

MỘT HƯỚNG TIẾP CẬN CỦA THUẬT TOÁN FICTITIOUS PLAY ĐỐI VỚI BÀI TOÁN PHÂN BỐ NGUỒN LỰC

Trịnh Bảo Ngọc, Huỳnh Quyết Thắng, Lê Công Thành, Lê Bá Trường Giang, Trần Quang Huy
Viện Công nghệ thông tin và truyền thông, Đại học Bách khoa Hà Nội

ngoctb@hanu.edu.vn, thanghq@soict.hust.edu.vn, thanhcls1316@gmail.com, giangpna98@gmail.com,
20164778@student.hust.edu.vn

TÓM TẮT: Phân bố nguồn nhân lực dự án là quá trình cân đối lại các nguồn lực trong thời gian thực hiện dự án, được thực hiện thường xuyên và chính xác nhằm giải quyết các vấn đề xung đột của dự án, vấn đề xảy ra với nguồn lực dự án ảnh hưởng tới tất cả các công việc của dự án. Chính vì sự quan trọng như vậy nên các vấn đề nội tại của Phân bố nguồn lực đáng được đem ra cân nhắc và tìm cách giải quyết. Mục tiêu của nghiên cứu này nhằm đưa ra mô hình toán học hiệu quả cho bài toán phân bố nguồn lực dưới dạng lý thuyết trò chơi thông qua việc tìm điểm cân bằng Nash. Từ mô hình đó, ta đưa ra phân bố xác suất của các chiến lược lựa chọn được đưa ra trong quá trình phân bố nguồn lực. Từ các số liệu đó, các tổ chức, công ty có thể sử dụng, tham khảo để quá trình phân bố nguồn lực diễn ra hiệu quả. Chúng tôi đề xuất một phương án mở rộng thuật toán Fictitious Play để phù hợp với mô hình bài toán phân bố nguồn lực cũng như nêu lên một số khó khăn của hướng đi này với bộ dữ liệu lớn

Từ khóa: Thuật toán Fictitious Play, Cân bằng Nash, Phân bố nguồn lực, Thuật toán CFR, Thuật toán CFR Plus, Lý thuyết trò chơi.

I. GIỚI THIỆU

Phân bố nguồn nhân lực là quá trình cân đối lại các nguồn lực trong suốt quá trình một tổ chức tồn tại và phát triển. Nó là một trong các nội dung quan trọng và là điều kiện cần thiết để đảm bảo tính cân đối. Phân tích nguồn lực là nội dung quan trọng trong quản trị chiến lược. Phân bố nguồn lực hợp lý là cơ sở để thực hiện các mục tiêu chiến lược một cách có hiệu quả. Đồng thời với đó, việc phân bố nguồn lực cần được thực hiện một cách công bằng trước nhu cầu của các bên. Phân phối nguồn lực một cách ngẫu hứng, thiếu căn cứ khoa học sẽ dẫn đến tình trạng lãng phí, kém hiệu quả trong việc sử dụng các nguồn lực và điều này sẽ dẫn đến việc thực hiện thất bại các mục tiêu đã đề ra. Chính vì tầm quan trọng như vậy, bài toán tối ưu hóa phân bố nguồn lực thường xuyên được đặt ra và việc giải quyết nó một cách tốt nhất đóng một vai trò quyết định trong hoạt động của tổ chức. Lấy ví dụ, các nguồn lực trong một công ty có thể được kể đến như: thông tin, tài chính, nhân sự, cơ sở vật chất, khách hàng, nhà cung cấp, công nghệ sử dụng, năng lực quản trị, năng lực kinh doanh, thương hiệu, uy tín, ... Trong bài báo này, chúng tôi phân tích và giải quyết vấn đề thông qua việc đảm bảo lợi ích cân bằng giữa các bên trong quá trình phân bố nguồn lực.

