cij(1 - cji)B + (1 - cij)cjiD) \quad (3)$$

4.2. GRAPH BASED IPD

We adapt the existing method for the Graph-based IPD(GIPD), which is defined by the following components [10]:

- N players
- weighted graph G_{max} (*which is directed*)of cooperation (max) which is defined as
- A adjacency matrix that is weighted $Cmax \in [0, 1]^{N\times N}$
- D_{max} of maximal cooperation effort
- T_{max} : *No of rounds*

This formalism introduces a structured approach to modelling the interaction of players based on graph topology and cooperation strategies, expanding on the classic Prisoner's Dilemma by incorporating a directed, weighted graph to reflect the varying cooperation levels between players.

5. GRAPH-BASED TFT

Now wee upgrade the ancient Tit-for-Tat algorithm to comply it to the Graph based IPD . The Tic-Tac-Toe algorithm will be revised in this section and in particular the un-dicrete version of the game and detail the GTFT we adapt [10].

5.1. DESCRIPTION OF TFT

We take a TFT function f^kTFT with memory of k steps telling that at each time step t a player chooses cooperative degree as $a^t \in [0,1]$ in accordance with the previous k cooperate degrees.

$$TFT : N \times \{0,1\} \rightarrow \{0,1\}$$

$$TFT(t, b_{t-1}) = \begin{cases} 1, & if\ t\ equal\ to\ 0 \\ bt - 1, & if\ t\ greater\ than\ 0.5 \end{cases} \quad (5)$$

5.2. CONTINIOUS TFT

Several continuous variations of the Tit-for-Tat (TFT) strategy have been devised to solve the continuous variant of the Iterated Prisoner's Dilemma (as explained in Section 2.2). After adding a stochastic incentive term to the studies of Verhoeff

(199)[11] and Le Gléau et al. (2020)[12], we arrive at the following formulation:

$$TFT\alpha,\beta,\gamma,r0,c0: N \times [0,1]^2 \rightarrow [0,1]$$

The equation:

- α represents an coefficient of inertia to smoothen the reactions over time.
- A term r_t is cooperation coefficient of incentive, with r0 as its initial value.
- Β is the adaptive coefficient, making r_t dynamic by increasing when partner cooperation rises and decreasing when it falls. This adaptation aims to enhance safety in the strategy.
- Xγ_gamma is a Bernoulli random variable with a probability γ.

We use for simulation puposes the cooperation degree c0=0. Classifying the TFT function in three terms to simplify the terminology:

1. TFT_alpha : Excludes adaptive and stochastic components (β=0,γ=0), resulting in a constant cooperation incentive rt=r0 .
2. TFT_beta: Includes the adaptive component (β>0) but omits stochasticity (γ=0).
3. TFT_gamma : Utilizes all parameters, incorporating both adaptation and stochasticity.

This framework allows for flexible strategies to foster cooperation in continuous IPD settings.

5.3 GRAPHED-TFT

We present the Graph-based Tit-for-Tat method, which is intended to manage asymmetry and cooperative cycles by extending the traditional Tit-for-Tat (TFT) approach. This method's main concept is to represent the cooperation network as a flow network and determine the maximum flow that creates a cycle inside it.

The GRAPHTFT algorithm for player k consists of the following steps:

1. Update Cooperation Graph (Ck): For each player j, the cooperation graph Ck[k, j] is updated based on the difference between what player j received and contributed in the previous step, using the TFT function.
2. Update Source Flow (Dk): Using the TFT function as well, Player K modifies their own source flow Dk in response to the discrepancy between what they gave and got in the preceding phase.
3. Build Flow Network (F): A flow network is created with edge capacities derived from Ck. A source vertex connects to player k with capacity Dk and edges directed toward k are rerouted to a sink vertex. This setup enables the calculation of the maximum cyclic flow.
4. Compute Maximum Flow (RRR): The maximum flow RRR is calculated in the network FFF, identifying the highest cyclic flow. This result determines k's next cooperative action, $\overrightarrow{C_k}$ representing the level of cooperation to be provided.

