О приближенном решении 3-х мерной задачи об упаковке на основе эвристик

В. В. Псиола

1. Введение

Всякий раз, когда требуется разместить товары на складе или загрузить их в транспортное средство, мы сталкиваемся с проблемой сложить груз так, чтобы он занимал как можно меньше места, или, иными словами, с задачей об оптимальной упаковке, так называемой 3-х мерной задачи о рюкзаке.

В математической литературе чаще встречается одномерный вариант задачи об упаковке. Было установлено, что эта задача является NP-полной [1]. При постановке задачи об упаковке в трехмерном пространстве, она будет заведомо сложнее одномерной, но по-прежнему может быть решена за полиномиальное время на недетерминированных вычислительных устройствах. Следовательно, трехмерная задача об упаковке является NP-полной.

К сожалению, все известные алгоритмы, решающие какую-нибудь из NP-полных задач и, в частности, задачу об упаковке, имеют экспоненциальную временную сложность [2], [3], что эквивалентно полному перебору всех возможных вариантов. Это означает, что даже при небольшом количестве грузов (около 50) время работы программы, реализующей такие алгоритмы, будет измеряться годами даже при использовании современных суперкомпьютеров. Другой проблемой, которая не позволяет применить классические алгоритмы на практике является набор дополнительный ограничений, которые должны быть выполнены при расстановке груза в заданном объеме. В зависимости от сферы деятельности организации, для которой требуется решение задачи расчета оптимальной упаковке, набор таких ограничений и дополнительных условий может быть очень разнообразным.

В большинстве случаев, алгоритмы, предлагающие точное решение задачи с трудом могут быть адаптированы для учета дополнительных ограничений.

Выходом из этой ситуации может послужить создание приближенных эвристических алгоритмов, которые за приемлемое время находят решение, близкое к оптимальному и при этом могут быть легко адаптированы для учета дополнительных ограничений.

В данной работе будет описан подобный алгоритм, который положен в основу действующей программной системы packer3d, предназначенной для расчета оптимальной укладки грузов в транспортные средства. Плотность заполнения в результате работы этого алгоритма составляет в среднем 80–90% от объема грузового отсека, а время работы для сотен ящиков — несколько минут. При этом алгоритм имеет высокую степень адаптивности к дополнительным ограничениям, возникающим на практике. Многие из этих ограничений, такие как грузоподъемность, максимальное давление на верхнюю грань ящика, сбалансированность давления на оси транспортного средства и другие, уже учтены в алгоритме и могут быть заданы при нахождении решения. Описанная программная система представлена в Интернете, на сайте www.packer3d.ru в виде бесплатного сервиса по расчету оптимальной укладки груза.

2. Постановка задачи

При решении задачи расчета оптимальной упаковки груза в транспортные средства или просто некоторый объем (например, складское пространство) на практике, возникает большое количество постановок задачи и дополнительных условий на укладку груза, которые должны быть соблюдены. Можно выделить наиболее простую (элементарную) постановку задачи, на решении которой базируется общее решение. Элементарную постановку задачи можно сформулировать следующим образом:

Дано:

1) K типов 3-х мерных ящиков (параллелепипедов), каждый из которых задан линейными размерами (s_1, s_2, s_3);

2) груз, представленный определенным количеством ящиков каждого типа $(N_1, N_2, \ldots, N_K)$;
3) 3-х мерный объем для укладки груза, заданный линейными размерами (v_x, v_y, v_z).

Требуется рассчитать точное положение ящиков в заданном объеме таким образом, чтобы его заполнение грузом было допустимым и наиболее эффективным по объему, то есть суммарный объем поместившихся ящиков должен быть максимальным из всех возможных. Допустимым положением груза считается такое, когда все упакованные ящики находятся внутри заданного объема, не пересекаются и полностью опираются на дно объема или другие ящики.

Даже в подобной элементарной постановке задачи не существует алгоритма, который находил бы наилучшее решение за приемлемое время. Однако помимо этого на практике требуется учесть ряд дополнительных ограничений на положения. Набор этих ограничений определяется спецификой той или иной организации, а так же областью применения системы расчета оптимальной упаковки. В большинстве случаев областью применения являются информационные системы логистики, обеспечивающие оптимизацию перевозки и хранения груза.