Bài toán phân bố nguồn lực có thể được giải quyết bằng cách sử dụng các mô hình trong lý thuyết trò chơi. Trong đó, các thành phần của tổ chức đóng vai trò như người chơi, còn nguồn lực mà họ được phân bổ thể hiện giá trị lợi ích mà họ thu được. Mỗi thành phần đều cố gắng giành được nhiều lợi ích nhất có thể và vì thế có thể xảy ra cạnh tranh lẫn nhau.

Thuật toán Fictitious Play, được giới thiệu lần đầu bởi Brown (1951), là một mô hình học phổ biến trong lý thuyết trò chơi [1, 2]. Trong thuật toán, hai người chơi thay phiên nhau thực hiện các hành động trong trò chơi, tại mỗi vòng lặp, mỗi người chơi sẽ chọn hành động đem lại lợi ích lớn nhất dựa trên tập các hành động trước đó của đối thủ. Trung bình của dãy các chiến lược, được lựa chọn, của mỗi người chơi sẽ hội tụ về điểm cân bằng Nash [2]. Tuy nhiên, thuật toán Fictitious Play có một số hạn chế: thứ nhất thuật toán bổ sung các chiến lược mới không tận dụng đầy đủ thông tin sau mỗi vòng lặp, theo đó vai trò của chiến lược mang lại nhiều lợi ích và ít lợi ích là không khác biệt; thứ hai thuật toán chỉ hỗ trợ cho mô hình hai người chơi, với sự có mặt của người chơi thứ ba, thuật toán không thể sử dụng được.

Hạn chế thứ nhất được khắc phục bằng các biến thể CFR và CFR+, được cải tiến từ thuật toán Fictitious Play gốc [6, 7]. Trong đó, một ma trận bổ sung để lưu trữ thông tin qua mỗi vòng lặp và đánh giá ảnh hưởng từ mỗi chiến lược mà người chơi chọn. Từ đó, tốc độ hội tụ của thuật toán được đẩy nhanh. Tuy nhiên cả hai biến thể này vẫn không thể giải quyết được hạn chế thứ hai. Để đối mặt với hạn chế đó, nhóm chúng tôi đề xuất mô hình mới, dựa trên ba thuật toán nêu trên, Fictitious Play, CFR, CFR+. Trong mô hình đề xuất, số người chơi tham gia được nâng lên n ($n > 2$).

Quá trình cập nhật dãy chiến thuật được thực hiện trên một ma trận payoff, thể hiện sự tương tác của từng chiến thuật với toàn bộ $n - 1$ người chơi còn lại.

Trong bài báo này, chúng tôi trình bày một hướng tiếp cận bài toán phân bố nguồn lực thông qua mô hình điểm cân bằng Nash của lý thuyết trò chơi, đồng thời giải quyết vấn đề này thông qua việc mở rộng thuật toán Fictitious Play cũng như Fictitious Play sử dụng ma trận CFR và CFR+ (các bản cải tiến hiệu quả của thuật toán nguyên bản). Trong hướng tiếp cận trên, chúng tôi mở rộng thuật toán Fictitious Play và các biến thể CFR, CFR+, bằng cách tăng số lượng người chơi tham gia. Chi tiết về mô hình mở rộng sẽ được trình bày trong mục C. Dựa vào dữ liệu thu được trong quá trình thực nghiệm, chúng tôi đánh giá và so sánh kết quả của thuật toán Fictitious Play với hai biến thể tương ứng là

CFR và CFR+. Một đánh giá khác cũng được đưa ra về tốc độ hội tụ của thuật toán theo số chiến lược cơ sở của người chơi. Toàn bộ phần đánh giá sẽ được trình bày cụ thể trong mục D. Mục E sẽ tổng hợp lại thành tựu thu được từ hướng mở rộng, đưa ra một số khó khăn còn vướng mắc và phương hướng khắc phục trong tương lai.

II. MÔ HÌNH HOÁ BÀI TOÁN PHÂN BỐ NGUỒN LỰC

2.1. Bài toán xung đột phân bố nguồn lực

Trong quản lý dự án, các nguồn lực có thể là: tiền, nhân sự, máy móc thiết bị, cơ sở vật chất cần được phân phối cho các nhóm, bộ phận khác nhau để thực hiện công việc dự án. Trong việc phân bổ, tiêu chí phụ thuộc vào đề xuất từ các bộ phận, sự cân nhắc của quản lý dự án, tình hình dự án hoặc nhiều nhân tố khác. Vậy nên có những tranh cãi, xung đột trong việc đòi hỏi nhiều tài nguyên hơn từ nhiều bộ phận.

