Giải quyết bài toán Weapon-Target Assignment sử dụng mô hình mạng hồi quy tuyến tính

Vấn đề về chỉ định mục tiêu - vũ khí là tìm một sự phân công hợp lý để nhắm đến mục tiêu sao cho giảm thiểu lớn nhất sự thiệt hại của kẻ thù đem lại cho lực lượng. Có thể xét trên phương diện khác, cũng chính là việc làm tăng tối đa hiệu quả tấn công của lực lượng lên kẻ thù.

Có nhiều phương án để giải quyết bài toán WTA (Weapon – Target assignment). Một số phương pháp cổ điển đã được áp dụng như combinatorial optimization, pseudo-boolean programming đã được áp dụng tuy nhiên dẫn đến sự phức tạp lớn trong tính toán, hậu quả là khó có thể tính toán đối với số lượng vũ khí lớn. Một hướng khác là sử dụng các giải thuật di truyền GA-Genetic Algorithm, có khả năng giải quyết được bài toán, tuy nhiên kết quả đạt được không đủ cho sự kỳ vọng.

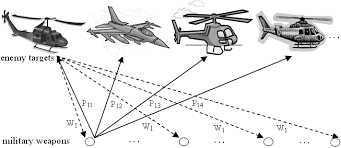
Báo cáo sau đây sẽ đề cập một phương pháp mới để giải quyết bài toán chỉ định vũ khí, trên cơ sở những nội dung sau

1. Đặt vấn đề
2. Mô hình toán của WTA
3. Thuật toán để giải quyết bài toán
4. Xây dựng phần mềm ứng dụng
5. Kết luận và các hướng phát triển

Sau đây sẽ là nội dung chi tiết

**1. Đặt vấn đề**

Có hai phương án cần được xét đến đó là WTA tĩnh và WTA động.



Trong phương án tĩnh, ta xét đến việc tối ưu hóa hiệu quả tấn công trong kịch bản đã biết trước số lượng mục tiêu - vũ khí cũng như xác suất tiêu diệt trong một lượt bắn là cố định và không xét đến sự biến đổi theo thời gian, đầu ra của bài toán SWTA sẽ là giá trị chỉ định hoặc không chỉ định của một vũ khí so với một mục tiêu. Khác với phương án tĩnh, phương án động có mô hình phức tạp hơn khá nhiều. DWTA có hai loại là shoot-look-shoot DWTA và 2-stage DWTA đều xem xét mô hình với sự thay đổi theo thời gian. 2-stage DWTA bao gồm giai đoạn đầu tính toán tương tự như SWTA, tuy nhiên có tính đến sự dự trữ vũ khí để cho giai đoạn sau. Shoot-look-shoot DWTA cũng sử dụng một phần của SWTA tuy nhiên lại tính đến khả năng bỏ sót mục tiêu từ đó xây dựng cách phân bổ vũ khí dữ cho đợt tấn công tiếp theo. Báo cáo này sẽ xem xét giải bài toán với mô hình tĩnh – SWTA.

Một nghiên cứu mới đây đã chỉ ra rằng mô hình mạng RNN là công cụ tốt để giải quyết bài toán đưa ra. Ý tưởng chính của nó là xây dựng một hàm không âm được gọi là hàm năng lượng. Hàm này sẽ phải đơn điệu giảm đến khi hệ đạt trạng thái cân bằng (hội tụ). Mặc dù mô hình RNN có một số giới hạn trong việc giải các chương trình zero-one với các hàm mục tiêu là hàm lồi. WTA là một vấn đề tối ưu hóa với ràng buộc là zero-one, tuy nhiên báo cáo này sẽ chỉ ra một mô hình RNN có thể sử dụng cho WTA.

Báo cáo sau đây dựa trên kết quả của bài báo **“*Projection Recurrent Neural Network Model: A New Strategy to Solve Weapon-Target Assignment Problem*”** của các tác giả “*Alireza Shojaeifard · Ali Nakhaei Amroudi · Amin Mansoori · Majid Erfanian*”.

Đăng trên tạp chí Springer Science - Business Media, LLC, part of Springer Nature 2019. *https://www.springerprofessional.de/en/projection-recurrent-neural-network-model-a-new-strategy-to-solv/16844850*

**2. Mô hình toán của WTA**

Trên các chiến trường hiện đại, nhiệm vụ quan trọng đối với chỉ huy là tìm ra một phương án thích hợp để bảo vệ tài sản của lực lượng cũng như tối đa hóa hiệu quả tấn công của đơn vị. Xét một ví dụ đối với hải quân, mục tiêu đe dọa có thể đến từ các tàu trên mặt nước, máy bay và có thể là tàu ngầm. Các vũ khí của đơn vị sẽ có xác suất tiêu diệt theo từng mục tiêu kể trên là khác nhau. Như vậy việc chỉ định mục tiêu – vũ khí sẽ ảnh hưởng trực tiếp đến sự thành bại của cuộc chiến. Sau đây ta sẽ xem xét trên phương diện toán học.

Ta sẽ sử dụng các ký hiệu như sau

– Số lượng vũ khí

 – Số lượng mục tiêu cần phải tiêu diệt

 – Giá trị nguyên biểu thị mức độ ưu tiên tấn công vào mục tiêu i, ui có giá trị càng lớn tương ứng với mức độ ưu tiên càng cao.

 – Xác suất tiêu diệt mục tiêu i trong một lần tấn công bởi vũ khí j

 – Một biến nhị phân cho biết vũ khí loại j có đang được ngắm đến mục tiêu i hay không. Với , ta có mục tiêu thứ i được chỉ định bởi vũ khí thứ j, tương tự mục tiêu thứ i không được chỉ định bởi vũ khí j thì 

Xét hàm số sau



Hàm F ứng với tổng xác suất bắn trượt, như vậy nếu F là cực tiểu, ta sẽ có hiệu suất tấn công là cao nhất. Như vậy, để giải bài toán chỉ định mục tiêu – vũ khí là tìm cực tiểu của hàm chi phí trên.

Trong trận chiến, ta cần tất cả các mục tiêu đều được chỉ định bởi vũ khí. Không có vũ khí nào mà không ngắm tới mục tiêu. Và một vũ khí chỉ được chỉ định vào một mục tiêu. Như vậy ta có điều kiện sau



Xét trường hợp số mục tiêu bằng số loại vũ khí, ta sẽ có mỗi loại vũ khí sẽ được gán với chỉ một mục tiêu, đặt W=T=N ta có



Tổng kết lại, ta có mô hình của WTA



Nhận thấy đây là bài toán tối ưu với mô hình là phi tuyến và ràng buộc zero-one. Tức là giá trị của x chỉ có thể có 2 khả năng là 0 hoặc 1.

**3. Thuật toán giải quyết bài toán**

1. **Cơ sở lý thuyết**

Với mô hình đã nêu trên, ta có thể chuyển từ ràng buộc zero – one sang ràng buộc tuyến tính



Sau đây ta sẽ chứng minh hàm F là hàm lồi

Xét hàm số 

Giả sử với  và , ta có



Từ đó ta có



Suy ra 

Hơn nữa, ta cũng có  là hàm lồi

và  với mọi 

Do  nên ta có



Như vậy ta chứng mình được f là hàm lồi, do F và f có mối quan hệ tuyến tính, do vậy F cũng là hàm lồi

* **Mô hình RNN**

Ta xem xét phương trình phi tuyến lồi với ràng buộc tuyến tính như sau



Bài toán tối ưu sẽ là



Ta cần xem xét một số bổ đề sau

***Bổ đề 1***

Với là một tập con lồi đóng của Rn, ta định nghĩa



Sẽ có các tính chất như sau

1. 

2. 

3.  là nghiệm tối ưu nếu và chỉ nếu với mọi , ta đều có



4. 

