

Краснощекова Алена Игоревна

студентка

Князев Антон Андреевич

студент

ФГБОУ ВО «Национальный исследовательский Московский
государственный строительный университет»
г. Москва

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ЗАДАЧИ КОММИВОЯЖЕРА СРЕДСТВАМИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ANYLOGIC

***Аннотация:** статья посвящена описанию способа решения задачи коммивояжера с помощью программного комплекса для имитационного моделирования AnyLogic. Решение представлено на примере формирования сети доставок строительных материалов от завода к складам с целью оптимизации парка транспортных средств, имеющих в наличии у предприятия.*

***Ключевые слова:** логистика, транспортная логистика, задача транспортной логистики, задача коммивояжера, AnyLogic, имитационное моделирование, решение задачи коммивояжера.*

Имитационное моделирование (ИМ) дает возможность ясно и четко описывать логический процесс задачи от ее постановки до получения результата. ИМ является разновидностью математического моделирования.

Рассмотрим практическую ситуацию. С завода по изготовлению металлоконструкций в Зеленограде необходимо доставить товар с помощью грузовиков на склады в города России (Санкт-Петербург, Ярославль, Тверь, Смоленск, Тула, Рязань, Владимир, Липецк, Воронеж, Казань, Самара). А также, нужно определить оптимальное количество фур для такого процесса.

Для решения данной задачи воспользуемся программным комплексом для имитационного моделирования – AnyLogic [1]. Начать следует с создания модели. Назовем ее «Доставка1».

Функционал программы позволяет подключить ГИС-карту. Эта функция дает возможность смоделировать процесс, приближенный к реальным условиям доставки.

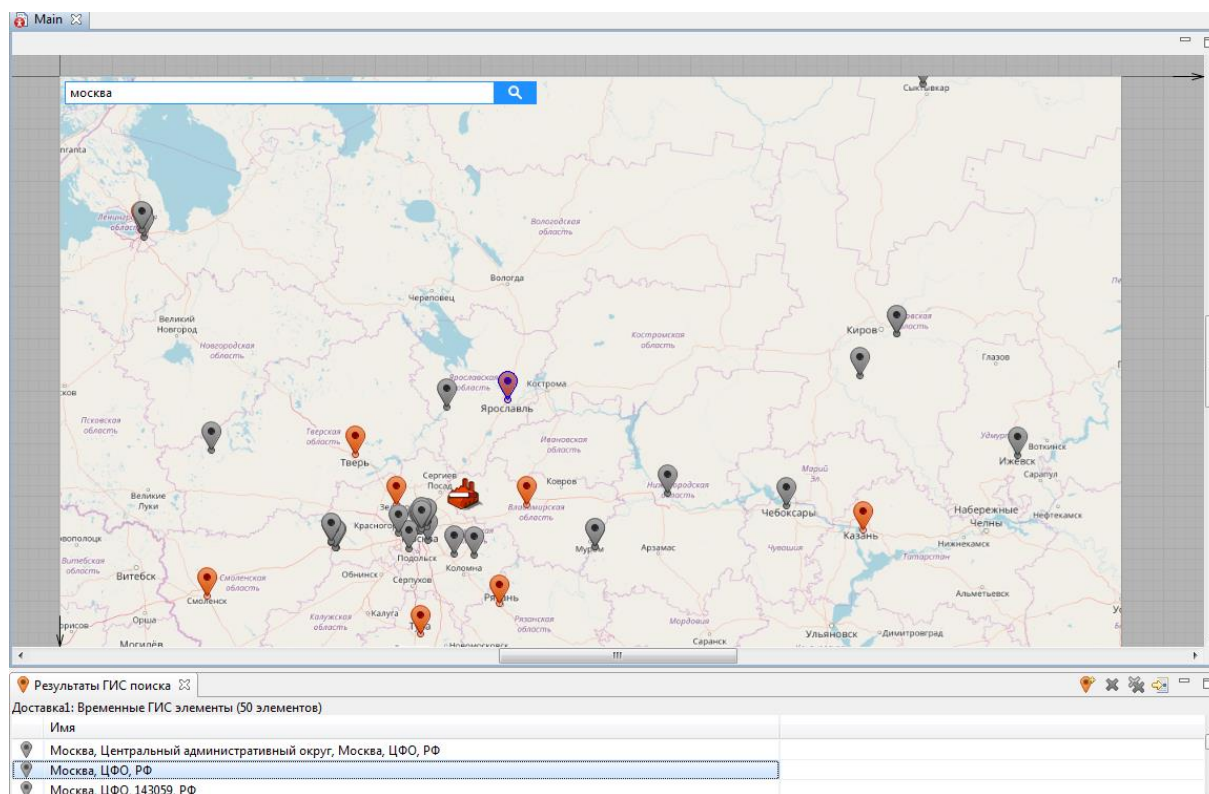


Рис. 1. Добавление точек на ГИС-карте

Для решения задачи сформируем все необходимые данные: добавим 11 точек городов-заказчиков и 1 точку-завод (рисунок 1). Для удобства размещения агентов в этих точках создадим коллекцию под названием «расположение Заказчиков».

