С.Л. Забелин, В.Д. Фроловский

ИССЛЕДОВАНИЕ МЕТАЭВРИСТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМАЛЬНОГО ГЕОМЕТРИЧЕСКОГО ПОКРЫТИЯ

(ФГОБУ ВПО «Сибирский государственный университет
телекоммуникаций и информатики»)

Области применения задач максимального геометрического покрытия охватывают большое количество сфер деятельности человека. Они появляются в системах воздушного и космического наблюдения, системах безопасности, агротехнических системах, системах проектирования и других.

Задача геометрического покрытия является частным случаем задачи оптимального проектирования и принадлежит к классу задач «раскроя и упаковки». Требуется расположить некоторые геометрические объекты на покрываемой поверхности таким образом, чтобы вся поверхность была покрыта целиком с наименьшей площадью перекрытий и промахов объектов, а также использовать минимум объектов. Актуальность данной задачи обусловлена также её принадлежностью к классу NP-трудных задач, сложность которых очень быстро возрастает с увеличением размерности. Для решения задач малой размерности можно использовать точные алгоритмы. Однако для задач большей размерности точные методы требуют значительных временных затрат и не дают хороших результатов. Эффективным является использование метаэвристических методов.

В статье рассматривается решение задач оптимального геометрического покрытия с помощью «первого подходящего» (*FirstFit*), вероятностного (*Probabilistic*), экстремального (*Extreme*) и муравьиного(*Ant*) алгоритмов. Реализация выполнена на языке программирования *C#.*

В основе перечисленных алгоритмов лежит метод наложения двумерных матриц.В двумерном пространстве имеется покрываемая поверхность S_0 и покрывающие прямоугольные объектыS_1,S_2,…,S_h, где h – общее количество заданных объектов различной размерности. Требуется расположить прямоугольные объекты из области Sна покрываемой поверхностиS_0 таким образом, чтобы вся поверхность была покрыта целиком, т.е. должно выполняться следующее условие:

$$S_0 \cap \left(\bigcup_{i=1}^{h} S_h \right) = S_0$$

Объекты могут быть использованы только один раз. В данной задаче принято, что все объекты имеют растровую структуру. В двумерной матрице каждая ячейка представляет собой пиксель на плоскости (*Ox; Oy).* Введены следующие обозначения:

1) I – количество пропусков «*skIp*» (точки со значением «1»);
2) S - количество успехов «*Success*» (точки со значением «2»);

3) *M* - количество промахов «*Miss*» (точки со значением «*a,b,c,…*»);
4) *O* - количество переполнений «Overflow» (точки со значением «*A,B,C,…*»);
5) *Sum* – количество точек всех используемых покрывающих объектов S_h .

Для определения эффективности покрытия введен коэффициент «Success», отражающий степень успешного расположения объектов на покрываемой поверхности*($0<K_S\leq 1$, $K_S\to 1$):*

$$K_S = \frac{S}{Sum}$$

В программе также рассчитываются и анализируются коэффициенты, отражающие количество промахов *($K_M\to 0$)*, переполнений *($K_O\to 0$)*, пропусков*($K_I\to 0$).*

Покрытие поверхности S_0 осуществляется объектамиS_h различной формы. Импортировать данные в программу можно рисунками с расширением *.bmp*или *.png*, которые преобразуются в матрицу.

Наложение объектов на поверхности осуществляется с помощью специальных «якорей» установленных на каждый объект. «Якорь» - это точка в объекте, которая имеет метку.

Для всех алгоритмов, кроме вероятностного, введена случайность выбора места расположения объекта, поэтому с каждым новым выполнением результат будет отличаться от предыдущего. Для ведения статистики выполнений с разными случайностями используется настраиваемый параметр, определяющий количество выполнений данного алгоритма. С помощью номера случайности всегда можно воспроизвести нужный вариант покрытия. Случайность алгоритма формируется автоматически при переносе его в область выполнения и при изменении количества выполнений в параметрах алгоритма и зависит от частоты процессора и времени создания. Случайность алгоритма можно задавать вручную. Случайность равная нулю говорит о том, что алгоритм работает без случайности, то есть берет первый подходящий вариант покрытия для каждого объекта.

