

Intl. Trans. in Op. Res. 18 (2011) 425-448 DOI:  
10.1111/j.1475-3995.2011.00804.xМЕЖДУНАРОДНЫЕ  
СДЕЛКИ В СФЕРЕ  
ОПЕРАЦИОННОЙ  
ДЕЯТЕЛЬНОСТИ  
ИССЛЕДОВАНИЕ

## Точный алгоритм для минимизации вершинных охранников на галереях

Марсело К. Коуту, Педро Ж. де Резенде и Сид К. де Соуза

*Институт вычислительной техники, Государственный университет Кампинаса, Кампинас 13083-852, Бразилия**E-mail: coutomarclo@gmail.com [Couto]; rezende@ic.unicamp.br [de Rezende]; cid@ic.unicamp.br [de Souza]*

Получено 19 ноября 2010 г.; получено в пересмотренном виде 20 декабря 2010 г.; принято 21 декабря 2010 г.

### Аннотация

Мы рассматриваем задачу [art gallery problem (AGP)] минимизации числа вершинных охранников, необходимых для наблюдения за художественной галереей, граница которой представляет собой  $n$ -вершинный простой многоугольник. В этой статье мы обобщаем и расширяем наши исследования точных подходов к решению AGP. В предыдущих работах мы предложили и протестировали точный алгоритм для случая ортогональных многоугольников. В этом алгоритме дискретизация, аппроксимирующая многоугольник, используется для формулировки экземпляра задачи о покрытии множества, которая впоследствии решается оптимально. Либо набор защит, характеризующий это решение, решает исходный экземпляр AGP, и алгоритм останавливается, либо дискретизация уточняется и начинается новая итерация. Эта процедура всегда сходится к оптимальному решению AGP, причем количество итераций сильно зависит от способа дискретизации многоугольника. Несмотря на то, что наилучшее теоретическое ограничение на сходимость составляет  $O(n^3)$  итераций, наши эксперименты показывают, что оптимальное решение всегда находится за небольшое их число, даже для случайных многоугольников со многими сотнями вершин. Здесь мы расширяем семейство классов многоугольников, к которым применим алгоритм, за счет включения в него неортогональных многоугольников. Кроме того, мы предлагаем новые стратегии дискретизации, что приводит к дополнительному анализу компромисса между временем предварительной обработки и временем обработки и достижению, в случае новой стратегии *Convex Vertices*, наиболее эффективной на сегодняшний день общей производительности. Мы сообщаем об экспериментах с простыми и ортогональными многоугольниками с количеством вершин до 2500, которые показали, что во всех случаях для получения точного решения требуется не более 15 минут на стандартном настольном компьютере. В итоге мы более чем в два раза увеличили размер самых больших экземпляров, решенных оптимально, по сравнению с нашими предыдущими экспериментами, которые уже были в пять раз больше, чем те, о которых ранее сообщалось в литературе.

**Ключевые слова:** проблема картинной галереи; точный алгоритм; покрытие множества; видимость

### 1. Введение

Согласно Хонсбергеру (1976), в 1973 году Виктор Клее задал Васеку Чв<sup>а</sup>талу вопрос об определении минимального количества сторожей, которые должны охранять художественную галерею, имеющую форму

$n$ -стенный простой многоугольник. Доказательство Чэ-Таля (1975) о том, что охранники  $\lfloor n/3 \rfloor$  иногда необходимы и всегда пригодны для этой цели, стало первым результатом в том, что стало совершенно новой исследований. Свидетельством широкого спектра литературы, появившейся с тех пор, является классическая книга (O'Rourke, 1987), а также обзоры Шермера (1992) и Уррутиа (2000) и множество журнальных статей, цитируемых в них. Совсем недавно вышла книга Гхоша (2007), в которой представлен обширный набор тем по проблемам видимости, возникшим из смежных вопросов, в ней приведено более 300 ссылок, в основном за последние 15 лет.

Среди самых ранних и важных результатов можно назвать доказательство NP-полноты Ли и Лина (1986) для смежной задачи минимизации, которая оставалась открытой более 10 лет. А именно, задав плоский простой многоугольник  $P$ , определить размещение *минимального* числа стационарных охранников, покрывающих  $P$ .

В литературе рассмотрено множество вариантов этой задачи. Исследуемая нами формулировка ограничивает размещение охранников вершинами многоугольника, который представляет собой внешнюю границу данной художественной галереи. В данной работе мы будем называть эту формулировку задачей о картинной галерее (AGP). Кроме того, мы рассматриваем общие простые многоугольники, а также подкласс простых ортогональных многоугольников, которые особенно актуальны из-за того, что большинство реальных зданий и галерей имеют ортогональную форму (Urrutia, 2000).

В одном из первых крупных результатов, касающихся последней проблемы, Kahn et al. (1983) утверждают, что для покрытия ортогонального многоугольника с  $n$  вершинами иногда необходимо и всегда достаточно  $\lfloor n/4 \rfloor$  охранников. Позже Шухардт и Хекер (1995) доказали, что минимизация числа охранников в этом варианте также является NP-трудной задачей, решив вопрос, который оставался открытым почти десять лет (Sack and Toussaint, 1988).

В прошлом было предложено несколько алгоритмов размещения, таких как Edelsbrunner et al. (1984) и Sack and Toussaint (1988), которые решают проблему правильного размещения ровно  $\lfloor n/4 \rfloor$  охранников, покрывающих заданную ортогональную галерею.

С другой стороны, в недавно опубликованной работе Ghosh (2010), основанной на Ghosh (1987), представлен алгоритм аппроксимации с временем  $O(n^4)$  для простых многоугольников, дающий решения в пределах  $\log n$  от оптимального. Другие результаты аппроксимации включают в себя работу Eidenbenz (2002), в которой описываются алгоритмы, разработанные для нескольких вариантов задач охраны местности, и работу Amit et al. (2007), в которой анализируются эвристики с экспериментальными доказательствами хорошей производительности по покрываемой площади и количеству охранников.

Другой подход, рассмотренный в работах Erdem and Sclaroff (2006) и Тана-Сэ et al. (2006), заключается в моделировании проблемы как дискретной комбинаторной задачи и последующем решении соответствующей оптимизационной задачи. В первом случае внутренняя часть многоугольника дискретизируется с помощью фиксированной сетки, что дает алгоритм аппроксимации, а во втором - эмпирический анализ точного метода последовательных аппроксимаций, основанного на доминировании областей видимости.

Наконец, в работе Couto et al. (2007) мы представили точный алгоритм для оптимального решения ортогональной AGP. В этом алгоритме мы итеративно дискретизируем и моделируем задачу как классическую задачу о покрытии множества (SCP) (Wolsey, 1998). Помимо демонстрации осуществимости этого подхода, мы показали, что на практике количество итераций, необходимых для решения экземпляров до 200 вершин, очень мало, а полученный алгоритм оказался достаточно экономным.

Хотя было показано, что количество итераций, выполняемых точным алгоритмом, предложенным нами в Couto et al. (2007) и улучшенным в Couto et al. (2008), полиномиально ограничено, его практическая производительность намного лучше в зависимости от способа дискретизации многоугольника - см. также Couto et al.

(2009a) или смотреть Couto et al. (2009b). Это становится очевидным, когда мы замечаем, что на каждой итерации экземпляр SCP, NP-трудной задачи, должен быть решен оптимально, в нашем случае с помощью решателя целочисленного программирования (ЦП).

В данной работе мы развиваем все наши предыдущие исследования и проводим тщательное экспериментальное изучение компромисса между количеством и характером методов дискретизации и количеством итераций, а также анализируем практическую жизнеспособность каждого подхода. Более того, если в наших предыдущих работах (Couto et al., 2007, 2008) рассматривались только ортогональные многоугольники, то здесь мы показываем, что тот же подход хорошо работает и для общих простых многоугольников. Помимо работы с новыми классами многоугольников, в данной работе представлены две новые и более совершенные стратегии дискретизации, которые сравниваются с ранее изученными. Все наши тестовые данные доступны в Couto et al. (2009c) и включают множество примеров для каждого размера вершинного множества, для различных классов многоугольников с количеством вершин до 2500.

Новые экспериментальные результаты значительно превосходят те, о которых мы сообщали в работах Couto et al. (2007, 2008). Это связано с исследованием альтернативных стратегий дискретизации, которые позволяют нам решать сложные задачи, а также справляться с существенным увеличением размера полигонов по сравнению с предыдущими результатами, сохраняя при этом низкое время выполнения.

В следующем разделе мы описываем процесс моделирования AGP как SCP и основные идеи, необходимые для описания алгоритма, который приводится в разделе 3, а также его доказательство корректности и сложности. Раздел 4 посвящен описанию альтернативных стратегий дискретизации входного многоугольника. Далее, в разделе 5, мы рассказываем о настройке тестовой среды и представляем различные классы используемых экземпляров. Следуя рекомендациям Johnson (2002), McGeoch and Moret (1999), Sanders, (2002) и Moret (1986), в разделе 5.2 мы проводим обширный экспериментальный анализ алгоритма с учетом нескольких стратегий дискретизации и включаем оценку различных сравнительных показателей. В последнем разделе сделаны заключительные замечания.

## 2. Моделирование

В примере AGP нам дается простой многоугольник, ограничивающий художественную галерею, и предлагается определить минимальное количество и оптимальное расположение вершинных охранников, чтобы вся галерея находилась под наблюдением. Предполагается, что у вершинных охранников дальность видимости составляет 360°. Подход, используемый алгоритмом, описанным в разделе 3, преобразует непрерывную задачу AGP в дискретную, которая, в свою очередь, может быть легко смоделирована как экземпляр SCP. Фактически, на протяжении последних двух десятилетий это была единственная известная методика разработки эффективных алгоритмов аппроксимации для АГП (Ghosh, 2010; Eidenbenz, 2002; Amit et al., 2007). Прежде чем мы представим наш алгоритм и установим его корректность, давайте рассмотрим некоторые основные определения. *Художественную галерею с  $n$  стенами* можно рассматривать как плоскую область, граница которой состоит из простого многоугольника (без отверстий)  $P$ . Множество вершин  $P$  обозначается  $V$ , а вершина  $v \in V$  называется *рефлексивной*, если внутренний угол при  $v$  равен  $4180^\circ$ . Если не возникает путаницы, то *точкой в  $P$*  будем понимать точку либо внутри, либо на границе  $P$ .

Считается, что любая точка  $y$  в  $P$  видна из любой другой точки  $x$  в  $P$  тогда и только тогда, когда замкнутый отрезок, соединяющий  $x$  и  $y$ , не пересекает внешнюю сторону  $P$ . Множество  $\text{Vis}(v)$  всех точек в  $P$ , видимых из вершины  $v \in V$ , называется *областью видимости  $v$* . Легко видеть, что  $\text{Vis}(v)$  всегда является звездой

многоугольник. Описание границы  $Vis(v)$  может быть вычислено за линейное время с помощью алгоритма, предложенного Ли (1983) и расширенного в работах Джо и Симпсона (1985, 1987).