**Sau đây sẽ là mô hình của RNN được xét đến**



Trong đó  và  được định nghĩa như trên

***Bổ đề 2***

là nghiệm tối ưu nếu và chỉ nếu nó là điểm cân bằng của bài toán RNN được nói đến ở trên

*Chứng minh*

Ta giả sử  là nghiệm tối ưu của phương trình, do đó 

Từ đó suy ra 

Chuyển vế ta có 

Từ tính chất 2 của bổ đề 3.1 ta có điều kiện cần và đủ để  là nghiệm tối ưu đó là 

Từ tính chất 3 của bổ đề 3.1 ta lại có 

Mặt khác, với  ta có 



Thỏa mãn phương trình (\*), do vậy giả thiết đã đúng.

* **Phân tích tính ổn định**

***Bổ đề 3***

Vế phải của phương trình (\*) thỏa mãn điều kiện Lipschizian

*Chứng minh*

Ta có



Theo định lý giá trị trung bình ta có



Giả thiết  ta có



Vậy ta chứng minh được vế phải của (\*) là Lipschitz

***Bổ đề 4***

Tồn tại nghiệm x(t) thỏa mãn phương trình (\*)

***Bổ đề 5***

Phép ánh xạ 

với là ánh xạ co

*Chứng minh*

Dựa vào tính chất thứ 2 của bổ đề 3.1, ta có phương trình sau

(\*\*)

Với  là ma trận đối xứng xác định dương



Trong đó là trị riêng của ma trận trên, với  ta có





Sử dụng công thức (\*\*) ta được



Chứng minh được ánh xạ là ánh xạ co

***Bổ đề 6***

Mô hình được nêu ở phương trình (\*) là ổn định theo cấp số nhân

*Chứng minh*

Dựa vào Bổ đề 2 và 4, xem xét tích phân sau





Trừ hai vế phương trình ta được



Hay có thể viết dưới dạng





Sử dụng bất phương trình thứ hai của bổ đề 3 ta được





Với  ta có điều phải chứng minh

Như vậy từ những bổ đề nêu trên ta có thể chứng minh được rằng mô hình WTA có nghiệm tối ưu cũng chính là nghiệm tối ưu của phương trình (\*), và nghiệm tối ưu của phương trình (\*) sẽ được tính toán bằng cách sử dụng mô hình trên.

1. **Mô phỏng thuật toán**

Sau đây ta sẽ tiến hành mô phỏng thuật toán của bài báo đang xem xét ở trên

Xét hàm số 

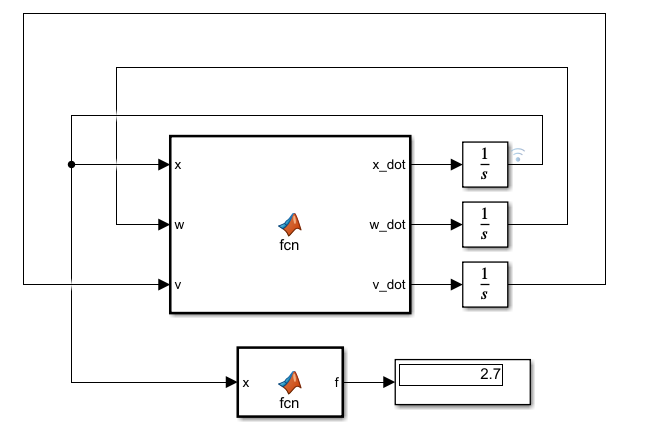
Gọi x\*, w\* và v\* là các giá trị tại điểm tối ưu, sẽ thỏa mãn hệ phương trình sau



Ta có hệ phương trình vi phân mô tả hệ



Ta thấy sau mỗi một chu kỳ các giá trị x, w và v sẽ được update 1 lần. Từ đó, tốc độ thu được kết quả sẽ phụ thuộc vào thời gian 1 lần lấy mẫu và tốc độ tính toán của thiết bị xử lý. Sử dụng công cụ Matlab/Simulink để mô phỏng, ta sẽ xây dựng mô hình có cấu trúc đơn giản như sau.



Để có thể kiểm chứng mô hình, trước tiên ta xét ví dụ đơn giản với N=3

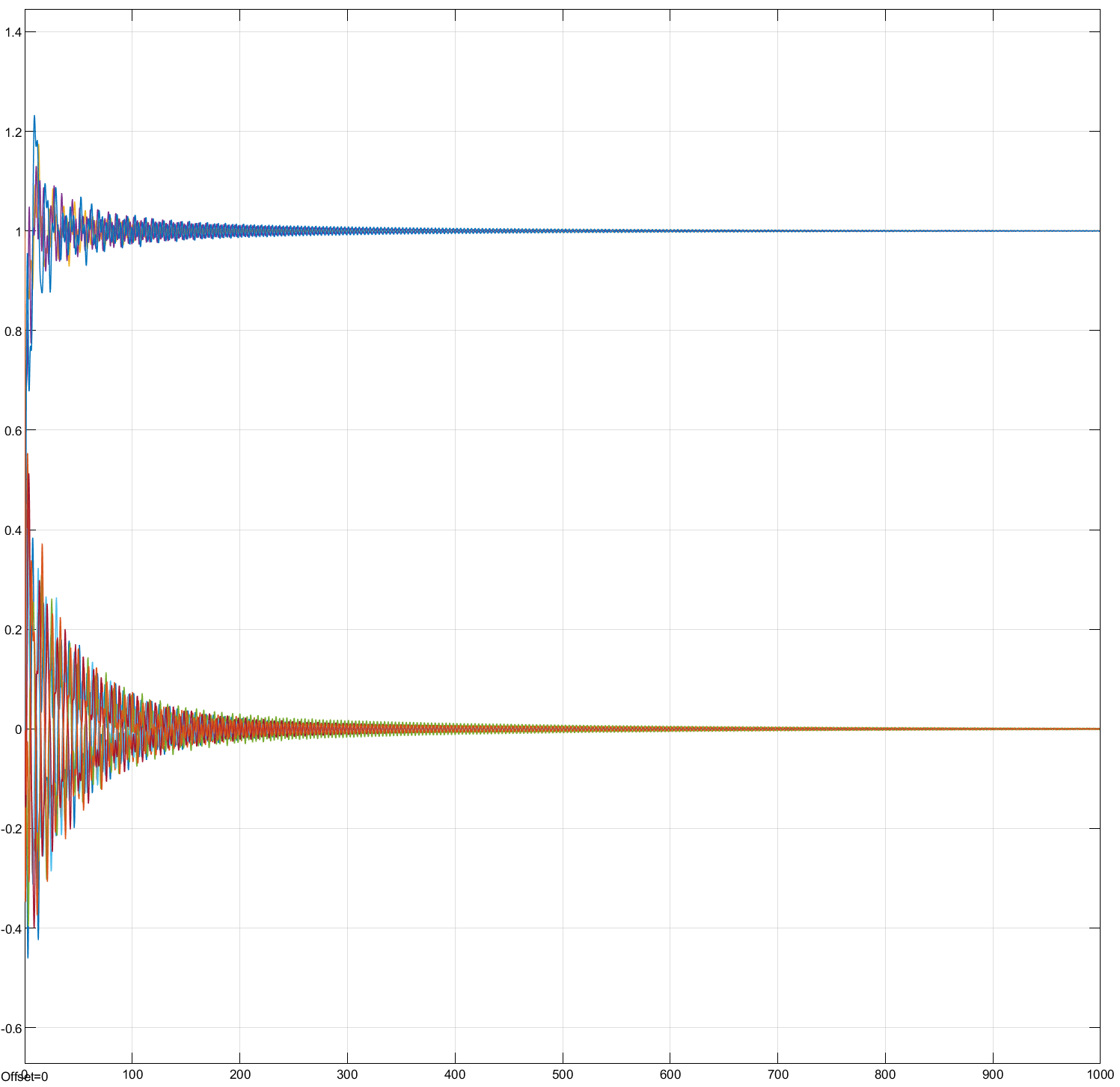
Với u = [1;3;2]. Mục tiêu 2 có độ ưu tiên cao nhất, sau đó đến mục tiêu 3 và cuối cùng là mục tiêu 1

