

Необходимо автоматизировать поиск равновесия в чистых и смешанных стратегиях в игре с 2 участниками у каждого по 2 стратегии. В задаче будет присутствовать параметр n , от которого будет зависеть ответ.

Краткая справка

Представление игры в матричной форме:

	Игрок 2		
Игрок 1		Стратегия 1	Стратегия 2
	Стратегия 1	(a,b)	(c,d)
	Стратегия 2	(e,f)	(g,n)

Здесь отображены выигрыши 1 и 2 игрока, если они играют определенные стратегии. Смысл в том, что игрок не знает какую стратегию выбирает другой, поэтому возникает неопределенность. Если, например, первый выбрал стратегию 1, а второй стратегию 2, то выигрыш первого составит c , а второго – d .

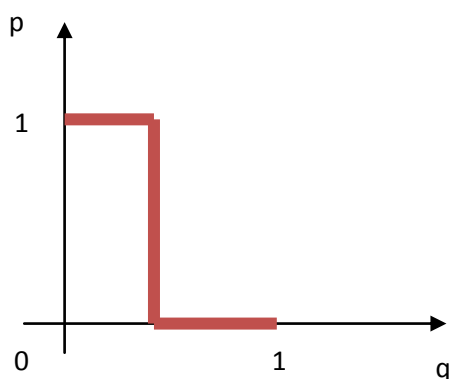
Равновесием в чистых стратегиях называется такой профиль стратегий для каждого из игрока, что ни одному из них не выгодно отклоняться (то есть, зная, что другой игрок играет равновесную стратегию, не выгодно самому играть что-то другое, отличное от равновесия).

Равновесие в смешанных это то же самое, только здесь игроки играют не определенные стратегии, а смешивают (то есть каждый выбирает вероятность с которой он играет стратегию 1). Ясно, что чистые стратегии включены в смешанные.

Будем искать все равновесия в смешанных стратегиях. Для этого обозначим вероятность того, что первый играет стратегию 1 как p , а то что 2й играет стратегию 1 как q . Тогда первый максимизирует свой ожидаемый выигрыш по той переменной, которую он выбирает сам (по p), при прочих равных условиях (воспринимая q как заданное):

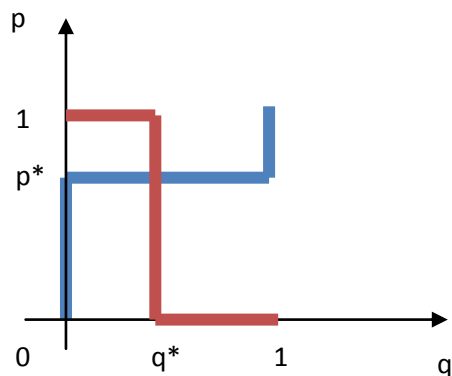
$$EU_1 = pq * a + p(1 - q) * c + (1 - p)q * e + (1 - p)(1 - q) * g \rightarrow \max$$

Ясно, что здесь надо найти производную. Если ее знак при каких то q всегда больше нуля, то при этих q $p=1$, если при каких то q производная меньше нуля, то там выбираем $p=0$, там, при том q , которое обнуляет производную, можно выбрать любое p от 0 до 1. Таким образом получаем функцию наилучшего ответа для первого игрока на действия второго. Ее график будет выглядеть примерно так (естественно, что график может быть «повернут» в другую сторону):



Теперь проделываем то же самое для второго игрока (максимизируем его ожидаемую полезность по q при заданных p), не забывая, что в нашей задаче решение будет зависеть от параметра n .

В итоге можем построить его функцию наилучшего ответа (на том же графике):



В итоге равновесия в чистых стратегиях тут не получили (графики не пересеклись в точках, где вероятности для обоих игроков или 0 или 1). Зато нашли равновесие в смешанных стратегиях (смесь стратегий 1 и 2 для игрока 1: играем стратегию 1 с вероятностью p^* , стратегию 2 с вероятностью $(1-p^*)$), аналогично для 2го. Формально ответ записывается:

$$\{p^*(\text{Стратегия 1}) \oplus (1 - p^*)(\text{Стратегия 2}); q^*(\text{Стратегия 1}) \oplus (1 - q^*)(\text{Стратегия 2})\}$$

Программа должна действовать следующим образом:

1. Пользователь забивает матрицу платежей (параметр n всегда будет расположен в одном и том же месте, его пользователь не изменяет)
2. Программа выдает ответ в формальном виде для разных групп значений n (то есть для n таких-то ответ такой-то, для других n такой-то, для третьих n такой-то). Ответ должен быть именно с условиями на n , а не для конкретных значений. То есть, эти группы значений параметров программа тоже должна находить.
3. Графики строить не обязательно, но желательно для каждой из групп значений параметров.