

СТАТИСТИЧЕСКИЙ АНАЛИЗ СТРАТЕГИЙ ИГРЫ «МОРСКОЙ БОЙ»Я.О. Дидыч¹

yaroslav.87@mail.ru

Г.Г. Малинецкий²

GMalin@keldysh.ru

¹ МГТУ им. Н.Э. Баумана, Москва, Российская Федерация² Институт прикладной математики им. М.В. Келдыша РАН, Москва, Российская Федерация**Аннотация**

Проведено сравнение алгоритмов действия атакующей и защищающейся сторон при игре «Морской бой». Вычислены статистические характеристики четырех алгоритмов атаки поля с варьируемыми параметрами, шести алгоритмов расстановки кораблей с вариацией параметров, а также трех алгоритмов добивания кораблей. Показано, что существует лучший из апробированных алгоритмов, наиболее эффективный как для случайных конфигураций кораблей, так и в случае применения алгоритмов защиты. Точно также найден лучший алгоритм расположения кораблей, обеспечивающий их защиту

Ключевые слова

Игра «Морской бой», алгоритмы расстановки кораблей, алгоритмы атаки, алгоритмы добивания, моделирование противоборства, метод Монте-Карло

Поступила в редакцию 19.09.2016
© МГТУ им. Н.Э. Баумана, 2017

Введение. Базовыми моделями для многих систем поддержки принятия решений и систем управления часто оказываются простейшие игры [1]. Анализу различных «школьных» игр специалисты по прикладной математике уделяли большое внимание [2]. Ряд исследований посвящен классической игре «Морской бой», отражающей ситуацию высокой неопределенности и неэффективности разведки, которая неоднократно возникала в военной истории [3–6], либо близким проблемам [7].

Цель настоящей работы — поиск наиболее эффективных алгоритмов, используемых для уничтожения объектов противника на поле боя, а также алгоритмов их защиты.

Исследование имеет широкий прикладной смысл, поскольку в реальной боевой обстановке фиксирование на определенной цели является сложной задачей. Выходом из положения становится стрельба в район поля действий противников согласно заданным алгоритмам.

Для симуляции боя здесь использована игра «Морской бой», на примере которой ищутся лучший алгоритм атаки объектов (кораблей), лучший алгоритм расстановки кораблей на поле боя и лучший алгоритм добивания объекта, если известно, что в него было попадание.

Эксперимент состоял в проведении симуляции игры «Морской бой» множеством алгоритмов атаки против множества алгоритмов расстановки. Игра

проходила до тех пор, пока атакующий игрок не уничтожил все корабли на поле. Основная измеряемая характеристика — количество выстрелов, необходимых для победы. По этой характеристике оценивается эффективность атаки и эффективность защиты, также на ее основе рассчитываются статистические показатели для серий игр: математическое ожидание числа шагов победы; стандартное отклонение распределения частоты побед по числу шагов, за которые эти победы достигались; процентилю указанного распределения. Согласно определению, N -й процентиль распределения величины — значение, левее которого на гистограмме распределения расположено N процентов наблюдений.

Каждый алгоритм атаки, как и каждый алгоритм расстановки кораблей на поле боя, описывается общей логикой работы и набором варьируемых параметров. Например, диагональный алгоритм атаки содержит типовую (неизменяемую) характеристику обстрела вдоль главной диагонали поля боя и изменяемые параметры, с помощью которых выбирается порядок диагоналей обстрела. Для алгоритма расстановки по краям неизменной остается необходимость разместить корабль вдоль края поля, изменяемой характеристикой — где, с точностью до координаты, будет установлен каждый корабль.

Играет каждый алгоритм атаки (в разрезе набора варьируемых параметров) с каждым алгоритмом расстановки (в разрезе набора варьируемых параметров). Отдельно для каждой игры собирается статистика алгоритмов добивания кораблей, которая затем суммируется.

Заданы следующие параметры симуляции:

- число атакующих алгоритмов в разрезе наборов параметров 818;
- число алгоритмов расстановок в разрезе наборов параметров 900;
- число симулированных сражений $900 \cdot 818 = 736200$.

Для алгоритмов атаки использовался «крестообразный» метод добивания, описанный ниже.

