РЕФЕРАТ

Отчет 45 с., 22 рис., 11 табл., 12 источн., 7 прил.

СТОХАСТИЧЕСКИЕ СИМУЛЯЦИИ БИОЛОГИЧЕСКОЙ МОДЕЛИ СТАЦИОНАРНЫХ СООБЩЕСТВ

Объектом исследования является модель стационарных биологических сообществ.

Цель работы – исследование данных симуляций о численности популяции биологических сообществ на плато.

В процессе работы разрабатывался алгоритм остановки симуляции на плато, генерировалась база данных симуляций и проводилось первичное исследование применимости методов машинного и глубинного обучения в предсказании численности популяции на плато по параметрам симуляции.

Научная новизна работы заключается в оптимизации процесса проведения симуляций, расширении базы данных результатов симуляции и применении методов машинного и глубинного обучения в задаче поиска численности популяции на плато.

В результате исследования:

- была изучена предметная область исследования;
- был разработан и реализован алгоритм остановки симуляций на плато;
- были проведены симуляции, результаты которых были собраны в датасет;
- была показана возможность применения методов машинного и глубинного обучения для предсказания численности популяции на плато.

Результаты, полученные в ходе данной курсовой работы, могут быть использованы в дальнейших исследованиях модели стационарных биологических сообществ.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Таблица 1 – Сокращения и термины с определениями.

Термин / Сокращение	**Определение**
Выброс	Значение временного ряда, значительно отклоняющееся от общего тренда.
Простое экспоненциальное сглаживание	Подход для избавления от выбросов во временных рядах, при котором новое значение ряда рассчитывается исходя из значения данных в данный момент времени и результатов сглаживания за предыдущий моменты времени с затухающим коэффициентом. Формула для расчета значения ряда: $$\hat{y}_t = a * y_t + (1 - a) * \hat{y}_{t-1}$$
SES	Сокращение от Simple exponential smoothing (просто экспоненциальное сглаживание)
Датасет	Коллекция данных, представленная в удобном для хранения и чтения виде.
Ггиперпараметр	Параметр в машинном обучении, который задается до начала обучения модели и используется для управления этим процессом.
MSE	Сокращение от Mean squared error (среднеквадратическая ошибка). Для двух векторов размерности n ычисляется по следующей формуле: $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
Функция потерь	Функция, с помощью которой оценивается точность предсказанного результата. В качестве входных данных функция принимает правильный ответ и предсказанное значение, а на выходе выдаёт потерю (числовую характеристику отклонение предсказания от истинного ответа)
Блокнот	В контексте работы: Jupyter notebook – интерактивный блокнот позволяющий запускать код на Python.
Ансамбль моделей	Метод машинного обучения, в котором используются несколько моделей обучаются для решения предложенной

	проблемы и объединяются для улучшения общего результата.
Беггинг	Метод машинного обучения, при котором однородные модели обучаются на разных наборах данных, а затем их прогнозы объединяются путем усреднения.
Полносвязный слой	Слой нейронной сети, в котором входные нейроны связаны со всеми выводными нейронами.
Функция активации	Нелинейное преобразование, использующееся в нейронных сетях для избавления от линейности и извлечения более сложных нелинейных признаков.
ReLU	Сокращение от Rectified linear unit; функция активации имеющая следующую формулу: ReLU = max(0, x)
Learning rate	Коэфициент скорости обучения; гиперпараметр для градиентных алгоритмов, позволяющий скоростью управлять коррекции весов на каждой итерации .
Эпоха в машинном обучении	Обучение модели на всех данных датасета. Обычно полное обучение модели проводится в несколько эпох.

1. ВВЕДЕНИЕ

Методы математического моделирования применяются во многих областях науки в том числе и биологии. С развитием компьютерных наук появились появилось множество моделей описывающих развитие биологических сообществ, вытеснения одним видом других, предсказания эпидемий и т. д.