Множество точек  $S$  является *охранным множеством* для  $P$ , если для каждой точки  $p \in P$  существует точка  $s \in S$ , такая, что  $p$  видна из  $s$ . Следовательно, *вершинное охранное множество*  $G$  - это любое подмножество вершин, такое, что  $\bigcup_{g \in G} Vis(g) = P$ . Другими словами, вершинное охранное множество для  $P$  задает позиции стационарных охранников, которые могут наблюдать за всей картинной галереей границы  $P$ . Таким образом, AGP сводится к нахождению наименьшего подмножества  $G \subseteq V$ , которое является вершинным охранным множеством для  $P$ .

Читатель, знакомый с SCP (Wolsey, 1998), должен быть, уже понял, что проблема нахождения наименьшего набора вершинных защит для  $P$  может рассматриваться как специфический SCP. А именно, мы хотим найти наименьшее по кардинальности множество звездообразных многоугольников (областей видимости вершин  $P$ ), объединение которых покрывает  $P$ . Заметьте, что, строго говоря, это *непрерывная* SCP, поскольку во внутренней части  $P$  имеется бесконечно много точек, которые нужно покрыть. Однако можно дискретизировать задачу, сгенерировав конечное число представительных точек в  $P$ , чтобы формулировка стала управляемой. Ниже мы увидим, как этот подход приведет нас к приемлемому решению исходной задачи.

Кстати, отметим, что недавно в работе Baumgartner et al. (2010) была предпринята попытка на основе ЛП оценить границы для количества охранников, необходимых для наблюдения за многоугольной художественной галереей, хотя при этом не было представлено никаких гарантий завершения или точности.

Теперь мы опишем, как решения последовательно уточняемых дискретных экземпляров гарантированно сходятся к оптимальному решению непрерывной задачи. Для этого рассмотрим произвольную дискретизацию  $P$  на конечное множество точек  $D(P)$ . Будем обозначать  $I(P, D(P))$  экземпляр дискретизированной AGP, полученный таким образом. ИС-формулировка соответствующего экземпляра SCP приведена ниже

$$\begin{aligned} z = \min \quad & \sum_{j \in V} x_j \\ \text{s.t.} \quad & \sum_{j \in V} a_{ij} x_j \geq 1, \text{ для всех } p_i \in D(P) \\ & x_j \in \{0, 1\}, \text{ для всех } j \in V, \end{aligned} \quad (1)$$

где двоичная переменная  $x_j$  устанавливается в 1 тогда и только тогда, когда вершина  $j$  из  $P$  выбрана в защитное множество. Более того, если точка  $p_i \in D(P)$  и вершина  $j$  из  $P$ , то  $a_{ij}$  - это двоичное значение, которое равно 1 тогда и только тогда, когда  $p_i \in Vis(j)$ .

Учитывая выполнимое решение  $x$  вышеприведенной ИС, пусть  $Z(x) = \{j \in V \mid x_j = 1\}$ . Ограничение (1) гласит, что каждая точка  $p_i \in D(P)$  видна хотя бы из одной выбранной позиции охраны в решении, а объективная функция минимизирует кардинальность  $z$  в  $Z(x)$ . Очевидно, что поскольку множество  $D(P)$  конечное, может случиться так, что  $Z(x)$  не образует вершинного защитного множества для  $P$ . В этом случае мы должны выбрать новую точку внутри каждой непокрытой области и включить эти точки в  $D(P)$ . Затем создается новый экземпляр SCP и решается соответствующая ИС, что приводит к итерационной процедуре. В конце раздела 3.2 мы установим сходимость этого процесса.

Фактическое количество необходимых итераций зависит от того, сколько непокрытых областей может быть последовательно сгенерировано. Поскольку стоимость каждой итерации связана с количеством ограничений в (1), возникает интересный компромисс, который приводит к попыткам выбора нескольких схем дискретизации. С другой, любой метод умного выбора начального

точки дискретизации будут иметь соответствующую стоимость во времени предварительной обработки, что открывает еще один интригующий вопрос обмена временем. Именно эти вопросы мы рассматриваем в разделе 4, где мы рассматриваем несколько возможных схем дискретизации, которые приводят к различным анализам производительности, обсуждаемым в разделе 5.

### 3. Описание алгоритма и доказательство корректности

Алгоритм состоит из двух этапов: *этапа предварительной обработки*, на котором строится начальная дискретизация, описанная в разделе 1, и ставится задача ИС, и *этапа решения*, на котором алгоритм выполняет итерации, как описано выше, решая задачи SCP для текущей дискретизации, пока не останется ни одной непокрытой области.

Как упоминалось ранее, множество решений  $Z(x)$  для дискретизированной формулировки в разделе 2 не всегда является защитным множеством для  $P$ , поскольку внутри  $P$  могут быть области, которые не видны ни из одной защиты в  $Z(x)$ .

Чтобы формализовать наши последующие рассуждения, мы начнем со следующего определения.

**Определение 1.** Пусть  $I(P, D(P))$  - экземпляр дискретизированной АГП с многоугольником  $P$  в качестве границы галереи и  $D(P)$  - дискретизацией  $P$ . Решение  $Z(x)$  этого экземпляра называется жизнеспособным, если  $Z(x)$  является защитным множеством для  $P$ , т.е.,

$$\bigcap_{g \in Z(x)} \text{Vis}(g) = P.$$

Любой точный метод для исходной АГП, решающий ее дискретизированную версию, должен учитывать тот факт, что решение  $I(P, D(P))$  может быть не обязательно жизнеспособным. Как мы увидим, наш алгоритм преодолевает эту трудность и всегда выдает жизнеспособное решение, последовательно уточняя заданную дискретизацию всякий раз, когда обнаруживает, что текущее решение не жизнеспособно. Более того, следующая теорема устанавливает, что решение, полученное с помощью этого итерационного процесса, также является минимальным.

**Теорема 1.** Пусть  $Z$  - решение экземпляра  $I(P, D(P))$  дискретизированной АГП. Если  $Z$  жизнеспособно, то  $Z$  оптимально.

**Доказательство.** Из того, что  $Z$  является решением задачи минимизации  $I(P, D(P))$ , следует, что  $Z$  оптимально в качестве вершинного защитного покрытия для множества  $D(P)$  точек, дискретизирующих многоугольник  $P$ , т.е.  $|Z|$  минимально среди кардинальностей всех вершинных защитных покрытий  $D(P)$ .

Теперь пусть  $Z^*$  - оптимальное вершинное защитное множество для  $P$  и пусть  $z^* \in Z^*$ . Поскольку  $Z^*$  также является вершинным защитным множеством для  $D(P)$ , мы должны иметь  $z^* \in Z$ . С другой стороны, поскольку  $Z$  жизнеспособно, из этого следует, что  $z \in Z^*$ . &

Теорема 1 устанавливает, что когда алгоритм находит решение дискретизированной формулировки, которое является жизнеспособным, это решение также является минимальным вершинным защитным покрытием для  $P$ , то есть является защитным множеством для  $P$ . Теперь мы можем подробно описать алгоритм, впервые предложенный нами в Couto et al. (2007). На *этапе предварительной обработки* выполняются три процедуры: первая вычисляет полигоны видимости для точек в  $V$ , вторая - начальную дискретизацию  $D(P)$  и третья - построение соответствующей IP-модели. На *этапе решения* дискретизированная задача

Последовательно решаются и уточняются, пока не будет найдено жизнеспособное (и оптимальное) решение.

### 3.1. Этап предварительной обработки

Основные этапы предварительной обработки приведены в Алгоритме 3.1.

Чтобы собрать формулировку, описанную в разделе 2, мы начинаем с построения начальной дискретизации  $D(P)$  многоугольника (шаг 1). В разделе 4 мы описываем альтернативные стратегии дискретизации и их влияние на эффективность алгоритма.

После построения дискретизации мы вычисляем, какие из ее точек находятся внутри области видимости каждой вершины в  $V$ , и включаем эти ограничения в формулировку SCP.

#### Алгоритм 3.1 Этап предварительной обработки

```

1:       $D(P) \leftarrow$  выбранная начальная дискретизация
 $P$ ; 2:      для каждого  $j \in V$  сделать
3:          Вычислите  $Vis(j)$ ;
4:          для каждой точки дискретизации  $p_i \in D(P)$  сделать
5:               $a_{ij} \leftarrow \text{Boolean}(p_i \in Vis(j))$ ; 6:
          end for
7:      конец для
  
```

Общая сложность шага 3 составляет  $O(n^2)$  (Joe and Simpson, 1985), и, если предположить, что  $m \leq |D(P)|$ , полная сложность шага 5 составляет  $O(m \log n)$ , поскольку определение местоположения каждой из  $m$  точек  $D(P)$  в звездообразном видимом  $n$ -полигоне может быть выполнено за время  $O(\log n)$ . Таким образом, общая сложность этапа предварительной обработки доминирует над сложностью шага 5, когда  $m \in O(n/\log n)$ , и над сложностью шага 3, в противном случае.

Результатом фазы предварительной обработки является IP-формулировка для SCP, которая после решения генерирует решение  $Z$ , которое, хотя и не обязательно является защитным множеством для  $P$ , всегда покрывает все точки в  $D(P)$ .

### 3.2. Фаза решения

На втором этапе алгоритма, начиная с формулировки ИС, полученной на этапе предобработки, мы решаем дискретизированный пример с последующим итеративным уточнением дискретизации до тех пор, пока решение не станет жизнеспособным. Это уточнение достигается путем создания еще одной точки в дискретизации для каждой непокрытой области (например, ее центроида) и добавления соответствующих ограничений к текущему SCP. Эти дополнительные точки улучшают формулировку и приводят к решению, близкому к жизнеспособному. Алгоритм 3.2 описывает шаги, выполняемые на этапе решения.

#### Алгоритм 3.2 Фаза решения 1:

```

      повторить
2:       $Z \leftarrow$  решение  $I(P, D(P))$ ;
3:      для каждого непокрытого региона  $R$  сделайте
4:           $c \leftarrow$  центроид  $R$ ;
5:           $D(P) \leftarrow D(P) \cup \{c\}$ ;
  
```

© 2011 Авторы.

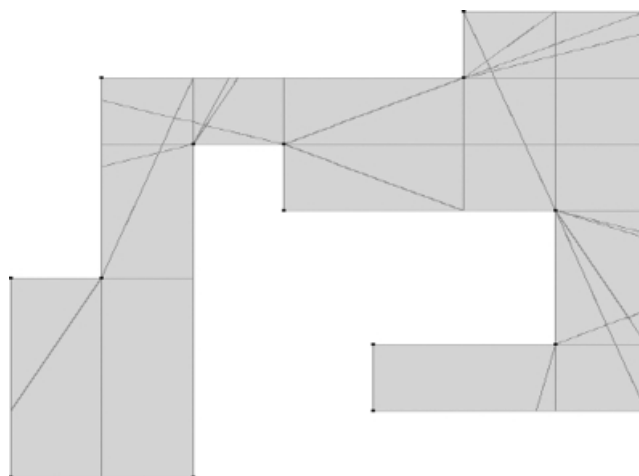


Рис. 1. Расположение видимости и атомарные полигоны видимости.

