



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

Утверждаю

Ректор университета

_____ О.Н. Федонин

«_____» _____ 2019г.

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

**ВЫПОЛНЕНИЕ ОПТИМИЗАЦИОННОГО
ЭКСПЕРИМЕНТА В ПРОГРАММНОМ КОМПЛЕКСЕ
ANYLOGIC**

Методические указания
к выполнению лабораторной работы №10
для студентов очной формы обучения
по направлению подготовки
09.03.01 «Информатика и вычислительная техника»

Брянск 2019

УДК 004.65

Компьютерное моделирование. Выполнение оптимизационного эксперимента в программном комплексе Anylogic [Электронный ресурс]: методические указания к выполнению лабораторной работы № 10 для студентов очной формы обучения по направлению подготовки 09.03.01 «Информатика и вычислительная техника». – Брянск: БГТУ, 2019. – 20 с.

Разработали:

А.А.Трубакова,

ст.преп.;

А.О. Трубаков,

канд. техн. наук, доц.

Рекомендовано кафедрой «Информатика и программное обеспечение»
БГТУ (протокол № 4 от № 9 от 07.06.2019г.)

Методические указания публикуются в авторской редакции

1. ЦЕЛЬ РАБОТЫ

Целью работы является ознакомление с основами построения оптимизационных экспериментов и методикой его проведения в среде *AnyLogic* на примере модели АТС.

Продолжительность работы – 2 часа.

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучение основных возможностей и принципов работы с оптимизационными экспериментами в среде имитационного моделирования *AnyLogic*.
2. Построение модели системы массового обслуживания АТС в среде *AnyLogic*.
3. Проведение оптимизационного эксперимента в среде *AnyLogic*.

3. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Чтобы настроить оптимизацию в *AnyLogic*, нужно выполнить следующие шаги:

Создать в разработанной модели оптимизационный эксперимент.

Задать оптимизационные параметры (компоненты изменяемого вектора исходных факторов x) и области их изменения.

Задать условие остановки модели после каждого прогона. Это может быть либо остановка по времени выполнения прогона, либо остановка по условиям, накладываемым на переменные модели.

Задать целевую функцию — ту функцию, значение w которой отражает предпочтительность вектора исходных факторов x . Значение w должно быть доступно в конце каждого прогона модели, оно будет использоваться оптимизатором.

Задать ограничения, которые в конце каждого прогона определяют, допустимо ли значение вектора исходных факторов x . Ограничения можно не задавать (т. е. это опционально).

Задать условия прекращения оптимизации.

4. ПОСТРОЕНИЕ МОДЕЛИ АТС

4.1. Общее описание модели

Рассмотрим модель системы предоставления услуг мобильной связи с помощью автоматической телефонной станции, на которую поступают вызовы. Поставщик сервиса выбирает оборудование автоматической телефонной станции, он оценивает параметры потока входных запросов (как часто люди будут звонить) и характеристики потока обслуживания (сколько времени требуется для получения консультации по телефону). Проблема состоит в определении числа каналов, которые дадут ему максимальную прибыль. Введение дополнительных каналов связи приведет к росту доходов вследствие уменьшения числа необслуженных вызовов, но потребует дополнительных вложений. Какое число каналов выбрать поставщику сервиса?

Будем считать, что за каждый отвергнутый вызов платится штраф (усредненные потери от неудовлетворенности клиента), а обслуженные вызовы приносят доход. Имея стоимость поддерживаемых каналов связи, можно в качестве целевой функции в этой модели выбрать прибыль, определяемую как разность доходов за обслуживание телефонных вызовов и расходов на оборудование, обслуживание станции и штрафы. Прибыль будем подсчитывать в среднем за единицу времени, например, за минуту. Она равна доходу минус расходы на оборудование (вместе со стоимостью обслуживания) и штрафы за отказ в обслуживании (все приведенное к стоимости в течение одной минуты).

4.2. Структура модели в среде AnyLogic

Откройте новый проект, назовите его *MobileCommunication*. Из Библиотеки моделирования процессов с помощью мыши перенесите блоки структуры системы в поле редактора структуры модели и соедините их в соответствии с рис. 2. Нам нужен один источник заявок (блок *source*), блок, решающий, отказать или нет пришедшей заявке в обслуживании в соответствии с некоторым условием (*selectOutput*), блок обработки входных заявок с использованием ресурсов – в нашем случае каналов связи (*service*), блок ресурсов (*resourcePool*) и два блока стока (*sink* и *sink1*) (рис. 2).