Одной из областей биологии, где оправдано применение компьютерных симуляций, является изучение сообществ растений. Моделирование в отличие от исследования реальных сообществ растений позволяет провести большое количество экспериментов за короткие сроки.

Наиболее приближенными к реальности являются пространственные модели, в которых учитывается поведение каждого индивида в отдельности, а не однородной в пространстве популяции в целом.

2. ПРЕДМЕТНАЯ ОБЛАСТЬ

2.1 Модель Дикманна-Лоу

В данной курсовой работе изучается модель Ульфа Дикманна [1], в которой рассматривается популяция растений со следующими условиями:

1) Организмы обитают в конечной области $\mathrm{A} \subset R^n$, $\mathrm{n} = 1, 2, 3$;
2) Каждый организм рассматривается как материальная точка (единственным отличием между двумя организмами является их положение в пространстве).

В данной модели существует 2 события, которые меняют состояние сообщества растений: рождение новых индивидов и смерть существующих. Данные события с заданной вероятностью могут происходить в любой момент времени.

Рождение нового организма в рамках модели подразумевает возникновение нового организма в произвольной точке пространства. Вероятность возникновения события зависит от вероятности того, что родитель воспроизведет новый организм (темпа рождаемости) и некоторой функции, зависящей от расстояния между родителем и потомком (ядра рождаемости). Последнее является математическим аналогом дальности «разбрасывания» семян растениями в реальном мире.

Вероятность наступления события смерти в модели складывается из двух составляющих: смертность от естественной среды (темп смертности) и смертность от конкуренции. При расчете вероятности смертности от конкуренции задается функция, учитывающая силу конкуренции и расстояние между конкурирующими индивидами (ядро конкуренции).

Таким образом модель имеет следующие параметры:

Таблица 2 – Параметры математической модели Ульфа Дикманна

Биологический параметр	**Описание**
$m(\xi)$	**Ядро рождаемости**, плотность вероятности возникновения нового потомка в точке x+ξ при условии, что родитель находится в точке x $\forall \xi\ \mathrm{m}(\xi) \geq 0: \int_{R^n} m(\xi)\, d\xi = 1;\ \lim_{x \to +inf} m(x) = 0;$ $\forall$ x, y $\in R^n$ ‖x‖ = ‖y‖ =>‖m(x)‖ = ‖m(y)‖
$w(\xi)$	**Ядро конкуренции,** $\forall \xi\ \mathrm{w}(\xi) \geq 0: \int_{R^n} w(\xi)\, d\xi = 1;$ $\lim_{x \to +inf} w(x) = 0;$ $\forall$ x, y $\in R^n$ ‖x‖ = ‖y‖ =>‖w(x)‖ = ‖w(y)‖ $\sum_{x' \in X} d'w(x - x')$ — плотность вероятности смерти в результате конкуренции организма в точке x от других организмов, X — множество точек всех организмов в пространстве
b	**Темп рождаемости**, плотность вероятности рождения индивида в любой точке пространства, b > 0
d	**Темп смертности**, плотность вероятности смертности от естественной среды, d ≥ 0
d'	**Сила конкуренции** индивидов, d' ≥ 0;

3. РАБОТА С СИМУЛЯЦИЯМИ

3.1 Описания генерации симуляций и их примеры

Используя репозиторий с исходным кодом нашей научной группы, были проведены симуляция, основанные на модели Дикманна-Лоу в одномерном пространстве.

Алгоритм симуляций описан на двух языках программирования. Для запуска и сохранения параметров используется язык R. Для ускорения выполнения симуляций для каждой эпохи выполняется код на C++. Эпохой называется один полный просчет всех одновременных событий во всех точках пространства.

В качестве параметров в симуляциях задается распределение ядер рождаемости и конкуренции, темпы рождаемости и конкуренции, сила конкуренции, длина рассматриваемого одномерного пространства, изначальная популяция. В данной работе рассматривались только нормальные распределения для ядер рождаемости и конкуренции с математическим ожиданием 0. Поэтому в качестве параметров для симуляций задавались соответствующие стандартные отклонения и радиус взаимодействия (для всех точек отдаленных от данной на расстояние большее радиуса взаимодействия сила конкуренции между индивидами считалась равной 0). Начальная популяция в данной работе задавалась как количество особей, а перед началом симуляции особи в случайном равномерно распределялись по всей протяженности пространства.