- 6: **Р** Добавьте в набор ограничений (1) новую строку  $r$ , соответствующую точке  $c$ :  

$$j \in V a_{rj} x_j \leq 1, \text{ где } a_{rj} \leftarrow \text{Boolean}(c \in \text{AVis}(j)), \delta j \in V;$$
- 7: конец для
- 8: пока  $Z$  не станет жизнеспособным

Остается доказать, что алгоритм 3.2 останавливается, поскольку из Теоремы 1 следует, что алгоритм точен и полученное решение действительно является защитным множеством для  $P$ .

**Определение 2.** Рассмотрим множество всех областей видимости вершин в  $V$ , объединение которых, очевидно, покрывает  $P$ . Ребра этих областей видимости вызывают расположение отрезков прямых внутри  $P$ , грани которых мы называем многоугольниками атомарной видимости (AVP) (см. рис. 1).

Чтобы определить наихудший случай для количества итераций, выполняемых алгоритмом, сначала заметим, что из определения AVP следует, что если центроид (или, тем более, любая точка внутри) AVP  $V$  виден из вершинной защиты, то вся область  $V$  также должна видна. Поскольку область видимости любой вершины может иметь не более  $O(n)$  ребер, индуцированное расположение генерируется из  $O(n^2)$  отрезков прямых и имеет общую сложность не более  $O(n^4)$  граней (или AVP). Заметим, что на шаге 3 любая непокрытая область (свидетельствующая о том, что  $Z$  не жизнеспособна) обязательно простым многоугольником, образованным объединением соседних AVP. Таким образом, верхняя граница на максимальное количество итераций алгоритма равна  $O(n^4)$ , и это подтверждает сходимость. Более того, из результата Bose et al. (2002) можно сделать вывод, что  $Y(n^3)$  является жестким ограничением на количество AVP, что улучшает приведенный выше результат для наихудшего случая. Однако на практике это все равно чрезвычайно завышены и должны рассматриваться исключительно как доказательство сходимости итерационного метода.

#### 4. Стратегии дискретизации

Как было показано в предыдущем разделе, сходимость алгоритма следует из верхней границы на количество непокрытых областей. Тем не менее, поскольку на каждой итерации решается экземпляр NP-

В идеале выбранная стратегия дискретизации должна достаточно легкой, чтобы экземпляры SCP, которые можно быстро решить, минимизируя количество итераций, необходимых для достижения оптимального решения.

Таким образом, существует компромисс между скоростью и точностью, который необходимо учитывать при разработке хорошей стратегии дискретизации. Хотя использование сложных геометрических свойств для построения более точной дискретизации может уменьшить количество итераций, выполняемых алгоритмом, соответствующие затраты на предварительную обработку могут перевесить преимущества.

В следующих разделах подробно рассматриваются несколько альтернативных вариантов дискретизации полигона и обсуждаются теоретические преимущества и возможные недостатки каждого из них.

#### 4.1. Одиночная вершина

Самая простая стратегия, которую можно рассмотреть, - это начать дискретизацию с одной вершины многоугольника

*P*. На первый , стратегия с *одной вершиной* может показаться слабой, поскольку одна точка в *P* не несет никакой геометрической информации, а решение дискретизированной AGP, связанное с этой простой дискретизацией, как ожидается, оставит несколько непокрытых областей, что приведет к множеству итераций алгоритма.

Однако размер экземпляров SCP, которые генерирует стратегия с *одной вершиной*, очень мал, и они не требуют затрат на предварительную обработку. Поэтому все же стоит определить, оправдывает ли себя эта стратегия.

#### 4.2. Все вершины

Разумный подход, позволяющий сократить количество итераций при использовании стратегии с *одной вершиной*, заключается в том, чтобы начать с большей дискретизации, точки которой адекватно распределены по *P*. Однако, чтобы сохранить преимущества предыдущей стратегии, количество точек в такой дискретизации должно быть небольшим и легко вычисляемым. Стратегия "*все вершины*" - это попытка достичь этой цели. Мы рассматриваем все еще разреженный случай, когда начальная дискретизация содержит все вершины многоугольника (см. рис. 2). Видно, что эта стратегия учитывает тот факт, что вершины многоугольника должны отражать достаточно геометрической информации, чтобы не возникало непокрытых областей вблизи выпуклых вершин. Более того, эксперименты показывают, что стратегия "*все вершины*" генерирует меньше непокрытых областей, чем стратегия "*одна вершина*", и в этом случае добавляются более значимые ограничения, что приводит к лучшим решениям на каждой итерации.

#### 4.3. Выпуклые вершины

Выпуклые вершины - явно более полезные точки дискретизации, чем рефлекторные, поскольку они легче охраняются, чем те. Поэтому, если какие-то вершины могут быть лишними при сборе информации о видимости, естественно предположить, что наиболее лишними являются рефлекторные вершины.

Эти наблюдения заставили нас рассмотреть стратегию *выпуклых вершин*, которая начинается дискретизации *P*, состоящей исключительно из выпуклых вершин (см. рис. 3). При этом мы еще больше уменьшаем размер дискретизации, сохраняя большую часть комбинаторных связей видимости ценой незначительного увеличения стоимости предварительной обработки.



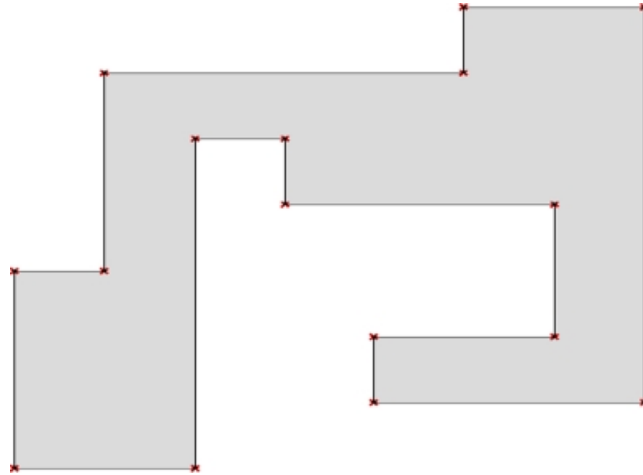


Рис. 2. Пример начальной дискретизации, используемой в стратегии "все вершины".

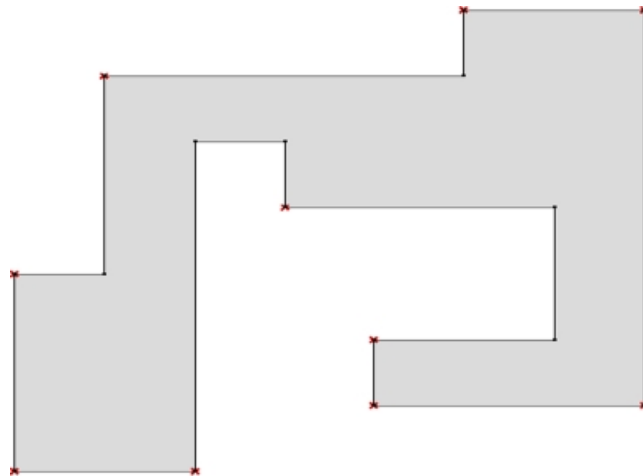


Рис. 3. Пример начальной дискретизации, используемой в стратегии выпуклых вершин.

Хотя можно было бы ожидать, что такое сокращение приведет к увеличению числа итераций, мы не обнаружили таких последствий. Кроме того, эта стратегия сохраняет все те же приятные особенности двух предыдущих вариантов, а именно: недорогой этап предварительной обработки и экземпляры SCP с малым числом ограничений на каждой итерации.

#### 4.4. AVPs

Рассмотренные до сих пор стратегии направлены на то, чтобы сохранить небольшое количество ограничений в начальном экземпляре SCP, пытаясь при этом сократить число итераций. Однако можно разработать стратегию, позволяющую сократить число итераций до одной.

© 2011 Авторы.

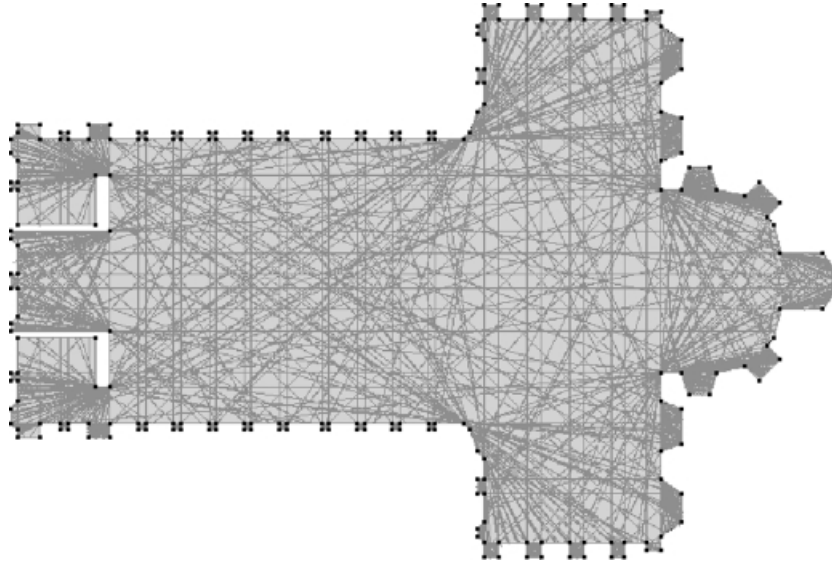


Рис. 4. Размерная галерея и ее расположение видимости, показывающее все атомарные полигоны видимости.

Для этого необходимо определить набор областей, которые, будучи покрытыми, гарантируют, что вся галерея находится под охраной. Более того, эти области должны обладать свойством, согласно которому, как только одна из их точек становится видимой из вершинной защиты, вся область оказывается под наблюдением этой защиты. Когда эти регионы определены, можно построить дискретизацию, выбрав по одной точке в каждом из них.

Как было показано в предыдущем разделе, AVPs из  $P$  удовлетворяют обоим свойствам, указанным в предыдущем параграфе. Поэтому начальная дискретизация, содержащая центроиды всех AVP, приводит к экземпляру SCP, который после решения дает оптимальный набор вершинных защит для  $P$ . Однако, поскольку AVP может быть  $O(n^3)$ , количество ограничений в модели SCP также может быть  $O(n^3)$ .

Хотя это доказывает, что мы можем решить задачу за одну итерацию алгоритма, построение дискретизации с центроидами всех AVP сложной галереи может привести к созданию огромного экземпляра SCP (см. рис. 4). Тем не менее, как показано далее, не все AVP должны быть представлены в наборе ограничений, чтобы гарантировать одну итерацию, что делает дискретизацию по всем AVP бессмысленной.