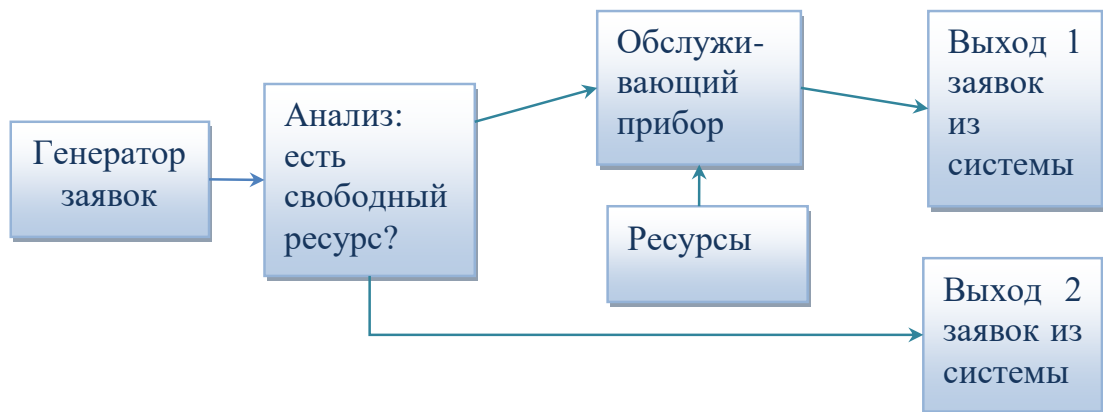


Рис. 1. Структура модели сервиса мобильной связи

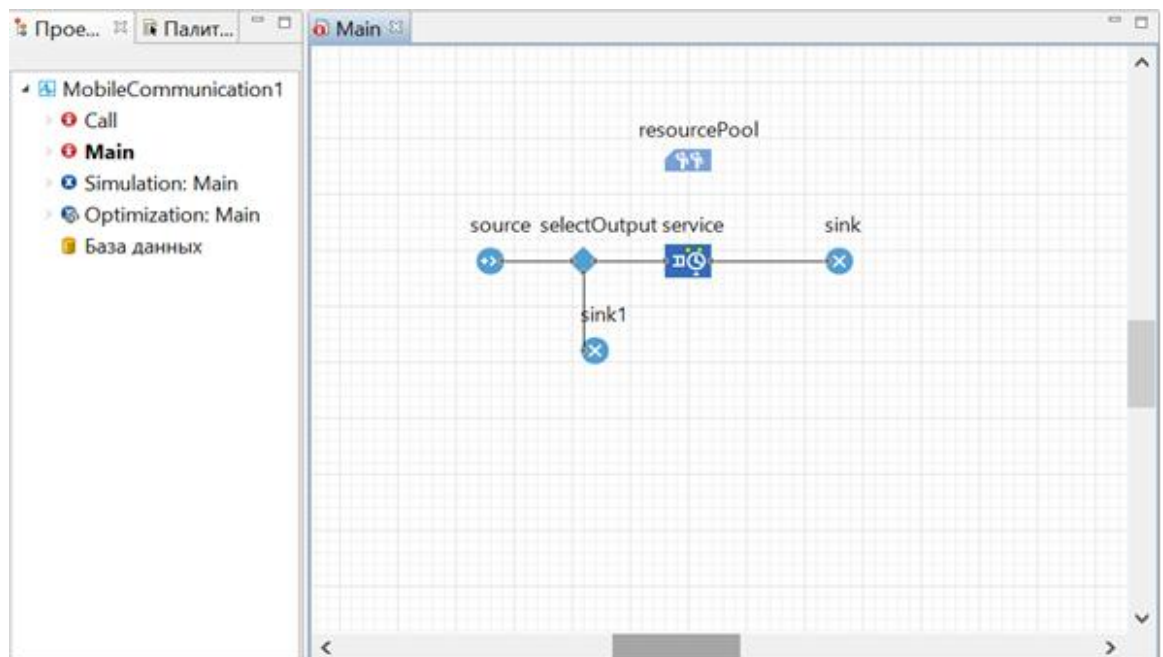


Рис. 2. Структура модели для решения проблемы мобильной связи в среде AnyLogic

Очевидно, что поток поступающих заявок и поток обслуживания имеют вероятностную природу. Для простоты будем считать, что они экспоненциальны. Введем в модель три параметра: целый N (число линий связи) и два вещественных: λ (интенсивность поступления заявок) и μ (интенсивность их обслуживания каждым каналом). Установим начальные значения для них $N=3$, $\lambda=1,5$ и $\mu=0,5$.

