

ПРИМЕНЕНИЕ ЭВРИСТИЧЕСКОГО АЛГОРИТМА РОЯ ЧАСТИЦ В РЕШЕНИИ ЗАДАЧИ ГЕОМЕТРИЧЕСКОГО ПОКРЫТИЯ

Гужва Антон Денисович

Новосибирск, Новосибирский Государственный Технический Университет, 09.04.01 Информатика и вычислительная техника

Представляется описание метода нахождения оптимального геометрического покрытия кругами ортогональных многоугольников с запретными участками. Метод основан на использовании гексагональной эвристики покрытия кругами и оптимизационном методе роя частиц. Проанализированы результаты выполнения и временные затраты, показано практическое применение данного метода, обозначено направление дальнейшего исследования.

Проблема покрытия большой области наименьшим количеством объектов определённого размера возникает в самых разных областях человеческой деятельности. При проектировании системы сотовой связи очень важно знать наиболее оптимальное расположение приёмо-передающих вышек. Правильное размещение датчиков пожарной сигнализации и видеокамер кругового обзора играют большую роль при создании систем безопасности. В сельском хозяйстве данная проблема возникает при проектировании системы полива. Во всех этих случаях для оценки и первичного анализа при разработке системы разумно поставить и решить задачу геометрического покрытия плоскости кругами. Нахождение алгоритмов решения данной задачи является актуальной проблемой.

В общем виде задачу можно описать следующим образом: имеется произвольная фигура на плоскости, которую нужно покрыть наименьшим количеством объектов так, чтобы не осталось ни одного непокрытого участка. Для её изучения будет использоваться конкретная разновидность задачи, в которой фигура является прямоугольником, внутри которого расположено некоторое количество запретных зон, а в роли объектов будут выступать круги равного радиуса. К прямоугольной форме или к форме ортогонального многоугольника, которую можно получить с помощью верной расстановки запретных зон, можно с определённой точностью свести практически любую фигуру произвольной формы, в зависимости от требуемого решения. Важно понимать, что обычно решение данной задачи не является конечным, а служит для первичной оценки при решении других задач, поэтому находить максимально точное решение, как правило, не требуется.

Постановка задачи звучит следующим образом. Дана прямоугольная область A с шириной W и длиной L . Для удобства вычислений вводится прямоугольная система координат с началом в левом нижнем углу и расположением осей Ox и Oy вдоль нижней и левой стороны соответственно. Внутри области A существует множество Z прямоугольных запретных зон, при этом стороны запретных зон параллельны сторонам области. $Z = \{Z_1, Z_2, \dots, Z_m\}$, где $Z_i = \langle z_x^i, z_y^i, z_w^i, z_l^i \rangle$, $i = 1, \dots, m$; Здесь z_x^i и z_y^i – координаты левого верхнего угла i -й запретной зоны, z_w^i и z_l^i – соответственно ширина и длина i -й запретной зоны. Также задан радиус r покрывающих кругов. Требуется покрыть область $A \setminus Z$ наименьшим числом кругов заданного радиуса.

Решение задачи можно представить в виде набора данных $R = \langle N, X, Y \rangle$, где N – количество покрывающих кругов, X и Y – векторы координат центров кругов. Тогда R будет допустимым покрытием, если выполняются следующие условия:

- Центры кругов располагаются внутри прямоугольника A :
 $\forall j = 1, \dots, N \quad x_j \geq 0; y_j \geq 0; x_j \leq L; y_j \leq W;$
- Ни один центр круга не располагается внутри запретных зон Z :
 $\forall j = 1, \dots, N \text{ и } \forall i = 1, \dots, m \quad (x_j - z_x^i)(x_j - z_x^i - z_l^i) \geq 0 \text{ или } (y_j - z_y^i)(y_j - z_y^i - z_w^i) \geq 0$
- Покрыта вся область $A \setminus Z$:
Если $(p_x; p_y)$ – произвольная точка на области $A \setminus Z$, то $\exists j: (p_x - x_j)^2 + (p_y - y_j)^2 \leq r^2$

Пример допустимого покрытия показан на рис. 1.