Và p = [0.6 0.5 0.8; 0.9 0.5 0.7; 0.8 0.9 0.4];

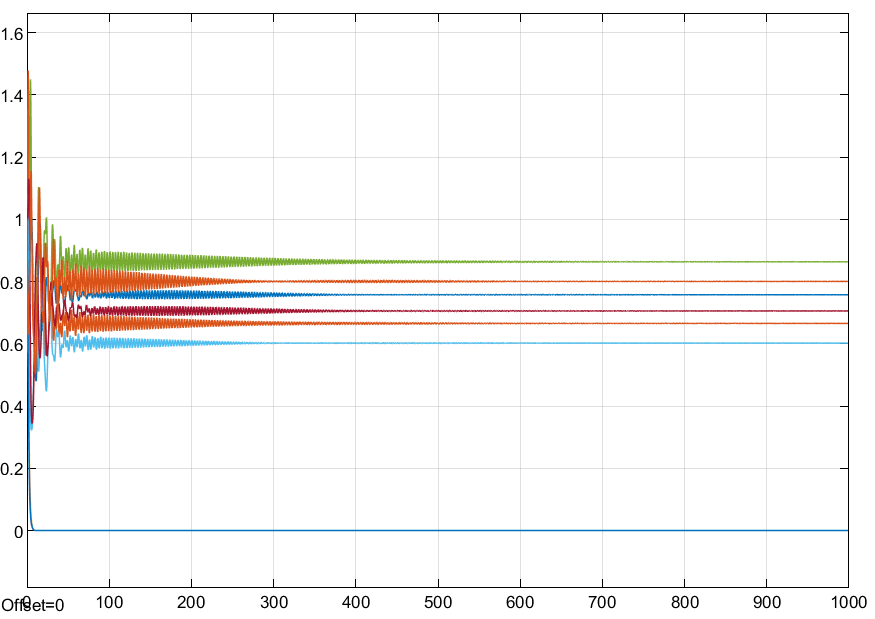
Điểm đầu ta chọn

x­0 = [1;0;0;0;1;0;0;0;1]; w0 = [1;1;1;1;1;1;1;1;1]; v0 = [1;1;1;1;1;1]

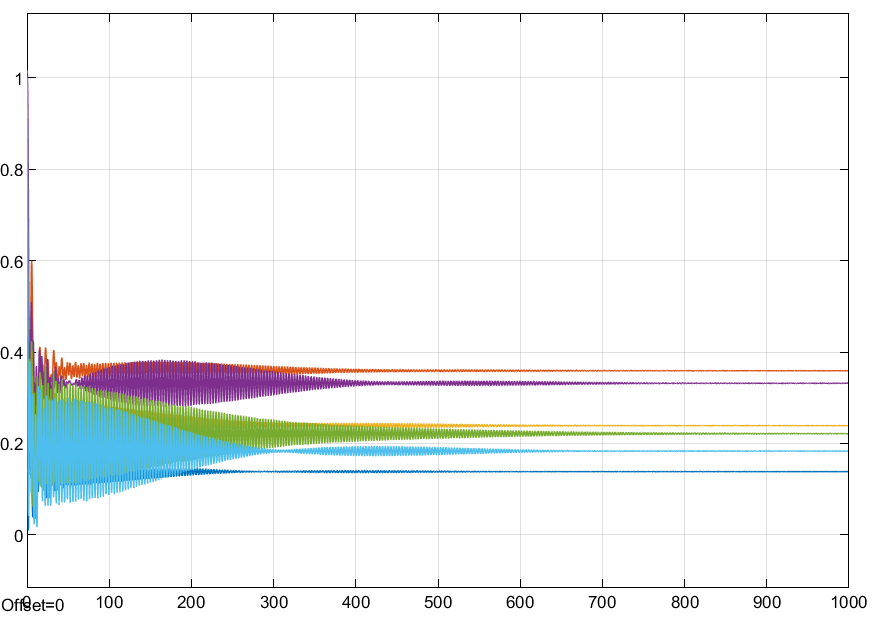
Ta thu được kết quả như sau



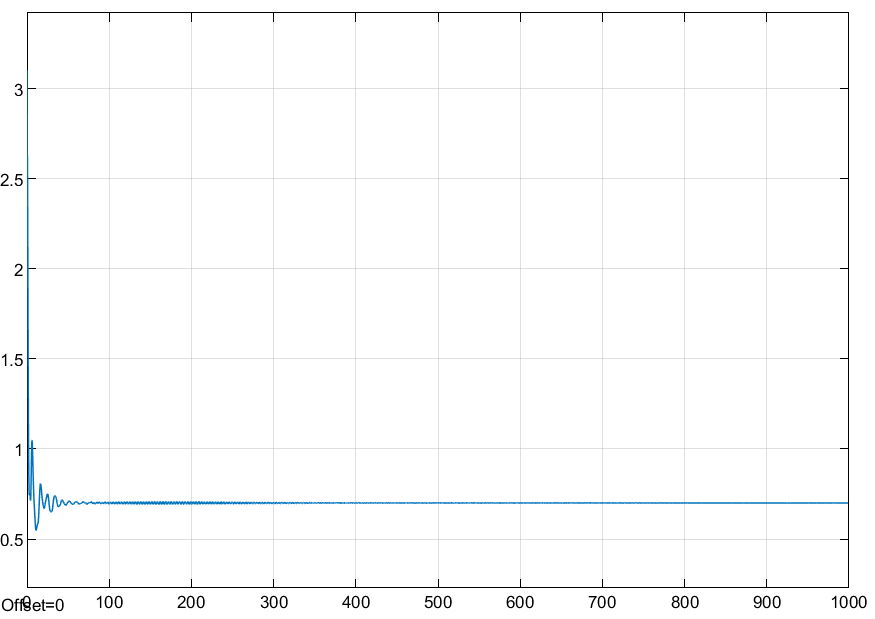
Đồ thị với x



Đồ thị với w



Đồ thị với v



Đồ thị với F

Ta nhận thấy, tại điểm hội tụ, các giá trị của x, w và v thỏa mãn hệ phương trình đã cho. Ta chọn chu kỳ trích mẫu để tính tích phân là Ta = 0.001 s. Thực hiện 1 000 000 vòng lặp do vậy trục thời gian mô phỏng tương ứng T =1000 s

Sau khi chuẩn hóa vector x về dạng ma trận, ta thu được kết quả

x = [0 0 1; 1 0 0; 0 1 0] và F = 0.7

Để chắc chắn rằng kết quả thu được là nhỏ nhất, ta sẽ thử từng trường hợp để có bảng giá trị ứng với từng bộ số x. Ta thu được bảng sau

|  |  |
| --- | --- |
| x | F |
| [1 0 0; 0 1 0; 0 0 1] | 3.1 |
| [1 0 0; 0 0 1; 0 1 0] | 1.5 |
| [0 1 0; 1 0 0; 0 0 1] | 2 |
| [0 1 0; 0 0 1; 1 0 0] | 1.8 |
| [0 0 1; 1 0 0; 0 1 0] | 0.7 |
| [0 0 1; 0 1 0; 1 0 0] | 2.1 |