3. Общая идея алгоритма

Общая идея алгоритма решения элементарной задачи укладки (см. 2) заключается в следующем (см. рис. 1):

1) данные для укладки (список ящиков) передаются алгоритму комбинирования ящиков в блоки;
2) набор ящиков комбинируются во всевозможные прямоугольные блоки, блоком называется набор ящиков, поставленных рядом, так что они образуют параллелепипед без щелей и пустот и представляющий собой аналог ящика с другими размерами (см. 4);
3) сформированный набор блоков передается в алгоритм 3-х мерной укладки, который состоит из 3 частей:

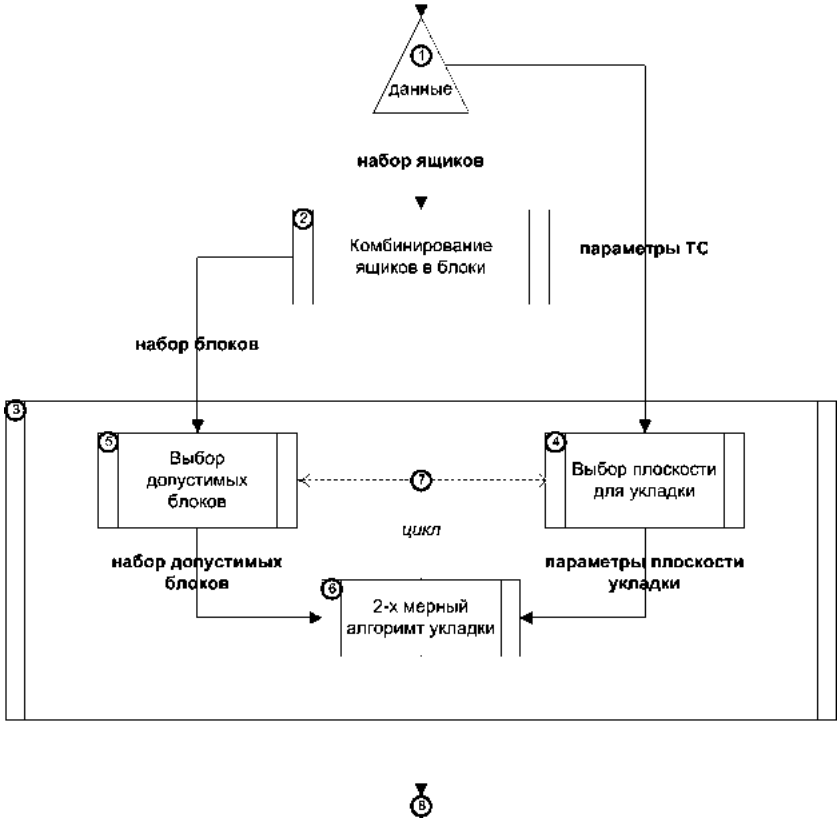

Рис. 1. Схема алгоритма.

а) первая часть выбирает плоскость для укладки ящиков, выбор плоскости для укладки является сведением задачи 3-х мерной укладки к задаче 2-х мерной укладки;
б) вторая часть выбирает набор блоков, допустимых для укладки в текущую плоскость, при этом все эти блоки должны быть одной высоты для сведения 3-х мерной задачи для укладки к 2-х мерной задаче;
в) третья часть является применением алгоритма 2-х мерной укладки к выбранной плоскости и допустимому набору блоков для укладки в эту плоскость;

4) алгоритм 3-х мерной укладки выполняется до тех пор, пока не уложены все ящики заказа и в ТС остается свободный объем для укладки ящиков;
5) после того как все ящики уложены или объем заполнен полностью заполнен расчет оптимальной укладки ящиков в ТС считается завершенным.

4. Алгоритм комбинирования предметов

Алгоритм комбинирования ящиков определяет способ формирования различных блоков из ящиков в форме параллелепипеда. Для формирования всевозможных блоков используются два более простых алгоритма:

- формирования блоков из произвольного количества однотипных ящиков одинаковой ориентации в пространстве;
- формирование блоков из двух разнотипных ящиков, грани которых совпадают по размеру.

Эти два алгоритма применяются последовательно к результату предыдущего применения, при этом из результата удаляются блоки ящиков, дублирующие друг друга. После того как последовательное применение этих шагов не образует новых блоков, приемлемых для укладки в ТС, алгоритм формирования всевозможных блоков завершает работу.