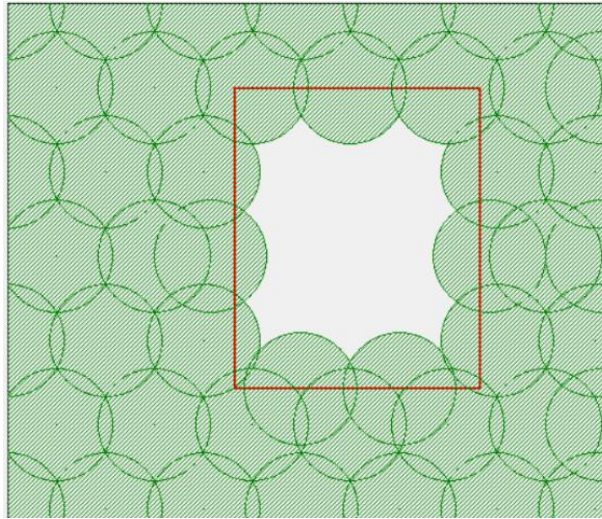


Рис. 1. Допустимое покрытие кругами прямоугольника с запретной зоной

Допустимое покрытие будет оптимальным, если число кругов N будет минимальным. К этому также можно добавить следующие условия: площадь пересечения кругов должна быть минимальной и площадь участков кругов, входящих в запретные зоны и выходящих за пределы области A должна быть минимальной.

Классическим методом решения задачи покрытия плоскости является «гексагональное покрытие». В этом покрытии центры кругов размещаются в узлах гексагональной решётки, которая строится по определённому алгоритму. При наличии запретных зон узлы, расположенные в них, объявляются запретными и удаляются, а образовавшиеся в результате этого пустые участки находят и покрываются. Таким образом, алгоритм гексагонального покрытия состоит из трёх этапов [1]:

1. Построение гексагональной решётки без учёта границ области A и запретных зон Z ;
2. Восстановление границ области и запретных зон; выявление и удаление неподходящих кругов;
3. Покрытие образовавшихся после этапа 2 пустых участков.

Структура гексагональной решётки показана на рис. 2

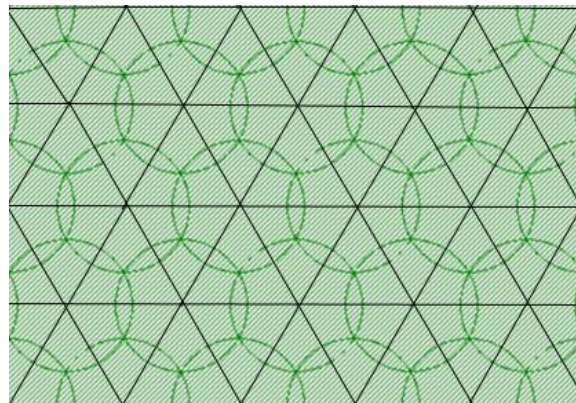


Рис. 2. Гексагональная решётка

Координаты узлов гексагональной решётки вычисляются следующим образом:

$$x_i = \begin{cases} 0, & \text{если круг – первый в чётном ряду} \\ \frac{r\sqrt{3}}{2}, & \text{если круг – первый в нечётном ряду} \\ x_{i-1} + r\sqrt{3}, & \text{если круг в том же ряду} \end{cases}$$

$$y_i = \begin{cases} y_{i-1}, & \text{если круг в том же ряду} \\ y_{i-1} + \frac{3}{2}r, & \text{если круг – первый в следующем ряду} \end{cases}$$

Здесь i – номер круга, r – радиус круга. Координаты первого круга вычисляются отдельно.

После удаления из решётки узлов, которые выходят за пределы области A или входят в запретные зоны, в покрытии образуются пустые участки, которые необходимо покрыть. Процесс поиска пустых участков достаточно трудоёмкий, поэтому его можно заменить поиском его границ [1].

Граница Γ пустых участков состоит из дуг окружностей и отрезков прямых, образованных кругами, границами запретных зон и области A . Для поиска Γ нужно определить непокрытые дуги и отрезки. Поиск дуг происходит по следующему алгоритму (поиск отрезков происходит аналогично):

1. Поиск покрытых дуг, образованных пересечением кругов с границей A ;
2. Поиск покрытых дуг, образованных пересечением кругов с границами Z ;
3. Поиск покрытых дуг, образованных пересечением кругов с другими кругами;
4. Объединение пересекающихся покрытых дуг;
5. Выделение из каждой окружности $C \setminus CA$, где C – окружность, CA – множество покрытых дуг окружности.