Таким образом у симуляций задавались следующие параметры:

Таблица 3 – Параметры симуляций.

Параметр симуляции	**Определение**
sd_b	$m(\xi)$~N(0, sd_b); Стандартное отклонение ядра рождаемости
sd_d	$w(\xi)$~N(0, sd_d); Стандартное отклонение ядра конкуренции
b	Темп рождаемости
d	Темп смертности
dd	d' Сила конкуренции
death_r	Радиус взаимодействия
area_length_x	Длина пространства
initial_pop	Изначальная численность популяции

В первом примере симуляции были заданы следующие значения параметров:

Таблица 4 – Значение параметров симуляций пример Рис1.

Параметр симуляции	**Значение**
sd_b	0.2
sd_d	1
b	1
d	0
dd	0.01
death_r	10
area_length_x	100
initial_pop	10

При заданных параметрах популяция в симуляции вышла на плато при значении около 1000 особей.

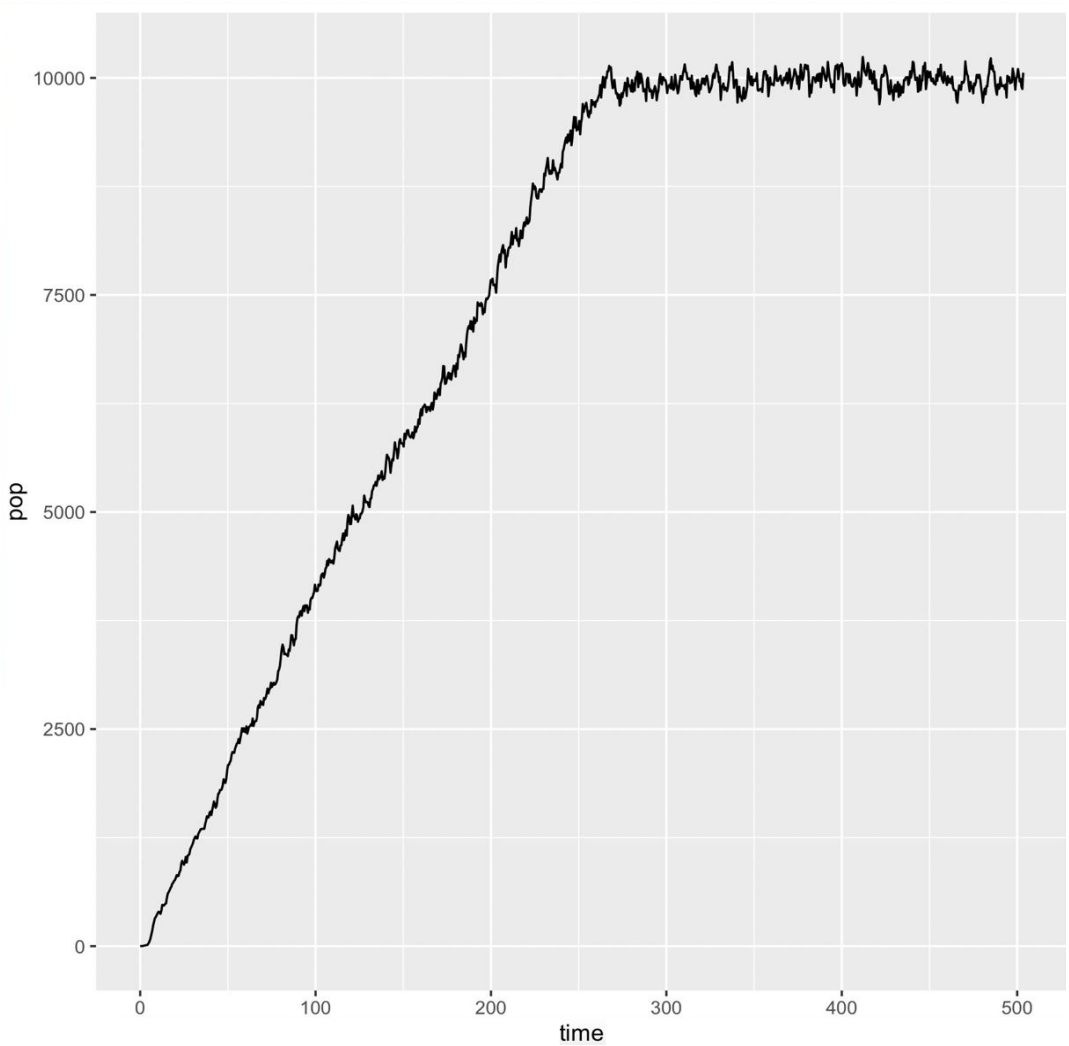


Рисунок 1 – Пример 1 графика численности популяции в симуляции.

Во втором примере параметры модели были изменены так, что за ограниченное при запуске программы количество эпох в симуляции график численности растений не успел достичь плато. В данном контексте под эпохой этап обработки симуляции (момент времени), когда для каждого индивида просчитывается вероятность его смерти или рождения им индивида, а далее в соответствии с данным критерием генерируется случайное событие.

Таблица 5 – Значение параметров симуляций пример Рис2.

Параметр симуляции	**Значение**
sd_b	0.2
sd_d	1
b	1
d	0
dd	0.1
death_r	10
area_length_x	100
initial_pop	10