В выводимых параметрах алгоритма можно увидеть коэффициенты «*Success*» для всех вариантов выполнения. Также для этих значений рассчитывается математическое ожидание и среднеквадратическое отклонение варианта покрытия по следующим формулам:

$$\overline{K_S} = \frac{1}{n}\sum_{i=1}^{n} K_{S\,i}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(K_{S\,i} - \overline{K_S})}$$

где $\overline{K_S}$ – математическое ожидание коэффициента «Success»;

σ – среднеквадратическое отклонение;

n – количество выполнений алгоритма;

K_S – коэффициент «Success».

Анализ работы алгоритмов проведен на поверхности размером 30 на 21 точку. Основные результаты сведены в таблицу 1.

Таблица 1. Коэффициент «*Success*» для исследуемых алгоритмов на поверхности размером 30 на 21 точку

Группа	Вариант алгоритма	Покрываемая поверхность			
		Кирпич	Кольцо	Ромашка	Случайное
к16	*01_Alg_First_Fit*	0,82031	0,52546	0,25947	0,48547
к16	*02_Alg_Probabilistic*	0,82031	0,52546	0,25947	0,48547
к16	*03_Alg_Extreme*	0,82031	0,52546	0,25947	0,48547
к16	*07_Alg_Ant*	0,82031	0,41164	0,23346	0,37778
к16	*08_Alg_Ant*	0,65625	0,54398	0,41827	0,47321
6к	*01_Alg_First_Fit*	0,74713	0,62470	0,50000	0,53456
6к	*02_Alg_Probabilistic*	1,00000	0,78873	0,71429	0,74364
6к	*03_Alg_Extreme*	0,75762	0,70801	0,62431	0,61735
6к	*07_Alg_Ant*	1,00000	0,59466	0,73203	0,54817
6к	*08_Alg_Ant*	1,00000	0,69110	0,75410	0,58081
п15	*01_Alg_First_Fit*	0,87246	0,52688	0,42000	0,53643
п15	*02_Alg_Probabilistic*	1,00000	0,46667	0,24571	0,45606
п15	*03_Alg_Extreme*	0,83472	0,51042	0,41290	0,51515
п15	*07_Alg_Ant*	0,91304	0,30588	0,25294	0,47317
п15	*08_Alg_Ant*	0,82609	0,45417	0,38095	0,56581
6п	*01_Alg_First_Fit*	0,75597	0,55952	0,45752	0,56964
6п	*02_Alg_Probabilistic*	0,93611	0,68966	0,43382	0,56544
6п	*03_Alg_Extreme*	0,76821	0,62385	0,60695	0,63316
6п	*07_Alg_Ant*	1,00000	0,66216	0,60580	0,73651
6п	*08_Alg_Ant*	1,00000	0,84709	0,63037	0,75000
г16	*01_Alg_First_Fit*	0,70588	0,46780	0,39583	0,45313
г16	*02_Alg_Probabilistic*	0,87216	0,54688	0,37723	0,40767
г16	*03_Alg_Extreme*	0,68137	0,51458	0,39583	0,43886
г16	*06_Alg_Ant*	0,64844	0,35547	0,32083	0,41815
г16	*07_Alg_Ant*	0,71875	0,44583	0,40179	0,47656
6г	*01_Alg_First_Fit*	0,53913	0,43654	0,36599	0,46485
6г	*02_Alg_Probabilistic*	0,85468	0,50110	0,45084	0,48323
6г	*03_Alg_Extreme*	0,60209	0,47667	0,54156	0,45816
6г	*06_Alg_Ant*	0,78723	0,44676	0,41190	0,47791
6г	*07_Alg_Ant*	0,80226	0,55023	0,48515	0,53188
т16	*01_Alg_First_Fit*	0,64183	0,50431	0,33902	0,45516