Чтобы покрыть непокрытую дугу, необходимо вычислить координаты центра круга, который бы её покрыл. После того, как координаты получены, нужно пересчитать границу Γ в соответствии с наличием нового круга. Данный процесс повторяется до тех пор, пока Γ не станет пустым. На выходе алгоритма получается решение R , допустимое для выполнения задачи.

В то же время, это решение нельзя с полной уверенностью назвать оптимальным. Во-первых, полученное решение в значительной степени зависит от координат начальной точки гексагональной решётки, по которой она и строится. На рис. 3 показаны две гексагональные решётки на втором этапе выполнения алгоритма с разными начальными точками. Можно увидеть, что у второй решётки граница Γ имеет значительно меньшую длину, а значит, кругов на её покрытие потребуется меньше, что непосредственно влияет на оптимальность решения. Во-вторых, используемый алгоритм покрытия непокрытых дуг имеет заметный недостаток – он может покрывать только дуги, размер которых не превышает $\pi/3$. Если дуга имеет больший размер, её приходится разбивать на несколько меньших, что отрицательно сказывается на оптимальном результате. Наилучшее расположение начальной точки, а также координаты покрывающих непокрытые участки кругов, можно найти, решив оптимизационную задачу одним из эвристических методов. Для работы на плоскости хорошо подходит метод роя частиц.

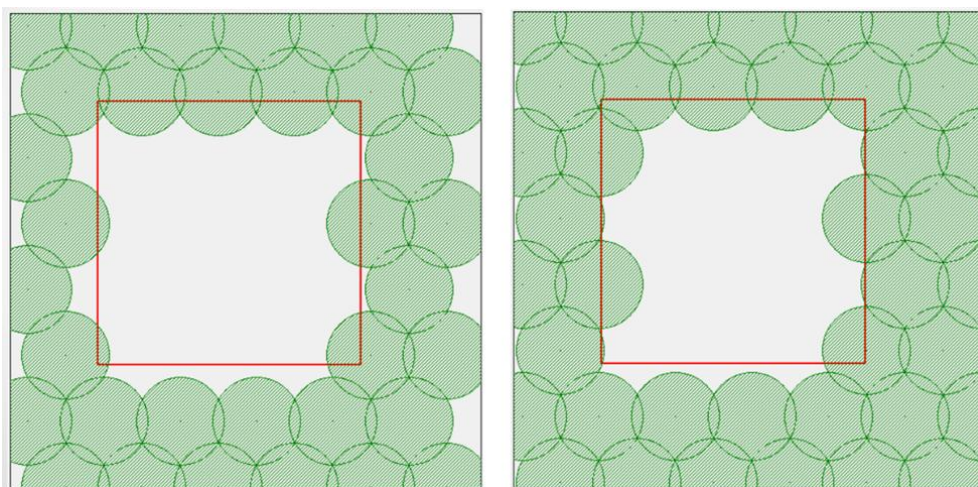


Рис. 3. Сравнение гексагональных решёток с разными начальными точками на втором этапе алгоритма

Метод роя частиц работает с множеством решений-кандидатов, последовательно улучшающихся путём перемещения их по пространству решений и обмена информации между ними. В контексте задачи покрытия решение-кандидат (частицу) удобно обозначить как $p_i = \langle x, y, R \rangle$, $i = 1, \dots, S$, где S – размер роя, x и y – координаты начальной точки решётки, а R – решение, с которым ассоциируется частица. Данное решение может быть индивидуальным для каждой частицы или общим, если оптимизируется покрытие пустых участков.

Каждая частица роя характеризуется набором двумерных векторов $\langle x_i, v_i, l_i, g \rangle$, где:

- x_i – положение частицы;
- v_i – скорость частицы, под действием которой она перемещается;
- l_i – локально лучшее решение, т.е. наилучшее из известных положений частицы i ;
- g – глобально лучшее решение, т.е. наилучшее известное состояние роя в целом.

Все эти векторы имеют одинаковую размерность и, в контексте задачи, представляют собой координаты точки на плоскости (x и y).

