In the Search for Optimal Portfolios of Counterstrategies in the Large Imperfect Information Games

Karolina Drabent DRABEKA1@FEL.CVUT.CZ
Ondřej Kubíček KUBICON3@FEL.CVUT.CZ
David Milec MILECDAV@FEL.CVUT.CZ
Viliam Lisý VILIAM.LISY@FEL.CVUT.CZ
Czech Technical University in Prague, Czech Republic

Abstract

Learning methods for optimizing strategies in very large multi-agent settings (a.k.a. games) often abstract the space of all possible strategies of the opponents to a small portfolio. Existing scalable methods for constructing these portfolios are either hand-designed for specific domains or heuristics without understood guarantees. To enable a more fundamental approach, this paper studies the problem in small normal form games. We formally define the optimization problem of finding the optimal portfolio for approximating the Nash equilibrium in two-player zero-sum games. We propose a method to approximate the optimal solution in small games. Even though this approximation is provably suboptimal, it is enough to demonstrate that the portfolios constructed by a recent scalable heuristic are very far from the optimum. We study the reasons and propose some improvements, but we are still far from closing the gap.

1. Introduction

Imperfect information extensive form games (IIGs) such as Dark Chess, Stratego, or Poker are relevant [3, 17] not only because these games are popular but also because they resemble real-world problems more than perfect information games. However, these games present greater challenges than perfect information games due to the complexity of hidden information [4]. Additionally, it is difficult to impossible to compute Nash Equilibrium in the domain of large games. Even though for the zero-sum games there exists a polynomial algorithm [7] the games are so large that it is intractable. Therefore many solutions are approximating the Nash Equilibrium, very often through abstractions [6, 9, 19]. Recently, a method called Multi-Valued States (MVS) [3, 4, 10] is used to solve large IIGs effectively. It uses a portfolio of strategies of the opponent to be able to adjust the player's strategy to a potentially changing opponent. While portfolios have appeared in various research works [1, 2, 10, 18] of competitive games, they have not been thoroughly studied. In these works, portfolios are found relying on heuristic methods or even crafted by hand.

In this paper, we explore the role of portfolios in optimizing strategies for large IIGs, with a focus on the theoretical foundations behind their use. Specifically, we investigate portfolios for strategy optimization in the context of the MVS approach. MVS leverages portfolios to enhance strategy robustness and efficiency, and we delve into this optimization process in detail. To better understand this issue, we transition to zero-sum normal form games, where we formally define a portfolio and introduce a relevant optimization metric. This allows us to frame the task of finding the optimal portfolio as an optimization problem. We systematically study this issue and propose an algorithm for finding the optimal portfolio and evaluating it. Additionally, we benchmark existing heuristic methods, analyzing their approximation quality and assessing the effectiveness of the portfolios they generate.

2. Related Work

An important portfolio searching algorithm is Double Oracle (DO) [14]. It generates subsets of strategies for each player by iteratively adding best responses to the opponent's current subset of actions. Lanctot et al. [11] introduced Policy-Space Response Oracles (PSRO), a generalized version of DO. Brown et al. [4] modified DO to be used in extensive form games and depth-limit solving.

Another class of methods is based on the transformations of the policy. In Multi-Valued States (MVS) [4] and Pluribus [3], a portfolio is found by ”bias approach”, which uses hard-coded transformations (e.g. multiplication of action probabilities) on the blueprint strategy (approximation of NE). However, Kubíček et al. [10] proposed an extended version of the ”bias approach”. It uses gradients from the Regularized Nash Dynamics [17] algorithm to perform transformations on the strategy.

Kroer et al. [8] define a Mixed-Integer Linear Program that computes an abstraction of an extensive form game that satisfies certain bounds on all levels at once. It can be used to create a portfolio, however, this is not scalable to bigger games and the portfolio is not guaranteed to be optimal.

On the other hand, related to finding the portfolio is work of [18]. They introduce a framework for a robust portfolio of strategies selection in Ad Hoc Teamwork Agents. This is defined in cooperative games. Certain definitions used by them are not fully applicable in the context of competitive zero-sum games. They introduce a concept of minimum coverage sets and use its approximation to create an algorithm for finding portfolios in the big space of strategies.

Implicit modelling uses portfolios for the opponent's exploitation. Bard et al. [2] create the portfolio by clustering strategies from the data. However, it does not scale well and is data-dependent.

3. Preliminaries

A two-player normal form game is a tuple $G = (N, A, u)$. $N = \{1, 2\}$ is a set of players, by i we refer to one of the players, and by $-i$ to their opponent. $A = A_1 \times A_2$ denotes a set of all available action profiles. We denote a set of all probability distributions over actions from A_i as $\Delta(A_i)$, and a set of all probability distributions over action profiles as $\Delta(A)$. For each player $i \in N$ we define an *(expected) utility function* $u_i : \Delta(A) \to \mathcal{R}$, which returns utility for a strategy profile $\pi = (\pi_1, ..., \pi_n) \in \Delta(A)$. $\pi(a)$ denotes the probability of players playing action profile $a \in \Delta(A)$. If $u_i(\pi) = -u_{-i}(\pi)$ for all $\pi \in \Delta(A)$, we say that the game is zero-sum. In this work we will only consider two-player zero-sum games (2p0s), additionally, we will use u as $u = u_1 = -u_2$.

