

А. Д. Тарасов, ст. преподаватель каф. "Автоматизированные системы обработки информации и управления", e-mail: adtarasov@mail.ru  
Оренбургский государственный аграрный университет

## Эффективность работы генетического алгоритма в задаче проектирования систем физической защиты

*Рассмотрен процесс решения задачи проектирования систем физической защиты с помощью стандартного генетического алгоритма. Выделены параметры алгоритма с наибольшим влиянием на процесс решения. Описаны эксперименты, доказывающие необходимость определения значений этих параметров с помощью адаптивного генетического алгоритма.*

**Ключевые слова:** система физической защиты, поддержка принятия решений, стандартный генетический алгоритм, адаптивный генетический алгоритм

### Введение

Под системами физической защиты (СФЗ) понимается совокупность инженерно-технических, сигнализационных, программно-аппаратных и иных средств для защиты имущества или объектов от хищений, диверсий и других неправомерных действий, позволяющая на заданном уровне создавать трудности, ограничивать возможности и увеличивать время проникновения нарушителя на объект охраны [1].

Критически важными объектами (КВО) являются объекты, нарушение функционирования которых приводит к потере управления, разрушению инфраструктуры, необратимому негативному изменению экономики субъекта или административной территориальной единицы либо существенному ухудшению на длительный период времени безопасности жизнедеятельности населения, проживающего на этих территориях [2].

Обеспечение безопасности КВО является важнейшей задачей, и процесс создания СФЗ КВО требует для проектировщика использования любых доступных средств, которые могут предоставить современные информационные технологии. Одним из таких средств является поддержка принятия решений.

Задача поддержки принятия решений при проектировании СФЗ и их анализе требует информации, отражающей опыт и знания экспертов, предоставления возможных вариантов решения, оценки этих решений и обоснованного выбора из них. В задаче принятия решения о составе СФЗ самым сложным и плохо формализуемым этапом, требующим применения процедур, которые используют знания экспертов, является концептуальное проектирование, под которым понимается анализ защищенности объекта, разработка практических рекоменда-

ций по созданию СФЗ, выбор варианта и состава инженерно-технических средств охраны (ИТСО).

Для решения задач такого рода предлагается использовать генетические алгоритмы (ГА). Так как задача обеспечения защиты критически важных объектов требует использования высокоэффективных методов решения, то при разработке ГА необходимо обеспечить максимальное качество работы алгоритма.

Генетические алгоритмы имеют множество преимуществ по сравнению с другими методами оптимизации [3], но также не лишены недостатков, один из которых — сильная зависимость эффективности работы алгоритма от выбранных параметров ГА. Такими параметрами являются вероятность кроссинговера, вероятность мутаций, число точек деления хромосом при кроссинговере и т. п. Обычно поиск лучших значений параметров проводится интуитивно, методом проб и ошибок, что приводит к выбору не самых удачных вариантов. Проведение экспериментов с оценкой работоспособности алгоритма при различных значениях параметров часто решает эту проблему.

Дополнительная сложность при определении наилучших значений параметров алгоритма вызывается следующей особенностью ГА: влияние многих параметров на эффективность работы изменяется на протяжении всего времени работы ГА. Следовательно, нужно определить не постоянные значения, а правила изменения параметров в процессе поиска решения. Например, желательно уменьшать вероятность мутации в зависимости от времени работы генетического алгоритма [4]. Кроме того, изменение параметров ГА может потребоваться при смене исходных данных задачи.

Таким образом, поиск наилучших значений параметров ГА надо проводить при каждом его за-

пуске, что возможно при использовании так называемых адаптивных генетических алгоритмов.

Выдвигаем предположение, что для решения задачи проектирования СФЗ критически важных объектов требуется разработка ГА с обязательной реализацией механизма адаптации.

В данной работе анализируется процесс решения задачи выбора состава ИТСО с помощью стандартного ГА, описанный в работе [5]. Выделены параметры ГА, значения которых сильно влияют на эффективность работы алгоритма, и проведены эксперименты, доказывающие необходимость определения значений этих параметров с помощью адаптивного ГА.

### 1. Использование ГА в задаче выбора состава ИТСО объекта

Понятие генетического алгоритма и особенности стандартного ГА описаны в работах [3] и [6].

