УДК 519.6: 629.7.036.3

Н.А. ГУЛЯЕВ, Ю.А. ЩЕРБАКОВА

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ДОСТАВКИ РЕСУРСОВ

На примере конкретной задачи рассмотрены алгоритмы для решения проблемы поиска кратчайшего пути, а также возможность оптимизации расходов. Предложена программная реализация модели развоза ресурсов по графу. Применено агентное моделирование для построения имитационной модели динамической системы. Рассмотрены основные вопросы, что возникают в процессе решения проблемы, и предложены пути их решения. В ходе моделирования был использован алгоритм Флойда (алгоритм динамического программирования). Получены результаты, которые могут быть использованы для прогнозирования поведения системы. Проведена работа по оптимизации выбора стратегии, собраны статистические данные, сделаны соответствующие выводы.

Ключевые слова: *имитационное моделирование, агентное моделирование, алгоритм Флойда, граф, прогнозирование.*

Введение

Имитационное моделирование – это одна из разновидностей аналогового моделирования, которая реализуется с помощью набора математических инструментальных средств, имитирующих компьютерных программ и технологий программирования, позволяющих провести целенаправленное исследование структуры и функций реального процесса в памяти компьютера в режиме «имитации», выполнить оптимизацию некоторых его параметров.

Целесообразность применения имитационных моделей возникает довольно часто, если:

– невозможно экспериментировать на реальном объекте;

– невозможно построить аналитическую модель: в системе есть время, причинные связи, нелинейности, стохастические (случайные) переменные;

– необходимо сымитировать поведение системы во времени.

Имитационная модель включает в себя большое число параметров, в частности, зависимость моделируемого процесса во времени (временная динамика) и в пространстве (пространственная динамика), логистика и т.д.

Одним из направлений применения имитационного моделирования является изучение дискретных состояний системы массового обслуживания (СМО) в непрерывном времени в форме моделирующего алгоритма. Этим объясняется актуальность данной задачи, ведь в реальном мире мы имеем дело с динамическими системами.

Формулировка задачи

В качестве примера возьмем развозку ресурса по заданному графу (в качестве вершин графов берутся районы города, а в качестве ребер – время, за которое машина с постоянной скоростью доедет к смежной вершине). Проблема оптимальной перевозки ресурсов по городу становится нетривиальной тогда, когда в системе введено время. Т.е. применением алгоритма для нахождения кратчайшего пути решение не ограничивается, и задача требует более тщательного подхода. Вопрос оптимизации в этом случае становится краеугольным камнем в сфере логистики и управлении запасами [2].

Имитационная модель содержит граф, состоящий из 11 вершин. Каждая вершина соответствует определенному району города. С каждого района поступают звонки с заказами. В одной из вершин находится база, которая имеет в наличии 3 машины.

Применение агентного моделирования позволяет представить поведение машины, при развозе товара, как совокупность состояний агента. Агент переходит из одного состояния в другое при определении его поведения на индивидуальном уровне. При этом глобальное изменение системы возникает как результат действия множества агентов.

Представление города как графа позволяет применять математические методы для решения практических задач. Граф можно представить в виде матрицы.

Постановка задачи:

1. Построить имитационную модель развоза ресурса.

2. Сделать возможным прогнозирование поведения системы – для проведения экспериментальных исследований.

Выбор программного обеспечения и алгоритма

Для решения поставленной задачи была выбрана среда AnyLogic 6.4.1. в связи с достаточно хорошим набором статистических возможностей, визуальных эффектов и элементов теории вероятности.

Для нахождения кратчайшего пути был использован алгоритм Флойда, который является алгоритмом динамического программирования.

Алгоритм Флойда - Уоршелла — динамический алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. Разработан в 1962 году Робертом Флойдом и Стивеном Уоршеллом [1].

Общее описание алгоритма:

Дан ориентированный граф G =< V, E > с матрицей смежности. Найти кратчайшее расстояние между всеми парами вершин.

При построении матрицы кратчайших расстояний между всеми вершинами графа используется идея построения на первоначальном этапе матрицы v(M,M) возможных одношаговых переходов из каждой вершины в каждую (т.е. в результате выполнения алгоритма образуется матрица кратчайших расстояний от любой одной вершины графа до любой другой или до самой себя):

v[i,k]=min{l[u]|beg u=i, end u=k},

где v[i,k] – кратчайшее расстояние между вершинами i и k

beg u – начало дуги

end u – конец дуги

min{l[u]|beg u=i, end u=k} – минимальное значение длины дуги и, начало который в вершине i, а конец дуги соответствует вершине k.