Правила игры «Морской бой». Играют два игрока. У каждого имеется поле размером 10×10 клеток, на котором перед началом игры располагаются корабли. Корабли бывают четырех видов:

- 1) единичный (4 шт.) — одна клетка поля;
- 2) двойной (3 шт.) — две клетки поля;
- 3) тройной (2 шт.) — три клетки поля;
- 4) четверной (1 шт.) — четыре клетки поля.

Корабли представляют собой горизонтальную или вертикальную конфигурацию клеток, расположенных по линии. Сгибать корабли запрещено. В процессе игры игроки делают выстрелы по полям расположения кораблей друг друга по очереди. Если один из игроков промахнулся — попал в поле, не занятое кораблем противника, — то ход переходит к сопернику. Пока промаха не произошло стрельба игрока продолжается.

Выигрыш одной стороны обеспечивается уничтожением кораблей противника за меньшее число шагов, чем у другой стороны. Можно выделить четыре параметра игры: 1) алгоритм атаки игрока 1; 2) алгоритм расстановки кораблей

игрока 1; 3) алгоритм атаки игрока 2; 4) алгоритм расстановки кораблей игрока 2. Найдем оптимальную стратегию одной стороны. Поскольку игрок 1 не может влиять на выбор атакующего алгоритма игрока 2, а также на выбор расстановки кораблей игрока 2, и наоборот, оптимизировать следует атакующие алгоритмы и расстановки одной из сторон без учета действий другой стороны.

Для удобства обозначения примем нумерацию строк матрицы поля расположения кораблей за X ($X = 0, \dots, 9$), колонок матрицы поля за Y ($Y = 0, \dots, 9$).

Алгоритмы расстановки. В процессе симуляции рассмотрено шесть алгоритмов расстановки кораблей:

- 1) по краям поля;
- 2) с одной стороны поля;
- 3) случайно в одном направлении;
- 4) случайно в двух направлениях;
- 5) полностью случайная;
- 6) в центре поля.

Расстановка кораблей по краям поля. Этот алгоритм расстановки выделяет полосу у края поля шириной одна клетка, т. е. координаты $X = 0, Y = 0, \dots, 9; X = 9, Y = 0, \dots, 9; X = 0, \dots, 9, Y = 0; X = 0, \dots, 9, Y = 9$. В рамках выделенной полосы случайно размещаются все корабли, кроме единичных. После размещения кораблей по всему полю в доступных клетках размещаются единичные корабли.

Расстановка кораблей с одной стороны поля. Такой алгоритм размещения заключается в выделении с произвольной стороны поля области шириной четыре клетки (например, $X = 0, \dots, 3, Y = 0, \dots, 9$). Затем в выделенной области произвольно размещаются все корабли, кроме единичных. Ограничение накладывается на ориентацию корабля (вертикальность или горизонтальность). Если выделены крайняя правая или левая область поля, то ориентация кораблей вертикальна, если верхняя или нижняя, то — горизонтальна. Если корабли в указанных начальных ограничениях шириной четыре клетки невозможно разместить на одном из шагов алгоритма размещения, то ширина поля увеличивается на единицу до тех пор, пока все корабли, кроме единичных, не будут размещены. В последнюю очередь размещаются единичные корабли на всем пространстве поля случайным образом.

Расстановка кораблей случайно в одном направлении. Этот алгоритм выбирает ориентацию всех кораблей (вертикальность или горизонтальность) и размещает корабли случайно по всему полю в заданной ориентации.

Расстановка кораблей случайно в двух направлениях. Алгоритм реализован так, чтобы после размещения на поле половина клеток кораблей, кроме единичных, принадлежала вертикальным кораблям, половина — горизонтальным. В результате при реализации алгоритма выбирается ориентация одного четверного и двух двойных кораблей, а остальные неединичные корабли имеют другую ориентацию.

Полностью случайная расстановка кораблей. В таком алгоритме размещения случайным образом выбирается как ориентация, так и координаты кораблей.

Расстановка кораблей в центре поля. Алгоритм выполнен так, чтобы максимально сгруппировать корабли в центре поля. Для этого выбирается квадрат в центре поля с координатами $X = \{4, 5\}$; $Y = \{4, 5\}$. Делается попытка разместить корабли в указанной области. Сперва размещаются единичные корабли, затем двойные, потом тройные и четверной. Если в выделенной области отсутствует место для размещения корабля, ее границы увеличиваются на единицу.