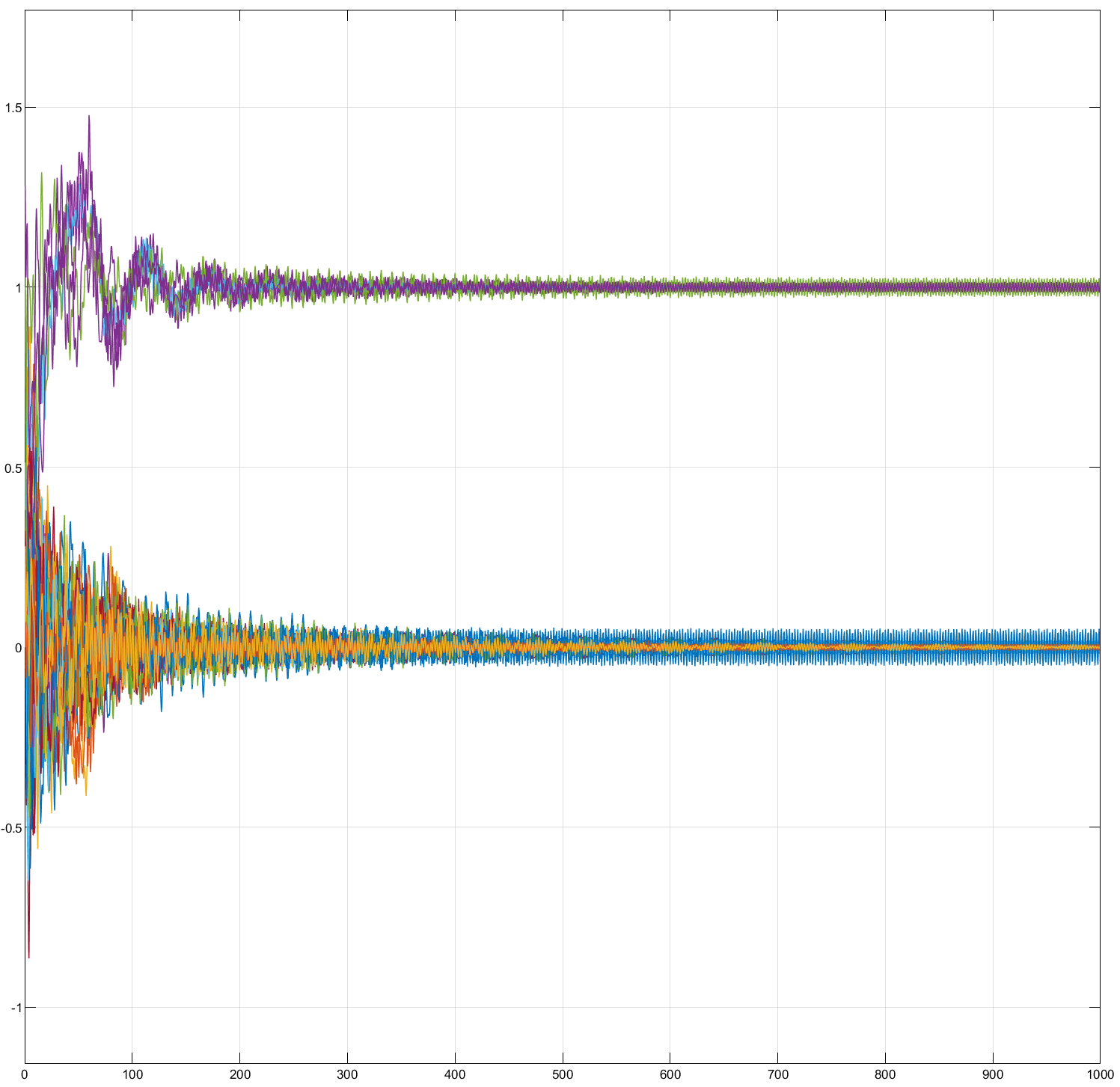
Như vậy ta thấy kết quả thu được là chính xác.

Xem xét bài toán WTA với N=5, có các tham số được cho như sau

(bộ tham số này cũng khớp với bài báo)



Với u và p đã cho ở trên ta thu được kết quả như sau



Đồ thị với x

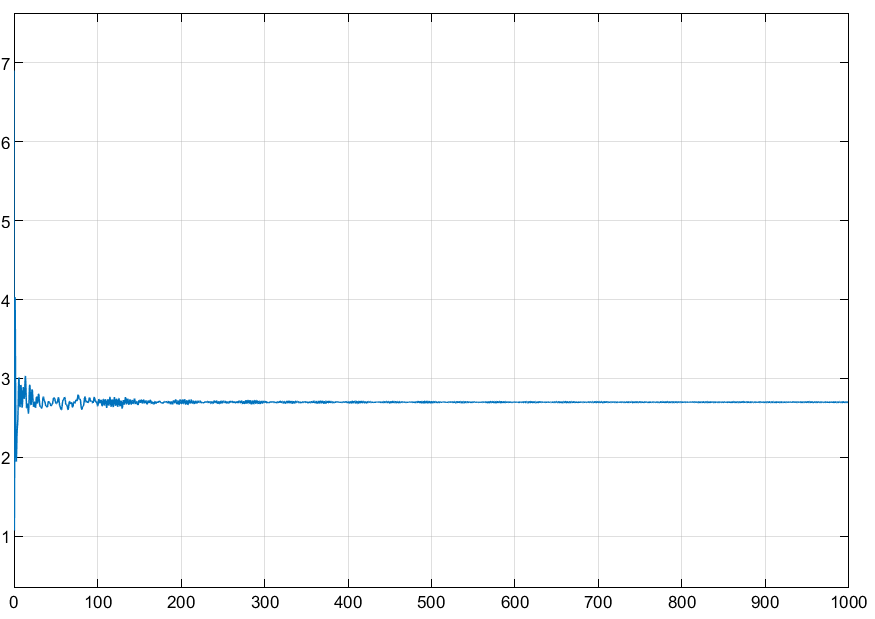
Như vậy ta thu được x =  và F = 2.7

Điều này có nghĩa ta sẽ ghép cặp

vũ khí 1 - mục tiêu 2; vũ khí 2 - mục tiêu 3; vũ khí 3 - mục tiêu 4;

vũ khí 4 - mục tiêu 1; vũ khí 5 - mục tiêu 5

Kết quả mô phỏng trùng khớp với kết quả của bài báo



Đồ thị với F

Như vậy, ta đã kết luận được khi sử dụng thuật toán trên, có thể giải quyết được bài toán chỉ định mục tiêu – vú khí. Tuy nhiên, còn tồn tại những hạn chế mà bài báo chưa giải quyết được, ta sẽ xem xét nguyên nhân cũng như tìm ra cách giải quyết trong phần sau.

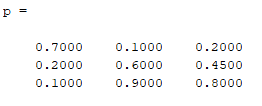
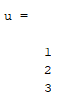
1. **Các vấn đề còn tồn tại và phương án xử lý**

* Trong quá trình mô phỏng, sẽ có nhiều lúc gặp phải trường hợp các biến trạng thái x không hội tụ về hai giá trị 0 hoặc 1 có thể do việc lựa chọn p và u ngẫu nhiên sẽ làm xuất hiện các cặp nghiệm cho ra kết quả giống nhau
* Chưa giải quyết được với số lượng mục tiêu và vũ khí là khác nhau
* Cần phải tối ưu về mặt tốc độ