Группа	Вариант алгоритма	Покрываемая поверхность			
		Кирпич	Кольцо	Ромашка	Случайное
т16	*02_Alg_Probabilistic*	0,86932	0,52232	0,31875	0,44048
т16	*03_Alg_Extreme*	0,69250	0,49792	0,35625	0,45313
т16	*06_Alg_Ant*	0,66927	0,43103	0,30417	0,36806
т16	*07_Alg_Ant*	0,70313	0,47991	0,39509	0,43598
6т	*01_Alg_First_Fit*	0,42141	0,38160	0,33794	0,34733
6т	*02_Alg_Probabilistic*	0,48077	0,44422	0,22842	0,41326
6т	*03_Alg_Extreme*	0,48705	0,40663	0,44796	0,41467
6т	*06_Alg_Ant*	0,68684	0,61578	0,41149	0,45984
6т	*07_Alg_Ant*	0,74247	0,61461	0,49517	0,46860
6к_6п_6г_6т	*01_Alg_First_Fit*	0,57080	0,44526	0,42245	0,45101
6к_6п_6г_6т	*02_Alg_Probabilistic*	0,90649	0,67848	0,44522	0,60000
6к_6п_6г_6т	*03_Alg_Extreme*	0,66788	0,59674	0,64045	0,55468
6к_6п_6г_6т	*06_Alg_Ant*	0,96568	0,81707	0,74671	0,70319
6к_6п_6г_6т	*07_Alg_Ant*	0,96423	0,85061	0,76316	0,70518
6с	*01_Alg_First_Fit*	0,32397	0,29073	0,28061	0,28337
6с	*02_Alg_Probabilistic*	0,37702	0,31478	0,31619	0,31343
6с	*03_Alg_Extreme*	0,36615	0,31419	0,37263	0,33824
6с	*06_Alg_Ant*	0,50116	0,45536	0,52661	0,36471
6с	*07_Alg_Ant*	0,51628	0,47748	0,36807	0,39671
6к_6п_6г_6т_6с	*01_Alg_First_Fit*	0,52106	0,41087	0,36031	0,42328
6к_6п_6г_6т_6с	*02_Alg_Probabilistic*	0,74069	0,64373	0,51629	0,52019
6к_6п_6г_6т_6с	*03_Alg_Extreme*	0,60347	0,51919	0,54315	0,50989
6к_6п_6г_6т_6с	*06_Alg_Ant*	0,94923	0,85938	0,61721	0,68898
6к_6п_6г_6т_6с	*07_Alg_Ant*	0,94444	0,84615	0,60286	0,71721

Условные обозначения:

а) «к» – квадрат, «п» – прямоугольник, «г» – г-образный, «т» – т-образный, «с» – сложный;

б) число перед буквой – количество видов объектов;

в) число после буквы – площадь единственного объекта в точках.

Алгоритм «первый подходящий» (*FirstFit*)

Алгоритм предполагает простую укладку первого попавшегося геометрического объекта на первую найденную пустую область на покрываемой поверхности. Покрытие происходит слева направо, сверху вниз.Управляемых параметров нет, кроме общих для всех алгоритмов. Он не гарантирует получения хорошего результата, но дает возможность построения хорошего начального приближения.

В управляемых параметрах установлено 1000 выполнений. Вследствие работы вероятности и большого количества повторений алгоритм может давать хорошие результаты, даже лучше более сложных алгоритмов. Например, при покрытии поверхности прямоугольниками одинаковой площади, данный алгоритм дает похожие или лучшие результаты в сравнении с более сложными алгоритмами (таблица 1, покрытие «п15»). Несмотря на большое количество выполнений, алгоритм не требует больших вычислительных ресурсов, так как случайный объект размещается на место в порядке следования «бегунка», и время работы алгоритма сравнительно не большое. Чем сложнее поверхность и более разнообразные покрывающие объекты, тем хуже работает «первый подходящий».

Вероятностный (*Probabilistic*) алгоритм

В вероятностном алгоритме каждому из S_h объектов присваивается вероятность использования в зависимости от площади объекта. Затем объекты упорядочиваются по уменьшению вероятности использования. Далее покрытие происходит некоторое количество раз по принципу алгоритма «первый подходящий», которое определяется применением доверительного коэффициента. Начальное значение доверительного коэффициента равно единице. Единственный управляемый коэффициент называется детализацией доверительности. Он показывает количество шагов одинаковой величины, за которые доверительный коэффициент уменьшиться до нуля. В этом алгоритме происходит примерка объектов на каждое место. Если объект удовлетворяет доверительному коэффициенту, то он устанавливается, иначе коэффициент уменьшается. На рисунке 1 приведено сравнение пяти алгоритмов, покрывающих шестью разными г-образными объектами.