Việc xác định các tiêu chí và đưa ra phương án tốt để phân chia nguồn lực là một công việc cần thiết cho quản lý dự án hỗ trợ trong việc ra quyết định. Với đặc thù các xung đột trong dự án, nơi mà từng bộ phận vẫn có thể sống sót và tiếp tục thực hiện nghĩa vụ của mình với nhiều mức độ tài nguyên cung cấp, nhưng rõ ràng là với các kết quả chất lượng khác nhau, có nghĩa là từng bộ phận - người chơi trong game - có những chiến lược riêng trong thực hiện game.

Để giải quyết bài toán, chúng ta phải định nghĩa đặc thù của từng bộ phận, các tiêu chí để đánh giá trong việc giúp người chơi lựa chọn chiến lược của mình - cách tài nguyên được phân phối. Khi đó, sẽ dễ hơn xác định 1 Nash Equilibria bằng 1 thuật toán tối ưu đa mục tiêu. Fictitious Play là một thuật toán dựa trên nền tảng toán học, mỗi vòng của thuật toán, các người chơi sẽ quyết định chiến lược của mình dựa theo các dữ liệu về các bước đi trước của các người chơi khác, với điều kiện tiên quyết là các chiến lược của người chơi được xác định trước và không thay đổi trong quá trình chơi. Điều này là phù hợp với bài toán được giao và có khả năng thực thi được.

2.2. Đầu vào bài toán

Đầu vào của bài toán là các nguồn lực cần được phân bổ trong dự án như: tiền, nhân sự, máy móc thiết bị, cơ sở vật chất. Thông qua các chiến lược của các nhóm, sự cạnh tranh và các yếu tố liên qua, chúng ta có thể thiết lập được các hàm mang tính tương đối để thể hiện sự tương quan giữa chiến thuật của các nhóm. Từ đó, chúng ta sẽ đưa ra được ma trận payoff. Với trường hợp tổng quát, với mỗi người chơi, ma trận payoff sẽ có n chiều ở đó mỗi chiều là một trong số các chiến thuật khả thi của từng nhóm.

2.3. Mô hình hoá bài toán

Đối với bài toán phân bố nguồn lực, giả sử rằng tổ chức T có m nguồn lực cần phân bổ a_1, a_2, \dots, a_m cho n đối tượng b_1, b_2, \dots, b_n . Khi đó, chúng ta có mô hình bài toán trong lý thuyết trò chơi như sau:

Bài toán gồm n người chơi b_1, b_2, \dots, b_n . Với mỗi b_i với $i = \overline{1, k}$ (k là số chiến thuật có thể của mỗi người chơi). Với mỗi b_i , ta định nghĩa $P_i = (p_{i1}, p_{i2}, \dots, p_{ik})$ là vector xác suất của chiến thuật mà người chơi b_i sử dụng. Thông qua ma trận payoff M được xây dựng dựa trên S_{ij} và các nguồn lực a_1, a_2, \dots, a_n , để giải quyết bài toán cân bằng nguồn lực, chúng ta cần tìm ra các bộ P_i thoả mãn điểm cân bằng Nash của bài toán - là bộ các vector P_i sao cho đối với từng người chơi đây là chiến thuật đem lại lợi ích tốt nhất cho họ cho dù những người chơi còn lại thực hiện bất kỳ chiến thuật nào [1].

III. GIẢI PHÁP ĐỀ XUẤT

Thuật toán Fictitious Play nguyên bản được sử dụng để giải quyết các bài trò chơi - 2 người nhằm để tìm ra điểm cân bằng Nash. Tuy nhiên trong thực tế, bài toán phân bố nguồn lực có nhiều thành phần hơn. Đó là động lực cho nhóm chúng tôi nghiên cứu, tìm kiếm hướng phát triển để giải quyết vấn đề khi số người chơi lớn hơn hai. Dựa vào ý tưởng cốt lõi của thuật toán, chúng tôi đề xuất giải thuật mở rộng của thuật toán Fictitious Play cho n người chơi để giải quyết bài toán phân bố nguồn lực.