Далее, нужно создать популяцию агентов на «main» для ранее найденных точек, которой присваивается название «Заказчики», и выбрать для нее анимацию. Для того, чтобы задать начальное количество агентов, записывается следующее выражение: *расположениеЗаказчиков.size()*

Эта функция возвращает количество элементов коллекции «расположение-Заказчиков» (11 штук). Далее, необходимо обозначить начальное положение этих точек. С помощью соответствующей вкладки для динамического значения узла нужно вписать выражение, показанное на рисунке 2.

Рис. 2. Редактирование свойств агента «заказчики»

Локальная переменная «index» возвращает номер популяции. По этому номеру осуществляется обращение к элементу из коллекции, которому будет соответствовать одна из точек-заказчиков на карте.

На следующем этапе создается предприятие, размещённое в Зеленограде, изготавливающее металлоконструкции. Необходимо выбрать новый тип агента с именем «Производство», который будет моделировать работу завода. Далее, через его свойства выбирается соответствующая точка на карте, где будет располагаться данное предприятие.

Следующий шаг – это создание агента «Фуры», с помощью которого будет осуществляться доставка продукции от завода к заказчикам. Выберем для него анимацию, зададим параметр «клиент», тип параметра – «Заказчик» (в котором будет храниться информация о том, кто сделал заказ и кому требуется доставка). Затем создадим целочисленный параметр «колвоФур», в котором будут содержаться соответствующие данные.

Теперь создается еще один тип агента, необходимый для моделирования процесса передачи заказа. Назовем его «Заказ», зададим параметр «потребитель» и тип «Заказчик».

Далее, необходимо вернуться к агенту «Фуры» и через его свойства задать начальное расположение на производстве соответствующей точкой на карте.

Так как изначальный размер популяции фур равен нулю, внутрь агента «Производство» добавляется блок процесса моделирования «resourcePool», его следует переименовать в «фуры» и задать значения количества, типа и остальных свойств в соответствии с рисунком 3.

Рис. 3. Создание параметров количества грузовиков

Далее, задается поведение агента «Заказчик», а именно сам процесс заказа металлических конструкций. Для этого используется диаграмма состояний: задаются состояния и интенсивность с помощью кода, представленного на рисунке

4. У «заказа» будет параметр – «Заказчик» который сформировал и отправил этот заказ. Для этого используется локальная переменная «this».

Теперь необходимо этот товар отправить. С помощью встроенной функцией AnyLogic «send», которая имеет два аргумента (что отправляем и кому отправляем), «отправляем» заказ. Затем, создается обратный переход от ожидания металлоконструкций к нормальной работе, который будет срабатывать при получении сообщения.

