**УДК 658.1:004.4**

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ВЫБОРА ОПТИМАЛЬНОЙ ИНВЕСТИЦИОННОЙ ПРОГРАММЫ ПРЕДПРИЯТИЯ**

Загирова Е.А.

В статье представлена техника программной реализации алгоритма для задачи выбора эффективного инвестиционного пакета проектов в условиях интервальной неопределенности исходных данных, с критерием формирования пакета в виде чистого дисконтированного дохода всей программы в целом. Предложена реализация двух моделей: с условием гарантированности дохода (максиминная стратегия) и в условиях неопределенности (дисперсия дохода как мера его риска).

Ключевые слова: инвестиционная программа, чистый дисконтированный доход, дисперсия, риск, программная реализация, вычислительный эксперимент

В современном мире участники бизнес-процессов, осознавая, что эффективная работа и развитие предприятия были возможны только при увеличением объема инвестиций, постепенно вывели эффективное управление инвестиционной деятельностью на первое место в стратегическом развитии предприятий. На данный момент времени существует множество моделей выбора инвестиционной политики [1].

В реализованных моделях чистый дисконтированный доход (ЧДД) принят как агрегированный показатель эффективности инвестиционной деятельности. В них также учитывается интервальная неопределенность исходных данных для исключения недопустимых и неэффективных решений.

# Постановка задачи

Инвестиционная программа представляет собой совокупность реализу­емых инвестиционных проектов предприятия, состоящих из перечня объек­тов инвестиций, их характеристик и объемов финансирования [2, 3].

Чистый дисконтированный доход (ЧДД) инвестиционной программы (сумма ЧДД проектов, которые были включены в инвестиционный пакет) выбран основным критерием формирования оптимальной инвес­тиционной программы предприятия.

Введены следующие обозначения [2, 3]:

*  – множество из  инвестиционных проектов, которые могут быть включены в состав инвестиционной программы;
*  – множество продолжительностей реализации инвестиционных проектов (в расчетных периодах);
* – горизонт планирования (число расчетных периодов);
*  – объемы финансирования инвестиционной программы предприятия по расчетным периодам.

Каждый из инвестиционных проектов  охарактеризован двумя показателями:

* – величина ЧДД, приведенного к моменту начала реализации проекта, если он был начат в период .
*  – потребность в финансировании инвестиционного проекта  в расчетный период  от начала реализации инвестиционной программы, при условии, что он будет начат в период .

Прогнозируемыми величи­нами являются показатели доходов и расходов, которые зависят от ряда факторов.  и  считаются случайными величинами. Получены интервальные оценки ЧДД , потребностей  и финансовых ресурсов предприятия  для каждого проекта и всех расчетных периодов.

Для построения модели введем булевы переменные



Поскольку реализация инвестиционного проекта  может начаться не позже чем в период , то должно выполняться следующее условие:



Условие реализуемости инвестиционной программы имеет вид



ЧДД для всего портфеля проектов равен



Осторожная (максиминная) стратегия направлена на получение макси­маль­ного гарантированного результата. Применяя нижние оценки ЧДД и объемов финансирования по расчетным периодам и верхние оценки потреб­ностей финансирования, гарантируется оптимальность значения следующей задачи [2, 3]:



 (1)

 (2)

 (3)

В данной работе также рассматривается реализация модели с учетом риска. Дисперсия чистого дисконтированного дохода  используется в качестве меры риска по аналогии с задачей Марковица-Тобина [2] при формирования портфеля ценных бумаг. При этом учитывается независимость между ЧДД и булев характер переменной .

В данном случае строятся эффективные инвестиционные программы в условиях риска (т.е. множество Парето в пространстве критериев «риск»-«ЧДД»). Наиболее подходящая инвестиционная программа выбирается на основании системы предпочтений лица принимающего решение при использовании систем поддержки принятия решений.

Ожидаемая величина ЧДД инвестиционной программы ищется как математическое ожидание, полагая, что чистый дисконтированный доход равномерно распределен на интервале , .

В итоге получаем, что эффективную инвестиционную программу в условиях риска можно найти при решении задач [2]:

 (5)

 (6)

для которых выполняются условия (1) - (3),  и  – допустимые уровни дисперсии и математического ожидания соответственно.

Все представленные задачи являются задачами булева линейного программирования, поэтому решены алгоритмом на основе динамического программирования.

# Алгоритм решения

Для решения задач использовался тот же алгоритм, что и для решения поиска оптимального инвестиционного пакета при условии гарантированности результата [3], но с некоторыми изменениями. Далее представлен алгоритм решения задачи, адаптированный к новы условиям.

Шаг 0. Инициализация. Составить список всех состояний на текущий момент , . Это состояния, в которых проекты либо не начались, либо активировались в момент . Исключить из списка все состояния, которые не удовлетворяют условиям, т.е. комбинация требует инвестиций больше, чем предприятие может выделить (2) или какой-либо из проектов комбинации будет активен и за пределами горизонта событий. В задаче (5) также проверяется математическое ожидание ().