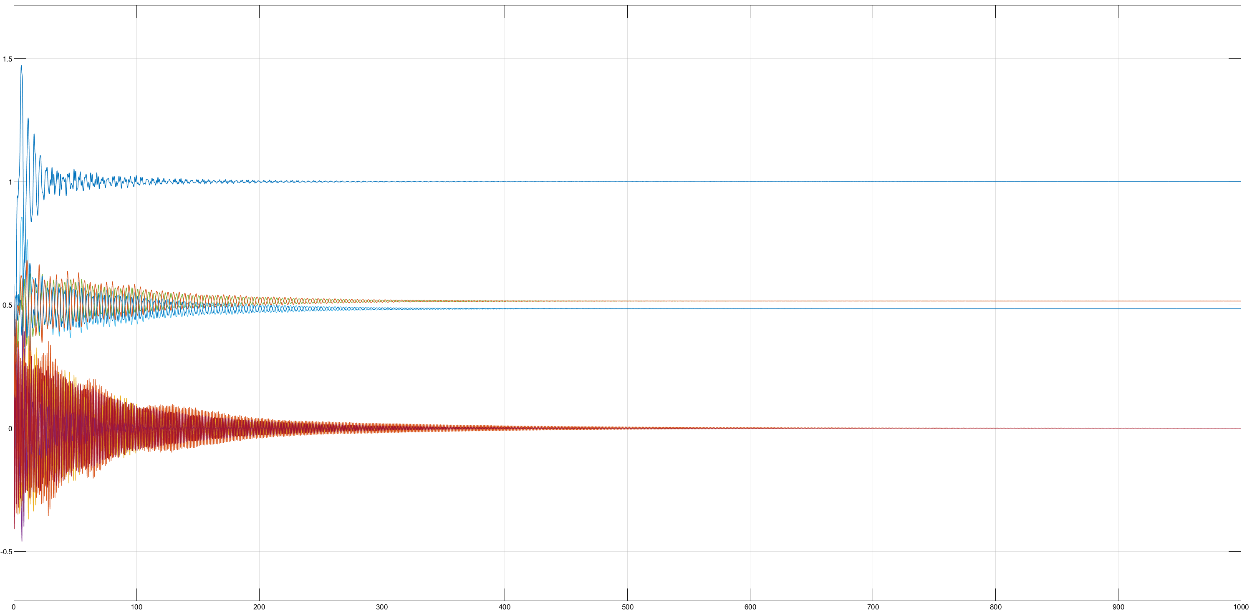
1. **Vấn đề 1**

Sau khi thử nghiệm, khẳng định được nguyên nhân làm cho các biến trạng thái không hội tụ về 0 là do khi chọn ngẫu nhiên u và p sẽ xuất hiện các khả năng tạo ra các tình huống khác nhau nhưng lại có kết quả giống nhau (giá trị hàm F là giống nhau).

Để mô tả hiện tượng này, ta lấy một ví dụ đơn giản với số mục tiêu và số vũ khí bằng 3. Giá trị u và p được chọn như sau

Sau khi chạy thuật toán ra được kết quả như hình



Ta thấy x không hội tụ về 1 hay 0 mà có giá trị hội tụ về 0.5. Tức là sẽ có 2 trường hợp cho ta giá trị hàm F là tối ưu

Kiểm chứng với bảng sau

|  |  |
| --- | --- |
| x | F |
| [1 0 0; 0 1 0; 0 0 1] | 1.7 |
| [1 0 0; 0 0 1; 0 1 0] | 1.7 |
| [0 1 0; 1 0 0; 0 0 1] | 3.1 |
| [0 1 0; 0 0 1; 1 0 0] | 4.7 |
| [0 0 1; 1 0 0; 0 1 0] | 2.7 |
| [0 0 1; 0 1 0; 1 0 0] | 4.3 |

Ta thấy giả thiết đã đưa ra phù hợp với thực tế.

Ta sẽ xử lý vấn đề trên như sau

* Sau khi đã có được giá trị hội tụ của mảng x, trước tiên với các giá trị hội tụ các xác xuất lớn, gần với giá trị 1 ta sẽ cho giá trị của nó bằng 1 và giá trị của các x khác với cùng giá trị hàng hoặc cùng giá trị cột bằng 0 để giảm khối lượng cần phải xử lý.
* Còn lại sẽ là các giá trị hội tụ trong khoảng mà ta cần phải xử lý, tiến hành như sau
* Phần tử x tại hàng i cột j sẽ được so sánh với toàn bộ x cùng hang và x cùng cột. Nếu thấy giá trị của x[i][j] là lớn nhất trong các giá trị đó, ta gán cho x[i][j] giá trị bằng 1
* Quét lần cuối trong mảng đã xử lý, nếu giá trị của x[i][j] bất kỳ khác 1 thì cho nó về 0.

1. **Vấn đề 2**

Thuật toán đang sử dụng là tối ưu hàm chi phí với ràng buộc tuyến tính. Toàn bộ mô hình này được xây dựng trên cơ sở số mục tiêu và vũ khí là bằng nhau và có giá trị cho trước. Như vậy, để có thể giải bài toán với số lượng mục tiêu và vũ khí là khác nhau ta sẽ có hai phương pháp. Phương pháp thứ nhất, ta xét lại mô hình tổng quát và xây dựng thuật toán cho mô hình mới. Phương pháp thứ hai là chuẩn hóa kịch bản để có thể sử dụng được thuật toán cũ. Sau đây ta sẽ sử dụng phương án thứ hai.

* **Số lượng mục tiêu nhiều hơn số lượng vũ khí**

- Để đảm bảo điều kiện thực hiện thuật toán, ta cần phải có đủ phần tử của ma trận p, do vậy đối với số lượng mục tiêu nhiều hơn vũ khí, ta sẽ tạo ra một số lượng vũ khí giả, với xác suất bắn trúng đối với tất cả các mục tiêu là bằng 0.

- Do hàm mục tiêu có dạng , với pij = 0 thì giá trị của x­ij là bao nhiêu cũng không ảnh hưởng đến kết quả thu được. Như vậy ta có thể kết luận phương pháp trên là sử dụng được.

* **Số lượng mục tiêu ít hơn số lượng vũ khí**

- Khi số lượng mục tiêu ít hơn số lượng vũ khí, để tăng khả năng tiêu diệt, sẽ có trường hợp nhiều hơn 1 vũ khí tấn công vào 1 mục tiêu. Ta sẽ thực hiện thuật toán như sau

- Tương tự như trường hợp trên, ta tạo ra các mục tiêu giả, với xác suất bắn trúng mục tiêu đó của toàn bộ vũ khí là bằng 0 và mức độ ưu tiên của nó cũng bằng 0. Như vậy hàm mục tiêu sẽ không bị ảnh hưởng bởi số mục tiêu giả đó.

- Chạy thuật toán lần thứ nhất, ta sẽ thu được kết quả trong đó có các vũ khí nhắm vào các mục tiêu giả. Ta sẽ đánh dấu lại những vũ khí đó.

- Tiếp theo ta sẽ chạy thuật toán lần thứ hai, với số lượng mục tiêu vẫn như cũ tuy nhiên số vũ khí chỉ là những vũ khí mà ta đã đánh dấu ở lần 1. Đưa bài toán về trường hợp thứ nhất – số lượng mục tiêu nhiều hơn số lượng vũ khí. Sau đó ta thực hiện chạy thuật toán với trường hợp kể trên.

1. **Vấn đề 3**

Hiện tại thuật toán đang để số vòng lặp là rất lớn để đảm bảo điều kiện hội tụ. Tuy nhiên đối với những kịch bản với số lượng mục tiêu và vũ khí là nhỏ, không nhất thiết phải trải qua quá nhiều vòng lặp để có thể hội tụ. Như vậy, nếu có thể giảm thiểu được số vòng lặp ta sẽ tối ưu được thời gian tính toán cho trường hợp số lượng vũ khí và mục tiêu là nhỏ.