#### 4.4.1. Теневые AVP

Как мы уже видели, решение экземпляра SCP с центроидами всех AVP очень дорого. Однако мы можем значительно уменьшить количество дискретизированных точек и при этом гарантировать, что алгоритм найдет минимальное количество охранников, необходимых для покрытия  $P$  после первой итерации. Для этого мы вводим понятие *теневого AVP*.

Изначально мы говорим, что отрезок прямой является *ребром видимости* для  $P$ , если существует вершина  $v \in P$  такая, что этот отрезок является ребром  $\text{Vis}(v)$ . Более того, считается, что ребро видимости  $e$ , исходящее из вершины  $v$ , является *собственным* для  $v$  тогда и только тогда, когда  $e$  не содержится ни в одном ребре  $P$ .

Обратите внимание, что, поскольку AVP - это грань в схеме, порожденной ребрами видимости, ребра AVP являются либо частями ребер  $P$ , либо частями соответствующих ребер видимости вершин  $P$ .

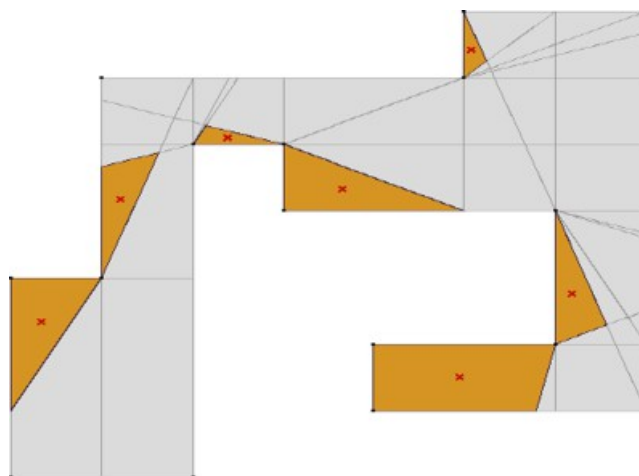


Рис. 5. Пример начальной дискретизации, используемой в стратегии *атомарных полигонов видимости тени*.

Мы говорим, что AVP  $S$  является *теневой AVP*, если существует подмножество  $U$  вершин  $V$  такое, что  $S$  не видна ни из одной вершины в  $U$  и единственные подходящие ребра видимости, порождающие  $S$ , исходят из вершин в  $U$ .

Стратегия дискретизации *теневого AVP* заключается в взятии центроидов каждой теневого AVP (см. рис. 5).

Теперь мы установим фундаментальную связь между оптимальными решениями дискретизированной AGP с *теневого* стратегией AVPs и решениями исходной AGP.

**Теорема 2.** Пусть  $I(P, D(P))$  - экземпляр дискретизированного AGP для многоугольника  $P$ , где  $D(P)$  множество центроидов теневого AVP  $P$ . Тогда  $G$  является вершинным защитным множеством для  $D(P)$  тогда и только тогда, когда  $G$  является вершинным защитным множеством для  $P$ .

*Доказательство.* Часть, связанная с необходимостью, тривиальна, так как  $D(P) \subseteq P$ , поэтому мы сосредоточимся на доказательстве достаточности. Предположим, что  $G \subseteq V$  является вершинным защитным множеством для  $D(P)$ , но не для  $P$ . Таким образом, существуют области  $P$ , не покрытые ни одной из вершин

$G$ . Пусть  $R$  - максимальная связная область, не покрытая

$G$ . Заметим, что  $R$  - это объединение (расходящихся) AVP.

Чтобы доказать, что хотя бы одна из этих AVP имеет тип shadow, заметим, что вся область  $R$  не видна ни одной вершине в  $G$ , чьи ребра видимости порождают  $R$ . Если  $R$  - AVP, то она по определению является shadow AVP. В противном случае существует вершина  $v_i \in V$ , имеющая ребро видимости  $e_{v_i}$ , которое пересекает и разделяет  $R$  на две другие области. Одна из этих областей соответствует стороне  $e_{v_i}$ , видимой из  $v_i$ , а противоположная - нет. Следовательно, по индуктивному аргументу, при последовательном разбиении  $R$  по крайней мере один теневого AVP обязательно будет содержаться в  $R$  и, следовательно, не будет покрыт.

Это противоречит гипотезе, поскольку  $G$  является защитным множеством для  $D(P)$ , которое состоит из центроидов всех теневого AVP. &

Из Теоремы 2 видно, что при использовании *теневого* стратегии AVPs алгоритм сходится за одну итерацию. Кроме того, когда мы ограничиваемся теновыми AVP, размер дискретизации значительно уменьшается по сравнению со стратегией AVP. Даже для сложных галерей (см. рис. 4 и 6) это уменьшение может быть достаточно большим, чтобы сделать алгоритм практичным.

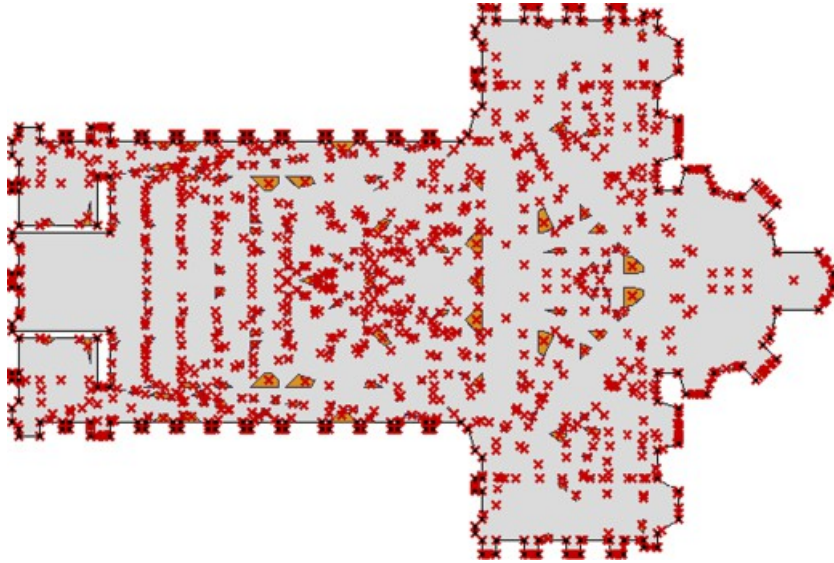


Рис. 6. Галерея большого размера и точки дискретизации, используемые в стратегии *атомарных полигонов видимости тени*.

Хотя *теневая* стратегия *AVPs* требует всего одной итерации алгоритма для нахождения оптимального решения, позже мы увидим, что время, затрачиваемое на фазу предварительной обработки, может стать узким местом алгоритма. Этот вопрос будет рассмотрен в разделе 5.2.

#### 4.5. Другие стратегии

Для полноты картины следует упомянуть, что в наших предыдущих работах рассматривались и другие стратегии, например, основанные на регулярной (Couto et al., 2007) и индуцированной (Couto et al., 2008) дискретизации сетки. Эксперименты показали, что ни одна из них не выдерживает конкуренции с обсуждаемыми в данном разделе стратегиями, которые являются гораздо более эффективными.

### 5. Вычислительные эксперименты

Теперь мы обсудим экспериментальные исследования, которые мы провели для оценки алгоритма, предложенного в разделе 3. В частности, мы анализируем поведение алгоритма отношению к различным стратегиям дискретизации, рассмотренным в предыдущем разделе.

Все наши программы были написаны на C11 и скомпилированы с помощью GNU g11 4.2, поверх CGAL.

3.2.1 (Computational Geometry Algorithms Library (CGAL), <http://www.cgal.org>). Был реализован алгоритм видимости Джо и Симпсона (1985), а для расчета моделей ИС, соответствующих экземплярам SCP, использовался Xpress v18.10.04 (XpressFOptimizer, <http://www.dashoptimization.com>).

Что касается аппаратного обеспечения, то мы использовали настольный ПК с Pentium IV с частотой 3,4 ГГц и 3 ГБ оперативной памяти под управлением GNU/Linux 2.6.24. Заметим, что во время тестирования на машине не было разрешено выполнение других процессов. Кроме того, в каждом случае алгоритм останавливался либо из-за того, что было найдено оптимальное решение, либо из-за того, что у программы заканчивалась память.

© 2011 Авторы.

### 5.1. Экземпляры

Чтобы считаться реалистичным методом решения AGP, алгоритм должен быть способен обрабатывать большое количество примеров с отличительными характеристиками. Таким образом, чтобы проверить устойчивость нашего алгоритма, мы разработали четыре класса примеров (см. рис. 7). Каждый из них отражает особые геометрические свойства, которые либо встречаются в реальных картинных галереях, либо представляют собой экстремальные ситуации, необходимые для отработки некоторых характеристик алгоритма. Набор всех экземпляров, использованных в экспериментах, о которых сообщается здесь, а также некоторые другие, можно загрузить из Couto et al. (2009c).

В первых двух классах экземпляров картинная галерея представлена в виде ортогонального или простого многоугольника, соответственно. Считается, что первые являются хорошими представителями многих реальных зданий художественных галерей. Последние два класса соответствуют многоугольникам, собранным из замкнутой версии фрактальной кривой фон Коха, см. Falconer (1990). Сгенерированные таким образом экземпляры имеют тенденцию иметь небольшие выступы на границе многоугольника, которые создают крошечные области, видимые только небольшим количеством вершин. Предположительно, эти примеры сложнее решить, чем аналогичные по размеру примеры двух первых классов. В качестве примера рассмотрим два экземпляра на рис. 8. Оба многоугольника имеют по 100 вершин, но схема видимости случайного ортогонального экземпляра имеет 2216 ребер и 1085 AVP (с 99 теньевыми AVP), в то время как схема, соответствующая случайному многоугольнику фон Коха (RvK), имеет 8794 ребра и 4420 AVP (с 264 теньевыми AVP).

Более подробная информация о том, как генерировать тестовые полигоны каждого класса, приведена ниже.

- (1) *Случайный ортогонал*: Случайный ортогональный экземпляр состоит из случайного ортогонального многоугольника с  $n$  вершинами, размещенного на сетке  $2^n \times 2^n$  единичных квадратов. Многоугольник генерируется без коллинеарных краев в соответствии с методом, описанным в Tóth и Bajuelos (2004).

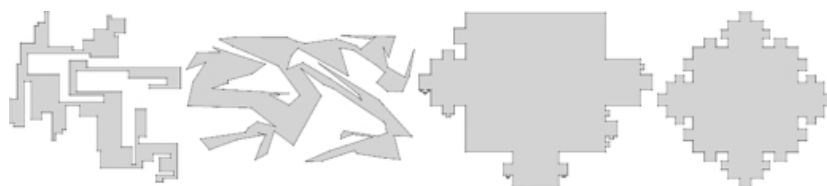


Рис. 7. Образцы многоугольников со 100 вершинами: случайный ортогональный, случайный простой, случайный фон Коха и полный фон Коха.