Каждый алгоритм расстановки кораблей представлен 150 вариациями. Результаты симуляции для алгоритмов расстановки приведены в табл. 1.

Таблица 1

Статистические результаты симуляции алгоритмов расстановки кораблей

Алгоритм расстановки кораблей	Математическое ожидание s	Стандартное отклонение σ	Процент выборки в интервале $(s - \sigma; s + \sigma)$	Процентиль			
				95	85	15	5
По краям поля	68,10	7,92	74,67	81	76	60	55
С одной стороны поля	66,30	8,72	72,31	81	75	57	52
Случайно в одном направлении	62,34	8,40	72,52	77	71	54	49
Случайно в двух направлениях	62,10	7,9	70,58	75	70	54	49
Полностью случайная	62,10	8,00	70,68	75	70	54	49
В центре поля	60,40	8,75	72,48	75	69	51	46

Согласно данным, приведенным в табл. 1, наибольшее математическое ожидание числа шагов потопления всех кораблей у алгоритма расстановки кораблей по краям поля, наименьшее — у алгоритма группировки кораблей в центре поля.

Для алгоритмов расстановки кораблей также были вычислены процентиля распределения (см. табл. 1). На основе процентилей можно сделать вывод, что гистограмма распределения результатов алгоритма расстановки кораблей по краям поля смещена в сторону увеличения шагов уничтожения поля, что подтверждает превосходство метода расстановки. Гистограмма расстановки кораблей в центре смещена в противоположную сторону.

Кроме игры против спектра исследуемых алгоритмов атаки были проведены расчеты при игре алгоритмов расстановки против 818 случайных алгоритмов атаки (метод Монте-Карло). На основе полученных результатов лидер остался неизменен — алгоритм расстановки кораблей по краям поля.

Вывод сделан на основе сравнения математических ожиданий числа шагов для победы над расстановкой, а также на основе сравнения процентилей распределений соответствующих алгоритмов защиты (табл. 2).

Таблица 2

Статистические результаты алгоритмов расстановки кораблей против случайных алгоритмов атаки

Алгоритм расстановки кораблей	Математическое ожидание s	Стандартное отклонение σ	Процент выборки в интервале $(s - \sigma; s + \sigma)$	Процентиль			
				95	85	15	5
По краям поля	66,08	7,95	71,81	78	74	58	52
С одной стороны поля	64,46	8,37	69,34	78	73	56	50
Случайно в одном направлении	60,31	8,18	70,81	74	69	52	47
Случайно в двух направлениях	60,40	7,62	73,80	73	68	53	48
Полностью случайная	60,70	7,51	71,36	73	68	53	48
В центре поля	60,34	7,41	72,36	73	68	53	48

Распределение алгоритма расстановки кораблей по краям поля смещено вправо, на что указывает наибольшее значение процентилей распределения числа выигрышей по числу затраченных шагов.

Если говорить про худший вариант при тестировании против спектра алгоритмов атаки — расстановка в центре поля — то при тестировании против случайных алгоритмов атаки результаты не столь заметны. У четырех расстановок с наименьшим математическим ожиданием числа шагов уничтожения поля результаты математических ожиданий приблизительно равны (см. табл. 2). Стандартные отклонения и проценты выборки в интервале одного стандартного отклонения от математического ожидания также не сильно отличаются. Похожая ситуация и при рассмотрении распределения результатов с позиции процентилей.

Учитывая изложенное выше, по итогам двух тестирований алгоритмов расстановки, безоговорочным лидером является расстановка кораблей по краям поля. Расстановка кораблей в центре поля — безусловный аутсайдер тестирования против спектра алгоритмов атаки, и входит в число четырех худших алгоритмов расстановки против случайного метода атаки.

Алгоритмы атаки. В настоящей работе из алгоритмов атаки были рассмотрены диагональный, линейный (обход змейкой), случайный и квадратный (обход по квадратам) алгоритмы.