3.1. Giải thuật Fictitious Play mở rộng cho n người chơi

Thuật toán Fictitious Play nguyên bản [3, 4] được sử dụng để giải quyết các bài trò chơi có 2 người nhằm để tìm ra điểm cân bằng Nash. Tuy nhiên trong thực tế, bài toán phân bố nguồn lực có nhiều thành phần hơn. Đó là động lực cho nhóm chúng tôi nghiên cứu, tìm kiếm hướng phát triển để giải quyết vấn đề khi số người chơi lớn hơn hai. Dựa vào ý tưởng cốt lõi của thuật toán, chúng tôi đề xuất giải thuật mở rộng của thuật toán Fictitious Play cho n người chơi để giải quyết bài toán phân bố nguồn lực.

Fictitious Play

Trong phần này, chúng tôi có ý tưởng mở rộng thuật toán Fictitious Play cho n người chơi. Thuật toán được trình bày trong **Thuật toán 1** trong đó:

- Mảng $payoff[player][i][s_1] \dots [s_{n-1}]$ biểu thị giá trị lợi ích của người chơi ($player$) ở chiến thuật khi các người chơi ($player + i$) sử dụng các chiến thuật s_i ;
- Mảng $Strategy[player][bestAction]$ là mảng thể hiện số các lần được chọn của từng chiến thuật qua các thể hệ đã qua;
- Mảng $Ns[player][i]$ được lấy thông qua hàm $getNormalizeStrategy()$ thể hiện xác suất được lựa chọn của chiến thuật i của người chơi $player$ sau các thể hệ đã qua.

Thuật toán 1: Fictitious Play Algorithm

Input: n (Số người chơi), $player$ (người chơi), k (số chiến thuật của mỗi người chơi)

```

1: Khởi tạo  $maxSum = -\infty, bestAction = -1$ 
2:  $ns[][] = getNormalizeStrategy()()$ 
3: for 1 to k do
4:    $sum = \sum(\prod_{i=1}^{n-1} (ns[player + i][s_i] * payoff[player][i][s_1] \dots [s_{n-1}]))$ 
5:   if  $sum > maxSum$  do
6:      $maxSum = sum$ 
7:      $bestAction = i$ 
8: end for
9:  $Strategy[player][bestAction] += 1$ 
Độ phức tạp  $O((n - 1)^k)$ 

```

Fictitious Play sử dụng kỹ thuật CFR

Bên cạnh thuật toán nguyên bản, Fictitious Play thường được biết đến với một phiên bản cải tiến hơn khi sử dụng ma trận CFR [5] (Counterfactual Regret Minimization) - một kỹ thuật để giải quyết các trò chơi dạng mở rộng với thông tin không hoàn hảo. Thuật toán được trình bày trong **Thuật toán 3** trong đó:

- Mảng $payoff$ vs $strategy$ giống như phần 1.a;
- Mảng $cs[][]$ được xây dựng dựa trên mảng $strategy[]$ vs $cfr[]$ sẽ được trình bày trong **Thuật toán 2**.

Thuật toán 2: Xây dựng mảng $cs[][]$

Input: n (Số người chơi), k (Số chiến thuật của mỗi người chơi)

```

1: Khởi tạo  $cs[][]$ 
2: For 1 to n do
3:    $sum = \sum_{j=0}^k (Max(0, cfr[i][j]))$ 
4: End for
5: If  $sum > 0$  do
6:    $cs[i][j] = \frac{Max(0, cfr[i][j])}{sum}$ 
7: else  $cs[i][j] = \frac{1}{k}$ 
8: return  $cs$ 
Độ phức tạp  $O(k^n)$ 

```

Fictitious Play sử dụng kỹ thuật CFR+

Kỹ thuật CFR+ được giới thiệu trong bài báo “Solving Large Imperfect Information Games Using CFR+” của Oskari Tammelin năm 2014 [6] như một biến thể tốt hơn của kỹ thuật CFR. Thuật toán sẽ được trình bày cụ thể trong **Thuật toán 4**, trong đó:

- Mảng $payoff$ vs $strategy$ giống như phần 1.a;
- Mảng $cs[][]$ được xây dựng dựa trên mảng $strategy[]$ vs $cfr[]$ sẽ được trình bày trong **Thuật toán 2**.