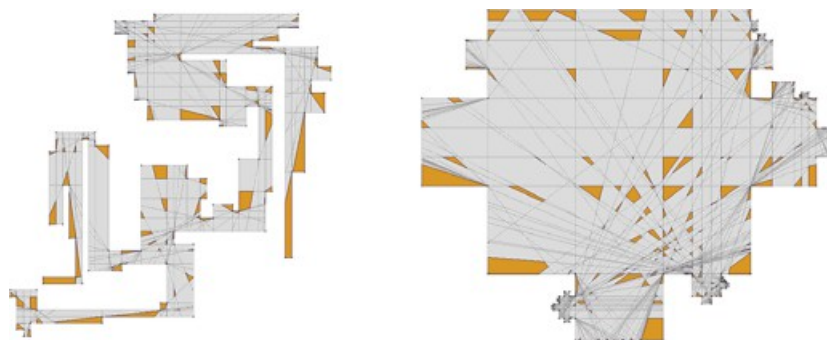


Рис. 8. Примеры случайных ортогональных и случайных многоугольников фон Коха одинакового размера, но разной сложности.



Рис. 9. Уровни многоугольников фон Коха.

- (2) *Случайный простой*: Каждый случайный простой экземпляр представляет собой случайный простой многоугольник, специальной процедурой, доступной в CGAL (n.d.). По сути, эта процедура начинает с равномерного распределения вершин многоугольника в заданном прямоугольнике и применяет метод устранения самопересечений с помощью 2-опт ходов.
- (3) *Полная кривая фон Коха (CvK)*: Здесь многоугольники генерируются на основе модифицированной версии кривой фон Коха, которая является фракталом с размерностью Хаусдорфа 1,34. Экземпляр создается, начиная с квадрата и рекурсивно заменяя каждое ребро на пять других ребер, как показано на рис. 9, где  $ar = st = ub$  и  $sr = tu = \frac{3}{4}ar$ . Воспользуемся рис. 9, чтобы ввести некоторые обозначения, необходимые для описания последнего класса экземпляров. Мы говорим, что ребро текущего многоугольника, которое остается над границей исходного квадрата, находится на уровне 0. Когда операция замены, показанная на рисунке, применяется к ребру  $e$  на уровне  $k$ , новые ребра, не коллинеарные с  $e$ , считаются находящимися на уровне  $k+1$ .
- (4) *RvK*: Экземпляр этого последнего класса строится следующим образом. Мы начинаем с квадрата и итеративно применяем операцию замены (из рис. 9) к некоторым ребрам, пока количество вершин многоугольника не достигнет *априорно* фиксированного предела. На каждой итерации мы случайным образом выбираем ребро, уровень которого меньше заданного параметра  $L$ , и случайным образом решаем, заменять его или нет.

Важно отметить, что для класса RvK экземпляры до 1000 вершин генерировались при  $L$ , равном 4. За пределами этого размера, когда число вершин приближается к числу вершин CvK-полигона уровня 4 (2500),  $L$  устанавливалось равным 5. Это, вероятно, объясняет разрыв, наблюдаемый в районе 1000 вершин на некоторых графиках, показанных в разделе 5.2 (см. рис. 13), где представлены результаты, полученные нашим алгоритмом для этого класса многоугольников.

Случайные экземпляры были сгенерированы для количества вершин  $n$  в диапазонах: [20, 100] с размером шага 20; (100, 1000] с размером шага 100 и (1000, 2500] с размером шага 250. Аналогичные размеры были выбраны для класса RvK. Наконец, класс CvK содержит, по конструкции, только четыре экземпляра с 20, 100, 500 и 2500 вершинами.

Чтобы придать нашим выводам статистическую значимость, необходимо определить размер выборки, то есть количество экземпляров, сгенерированных для каждого значения  $n$  для классов random orthogonal, random simple и RvK. Для этого мы проанализировали дисперсию результатов, полученных нашим алгоритмом, при изменении размера выборки  $s$ . Мы заметили, что при  $s \geq 30$  дисперсия практически не изменилась, и поэтому решили генерировать 30 экземпляров для каждого значения  $n$ .

Таким образом, в общей сложности наш набор данных состоит из 1804 экземпляров, имеющих от 20 до 2500 вершин каждый. Стоит отметить, что наши самые большие экземпляры более чем в два раза превышают размеры самых больших экземпляров, оптимальные решения которых встречаются в литературе.

Для полноты картины отметим, что в наших предыдущих работах (Couto et al., 2007, 2008) рассматривались и другие классы экземпляров, в том числе многоугольники FAT, представленные в работе Tomàs et al. (2006). Эти классы не были включены в представленное здесь исследование, поскольку они гораздо менее сложны, чем те, которые рассматриваются в экспериментах, представленных в данной работе. Например, FAT-полигоны любого размера всегда имеют аналитическое оптимальное решение, состоящее только из двух охранников.

В следующем разделе представлен обширный экспериментальный анализ алгоритма с учетом нескольких стратегий дискретизации.

## 5.2. Результаты

Теперь мы обсудим экспериментальную оценку различных стратегий дискретизации, описанных в разделе 4. Все значения, приведенные в этом разделе, являются средними результатами для 30 экземпляров каждого размера или 30 запусков одного экземпляра в случае класса CvK, поскольку для этого класса существует единственный экземпляр каждого размера.

Напомним, что размер дискретизации определяет количество ограничений в экземпляре SCP, решаемом на втором шаге алгоритма 3.2. Начнем с анализа зависимости между типами дискретизации и временем, затрачиваемым предложенным алгоритмом. Многоугольники CvK особенно хорошо подходят для этого анализа, поскольку они иллюстрируют две экстремальные ситуации. В стратегиях *с одной вершиной*, *выпуклыми вершинами* и *всеми вершинами* начальные дискретизации соответствуют очень разреженным сеткам, размеры которых увеличиваются лишь на небольшую величину в течение итераций. С другой, для *теневых АВП* одна итерация алгоритма приносит успех, но за счет построения чрезвычайно плотной сетки. В таблице 1 приведены эти результаты.

Обратите внимание, что стратегии *с одной вершиной*, *выпуклыми вершинами* и *всеми вершинами* действительно дают небольшие дискретизации, размер которых линейно увеличивается с ростом числа вершин  $P$ . Что касается стратегии *теневых АВП*, то размер дискретизации растет очень быстро, настолько, что мы не смогли решить самый большой экземпляр CvK из-за нехватки памяти. Большие сетки увеличивают количество ограничений в формулировке ИС, что значительно увеличивает время, необходимое для оптимального решения экземпляра SCP. Меньше всего времени тратится на стратегию *с выпуклыми вершинами*, затем следуют стратегии *с одной вершиной* и *со всеми вершинами*. Как видно, относительный порядок между этими стратегиями остается неизменным по отношению к размерам конечных дискретизаций.

С другой, на рис. 10 показано количество дискретизированных точек, требуемых каждой стратегией для достижения оптимального решения AGP для *случайных ортогональных*, *случайных простых* и *RvK* многоугольников. Видно, что *теневая* стратегия *AVPs* следует той же схеме, что и в случае CvK для RvK, причем размер дискретизации быстро растет по мере увеличения числа вершин многоугольников. Тем не менее, как мы увидим, подход *shadow AVPs* хорошо подходит для случайных многоугольников. Кривые, соответствующие стратегии *"все вершины"*, говорят о том, что набор вершин многоугольника является хорошей заявкой на начальную дискретизацию, поскольку для ее достижения требуется добавить немного новых точек.

Таблица 1  
Результаты для полных многоугольников фон Коха

# вершин	Окончательный размер дискретизации				Общее время (в секундах)			
	20	100	500	2500	20	100	500	2500
Одиночная вершина	9	82	326	1185	0.04	1.23	26.26	808.95
Выпуклые вершины	12	60	279	1403	0.04	0.84	21.09	720.99
Все вершины	20	115	552	2687	0.03	1.22	27.50	828.54
Полигоны атомарной видимости теней	20	244	5029	-	0.06	2.43	143.75	-

© 2011 Авторы.



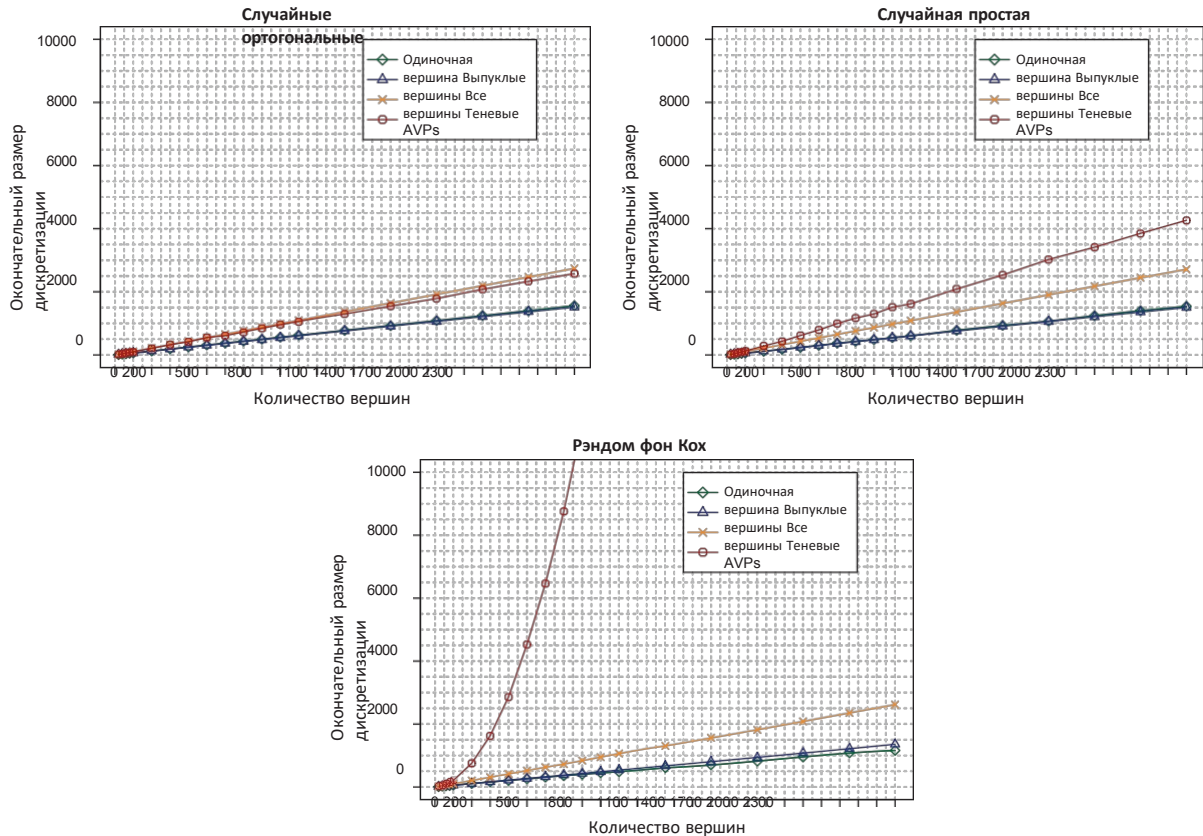


Рис. 10. Окончательный размер дискретизации по типам полигонов.