5. Алгоритм 3-х мерной задачи (сведение к 2-х мерной)

Алгоритм решения 3-х мерной задачи представляет собой последовательное решение задачи 2-х мерной укладки предметов:

1) из ящиков и блоков, сформированных на этапе 4, формируются наборы предметов одинаковой высоты, при этом учитываются все возможные положения предмета в пространстве, то

есть каждый из предметов, имеющий разные линейные размеры, войдет в 3 набора, соответствующих каждому из его линейных размеров (если они все разные);

2) сформированные наборы сортируются в порядке убывания их функционала качества (см. 9), после чего считается, что первый набор является наиболее оптимальных для укладки в ТС в текущий момент времени;
3) запускается цикл, в котором на каждом шаге в ТС упаковывается наилучший (с точки зрения функционала 9) набор предметов:
   а) в первую очередь вычисляется область укладки, в которую на данном шаге цикла должны быть помещены предметы (см. 6);
   б) выбранная область определяется некоторым плоским основанием с ломаной границей и высотой, допустимой для укладки предметов (см. 6);
   в) из отсортированного набора предметов выбирается ближайший набор, чья высота меньше или равна высоте области укладки;
   г) для выбранных области и набора предметов запускается алгоритм решения 2-х мерной задачи укладки (см. 7);
   д) после того как предметы размещены в ТС, меняется состав сформированных наборов укладки (из них исключаются уже размещенные предметы), поэтому запускается пересортировка всех наборов предметов в соответствии с убыванием функционала качества 9;
4) цикл завершается, когда в очередную область укладки не получилось поставить не одного предмета, при этом процедура расчета считается завершенной.

6. Выбор области укладки для 3-х мерного алгоритма

На каждом шаге алгоритма 3-х мерной укладки (см. 5), в уже частично заполненном ТС, необходимо выбрать некоторую область,

в которую будут упаковываться предметы одинаковой высоты с помощью алгоритма 2-х мерной укладки. Эта область должна иметь плоское основание с ломаной границей. Подобные плоскости образуются в процессе работы алгоритма 3-х мерной укладки предметов, когда ящики и блоки одинаковой высоты устанавливаются рядом и на одному уровне. Поверхность таких предметов и образует некоторую область для укладки других предметов. Таким образом подобная область укладки определяется двумя параллельными прямыми (борта ТС) и двумя прямоугольными ломаными (край установленных предметов и граница начальной области, куда они были установлены).

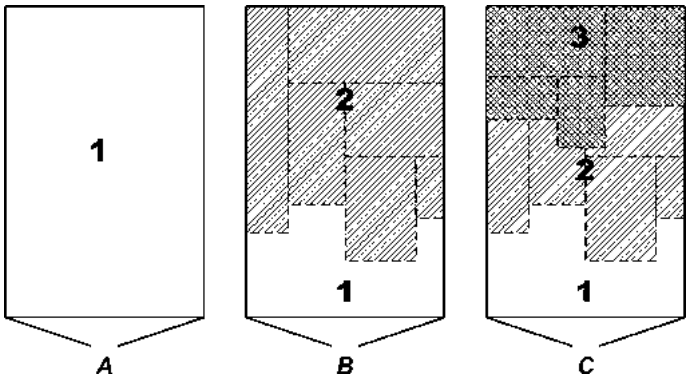

Рис. 2. Области укладки ящиков.

Алгоритм выбора такой плоскости следующий:

1) Изначально такой областью является все пространство ТС (A1 на рисунке 2).
2) После установки в эту область нескольких предметов одинаковой высоты, становятся доступными для использования 2 области с плоским основанием: та что осталась от основания (B1 на рисунке 2) и та что сформирована установленными предметами (B2 на рисунке 2).
3) В качестве следующей для укладки области выбирается область над установленными на предыдущем шаге предметами (то есть на рисунке 2 последовательно заполняются области A1, B2, C3 и т. п.)

4) В том случае, если сверху установленных предметов не удалось больше ничего поставить, выбирается оставшаяся область от предыдущего шага (C2 на рисунке 2).
5) Если не удалось заполнить и оставшуюся область, то осуществляется спуск еще на один шаг ниже (B1 на рисунке 2).

7. Алгоритм 2-х мерной укладки

Алгоритм укладки 2-х мерных предметов используется в алгоритме укладки 3-х мерных предметов 5. При этом 2-х мерный алгоритм должен вычислить оптимальную укладку 2-х мерных предметов в область, заданную двумя прямоугольными ломаными границами (внутренней и внешней). Боковые границы области определяются как прямые между концами внутренней и внешней ломаных. Ломаную линию будем называть прямоугольной, если все углы излома равны $+90^\circ$ или -90° (см. рисунок 3). Алгоритм 2-х мерной укладки заключается в выполнении следующих шагов:

1) все предметы сортируются в порядке убывания функционала качества (см. 10);
2) для каждого предмета в полученном списке выполняются следующие действия:
3) если площадь предмета больше пустой площади объема или его линейный размер больше максимального линейного размера объема для укладки, то пропускаем этот предмет и переходим у следующему;
   а) для каждой из 4-х возможных ориентаций предмета в пространстве:
   б) для каждой из 4-х вершин предмета:
   в) для каждой из вершин внутренней ломанной и для каждой проекции вершин внешней ломанной на внутренюю:
      – вычисляется новая область укладки, которая формируется из старой области и этим предметом, установленным в выбранной ориентации так, что выбранная вершина предмета совпадает с выбранной вершиной ломаной;

- если в выбранном положении предмет пересекается с какой-либо границей области, то такое положение считается не допустимым и алгоритм переходит к следующему;
- если выбранное положение предмета допустимо, то новая область укладки, сформированная этим положением, добавляется в список для анализа на шаге 3е.