Для выбора состава ИТСО с помощью стандартного ГА применяется логическая модель, которая представляет структуру объекта в виде графа, где вершины — это зоны объекта, а ребра — рубежи. Часть зон является критическими элементами (КЭ) объекта, требующими защиты. Защита КЭ обеспечивается комплексом ИТСО, состав которого представляется в виде множества точек контроля (ТК). Физически каждая ТК может включать в себя несколько средств комплекса ИТСО, выполняющих одну общую функцию: точка обнаружения (ТО); точка доступа (ТД); точка видеонаблюдения (ТВ); точка задержки (ТЗ). Итого можно выделить четыре типа точек контроля, причем для защиты одной зоны может потребоваться несколько точек. Решение задачи нахождения оптимального расположения ТК на графе объекта подробно описано в работе [5].

Целевая функция ГА данной задачи состоит из двух частей. Первая часть функции отвечает за то, чтобы набор ТК на каждом из всех возможных путей к КЭ соответствовал требуемому набору ТК для данного КЭ или превышал его, т. е. все КЭ должны быть защищены. Вторая часть функции отвечает за минимизацию общего числа ТК на объекте (чем меньше ТК, тем меньше затраты на создание комплекса ИТСО).

Для получения оптимального решения соблюдение условий первой части целевой функции важнее, чем соблюдение условий второй части, так как если не все КЭ защищены, проект системы защиты непригоден независимо от затрат. Поэтому в задаче используются так называемые уровни влияния частей целевой функции [5]. Два числа  $A$  и  $B$ , соотношение которых задает превышение значения одной части целевой функции над другой, участвуют в расчете эффективностей хромосом:

$$\mu(h_i) = (\mu_1(h_i) \times A + \mu_2(h_i) \times B) / (A + B),$$

где  $\mu_1(h_i)$  — значение эффективности  $i$ -й хромосомы для первой части целевой функции,  $\mu_2(h_i)$  — значение эффективности для второй части целевой функции.

Чем больше разница между числами  $A$  и  $B$ , тем сильнее одна часть целевой функции влияет на эффективность хромосом (и на получаемые решения). Делаем вывод, что уровни влияния частей целевой функции являются одними из параметров ГА, от которых сильно зависит эффективность работы ГА. Эксперименты, представленные ниже, показали, насколько меняется скорость поиска оптимального решения при изменении отношения уровней влияния.

Вероятность мутации также является одним из значимых параметров ГА, оптимальное значение которого непредсказуемо, но влияет как на скорость работы, так и на саму возможность получения оптимального решения.

Таким образом, в процессе решения задачи выбора состава ИТСО объекта с использованием стандартного ГА определяем два параметра, значения которых, предположительно, являются наиболее важными для эффективной работы алгоритма. Это уровни влияния частей целевой функции и вероятность мутации хромосом.

### 2. Оценка оптимальности решения, полученного с помощью ГА

Набор исходных данных для тестирования работоспособности алгоритма описывает условный объект (рис. 1). Граф объекта достаточно простой для поиска путей вручную, что будет использовано при оценке оптимальности решения. Исходные данные следующие: точки проникновения — зона 1 и 4, критические элементы — зона 10 и 12. Ограничения на число ТК у всех вершин и ребер одинаковое: минимальное — ноль, максимальное — три для ТК каждого типа. Требования по ТК для критических элементов выбраны случайным образом (табл. 1).

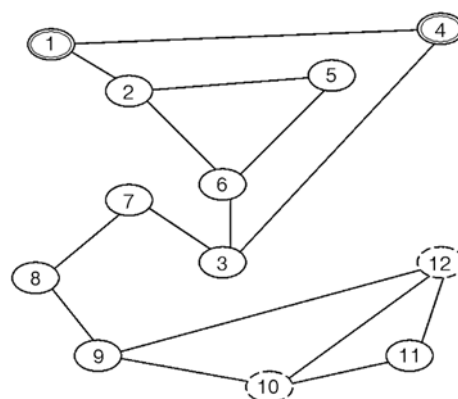


Рис. 1. Граф условного объекта

Таблица 1

Требования по числу ТК для условного объекта

Критические элементы	Точки обнаружения (ТО)	Точки доступа (ТД)	Точки видеонаблюдения (ТВ)	Точки задержки (ТЗ)
Зона 10	25	1	17	4
Зона 12	10	17	16	3

**Набор точек контроля в оптимальном решении для условного объекта**

Зона или рубеж	ТО	ТД	ТВ	ТЗ
Зона 3	3	3	2	0
Зона 4	1	0	0	0
Зона 7	3	3	1	1
Зона 8	3	3	3	0
Зона 9	3	3	3	1
Зона 10	3	0	1	1
Зона 12	0	2	0	0
Рубеж 6—3	1	0	0	0
Рубеж 3—7	3	1	3	1
Рубеж 7—8	3	2	1	0
Рубеж 8—9	3	0	3	0
Сумма	26	17	17	4

Для проверки оптимальности полученного генетическим алгоритмом результата нужно найти решение задачи вручную. Если существует несколько оптимальных решений (с одинаковым соответствием целевой функции), то необходимо определить, какие общие свойства должны иметь решения, чтобы считаться оптимальными. На основе рассчитанных свойств можно доказать, что результат, полученный ГА, оптимален.