Работа алгоритма роя частиц настраивается с помощью ряда коэффициентов, влияющих на перемещение частиц в пространстве решений:

- c_1 – когнитивный коэффициент, определяющий степень влияния локально лучшего решения на скорость;
- c_2 – социальный коэффициент, определяющий степень влияния глобально лучшего решения на скорость;
- $\chi = \frac{2}{c-2+\sqrt{c^2-4c}}$ – коэффициент стягивания, предназначенный для того, чтобы не допустить чрезмерного разлёта частиц (здесь $c = c_1 + c_2$) [3];
- ω – коэффициент инерции, позволяющий изменять баланс между тщательностью исследования пространства и сходимостью [4].

В зависимости от задачи, может быть использован один из двух коэффициентов χ или ω . Совместное их использование возможно, но может привести к последствиям, не исследованным до конца.

Алгоритм роя частиц состоит из двух стадий: инициализация и вычисление. Инициализация задаёт начальные значения для всех частиц роя. Для отдельно взятой частицы i она выглядит следующим образом:

1. Получить начальное положение частицы случайным образом с помощью равномерного распределения;
2. Присвоить локально лучшему решению значение начального положения частицы;
3. Если локально лучшее решение лучше текущего глобально лучшего решения, то обновить его значением локально лучшего решения частицы;
4. Получить начальную скорость частицы случайным образом с помощью равномерного распределения.

В роли критерия, оценивающего качество решения можно выбрать её количество кругов либо площадь пересечения кругов друг с другом, с запретными зонами и границами области A . Также могут быть использованы оба эти критерия.

Вычисление происходит итерационно до тех пор, пока не будет выполнено условие. Этим условием может быть достижение определённого количества итераций либо достижение определённого значения целевой функции. Это значение может быть получено оценочно по формуле $N' = \frac{S_A K}{S}$, где S_A – площадь фигуры $A \setminus Z$, $K = 1,22$ – коэффициент, учитывающий перекрытие кругов в гексагональном покрытии, S – площадь круга [1]. Рационально выбрать оба этих условия, так как оценочное значение N' не является окончательным решением.

На итерации t алгоритм обновляет значения частицы i следующим образом:

1. Получить значения случайных величин r_1 и r_2 , равномерно распределённых в промежутке от 0 до 1;
2. Обновить скорость частицы по формуле:
 $V_i^t = \chi(\omega V_i^{t-1} + c_1 r_1 (x_i^{t-1} + l_i^{t-1}) + c_2 r_2 (x_i^{t-1} + g^{t-1}));$
3. Обновить положение частицы в пространстве:
 $x_i^t = x_i^{t-1} + V_i^t;$
4. Если новое положение частицы лучше текущего локально лучшего значения, то обновить его;
5. Если локально лучшее положение частицы лучше текущего глобально лучшего значения, то обновить его.

По окончании вычисления g будет являться искомым решением.

Оценка решений происходит по-разному, в зависимости от того, к чему применяется данный алгоритм. При нахождении наилучшей начальной точки оптимизация идёт по пути минимизации числа покрывающих кругов, а также площади пересекающихся участков кругов. Таким образом, каждая частица содержит отдельное допустимое решение и координаты начальной точки для этого решения. Подход к оптимизации покрытия пустых участков несколько другой. Положение частицы обозначает координаты центра круга, который должен покрыть пустой участок с максимальной эффективностью, т.е. минимизировать длину оставшейся границы G . При обоих подходах сложно оценить нижнюю границу решения, поэтому в качестве критерия остановки используется только количество итераций.

Для оценки качества разработанных алгоритмов проведен ряд численных экспериментов на задачах с различными параметрами. Для проведения экспериментов была выбрана область A в виде квадрата со

стороной 500 пикселей. Количество запретных зон Z варьировалось от 1 до 3. Их размеры задавались случайным образом так, чтобы их общая площадь составляла 30, 50 и 70 процентов общей площади области A . Задачи разделены на 9 классов по числу препятствий и занимаемой ими площади. Для каждой задачи исследовалось 10 конфигураций запретных зон. Для покрывающих кругов были взяты радиусы 30 и 40 пикселей.