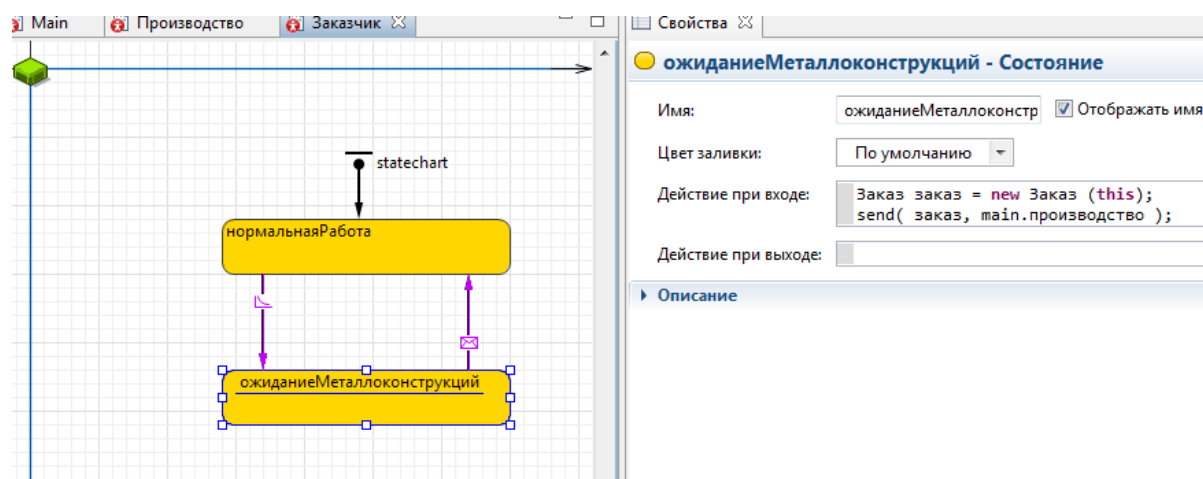


Рис. 4. Поведение агента «Заказчик»

Далее, следует вернуться обратно к агенту «Производство», чтобы задать логику обработки заказов. Для этого будет использоваться библиотека моделирования процессов. Точкой входа в процесс будет блок «Enter», в который будут помещаться заказы. Как только заказ поступает на производство, он встает в очередь на ожидание ресурсов, которыми являются фуры, выбираемые в блоке «seize». После «захвата ресурсов» необходимо осуществить погрузку и отправку фуры к заказчику. Для этого используется подпроцесс для ресурсов, который будет начинаться с блока «resourceTaskStart» с типом «Фуры». Затем, нужно выбрать блок «delay», с помощью которого будет моделироваться процесс «погрузки». Для задания времени погрузки используется равномерное распределение от двух до трех часов: $uniform(2,3)$.

Далее, выбирается блок «moveTo» для «отправки» фуры к заказчику и ему задается имя «кЗаказчику». Точкой назначения будет служить «Заказчик».

Данная информация хранится внутри фуры в параметре «клиент». Обратимся к нему: *agent.клиент*.

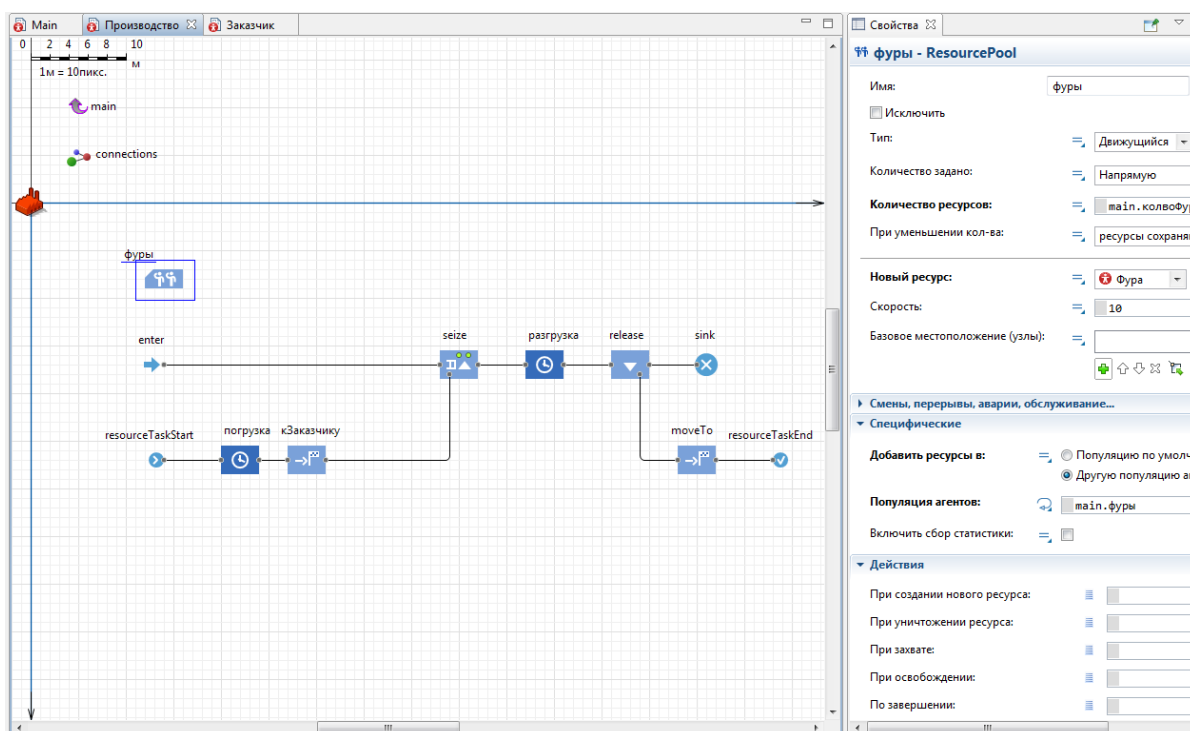
При захвате ресурса нужно передать информацию, содержащуюся внутри заказа, в фуру, чтобы она знала, куда именно нужно доставить товар. Соответственно, в свойствах блока «seize» на вкладке действия в поле «при захвате ресурса» нужно написать следующее выражение:

((Фура)unit).клиент = agent.потребитель;

Затем соединить блок «кЗаказчику» с нижним портом блока «seize».

Как только фура приехала к заказчику, необходимо осуществить разгрузку. Моделируется этот процесс так же с помощью блока «delay». Далее, задается равномерное время разгрузки от двух до трех часов.

После того, как закончился данный процесс, освобождается фура при помощи блока «release». Самого агента нужно отправить в блок «sink». Там он будет удален, а фура будет отправлена обратно в Зеленоград, для чего нужно опять воспользоваться блоком «moveTo». Зададим для него место назначения «агент», который расположен на «main». Это будет «Производство». Этот процесс необходимо закончить блоком «resourceTaskEnd» для того, чтобы ресурс вернулся к следующему агенту.



Теперь определим, какое количество грузовиков необходимо для оптимального распределения ресурсов. Для этого нужно запустить модель еще раз и перейти внутрь агента «Производство», чтобы посмотреть текущую загрузку ресурсов (рисунок 7).

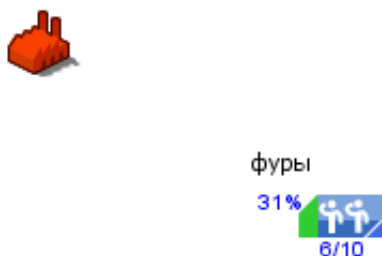


Рис. 7. Загрузка ресурсов

Из рисунка видно, что загрузка ресурсов колеблется в районе 30–32 процентов. Это означает, что количество фур, которое используется (в данном случае 10), избыточное.

Для того, чтобы оптимизировать число грузовиков, нужно создать эксперимент «Оптимизация» и задать свойства в соответствии с рисунком 8.

Optimization - Оптимизационный эксперимент

Имя: Optimization ☐ Исключить

Агент верхнего уровня: Main

Целевая функция: ☐ минимизировать ☒ максимизировать

root.производство.фуры.utilization()

☒ Количество итераций: 500

☐ Автоматическая остановка

Максимальный размер памяти: 512 M6

Параметры

Параметры:

Параметр	Тип	Значение			
		Мин.	Макс.	Шаг	Начальное
колвоФур	дискретный	1	10	1	

Модельное время

Ограничения

Требования

Требования (проверяются после "прогона" для определения того, допустимо ли найденное решение):

Вкл.	Выражение	Тип	Гран...
<input checked="" type="checkbox"/>	root.производство.фуры.utilization()	<=	0.85

Случайность

Рис. 8. Свойства эксперимента оптимизации

В свойствах задаются значения количества фур от 1 до 10, так как больше 10 не потребуется, а в качестве целевой функции используется средняя загрузка всех фур, ее необходимо максимизировать. «root» – это обращение к агенту «Производство», а «utilization» возвращает среднюю загрузку. Так же дополнительным условием будет то, что загрузка не должна превышать 85% для большей реалистичности задачи, потому что стоит учесть некоторое время, которое может потребоваться для ремонта транспортных средств, их технического обслуживания и т. д. Цель эксперимента – посмотреть, при каком значении ресурса загрузка будет максимальной (рисунок 9).

