электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.023, 519.147

Особенности адаптации алгоритмов двумерной ортогональной упаковки

***Рыжиков А.О.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
Кафедра «Компьютерные системы автоматизации производства»

Научный руководитель: Хоботов Е.Н., д.т.н., профессор,
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
gss@bmstu.ru

Введение

В настоящее время на многих промышленных предприятиях достаточно часто используются алгоритмы двумерной ортогональной упаковки. Эти алгоритмы позволяют размещать прямоугольные элементы в прямоугольном контейнере таким образом, чтобы рабочая область была заполнена по возможности максимально. Подобная задача является NP-полной [1], и для ее решения чаще всего используются различные приближенные методы. Один из таких методов представлен в статье [2].

В рамках данной статьи предлагается алгоритм, основанный на идеях и принципах построения методов двумерной упаковки для решения задач по заполнению вибростолов и площади формовочного цеха на заводах железобетонных изделий с учетом особенностей их работы.

Постановка задачи

Пусть имеется прямоугольный контейнер C_0, имеющий длину W_0 и ширину H_0 соответственно, в котором требуется разместить n прямоугольных форм для изготовления изделий R_i, имеющих длину w_i, ширину h_i. Каждое изделие находится на столе определенное количество времени t_i. В задаче требуется разместить в контейнере наибольшее количество изделий из заданного множества R_i так, чтобы:

- оставшаяся свободной от изделий площадь контейнера была бы минимальной;
- суммарное время занятости контейнера этим набором изделий было бы минимально.

Накладываемые *ограничения* для этой задачи имеют вид:

1. Каждый край изделия должен быть параллелен краю контейнера.
2. Изделия не должны перекрывать друг друга.
3. Изделия не могут находиться на одном месте в одно время.

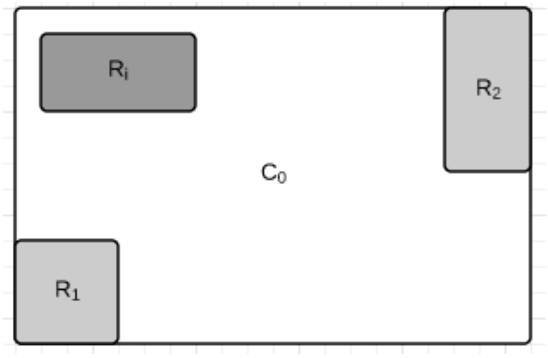

Рис. 1. Варианты СОА в контейнере

При выполнении задачи принимаются определенные *допущения*:

1. Все изделия прямоугольной формы.
2. Вероятность появления брака любого рода равна нулю.
3. Расстояние между изделиями не учитывается.
4. Время на установку и удаление изделия из контейнера не учитывается.
5. Время обработки изделия больше нуля.

Эвристический поиск

В качестве базы для разработки нового алгоритма используется описанные в статье [2] квазичеловеческий эвристический алгоритм, а также понятие «главной точки».

Согласно [2], «в квазичеловеческом эвристическом методе заполнения полостей процесс упаковки рассматривается как имитация деятельности человеческой логики. Эвристическим метод на основе заложенных в него правил, указывает путь к конечной цели алгоритма (заполненный контейнер) и определяет, каким образом лучше расположить в контейнере каждую форму изготавливаемого изделия». «Главной точкой», согласно [2], является точка на поверхности контейнера, определяющей положение детали на столе. В дальнейшем, изменив терминологию [2], однозначное положение, определенное с помощью главной точки, будем называть Common Orderable Action, сокращенно – СОА.