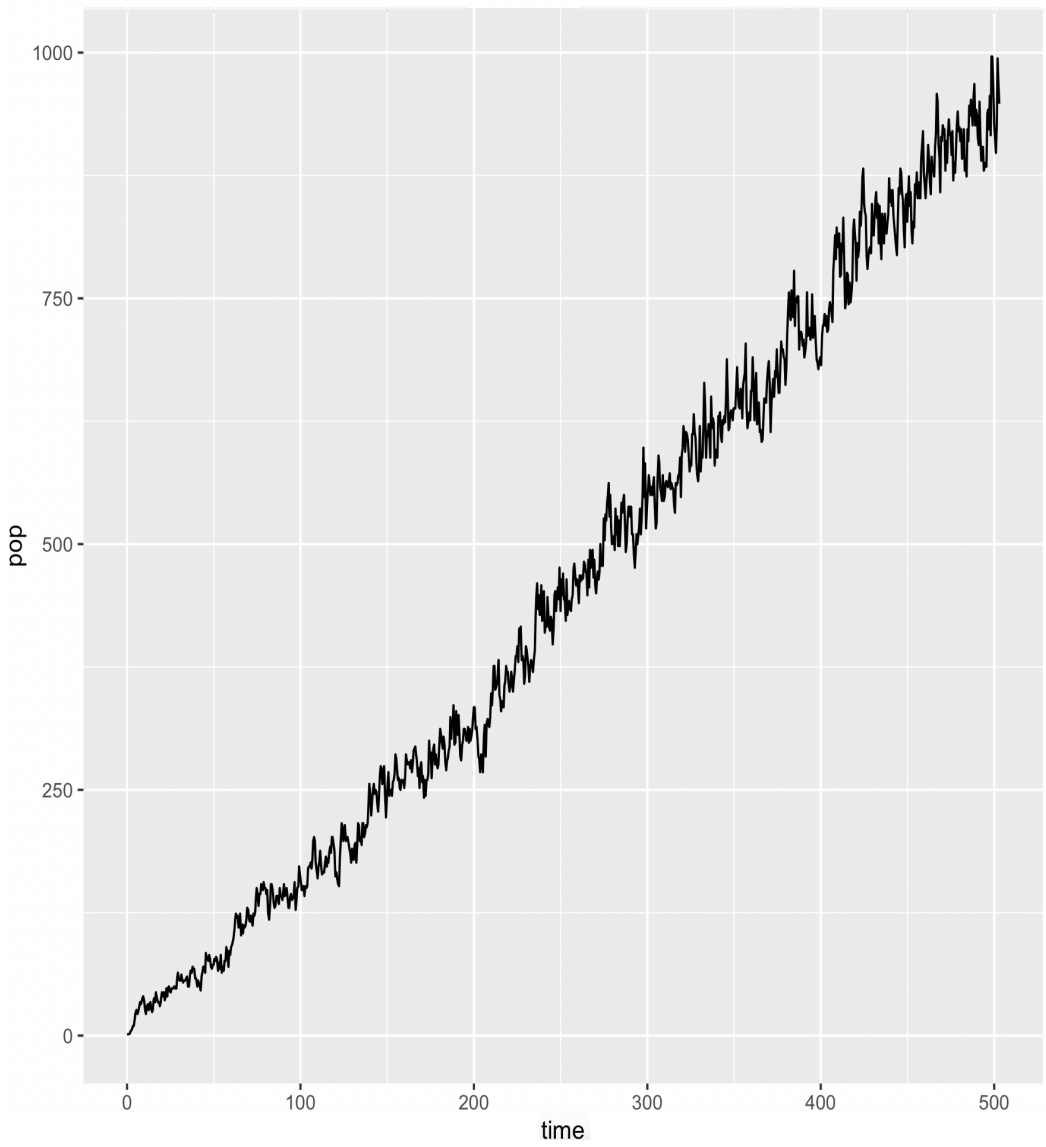


Рисунок 2 – Пример 2 графика численности популяции в симуляции.

Приведенные эксперименты демонстрируют одну из главных проблем в существующем подходе к проведению симуляций. Так как одним из важных критериев для завершения симуляции является достижение плато – состояния равновесия для данного вида растений, при котором смертность компенсируется рождаемостью, завершение симуляции для оптимального использование вычислительных ресурсов должно происходить после выхода графика плато. Таким образом одной из целей данной курсовой работы стало изучение и подбор алгоритма для программного определения выхода симуляции на плато, а также внесение соответствующих изменений в код генератора экспериментов.

3.2 Применение сглаживания на результатах симуляций

3.2.1 Обоснование применения сглаживания численности популяции

Так как в основе симуляций лежат стохастические процессы, график численности популяции имеет, в том числе и на плато, некоторые колебания вокруг ярко выраженного тренда. Поэтому для вычленения этого тренда и более точного определения момента выхода численности популяции на плато необходимо избавится от выбросов в данных. Одним из важных аспектов при