Проанализируем задачу с текущим набором исходных данных. Оптимальное решение должно максимально возможно соответствовать целевой функции (защита всех КЭ при минимально возможном числе ТК). Определяем свойства оптимального решения через следующие умозаключения.

Число ТК на всех путях (на всем объекте) в идеальном случае строго равно требуемому числу ТК для КЭ (меньшее число не сможет защитить все КЭ, большее нежелательно по правилам второй части целевой функции). Если КЭ несколько, то в идеальном наборе ТК число ТК каждого типа равно максимальному среди требований для защиты каждого КЭ. Другими словами, минимальное число ТК на объекте должно соответствовать или превышать требуемое для защиты каждого КЭ, а значит, в идеале равно наибольшему числу среди заданных наборов ТК, требуемых для всех КЭ. По табл. 1 определяем идеальный набор ТК на всем объекте: 25 ТО, 17 ТД, 17 ТВ, 4 ТЗ.

Теперь проверим, возможно ли защитить все пути таким минимально допустимым набором точек. Фактически требуется найти вершины или ребра, через которые идут все пути, и расставить на них нужное число ТК. Таких вершин и ребер в графе семь. Все они расположены на участке графа между вершинами 3 и 9: 3—7—8—9. Ограничения на число ТК в вершинах и ребрах позволяют разместить на этом участке не более 21 ТК каждого вида (по три на вершинах и ребрах). Этого достаточно для ТД, ТВ и ТЗ, значит, их число в оптимальном решении равно 17, 17 и 4 соответственно.

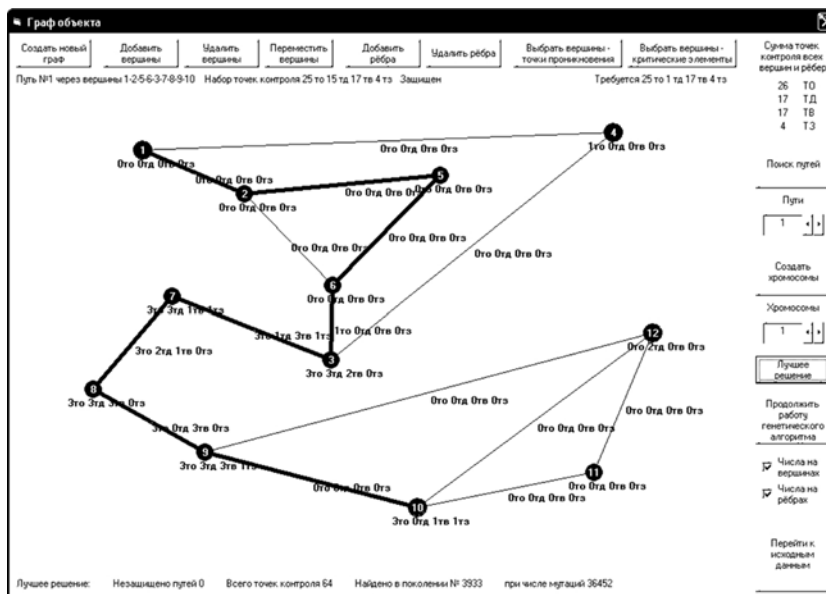
Нужное число точек обнаружения не помещается на выделенном участке графа. 21 ТО в идеальном решении расположены на этом участке. Оставшиеся четыре ТО требуют поиска оптимального расположения в других участках. Эти точки должны защищать только зону 10, так как для защиты зоны 12 достаточно 21 точки, которые уже размещены. Расположим три допустимые по ограничениям ТО в самой зоне 10. Остается одна недостающая ТО.