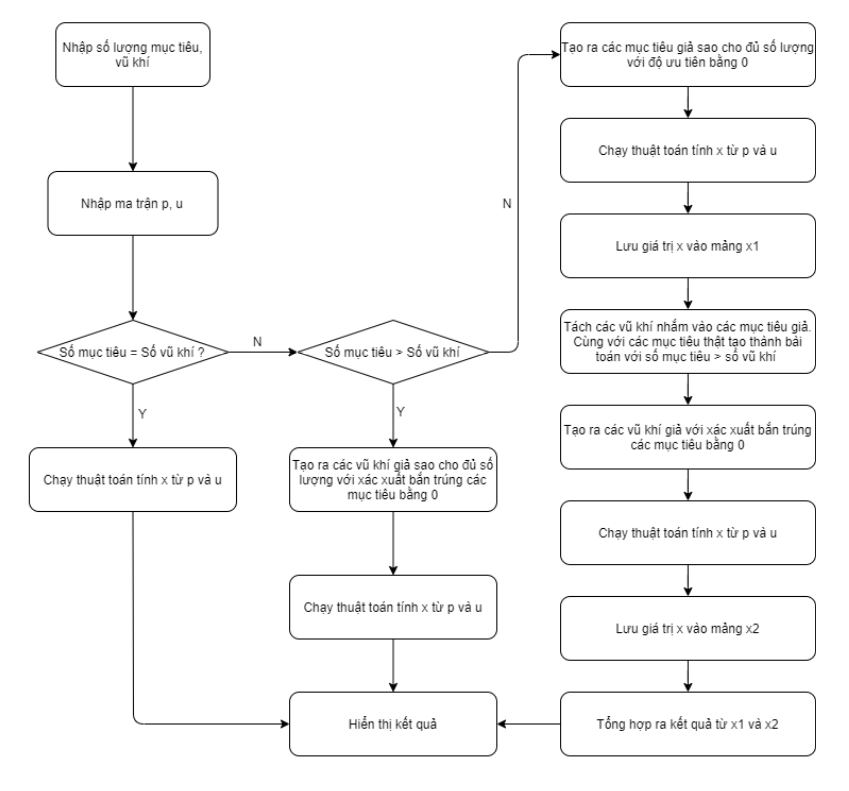
Để giảm thiểu số vòng lặp cần phải trải qua, ta đưa thêm một điều kiện dừng của thuật toán đó là giá trị đạo hàm của tất cả các giá trị trong ma trận v nhỏ hơn một giá trị  đủ nhỏ. Trong quá trình mô phỏng, nhận thấy việc hội tụ của biến trạng thái F là nhanh hơn so với thời gian hội tụ của biến v. Trong đó giá trị đạo hàm của biến v mô tả sự thỏa mãn điều kiện tuyến tính của mô hình tại thời điểm xét. Như vậy, việc ta chọn điều kiện dừng của thuật toán như trên sẽ thỏa mãn được giá trị của F tiến tới lân cận tối ưu cũng như điều kiện ràng buộc tuyến tính được thoả mãn.

1. **Xây dựng phần mềm ứng dụng**

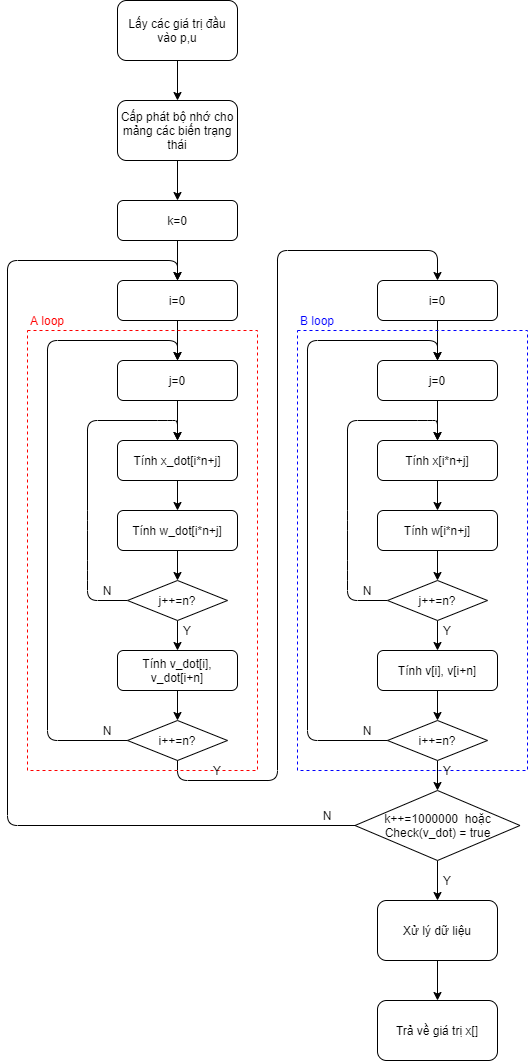
Sau khi đã có ý tưởng về thuật toán cũng như các hiệu chỉnh về mặt tối ưu, ta tiến hành xây dựng ứng dụng để giải quyết bài toán WTA

* 1. **Sơ đồ thuật toán**

Sau đây là sơ đồ tổng quát của chương trình



Sơ đồ chi tiết thuật toán đưa ra giá trị của x từ u và p



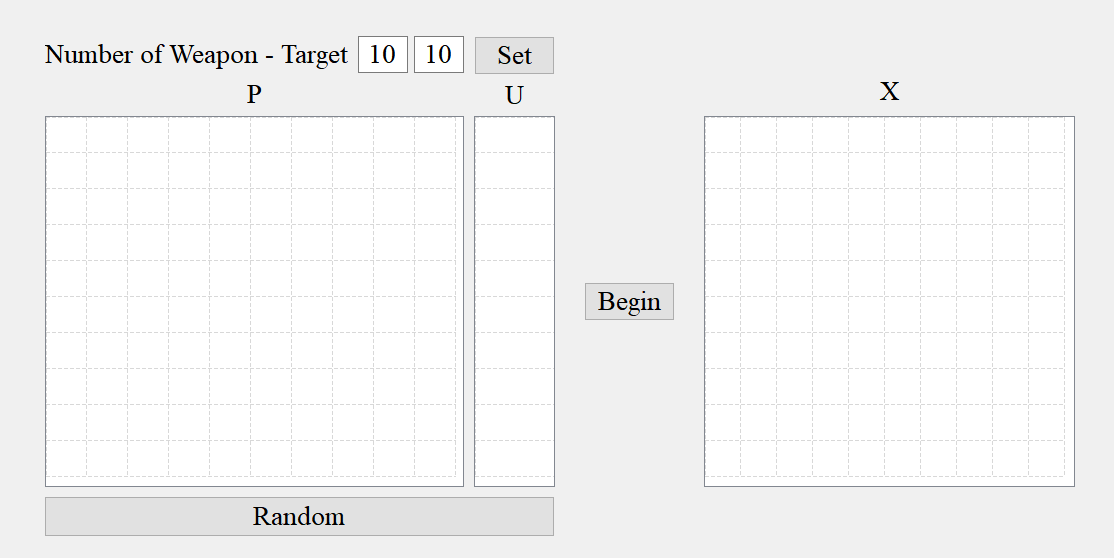
Mô tả chung:

* Người sử dụng nhập số lượng vũ khí – mục tiêu. Ma trận p và u có thể tùy chọn nhập vào hoặc sử dụng giá trị ngẫu nhiên.
* Chương trình bao gồm 1 vòng lặp chính, có nhiệm vụ tính giá trị của các biến trạng thái của vòng lặp thứ k+1 từ giá trị của các biến trạng thái của vòng lặp thứ k. Vòng lặp này buộc phải tính tuần tự.
* Trong vòng lặp này ta lại thực hiện 2 vòng lặp nhỏ, kí hiệu là A và B
* Vòng lặp A ta sẽ thực hiện tính giá trị đạo hàm của các biến trạng thái từ giá trị đầu vào của nó tại chu kỳ thứ k và giá trị của hai ma trận p và u. Với số phép tính ứng với biến trạng thái x\_dot là n\*n, ứng với biến trạng thái w\_dot là n\*n và với biến trạng thái v\_dot là n\*2.
* Vòng lặp B ta sẽ thực hiện tính giá trị của các biến trạng thái tại vòng lặp thứ k+1 bằng giá trị và giá trị đạo hàm của nó tại thời điểm k theo cách tính sai phân. Với số phép tính ứng với biến trạng thái x là n\*n, ứng với biến trạng thái w là n\*n và với biến trạng thái v là n\*2.
* Ở cuối mỗi vòng lặp lớn, ta có thêm điều kiện dừng của thuật toán.
* Sau khi đã chạy qua vòng lặp chính, ta sẽ tiến hành xử lý dữ liệu nhận được. Dữ liệu lúc này được coi là đã hội tụ, ta sẽ xử lý các trường hợp biến trạng thái x không hội tụ về 0 hoặc 1.
* Sau khi đã hoàn tất xử lý, ta đưa kết quả ra giao diện bằng cách hiển thị giá trị 1 và 0 ứng với chỉ định và không chỉ định mục tiêu. Đồng thời tại các vị trí chỉ định mục tiêu ta sẽ tô nền màu đỏ.
* Sau khi thực hiện hết tất cả các bước trên, ta in ra màn hình thời gian thực hiện toàn bộ thuật toán.
* Các giá trị hằng số:
* Chu kỳ trích mẫu Ta = 0.001
* Số vòng lặp tối đa k =1 000 000
* Giá trị điều kiện dừng
  1. **Xây dựng giao diện người dùng**

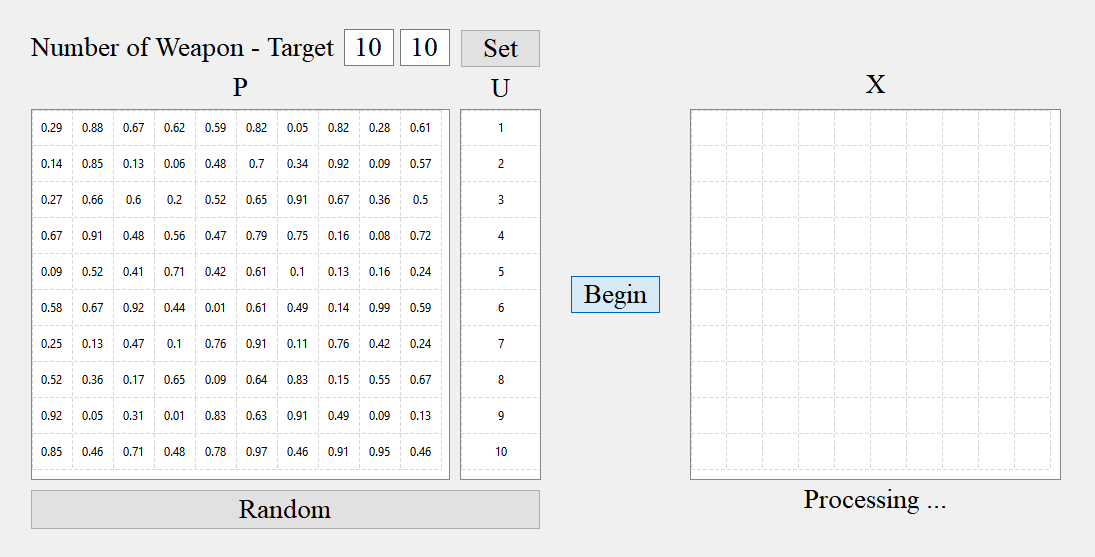
Ta sẽ sử dụng công cụ QtCreator để xây dựng giao diện, với yêu cầu như sau

* Có thể nhập số lượng mục tiêu, số lượng vũ khí
* Có thể nhập bảng giá trị của p và u
* Nếu không có giá trị p và u nhập vào, có thể tự xuất dữ liệu random
* Hiển thị kết quả và thời gian chạy thuật toán

Giao diện sau khi xây dựng



Giao diện lúc đang trong quá trình chạy thuật toán



Giao diện khi đã chạy xong thuật toán



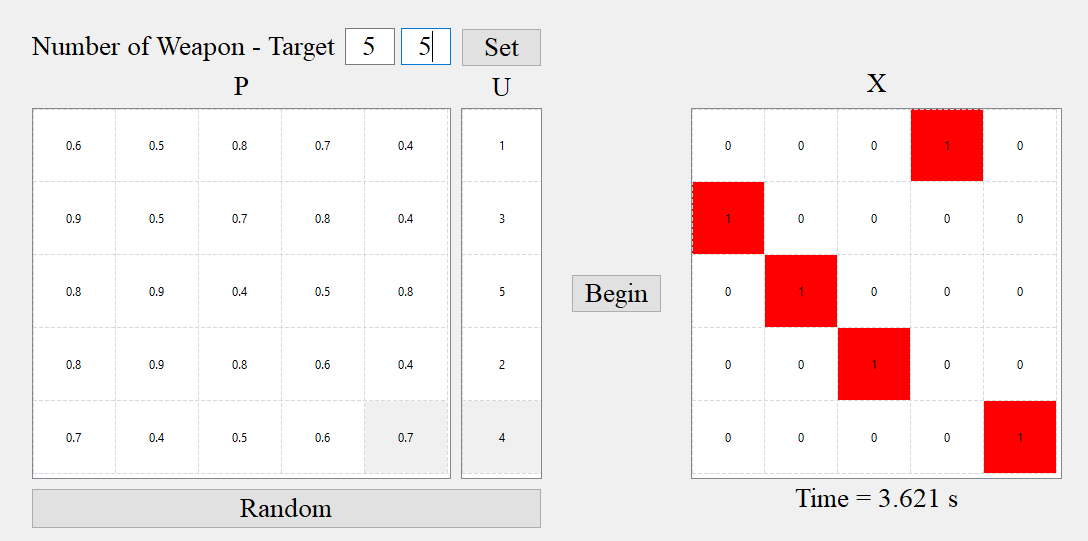
* 1. **Thử nghiệm và kiểm tra**

Thử nghiệm với tình huống của bài báo với N=5

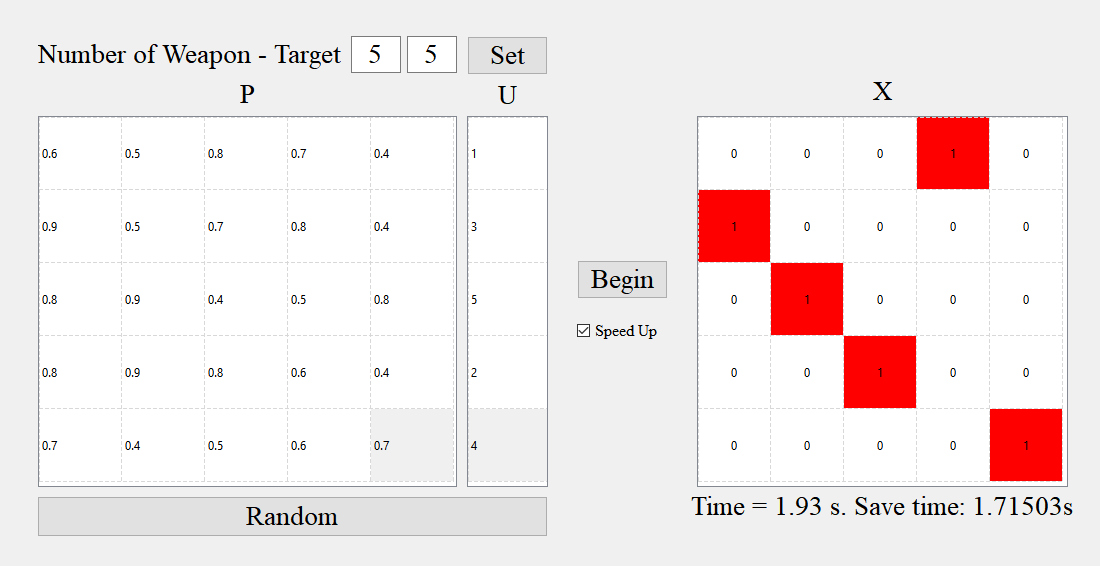
Các giá trị u và p như sau



Ta thu được kết quả



Sử dụng chế độ Speed Up bằng cách thêm điều kiện dừng như đã trình bày ở mục 3.3.3, chọn giá trị  cải thiện được khá nhiều thời gian đối với số lượng mục tiêu – vũ khí là nhỏ.