Диагональный алгоритм характеризуется выстрелами по диагоналям, параллельным главной диагонали поля расположения кораблей. В алгоритме используются два параметра. Первый — координата Y серии выстрелов, второй — шаг

для серии выстрелов. Для того чтобы обеспечить полный прострел поля, необходимо иметь несколько серий, характеризующихся координатами Y первой ячейки выстрела и шагом стрельбы. Координата X первой ячейки выстрела всегда равна нулю. Когда выбрана ячейка для выстрела, происходит стрельба параллельно главной диагонали с постоянным шагом единица. После достижения края поля вновь выбирается первая ячейка для новой диагонали обстрела с учетом шага. Обход осуществляется справа налево. Если для заданного шага и начальной диагонали обстрела выстрелы более невозможны, выполняется переход к следующему шагу и новой начальной координате отсчета диагоналей.

Линейный алгоритм характеризуется выстрелами снизу вверх с определенным шагом. В качестве варьируемых параметров используются начальная координата (например, $(0, 1)$) и шаг (например, 5). В дальнейшем идет обстрел первой колонки поля, затем второй, третьей с шагом 5 , пока не будет достигнут конец поля. Например, первый выстрел $(0, 1)$, второй $(0, 6)$, третий $(1, 2)$ и т. д. После достижения конца поля выбираются новая начальная координата и шаг, вновь происходит обстрел. Так продолжается до тех пор, пока все корабли не будут уничтожены.

Случайный алгоритм характеризуется полностью случайной стрельбой. Координаты (X, Y) выбираются произвольно в диапазоне $0, \dots, 9$.

Квадратный алгоритм заключается в обстреле поля противника по квадратам. Сначала обстреливаются четыре угловых точки квадрата, затем четыре соседние к ним по часовой стрелке, после этого соседние к обстрелянным на предыдущем шаге и т. д. Варьируемыми параметрами являются размер стороны квадрата и координаты первой точки угла.

В разрезе параметров из алгоритмов атаки симулировались:

- 100 разновидностей диагонального алгоритма;
- 49 разновидностей квадратного алгоритма;
- 269 разновидностей линейного алгоритма;
- 400 разновидностей случайного алгоритма.

В ходе проведенных симуляций были получены статистические результаты для алгоритмов атаки (табл. 3).

Таблица 3

Статистические результаты для алгоритмов атаки

Алгоритм	Математическое ожидание s	Стандартное отклонение σ	Процент выборки в интервале $(s - \sigma; s + \sigma)$	Процентиль			
				95	85	15	4
Диагональный	56,19	7,80	68,79	70	64	48	44
Случайный	62,39	8,20	70,06	76	71	54	49
Линейный	67,01	7,40	71,80	79	75	59	55
Квадратный	68,87	8,84	70,01	86	78	60	56