Согласно графу объекта одна ТО в любых вершинах и ребрах, оставшихся не заполненными точками обнаружения, не сможет защитить все пути. Граф разветвляется на два направления до и после участка 3—7—8—9. От-

сюда вывод: необходимо разместить на графе минимум две ТО для защиты всех путей. Эти точки могут быть расположены практически в любых оставшихся незанятыми вершинах или ребрах, но только с условием, что любой путь пройдет хотя бы через одну из двух точек. Например, участки графа 9—12; 9—10 или 3—6; 3—4 или обе точки проникновения — зоны 1 и 4.

Таким образом, в оптимальном решении должно быть на одну ТО больше, чем в идеальном наборе: 26 ТО, 17 ТД, 17 ТВ, 4 ТЗ — всего 64 точки контроля. Если генетическим алгоритмом будет найдена хромосома с подобным набором ТК, значит, алгоритм пригоден для поиска оптимальных решений.

Решение было найдено программой [7] несколько раз с различными настройками ГА (см. следующий раздел). Полученные решения не полностью соответствуют предсказанному расположению, но набор ТК всегда равен оптимальному: 26 ТО, 17 ТД, 17 ТВ, 4 ТЗ. Пример решения в интерфейсе программы показан на рис. 2 и в табл. 2.



**Рис. 2. Оптимальное решение для условного объекта**

### 3. Проверка зависимости работы ГА от уровня влияния частей целевой функции и вероятности мутации

Проведем эксперименты для выяснения наилучших значений параметров задачи, при которых скорость поиска решения наибольшая.

В проводимых экспериментах доказывается, что неравные уровни влияния для двух частей целевой функции значительно ускоряют поиск решения по сравнению с равным (при большем влиянии первой части функции). Также проверяется скорость работы при различных соотношениях уровней влияния.

Вторым проверяемым параметром задачи является вероятность мутации. Чем выше эта вероятность, тем быстрее среди хромосом появляются решения, более близкие к оптимальному решению. Если же мутации появляются часто, то происходит слишком быстрое изменение хромосом. В этом случае решения, близкие к оптимальному, не успевают пройти процедуру кроссинговера и закрепить свои свойства в будущих поколениях. Это замедляет или делает полностью невозможным поиск оптимального решения.

Были проведены эксперименты с использованием описанного выше набора исходных данных. Цель:

определить скорость поиска решения при различных параметрах. Скорость определялась по числу итераций алгоритма, прошедших до появления хромосомы с оптимальным решением. Номер итерации соответствует номеру поколения хромосом. Дополнительно анализировали число произошедших мутаций.

В первой части экспериментов постоянным параметром была вероятность мутации, равная 0,3, и переменным параметром — уровни влияния частей целевой функции. Проверяли 10 вариантов отношения уровней влияния второй части целевой функции к первой, начиная от 0,5 (первая влияет в 2 раза сильнее второй) до 1 (уровни влияния одинаковы).

Во второй части экспериментов постоянный параметр — отношение уровней влияния — равен 0,75. Переменный параметр — вероятность мутации — изменялся в диапазоне от 0,05 до 0,5. С данной вероятностью каждая хромосома в поколении подвергается мутации одного гена.

В работе с генетическими алгоритмами слишком велик фактор случайности, и, чтобы уменьшить непредсказуемость результатов, эксперимент с каждым вариантом проводили трижды (табл. 3, 4).

Делаем следующий вывод: максимальная скорость поиска решения получена при значении от-

Таблица 3

Зависимость скорости поиска оптимального решения от отношения уровней влияния частей целевой функции

№	Отношение уровня влияния 2-й части целевой функции к 1-й	Вероятность мутации	Номер поколения с оптимальным решением	Число произошедших мутаций
1	0,5	0,3	8 955	79 208
2	0,5	0,3	13 180	116 532
3	0,5	0,3	12 536	110 620
4	0,6	0,3	4 017	37 003
5	0,6	0,3	5 896	53 940
6	0,6	0,3	6 692	59 474
7	0,7	0,3	2 913	26 785
8	0,7	0,3	3 462	31 596
9	0,7	0,3	4 852	43 596
10	0,75	0,3	2 632	24 783
11	0,75	0,3	3 314	29 518
12	0,75	0,3	2 586	23 024
13	0,8	0,3	3 493	31 969
14	0,8	0,3	2 184	20 868
15	0,8	0,3	2 898	26 991
16	0,85	0,3	3 623	32 275
17	0,85	0,3	2 454	22 563
18	0,85	0,3	3 450	31 619
19	0,9	0,3	5 360	47 210
20	0,9	0,3	3 292	30 222
21	0,9	0,3	2 793	25 514
22	0,95	0,3	5 654	47 670
23	0,95	0,3	7 344	63 477
24	0,95	0,3	7 956	69 446
25	0,97	0,3	9 480	83 922
26	0,97	0,3	11 706	103 320
27	0,97	0,3	11 505	101 910
28	1	0,3	16 663	145 513
29	1	0,3	14 534	126 493
30	1	0,3	13 015	115 193