Để có thể kiểm chứng độ chính xác, ta sẽ chạy thuật toán với tình huống biết trước kết quả như sau

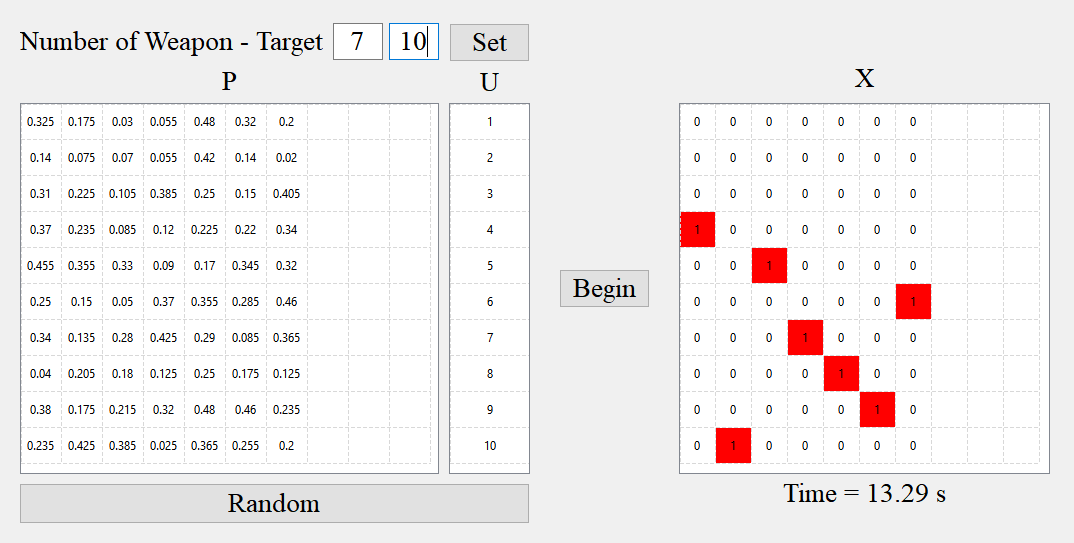
* Số lượng N=T=W=10
* p sẽ là ma trận sao cho đường chéo có xác suất là 0.9, còn lại các giá trị khác sẽ là ngẫu nhiên và nhỏ hơn 0.9
* u là mảng với các giá trị tăng dần từ 1 đến 10



Nhận thấy ta thu được kết quả x là ma trận đơn vị. Đúng với giả thiết đặt ra. Tiếp theo ta sẽ chạy thuật toán nhiều lần với số lượng mục tiêu – vũ khí là khác nhau để có thời gian chạy thuật toán trong mỗi trường hợp, ta thu được bảng sau

|  |  |
| --- | --- |
| N | Time |
| 5 | 3.45 s |
| 10 | 14.50 s |
| 20 | 64.15 s |
| 50 | 306.47 s |
| 80 | 822.06 s |
| 100 | 1420.59 s |
| 200 | > 40 phút |

Đối với trường hợp số lượng mục tiêu và vũ khí khác nhau



Đối với trường hợp số lượng mục tiêu nhiều hơn số lượng vũ khí, ta phải chấp nhận có một số mục tiêu không được chỉ định. Mục tiêu không được chỉ định là mục tiêu mà giá trị của X tại hàng ứng với mục tiêu đó là bằng 0



Đối với trường hợp số lượng vũ khí lớn hơn số lượng mục tiêu, sẽ có trường hợp mục tiêu được chỉ định bởi 2 vũ khí, ứng với hàng có hai giá trị của ma trận X bằng 1 và được tô 2 màu đỏ và vàng tương ứng. Tuy nhiên do việc phải chạy thuật toán 2 lần dẫn đến việc thời gian tăng lên đáng kể.

1. **Kết luận và hướng phát triển**

* Sau khi đã hoàn tất xây dựng ứng dụng để giải quyết bài toán WTA theo hướng sử dụng mô hình mạng RNN, ta rút ra được những kết luận sau
* Ưu điểm lớn nhất của phương pháp trên đó là sự hội tụ. Đối với các phương pháp khác, ví dụ như phương pháp sử dụng giải thuật di truyền, rất có thể hệ sẽ gặp phải tình trạng rơi vào miền tối ưu địa phương, tức là chỉ tối ưu trong một vùng lân cận nhỏ. Trong khi phương pháp này đã được chứng minh là có tính ổn định toàn cục.
* Ngoài ra, việc chọn thời điểm ban đầu của các biến trạng thái nếu sử dụng các phương pháp tối ưu khác sẽ ảnh hưởng đến kết quả. Tuy nhiên đối với phương pháp này, do đã được chứng minh là tự động tiến về tối ưu toàn cục, do vậy không phụ thuộc vào điểm đầu.
* Tuy nhiên, nếu chỉ sử dụng mình thuật toán của bài báo nói trên, ta sẽ gặp phải trường hợp không xử lí được, do vậy cần phải có khâu trung gian để xử lý dữ liệu sau khi đã có được điểm hội tụ của hệ. Cần lưu ý thêm rằng việc chuyển từ ràng buộc zero – one về ràng buộc tuyến tính không phải là phép biến đổi tương đương. Do vậy khâu xử lý dữ liệu đầu ra thực sự cần thiết.
* Đối với số mục tiêu và vũ khí là khác nhau, cần phải xử lý dữ liệu đầu vào lẫn đầu ra để có thể cho ra kết quả hợp lý. Trong nhiều trường hợp có thể làm ảnh hưởng đến thời gian chạy thuật toán.
* Với kết quả đã đạt được kể trên, ta có thể phát triển thêm theo hướng sau
* Như sơ đồ nguyên lý đã nêu ở trên, ta có thể thấy giá trị của các ma trận x\_dot, w\_dot, v\_dot cũng như các ma trận x, w, v có thể được tính toán một cách độc lập và song song với nhau. Trong khi đó chương trình hiện tại lại đang tính toán một cách tuần tự.
* Như vậy, nếu có thể tính song song các giá trị của x, w, v sẽ giảm được thời gian tính toán cho các kịch bản với số lượng mục tiêu và vũ khí lớn. Một trong những cách để tính toán song song đó là sử dụng GPU với CUDA kit. Phương pháp này kết hợp với thuật toán đã có hy vọng sẽ cải thiện được phần nào tốc độ tính toán để đưa ra kết quả.

Danh mục tài liệu tham khảo

1. *Projection Recurrent Neural Network Model: A New Strategy to Solve Weapon-Target Assignment Problem Alireza Shojaeifard1 · Ali Nakhaei Amroudi1 · Amin Mansoori2 · Majid Erfanian3*
2. *Lee Z-J, Su S-F, Lee C-Y (2003) Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Trans Syst Man Cybern Part B Cybern 33*
3. *Mansoori A, Eshaghnezhad M, Effati S (2019) Recurrent neural network model: a new strategy to solve fuzzy matrix games. IEEE Trans Neural Netw Learn Syst*