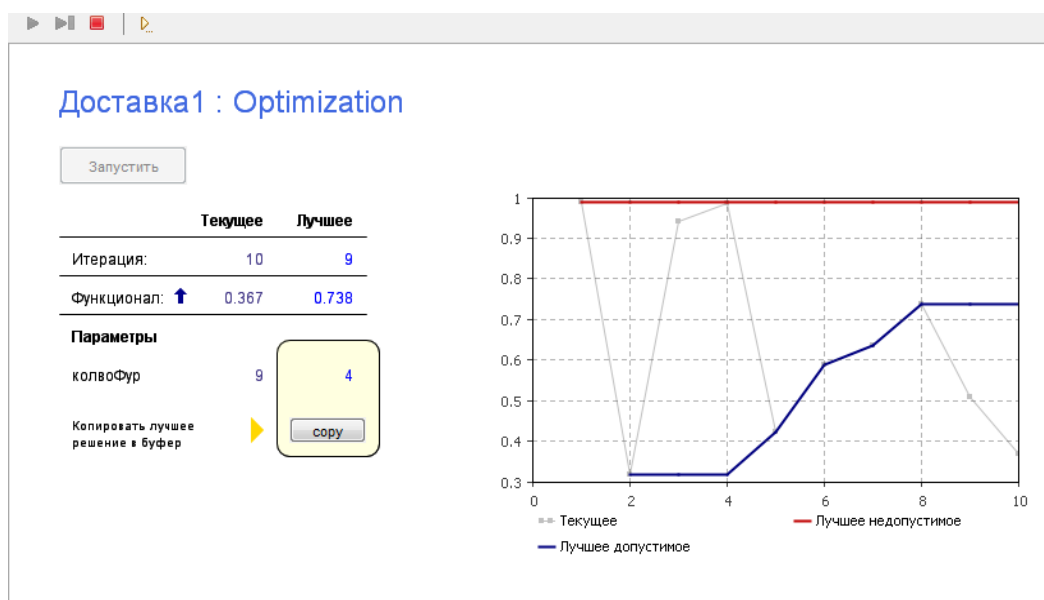


Рис. 9. Процесс оптимизации

Исходя из результатов этого эксперимента, видно, что лучшая итерация под номером 9, количество фур, достаточное для рационального использования в процессе доставки, равно 4, при этом загрузка ресурсов около 74%.

Далее, нужно подставить полученное значение в количество фур и запустить заново симуляцию процесса (рисунок 10).

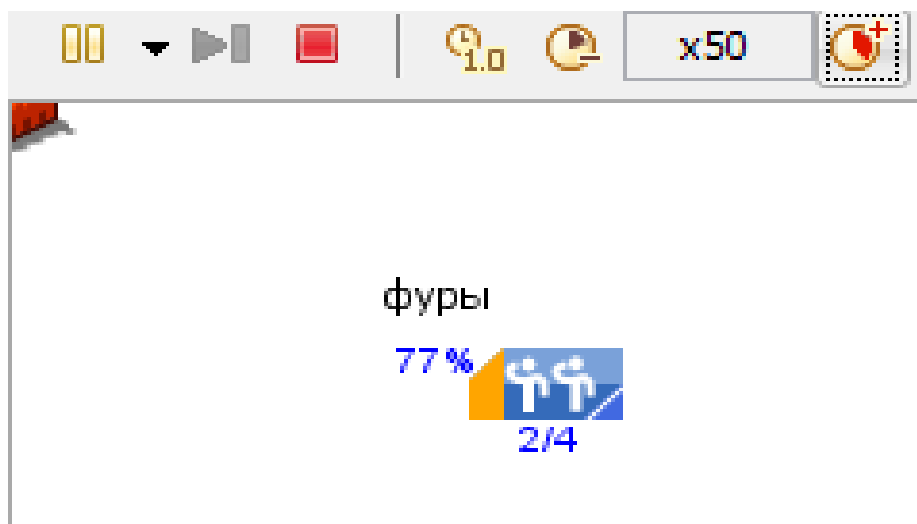


Рис. 10. Результаты

На рисунке 10 видно, что процент загрузки ресурсов колеблется в районе 77–80 процентов, следовательно, подставленное значение является оптимальным. Таким образом можно сделать вывод о том, что использование программного обеспечения Anylogic в моделировании каких-либо процессов позволяет воспроизвести событие с условиями, приближенными к реальности, а также добиться оптимального использования ресурсов, что способствует экономии финансовых средств. Данная тематика актуальна на сегодняшний день, а продукт исследования пользуется спросом практически во всех транспортных кампаниях и предприятиях.

Список литературы

1. AnyLogic. Многоподходное имитационное моделирование [Электронный ресурс]. – Режим доступа: <http://www.anylogic.ru/>
2. Борознов В.О. Исследование решения задачи коммивояжера // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. – 2009. – №2. – С. 147–151.
3. Мезенцев К.Н. Моделирование систем в среде AnyLogic 6.4.1: Учеб. пособие. Ч. 2. – М., 2011. – 108 с.
4. Харитонов А.М. Модели и методы применения и развития автоматизированных систем / А.М. Харитонов, Е.С. Сергеев, П.В. Пигачёв // Современные проблемы науки и образования. – 2015. – №1–1. – 398 с.