4.3. Параметры модели

Для определения характеристик эффективности системы установим следующие параметры для блоков системы.

Блок *source*. У параметра «Время между прибытиями» оставим экспоненциальное распределение, но установим у этого распределения параметр $\lambda - \text{exponential}(\lambda)$.

Блок *selectOutput* посылает принятые заявки на свой выход. В нашей модели условие должно быть истинно только в том случае, когда имеются свободные линии. Число обрабатываемых в этом блоке заявок можно получить, обратившись к функции *delaySize()*. Итак, в поле «Условие» параметра «Выход true выбирается» блока *selectOutput* нужно установить $\text{service.delaySize()} < N$.

Блок *service*. Здесь параметр «Время задержки» установите, как $\text{exponential}(\mu)$. В «Наборы ресурсов» добавьте *resourcePool*. Число ресурсов на одну входную заявку установите в 1 (параметр справа от *resourcePool*), он определяет длину очереди в процессоре, но у нас длина очереди не будет влиять на работу блока (мы определили, что в блок приходит заявка, если только для нее есть ресурс). Остальные параметры оставьте неизменными.

Блок *resourcePool* требует установки только одного параметра «Количество ресурсов», установите для него значение N .

Запустите модель в режиме виртуального времени. Через несколько секунд некоторые из требуемых характеристик уже появятся в анимированном окне структуры (рис. 3).

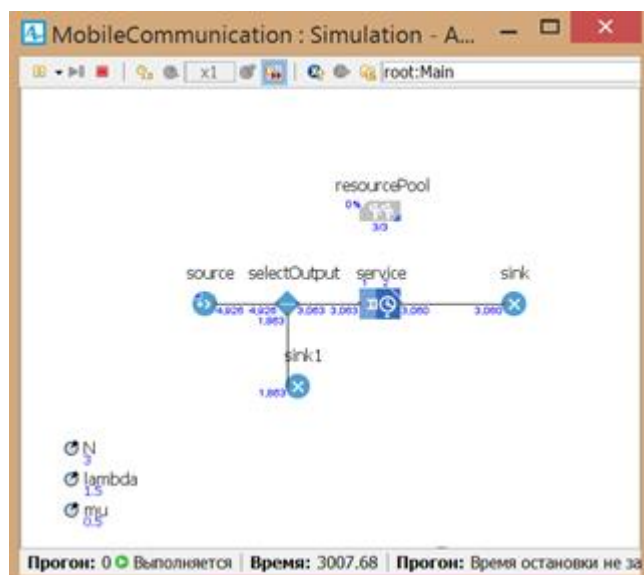


Рис. 3. Статистические характеристики модели

4.4. Статистические характеристики модели

Из рис. 3 видно, что всего пришло 4926 вызовов, 3063 из них обслужено, остальным (1863) отказано. Таким образом:

- относительная пропускная способность системы, т. е. средняя доля пришедших заявок, обслуживаемых системой, 3063 (62 %);
- вероятность отказа заявке в обслуживании 1863 (38 %).

Однако статистические характеристики системы играют лишь вспомогательную роль при анализе систем. Действительной целью моделирования является анализ возможной прибыли при различных значениях вектора входных параметров системы. В нашей задаче единственным таким параметром является n – число поддерживаемых каналов связи.