Таблица 4

Зависимость скорости поиска оптимального решения от вероятности мутации

№	Отношение уровня влияния 2-й части целевой функции к 1-й	Вероятность мутации	Номер поколения с оптимальным решением	Число произошедших мутаций
1	0,75	0,05	10 401	16 047
2	0,75	0,05	10 942	16 646
3	0,75	0,05	7 911	12 097
4	0,75	0,1	5 183	15 214
5	0,75	0,1	6 276	19 417
6	0,75	0,1	5 187	15 794
7	0,75	0,15	4 402	20 192
8	0,75	0,15	3 851	17 985
9	0,75	0,15	4 391	20 196
10	0,75	0,2	4 270	25 332
11	0,75	0,2	4 314	25 659
12	0,75	0,2	6 217	36 262
13	0,75	0,25	2 447	19 360
14	0,75	0,25	3 888	29 448
15	0,75	0,25	3 219	25 601
16	0,75	0,3	2 632	24 783
17	0,75	0,3	3 281	30 287
18	0,75	0,3	2 345	22 017
19	0,75	0,35	2 677	28 779
20	0,75	0,35	3 046	31 642
21	0,75	0,35	3 132	33 249
22	0,75	0,4	2 919	39 981
23	0,75	0,4	5 999	80 892
24	0,75	0,4	4 411	53 757
25	0,75	0,45	5 975	71 911
26	0,75	0,45	3 940	54 151
27	0,75	0,45	9 002	108 710
28	0,75	0,5	19 940	298 566
29	0,75	0,5	17 502	261 759
30	0,75	0,5	9 377	141 192

ношения уровня влияния второй части целевой функции к первой в диапазоне  $0,75 \dots 0,9$ . То есть часть функции, отвечающая за соблюдение защищенности КЭ, должна влиять сильнее, чем часть функции, минимизирующая число ТК на объекте на  $10 \dots 25$  %. Наилучший вариант, когда влияние второй части функции составляет  $0,75$  от влияния первой. В пределах погрешности, вызываемой случайными процессами, вариант отношения  $0,9$  также показывает высокие результаты.

Как и предсказывалось, равные уровни влияния значительно замедляют работу алгоритма (практически в 3 раза). На рис. 3 показано начало поиска решений для модельного объекта с отношением уровней влияния частей целевой функции, равным единице. Графики отображают суммарное несоответствие хромосом частям целевой функции: нижний график — недостаток ТК для защиты КЭ, верхний — суммарное число ТК. Видно, что несоответствия хромосом частям функции становятся практически одинаковыми, что замедляет приближение к оптимальному решению, так как в оптимальном решении значение первой части функции должно быть нулевым, а от второй части — намного больше нуля [5]. Близкие к единице отношения  $0,95$  и  $0,97$  также требуют больше итераций алгоритма для поиска решения.

Отношения, меньшие  $0,75$ , показывают замедление работы: чем больше разница между уровнями влияния частей функции, тем дольше идет поиск решения. Это объясняется следующим: хромосомы, оптимальные по первой части целевой функции (отклонение равно нулю), при различных отклонениях по второй части целевой функции получают примерно равные значения вероятности попадания в родительский пул (так как влияние второй части функции слишком мало).

Вероятность попадания оптимального решения среди хромосом в родительский пул получается низкой, и поиск решения замедляется. На рис. 4 и 5 видно, что скорость сходимости по второй части целевой функции (верхний график) меньше при отношении  $0,5$ , чем при  $0,75$ .

Вывод по вероятности мутации: максимальная скорость работы алгоритма получена при значениях  $0,25 \dots 0,35$ . Меньшие вероятности показывают приблизительно линейную зависимость — чем ниже вероятность мутации, тем больше требуется итераций алгоритма для поиска решения. Это, очевидно, объясняется уменьшением вероятности появления хромосомы, которая будет ближе других к оптимальному решению. На графиках можно увидеть, что вероятность, равная  $0,05$ , вызывает замедление поиска решения задачи (рис. 6). Обнаруживается низкая сходимости по обеим частям целевой функции при сравнении с вероятностью мутации  $0,3$  при том же отношении уровней влияния частей целевой функции (см. рис. 4).