г) если для данного предмета нет ни одного допустимого положения, то алгоритм переходит к следующему предмету;

д) после перебора всех возможных положений предмета, сформированные новые области укладки, которые получаются при установке предмета в каждом из таких положений, сортируются в порядке убывания функционала качества (см. 10);

е) в полученном списке областей выбирается первая (с наибольшим значением функционала качества) и она определяет наилучшую позицию выбранного предмета в данной области, именно в этой позиции предмет и размещается;

ж) очевидно, что если предметы из списка, стоящие перед текущим не смогли поместиться в предыдущую область укладки, то они не смогут поместиться и в новую, образованную этим установленным предметом, поэтому они исключается из дальнейшего перебора, а рассматриваются только следующие за текущим предметы.

4) цикл и алгоритма 2-х мерной укладки завершается по окончании перебора всех 2-х мерных предметов.

8. Использование эвристик

В различных местах алгоритма, для нахождения наилучшего решения, необходимо перебрать все возможные варианты сортировки некоторой последовательности элементов. К таким местам относятся пункт 2 алгоритма 5 и пункты 1 и 3д алгоритма 7. Однако подобный подход (полный перебор) приведет к неприемлемому долгому времени расчета. Поэтому, из некоторых эвристических соображений и на основе моделирования практических ситуаций формирует-

ся функционал качества элементов такой последовательность и все элементы сортируются в порядке убывания его значения. Отсортированные таким образом элементы последовательности и считаются единственным вариантом, которые требуется рассмотреть. В каждом случае функционал качества представляет собой взвешенную сумму некоторых характеристик элемента. Набор этих характеристик и коэффициенты определяются как раз из эвристических соображений на основе анализа различных задач и вариантов укладки предметов.

Во всех случаях интуитивно ясно, что все предложенные характеристики элементов в функционалах влияют на качество укладки, однако задача подбора коэффициентов для них является нетривиальной. Во-первых, сложно оценить в совокупности, какая из характеристик является более или менее важной, а во-вторых, в разных условиях (например, при загрузке бытовой техники или разобранной мебели) для оптимальной загрузки, наборы этих характеристик могу существенно отличаться. Поэтому для подбора оптимального набора характеристик используются различные математические методы. В частности оптимизация может быть проведена методами генетических алгоритмов [4] или с использованием различных парадигм нейронных сетей [5].

9. Функционал качества наборов 3-х мерных предметов

Функционал качества (см. 8) набора 3-х мерных предметов одинаковой высоты определяется следующим образом:

$$\mathfrak{A} = \alpha_1 * h + \alpha_2 * \sqrt[3]{\sum_{i\in M} v(i)} + \alpha_3 * \sqrt{\sum_{i\in M} s(i)} + \alpha_4 * \sum_{i\in M} p(i) +$$

$$+ \alpha_5 * \sqrt[3]{\sum_{i\in N} q(i) * v(i)} + \alpha_6 * \sqrt{\sum_{i\in N} q(i) * s(i)} + \alpha_7 * \sum_{i\in N} q(i) * p(i) +$$

$$+\alpha_8 * \sqrt[3]{\frac{\sum_{i\in N} q(i)*v(i)}{\sum_{i\in N} q(i)}} + \alpha_9 * \sqrt{\frac{\sum_{i\in N} q(i)*s(i)}{\sum_{i\in N} q(i)}} + \alpha_{10} * \frac{\sum_{i\in N} q(i)*p(i)}{\sum_{i\in N} q(i)} +$$

$$+\alpha_{11} * \frac{\sum_{i\in N} q(i) * \tilde{p}(i)}{\sum_{i\in N} q(i)} + \alpha_{12} * \sqrt{\frac{\sum_{i\in N} q(i) * \tilde{s}(i)}{\sum_{i\in N} q(i)}} +$$
$$+\alpha_{13} * \sqrt[3]{\sigma_v(N)} + \alpha_{14} * \sqrt{\sigma_s(N)} + \alpha_{15} * \sigma_p(N) + \alpha_{16} * \sigma_{\tilde{p}}(N) + \alpha_{17} * \sqrt{\sigma_{\tilde{s}}(N)},$$