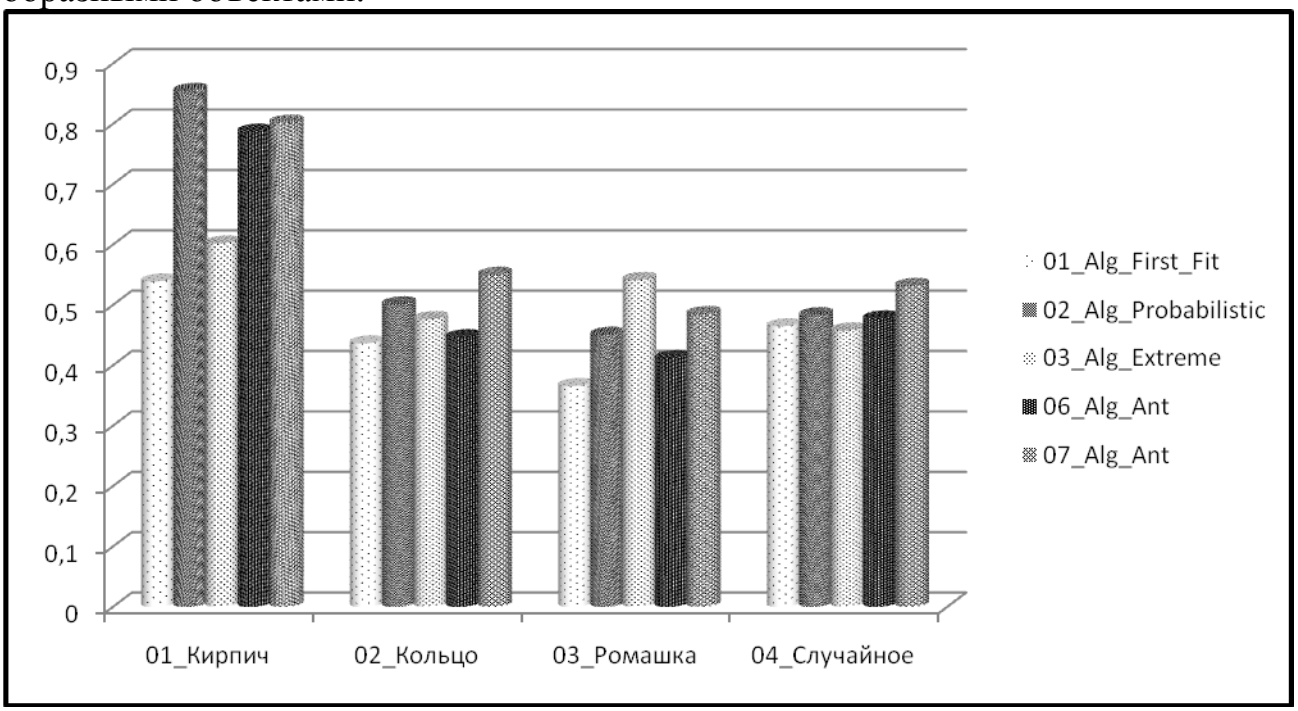

Рис. 1. Сравнение работы алгоритмов на четырех разных поверхностях, используя шесть г-образных объектов (6г)

Вероятностный алгоритм хорошо работает на поверхностях с равной шириной (например, «кирпич») и с объектами разной формы и с равномерным изменением площади. Преимущество работы рассматриваемого алгоритма видны на поверхности вида «кирпич», покрываемой шестью видами г-образных объектов, площадь которых соответственно 3, 5, 12, 16, 22, 29 точек. Покрытие изображено на рисунке 2. Однако и на других поверхностях он дает хорошие результаты с условием, что в приведенном примере площадь объектов изменяется равномерно.