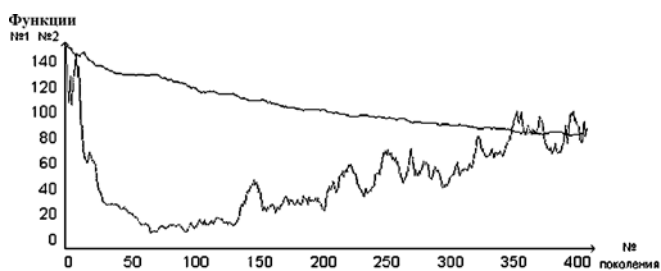


Рис. 3. Графики суммарного несоответствия частям целевой функции (вероятность мутации  $0,3$ , отношение уровней влияния  $1$ )

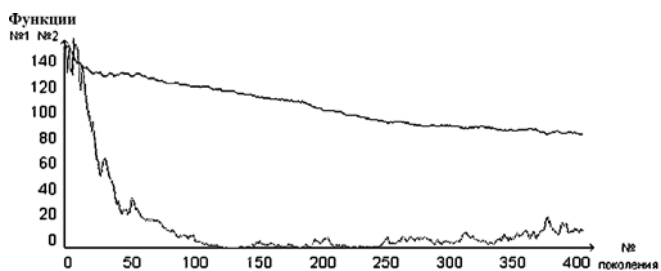


Рис. 4. Графики суммарного несоответствия частям целевой функции (вероятность мутации  $0,3$ , отношение уровней влияния  $0,75$ )

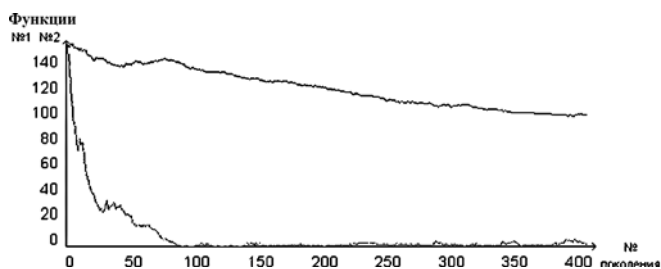


Рис. 5. Графики суммарного несоответствия частям целевой функции (вероятность мутации  $0,3$ , отношение уровней влияния  $0,5$ )

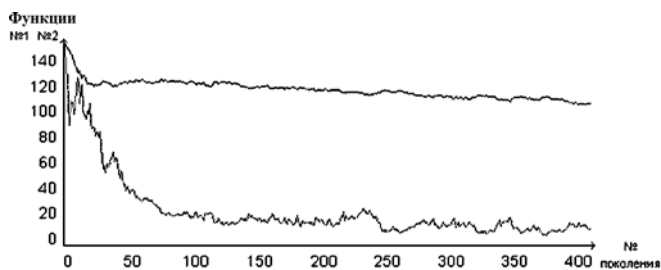


Рис. 6. Графики суммарного несоответствия частям целевой функции (вероятность мутации  $0,05$ , отношение уровней влияния  $0,75$ )

Слишком большое число мутаций разрушает хромосомы с эффективными решениями раньше, чем кроссинговер сохранит их свойства в следующих поколениях. Замедление поиска решения при вероятности  $0,5$  можно обнаружить в момент приближения свойств хромосом к оптимальному решению (рис. 7). Графики показывают в среднем большее суммарное несоответствие первой части целевой функции (нижний график) по сравнению с вероятностью мутации  $0,3$  при том же отношении уровней

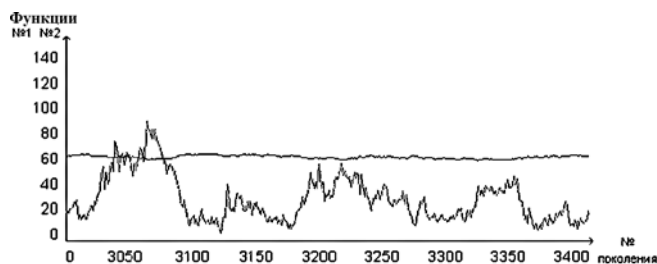


Рис. 7. Графики суммарного несоответствия частям целевой функции при приближении к оптимальному решению (вероятность мутации 0,5, отношение уровней влияния 0,75)