подборе алгоритмов сглаживания данных была необходимость реализации метода в режиме реального времени, при отсутствии данных о конечной длительности симуляции (количестве эпох). Для этого были рассмотрены следующие алгоритмы сглаживания данных: скользящее среднее, простое экспоненциальное сглаживание, последовательное применение простого экспоненциального сглаживания. Алгоритмы и дополнительные иллюстрации подбора алгоритма сглаживания представлены в блокноте (Приложение С).

3.2.2 Метод скользящего среднего

Первым и самым простым алгоритмом сглаживания из примененных к данным симуляций был алгоритм скользящего среднего. При этом подходе к сглаживанию вычисление нового значения ряда сводится к вычислению среднего значения k предыдущих членов ряда:

$$\widehat{\boldsymbol{y}}_{\boldsymbol{t}} = \frac{\mathbf{1}}{\boldsymbol{k}} \sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{k}-\mathbf{1}} \boldsymbol{y}_{\boldsymbol{t}-\boldsymbol{i}}$$

где **y** – исходный ряд, $\widehat{\boldsymbol{y}}$ значения сглаженного ряда.

Для демонстрации работы алгоритма приведен пример сглаживания симуляции методом скользящего среднего с размерами окон 10, 50, 100 при общей длительности симуляции 1000 эпох.