A *pure strategy* π_i for player i is an action $a \in A_i$. A *mixed strategy* π_i is a probability distribution over A_i. A strategy $BR_i(\pi_{-i}) \in \Delta(A_i)$ is best response $BR_i(\pi_{-i}) \in \operatorname{argmax}_{\pi_i \in \Delta(A_i)} u_i(\pi_i, \pi_{-i})$. We call a strategy profile π^* a Nash Equilibrium (NE) if, for all agents i, π^*_i is a best response to π^*_{-i}. An ϵ-Nash Equilibrium (ϵ-NE), $\epsilon \geq 0$, is a strategy profile π such that no player can gain more than ϵ by unilaterally deviating from their strategy. A *game value* $u^* = u(\pi^*)$ of 2p0s game is the utility value of Nash Equilibrium π^*. In 2p0s game the Maximum Entropy Correlated Equilibrium (Maxent) [15] is $\pi^{ME} = \operatorname{argmax}_{\pi^* \in NE} H(\pi)$, where $H(\pi)$ is the (Shannon) entropy of the strategy $H(\pi) = \sum_{a \in A} \pi(a) ln(\frac{1}{\pi(a)})$.

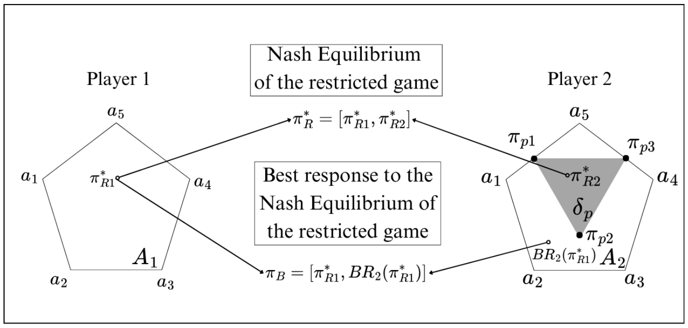

Nash Equilibrium of the restricted game
Player 1
Player 2
a_5
$\pi^*_R = [\pi^*_{R1}, \pi^*_{R2}]$
a_5
π_{p1}
π_{p3}
a_1
π^*_{R1}
a_4
Best response to the Nash Equilibrium of the restricted game
a_1
π^*_{R2}
a_4
δ_p
π_{p2}
$BR_2(\pi^*_{R1})$
A_2
A_1
a_2
a_3
$\pi_B = [\pi^*_{R1}, BR_2(\pi^*_{R1})]$
a_2
a_3

Figure 1: This diagram illustrates the strategy optimization process using a portfolio P in a game G. The strategy space is a multidimensional simplex mapped to two dimensions for simplicity. On the left, the simplex represents the strategy space of Player 1, $\Delta(A_1)$, while on the right, of Player 2, $\Delta(A_2)$. Within the latter, the gray simplex represents the strategy space defined by the portfolio $\Delta(P)$. The NE of the restricted game, π^*_R, is computed for $\Delta(A_1)$ and $\Delta(P)$. Then, π_B is computed by fixing Player 1's strategy and finding best response of Player 2 in the full strategy space $\Delta(A_2)$.

4. Portfolios

A portfolio is a subset of possible actions or strategies that a player will use instead of their actions in the original game. We now formally define a portfolio.

Definition 1 *A Mixed Portfolio of size $k \in \mathbb{N}^+$ of player i in the normal form game G is a subset of mixed strategies $P = \{\pi_{p1}, \pi_{p2}, ..., \pi_{pk}\}, P \subseteq \Delta(A_i)$, so that $|P| = k$.*

Additionally, we define *Pure Portfolio* P as such that consist only of pure strategies, which means that portfolio P of player i is a subset of actions $P \subseteq A_i$.

A normal form game G can be restricted by a portfolio for a player i. In that case, that player is allowed to only choose the actions from the portfolio.

Definition 2 *A normal form game $G = (N, A, u)$ restricted by portfolio $P = \{\pi_{p1}, ..., \pi_{pk}\}$ of player i is called a restricted normal form game $G_R(G, P, i) = (N, A', u')$, where actions are formed as $A' = A'_1 \times A'_2, A'_i = P, A'_{-i} = A_{-i}$ and utility matrix is constructed with $u'(\pi'_i, \pi_{-i}) = \Sigma^k_{j=1} \pi'_i(j) \cdot u(\pi_{pj}, \pi_{-i}), \pi'_i \in \Delta(P)$.*

We denote a set of all NE of restricted game $G_R(G, P, 2)$ as: $\pi^*_R(G, P) = = \text{argmax}_{\pi_1 \in \Delta(A_1)} \min_{\pi_2 \in \Delta(P)} u(\pi_1, \pi_2)$

4.1. Strategy Optimization

In this paper, we focus on identifying the opponent's portfolio that enables the best performance in the original game, as in e.g. MVS [4]. We define this process within the normal form games and assume that strategy optimization is conducted for player 1 while player 2 is restricted by the portfolio. Initially, we are given a two-player zero-sum normal form game, $G = (N, A, u)$, and a portfolio P of size k. Then a restricted game, $G_R(G, P, 2)$, is constructed and one of its NE, π^*_R, is computed. Player 1 adopts this strategy in the original game G, while Player 2 plays best response, $BR_2(\pi^*_{R1})$. This defines portfolio utility, which says how little a player can lose by assuming the portfolio while facing a rational opponent. This process is illustrated in Figure 1.