Шаг 1. Динамика системы во времени. Начать цикл по времени. На каждой итерации переменная времени увеличивается на 1. Если данная переменная сравняется по значению с горизонтом событий, то цикл прекращается и происходит переход к шагу 3, иначе – к шагу 2.

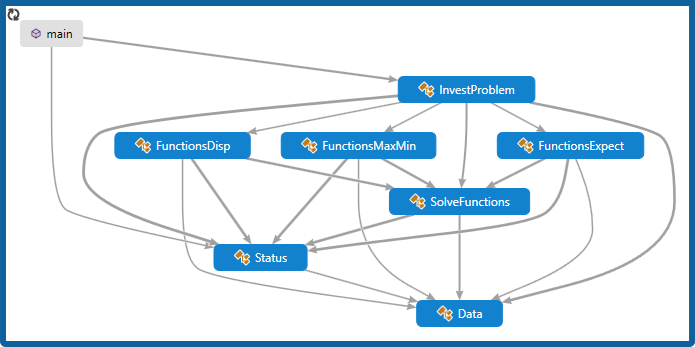
Шаг 2. Тело цикла. Составить новый список состояний. Для этого просмотреть все состояния системы из списка состояний. Построить все возможные состояния системы исходя из предыдущего состояния, т.е. добавить в комбинацию уже активных проектов другие возможные комбинации проектов. При этом проведя такую же проверку каждого состояния, как и в шаге 0. Если не удалось активировать еще проекты, оставить текущее состояние системы. Удалить старый список состояний.

Шаг 3. Поиск оптимума. В зависимости от задачи, найти максимальное по ЧДД (1), по математическому ожиданию (5) или минимальное по дисперсии (6) состояние системы. Это и будет решением задачи.

Замечание. Проверка условия  в задаче (6) проходит в самом конце алгоритма при переборе списка всех найденных состояний в поиске оптимального, т.к. при более раннем применении оно может отсечь возможные варианты, где основной доход от проектов приходит в более поздние периоды.

# Изменения в реализации

Для работы с несколькими задачами были внесены изменения в уже имеющуюся реализацию [3]. Т.к. алгоритм в основном сохраняется, от задачи к задаче меняются только некоторые ограничения и условия поиска оптимума, то была разработана иерархия классов. На диаграмме зависимостей отображены основные структурные изменения (рис. 1).



*Рисунок 1 – Диаграмма зависимостей*

В основном классе InvestProblem из всех методов сохранился только метод Solve, все остальные сгруппированы в абстрактном классе SolveFunctions, где виртуальными методами объявлены addToStatusList() и checkStatus(). От него наследуют классы FunctionsMaxMin, FunctionsExpect и FunctionsDisp, в которых данные виртуальные методы переопределяются в соответствии с решаемой задачей, т.е. в условиях гарантированности результата, максимального математического ожидания или минимальной дисперсии соответственно.

Для решения новых типов задач были внесены изменения и в классы Data и Status. В Data теперь также хранится идентификатор задачи и уровень показателя, т.е. для задачи (5) – верхняя граница дисперсии (максимально возможный риск), для задачи (6) – нижняя граница математического ожидания (минимальный ожидаемый ЧДД). Теперь необходимо хранить не только верхние или нижние границы показателей, но оба. В классе Status теперь дополнительно хранится математическое ожидание и дисперсия состояния, которые просчитываются при создании объекта класса.

Был переработан метод бинарного перебора всех вариантов состояний системы, ранее осуществлявшийся рекурсивным перебором по алгоритму Грея [3]. Теперь его выполняет метод generateProjectsTimes() (рис. 2), который на вход получает кроме времени и базового вектора [3] список векторов, куда записываются новые варианты возможных изменений состояний.

|  |
| --- |
| void SolveFunctions::generateProjectsTimes(  int time,  const vector <int>& baseVec,  vector<vector<int>>& List){  //количество проектов, которые никогда не активировались  int k = 0; for (int i = 0; i < baseVec.size(); i++) if (baseVec[i] == -1) k++;  vector <int> empty(k, -1);  List.push\_back(empty);  for (int i = 0; i < k; i++){  int n = List.size();  for (int j = 0; j < n; j++){  List.push\_back(List[j]);  List[n+j][i] = time;  }  }  } |

Рисунок 2 – Метод generateProjectsTimes()