А затем эта матрица пересчитывается тройным циклом по множеству вершин, причем внешний цикл – по промежуточной вершине:

Псевдокод:

```
for (i in M) do
for (i1,i2 in M) do
     if ( v[i1,i2] > v[i1,i]+v[i,i2] ) then
v[i1,i2]:= v[i1,i]+v[i,i2]
end
end
end
```

Смысл алгоритма в том, что когда при фиксированном i перебираются все возможные пары i1 и i2, то происходит сравнение лучшего из имеющихся к данному моменту путей от i1 до i2 путем, составленным из лучшего пути от i1 до i и лучшего пути от i до i2., и в результате этого сравнения может быть улучшен путь от i1 до i2.

Сам путь, на котором достигается максимум, легко строится, если одновременно с матрицей v пересчитывать матрицу d[M,M], в которой d[i,k] – выходящая из i дуга, с которой начинается путь, ведущий в k (для пересчета этой матрицы при каждом исполнении оператора присваивания внутри тройного цикла должен исполняться еще один оператор d[i1,i2]:=d[i1,i] – переход из i1 в i2 должен быть таким же, как в i). По завершении расчетов путь из любой вершины в любую легко создается по данным матрицы d.

Создание имитационной модели

Для построения имитационной модели мы будем использовать агентное моделирование.

Под агентом в агентном моделировании понимается элемент модели, который может иметь поведение, память (историю), контакты и т.д. Агенты могут моделировать людей, компании, проекты, автомобили, города, животных, корабли, товары и т.д.

Агент «Car»

Для каждой машины в нашей модели мы формируем очередь с 3 мест. А значит, единственный параметр, который передаётся машине – это номер вершины графа с заказом.

Построим блок-схему «Состояние агента» (рис. 1) для агента, а также определим набор переменных и методов для него.

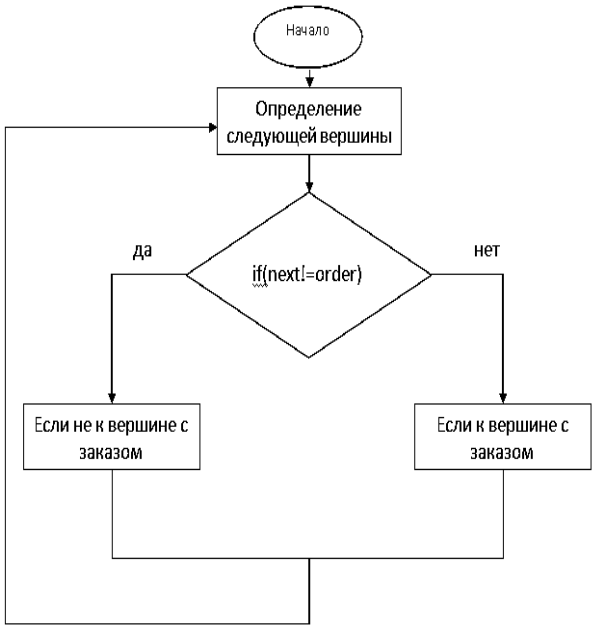

Рис. 1. «Состояние агента»

Главной особенностью внедрения агента является то, что описывая его поведение, можем прогнозировать поведение всей системы. А значит, конец работы программы определяет пользователь программы. В качестве примера ограничим работу агента по времени. В соответствии с этим сбор статистических данных будет производиться за выбранный нами период.

В среде AnyLogic кроме переменных и функций имеются события. Событие является самым простым способом планирования действий в модели. События часто используются для моделирования задержек и таймаутов.

Параметры, функции и события:

- переменные:

speed *– скорость машины.*
rotation *– угол поворота*
km *– километраж*
isMoving *– машина в движении или стоит*
amount *– количество принятых заказов*
home *– номер вершины где находится*
next *– номер вершины к которой едет в данный момент*
order *– номер вершины с заказом*
previos *– переменная для метода* ***service***
pizzas *– сколько загружено всего*
pizzas_help *– сколько не заказанных осталось*

- функции:

amount_bell *– возвращает число заказов с вершины*
driving *– определяет путь к вершине с заказом*

события:

go_next *– метод для запуска движения*
Moving *– проверяет в движении машина или нет*
service *– отключаем анимацию захваченного заказа*
optimisation *– обслуживаем заказ с очереди, если он на пути движения*

Программная реализация алгоритма Флойда

Функция имеет два аргумента home и demand (начальное положение и вершина заказа соответственно), и возвращает номер вершины на следующем шагу движения машины. Также она использует матрицу T, как необходимые данные для алгоритма Флойда.