Definition 3 *The set of utilities of portfolio P in a game G is the set of all the utilities that player 1 can get when assuming the portfolio: $U_P(G,P) = \{\min_{\sigma_2 \in \Delta(A_2)} u(\pi_1, \sigma_2) \mid \pi \in \pi^*_R(G,P)\}$.*

$U_P(G,P)$ is a set rather than a single value, as multiple equilibria can exist in the restricted game G_R. While they share the same utility in G_R, their utilities in the original game G may differ. The following definition allows us to specify which one is selected.

Definition 4 *Depending on the rule selecting the equilibrium in the restricted game there are the following utilities of portfolio P in the game G:*

- *Optimistic Utility of Portfolio $u_{OPT}(G,P) = \max_{\sigma \in \pi^*_R(G,P)} \min_{\sigma_2 \in \Delta(A_2)} u(\sigma_1, \sigma_2)$*
- *Pessimistic Utility of Portfolio $u_{PES}(G,P) = \min_{\sigma \in \pi^*_R(G,P)} \min_{\sigma_2 \in \Delta(A_2)} u(\sigma_1, \sigma_2)$*
- *Maximum Entropy Utility of Portfolio $u_{ME}(G,P) = \min_{\sigma_2 \in \Delta(A_2)} u(\pi^{ME}(G_R(G,P,2)), \sigma_2)$*
- *Algorithm A Utility of Portfolio $u_A(G,P) = \min_{\sigma_2 \in \Delta(A_2)} u(A(G_R(G,P,2)), \sigma_2)$, where algorithm A is an algorithm that computes a NE in a game*

Note that $u_{OPT}(G,P), u_{PES}(G,P), u_{ME}(G,P), u_A(G,P) \in U_P(G,P)$. For any game G and portfolio P, the portfolio utility u_P is bounded by optimistic and pessimistic utilities: $u_{OPT}(G,P) \geq u_P(G,P) \geq u_{PES}(G,P)$. However, computing these utilities requires knowledge beyond the restricted game. Nevertheless, we focus on the pessimistic case as it reflects the portfolio's performance in the worst-case scenario. We further discuss this issue in the Appendix D.

Since the objective is to find a strategy the closest to NE strategy, we want to know how much worse the player is for assuming the portfolio ($u_P(G,P)$) from playing without any restriction on actions ($u(\pi^*(G))$). In other words, we want to know the cost of using the portfolio:

Definition 5 *For a given portfolio utility definition u_P, a cost of portfolio P in the game G is defined as $C(G,P,u_P) = |u(\pi^*(G) - u_P(G,P)|$.*

The optimal portfolio is the one that minimizes that cost. We elaborate on it in the Appendix B.

4.2. Are Mixed Portfolios needed?

We observed that mixed portfolios are more expressive than pure portfolios of the same size.

Observation 6 *For any portfolio utility u_P, any game $G = (N,A,u)$ and portfolio size $k \leq |N|$, there exists mixed portfolio P_m of size k that's cost is at least as low as any pure portfolio P_p:*

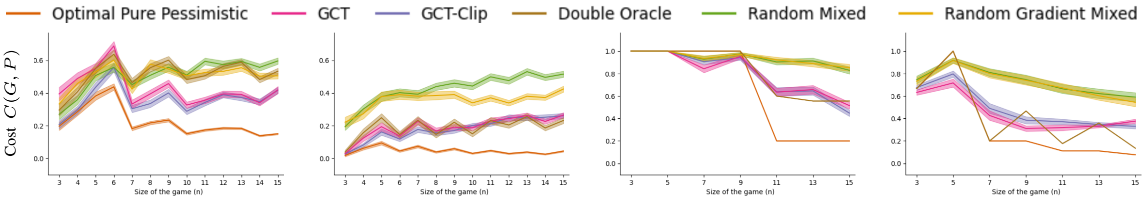
$$\exists_{P_m \subseteq \Delta(A_2)} \forall_{P_p \subseteq \Delta(A_2)} C(G, P_m, u_P) \leq C(G, P_p, u_P)$$

Theorem 7 *There exist games and portfolio sizes for which the above inequality is strict.*

We provide the proofs for Observation 6 and Theorem 7 in Appendix C.

5. Evaluation

The evaluation procedure starts with a portfolio P and a game G as input. Next, the NE π is found in the restricted game $G_R(G,P,2)$, based on the portfolio type. Once the equilibrium is identified, the best response for Player 2, $BR_2(\pi_1)$, is determined and its value, v_{br}, is computed. Afterwards, the value of the game G, v, is calculated. Finally, the evaluation returns the cost of portfolio $|v - v_{br}|$.



$(a)\ k_{rate} = 0.3$ $(b)\ k_{rate} = 0.5$ $(c)\ k_{rate} = 0.3$ $(d)\ k_{rate} = 0.5$

Figure 2: Figures (a) and (b) present results for the generalized RPS game (for 100 seeds). Figures (c) and (d) present portfolio results for random games. 100 different games were generated (seed 42), for each size n. In both experiments, Gradient-based methods use a scaling factor of 0.3. The average and standard error of pessimistic evaluation is shown.