Скорость работы данного алгоритма высока, так как покрытие совершается за один проход, однако скорость подбора объекта на выбранное место зависит от детализации доверительности.

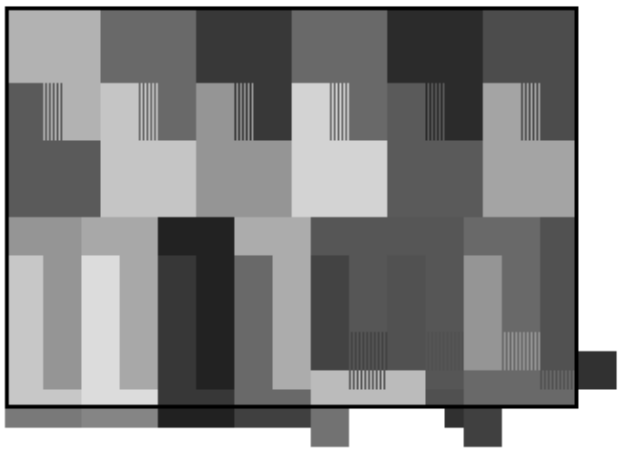

Рис. 2. Покрытие поверхности «кирпич» вероятностным алгоритмом (6г)

Экстремальный (*Extreme*) алгоритм

Экстремальный алгоритм работаеттолько с одним решением, которое шаг за шагом улучшается. Основная стратегия метода в том, чтобы найти те компоненты решения, которые влияют на него наихудшим образом, и постепенно улучшать их.

Применительно к задаче покрытия данный алгоритм будет работать следующим образом. Есть покрываемая поверхность S_0. Для покрытия этой области применен алгоритм «первый подходящий».Поверхность покрытия разбивается на две части и вычисляются их коэффициенты «*Success*». Принято, что объекты, попавшие на линию раздела принадлежат части с меньшим коэффициентом. Все объекты, принадлежащие области поверхности с худшим коэффициентом удаляются, и осуществляется покрытие всей поверхности с учетом оставшихся объектов. После полного повторного покрытия происходит расчет коэффициентов эффективности. Если коэффициент «*Success*» всей новой области меньше либо равен коэффициенту предыдущей области, то происходит возврат к старому варианту покрытия, иначе сохраняется новый результат. И так до тех пор, пока идет улучшение при заданной глубине рекурсии (количество попыток).

Экстремальный алгоритм работает с установленным количеством попыток равным тысячи раз и количеством выполнений равным ста. Из-за большого

количества простейших покрытий время выполнения алгоритма увеличивается. Например, оно в сотни раз больше времени выполнения «первого подходящего» и вероятностного алгоритма, однако почти в два раза меньше времени выполнения муравьиного алгоритма для покрываемой поверхности «ромашка» (таблица 1, объекты «6г»).

Как видно из графиков на рисунке 1 экстремальный алгоритм совершил покрытие поверхности «ромашка» лучше приведенных алгоритмов. Это обусловлено большим количеством вариантов проблемных участков поверхности, однако данный алгоритм работает не стабильно. Его результаты могут оставаться на уровне «первого подходящего».