оптимальное решение экземпляра AGP во всех трех рассматриваемых классах. Заметим, что то же самое наблюдение применимо к стратегиям *с одной вершиной* и *выпуклыми вершинами*. Отсюда можно сделать вывод, что малые хорошо подобранные правильные подмножества  $V$  могут с достаточной степенью вероятности отразить соответствующую часть трудности задачи. Однако, как мы вскоре увидим, стратегии, начинающиеся с мизерных дискретизаций, крайним случаем которых является *единственная вершина*, могут привести к тому, что алгоритм будет слишком много итерироваться, увеличивая время вычислений.

На рисунке 11 показано количество итераций, необходимое каждой стратегии для достижения оптимального решения для трех классов случайных многоугольников. Линии, соответствующие *теневой* стратегии *AVPs*, приравнены к постоянной функции значения один и включены в графики исключительно в качестве ссылки. Напомним, что из-за нехватки памяти мы успешно запустили нашу программу для класса RvK для экземпляров, содержащих не более 2250 вершин.

Рассмотрим теперь стратегию *с одной вершиной*. Для *случайных ортогональных* и *случайных простых* многоугольников любая единственная точка  $P$  отражает только строго локальную информацию о форме многоугольника. Таким образом, начав с унитарной дискретизации, следует ожидать нескольких итераций, прежде чем  $D(P)$  станет достаточно плотным, чтобы отразить форму  $P$ . Эта ситуация очень наглядно представлена на *примере одиночной вершины*



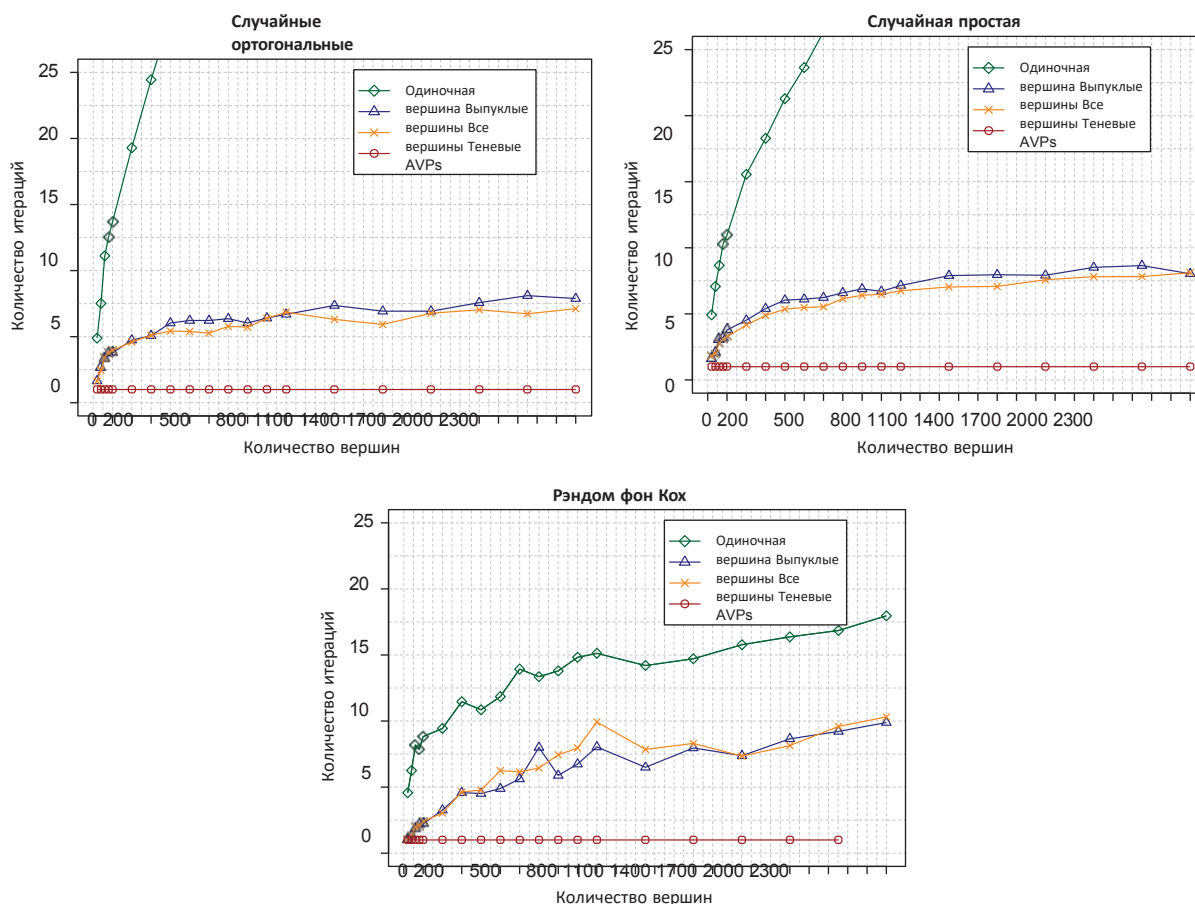


Рис. 11. Количество итераций в зависимости от типа полигона.

кривые *вершин* для случайных ортогональных и случайных простых классов. Как и для экземпляров RvK, многоугольник видимости большинства вершин соответствует большей части  $P$ , что приводит к множеству многократно покрытых областей внутри  $P$ . Таким образом, даже если мы начинаем с одиночной вершины, сходимость происходит гораздо быстрее, чем для двух других классов.

Теперь, глядя на три непостоянные кривые внутри каждого графика, мы видим, что количество итераций увеличивается по мере уменьшения размера *исходных* дискретизаций. Что касается размера входного многоугольника, то количество итераций остается незначительным по сравнению с теоретическим ограничением  $Y(n^3)$  (см. раздел 3.2). Что касается RvK-полигонов, то количество итераций растет немного быстрее с увеличением размера экземпляра, но все равно остается достаточно малым. С другой стороны, близость кривых для *выпуклых вершин* и *всех вершин* показывает, что только выпуклые вершины достаточно хорошо передают форму  $P$ , чтобы обойтись без рефлекторных вершин. Поэтому стратегия *выпуклых вершин* требует чуть больше итераций, чем стратегия *всех вершин*.

На рисунке 12 представлена диаграмма в виде квадрата с количеством итераций, выполненным алгоритмом с использованием стратегии *выпуклых вершин* для класса RvK. Для большинства размеров экземпляров видно, что

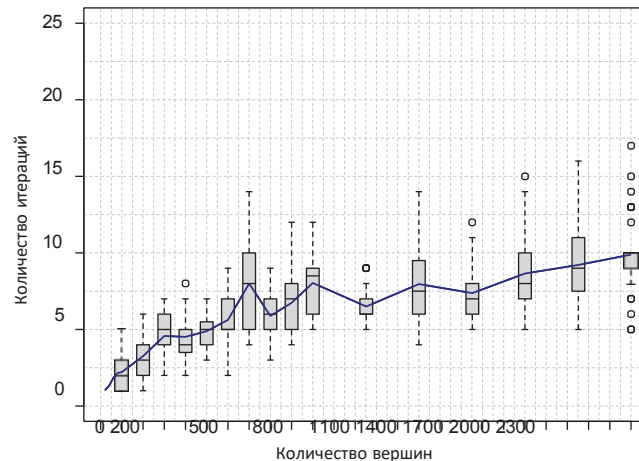


Рис. 12. График, показывающий количество итераций для случайных многоугольников фон Коха при использовании стратегии *выпуклых вершин*.

Средние и медианные значения очень близки. Разница между верхним и нижним квартилями никогда не превышает 5, и во всех случаях, кроме самых крупных, не встречается более одного выброса. Даже для многоугольников размером 2500, поведение которых кажется необычным, из 30 экземпляров восемь выбросов являются вполне приемлемыми. Это подтверждает устойчивость предложенного алгоритма с данной стратегией.

На рисунке 13 показано общее время, включая фазы предварительной обработки и обработки, затраченное на решение примеров из трех случайных классов многоугольников. Обратите внимание, что все графики построены в одном масштабе. Чтобы правильно проанализировать этот график, необходимо помнить о предыдущем обсуждении количества итераций и размера дискретизации, производимой каждой из альтернативных стратегий. Во-первых, обратите внимание, что стратегии *"Выпуклые вершины"* и *"Все вершины"* приводят к очень близкому времени вычислений. Ранее мы видели, что количество их итераций невелико и очень близко друг к другу. Мы также заметили, что размер конечных дискретизаций медленно (почти линейно) растет с размером экземпляра и что размер дискретизации, соответствующий стратегии *"Все вершины"*, лишь немного больше, чем размер дискретизации для стратегии *"Выпуклые вершины"*. Такое сходство объясняется тем, что в обоих случаях решатель ИС должен вычислять более легкий экземпляр SCP на каждой итерации алгоритма. Алгоритмы, полученные с помощью этих двух стратегий, не только быстрые, но и очень надежные. Чтобы убедиться в этом, заметим, что кривые для *всех* стратегии *вершин* и *выпуклых вершин* остаются абсолютно одинаковыми при изменении классов экземпляров.

С другой стороны, стратегия *с одной вершиной* плохо ведет себя для классов *случайных ортогональных* и *случайных простых*. Хотя в среднем она всегда дает наименьшую дискретизацию, количество итераций, требуемых этой стратегией, растет очень быстро. Хотя можно было бы также ожидать, что легкие экземпляры SCP будут оптимизироваться на каждой итерации, накладные расходы на многочисленные вызовы IP-решателя превосходят преимущества небольших экземпляров. Только для многоугольников RvK стратегия *одной вершины* становится конкурентоспособной по отношению к стратегиям *выпуклых вершин* и *всех вершин*, что предсказуемо, поскольку для этих многоугольников мы видим, что стратегия *одной вершины* обычно выполняет всего 10 итераций по сравнению со средним значением двух других стратегий.