4.5. Проблема оптимизации. Целевая функция

Введем понятие Тарифного плана – аналога Соглашения о качестве сервиса (*Service Level Agreement*), которое широко используется в современных системах предоставления сервиса для расчета стоимости сервиса. В таком соглашении обычно оговариваются как стоимость сервиса при его надлежащем качестве, так и штраф поставщика сервиса за низкое

качество обслуживания. Пусть в соответствии с тарифным планом каждая обслуженная заявка приносит поставщику сервиса некоторый доход в соответствии с правилом посекундной тарификации после первой минуты, а за каждую отклоненную заявку поставщик сервиса должен заплатить штраф. Покупка оборудования АТС и содержание каждого канала обходятся в некоторую сумму, зависящую от максимального числа каналов АТС. Прибыль, полученную владельцем АТС, можно вычислить как доход за обслуживание вызовов минус штраф за отвергнутые вызовы и расходы на оборудование: $\text{прибыль} = \text{доход} - \text{штраф} - \text{стоимость оборудования}$

Увеличение числа каналов, обеспечивающих связь, уменьшает штраф и повышает доход, но при этом растут и расходы на оборудование. Варьируя число каналов при конкретных значениях остальных параметров системы, можно найти то оптимальное число каналов, которое принесет максимальную прибыль. Именно значение прибыли в этой задаче играет роль значения w функции полезности.

Ясно, что аналитически проблему эту не решить, так как все функции здесь неаналитические. Поскольку доход и штрафы можно подсчитать с помощью имитационной модели, то вся проблема является типичной проблемой имитационного моделирования. Прибегнем к оптимизации, использующей имитационную модель для нахождения значения заданного функционала (прибыли) при конкретных значениях параметров.

Доход от каждой обслуженной заявки зависит от времени соединения (например, поставщик сервиса хочет объявить повременную тарификацию после первой минуты). Любая подобная зависимость в *AnyLogic* легко задается с помощью функции. Отнеся суммарный доход (назовем его в модели *Income*) к временному периоду моделирования, получим доход в единицу времени (назовем его *gain*). Доход в единицу времени является случайной величиной, мы будем использовать его среднее значение.

Рассмотрим теперь расходы. Они состоят из штрафов за отвергнутые вызовы и затрат на оборудование. Выплачиваемый штраф нужно

подсчитывать для всех отвергнутых вызовов. Пусть *penalty* – средний штраф в единицу времени по всем отвергнутым вызовам. Это также случайная величина, среднее значение которой будет использоваться при поиске минимума целевой функции.

4.6. Наборы данных

В нашей задаче полученный в каждый момент времени средний доход *Gain* так же, как и многие другие значения в моделях систем массового обслуживания (длина очереди и время нахождения каждой заявки в очереди в некоторый момент времени и т. п.), будет только конкретной реализацией соответствующих величин, имеющих стохастическую природу. В общем случае все параметры производительности систем, функционирующих в условиях неопределенности, являются стохастическими. При анализе таких систем представляют интерес не конкретные значения этих параметров, которые являются случайными величинами, а статистические характеристики всего набора реализаций таких случайных величин – среднее, максимальное и минимальное значения, доверительный интервал и т. п.

Для удобства работы с коллекциями данных в *AnyLogic* есть раздел Палитра → Статистика, в котором находятся элементы, необходимые для сбора данных. С их помощью можно не только хранить все конкретные значения, но и автоматически подсчитать статистические характеристики.

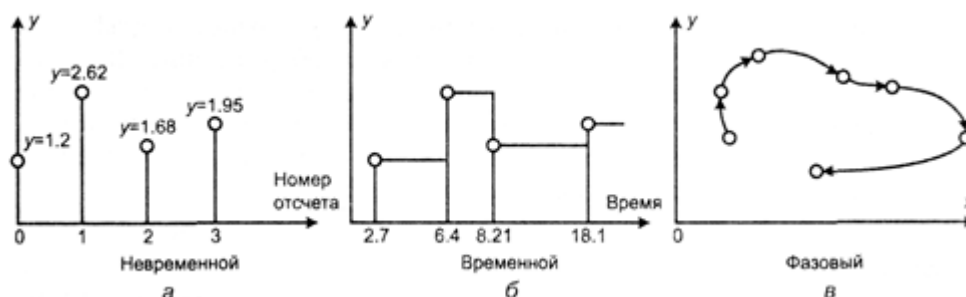


Рис. 4. Типы наборов данных

Наборы данных можно охарактеризовать тремя разными способами, показанными на рис. 4. *Невременной тип* – это просто последовательность

реализаций некоторой случайной величины. Статистические характеристики такого набора данных определяются стандартным образом. Совершенно другой тип наборов данных – *временной*. Для временных наборов данных хранятся не только каждое конкретное значение случайной величины, но и момент времени, в который это значение было установлено. При подсчете статистических характеристик такого набора данных учитывается длительность временного интервала в свойствах элемента, в течение которого случайная величина имела конкретное значение, то есть значение случайной величины учитывается с весом, равным этому временному интервалу. *Фазовый набор данных* – это последовательность пар значений, из которых первое понимается как аргумент, а второе – как значение функции от этого аргумента. Очевидно, что время ожидания в очереди представляется невременным набором данных. С другой стороны, длина очереди – это *временной набор данных*.