5.1. Algorithm for pessimistic portfolio evaluation

We present the Mixed-Integer Linear Program (MILP) algorithm for the pessimistic evaluation of (mixed) portfolio $P : k \times |A_1|, k \in \{1, ..., |A_2|\}$ in the normal form game $G = (U, (A_1 \times A_2), \{1, 2\})$. For this program the value v_R of the restricted game $G_R(G, P, 2)$ needs to be provided. We assume that the utility matrix U is normalized so the values are from $[-1, 1]$. That allows us to use a big enough constant $M = 10$. Additionally, we calculate utility matrix $U_R : |A_1| \times k$ of G_R as $U_R = PU$. $e_i \in \{0, 1\}^{|A_2|}$ denotes vector with 1 on position i and 0 otherwise. Variables $x \in [0, 1]^{|A_1|}$ denote player 1's strategy and $v_o \in [min(U), max(U)]$ is the utility of portfolio. Lastly, binary variables $b \in \{0, 1\}^{|A_2|}$ signify which action is best response of player 2.

We make sure that strategy is a distribution (1) and that there is only one best response picked(2). We also fix player 1's strategy to be NE (3) and bound best response value v_o (4).

$$1)\sum_{i\in\{1,..,|A_1|\}} x_i = 1, \quad 2)\sum_{i\in\{1,..,|A_2|\}} b_i = 1, \quad 3) xU_R \geq v_r \mathbb{1}$$

$$4)\forall_{i\in\{1,...,|A_2|\}} : xUe_i \leq v_o + M(1 - b_i)$$

Finally, the objective is to minimize v_o.

6. Experiments

We employ methods relevant to the portfolio search problem, utilizing scalable heuristics that do not guarantee optimal portfolios. Additionally, we include a non-scalable optimal method.

Gradient Cluster Transformations (GCT) [10], operates in three stages. Firstly, it computes the gradients of an algorithm that approaches Nash Equilibrium, though full convergence is not required. Next, the gradients are normalized and clustered to form k distinct clusters, representing various directions in the strategy space. Finally, gradients are drawn from each of these k clusters, multiplied by a scaling factor c, and added to a grounding strategy, such as an approximate NE, to form the portfolio. We adapt GCT for the 2p0s NFG setting. Like the original authors, we use the Regularized Nash Dynamics (R-NaD) algorithm [17] [16] as the iterative algorithm. R-NaD performs efficiently on GPUs, making it well-suited for large imperfect information games. We use it with 10k gradients. Additionally, we modify the GCT method, which clips the gradients to be non-negative before normalization, we refer to it as GCT-Clip.

The Double Oracle (DO) [14] method iteratively expands each player's strategy subset by adding best responses to the opponent's current actions. We start with responses to the uniform strategy and continue until player 2's subset reaches the target portfolio size.

We use two random methods: Random Mixed Portfolio and Random Gradient Portfolio. The former generates random strategies with values drawn from uniform distribution and then normalized. The latter allows for a direct analysis of the impact of the gradients generated by GCT. It generates random gradients, which are normalized and added to the Nash Equilibrium strategy, scaled by a constant factor as in GCT. Together, these components form the portfolio.

Additionally, we compute the optimal pure pessimistic portfolio (OPPP) via brute force, enumerating all pure portfolios and evaluating them using the algorithm in Subsection 5.1.

7. Results

7.1. Random games

Random games [13] enable easy generation of games at any size. We create them by sampling utility integer values from $[-1e07, 1e07]$ and normalizing them to the range [-1, 1]. Empirically, these games typically have a single NE. We conducted experiments on games with action sizes $|A_1| = |A_2| \in \{3, \ldots, 15\}$, using portfolio size rates of 0.3 and 0.5. Portfolio size k is calculated as $k = \max(1, \lfloor k_{rate} \cdot n \rfloor)$, with $k = 2$ for $n = 3$ and $k_{rate} \geq 0.5$. Results in Figure 2 show a "zig-zag" pattern, caused by k increasing by 1. None of the methods outperform OPPP. For $k_{rate} = 0.3$ GCT-Clip outperforms GCT and DO. GCT outperformed DO for $n > 6$. However, for $k_{rate} = 0.5$ they all perform with similar quality. Non-random methods achieve lower costs for $k_{rate} = 0.5$, as larger portfolios retain more information. This effect is especially pronounced for DO, likely due to its iterative nature: with smaller portfolios, there's a lower chance of selecting the correct strategies. Random Gradient Mixed performs worse than GCT, indicating the usefulness of R-NaD gradients in GCT. A more detailed GCT study can be found in Appendix A.

7.2. Generalized Rock Paper Scissors

To assess methods on a game with a specific structure, we performed experiments on the generalized Rock Paper Scissors game, as it is a popular zero-sum game [12]. We use the generalization proposed by Gergely et al. [5], defined for odd sizes n with each action winning and losing exactly $\lfloor \frac{n}{2} \rfloor$ times. Uniform strategy is the only NE for all n. The results are presented in the Figure 2. Interestingly, in this game, the difference between GCT and GCT-Clip is different than in Random Games. Moreover, for some sizes, OPPP is reached by DO or even outperformed by GCT.