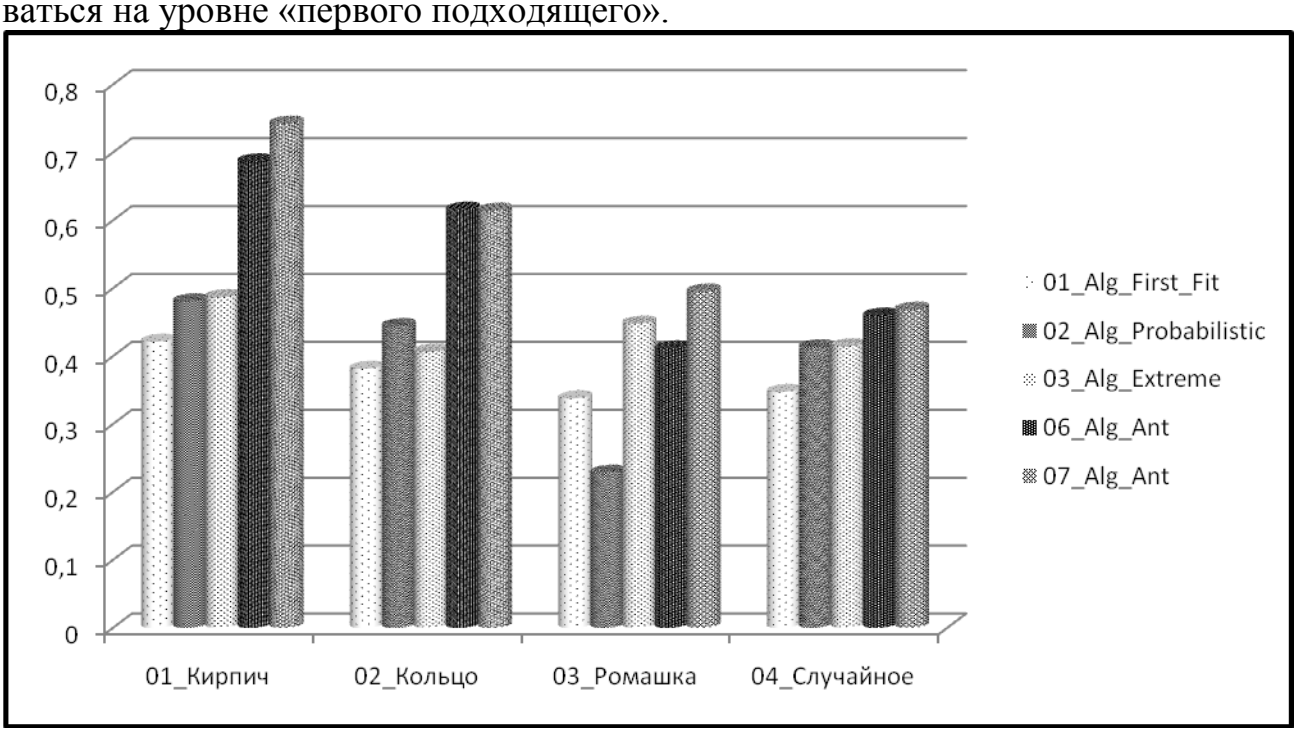

Рис. 3. Сравнение работы алгоритмов на четырех разных поверхностях, используя шесть т-образных объектов (6т)

На рисунке 3 приведено сравнение экстремального алгоритма с остальными, результаты которого часто уступают только муравьиному алгоритму. Из таблицы 1 видно, что данный алгоритм работает лучше с покрывающими объектами разной формы и площади.

Муравьиный (*Ant*) алгоритм

Муравьиный алгоритм основан на тех же принципах, по которым муравьи находят кратчайший путь между двумя точками. Муравей располагает набором из всех геометрических объектов $S_1, \ldots, S_h$ и покрываемой поверхностью S_0.

Рассмотрим реализацию муравьиного алгоритма для задачи геометрического покрытия. В предложенном исполнении алгоритма используются понятия «гордой» и «жадной» укладки. Сначала покрывающие объекты с одинаковой площадью объединяются в группы. Далее происходит сортировка получившихся групп по убыванию площади. При выполнении цикла каждой группе присваивается статус. Группа со статусом «жадной» укладки производит размещение объектов на покрываемой поверхности оптимальным способом. Оптималь-

ный способ укладки – это размещение объектов, путем полного прохождения каждым объектом каждой точки карты и выбора лучшего положения данного объекта на карте. Лучшее размещение объекта выбирается из двух условий: «жадность» и «эксклюзив». «Жадность» определяется количеством «1» под объектом с данным поворотом и на данной точке карты, а «эксклюзив» определяется суммой «1» для четырех поворотов на данной точке карты. Лучшим положением является точка на карте с максимальной «жадностью». Если точек карты с максимальной «жадностью» несколько, то лучшим положением выбирается точка с минимальным «эксклюзивом». Если точек с максимальной «жадностью» и минимальным «эксклюзивом» несколько, то точка для расположения объекта выбирается случайным образом из полученных лучших вариантов. Данным условием достигается случайность алгоритма.