Если y – случайная величина, для которой нам хочется подсчитать ее статистические характеристики, то в модели нужно определить y как «статистику», и в момент, когда y примет очередное значение y , вызвать необходимую функцию. Полный список методов, доступных для получения статистической информации случайной величины, приведен в Справке *AnyLogic*.

Для сбора реализаций случайных величин *gain* и *penalty* введите статистики с этими именами, выбрав в Палитре раздел «Статистика» и в нем, элемент «*статистика*» сбора данных. Кроме имени никаких других параметров в окне свойств этих объектов менять не нужно.

4.7. Доход от обслуживания вызовов

В нашей модели сделаем следующие добавления. Все цены будем считать в у. е., единица модельного времени будет соответствовать одной минуте реального времени.

Для подсчета длительности состоявшегося телефонного разговора добавим нового агента, которого назовем *Call*. Этот агент будет иметь два параметра: *tStart* и *tFinish*.

В свойствах блока *source* есть параметр «Новый агент», выберем, используя выпадающее меню, созданный агент *Call*.

В блоке *service* параметр *onEnter* определим как:

```
((Call)agent).tStart=time( );
```

а параметр *onExit* как:

```
((Call)agent).tFinish=time( );
```

Операция кастинга имеет меньший приоритет, чем точка, поэтому, это выражение нужно взять в скобки, чтобы затем использовать «точечную» операцию доступа к данным этого объекта.

Теперь после обработки каждая заявка будет хранить момент начала и момент конца разговора. Эти значения нужны для того, чтобы начислить плату за обслуживание заявки.

Зададим теперь правило расчета за состоявшиеся телефонные переговоры.

Его удобно определить алгоритмической функцией. Создайте новую алгоритмическую функцию и назовите ее *callPrice* (Палитра → Агент → функция). У нее должен быть один параметр типа *double*, назовите его *t*. Саму функцию в свойствах элемента «Тело функции» определите так:

```
return t<=1 ? minPrice : t*minPrice;
```

Функция *callPrice(t)* определяет стоимость разговора так: если длительность *t* разговора не больше одной минуты, то взимается плата за одну минуту в размере *minPrice*, если больше, то *t*minPrice*. Параметр *minPrice* (цена минуты разговора) задайте как вещественный параметр модели со значением 0.12.

Введите в модель вещественную переменную *Income*, суммирующую полученный с течением времени доход от предоставления соединения (с

начальным значением 0). Каждый раз, как обслуженная заявка попадает в блок *sink*, величина *income* должна возрасти на величину:

$$callPrice(((Call)agent).tFinish - ((Call)agent).tStart);$$

Как только получено новое значение суммарной величины дохода *Income*, можно вычислить новую реализацию среднего дохода *gain*, которая равна $Income/time()$. Поэтому в поле свойств с именем «При входе» элемента *sink* следует вставить следующие операторы:

$$Income += callPrice(((Call)agent).tFinish - ((Call)agent).tStart);$$
$$gain.add(Income/time());$$

4.8. Штраф за необслуженные вызовы

В соответствии с тарифным планом за каждый необслуженный вызов поставщик сервиса платит фиксированный штраф. Примем значение этого штрафа 1 у. е. Подсчитаем сумму штрафа за все пропущенные вызовы. Введем новый параметр модели *penaltyPercall* (штраф за необслуженный вызов) со значением 1.0. Кроме того, введем в модель переменную *Penalties* для подсчета полной суммы штрафа с начальным значением 0. Для получения всей суммы штрафа каждый раз, как только заявка отбрасывается, следует увеличить суммарное значение штрафа *Penalties* на величину *penaltyPercall*, а также добавить в набор данных *penalty* ее новую реализацию, равную $Penalties/time()$.