где

h — высота ящиков в наборе;

N — множество типов ящиков и блоков ящиков входящих в набор;

M — множество типов ящиков, из которых состоят блоки, входящие в набор;

$v(i)$ — объем ящика или блока типа i;

$s(i)$ — суммарная площадь всех граней ящика или блока типа i;

$p(i)$ — суммарная длина линейных размеров ящика или блока типа i;

$q(i)$ — количество ящиков или блока типа i в наборе;

$\tilde{s}(i)$ — разница площадей максимальной и минимальной граней ящика или блока типа i;

$\tilde{p}(i)$ — разница максимального и минимального линейных размеров ящика или блока типа i;

$\sigma_x(Y)$ — средне квадратичное отклонение функции x по всем элементам множества Y;

Значения коэффициентов α_i выбираются из естественных предположений, например, очевидно, что сначала надо попытаться уложить набор максимального объема, а затем «дозаполнять щели» более мелкими наборами, отсюда α_2 и α_5 должны иметь большие положительный значения. С другой стороны, проще сначала расставить «плоские» ящики, типа стекла и фанеры, а потом упаковывать «кубические» ящики, типа бытовой техники, отсюда α_{11} и α_{12} тоже должны иметь большие положительные значения. Из подобных эвристических соображений, путем анализа большого количества различных особенностей и вариантов задач на расчет оптимальной укладки были определены и остальные характеристики.

10. Функционал качества 2-х мерных предметов

Функционал качества (см. 8) 2-х мерных предметов, который используется в алгоритме 4.5, определяется следующим образом:

$$\mathfrak{B} = \beta_1 * \frac{\sqrt{s(i)}}{c(j)} + \beta_2 * \sqrt{s(i)} + \beta_3 * p(i) + \beta_4 * \tilde{p}(i),$$

где

$c(i)$ — количество ящиков, входящих в состав 2-х мерного предмета i;

$s(i)$ — площадь 2-х мерного предмета i;

$p(i)$ — периметр 2-х мерного предмета i;

$\tilde{p}(i)$ — разница длин минимального и максимального ребра 2-х мерного предмета i;

Значения коэффициентов β_i выбираются из естественных предположений, например, очевидно, что сначала надо попытаться уложить предметы максимальной площади, а затем «дозаполнять щели» более мелкими предметами, отсюда β_2 должен иметь большое положительное значение. С другой стороны, проще сначала расставить «узкие» предметы, типа стекла и фанеры, а потом упаковывать «квадратные» предметы, типа бытовой техники, отсюда β_4 тоже должен иметь большое положительное значение и т. п.

11. Функционал качества областей укладки

Функционал качества (см. 8) областей укладки, который используется в алгоритме 7, определяется следующим образом:

$$\mathfrak{C} = \gamma_1 * \mathfrak{B} + \gamma_2 * \sqrt{S} + \gamma_3 * \sum_{i \in N} l_i + \gamma_4 * \sum_{i \in N \wedge i:2} l_i + \gamma_5 * l_{\min} + \gamma_6 * l_{\max} +$$
$$+ \gamma_7 * \mu_{l_i}(i \in N) + \gamma_8 * \mu_{l_i}(i \in N \wedge x_i > x_{i+1}) + \gamma_9 * \mu_{l_{i+1}}(i \in T) +$$
$$+ \gamma_{10} * \sigma_{l_{i+1}}(i \in T) + \gamma_{11} * \mu_{\min(l_i, l_{i+2})}(i \in T) + \gamma_{12} * \sigma_{\min(l_i, l_{i+2})}(i \in T) +$$
$$+ \gamma_{11} * \mu_{|\min(l_i, l_{i+2}) - l_{i+1}|}(i \in T) + \gamma_{12} * \sigma_{|\min(l_i, l_{i+2}) - l_{i+1}|}(i \in T),$$