Коэффициенты работы роя частиц задавались случайным образом: для инерциального коэффициента от 0.8 до 1.2 и для когнитивного и социального коэффициента от 1.5 до 3, причём их сумма, в соответствии с условием работы алгоритма, не меньше 4. В каждом решении использовалось 20 частиц, вычисление длилось 50 итераций.

Примеры покрытий изображены на рисунках 4 и 5. Границы области A обозначены чёрным цветом, границы запретных зон – красным цветом.

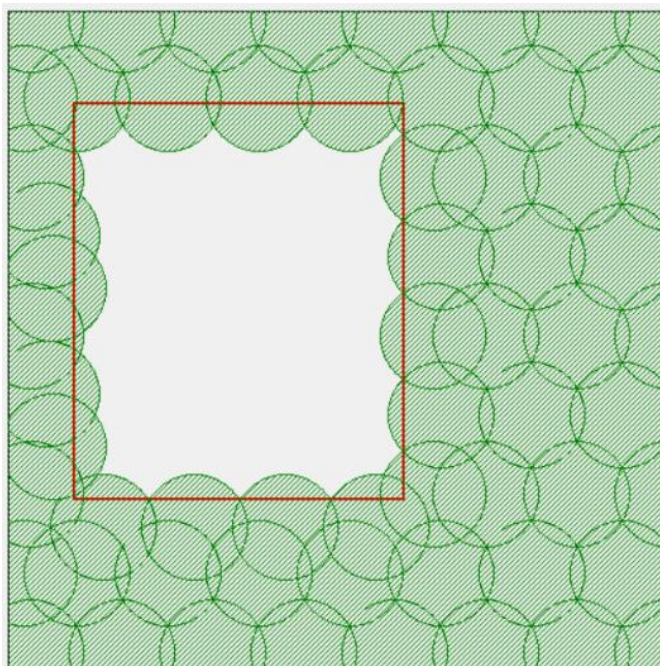


Рис. 4. Пример решения задачи класса №1 с оптимизированным покрытием пустых участков (круги радиуса 40 пикселей)

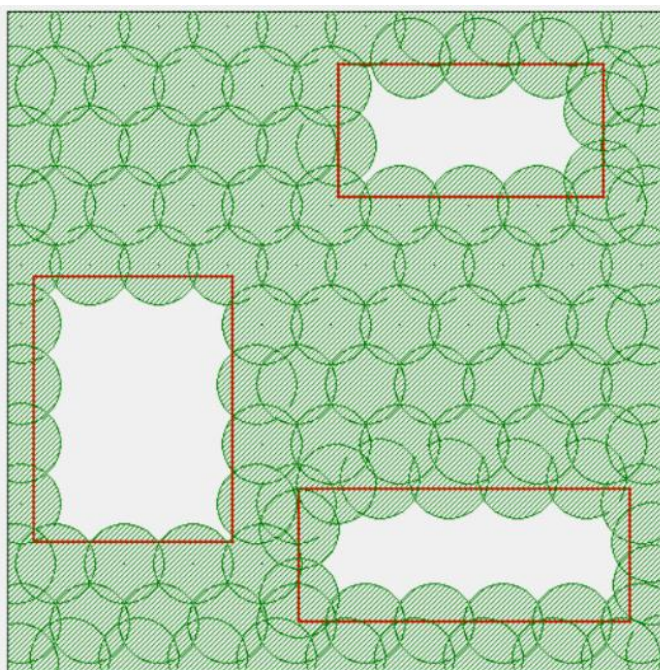


Рисунок 3.3 – Пример решения задачи класса №7 с оптимальной начальной точкой (круги радиуса 30)

Результаты экспериментов приведены в таблицах 1 и 2 для кругов с радиусом 40 и 30 пикселей соответственно. Обозначения в таблицах: N_z – количество препятствий в задаче, F – площадь свободного пространства, где могут размещаться центры покрытых кругов, в процентах от общей площади, n_1 – среднее количество кругов в решении, полученном неоптимизированным гексагональным алгоритмом, n_2 – среднее количество кругов в решении с оптимальной начальной точкой, n_3 – среднее количество кругов в решении с оптимизированным покрытием пустых участков, n_4 – среднее количество кругов в решении, оптимизированном обоими алгоритмами.