Все необслуженные заявки в модели направляются в блок *sink1*. Для подсчета штрафа нам не нужны никакие параметры новой пришедшей в этот блок заявки, важен лишь сам факт прихода заявки. Поэтому для учета штрафа в поле «При входе» блока *sink1* вставим операторы:

$$Penalties += penaltyPercall;$$
$$penalty.add(Penalties/time());$$

Напомним, что переменная *Penalties* принимает значение общего штрафа, в то время как набор данных *penalty* – это случайная величина, равная среднему штрафу за единицу времени.

4.9. Приведенная стоимость оборудования

Определим теперь функцию зависимости затрат на оборудование от числа каналов. Отнеся затраты на оборудование к периоду морального устаревания оборудования (например, 5 лет), получим значение приведенных капитальных затрат.

Пусть для организации АТС может быть закуплено оборудование разных типов, так что стоимость поддержания одного канала зависит от числа каналов сложным образом. Пусть простая АТС, обслуживающая до 10 каналов одновременно, вместе с затратами на поддержку каналов в течение 5 лет стоит 0.012 у. е. в пересчете на 1 минуту, если использовать все 10 каналов, но только 0.01 у. е, если использовать 1 канал. Более продвинутая АТС, обслуживающая до 50 каналов, стоит 0.05 у. е. (в пересчете на 1 минуту обслуживания в течение 5 лет), но если использовать только 11 ее каналов, стоит 0.013 у. е. и т. п.

Учет затрат на оборудование и его обслуживание, зависящих нелинейным образом от числа каналов n , выполним с помощью табличной функции, которая легко определяется в *AnyLogic*. Назовем табличную функцию (Палитра → Агент → табличная функция), дающую значение затрат на оборудование от числа каналов, приведенное к минуте времени, *equipmentPrice*. Заполним табличные данные *equipmentPrice* согласно рис. 5. В свойствах таблицы можно выбрать как закон интерполяции между заданными таблично значениями, так и правила принятия решения, когда аргумент выходит за обозначенные пределы. Определим эти законы разумным образом, например, простой линейной интерполяцией и экстраполяцией, если аргумент выходит за пределы.

Интерполяция: Линейный	
Если аргумент выходит за пределы: Экстраполяция	
Табличные данные	
Аргумент	Значение
1	0.01
10	0.012
11	0.013
50	0.05
51	0.05
100	0.08

Интерполяция: Линейный	
Если аргумент выходит за пределы: Экстраполяция	
Табличные данные	
Аргумент	Значение
50	0.05
51	0.05
100	0.08
101	0.08
1000	0.3

Рис. 5. Табличная функция, задающая стоимость оборудования

Расходы на оборудование при числе каналов n , приведенные к единице времени, теперь можно подсчитать вызовом функции $equipmentPrice(N)$.

4.10. Целевая функция

Средний доход в нашей модели определяется случайной величиной $gain$, средний штраф определяется случайной величиной $penalty$, стоимость оборудования, уже приведенная к единице модельного времени, определяется функцией $equipmentPrice(N)$. Для подсчета прибыли (в единицу времени) введем в модель функцию $BenefitMean$ (среднюю прибыль), определяемую формулой так:

$$return\ gain.mean() - penalty.mean() - equipmentPrice(N).$$

Здесь $gain.mean$ и $penalty.mean$ – математические ожидания величин $gain$ и $penalty$.

Запустим модель. Можно видеть, что прибыль в единицу времени (переменная $BenefitMean$) при установленных параметрах отрицательна. Изменим значение n , установив его 20. Прибыль стала положительной, порядка 0.35 у. е. в минуту. При числе каналов n , равном 100, прибыль становится меньше. Следовательно, существует оптимальное число каналов, обеспечивающее максимальную прибыль.

Для нахождения оптимального числа каналов, дающего максимальную прибыль, можно либо перебрать все значения n от 1 до, например, 100 в эксперименте для варьирования параметров, либо использовать эксперимент

с оптимизацией. Оптимизационный эксперимент в *AnyLogic* позволяет найти такие значения параметров модели, при которых обращается в минимум или максимум некоторая определенная пользователем целевая функция. Значения целевой функции подсчитываются в *AnyLogic* каждый раз по окончании очередного выполнения модели, и алгоритм оптимизации автоматически выбирает новые значения параметров для очередного запуска модели. Для оптимизации пользователь должен в соответствующих окнах задать функционал, который следует минимизировать либо максимизировать, задать параметры и ограничения их диапазона, в которых должна выполняться оптимизация, а также указать ограничения, определяющие класс допустимых решений. Затем пользователь должен запустить оптимизацию, и пакет *OptQuest* будет использовать метаэвристику рассеянного поиска для выбора очередных значений входных параметров на основании значений целевой функции, полученных на предыдущих прогонах модели.