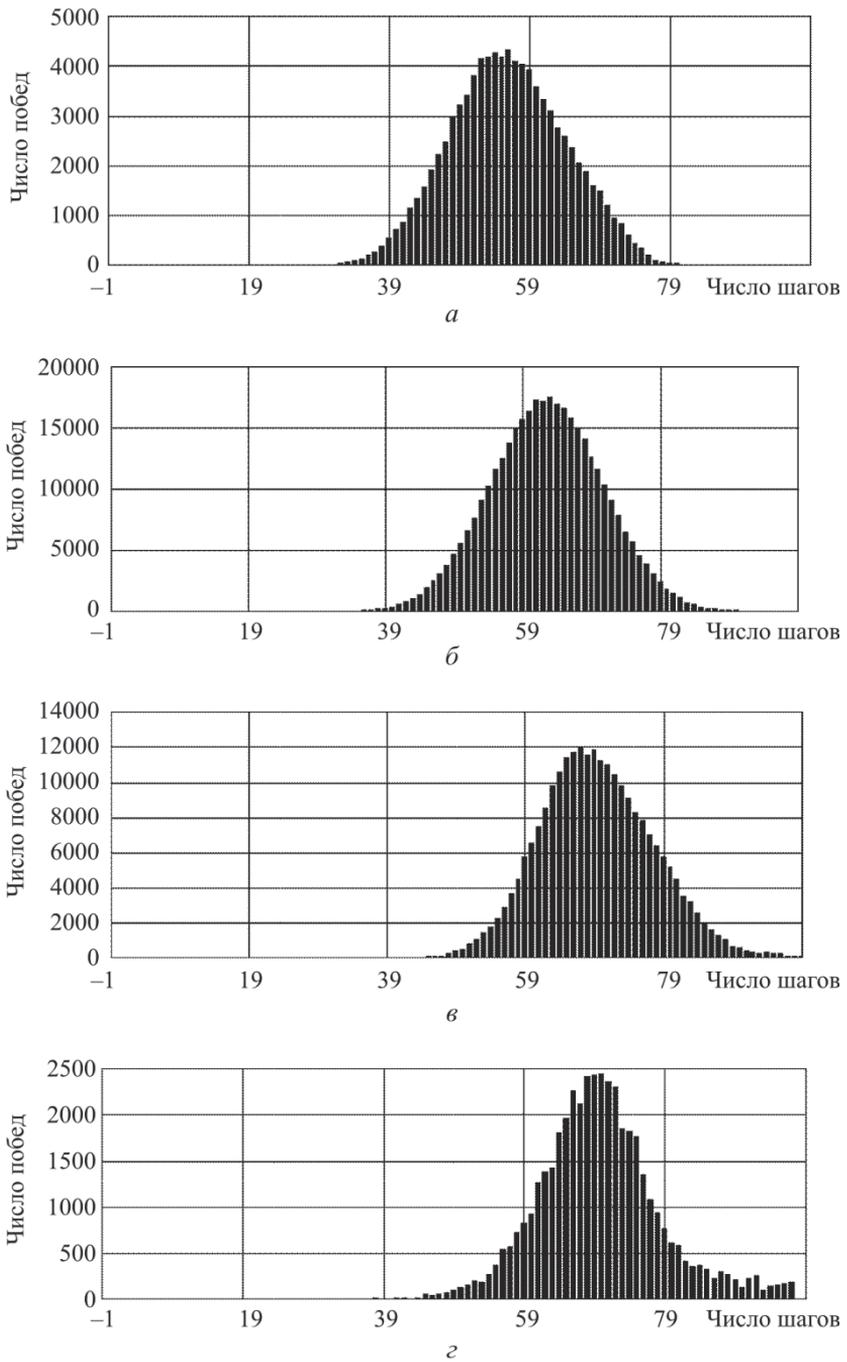


Рис. 1. Гистограммы диагонального (*a*), случайного (*б*), линейного (*в*) и квадратного (*г*) алгоритмов

Вычислено распределение числа побед по числу шагов для каждого набора варьируемых параметров, а затем результаты распределения просуммированы для каждого алгоритма.

Согласно данным, приведенным в табл. 3, наименьшее математическое ожидание шагов для уничтожения поля противника у диагонального алгоритма, наибольшее — у квадратного алгоритма. Стандартные отклонения распределений у всех алгоритмов атаки примерно одинаковы, как и процент выборки в соответствующих интервалах отклонений.

Значение процентилей распределения диагонального алгоритма атаки меньше соответствующих значений процентилей конкурирующих алгоритмов (см. табл. 3). Поскольку показатели стандартных отклонений алгоритмов отличаются незначительно, это свидетельствует о смещении гистограммы диагонального алгоритма влево при сохранении ее формы по сравнению с другими алгоритмами атаки. Аналогично гистограмма квадратного алгоритма атаки смещена вправо — в сторону наибольших шагов, необходимых для победы. Смещение наглядно продемонстрировано на рис. 1.

Кроме сражений против указанных алгоритмов расстановок были проведены расчеты игры алгоритмов атаки против 900 случайных (метод Монте-Карло) алгоритмов расстановки кораблей. Ранжирование кораблей осталось прежним (табл. 4). При игре против случайной расстановки конфигураций у алгоритмов атаки уменьшилось математическое ожидание числа шагов победы, однако лидер и аутсайдеры не изменились. Подобные изменения коснулись и расчета значений процентилей выборок результатов алгоритмов: большинство соответствующих значений стали меньше, но при этом ранжирование не изменилось.

Таблица 4

Статистические результаты алгоритмов атаки при игре со случайным алгоритмом расстановки кораблей

Алгоритм	Математическое ожидание s	Стандартное отклонение σ	Процент выборки в интервале $(s-\sigma; s+\sigma)$	Процентиль			
				95	85	15	5
Диагональный	53,19	5,19	75,39	62	59	48	45
Случайный	61,54	7,83	72,66	75	70	54	49
Линейный	62,84	6,77	69,95	75	70	56	52
Квадратный	65,52	7,25	70,47	78	73	58	54

Как и в случае тестирования против спектра алгоритмов защиты, так и в случае тестирования против случайных алгоритмов расстановки, наихудший и наилучший алгоритмы атаки одни и те же. Лучший алгоритм атаки — диагональный, худший — квадратный.