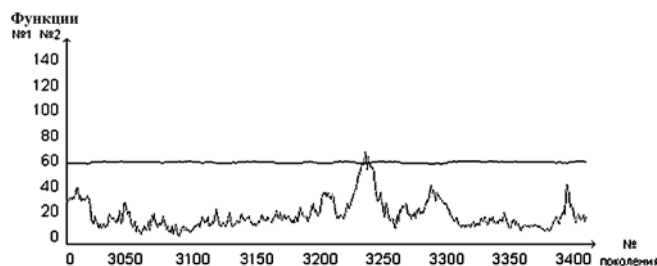


Рис. 8. Графики суммарного несоответствия частям целевой функции при приближении к оптимальному решению (вероятность мутации 0,3, отношение уровней влияния 0,75)

влияния частей целевой функции (рис. 8). При этом в начале работы ГА высокая вероятность мутации увеличивает скорость сходимости.

Делаем вывод, что вероятность мутации необходимо менять при приближении к оптимальному решению: чем дольше работает алгоритм, тем ниже должна быть вероятность.

### Заключение

Использование ГА для решения задачи выбора состава СФЗ критически важных объектов требует тщательной настройки параметров алгоритма, так как это влияет на эффективность работы и оптимальность полученных решений. Выбор наилучших значений параметров, подходящих для любых исходных данных, требует постоянного проведения

экспериментов с доказательством оптимальности полученных решений. Кроме того, практически невозможно определить, как правильно должны изменяться параметры во время работы алгоритма для различных исходных данных. Следовательно, необходимо включить в описанный алгоритм механизм адаптации для регулирования значений наиболее важных параметров.

Был проведен анализ процесса решения задачи выбора состава ИТСО с помощью стандартного ГА. В приведенном примере были определены два наиболее важных параметра: уровни влияния частей целевой функции и вероятность мутации. Эксперименты показали ощутимое воздействие значений этих параметров на работоспособность алгоритма.

Определена цель дальнейших исследований: необходимо провести модификацию программы для добавления механизма вычисления значений двух указанных параметров во время работы ГА. Полученный адаптивный генетический алгоритм будет решать задачу выбора состава ИТСО с высокой эффективностью.

### Список литературы

1. Магауенов Р. Г. Системы охранной сигнализации: основы теории и принципы построения. М.: Горячая линия — Телеком, 2004. 367 с.
2. Перминов А. Н. О новом подходе к решению задачи мониторинга критически важных объектов и (или) потенциально опасных объектов инфраструктуры Российской Федерации и опасных грузов: цели, задачи и направления развития // Право и безопасность. 2005. № 4. С. 41–46.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. М.: Горячая линия — Телеком, 2006. 452 с.
4. Холланд Дж. Генетические алгоритмы // В мире науки. 1992. № 9—10. С. 32.
5. Боровский А. С. Обобщенная модель системы физической защиты как объект автоматизированного проектирования // Вестник компьютерных и информационных технологий. 2014. № 10. С. 45–52.
6. Скурихин А. Генетические алгоритмы // Новости искусственного интеллекта. 1995. № 4. С. 6–17.
7. Свидетельство о государственной регистрации программы для ЭВМ № 2014615742 "GenalgSfz". Авторы: Тарасов Андрей Дмитриевич, Боровский Александр Сергеевич. Заявка № 2014613355. Дата поступления 15 апреля 2014 г. Зарегистрировано в Реестре программ для ЭВМ 02 июня 2014 г.

A. D. Tarasov, Senior Lecturer of "Automated systems of information processing and control" department, e-mail: adtarasov@mail.ru, State agrarian university of Orenburg

## Genetic Algorithm Effectiveness Used in Problem of Physical Defense System Designing

*Is considered the problem of decision-making for potentially-dangerous objects physical defense system structure. The problem stage with experts knowledge using — the conceptual designing, which including object defense analysis, practical recommendations for physical defense system creation, options of engineering defense units structure. It is observed problem solving process of engineering defense units structure designing with Standard Genetic Algorithm using. Method based on object of defense representation in graph form. The example of engineering defense units structure searching with the program is shown. Algorithm parameters with the greatest influence on decision process were found. Experiments are described which show: these parameters must be determined with Adaptive Genetic Algorithm using.*

**Keywords:** physical protection system, decision support system, standard genetic algorithm, adaptive genetic algorithm.