где

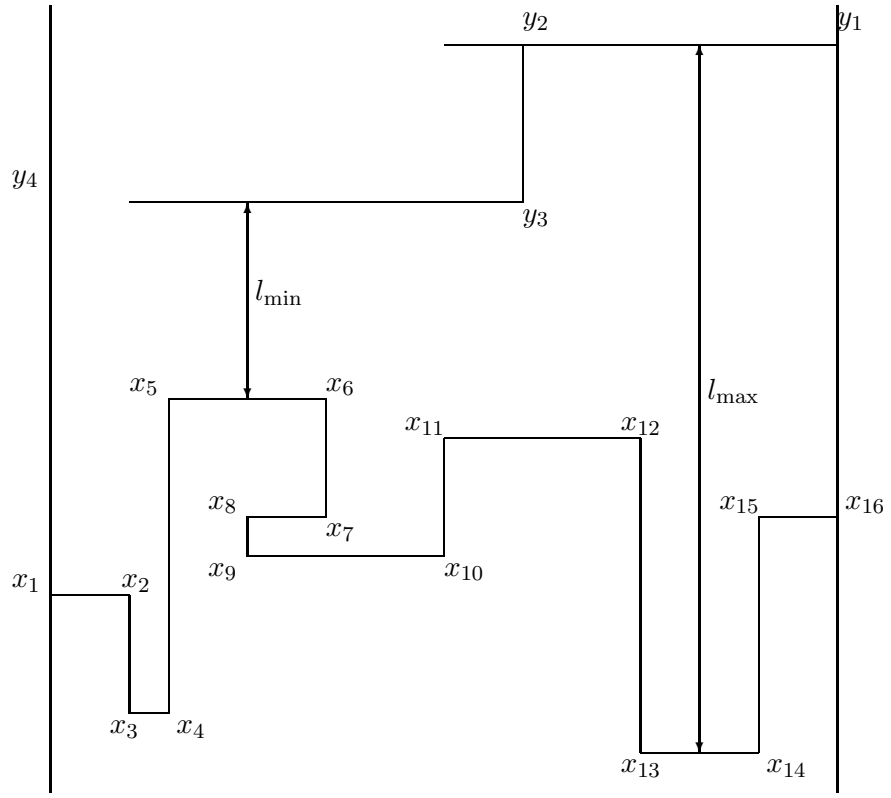

Рис. 3. Область загрузки

$\mathfrak{B}$ — значение функционала качества 2-х мерного предмета (см. 10), после установки которого образована данная область (см. 7);

S — площадь 2-х мерного предмета (см. 10), после установки которого образована данная область (см. 7);

N — множество вершин внутренней ломаной (исключая последнюю);

T — множество вершин внутренней ломаной, которые вместе с тремя последующими вершинами образуют тупик, под тупиком подразумевается последовательность из 3-х участков ломаных, крайние из которых имеют разную ориентацию (на рис. 3 это точки x_2, x_7 и x_{12});

l_i — длина отрезка (x_i, x_{i+1}) внутренней ломаной;

$l_{\min}$ — минимальное расстояние между внутренней и внешней границами области;

$l_{\max}$ — максимальное расстояние между внутренней и внешней границами области;

$\mu_x(Y)$ — математическое ожидание функции x по всем элементам множества Y;

$\sigma_x(Y)$ — среднеквадратичное отклонение функции x по всем элементам множества Y;

Этот набор характеристик выбран из некоторых эвристических соображений, например, очевидно, появление очень глубоких и узких щелей в границе области (тупиков) существенно затруднит последующую укладку предметов, отсюда значения коэффициентов $\gamma_{10} - \gamma_{14}$ должны иметь маленький или отрицательный коэффициент в функционале и т. п.

12. Внедрения и особенности программной реализации

Описанный алгоритм имеет сугубо практическое значение и реализован в виде различных программных систем. Во-первых, это внедрения в информационные системы различных предприятий. Во-вторых, алгоритм реализован в виде линейки программных продуктов. В-третьих, алгоритм представлен в интернете виде бесплатного сервиса по расчету оптимальной укладки грузов в транспортные средства на сайте www.packer3d.ru. Интерес, который пользователи проявляют как к программным продуктам, так и к online-сервису показывает, что реализация вполне приемлема и удобна для практического использования. По отзывам и анализу статистики использования можно сделать вывод, что успешное внедрение алгоритма на предприятии позволяет снизить расходы на грузоперевозки до 20%. При этом скорость работы данной реализации алгоритма вполне приемлема при решении задач, возникающих на практике. Например, даже с учетом различных дополнительных ограничений, скорость рас-