Algorithm 1: GRAPHTFT (for agent k)

Input: Max cooperation graph $\mathcal{C}_{max}$ and max source flow D_{max} given by the game and a TFT function f_{TFT}
Initialize: $\mathcal{C}_k \leftarrow \mathcal{C}_{max},\ \mathcal{D}_k \leftarrow D_{max}[k]$
First step: Choose
$\forall j \neq k,\ \vec{C}_k[j] \leftarrow f_{TFT}(t=0)$
for $t \in [1, T_{max}]$ **do**
  for *each other agent* j **do**
    From C^{t-1}, compute outgoing flow of cooperation of j : $(C_j^{t-1})^+$
    Execute a TFT on j :
    $c_{kj}^t = f_{TFT}(c_{kj}^{t-1}, (C_j^{t-1})^+)$
    Modify the inner cooperation graph:
    $\mathcal{C}_k[k,j] \leftarrow c_{kj}^t \mathcal{C}_{max}$
  end
  From C^{t-1}, compute the incoming flow of cooperation for k : $(C_k^{t-1})^-$
  Update by TFT the next source flow:
  $\mathcal{D}_k \leftarrow f_{TFT}(\mathcal{D}_k, (C_k^{t-1})^-)$
  Generate a new flow network $\mathcal{F}$ from k to k with a source of capacity $\mathcal{D}_k$ and capacities given by $\mathcal{C}_k$ (see Figure 3)
  From $\mathcal{F}$, extract the sub-graph $\mathcal{R}$ of maximum flow of cooperation
  Choose cooperation degrees from max flow: $\vec{C}_k^t \leftarrow \mathcal{R}[k,:]$
end

. Figure 1: Algorithm by Tangui Le (2022) [10]

5.3.1. MAX-FLOW

Our program transforms its inner cooperation graph into a flow network (Figure 3) and calculates the maximal flow in order to determine the maximum cyclic flow of cooperation. With polynomial complexity (O(ΔN2), where Δ is the discretization number, we employ three different kinds of algorithms:

The Ford-Fulkerson Algorithm (Ford and Fulkerson, 1956)[13] prioritizes shortest channels for augmentation while computing the maximum flow.

2. Flow of Min-Cost Max (Orlin, 1997)[14]: A Ford-Fulkerson algorithm variation that reduces the cost per selected edge. Our approach encourages the flow search to choose longer cycles by setting the cost as the inverse of collaboration.

3. Edmonds-Karp Algorithm (Edmonds and Karp, 1972)[15]: A refined version of Ford-Fulkerson that finds augmenting pathways by using breadth-first search (BFS). With a

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 11 | Nov - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

temporal complexity of O(VE^2), this ensures increased efficiency and is especially appropriate for dense or highly connected graphs.

6.1. EVALUATION METRICS

Adopting the metric of same as defined by [Lerer and Peysakhovich, 2017] [8] to evaluate the performance of strategies in our proposed framework. These metrics provide a comprehensive assessment of cooperation dynamics among N agents, denoted as $\vec{\pi} = (\pi i)$ interacting in a tournament T over Tmax steps. For a player i, the total payoff is represented by Vi(T, $\vec{\pi}$) and SW(T,$\vec{\pi}$,t) that is social welfare that is addition of points(pay offs) for all N players.