8. Conclusions

We formally defined portfolios and their optimization problem, analyzed their dynamics, and developed an algorithm to find optimal pure portfolios. Using this algorithm, we benchmarked computationally efficient methods and assessed their performance. While we observed improvements in the GCT method, it does not achieve the cost of the optimal pure portfolio, indicating room for further enhancement. Future work should focus on designing an algorithm for finding optimal mixed portfolios. It would also be beneficial to test GCT-Clip with SEPOT [10] in large games to assess it. Additionally, exploring other use cases for portfolios would be valuable.

References

[1] Nolan Bard, Michael Johanson, Neil Burch, and Michael Bowling. Online implicit agent modelling.

[2] Nolan DC Bard. Online agent modelling in human-scale problems. URL https://era.library.ualberta.ca/items/82723f7a-1a91-4613-acb5-109fe8de285c.

[3] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365 (6456):885–890, 2019. doi: 10.1126/science.aay2400. URL https://www.science.org/doi/abs/10.1126/science.aay2400.

[4] Noam Brown, Tuomas Sandholm, and Brandon Amos. Depth-limited solving for imperfect-information games. URL http://arxiv.org/abs/1805.08195.

[5] Mali Imre Gergely and Gabriela Czibula. Policy-based reinforcement learning in the generalized rock-paper-scissors game. *ESANN 2023 proceesdings*, 2023. URL https://api.semanticscholar.org/CorpusID:262073109.

[6] Andrew Gilpin and Tuomas Sandholm. Lossless abstraction of imperfect information games. 54. doi: 10.1145/1284320.1284324.

[7] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996. ISSN 0899-8256. doi: https://doi.org/10.1006/game.1996.0051. URL https://www.sciencedirect.com/science/article/pii/S0899825696900512.

[8] Christian Kroer and Tuomas Sandholm. Extensive-form game abstraction with bounds. In *Proceedings of the fifteenth ACM conference on Economics and computation*, EC '14, pages 621–638. Association for Computing Machinery, . ISBN 978-1-4503-2565-3. doi: 10.1145/2600057.2602905. URL https://dl.acm.org/doi/10.1145/2600057.2602905.

[9] Christian Kroer and Tuomas Sandholm. A unified framework for extensive-form game abstraction with bounds. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., . URL https://proceedings.neurips.cc/paper/2018/hash/aa942ab2bfa6ebda4840e7360ce6e7ef-Abstract.html.

[10] Ondřej Kubíček, Neil Burch, and Viliam Lisý. Look-ahead Search on Top of Policy Networks in Imperfect Information Games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 8 2024. doi: 10.24963/ijcai.2024/480. URL https://doi.org/10.24963/ijcai.2024/480.

[11] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. URL http://arxiv.org/abs/1711.00832.

[12] Marc Lanctot, John Schultz, Neil Burch, Max Olan Smith, Daniel Hennes, Thomas Anthony, and Julien Perolat. Population-based evaluation in repeated rock-paper-scissors as a benchmark for multiagent reinforcement learning, 2023. URL https://arxiv.org/abs/2303.03196.

[13] Chun Kai Ling, Fei Fang, and J. Zico Kolter. Large scale learning of agent rationality in two-player zero-sum games, 2019. URL https://arxiv.org/abs/1903.04101.

[14] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 536–543, 2003.

[15] Luis E. Ortiz, Robert E. Schapire, and Sham M. Kakade. Maximum entropy correlated equilibria. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 347–354, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL https://proceedings.mlr.press/v2/ortiz07a.html.

[16] Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, et al. From poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization. In *International Conference on Machine Learning*, pages 8525–8535. PMLR, 2021.

[17] Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623): 990–996, 2022.

[18] Muhammad Rahman, Jiaxun Cui, and Peter Stone. Minimum coverage sets for training robust ad hoc teamwork agents. 38(16):17523–17530. ISSN 2374-3468. doi: 10.1609/aaai.v38i16.29702. URL https://ojs.aaai.org/index.php/AAAI/article/view/29702. Number: 16.

[19] Tuomas Sandholm. Abstraction for solving large incomplete-information games. 29(1). ISSN 2374-3468. doi: 10.1609/aaai.v29i1.9757. URL https://ojs.aaai.org/index.php/AAAI/article/view/9757. Number: 1.

Appendix A. Qualitative analysis and GCT study

A.1. GCT study on random games

In order to measure how ”accidental” is GCT method and whether the gradients generated by it are useful in leading to the portfolio, we analyse the importance of the three stages of GCT: choosing the gradients(1), normalization of the gradients(2) and choosing a grounding strategy for the gradients(3).

To analyse the first stage and determine which gradients are relevant, we performed experiments to check the influence of the number of gradients used. The results are presented in the Figure 4. It can be observed that in general, GCT with only the first 500 or 1000 gradients achieves worse results. The difference grows with the portfolio size, as other gradients sizes improve the cost.

This proves that gradients from R-NaD from later iterations are useful. However, for a bigger number of gradients, it is not clear which one is the best. This suggests that gradients from the end do not contribute to finding good directions from the NE. That might be caused by making gradients more repetitive and losing more significant gradients. Or because gradients from later iterations are changing policy by a small difference, which can be not significant, especially when k is smaller. This is supported by the fact that the bigger k, the more cost decreases among the higher gradients number.