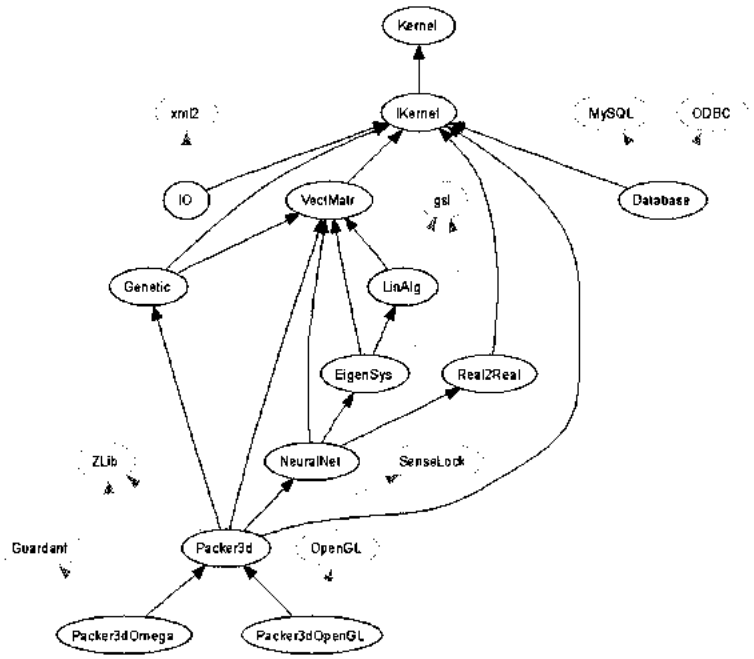

Рис. 4. Зависимость библиотек C++ классов.

чета оптимальной укладки 4000 ящиков 100 различных типов составляет **меньше минуты** на компьютере с процессорной мощностью 2 гигагерца. Это при этом, что любой алгоритм поиска *наилучшего* решения будет работать неприемлемо долго (годы) уже для задачи из нескольких десятков ящиков. Качество работы алгоритма (заполнения объема) существенным образом от параметров предметов, однако на практике качество в среднем составляет 80–90%, что является более высоким показателем, по сравнению с качеством заполнения транспортного средства человеком (для количества ящиков больше 100 и типов ящиков больше 10).

Алгоритм реализован в виде библиотеки методов на языке программирования C++. С учетом дополнительных функций и ограничений, которые учитывает алгоритм, размеры исходных текстов составляют примерно 50 тысяч строк кода, распределенных по 158 файлам. Целиком алгоритм реализован в нескольких библиотеках C++

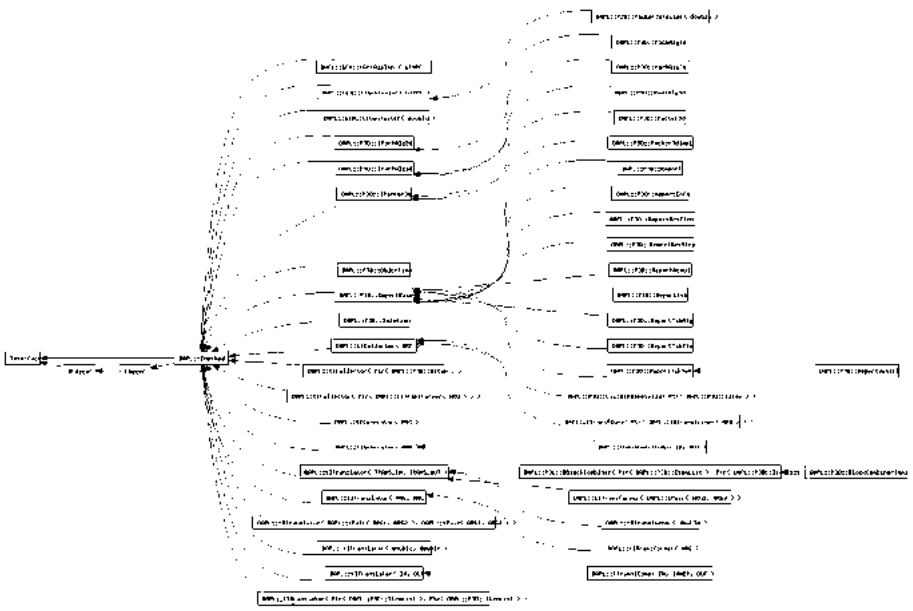

Рис. 5. Иерархия классов, реализующих методы алгоритма.

классов, которые используя друг друга образуют граф зависимостей, представленный на рисунке 4 (серыми стрелками обозначены зависимости от сторонних библиотек).

Структура объектно-ориентированной иерархии алгоритма такова, что допускает простое расширение базовой части алгоритма различными модификациями без существенного изменения ядра. Именно подобная структура легко расширять возможности реализованных программных систем и адаптировать алгоритм под различные требования конечных пользователей. Иерархия включает в себя 186 классов и интерфейсов, которые реализуют алгоритм. Показательная часть основной ветки иерархии классов представлена на рисунках 5 и 6.