Теперь мы проанализируем поведение алгоритма при использовании *теневой* стратегии *AVPs*. Этот вариант алгоритма превосходит все остальные для *случайных ортогональных* и *случайных простых* классов, но

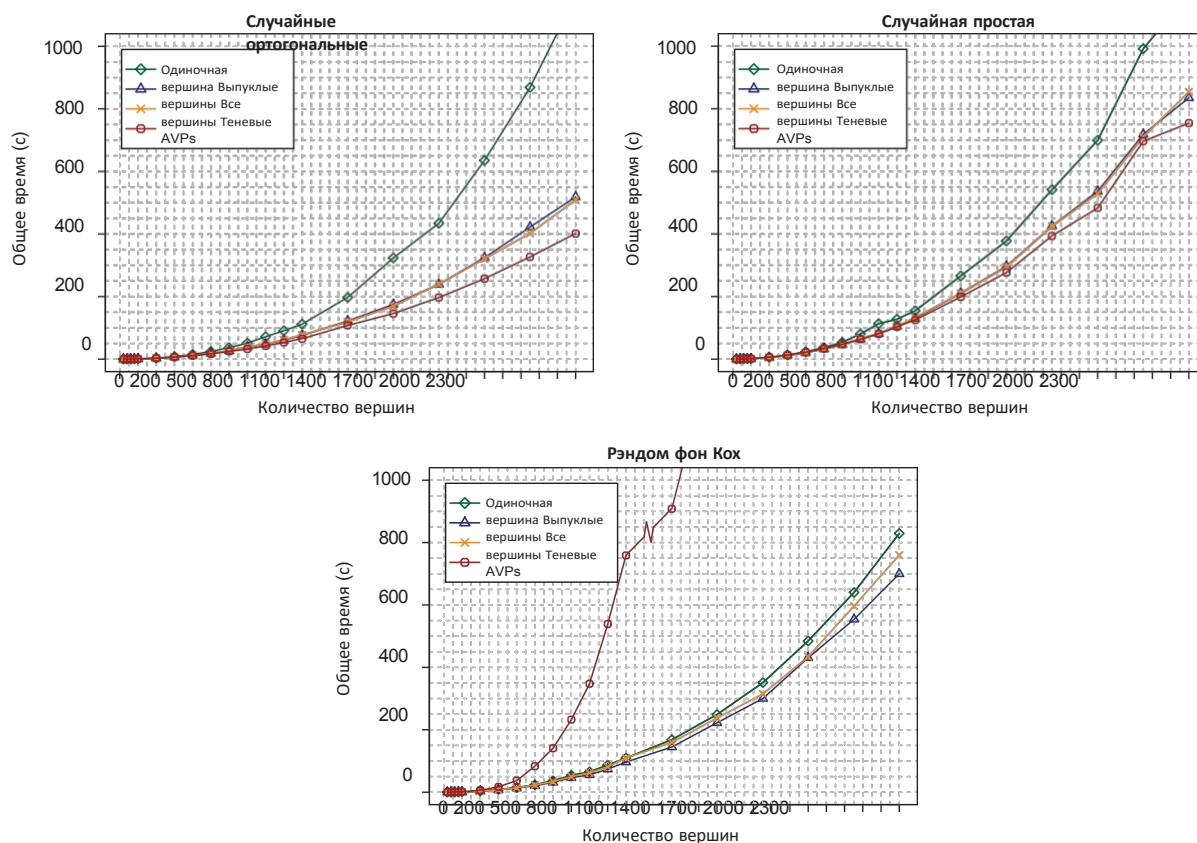


Рис. 13. Общее время по типам полигонов.

то для экземпляров RvK с несколькими сотнями вершин он становится чрезмерно медленным. Чтобы объяснить эти результаты, мы снова обратимся к рис. 10.

В классе RvK, напомним, количество теневых AVP быстро растет с размером экземпляра. Таким образом, хотя решается один экземпляр SCP, время, затрачиваемое на вычисление матрицы ограничений, связанной с этим экземпляром, огромно. Кратко отложим обсуждение этого последнего вопроса. На рис. 10 видно, что для *случайных ортогональных* и *случайных простых* классов итоговая дискретизация или, что то же самое, количество теневых AVP растет почти линейно с размером экземпляра. На самом деле, оно не намного больше, чем размер конечных дискретизаций, получаемых стратегиями "*Выпуклые вершины*" и "*Все вершины*". Однако мы знаем, что в этом достаточно одной итерации алгоритма. Поэтому размер единственного решаемого экземпляра SCP не намного больше размера экземпляра, решенного в последней итерации стратегий "*Все вершины*" и "*Выпуклые вершины*". Теперь обратим внимание на время, затрачиваемое алгоритмом на каждом этапе относительно стратегий дискретизации. Напомним, что этап предварительной обработки состоит из трех процедур. Первая является общей для всех стратегий и вычисляет полигоны видимости каждой вершины. Вторая вычисляет начальную дискретизацию, и ее стоимость сильно зависит от выбора

© 2011 Авторы.

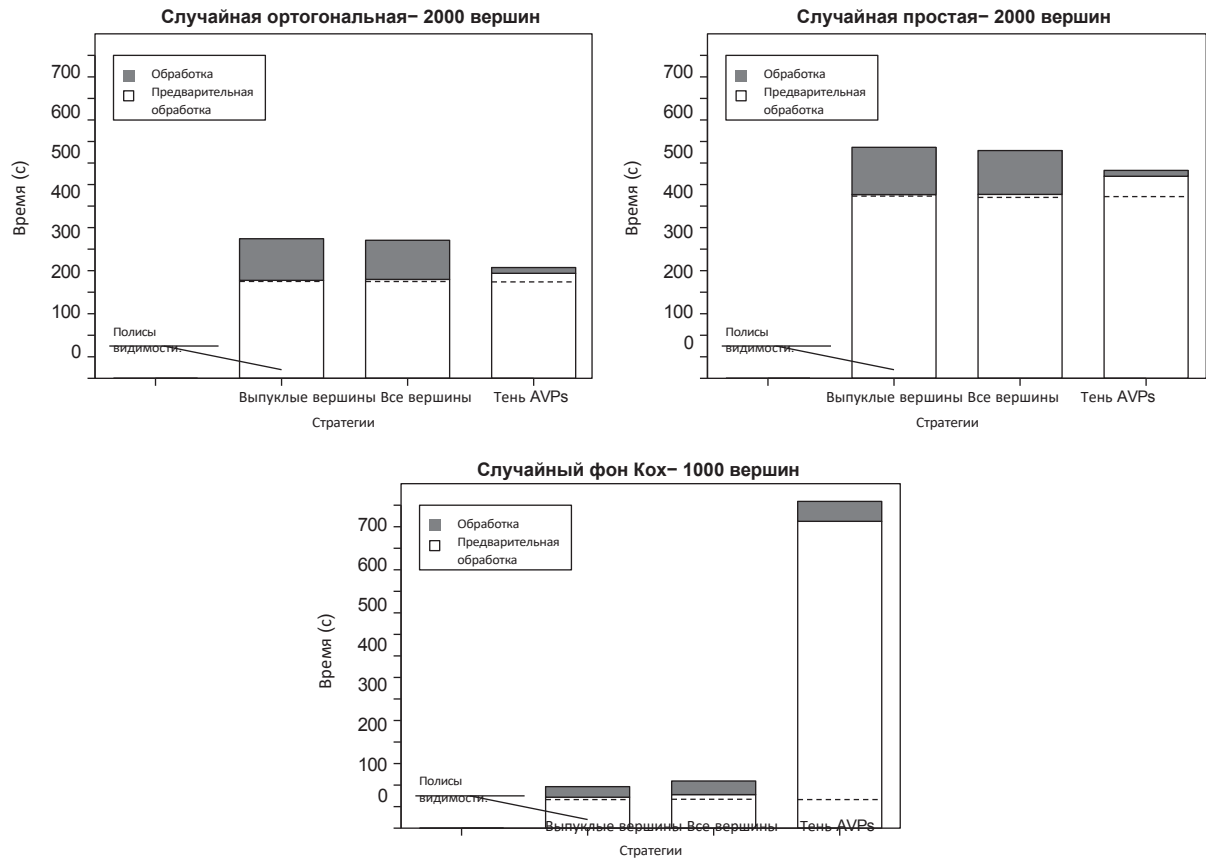


Рис. 14. Разбиение времени выполнения на обработку и препроцессинг для полигонов случайных классов.

стратегия, которую необходимо реализовать. Наихудший случай соответствует стратегии *теневого AVPs*, поскольку она требует вычисления всех AVPs и определения теневого AVPs вместе с их центроидами. С другой стороны, стратегии *"одна вершина"* и *"все вершины"* не требуют никаких вычислений для второй процедуры, в то время как для стратегии *"выпуклые вершины"* необходимо провести несколько недорогих вычислений, чтобы определить, какие вершины являются выпуклыми. Наконец, в третьей процедуре этапа предварительной обработки необходимо построить начальную модель ИС, и время, затрачиваемое на это, зависит от размера дискретизации. Это явно благоприятствует стратегии с *одной вершиной*, а также, хотя и в незначительной степени, стратегиям с *выпуклыми вершинами* и *всеми вершинами*.

На рисунке 14 показано время вычислений *случайных ортогональных* и *случайных простых* многоугольников на 2000 вершин и многоугольников RvK на 1000 вершин. Обратите внимание, что для удобства сравнения на всех трех графиках используется один и тот же масштаб - именно по этой причине в классе RvK рассматривались более мелкие экземпляры. Столбики на этих диаграммах показывают долю общего времени, затраченного на этапы обработки и препроцессинга, а для последнего - долю, затраченную на процедуру вычисления полигонов видимости.

Видно, что время, затраченное на фазу предварительной обработки, соответствует приведенному выше обсуждению: стратегия *shadow AVPs* является наиболее трудоемкой для RvK-полигонов. Для *случайных ортогональных* и *случайных простых* классов время предварительной обработки для стратегии *shadow AVPs* примерно такое же, как и для двух других стратегий, и ее преимущество проявляется только на этапе обработки. Первые два графика также иллюстрируют тот факт, что случайные простые полигоны имеют более сложную структуру видимости, чем полигоны ортогонального класса.

Удивительно то, что, хотя на этапе решения мы решаем NP-трудные задачи, во всех случаях большая часть временных затрат приходится на этап предобработки, который полностью полиномиален. Необычайное развитие решателей ИС, а также тот факт, что экземпляры SCP, возникающие в AGP, являются одними из самых простых, объясняет это кажущееся нелогичным поведение алгоритма. Таким образом, прорыв в производительности нашего алгоритма был бы достигнут, если бы удалось придумать дискретизацию, получаемую с помощью очень быстрой процедуры и, в то же время, удовлетворяющую свойству, что для достижения оптимума экземпляра AGP достаточно одной итерации. Сравнивая размеры конечных дискретизаций различных стратегий, показанных ранее, можно сделать вывод, что возможности для таких улучшений есть.

## 6. Выводы и замечания

В данной работе мы обобщили и расширили наши исследования, посвященные точным подходам к решению AGP. В предыдущих работах (Couto et al., 2007, 2008) мы сосредоточились на галереях, представленных ортогональными многоугольниками, и предложили алгоритм, также рассмотренный в разделе 3, основанный на последовательной дискретизации входного многоугольника. Наши предыдущие вычислительные эксперименты были ограничены экземплярами с количеством вершин не более тысячи, в то время как здесь мы расширили алгоритм для работы с неортогональными многоугольниками и протестировали его на экземплярах с количеством вершин до 2500. Кроме того, мы предложили новые стратегии дискретизации, поскольку алгоритм очень чувствителен к выбору дискретизации. В результате мы предложили стратегию *выпуклых вершин*, которая показала наилучшую производительность на сегодняшний день.

Напомним, что точный алгоритм опирается не только на дискретизацию внутренней части входного многоугольника, но и на моделирование этой упрощенной дискретной задачи как SCP. Полученный экземпляр SCP решается до оптимальности с помощью IP-решателя, а если остаются непокрытые области, то включаются дополнительные ограничения и процесс повторяется. Очевидно, что производительность алгоритма зависит и от количества таких итераций.