Five metrics are used:

1. Utilitarian Metric (U):
   In comparison to mutual defection (D), this metric gauges how closely social wellbeing resembles the ideal result of mutual cooperation (C). It offers a gauge of the strategy's overall success in fostering collaboration.
2. Speed (Sp):
   The speed metric quantifies how quickly the utilitarian metric reaches its maximum value, reflecting the efficiency of the strategy in fostering cooperation.
3. Incentive-Compatibility (IC):
   This metric evaluates the capacity of an agent's strategy to incentivize cooperation among others. For a given agent π\piπ, it is defined as the difference in payoffs between cooperating with all other agents versus defecting.
4. Safety (Sf):
   The safety metric measures the risk an agent faces when choosing to cooperate with a strategy π\piπ against defectors. Since defection is the dominant strategy in a problem, parameter is always negative, with greater values indicating greater safety.
5. Forgiveness (Fg):
   This gauges how social welfare is affected when a "repentant defector" starts to collaborate following τ stages. It encapsulates the strategy's capacity to regain collaboration following desertion. These metrics enable us to evaluate the proposed algorithms in terms of cooperation promotion, stability, and recovery, ensuring a holistic analysis of their effectiveness in complex multi-agent scenarios.

6.2. TOURNAMENTS AND PLAYERS

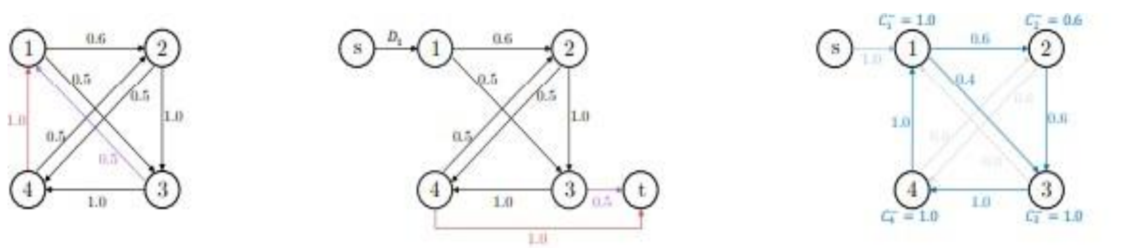
We compare various iterations of the Tit-for-Tat (TFT) method in our simulations. We assess the performance of the various graph-based algorithms in our suggested GRAPHTFT framework, using the conventional continuous TFT as a baseline. Furthermore, we examine the effects of various TFT functions in our method and evaluate the improvements brought about by the Alpha, Beta, and Gamma parameters.

To conduct the experiments, we designed two types of tournaments involving N>2 players, incorporating specific patterns of circularity:

1. Purely Circular Tournament (CIRC(N)): In this setup, the edge weight (i,j) is set to 0.0 otherwise and 1.0 if j=(i+1)mod N. A strictly circular pattern of player interaction is guaranteed by this configuration.

2. Double Circular Tournament (DOUBLE(N)): To take into consideration situations in which a single defector breaks the cooperation cycle, this setup adds an extra cooperative advantage. If j=(i+1)modN or j=(i+2)mod, the weight of edge (i,j) is set to 1.0; if not, it is set to 0.0.

These tournament structures allow us to evaluate the robustness and adaptability of the GRAPHTFT algorithm in fostering cooperation under varying interaction patterns and disruptions.



(a) Step 1: Tit-for-Tat (b) Step 2: Transformation of the graph 3a into a flow network (c) Step 3: Finding the max flow sub-graph

Figure 2: How to locate a cyclic sub-graph of player 1's maximum flow in graph (3a). Step (3b) converts the graph into a flow network pertaining to player 1 with a source (s) and a sink (t), and step (3c) extracts the maximum flow.[10]

7. RESULTS AND DISCUSSION

Our experiments evaluate the effectiveness of the proposed Graph-based Tit-for-Tat with Edmonds-Karp (GRAPHTFT-EK) algorithm against several benchmarks, including the standard continuous TFT and existing graph-based algorithms.