Алгоритмы добивания. Проанализированы три вида алгоритмов добивания.

Первый алгоритм — *крестообразное добивание*. Сначала делается попытка добивать вверх от переданной точки попадания. Добивание идет последовательно клетка за клеткой, пока не происходит промах. Если при движении вверх было хотя бы одно попадание, то делается вывод о вертикальном расположении корабля.

После промаха при движении вверх и в случае установления вертикальности корабля идет движение вниз до первого промаха. После промаха при движении вверх и в случае неопределенности ориентации корабля идет движение вправо. В случае промаха при движении вправо дальнейшее движение продолжается влево от начальной точки попадания. После промаха при движении прицела добивания влево начинается движение вниз. Таким образом, будут использованы все варианты расстановки кораблей. На каждом шаге добивания делается проверка гипотезы о том, что корабль не уничтожен. Если в результате проверки устанавливается подбитие всех клеток корабля, добивание прекращается.

Второй алгоритм добивания — *максимальное исключение* — добивание в сторону, где в случае попадания в следующую клетку будет исключено наибольшее число полей вокруг. При подсчете предполагается, что убитая клетка будет крайняя для корабля.

Третий алгоритм добивания — *минимальное исключение* — аналогичен второму алгоритму, за исключением правила, в соответствии с которым выбор делается в сторону клетки, при уничтожении которой будет исключено наименьшее число клеток вокруг нее.

При симуляции алгоритмов добивания в результатах учитывались все выстрелы: как промахи, так и попадания в процессе добивания.

Применение алгоритма максимального исключения при добивании уменьшает ожидаемое число шагов атаки. Стандартное отклонение этого алгоритма также является наименьшим, что свидетельствует в пользу его использования вместо крестообразного алгоритма или алгоритма минимального исключения. Данные, приведенные в табл. 5, подтверждают этот вывод: значения всех вычисленных процентилей алгоритма максимального исключения наименьшие, т. е. распределение числа уничтоженных кораблей по числу необходимых шагов смещено влево по сравнению с другими алгоритмами добивания.

Таблица 5

Статистические результаты для алгоритмов добивания

Алгоритм	Математическое ожидание s	Стандартное отклонение σ	Процент выборки в интервале $(s - \sigma; s + \sigma)$	Процентиль			
				95	85	15	5
Крестообразный	17,77	3,09	77,19	23	21	15	13
Максимальное исключение	16,79	2,64	74,12	21	19	14	12
Минимальное исключение	18,40	3,28	71,31	24	22	15	13

Наихудший алгоритм добивания — алгоритм наименьшего исключения дополнительных клеток вокруг корабля с учетом предположения, что поражаемая клетка будет крайней. Это следует из математического ожидания числа шагов уничтожения кораблей для указанного алгоритма, а также из общего смещения распределения результатов в сторону большего числа шагов (см. табл. 5).

Лучшие алгоритмы рассмотренных действий наглядно показаны на рис. 2.

Взгляд с позиции теории игр. Рассмотренные алгоритмы были проанализированы с позиции применения теории игр. Применялись те же алгоритмы расстановки кораблей и атаки, при этом варьировался размер поля. Использованы поля размерами 10×10 , 20×20 , 30×30 . Получены платежные матрицы, представленные в табл. 6–8.

Элементы платежных матриц — математическое ожидание игры чистой стратегии атаки и защиты между собой. При этом число используемых вариаций параметров для стратегий атаки составляет 150; для защиты — 100.

Матрицы, приведенные в табл. 6, 7, имеют седловую точку на пересечении случайного алгоритма атаки и алгоритма расстановки по краям поля. Платежная матрица, представленная в табл. 8, отличается от двух предыдущих: она не имеет седловой точки, но решается в смешанных стратегиях. Решение представляет собой вектор $(0,728; 0,003; 0; 0; 0; 0,269)$ для атаки и вектор $(0,022; 0,978; 0; 0)$ для расстановки (защиты). В таком случае цена игры 721,73. Отличие платежных матриц игр на полях размерами 30×30 и 10×10 может быть объяснено нарушением отношения размеров кораблей к размеру поля игры.