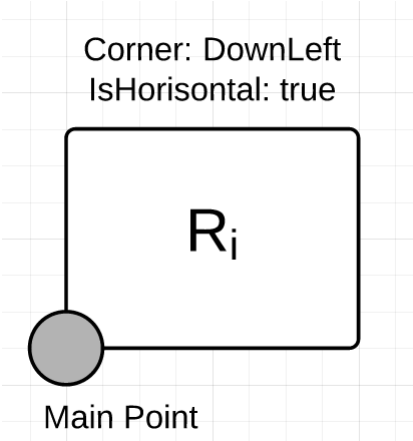

Рис. 2. Представление и основные параметры СОА

Таким образом, блок-схема размещения, согласно [2], выглядит следующим образом:

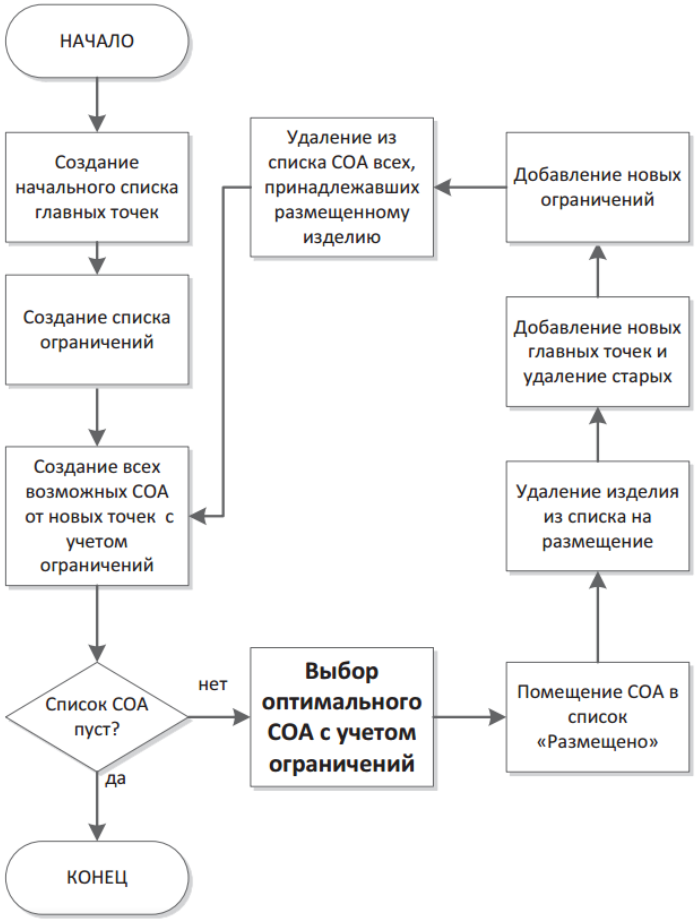

Рис. 3. Приближенная блок-схема работы алгоритма из [2]

Для повышения эффективности заполнения контейнера формами имеет смысл использовать несколько правил. Более того, правила можно компоновать и изменять в

процессе работы алгоритма. Кроме того, во многих случаях хорошие результаты дают [4] правила, основанные на минимизации размеров полостей между уже размещенными формами изделий. Цель этих правил – максимально заполнить пустоты, образуемые в процессе размещения форм.