Группа со статусом «гордой» укладки производит размещение объектов на покрываемой поверхности оптимальным способом с дополнительным условием. Ограничением для данного размещение является условие, при котором отношение «жадности» в данной точке карты с данным поворотом к площади покрывающего объекта равно единице. Объект в данном виде укладки может быть расположен на карте только при соблюдении этого условия. Также как и в «жадной» укладке при получении нескольких точек с идеальной «жадностью», объект располагается в точке с минимальным «эксклюзивом». Если точек с идеальной «жадностью» и минимальным «эксклюзивом» несколько, то точка для расположения объекта выбирается случайным образом из полученных лучших вариантов.

Покрытие поверхности осуществляется группами объектов. В муравьином алгоритме происходит множество проходов по группам с изменением статуса. После окончания цикла осуществляется вывод лучшего результата и расчет всех коэффициентов.

Рассматриваемый алгоритм по заданному значению выполняется 100 раз. Покрытие осуществляется:

а) по неполной карте (06_*Alg_Ant*) –поиск точки осуществляется только по непокрытой области поверхности;
б) по полной карте (07_*Alg_Ant*) – поиск точки осуществляется по всей матрице, включая точки, не принадлежащие поверхности.

Второй метод значительно увеличивает работу алгоритма из-за необходимости при укладке каждого объекта искать место по всей матрице. Однако этот метод позволяет добиваться более высоких результатов для поверхностей и объектов сложной формы.

Муравьиный алгоритм плохо работает с объектами одного размера, потому что он построен на оптимальном размещении групп объектов разной площади. Это видно на диаграмме в таблице 1, где поверхности покрываются прямоугольными объектами одинаковой формы и площади (п15). Однако уже на поверхности «случайное» муравьиный алгоритм с полной картой превосходит другие алгоритмы, так как эта поверхность является неоднородной и имеет изорванные края.

Расположение покрывающих объектов разного вида и площади на поверхностях сложной формы с помощью муравьиного алгоритма осуществляется значительно лучше за счет прохождения полной карты и преимущества «гордой» и «жадной» укладки.Сравнительный анализ работы алгоритма приведен на рисунке 4.