Исходные тексты алгоритма зарегистрированы в реестре программ для ЭВМ федеральной службы по интеллектуальной собственности, патентам и товарным знакам, свидетельство № 2007614694.

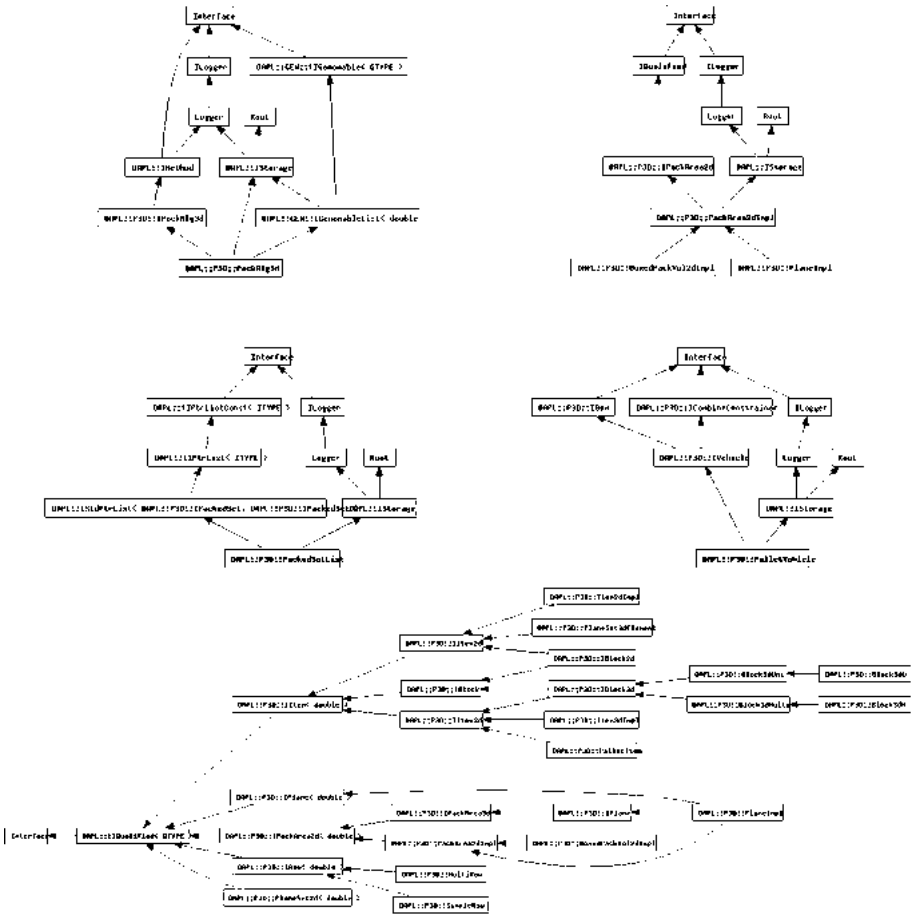

Рис. 6. Локальная иерархия базовых классов алгоритма.

13. Заключение

В работе предложен алгоритм расчета оптимальной укладки ящиков в транспортное средство удобный для использования при решении практических задач. Алгоритм не рассчитан на нахождение наилучшего решения задачи о рюкзаке, однако его высокая скорость и вполне приемлемый результат позволяют использовать его в реальных задачах, возникающих на практике. Помимо высокой скорости

работы основным преимуществом алгоритма является его высокая адаптивность, позволяющая реализовывать модификации алгоритма для учета дополнительных ограничений и легко модернизировать алгоритм для нужд конечных пользователей. Формат данной статьи не позволяет описать подробно структуру алгоритма, а также все его **реализованные** модификации, которые позволяют эффективно применять его на практике. Это будет сделано в последующих работах.

Хотелось бы отметить особое участие в разработке алгоритма и выразить огромную благодарность моему научному руководителю Строгалову Александру Сергеевичу. Именно совместно с ним была выработана основная идея алгоритма, а его ценные советы и замечания помогли решить много принципиальных трудностей, возникавших при адаптации алгоритма к новым требованиям.

Список литературы

[1] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

[2] Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.

[3] Ху Т. Ч., Шинг М. Т. Комбинаторные алгоритмы. Нижний Новгород: Изд-во Нижегородского гос. университета им. Н. И. Лобачевского, 2004.

[4] Whitley Darrell. A genetic algorithm tutorial // Statistics and computing. 4. 65–85. 1994.

[5] Псиола В. В. Обзор основных нейросетевых моделей // Интеллектуальные системы. Т. 4. Вып. 3–4. 1999.