Thuật toán 3: Fictitious Play - CFR

Input: n (Số người chơi), k (Số chiến thuật), $player$ (người chơi thực hiện thuật toán)

```

1: Khởi tạo  $maxSum = -\infty, bestAction = -1$ 
2:  $ns[][] = getCurrentStrategy()()$ 
3: for 1 to k:

```

```

4:    $cfu[i] = \sum (\prod_{i=1}^{n-1} (ns[(player + i) \% n][s_i]) * payoff[player][i][s_1] \dots [s_{n-1}])$ 
5:    $ev += cs[player][i] * cfu[i]$ 
6: end for
7: for 1 to k:
8:    $cfr[player][i] += cfu[i] - ev$ 
9:    $strategy[player][i] += cs[player][i]$ 
10: end for
Độ phức tạp  $O((n - 1)^k)$ 

```

Thuật toán 4: Fictitious Play - CFR+

Input: n (Số người chơi), k (Số chiến thuật), $player$ (người chơi thực hiện thuật toán)

```

1: Khởi tạo  $maxSum = -\infty$ ,  $bestAction = -1$ 
2:  $ns[][] = getCurrentStrategy()()$ 
3: for 1 to k:
4:    $cfu[i] = \sum (\prod_{i=1}^{n-1} (ns[(player + i) \% n][s_i]) * payoff[player][i][s_1] \dots [s_{n-1}])$ 
5:    $ev += cs[player][i] * cfu[i]$ 
6: end for
7: for 1 to k:
8:    $cfr[player][i] = Max(0, cfr[player][i] + cfu[i] - ev)$ 
9: end for
10: if  $iterationCount > delay$  do
11:    $w = 2^{wmode}$ 
12: else  $w = 0$ 
13: for 1 to k do
14:    $strategy[player][i] += cs[player][i] * w$ 
Độ phức tạp  $O((n - 1)^k)$ 

```

Trong thuật toán đề xuất trên, chúng tôi sử dụng ma trận payoff n chiều với n là số người chơi tham gia, thay vì ma trận hai chiều. Ma trận này cho phép sử dụng miêu tả đầy đủ giá trị nhận được của một người chơi nhận được tương ứng với mỗi hành động cụ thể của các người chơi còn lại.

Thuật toán 2 là bước cập nhật chiến thuật mới và chỉ xuất hiện trong thuật toán CFR và CFR+. Trong thuật toán này, mảng cs đóng vai trò như thành phần để bổ sung vào dãy chiến lược trước đó và được tính toán dựa trên ma trận payoff của người chơi tương ứng. Thuật toán cho thấy, giá trị lợi ích đạt được từ một chiến lược tỷ lệ thuận với đóng góp của nó trong lần cập nhật tiếp theo.

3.2. Phương án lựa chọn chiến thuật

Đối với trường hợp bài toán có n người chơi và có k chiến thuật, khi đó, ở mỗi vòng lặp bước ở thuật toán, thuật toán thực hiện tối thiểu $k \cdot n \cdot k^n$ bước làm do đó độ phức tạp của thuật toán tối thiểu sẽ là $O(n \cdot k^n)$. Với độ phức tạp $O(n \cdot k^n)$, nếu k lớn, tốc độ của thuật toán sẽ vô cùng chậm, thậm chí vượt quá khả năng xử lý dữ liệu của các máy tính. Do đó, việc hạn chế k , thông qua các chiến lược lựa chọn chiến thuật là cần thiết để cải thiện khả năng xử lý của thuật toán.