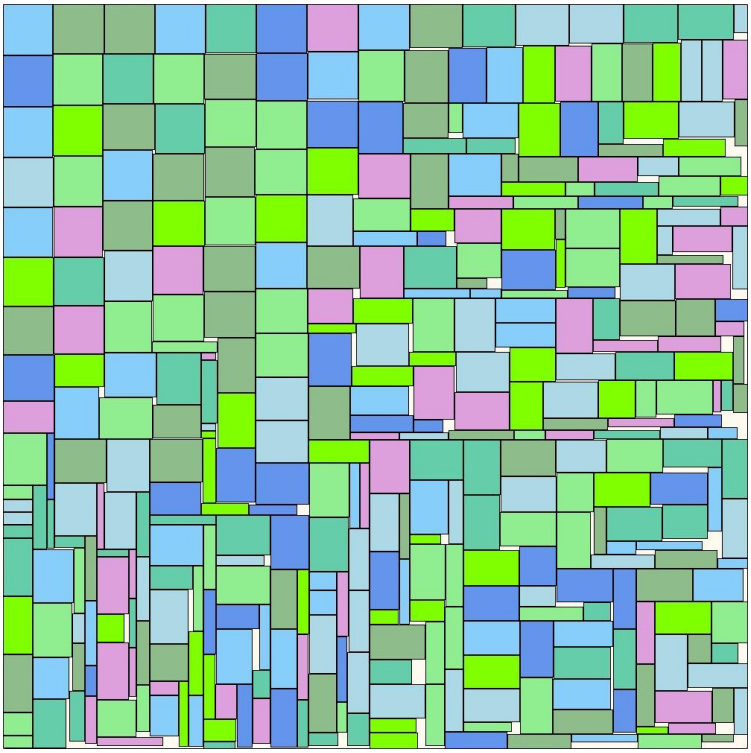

Рис. 4. Результат работы алгоритма по правилу минимизации нишевых полостей

Несмотря на все плюсы алгоритмов, основанных на размещении относительно главных точек, они недостаточно гибки в тех условиях, когда необходимо быстрое размещение. Потерю в скорости можно компенсировать последовательным использованием сначала примитивных алгоритмов, быстро создающих солидный процент заполнения. После работы такого рода алгоритмов целесообразно использовать вышеописанные эвристики, чтобы заполнить образовавшиеся пустоты.

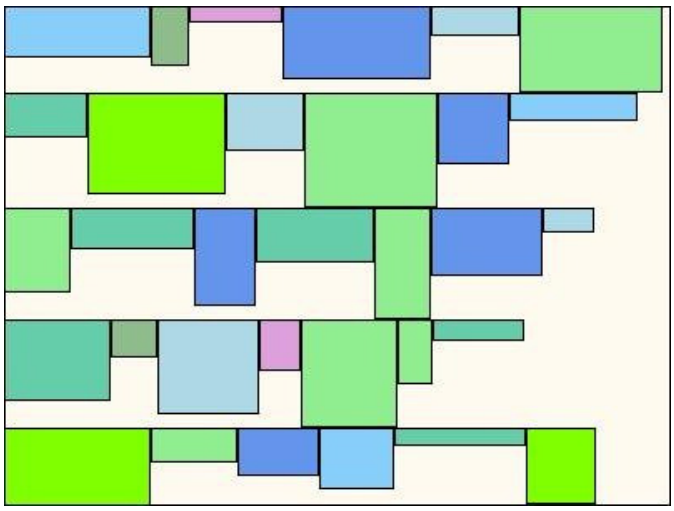

Рис. 5. Результат работы примитивного алгоритма

Стоит отметить, что в описанном выше алгоритме еще не рассматривалась задача динамического размещения форм изделий. В реальном производстве формы изделия убираются и помещаются на стол в процессе его работы. Имеет смысл так размещать формы изделий, чтобы процесс удаления ка с вибростолов, так из цеха формовки происходил с формами изделий, близко расположенными друг к другу. Для этого в алгоритм необходимо внести некоторые коррективы. Введем несколько определений.

Таймлайн (англ. timeline, «линия времени») − шкала, относительно которой происходит отсчет времени размещения форм.

Отметка (англ. timestamp) − определенный момент времени на таймлайне, служащий для учета размещения форм изделий. Отсчетная точка таймлайна также является отметкой. Отметка возникает в момент размещения форм изделия на столе и характеризуется окончанием размещения изделия. Например, отметка 2 на рисунке 6 образована от изделий 1 и 2, отметка 3 – от изделий 3, 4, 5 и т.д.