Таблица 6

Платежные матрицы игры на поле размерами 10×10

Алгоритм расстановки кораблей	Алгоритм атаки			
	Диагональный	Случайный	Линейный	Квадратный
По краям поля	68,14	65,48	69,84	79,79
С одной стороны поля	60,29	64,45	68,43	69,61
Случайно в одном направлении	53,28	60,88	63,79	70,72
Случайно в двух направлениях	53,56	60,16	64,32	69,47
Полностью случайная	52,34	61,51	65,59	68,87
В центре поля	48,86	60,04	63,26	69,08

Таблица 7

Платежная матрица игры на поле размерами 20×20

Алгоритм расстановки кораблей	Алгоритм атаки			
	Диагональный	Случайный	Линейный	Квадратный
По краям поля	333,20	312,67	366,60	375,31
С одной стороны поля	305,44	311,60	368,49	369,93
Случайно в одном направлении	302,34	283,41	350,27	353,56
Случайно в двух направлениях	293,62	297,35	353,72	354,06
Полностью случайная	289,04	303,77	351,28	351,06
В центре поля	247,06	305,19	358,35	361,63

Платежная матрица игры на поле размерами 30×30

Алгоритм расстановки кораблей	Алгоритм атаки			
	Диагональный	Случайный	Линейный	Квадратный
По краям поля	774,16	720,60	864,59	873,24
С одной стороны поля	777,72	718,05	864,56	872,16
Случайно в одном направлении	715,73	703,51	849,00	853,80
Случайно в двух направлениях	722,25	701,97	857,44	853,25
Полностью случайная	714,47	692,62	850,27	848,12
В центре поля	591,80	724,66	858,80	861,10

Интерпретируя полученный статистический результат с позиции теории игр, следует отметить, что если сторона защиты (расстановки) действует не оптимальным для себя способом, применяя смешанную стратегию с вектором $(1/6; 1/6; 1/6; 1/6; 1/6; 1/6)$, то стороне атаки выгодно отклониться от своей оптимальной чистой стратегии и применять диагональный алгоритм.

Выводы. Из алгоритмов расстановки кораблей наилучшие статистические результаты у алгоритма расстановки кораблей по краям поля, т. е. наибольшее число шагов для уничтожения всех кораблей с подобной расстановкой и наибольшее смещение распределения вправо относительно других алгоритмов расстановки. Наихудшие результаты у алгоритма расстановки кораблей в центре поля, причем алгоритм расстановки кораблей в центре показывает безоговорочно худшие результаты при проверке против групп тестируемых алгоритмов атаки, однако является лишь одним из худших при тестировании против случайных алгоритмов. Алгоритм расстановки кораблей по краям поля подтвердил эффективность как в игре против моделируемых специальных алгоритмов атаки, так и против алгоритма случайной стрельбы (метод Монте-Карло).

Алгоритм атаки, показывающий лучшие статистические характеристики, — диагональный алгоритм. Он имеет наименьшее математическое ожидание уничтожения поля противника, а также наилучшие показатели смещения распределения относительно случайного, линейного и квадратного алгоритмов. Диагональный алгоритм подтверждает лидерство и при тестировании против случайного алгоритма расстановки кораблей. Худшим алгоритмом атаки является квадратный алгоритм, причем он худший и при тестировании как против выбранных алгоритмов защиты, так и против алгоритмов случайной расстановки кораблей.

Наилучшие показатели у алгоритма максимального исключения из дальнейшей игры наибольшего числа клеток. Наихудшие результаты у алгоритма минимального исключения клеток, если попадание произойдет и уничтоженная клетка будет крайней для корабля.

Прикладной смысл исследования заключается в том, что при увеличении точности определения целей на поле боя значительно усложняется аппаратура. Наиболее простой способ обойти такую проблему — использование оптимальных алгоритмов атаки на все цели поля боя, которые были определены выше.

Авторы выражают благодарность А.В. Подлазову, обратившему внимание авторов на методы статистического анализа больших серий расчетов и важность алгоритмов добывания, а также А.А. Кочкарову, увидевшему связь рассмотренной задачи с проблемами управления многоагентными мобильными системами.