Листинг:

```
int n=11;
int[][] p=newint[12][12];
double[][] T=
{
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 7.6, 2.1, 4.4, 100, 100, 100,
100, 100, 100, 100},
{0, 7.6, 0, 100, 100, 2.4, 100, 100,
100, 3.4, 100, 100},
{0, 2.1, 100, 100, 4.7, 100, 100,
100, 100, 100, 100, 100},
{0, 4.4, 100, 4.7, 0, 2.6, 100, 100,
3.1, 100, 3.5, 100},
{0, 100, 2.4, 100, 2.6, 0, 100, 100,
100, 100, 3.5, 100},
{0, 100, 100, 100, 100, 100, 0, 3.6,
3.6, 100, 100, 100},
{0, 100, 100, 100, 100, 100, 3.6, 0,
100, 100, 100, 6.6},
{0, 100, 100, 100, 3.1, 100, 3.6,
100, 0, 100, 3, 100},
{0, 100, 3.4, 100, 100, 100, 100,
100, 100, 0, 4.3, 100},
{0, 100, 100, 100, 3.5, 3.5, 100,
100, 3, 4.3, 0, 3.6},
{0, 100, 100, 100, 100, 100, 100,
6.6, 100, 100, 3.6, 0}
};
for ( int k = 1; k < n+1; k++ )
{
for ( int i = 1; i < n+1; i++ )
      {
      for (int j = 1; j < n+1; j++)
            {
if ( T[i][j]>( T[i][k] + T[k][j] ) )
                  {
      T[i][j] = T[i][k] + T[k][j];
      p[i][j]=k;
                  }
            }
      }
}
int s=0;
int k=p[home][demand];
s=k;
if(s!=0)
{
while(k>0)
{
k=p[home][k];
if(k!=0)
{s=k;}
}
}
else s=demand;
returns;
```

Проблема формирования очереди при наличии нескольких агентов затрагивает вопросы оптимального выбора машины для обслуживания и возможности удовлетворения заказа.

Например, нам нужно выбрать ближайшую машину к месту заказа, но не забыть, что количество ресурсов на ней должно быть больше либо равно количеству заказанных.

Как же происходит управление очередью?

И как сама очередь формируется?

Для решения этих вопросов составляется блок – схема «Управления заказами» (рис. 2). А также вносятся критерии для добавления в очередь того или иного заказа.

Появление заказа с любого района города является событием происходящее с заданной интенсивностью и используется для моделирования потока независимых событий.

Событие происходит для каждой вершины отдельно. Функция ***queue*** используется для формирования очереди.

Далее будут приведены переменные и функции, которые использовались для построения имитационной модели.

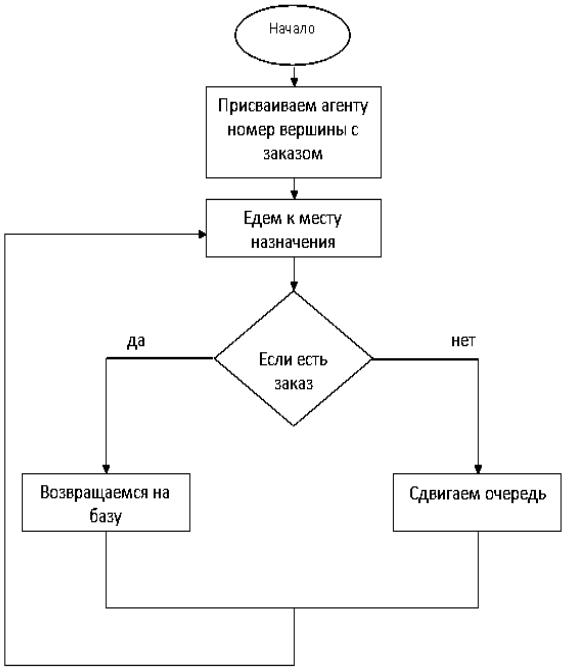

Рис. 2. «Управление заказами»

Переменные:
one_call,…,eleven_call – *количество заказов в каждой вершине*
one_bool,…,eleven_bool – тип boolean *(если заказ на пути движения любой из машин – мы его принимаем)*
point1, point2, point3 – *очередь для 1 машины*
point4, point5, point6 – *очередь для 2 машины*
point7, point8, point9 – *очередь для 3 машины*
all_order – *все заказы*
lost – *потерянные клиенты*
served1 – *обслужила первая машина*
served2 – *обслужила вторая машина*
served3 – *обслужила третья машина*

Функции:
visualization – *визуализация заказа*
road_check – *проверяет возможность обслужить заказ на пути к заданной вершине*
Вспомогательные переменные агента car:
pizzas – *сколько загружено всего*
pizzas_help – *сколько не заказанных осталось*

Использование и получение данных

Для имитации реального процесса нам будут нужны статистические данные такие как:

– примерное время перемещения между смежными вершинами графа (для заполнения матрицы Флойда);

– интенсивность заказов с каждой вершины графа.

После этого подбираем усредненную скорость для машин в нашей модели и запускаем программу. На выходе получаем:

– количество проданного товара (в ед.) для каждой машины в отдельности и суммарное количество;

– расстояние, которое проехала каждая машина в отдельности.