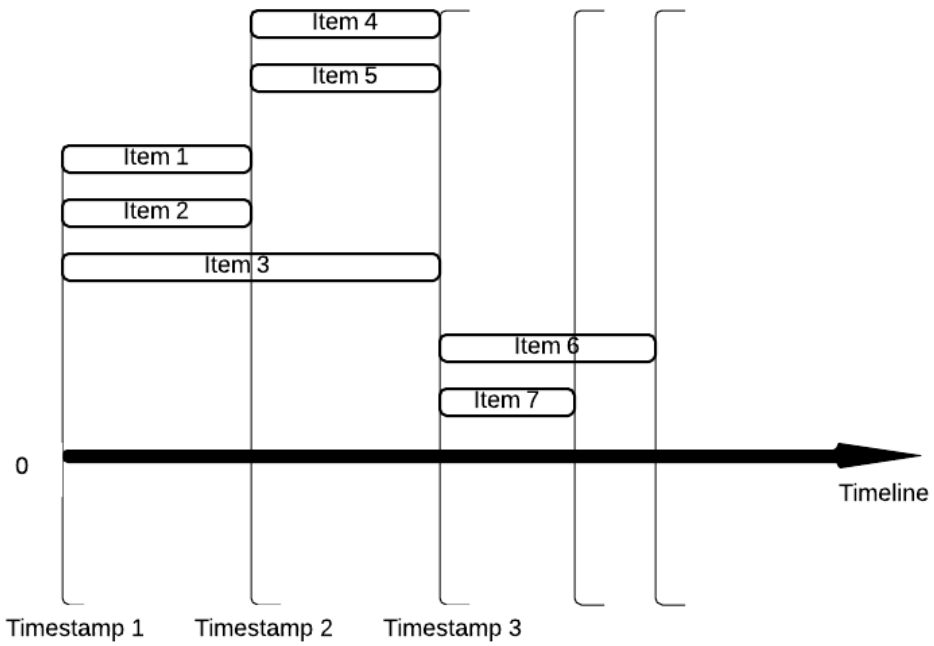

Рис. 6. Организация учета времени

Таким образом, предложенная на рис. 3 блок-схема нуждается в адаптации. Опуская уже описанные на рис. 3 детали, алгоритм может быть изменен согласно рис. 6.

Если в данный момент времени время приближается к намеченной отметке, удаляются формы изделий, размещенные до достижения отметки, и попробуем на освободившихся площадях разместить новые. Особое внимание обращается на тот факт, что порой возможность размещения не является целесообразной из-за сложности транспортировки форм, и поэтому необходимы простые правила, позволяющие определить необходимость размещения изделия. Когда произошло размещение, проверяется, есть ли возможность и необходимость размещения следующих изделий. Если больше изделий разместить нельзя, то выполняется переход к следующей отметке. Такой процесс повторяется до тех пор, пока временные отметки не закончатся.