Таблица 1

№	N_z	F, %	n_1	n_2	n_3	n_4
1	1	70	64	61	63	60
2	2	70	62	59	60	59
3	3	70	71	68	67	66
4	1	50	55	50	51	50
5	2	50	57	55	54	54
6	3	50	65	63	59	60
7	1	30	46	44	43	43
8	2	30	48	47	46	46
9	3	30	53	49	51	48

Таблица 2

№	N_z	F, %	n_1	n_2	n_3	n_4
1	1	70	113	102	106	102
2	2	70	100	99	100	99
3	3	70	117	114	113	110
4	1	50	93	89	86	84
5	2	50	98	88	96	86
6	3	50	98	95	95	91
7	1	30	70	66	66	64
8	2	30	72	69	70	68
9	3	30	77	69	72	68

Из полученных результатов можно увидеть, что оптимизация гексагонального алгоритма даёт лучший результат в каждом случае. Оптимизация начальной точки уменьшает количество кругов в покрытии в среднем на 4.8-5.4%, при этом этот процент тем больше, чем меньше соотношение радиуса круга к площади покрытия. Оптимизация покрытия пустых участков без оптимизации начальной точки улучшает решение, но зачастую не так сильно, как оптимизация начальной точки. В среднем количество кругов в покрытии уменьшается на 4.1-5%, это число зависит только от конфигурации запретных зон на области покрытия. Решение, полученное и с помощью оптимизации начальной точки, и с помощью оптимизации покрытия пустых участков, как правило, является наилучшим. По сравнению с неоптимальным решением количество кругов уменьшается на 6.7-7.8%. Однако следует отметить, что такое решение не всегда является лучше, чем решение, полученное только одним оптимизированным алгоритмом, иногда такие решения одинаковы.

Если изучить влияние алгоритма с оптимальным покрытием пустых участков, можно увидеть, что во многих случаях не наблюдается значительного улучшения решения. Особенно это касается применения этого алгоритма к решению с оптимальной начальной точкой. Выгода в этих случаях может либо отсутствовать, либо составлять около 1% от общего числа кругов, при этом затраты вычислительных ресурсов и, соответственно, времени гораздо выше. В связи с этим его использование следует рассматривать не во всех случаях.

Нужно добавить, что разработанный алгоритм оптимизированного гексагонального покрытия требует значительных вычислительных и, как следствие, временных затрат, которые на несколько порядков больше, чем классический гексагональный алгоритм. При использовании многоядерных процессоров или вычислительных кластеров, время выполнения может значительно снизиться, так как вычисление частиц идёт независимо друг от друга.

Дальнейшие исследования по данному вопросу заключаются в модификации используемого алгоритма оптимизации, а именно: подбор оптимальных параметров, применение других разновидностей метода, таких как полноинформированный метод роя частиц, применение топологии соседей для обмена

информацией. Кроме того, нужно исследовать возможное применение алгоритма к областям с произвольной формой. Сложность в этом случае будет с определением границы Γ , при этом сам алгоритм не требует серьезных изменений. Следует также помнить, что данный алгоритм может быть использован для оценки при решении задач в разных областях, таких как связь, безопасность, астрономия, строительство и другие, однако практическую пользу от его применения можно получить не во всех случаях.

Список использованных источников

1. Кузнецов В.Ю. Методы покрытия многосвязных ортогональных многоугольников для задач оптимального размещения сенсоров в области мониторинга: диссертация канд. тех. наук. Уфим. гос. авиац.-тех. университет, Уфа, 2009.
2. J. Kennedy Particle swarm optimization / J. Kennedy, R. C. Eberhart // Proc. of IEEE International Conference on Neural Networks Vol. IV. – Piscataway, 1995. – P. 1942 – 1947.
3. M. Clerc, The particle swarm – explosion, stability, and convergence in a multidimensional complex space [текст]/M. Clerc, J. Kennedy // IEEE Transactions on Evolutionary Computation. – 2002. – №6 (1). – P. 58 – 73.
4. Y. Shi, A modified particle swarm optimizer / Y. Shi, R. C. Eberhart // Proc. of IEEE International Conference on Evolutionary Computation. – Piscataway, 1998. – P. 69 – 73.