Giả sử có m nguồn tài nguyên là a_1, a_2, \dots, a_m có k chiến thuật s_1, s_2, \dots, s_k . Khi đó giả sử chiến thuật s_i có giá trị tài nguyên là $a_j[i]$ với tài nguyên a_j . Khi đó đặt $value[i] = f(a_1[i], a_2[i], \dots, a_k[i])$ là hàm biểu thị giá trị của chiến thuật s_i . Chiến lược lựa chọn các chiến thuật thực hiện theo thuật toán 5 sau trong đó $delta[i]$ là giá trị độ lệch của $value[i]$ với giá trị trung bình, $rank[i]$ là xếp hạng của chiến thuật s_i theo giá trị $delta$, x là số lượng chiến thuật chúng ta muốn chọn.

Thuật toán 5:

```

1: Khởi tạo  $rank[i] = i$ 
2:  $\overline{value} = \frac{\sum_j^m (value[j])}{m}$ 
3:  $\Delta_i = |value[i] - \overline{value}|$ 
4: Cập nhật  $rank[]$  theo mảng  $\Delta$ 
5: Loại bỏ các chiến thuật  $s_k$  nếu  $rank[k] > x$ 

```

Thuật toán 5 tính toán những giá trị lợi ích mà một chiến lược mang lại cho người chơi khi mà chưa xem xét đến tương tác với các người chơi còn lại. Mục đích nhằm dự đoán các chiến lược mà không có lợi cho quá trình tìm kiếm lời giải.

Thuật toán trên loại bỏ những chiến lược mà giá trị, nằm của chúng nằm quá xa so với giá trị trung bình và vì thế không đem lại hiệu quả trong việc tìm kiếm lời giải bài toán. Bằng cách không xem xét những chiến lược này, kích thước không gian tìm kiếm giảm xuống nhanh chóng.

IV. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1. Dữ liệu thực nghiệm

Trong bài báo này, chúng tôi thực nghiệm với các phương án thực nghiệm sau: với bộ dữ liệu ở bảng 1, 2 và các tham số đầu vào trong thuật toán ở bảng 3.

Bảng 1. Dữ liệu nhân sự của dự án

STT	Tên	Năng lực	Lương(triệu)	Kỹ năng
1	Hà Quang Minh	1	17,0	PHP, JavaScript ,Python
2	Bùi Tuấn Anh	1	15,0	MySQL, C++, C#
3	Vũ Minh Tuấn	2	15,0	Python, Java, C++
4	Trần Quang Đức	2	9,5	Java, C#
5	Nguyễn Thị Tuyết Nhung	3	9,5	PHP, JavaScript
6	Nguyễn Thị Hải Yến	4	9,5	MySQL, C#
7	Kiều Thanh Hải	5	9	MySQL, C++
8	Hà Vĩnh Lăng	6	9	Java, C++
9	Kiều Xuân Việt	7	9	JavaScript, Python

Bảng 2. Yêu cầu kỹ năng của từng dự án

Dự án	Yêu cầu kỹ năng (cần một trong số các kỹ năng)
1	PHP, MySQL, JavaScript
2	JavaScript, Python, Java
3	Java, C++, C#

Bảng 3. Các tham số đầu vào

Tham số	Giá trị
Số người chơi (n)	3
Số chiến thuật	5, 20
wmode	0
delay	0
epsilon(e)	0,0001
time (t)	0,40(s)

4.2. Môi trường thực nghiệm

- Hệ điều hành: MacOS Sierra 10.12.6;
- Chip: 2.2 GHz Intel Core i7;
- Máy: Macbook Pro Mid 2015.

4.3. Kết quả thực nghiệm

Kết quả thực nghiệm được thực hiện ở trong bảng 4, bảng 5. Bảng 4 biểu diễn kết quả so sánh giữa 3 thuật toán kết quả được ghi nhận khi giá trị epsilon đạt ngưỡng 0.0001 và số lượng chiến thuật được giới hạn bằng 5. Kết quả cho thấy thuật toán CFR+ có tốc độ hội tụ nhanh vượt trội, trong khi thuật toán Fictitious Play là tồi nhất. Điều này là hệ quả của sự khác biệt trong quá trình cập nhật dãy chiến thuật từ các dữ liệu sau mỗi vòng lặp. CFR+ tận dụng tối đa thông tin thu được dựa vào ảnh hưởng của từng chiến thuật lên từng người chơi cho và cho tốc độ hội tụ nhanh nhất.