# Примеры решения

Рассмотрим применение данного кода к решению задачи. На рис. 3 представлена схема того, как задается условие задачи. Данные разделены пустой строкой на пять блоков. В первом блоке четыре строки. В первой строке записан идентификатор задачи **ID**, количество проектов **n**, горизонт планирования m (начиная с нулевого периода) и значение критерия **e/d**, т.е. ограничение для дисперсии (5) или матожидания (6). Во второй – продолжительность каждого проекта **L**, в третьей и в четвертой – нижняя **Rl** и верхняя **Ru** граница имеющихся у предприятия ресурсов в каждый период времени. Во втором блоке приводятся нижние оценки ЧДД проектов **NPV\_l** в каждый период времени возможного начала реализации . В каждой строке отображена информация по одному проекту. В третьем блоке приводятся верхние оценки ЧДД проектов **NPV\_u**. В четвертом блоке записан нижний уровень затрат **Inv\_l** на каждый проект в каждый период времени, начиная с момента активации проекта по тому же принципу, что и предыдущий блок. В пятом блоке записана верхняя граница затрат на каждый проект **Inv\_u**.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | n | m | e / d |
| L1 | L2 | … | Ln |
| Rl 1 | Rl 2 | … | Rl m |
| Ru 1 | Ru 2 | … | Ru m |
|  | | | |
| NPV\_l\_1\_0 | NPV\_l\_1\_1 | … | NPV\_l\_1\_m-L1 |
| … | … | … | … |
| NPV\_l\_n\_0 | NPV\_l\_n\_1 | … | NPV\_l\_n\_m-Ln |
|  | | | |
| NPV\_u\_1\_0 | NPV\_u\_1\_1 | … | NPV\_u\_1\_m-L1 |
| … | … | … | … |
| NPV\_u\_n\_0 | NPV\_u\_n\_1 | … | NPV\_u\_n\_m-Ln |
|  | | | |
| Inv\_l\_1\_0 | Inv\_l\_1\_1 | … | Inv\_l\_1\_L1 |
| … | … | … | … |
| Inv\_l\_n\_0 | Inv\_l\_n\_1 | … | Inv\_l\_n\_Ln |
|  | | | |
| Inv\_u\_1\_0 | Inv\_u\_1\_1 | … | Inv\_u\_1\_L1 |
| … | … | … | … |
| Inv\_u\_n\_0 | Inv\_u\_n\_1 | … | Inv\_u\_n\_Ln |

*Рисунок 3 – Схема условия*

Результаты решения задачи для пакета из шести инновационных проектов представлены на рис. 5. В первом столбце показан номер проекта, во втором – период, в который следует начать проект. -1 означает то, что проект не должен быть выполнен. При формировании инвестиционной программы по максиминной стратегии гарантированный ЧДД программы равен 80 (рис. 4.а). Это решение совпало с решением задачи поиска минимальной дисперсии при условии, что матожидание будет больше 80 (рис. 4.в). При решении задачи поиска максимального матожидания с ограничением дисперсии в пределах 40 получили гарантированный ЧДД 47, ожидаемую прибыль – 57 (рис. 4.б).

# Заключение

Предложенная реализация позволяет найти оптимальный пакет инвестиционных проектов при условии гарантированности результата, а также в условиях неопределенности. Данная реализация не позволяет решать задачи большой размерности, что показано на рис. 5. Заметим, что в течении работа цикла, каждая итерация продолжается дольше, чем предыдущая, но скорость роста уменьшается до 20% к количеству проектов равному 8 (рис. 6).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NPV | 80 |  | NPV | 47 |  | NPV | 80 |
| Expectancy | 95 |  | Expectancy | 57 |  | Expectancy | 95 |
| Dispersion | 41.6667 |  | Dispersion | 33.3333 |  | Dispersion | 41.6667 |
| Project | time |  | Project | time |  | Project | time |
| 1 | -1 |  | 1 | -1 |  | 1 | -1 |
| 2 | -1 |  | 2 | -1 |  | 2 | -1 |
| 3 | 0 |  | 3 | -1 |  | 3 | 0 |
| 4 | -1 |  | 4 | -1 |  | 4 | -1 |
| 5 | -1 |  | 5 | 0 |  | 5 | -1 |
| 6 | 4 |  | 6 | -1 |  | 6 | 4 |
|  |  |  |  |  |  |  |  |
| а) | |  | б) | |  | в) | |

*Рисунок 4 – Решение задачи*

*a) максимин, b) матожидание, c) дисперсия*

Рисунок 5 – *Время выполнения программы*

Рисунок 6 – *Скорость выполнения итерации цикла по отношению к предыдущей итерации, %*

Направление дальнейшего исследования – построение программного обеспечения, позволяющего решать задачи большой размерности, а также реализовать методы поиска оптимального портфеля проектов с учетом вероятности недостижимости заданного уровня чистого дисконтированного дохода.

# Библиографический список

1. Кибзун, А. И. Алгоритм решения обобщенной задачи Марковица [текст] / А. И. Кибзун, А. И. Чернобровов ; Автоматика и телемеханика. – 2011. – № 2. – C. 77–92.
2. A. V. Panyukov and E. N. Kozina Forming of the Competitive Investment Programs for Enterprises Workshop on Computer Modelling in Decision Making (CMDM 2016). CEUR Workshop Proceedings. Vol. 1726. pp. 89-99. URL http://ceur-ws.org/Vol-1726/paper-09.pdf
3. Панюков, А.В. Программная реализация максиминной стратегии управления инвестиционной деятельностью предприятия [текст]/ А.В. Панюков, Е.А. Загирова; Информационные технологии и системы. – 2017. – С. 211-219.