For the second stage of GCT, we test the importance of normalization of the gradients and gradient scaling factor. We compare no normalization, normalization of the length of the gradient and third variant (old/clipping) which is clipping the values of gradients to be from 0 to 1 before normalization. Results are presented in the Figure 3. It is apparent that normalization is necessary, which makes sense because some gradients might be significantly smaller than others, especially the ones nearer convergence. Additionally, clustering them with length is more messy(?) - it's introducing additional dimension. However, surprisingly the normalization method that clips the gradients to positive value firsts has better results. We also tested the influence of the clustering constant gradients' multiplication factor for 20 values from 0.01 to 1.0 and observed no difference in the results, therefore we set this constant to 0.3.

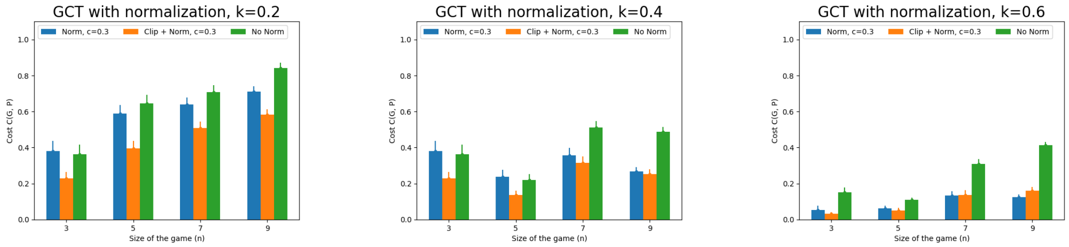
To test the importance of the grounding strategy for the GCT method we performed an experiment with the GCT gradients attached to different grounding strategies. We compare using different approximations of Nash Equilibrium(ϵ-Nash Equilibrium), depending on the ϵ_{ne}. Here, we take the strategy that is on the edge of ϵ-NE for player 1. Results are presented in the Figure 5.

It is visible that for $\epsilon_{ne} \leq 1.0$, the costs are almost the same. For bigger values, it is increasing the cost significantly. It is a positive outcome in the meaning that in terms of large imperfect information games, computing ϵ-NE with very small ϵ_{ne} is costly.

A.2. Qualitative analysis

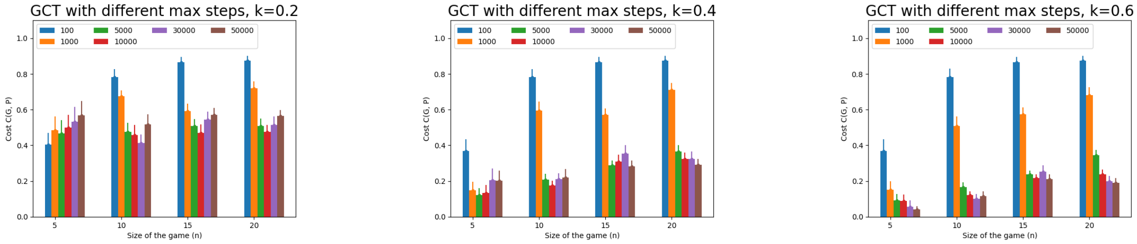
To understand portfolios more we performed qualitative analysis. Firstly, we look at the games in which GCT achieved (much) worse results than other methods, meaning a difference of at least 0.3.

When an action is dominated it results in the GCT method not using this action at all in the portfolio. On one hand, this is a desired feature as often dominated actions are not necessary in the portfolio. On the other hand, there are many games where the Optimal Pure Pessimistic Portfolio contains actions from outside of NE support (when the portfolio size is smaller or equal to the size



$(a)\ k_{rate} = 0.2$ $(b)\ k_{rate} = 0.4$ $(c)\ k_{rate} = 0.6$

Figure 3: Normalization of gradients in GCT method. Average and standard error from 50 random games is shown.



$(a)\ k_{rate} = 0.2$ $(b)\ k_{rate} = 0.4$ $(c)\ k_{rate} = 0.6$

Figure 4: Number of gradients in GCT method. The average and standard error of pessimistic evaluation from 20 random games is shown. Original normalization is used.

of the support). This results in many games in which the GPT portfolio performs much worse than the Optimal Pure Pessimistic Portfolio.

	B_1	B_2	B_3
A_1	-0.99	0.66	0.37
A_2	-0.15	-0.16	0.87
A_3	-1.00	0.48	-0.72

Table 1: Game of size 3.

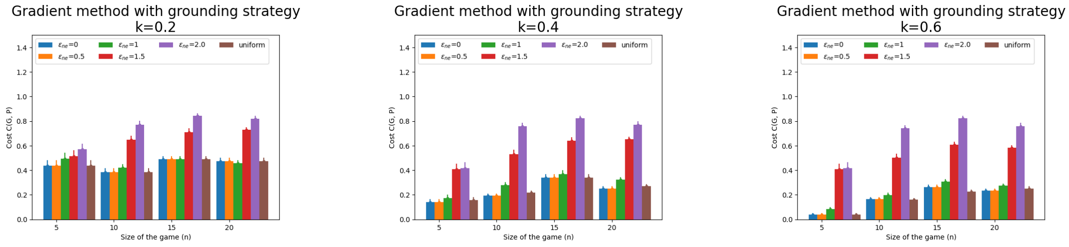
	B_1	B_2	B_3	B_4
A_1	0.38	0.60	0.56	-0.65
A_2	-0.29	-0.07	-0.00	-0.98
A_3	0.10	-0.74	0.52	0.39
A_4	0.90	0.52	-0.29	1.00

Table 2: Game of size 4.