Bảng 5 ghi nhận kết quả tại thời điểm 0,40 s, đánh giá giá trị epsilon giữa các thuật toán và giữa hai mức giới hạn số lượng chiến thuật là 5 và 20. Việc tăng số lượng chiến thuật khá dễ cho người chơi dẫn đến dữ liệu cần tính toán lớn hơn. Do đó, thời gian để hoàn thành vòng lặp cũng tăng theo và kết quả là tốc độ hội tụ của thuật toán vì như vậy mà giảm đi đáng kể.

Bảng 4. Bảng kết quả thực nghiệm

Số vòng lặp	Fictitious Play		CFR		CFR +	
	t(s)	epsilon	t(s)	epsilon	t(s)	Epsilon
1	0,00	0,70685	0,00	0,27692	0,00	0,27692
100	0,01	0,03049	0,01	0,01579	0,01	0,01802
200	0,02	0,01516	0,02	0,00775	0,02	0,00913
300	0,02	0,01010	0,02	0,00513	0,02	0,00579
400	0,02	0,00977	0,02	0,00384	0,02	0,00443
500	0,03	0,00764	0,03	0,00306	0,03	0,00385
1000	0,04	0,00364	0,04	0,00233	0,04	0,00192
5000	0,06	0,00085	0,06	0,00074	0,09	0,00049
10000	0,08	0,00053	0,08	0,00047	0,11	0,00026
20000	0,11	0,00035	0,13	0,00027	0,14	0,00015
30000	0,14	0,00025	0,17	0,00021	0,19	0,00011
32814	x	x	x	x	0,20	0,00010
40000	0,17	0,00020	0,21	0,00019	x	X
50000	0,19	0,00018	0,26	0,00019	x	X
79453	x	x	0,36	0,00010	x	X
100000	0,32	0,00010	x	x	x	X
100995	0,32	0,00010	x	x	x	X

Bảng 5. Bảng kết quả thực nghiệm giới hạn thời gian chạy là 0,4 s

Số chiến thuật lựa chọn	Fictitious Play		CFR		CFR +	
	t(s)	epsilon	t(s)	epsilon	t(s)	epsilon
5	0,4	0,00010	0,4	0,00008	0,4	0,00009
20	0,4	0,02565	0,4	0,04491	0,4	0,02777

4.4. Đánh giá

Từ bảng kết quả thực nghiệm, chúng ta thấy với cả ba thuật toán tốc độ hội tụ khá nhanh với trường hợp các chiến thuật được đánh giá và thu gọn lại về 5. Trong đó, thuật toán CFR+ có tốc độ hội tụ nhanh nhất sau đó đến Fictitious Play và CFR. Đối với trường hợp số lượng chiến thuật được giữ nguyên (lấy toàn bộ các bộ chiến thuật có thể có) thuật toán cho thấy sự hội tụ tương đối chậm khi trả ra kết quả không được tốt.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi đã đề xuất ra một mô hình mở rộng thuật toán Fictitious Play và các biến thể để giải quyết bài toán phân bổ nguồn lực. Trong thuật toán nguyên bản, Fictitious Play được áp dụng cho bài toán trò chơi với hai người, tuy vậy áp dụng cho bài toán phân bổ nguồn lực (khi mà số người chơi lớn), thuật toán mở rộng lấy tư tưởng chủ đạo từ thuật toán nguyên bản, vẫn cho thấy sự hiệu quả trong quá trình giải quyết bài toán với bộ dữ liệu không quá lớn. Tuy vậy, với trường hợp dữ liệu lớn khi tổ hợp các chiến thuật nhiều, thuật toán có độ phức tạp tương đối cao, cũng như tốc độ hội tụ tương đối thấp, do đó thuật toán cần kết hợp sử dụng một chiến lược kết hợp để đánh giá và rút gọn số lượng các chiến thuật một cách hợp lý.