Щелчком правой кнопкой мыши на *MobileCommunication* и в появившемся окне выберем создать эксперимент. Тип эксперимента указываем оптимизация. В окне свойств эксперимента установим *root.BenefitMean()* в качестве целевой функции, которую нужно максимизировать. Изменяемым параметром является *n*: верхнюю границу *n* установим 100 (значение Макс) будем идти с шагом 1 от единицы.

После настройки параметров, в свойствах оптимизационного эксперимента нажмите кнопку «Создать интерфейс».

Запустим оптимизационный эксперимент: наилучшее число каналов 54, максимальная прибыль поставщика сервиса в этом случае будет 0.404 у. е. в минуту (рис. 6).

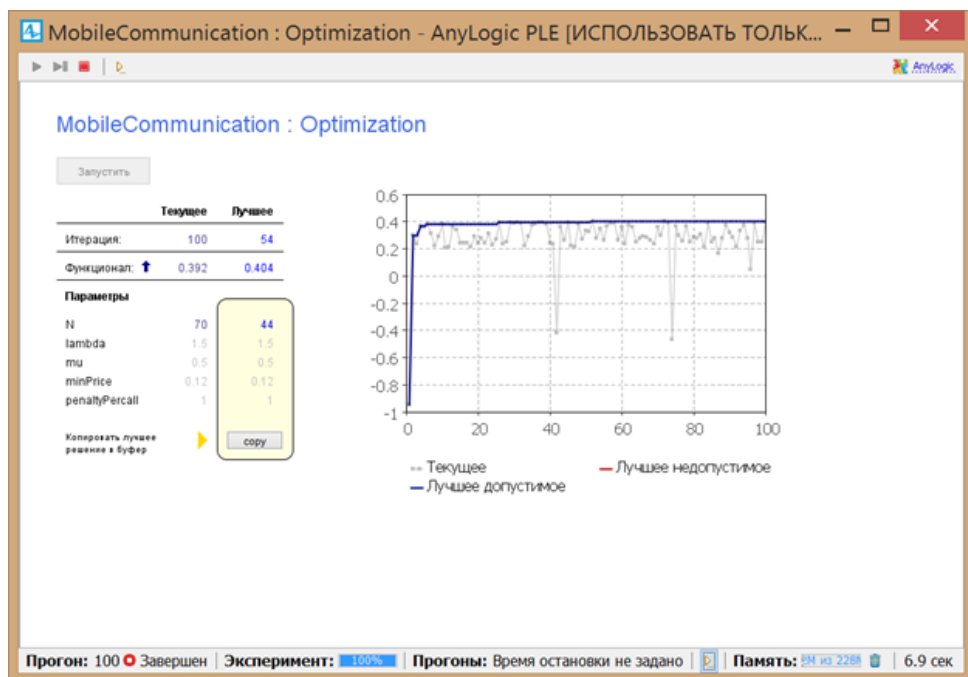


Рис. 6. Результат оптимизационного эксперимента

4. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

На построенной модели АТС проведите оптимизационный эксперимент в соответствии с параметрами вашего варианта.

Задайте параметры модели согласно вашему варианту (номер по списку группы).

Таблица 1.
Параметры модели АТС

Вариант №	Штраф за необслуженный вызов	Стоимость минуты разговора
1	1	0.05
2	1.05	0.06
3	1.06	0.07
4	1.07	0.08
5	1.08	0.09
6	1.09	0.10
7	1.10	0.11

8	1.11	0.12
9	1.12	0.13
10	1.13	0.14
11	1.14	0.05
12	1.15	0.06
13	0.99	0.07
14	0.98	0.08
15	0.97	0.09
16	0.96	0.10
17	0.95	0.11
18	0.94	0.12
19	0.93	0.13
20	0.92	0.14

Формой отчета по данной лабораторной работе является построенная в системе *AnyLogic* модель.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается технология построения оптимизационных моделей?
2. Что такое целевая функция и какова ее роль при построении модели? Что является целевой функцией в модели АТС?
3. В чем заключается смысл проведения оптимизационного эксперимента?
4. Какие параметры определяют в модели АТС штраф за необслуженный вызов, цену минуты разговора?
5. Как определяется значение затрат на оборудование от числа каналов?