Table 3: Games in which GCT was much worse than OPPP.

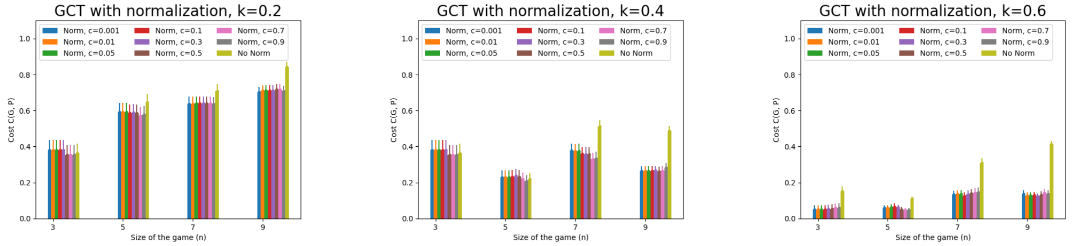
For game, with utility matrix presented in Table 1, and $k = 1$ GCT found portfolio $\{[0.431, 0.569, 0.000]\}$ with cost 0.837, whereas OPPP portfolio is $\{[1, 0, 0]\}$ with cost 0.007. For comparison, a portfolio containing strategy $\{[0.5, 0.5, 0.0]\}$, which is very close to NE has the same cost as OPPP.

For the next game 2, GCT found bad portfolios for both k_{rate}. For $k = 1$, OPPP has an action that is not in the support of NE while OPPP finds $[1, 0, 0, 0]$ with cost 0.517, while GCT achieves cost 0.971 and portfolio $\{[0, 0.198, 0.523, 0.279]\}$. For comparison, a portfolio containing strategy $\{[0.0, 0.2, 0.4, 0.4]\}$, which has the same cost as OPPP.



(a) $k_{rate} = 0.2$ (b) $k_{rate} = 0.4$ (c) $k_{rate} = 0.6$

Figure 5: Grounding strategy in the GCT method: Average and standard error of pessimistic evaluation over 50 random games are shown, using original normalization. For each ϵ_{ne}, the least beneficial strategy that is an ϵ_{ne}-NE was chosen. 'Uniform' refers to the uniform strategy.



(a) Original Normalization (b) Original Normalization (c) Original Normalization

Figure 6: GCT method with different scaling factors. Average with standard error on 50 random games each is presented. Original Normalization.

We now discuss the same game (Table 2) but for $k = 2$. OPPP found portfolio $\{[0, 1, 0, 0]$, $[0, 0, 0, 1]\}$ with cost 0.284. GCT found portfolio $\{[0.00, 0.176, 0.644, 0.180]$, $[0.00, 0.238, 0.340, 0.422]\}$ with cost 0.677.

Additionally, we look at two games. Firstly, in the Rock Paper Scissors game in which GCT found a portfolio
$\{[0.26, 0.24, 0.50], [0.41, 0.42, 0.17]\}$, with cost 0.48, which is better than pure optimal pessimistic but not optimal. Then we look at the games from example 7 in which GCT and DO both find optimal portfolios.

Appendix B. Details on optimizing portfolio cost

Definition 1 *For a utility portfolio u_P, the optimal portfolio of size k in the game G is a portfolio of player 2 defined as $P^* \in \operatorname{argmin}_{P \subseteq \Delta(A_2), |P|=k} |u(\pi^*(G)) - u_P(G,P)|$*

Next, we will demonstrate that the above optimization criterion can be simplified. To achieve this, we require the following lemma:

Lemma 2 *Utility u^* in the Nash Equilibrium of 2p0s game G is bigger or equal than any utility of portfolio $u_P(G,P) \in U_P(G,P)$: $u^* \geq u_P(G,P)$*

Proof Since G is a zero-sum game, for any strategy $\pi_1 \in \Delta(A_1)$ we have following inequality: $u* = u(\pi_1^*, \pi_2^*) \geq u(\pi_1, \pi_2^*) = -u_2(\pi_1, \pi_2^*)$. In particular, when $\pi_1 = \pi_{R1}^*$ we have that $u* \geq u(\pi_{R1}^*, \pi_2^*) = -u_2(\pi_{R1}^*, \pi_2^*)$. This is equivalent to:

$$-u^* \leq u_2(\pi_{R1}^*, \pi_2^*) \leq u_2(\pi_{R1}^*, BR_2(\pi_{R1}^*)) =$$
$$= -u(\pi_{R1}^*, BR_2(\pi_{R1}^*)) = -u_P(G,P) \in U_P(G,P)$$

Therefore $u^* \geq u_P(G,P)$, for all $u_P(G,P) \in U_P(G,P)$ ■

Now we introduce the simplification of the Definition 1.