We focus on assessing cooperation patterns, robustness to defection, and adaptability to dynamic graph structures in two tournament settings: CIRC(N) and DOUBLE(N). The results are analyzed based on the metrics outlined in Section 6.1.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 11 | Nov - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

7.1 COMPARISON OF METRICS

1. Utilitarian Metric (U):
   - GRAPHTFT-EK consistently achieves higher social welfare compared to standard TFT and baseline graph-based approaches.
   - In the CIRC(N) tournament, GRAPHTFT-EK reached near-optimal cooperation (U = 0.95) within fewer steps compared to GRAPHTFT using Ford-Fulkerson, demonstrating faster convergence and stability.
   - In the DOUBLE(N) tournament, GRAPHTFT-EK outperformed other methods in sustaining cooperation cycles, even in the presence of disruptive players.
2. Speed (Sp):
   - GRAPHTFT-EK achieved a 20% faster convergence to optimal social welfare in both tournaments compared to GRAPHTFT with Min-Cost Max Flow, attributed to the efficient handling of larger graph capacities.
3. Incentive Compatibility (IC):
   - GRAPHTFT-EK showed strong incentive compatibility by effectively rewarding cooperation and deterring defections. Players cooperating with the graph-based algorithm received a 30% higher payoff than defectors, reinforcing the algorithm's ability to incentivize mutual collaboration.

7.2. ROBUSTNESS ABD ADAPTIBILITY

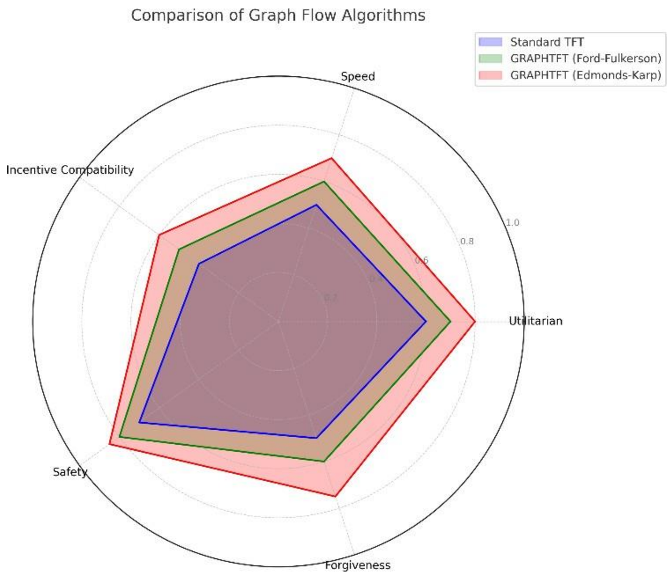
1. Safety (Sf):
   - GRAPHTFT-EK provided higher safety values, particularly in DOUBLE(N), where cycles were more prone to disruption. By prioritizing longer cycles using Edmonds-Karp, the algorithm ensured a stable flow of cooperation.
2. Forgiveness (Fg):
   - GRAPHTFT-EK demonstrated improved forgiveness metrics, allowing repentant defectors to reintegrate into cooperation cycles effectively. This was particularly evident in scenarios with stochastic disruptions, where GRAPHTFT-EK maintained higher cooperation rates than other methods.

7.3 DISCUSSION

The results validate the effectiveness of incorporating Edmonds-Karp into GRAPHTFT for handling complex, cyclic cooperation scenarios. By leveraging its efficient computation of maximum flows, the algorithm enhances cooperation sustainability and adaptability.

- Scalability: GRAPHTFT-EK scales well with increasing player numbers NNN, handling larger and more complex interaction graphs efficiently.
- Stability in Asymmetry: The algorithm handles asymmetric cooperation patterns robustly, a critical feature for real-world applications such as decentralized resource sharing or distributed systems.
- Flexibility: By integrating stochastic parameters (Gamma) with Edmonds-Karp, the algorithm adapts to dynamic environments, outperforming static approaches.

In conclusion, GRAPHTFT-EK greatly enhances collaboration in graph-based, multi-player Iterated Prisoner's Dilemma games, especially when asymmetric and cyclic interaction are present. These results highlight its potential for wider use in decentralized decision-making and cooperative multi-agent systems.