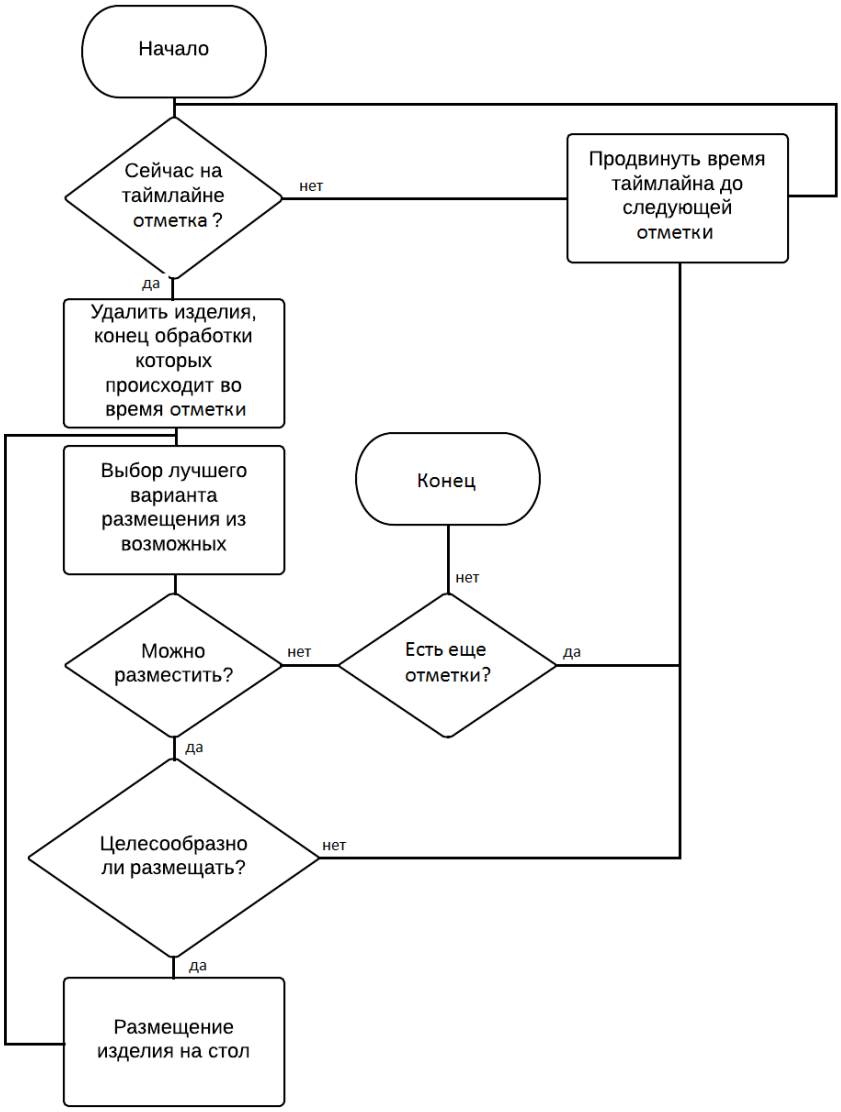

Рис. 6. Блок-схема алгоритма с учетом времени

В связи с особенностями работы алгоритма, а именно в необходимости добавления новых правил размещения, архитектура системы должна быть гибкой. Как следствие, система должна удовлетворять принципам гибкой разработки программного обеспечения и применять необходимые паттерны проектирования.

Поэтому в процессе разработки использовалась методология *TDD* (Test-Driven Development), основанная на написании модульных тестов для описания работы программы, а также обеспечивающая полное покрытие системы тестами.

Для удобного внедрения иных алгоритмов использовался паттерн стратегии. Сам модуль, описанный в этой статье, практически устроен как библиотека, позволяющая с помощью набора простых в использовании методов получать расчеты, которые с легкостью

можно интегрировать в форматы *XML* или *JSON* для вывода данных в обычное приложение или веб-страницу. Формат библиотеки позволяет легко расширить алгоритмы в случае необходимости и обеспечивает стабильную работу без глубокого погружения в архитектуру программы.

Описанный в статье модуль является частью разрабатывающейся системы управления заводом железобетонных изделий. Система основана на клиент-серверной архитектуре и располагается на сервере, на котором расположена база данных предприятия. Клиент же получает данные о доступных в данный момент времени изделиях и посылает запрос на сервер, где происходят алгоритмы упаковки.

Дальнейшие перспективы

В дальнейших разработках планируется добавить эвристики алгоритмы размещения форм изделий, разработать интерфейсы, позволяющие пользователям выбирать и комбинировать эвристики для достижения наилучшего результата. Имеет смысл модернизация представленного метода учета времени, и разработка правил, которые способны подстраиваться под эти особенности технологического процесса. Сам учет времени также планируется улучшить, добавив привязку существующей статистики к предлагаемому размещению. Кроме того, предполагается проводить работы над совершенствованием алгоритмов, обеспечивая их возможностью возвращения на несколько шагов назад, чтобы на основе анализа статистики можно было менять эвристику в процессе работы.

Заключение

Несмотря на большое количество вариаций алгоритмов ортогональной упаковки, порой возникает необходимой серьезной адаптации алгоритмов под реальные нужды автоматизации реальных процессов. Данный алгоритм легко расширяем и, по сути, представляет из себя базу для применения других правил, позволяющих успешно внедрять модули, адаптированные под различного рода задачи. Автор надеется, что гибкая методология разработки и продуманная архитектура продукта позволит создавать легко поддерживаемые и адаптируемые программные комплексы для успешной автоматизации производственных процессов.

Список литературы

1. Gary M.R., Jonson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. A Series of Books in the Mathematical Sciences. W.H. Freeman and Co., 1979. 338 p.

2. Рыжиков А.О. Алгоритм двумерного размещения форм в контейнер // Молодежный научно-технический вестник. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 11. Режим доступа: http://sntbul.bmstu.ru/doc/635531.html (дата обращения 20.03.2014).
3. Брукс Ф. Мифический человеко-месяц, или как создаются программные системы. СПб.: Символ-Плюс, 2013. 304 с.
4. Djuabing Chen, Venki Chuan. A Novel Quasi-human Heuristic Algorithm for Two-dimensional Rectangle Packing Problem // IJCSNS International Journal of Computer Science and Network Security. 2006. Vol.6. No.12. Pp. 115-119.