Theorem 3 *For a portfolio utility u_P, the optimal portfolio of player 2 of size k is characterized by $P^* \in \operatorname{argmax}_{P \subseteq \Delta(A_2), |P|=k} u_P(G,P)$*

Proof Thanks to Lemma 2, the absolute value in the definition of cost can be removed: $C(G,P,u_P) = u^* - u_P(G,P)$. Moreover, game value u^* does not depend on portfolio, therefore, minimizing $C(G,P,u_P)$ is equivalent to minimizing $-u_P(G,P)$ or maximizing $u_P(G,P)$. ■

Appendix C. Proof of Observation 6 and Theorem 7

Observation 6 is trivially proven because mixed portfolios contain pure portfolios. Below we present the proof of the Theorem 7.

Proof

	R	**P**	**S**
R	0	-1	1
P	1	0	-1
S	-1	1	0

Table 4: Utility matrix of Rock Paper Scissors(RPS)

	R	**P**
R	0	-1
P	1	0
S	-1	1

Table 5: Utility matrix of G_R(RPS, $\{R, P\}, 2)$

	σ_1	σ_2
R	-0.5	0
P	0.5	-0.5
S	0	0.5

Table 6: Utility matrix of $G_R(RPS,$ $\{(0.5, 0.5, 0.0), (0.0, 0.5, 0.5)\}, 2)$

We prove the opposite by counterexample. We will show that for a specific game, there exists a mixed portfolio with a lower cost than any pure portfolio. We will show that for optimistic and pessimistic utility, which is enough because those types bound others. Let G be a Rock Paper Scissors game (Table 4) and portfolio size be $k = 2$. In this case, an optimal portfolio is any portfolio that contains two distinct actions, then without loss of generality $P_{pure} = \{(1, 0, 0), (0, 1, 0)\}$ is the optimal pure portfolio. Utility of restricted game $G_R(G, P_{pure}, 2)$ is presented in the Table 5. The optimistic and pessimistic NE both result in the following strategy of player 1: $\pi^*_{R1} = (0, \frac{2}{3}, \frac{1}{3})$ and best response of player 2 in the original game G to be $BR_2(\pi^*_{R1}) = (0, 0, 1)$. Therefore, the cost of this portfolio is: $C(G, P_{pure}, u_{P\_OPT}) = C(G, P_{pure}, u_{P\_PES}) = |0 - \frac{2}{3}| \approx 0.67$

Now, we will show a mixed portfolio of size $k = 2$, that has a lower cost for this game than 0.67 with $P_{mixed} = \{(0.5, 0.5, 0), (0., 0.5, 0.5)\}$. Utility of restricted game $G_R(G, P_{mixed}, 2)$ is presented in the Table 6. Again, the optimistic and pessimistic NE both result in the following strategy of player 1: $\pi^*_{R1} = (0, \frac{1}{3}, \frac{2}{3})$, with best response of player 2 to be $BR_2(\pi^*_{R1}) = (0, 0, 1)$. Therefore the cost of this portfolio is: $C(G, P_{mixed}, u_{P\_OPT}) = C(G, P_{mixed}, u_{P\_PES}) = |0 - \frac{1}{3}| \approx 0.33$

■

Appendix D. Discussion about different utilities of portfolio

	B_1	B_2	B_3	B_4
A_1	0	0	1	-1
A_2	0	0	-b	1

Table 7: Utility matrix of a game with ambiguous optimal optimistic portfolios. $b > 1$ is parameter.

While the optimal portfolio with optimistic utility is easier to find, it may not be useful for strategy calculation. Since the NE selected in the restricted game is always beneficial for player 1, the portfolios are chosen ignoring the difficulty in finding it, promoting ambiguous restricted games. A portfolio carrying less information (not capturing full-game dynamics) can make it harder to identify the correct NE with a method that relies solely on the restricted game.

As an example, consider a game G with the utility matrix in Table 7 and $b > 1$. The NE for G is $\pi^* = \{(\frac{1+b}{3+b}, \frac{2}{3+b}), (0, 0, \frac{2}{3+b}, \frac{1+b}{3+b})\}$. The desired portfolio of size $k = 2$ should include actions B_3 and B_4, as they are necessary to identify the NE, while actions B_1 and B_2 are redundant. However, both $\{B_3, B_4\}$ and $\{B_1, B_2\}$ are optimal optimistic portfolios, because in the restricted game $G_R(G, \{B_1, B_2\}, 2)$, any strategy is a NE so π_1^* can be chosen. Unfortunately, there is no guarantee the method using only the restricted game will find π_1^*. Even if it did for one value of b, it would fail for others, leading to portfolio exploitability. Thus, the optimistic portfolio concept is not ideal, as it does not capture what we are truly aiming to optimize.

Therefore, the optimal pessimistic portfolio is more relevant, as it guarantees a maximum cost regardless of the method. However, its optimization formula is more complex, making it harder to find the optimal algorithm. If the NE-finding method is known, such as Regret Matching+, computation becomes more precise. This simplifies the process but reduces generality.

On the other hand, Maximum Entropy Equilibrium can be useful when an equilibrium must be chosen, as it's easier to compute than the pessimistic version and doesn't require full game information. It's also method-independent, making it more general, though it's uncertain if the chosen NE-finding method will yield a similar result.