В то время как основное внимание уделялось новым стратегиям реализации шага дискретизации, был проведен тщательный эксперимент для оценки компромисса между количеством итераций и временем, затрачиваемым множеством вариантов алгоритма, возникающих при использовании альтернативных методов дискретизации.

Подтверждая результаты наших предыдущих работ, предложенный алгоритм показал отличную общую производительность. Он также оказался устойчивым в том смысле, что смог справиться с экземплярами из широкого спектра классов многоугольников. Более того, самые быстрые варианты алгоритма очень быстро находили решения для экземпляров с более чем 2000 вершинами. Это более чем в два раза превышает размер самых больших решенных нами ранее экземпляров, которые, в свою очередь, были в пять раз больше тех, о которых сообщалось ранее в литературе. Стратегия *выпуклых вершин*, предложенная в разделе 4, дает разреженную дискретизацию и, как следствие, небольшие экземпляры SCP. Как видно из таблицы 2, это приводит к очень быстрой реализации для экземпляров с количеством вершин до 2500.

Таблиц  
а 2

Общее время (в секундах) для стратегии выпуклых вершин

n	Классы многоугольников			
	Случайные ортогональные	Случайная простая	RvK	CvK
100	0.74	1.38	0.76	0.84
500	18.64	32.48	20.82	21.09
2500	518.60	834.78	699.74	720.99

RvK - случайный фон Кох; CvK - полный фон Кох.

С другой стороны, как мы заметили в Couto et al. (2008), очевидное преимущество дискретизации, обеспечивающей точное решение после одной итерации алгоритма, не было подтверждено. В частности, это произошло со стратегией *теневых AVPs*, неэффективность которой была обусловлена дорогостоящим этапом предварительной обработки, на котором вычисляются теневые AVPs. Для этих вычислений мы алгоритм полиномиального времени, реализованный с помощью мощных структур данных и эффективных библиотечных пакетов для выполнения необходимых геометрических операций. И все же нам не удалось существенно снизить время препроцессирования точного алгоритма в рамках стратегии *теневых AVP*.

Вопреки ожиданиям, в случае *теневых AVP* препроцессирование оказалось более затратным по времени, чем решение экземпляра SCP, известной NP-трудной задачи. Успех этого алгоритма можно приписать необычайному развитию решателей ИС в последние годы. Достижения в этой области сделали возможным решение больших экземпляров SCP за очень малое время.

Остается открытым вопрос, сможем ли мы найти еще одну дискретизацию, приводящую к одной итерации алгоритма, которая вычисляется за время, ограниченное полиномом очень малой степени от числа вершин. Это перспективная тема для будущих исследований, которая может оказаться полезной для нашего алгоритма.

В качестве заключительного замечания отметим, что мы сосредоточились на проблеме охраны вершин, поскольку это, вероятно, тот вариант общей AGP, которому до сих пор уделялось наибольшее внимание в литературе, хотя точный алгоритм до сих пор не найден. Тем не менее, представленные здесь методы, безусловно, могут быть адаптированы к любому случаю, в котором кандидаты в охранники образуют выполнимое конечное множество. Более того, актуальный случай простых многоугольников с отверстиями, который не рассматривался в данной работе, требует более сложной реализации для точного вычисления многоугольников видимости. Это одно из направлений наших дальнейших исследований по данной теме.

## Благодарности

Исследование было частично поддержано CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - Grant #301732/2007-8, #472504/2007-0, #483177/2009-1, #473867/ 2010-9 и FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo - Grant #07/ 52015-0 и Grant from FAEPEX/Unicamp.

© 2011 Авторы.

## Ссылки

- Амит, Й., Митчелл, Дж.С.Б., Пакер, Э., 2007. Определение местоположения охранников для обеспечения видимости многоугольников. Труды семинара по разработке алгоритмов и экспериментам, Новый Орлеан, Лос-Анджелес, 6 января 2007 г., стр. 120-134.
- Баумгартнер, Т., Фекете, С.П., Крӑлтер, А., Шмидт, К., 2010. Точные решения и границы для общих задач о художественных галереях. Труды 12-го семинара по разработке алгоритмов и экспериментам (ALENEX), Остин, Техас, 16 января 2010 г., стр. 11-22.
- Бозе, П., Любив, А., Манро, Дж. И., 2002. Безопасные запросы видимости в простых многоугольниках. *Вычислительная геометрия* 23, 3, 313-335.
- Чу Тал, В., 1975. Комбинаторная теорема в геометрии плоскости. *Журнал комбинаторной теории Серия В* 18, 39-41.
- Couto, M.C., de Rezende, P.J., de Souza, C.C., 2009a. IP-решение проблемы художественной галереи. В материалах 25-го ежегодного симпозиума ACM по вычислительной геометрии, Орхусский университет, Дания, 8 июня 2009 г., стр. 88-89. [dx.doi.org/10.1145/1542362.1542378](https://doi.org/10.1145/1542362.1542378).
- Куто, М.К., де Резенде, П.Ж., де Соуза, К.К., 2009b. IP-решение проблемы художественной галереи. Видео, опубликованное на 18-м ежегодном видео/мультимедийном обзоре по вычислительной геометрии, Орхусский университет, Дания, 8 июня 2009 г. Доступно на сайте <http://www.computational-geometry.org/SoCG-videos/socg09video>.
- Couto, M.C., de Souza, C.C., de Rezende, P.J., 2007. Точный и эффективный алгоритм для решения задачи ортогональной галереи. Труды XX Бразильского симпозиума по компьютерной графике и обработке изображений, IEEE Computer Society, Belo Horizonte, MG, Brazil, pp. 87-94.
- Куто, М.К., де Соуза, К.К., де Резенде, П.Ж., 2008. Экспериментальная оценка точного алгоритма для проблемы ортогональной галереи. В Трудях WEA, том 5038 Лекций по информатике, стр. 101-113. Доступно на сайте <http://dblp.uni-trier.de/db/conf/wea/wea2008.html#CoutoSR08> (доступ получен 4 июня 2008 г.).
- Couto, M.C., de Souza, C.C., de Rezende, P.J., 2009c. Экземпляры для проблемы художественной галереи. Доступно по адресу <http://www.ic.unicamp.br/~cid/Problem-instances/Art-Gallery> (доступ получен 28 ноября 2009 г.).
- Эдельсбруннер, Х., О'Рурк, Ж., Вельцль, Е., 1984. Расстановка охранников в прямолинейных художественных галереях. *Компьютерное зрение, графика и обработка изображений* 27, 167-176.
- Эйденбенц, С., 2002. Аппроксимационные алгоритмы для охраны местности. *Information Processing Letters* 82, 2, 99-105.
- Эрдем, У.М., Склярф, С., 2006. Автоматизированное расположение камер для удовлетворения требований к покрытию, специфичных для конкретной задачи и плана этажа. *Компьютерное зрение и понимание изображений*, 103, 3, 156-169.
- Фалькoner, К., 1990. *Фрактальная геометрия, математические основы и приложения*. John Wiley & Sons, Chichester, pp. 120-121.
- Гхош, С.К., 1987. Аппроксимационные алгоритмы для задач о художественных галереях. В материалах конгресса Канадского общества обработки информации, Виннипег, Манитоба, Канада, 12 мая 1987 г., с. 429-434.
- Гхош, С.К., 2007. *Алгоритмы видимости на плоскости*. Cambridge University Press, New York.
- Гхош, С.К., 2010. Аппроксимационные алгоритмы для задач о художественной галерее в многоугольниках. *Дискретная прикладная математика* 158, 6, 718-722.
- Хонсбергер, Р., 1976. *Mathematical Gems II (Dolciani Mathematical Expositions, No. 2)*. Математическая ассоциация Америки, Вашингтон, округ Колумбия.
- Джо, Б., Симпсон, Р.Б., 1985. Видимость простого многоугольника из точки. Отчет CS-85-38, математики и компьютерных наук, Дрексельский университет, Филадельфия, штат Пенсильвания.
- Джо, Б., Симпсон, Р.Б., 1987. Поправка к алгоритму многоугольника видимости Ли. *Поведение и информационные технологии* 27, 458-473.
- Джонсон, Д.С., 2002. Руководство теоретика по экспериментальному анализу алгоритмов. В книге Goldwasser, M.H., et al. (eds) *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*. AMS, Провиденс, РИ, стр. 215-250.
- Кан, Дж., Клау, М.М., Клейтман, Д., 1983. Традиционные галереи требуют меньшего количества сторожей. *SIAM Journal of Algebraic and Discrete Methods* 4, 194-206.
- Ли, Д.Т., 1983. Видимость простого многоугольника. *Компьютерное зрение, графика и обработка изображений* 22, 207-221.
- Ли, Д.Т., Лин, А.К., 1986. Вычислительная сложность задач художественной галереи. *IEEE Transactions on Information Theory* 32, 2, 276-282.
- МакГеох, К.К., Морет, Б.М.Е., 1999. Как представить доклад об экспериментальной работе с алгоритмами. *Новости SIGACT*, том 30.

- Морет, Б., 1986. На пути к дисциплине экспериментальной алгоритмики. В материалах 5-й конференции DIMACS Challenge, Центр DIMACS, Университет Ратгерса, 28-30 октября 1996 года.
- О'Рурк, Дж., 1987. *Теоремы и алгоритмы художественной галереи*. Оксфорд Университи Пресс, Лондон.
- Sack, J.-R., Toussaint, G.T., 1988. Размещение стражей в прямолинейных многоугольниках. В Toussaint, G.T. (ed.) *Computational Morphology*. North-Holland, Amsterdam, pp. 153-175.
- Сандерс, П., 2002. *Представление данных экспериментов в алгоритмике*. Springer-Verlag, New York, NY, стр. 181-196.
- Schuchardt, D., Hecker, H.-D., 1995. Две NP-трудные проблемы арт-галереи для орто-полигонов. *Mathematical Logic Quarterly* 41, 261-267.
- Шермер, Т.К., 1992. Последние результаты в художественных галереях. *Труды IEEE* 80, 9, 1384-1399.
- Tomas, A.P., Bajuelos, A.L., 2004. Генерация случайных ортогональных многоугольников. In Conejo, R., Urretavizcaya, M., Perez-de-la-Cruz, J.-L. (eds) *Current Topics in Artificial Intelligence, Volume 3040 of LNCS*. Springer, Berlin, pp. 364- 373.
- Tomas, A.P., Bajuelos, A.L., Marques, F., 2006. О проблемах видимости на плоскости - решение задач о минимальной охране вершин путем последовательных приближений. Труды 9-го Международного симпозиума по искусственному интеллекту и математике, Форт-Лодердейл, штат Флорида, 4 января 2006 года.
- Уррутиа, Х., 2000. Художественная галерея и проблемы освещения. В Sack, J.-R., Urrutia, J. (eds) *Handbook of Computational Geometry*. North-Holland, Amsterdam, pp. 973-1027.
- Вулси, Л.А., 1998. *Целочисленное программирование*. Wiley-Interscience, New York.