ЛИТЕРАТУРА

1. Ларичев О.И. Теория и методы принятия решений, а также Хроника событий в Волшебных странах. М.: Логос, 2002. 392 с.
2. Berlekamp E.R., Conway G.H., Guy R.K. Winning ways for your mathematical plays. Vol. I. Wellesley, Massachusetts: A.K. Peters, 2001. 293 p.
3. Хемди А. Таха. Введение в исследование операций; пер. с англ. М.: Издательский дом «Вильямс», 2005. 912 с.
4. Евменов В.П. Интеллектуальные системы управления. М.: Книжный дом «ЛИБРОКОМ», 2009. 304 с.
5. Мусеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации. М.: Наука, 1978. 352 с.
6. Вильямс Дж.Д. Совершенный стратег, или Букварь по теории стратегических игр / пер. с англ. М.: Советское радио, 1960. 270 с.
7. Sevenster M. Battleships as decision problem // ICGA Journal. 2004. Vol. 27. No. 3. P. 142–149.

Дидыч Ярослав Олегович — магистрант кафедры «Прикладная математика» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская, д. 5, стр. 1).

Малинецкий Георгий Геннадьевич — д-р физ.-мат. наук, профессор Института прикладной математики им. М.В. Келдыша РАН (ИПМ РАН) (Российская Федерация, 125047, Москва, Миусская пл., д. 4).

Просьба ссылаться на эту статью следующим образом:

Дидыч Я.О., Малинецкий Г.Г. Статистический анализ стратегий игры «Морской бой» // Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки. 2017. № 4. С. 122–135.
DOI: 10.18698/1812-3368-2017-4-122-135

STATISTICAL ANALYSIS OF "BATTLESHIP" STRATEGIES

Ya.O. Didych¹

yaroslav.87@mail.ru

G.G. Malinetskiy²

GMalin@keldysh.ru

¹Bauman Moscow State Technical University, Moscow, Russian Federation

²Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, Russian Federation

Abstract

The article gives the comparison between attacking and defensive algorithms in "Battleship" game. We calculated statistical characteristics of four attacking algorithms with variable parameters, as well as six defensive algorithms with variable parameters and three algorithms of finishing off the ships. Findings of the research show that there exists the most effective algorithm both for random ship configurations, and when specific defensive algorithms are applied. Similarly, the best defensive algorithm for locating ships is provided

Keywords

"Battleship" game, ship placement algorithms, attacking algorithms, finishing off algorithms, simulating confrontation, Monte-Carlo method

REFERENCES

- [1] Larichev O.I. Teoriya i metodi prinyatiya resheniy, a tackzhe Hronika sobitiy v Volshebnyh stranah [Theory and methods of decision making and Chronicles of events in Magic countries]. Moscow, Logos Publ., 2002. 392 p.
- [2] Berlekamp E.R., Conway G.H., Guy R.K. Winning ways for your mathematical plays. Vol. I. Wellesley, Massachusetts, A.K. Peters, 2001. 293 p.
- [3] Hamdy A. Taha. Operations research: An introduction. Pearson Education, New Jersey, 2003. 832 p.
- [4] Evmenov V.P. Intellektualnie sistemi upravleniya [Intelligent control systems]. Moscow, "LIBROKOM" Book house, 2009. 304 p.
- [5] Moiseev N.N., Ivanilov Yu.P., Stolyarova E.M. Metodi optimizatsii [Optimization methods]. Moscow, Nauka Publ., 1978. 352 p.
- [6] Williams J.D. The Complete strategist: Being a primer on the theory of games of strategy. New York, Dover Publications, 1986. 268 p.
- [7] Sevenster M. Battleships as decision problem. *ICGA Journal*, 2004, vol. 27, no. 3, pp. 142–149.

Didych Ya.O. — Master of Applied Mathematics Department, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Malinetckiy G.G. — Dr. Sc. (Phys.-Math.), Professor of Keldysh Institute of Applied Mathematics, Russian Academy of Sciences (Miuskaya ploshchad' 4, Moscow, 125047 Russian Federation).

Please cite this article in English as:

Didych Ya.O., Malinetckiy G.G. Statistical Analysis of "Battleship" Strategies. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Estestv. Nauki* [Herald of the Bauman Moscow State Tech. Univ., Nat. Sci.], 2017, no. 4, pp. 122–135. DOI: 10.18698/1812-3368-2017-4-122-135