Mặc dù hướng tiếp cận này cho thấy tiềm năng phát triển nhưng vẫn còn nhiều khó khăn cần được giải quyết. Khi số lượng người chơi gia tăng cũng như bộ dữ liệu đầu vào lớn, thuật toán có độ phức tạp cũng như thời gian tính toán tăng nhanh (cấp số mũ theo số người chơi) do kích thước lớn của ma trận payoff. Do đó việc đề ra chiến lược để khống chế kích thước của ma trận payoff là cần thiết để đảm bảo tốc độ cũng như kết quả của thuật toán. Bên cạnh đó, vấn đề xây dựng ma trận payoff cũng cần được quan tâm và nghiên cứu để đảm bảo tính chính xác tương đối của ma trận payoff so với thực tế. Việc xây dựng ma trận payoff cần được cân nhắc và tính toán trong từng trường hợp cụ thể để đảm bảo độ chính xác của thuật toán.

VI. TÀI LIỆU THAM KHẢO

- [1] Martin J. Osborne (2003). "An Introduction to Game Theory". Oxford University Press; 1st edition, ISBN-13: 978-0195128956, 560 pages.
- [2] Johannes Heinrich, Marc Lanctot, David Silver (2015). "Fictitious Self-Playing Extensive-Form Games". Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

- [3] Enrico H. Gerding, Zinovi Rabinovich Andrew Byde, Edith Elkind (2008). “Approximating Mixed Nash Equilibria using Smooth Fictitious Play in Simultaneous Auctions”. In Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems, Volume 3, pp.1577-1580.
- [4] Theodore J. Lambert III, Marina A. Epelman, Robert. Smith (2005). “A fictitious play approach to large-scale optimization”. Journal of Operations Research, Volume 53 Issue 3, May 2005, pp.477-489.
- [5] Andrew Gilpin, Samid Hoda, Javier Pena, and Tuomas Sandholm (2007). “Gradient-based Algorithms for Finding Nash Equilibria in Extensive Form Games”. In: Deng X., Graham F. C. (eds) Internet and Network Economics. WINE 2007. Lecture Notes in Computer Science, vol 4858. Springer, Berlin, Heidelberg.
- [6] Oskari Tammelin (2014). “Solving Large Imperfect Information Games Using CFR+”. CoRR abs/1407.5042.
- [7] Dohmatob E. (2015). A simple and efficient algorithm for computing approximate Nash equilibria in two-person zero-sum sequential games with incomplete information. CoRR, abs/1507.07901.
- [8] Elvis Dohmatob (2016). A simple algorithm for computing Nash-equilibria in incomplete information games. In OPT2016 - NIPS workshop on optimization for machine learning, Dec 2016, Barcelona, Spain. 2016.

A NEW FICTITIOUS PLAY ALGORITHM APPROACH FOR RESOURCES ALLOCATION

Trinh Bao Ngoc, Huynh Quyet Thang, Le Cong Thanh, Le Ba Truong Giang, Tran Quang Huy

ABSTRACT: Resources allocation in a project is the process of balancing available project resources among teams in project organization, this process must perform regularly, precisely to keep project on the right track, conflicts in resources allocation may cause lots of issues to all project activities. That is the reason why, the problem in resources allocation need to take into account appropriately. The purpose of this research aims to propose the suitable mathematical model to the problem of resources allocation which is to be consider as a game in Game Theory, and then, finding Nash Equilibria for this game as a solution for the problem. In this model, we will find sampling of probability distribution in selected strategies of players in the game of allotting resources. Project managers and their teams can take advantage of this strategies to make a right decision. We introduce a new approach of Fictitious Play algorithm in resources allocation problem and analyze pros and cons of this method when dealing with a large amount of data of game.

Keywords: Fictitious Play Algorithm, Nash Equilibria, Resources allocation, CFR Algorithm, CFR Plus Algorithm, Game theory.