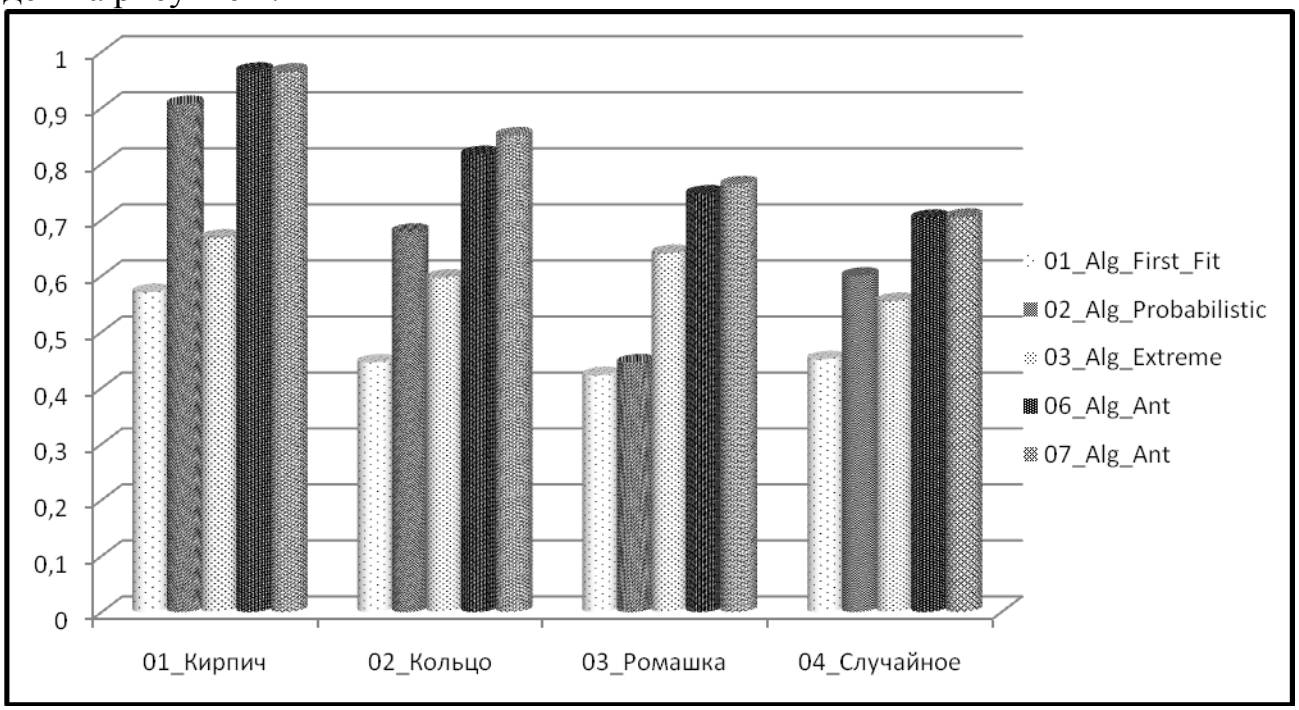

Рис. 4. Сравнение работы алгоритмов на четырех разных поверхностях, используя по шесть квадратов, прямоугольников, г-образных и т-образных объектов (6к_6п_6г_6т)

На четырех разных поверхностях с использованием большого количества разной формы и площади объектов муравьиный алгоритм показывает более высокие результаты. Отдельно на рисунке 5 приведено покрытие поверхности «ромашка» муравьиным алгоритмом с полной картой объектами различной формы и площади.

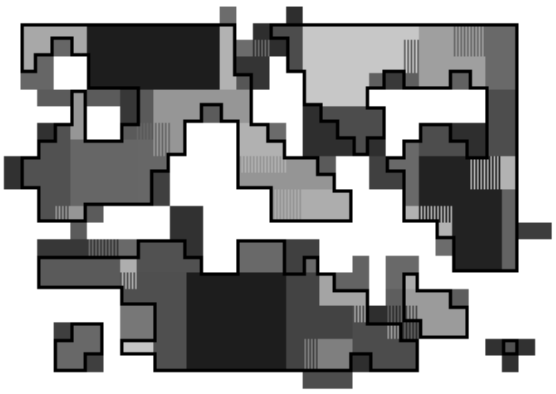

Рис. 5. Покрытие поверхности «случайное» муравьиным алгоритмом (6к_6п_6г_6т)

Литература

1. Фроловский В.Д. Приближенные методы решения *NP*-трудных задач в системах автоматизации проектирования. Новосибирск: НГТУ, 2006. – 100 с.

2. Филиппова А.С., Кузнецов В.Ю. Задачи о минимальном покрытии ортогональных многоугольников с запретными участками // Информационные технологии, 2008. – №9 (145). – с. 60 – 65

3. Мухачева А.С. простые эвристики для решения двумерной задачи максимального покрытия // Принятие решений в условиях неопределенности. Межвузовский научный сборник. Вып.2. Ч.1. Уфа: УГАТУ, 2005. – с. 24 – 32

4. Dorigo M., Maniezzo V., Colorni A. The ant system: Optimization by a colony of cooperating agents // IEEE Transactions on Systems, Man and Cybernetics, 1996. – № 26. – pp. 29 – 41

5. Кузнецов В.Ю. Задача покрытия и ее приложения // Интеллектуальные системы обработки информации и управления: сб. ст. рег. зимн.шк.-сем. Аспирантов и молодых ученых, Т.1, Уфа: УГАТУ, 2006. – с. 151 – 172

6. Фроловский В.Д. Автоматизация проектирования управляющих программ тепловой резки металла на оборудовании с ЧПУ // Информационные технологии в проектировании и производстве, 2005. – № 4. – с. 35 – 51

7. Кузнецов В.Ю. Задачи покрытия ортогональных многоугольников с запретными участками. // Вестник УГАТУ. Серия: «Управление, вычислительная техника и информатика», Т. 10, №2(27). Уфа: УГАТУ, 2008. – с. 177 – 182