6. Для чего в модели используется линейная интерполяция?
7. Какова роль функции *BenefitMean* при моделировании и построении оптимизационного эксперимента?
8. Как по результатам оптимизационного эксперимента можно проанализировать максимальное значение прибыли, отсутствие штрафов за необслуженные вызовы?

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Campbell, S.L. Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4. / Campbell S.L., Chancelier J.P., Nikoukhah R. – 2-е изд. – New York: Springer, 2010. – 329 с.
2. Боев, В.Д. Компьютерное моделирование [Электронный ресурс] / В.Д. Боев, Р.П. Сыпченко. – 2-е изд. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 525 с. – Режим доступа: <http://www.iprbookshop.ru/73655.html>.
3. Введение в математическое моделирование [Электронный ресурс]: учебное пособие / В.Н. Ашихмин [и др.]. – М.: Логос, 2016. – 440 с. – Режим доступа: <http://www.iprbookshop.ru/66414.html>.
4. Тупик, Н.В. Компьютерное моделирование [Электронный ресурс]: учебное пособие / Н.В. Тупик. – 2-е изд. – Саратов: Вузовское образование, 2019. – 230 с. – Режим доступа: <http://www.iprbookshop.ru/79639.html>.
5. Осоргин, А.Е. AnyLogic 6. Лабораторный практикум. / А.Е. Осоргин. – Самара: ПГК, 2011. – 100 с.
6. Карпов, Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 /Ю. Карпов. – СПб.: БХВ-Петербург, 2005. – 400 с.

7. Лоу, А.М. Имитационное моделирование /А.М.Лоу, В.Д.Кельтон. – 3-е изд. – СПб.: Питер, 2004. –847 с.
8. Маликов, Р.Ф. Практикум по моделированию сложных систем в среде AnyLogic 6 [Электронный ресурс]: учебное пособие / Р.Ф.Маликов. – Уфа: Издательство БГПУ, 2013. –296 с. – Режим доступа: https://www.anylogic.ru/upload/Books_ru/Практикум_по_ИМ_16-04-14.pdf
9. Каталевский, Д. Ю. Основы имитационного моделирования и системного анализа в управлении [Электронный ресурс]: учебное пособие / Д. Ю. Каталевский. – 2-е изд. – М.: Издательский дом «Дело» РАНХиГС, 2015. – 496 с. – Режим доступа: https://www.anylogic.ru/upload/pdf/katalevsky_osnovy_imitatsionnogo_modelirovania.pdf

Компьютерное моделирование. Выполнение оптимизационного эксперимента в программном комплексе Anylogic: методические указания к выполнению лабораторной работы № 10 для студентов очной формы обучения по направлению подготовки 09.03.01 «Информатика и вычислительная техника»

ТРУБАКОВА АННА АЛЕКСЕЕВНА
ТРУБАКОВ АНДРЕЙ ОЛЕГОВИЧ

Научный редактор Д. А. Коростелев
Компьютерный набор А.А. Трубакова
Иллюстрации А.А. Трубакова

Подписано в печать __.__.__. Усл.печ.л. 1,16 Уч.-изд.л. 1,16

Брянский государственный технический университет
241035, Брянск, бульвар 50 лет Октября, 7 БГТУ
Кафедра «Информатика и программное обеспечение», тел. 56-09-84

Сопроводительный лист на издание в авторской редакции

Название работы Компьютерное моделирование. Выполнение
оптимизационного эксперимента в программном комплексе Anylogic:
методические указания к выполнению лабораторной работы № 10 для
студентов очной формы обучения по направлению подготовки 09.03.01
«Информатика и вычислительная техника»

Актуальность и соответствующий научно-методический уровень
подтверждаю _____

(подпись научного редактора)

Рукопись сверена и проверена автором _____

(подпись автора)

Рекомендуется к изданию _____

(подпись заведующего кафедрой)