После экспериментальных исследований полученные данные можем использовать для нахождения чистого дохода:

$$P = n (c – s) –k,$$

где

P – чистый доход;

n – количество проданного товара;

c – цена на товар;

s – себестоимость товара;

k – издержки (затраты на бензин и т.д.).

Полученные результаты могут быть обработаны специалистами, и в соответствии с этим, сделаны предложения по максимизации доходов. Например, отказаться от обслуживания одного из районов, или устанавливать разную цену для каждого района.

Заключение

Рассмотрена система доставки ресурса по графу. Для нахождения оптимального пути был использован алгоритм Флойда (программную реализацию которого вмещает в себе одна из функций).

Предложенная работа демонстрирует основные достоинства имитационного моделирования:

1. Возможность описания поведения компонент (элементов) процессов или систем на высоком уровне детализации.

2. Отсутствие ограничений между параметрами имитационного моделирования и состоянием внешней среды реальных процессов и систем.

3. Возможность исследования динамики взаимодействия компонент во времени и пространстве параметров системы.

Эти достоинства обеспечивают имитационному методу широкое распространение.

Для решения поставленной задачи использовалось агентное моделирование, что дало возможность прослеживать поведение системы. Применение реализованной имитационной модели позволяет вы-

брать оптимальную стратегию продажи ресурса. Это возможно при наличии статистических данных связанных с зависимостью между ценой на товар и интенсивностью заказов в каждом районе. Таким образом, при имитационном моделировании рассмотренной задачи речь идет о статистическом имитационном моделировании.

Реализация очень проста:

– вводим начальные данные;

– устанавливаем время эксперимента;

– проводим эксперимент несколько раз (составляем выборку);

– сравниваем доход для каждой цены на товар;

– выбираем оптимальный вариант.

Имитационном моделирование является одним из наиболее широко используемых методов при решении задач синтеза и анализа сложных процессов и систем. Особенно это относится к ситуации, когда результат процесса зависит от принятием руководителем в сложившейся обстановке того или иного решения.

Литература

1. Алгоритмы построения и анализ, второе издание: пер. с англ. [Текст]/ Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М.: Издательский дом «Вильямс», 2005. – 1296 с.

2. Мешкова, Л.Л. Логистика в сфере материальных услуг (на примере снабженческо - заготовительных транспортных услуг) [Текст] / Л.Л. Мешкова, И.И. Белоус, Н.М. Фролов. – Томбов: изд-во ТГТУ, 2002. – 99 с.

3. Осоргин, А.Е. AnyLogic 6 лабораторный практикум [Текст] / А.Е. Осоргин. – Самара: ПГК, 2011. – 100 с.

4. Бочкарев, А.А. Планирование и моделирование цепи поставок [Текст]: справочное пособие / А.А. Бочкарев. – М.: изд-во Альфа-Пресс, 2008. – 192 с.

Поступила в редакцию 10.02.2012

Рецензент: д-р физ.-мат. наук, проф., проф. каф. информатики М.Л. Угрюмов, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ДОСТАВКИ РЕСУРСІВ

М.О. Гуляєв, Ю.А. Щербакова

На прикладі конкретної задачі був розглянутій алгоритмдля вирішення проблеми пошуку найкоротшого шляху, а також можливість оптимізації витрат. Запропонована програмна реалізація моделі розвезення ресурсу по графу. Застосовано агентське моделювання для побудови імітаційної моделі динамічної системи. Розглянуто основні питання, що виникають в процесі розв'язання проблеми, також запропоновані шляхи їх вирішення. В ході моделювання був використаний алгоритм Флойда (алгоритм динамічного програмування). Отримані результати, які можуть бути використані для прогнозування поведінки системи. Проведена робота з оптимізації вибору стратегії, зібрані статистичні дані, зроблені відповідні висновки.

Ключеві слова: імітаційне моделювання, алгоритм Флойда, агенте моделювання, граф, прогнозування.

SIMULATION MODELING OF RESOURCE DELIVERY

N.A. Guliaev, Y.A. Scherbakova

On the example of a specific task considered algorithms for solving the problem of finding the shortest path, as well as the opportunity to optimize costs. We propose a software implementation of the model delivers resources for the count. An agent design is applied for the construction of simulation model of the dynamic system Basic questions which arise up in the process of decision of problem are considered, and also ways of their decision are offered. During a design there was used Floyd'salgorithm (algorithm of the dynamic programming). There sults are got which can be used for prediction of conduct of the system. The work are carried out to optimize the choice of strategy, statistical data are collected, appropriate conclusion sare made.

Key words: simulation, Floyd's algorithm, graph, prediction, agent design.

Гуляев Николай Александрович – студент каф. высшей математики, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, e-mail: guliaev_kolia@bk.ru .

Щербакова Юнна Анатольевна – канд. физ.-мат. наук, ст. преп. каф. высшей математики, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, e-mail: yunne@yandex.ru