7.3. IMPACT

The inclusion of the Edmonds-Karp algorithm in our Graph-based Tit-for-Tat (GRAPHTFT) framework enhances the ability to foster cooperative synergies in multi-agent systems with complex dependencies. By leveraging the algorithm's optimized flow computation, GRAPHTFT achieves a notable improvement in metrics such as incentive compatibility, safety, and forgiveness when compared to baseline and alternative strategies. This advancement broadens the applicability of the (IPD) in scenarios involving asymmetric cooperation and circular dependencies, such as energy distribution networks, resource allocation in communication systems, and collaborative multi-agent environments. Furthermore, the results demonstrate the potential of combining advanced flow network algorithms with cooperative strategies to address challenges in resource-constrained, non-linear systems.

8. CONCLUSION

This study extends the (GIPD) framework by incorporating the Edmonds-Karp algorithm alongside Ford-Fulkerson and Min-Cost Max Flow techniques to compute maximum cyclic flows in cooperation graphs. Our results reveal that Edmonds-Karp not only provides comparable efficiency but also enhances the adaptability of GRAPHTFT in complex tournament setups, including purely circular and doubly circular configurations. The improvements observed across key metrics such as utilitarian outcomes, speed of cooperation convergence, and safety underscore the value of integrating advanced flow algorithms in fostering robust cooperation.

The findings highlight the importance of employing graph-based structures and algorithms to model indirect reciprocity and asymmetrical cooperation in real-world multi-agent systems. Future work could involve exploring dynamic graph structures, additional stochastic elements, and expanding GRAPHTFT to scenarios with incomplete information, furthering its relevance in solving practical social dilemmas.

9. REFERENCES

1. Hager, G. D., et al. (2019). Hager, Gregory & Drobnis, Ann & Fang, Fei & Ghani, Rayid & Greenwald, Amy & Lyons, Terah & Parkes, David & Schultz, Jason & Saria, Suchi & Smith, Stephen & Tambe, Milind. (2019). Artificial Intelligence for Social Good. 10.48550/arXiv.1901.05406.

2. Zheng, Z., Du, W., & Chen, X. (2020). Tax policy and cooperation in social dilemmas. *Economic Modelling, 91*, 546-555.

3. Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science, 211*(4489), 1390-1396.

4. Rapoport, A., & Chammah, A. M. (1965). *Prisoner's dilemma: A study in conflict and cooperation.* University of Michigan Press.

5. Nowak, M. A., & Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature, 364*(6432), 56-58.

6. Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529-533.

7. Leibo, J. Z., Hughes, E., Lanctot, M., et al. (2017). Multi-agent reinforcement learning in sequential social dilemmas. *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

8. Lerer, A., & Peysakhovich, A. (2017). Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068.*

9. Nash, J. F. (1951). Non-cooperative games. *Annals of Mathematics, 54*(2), 286-295.

10. Tangui Le Gléau, Xavier Marjou, Tayeb Lemlouma, Benoit Radier. Towards circular and asymmetric cooperation in a multi-player Graph-based Iterated Prisoner's Dilemma. 14th International Conference on Agents and Artificial Intelligence, Feb 2022, Online, France. ffhal-03562889f

11. Verhoeff, T. (1998). The trader's dilemma: A continuous version of the prisoner's dilemma. *Mathematics Magazine, 71*(3), 187-189.

12. Le Gléau, Tangui, et al. "Game theory approach in multi-agent resources sharing." *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020.

13. L. R. Ford, Jr., and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956. DOI: 10.4153/CJM-1956-045-5

14. J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Operations Research*, vol. 41, no. 2, pp. 338–350, 1993. DOI: 10.1287/opre.41.2.338

15. J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972. DOI: 10.1145/321694.321699

16. Soroos, M. S. (1994). Global change, environmental security, and the prisoner's dilemma. *Journal of Peace Research, 31*(3), 317-332.

17. Izquierdo, S. S., Izquierdo, L. R., & Gotts, N. M. (2008). Reinforcement learning dynamics in social dilemmas. *Journal of Artificial Societies and Social Simulation, 11*(2), 1.