## References

1. **Magauenov R. G.** *Sistemy okhrannoi signalizatsii: osnovy teorii i principy postroeniya* (Guard Signalization Systems: Base of Theory and Build Principles), Moscow, Goryachaya liniya — Telekom, 2004, 367 p. (in Russian).
2. **Perminov A. N.** *Pravo i bezopasnost'*, 2005, no. 4, pp. 41—46 (in Russian).
3. **Rutkovskaya D., Pilin'skij M., Rutkovskij L.** *Neironnye seti, geneticheskie algoritmy i nechtotkie sistemi* (Neuron Networks, Genetic

Algorithms and Fuzzy Systems), Moscow, Goryachaya liniya — Telekom, 2006, 452 p. (in Russian).

4. **Holland G.** *V mire nauki*, 1992, no. 9—10, pp. 32 (in Russian).
5. **Borovskij A. S.** *Vestnik komp'uternikh i informatsionnykh tekhnologiy*, 2014, no. 10, pp. 45—52 (in Russian).
6. **Skurikhin A.** *Novosti iskusstvennogo intellekta*, 1995, no. 4, pp. 6—17 (in Russian).
7. **Tarasov A. D., Borovskij A. S.** *Svidetelstvo o gosudarstvennoy registratsii programmy dlya EVM № 2014615742 "GenalgSfz"*, 02.06.2014. (in Russian).

УДК 519.17, 519.8

**А. И. Николаев**, стажер-исследователь, ainikolaev@hse.ru,

Лаборатория алгоритмов и технологий анализа сетевых структур,

Национальный исследовательский университет Высшая школа экономики, Нижний Новгород

## Эффективный подход на основе машинного обучения к решению задачи о максимальной клике

*Представлен новый подход к решению задачи о максимальной клике. Предложенный подход состоит в том, что для данного графа с помощью машинного обучения выбирается наиболее быстрый алгоритм из нескольких алгоритмов, решающих задачу о максимальной клике. После чего выбранный алгоритм применяется для решения задачи о максимальной клике в этом графе. Вычислительные эксперименты на графах библиотеки DIMACS показывают, что представленный подход позволяет с высокой точностью выбрать наиболее быстрый алгоритм из нескольких рассматриваемых.*

**Ключевые слова:** задача о максимальной клике, точные алгоритмы, деревья классификации

### Введение

Задача о максимальной клике в простом неориентированном графе  $G = (V, E)$  является классической задачей теории графов. Клик (или полным подграфом) графа  $G$  называется такое подмножество его вершин, в котором любые две вершины соединены ребром. Клика, которая не содержится в клике большего размера, называется *максимальной по включению*. Задача о максимальной клике (ЗМК) состоит в том, чтобы для заданного графа  $G$  найти клику максимального размера.

ЗМК является  $NP$ -трудной задачей [4]. Данная задача имеет большое число приложений. В частности, эта задача возникает в теории кодирования [8], биоинформатике [2], анализе социальных сетей, анализе сетей фондовых рынков [1].

Для решения ЗМК в литературе предложено большое число как точных алгоритмов, так и приближенных (эвристик). Подробный обзор точных и приближенных алгоритмов для ЗМК может быть найден в работе [10]. В соответствии с результатами, опубликованными в этом обзоре, алгоритмы MaxCLQ [5] и MCS [9] являются наиболее эффективными современными алгоритмами для точного решения ЗМК.

Так как ЗМК относится к классу  $NP$ -трудных задач, то теоретический анализ точных алгоритмов

для решения ЗМК затруднителен. Вместо этого проводят эмпирические исследования их характеристик на тестовых графах. Например, для тестирования алгоритмов, решающих ЗМК, используют графы библиотеки DIMACS [12]. При этом во многих работах, посвященных данной задаче, делается вывод о том, что новый алгоритм "лучше" других, если он затрачивает на решение всех тестовых графов меньше времени, чем другие алгоритмы. Хотя тот факт, что одни алгоритмы оказываются лучше других алгоритмов на конкретных графах, зачастую никак не учитывается.

Идеальным решением данной проблемы является обращение к некоему "оракулу", который знает, сколько времени тратит каждый из известных алгоритмов на решение данного примера. После чего оракул выбирает алгоритм, который решает рассматриваемую задачу за минимальное время. К сожалению, совершенного оракула, который за короткое время выбирает наиболее быстрый алгоритм, для таких задач не существует, так как задача определения алгоритма является еще более сложной с вычислительной точки зрения. Поэтому на практике мы можем рассматривать только эвристические алгоритмы для предсказания наиболее быстрого алгоритма из предложенного множества алгоритмов.