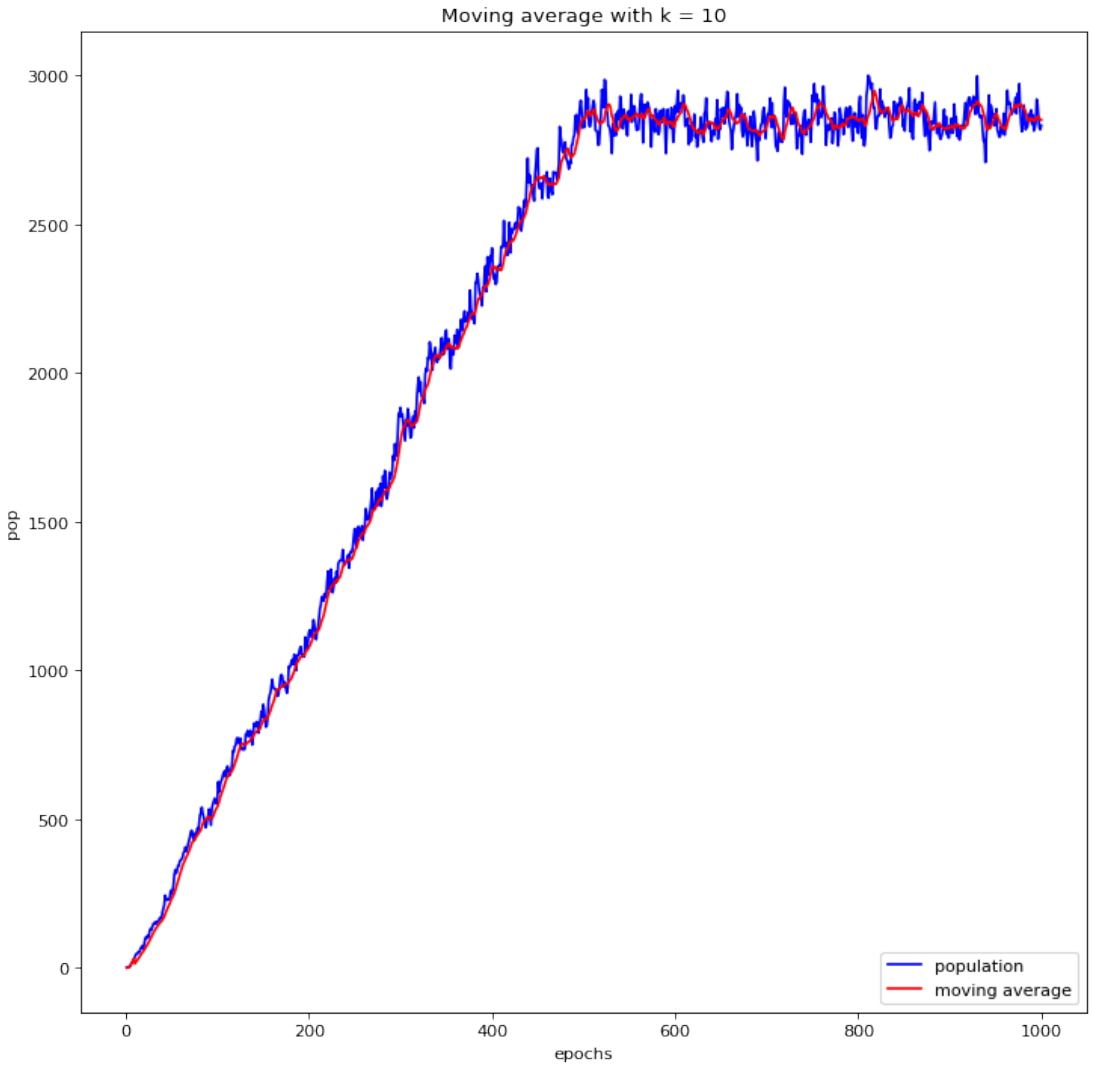


Рисунок 3 – Результат применения алгоритма скользящего среднего с размером окна 10 к данным симуляций.

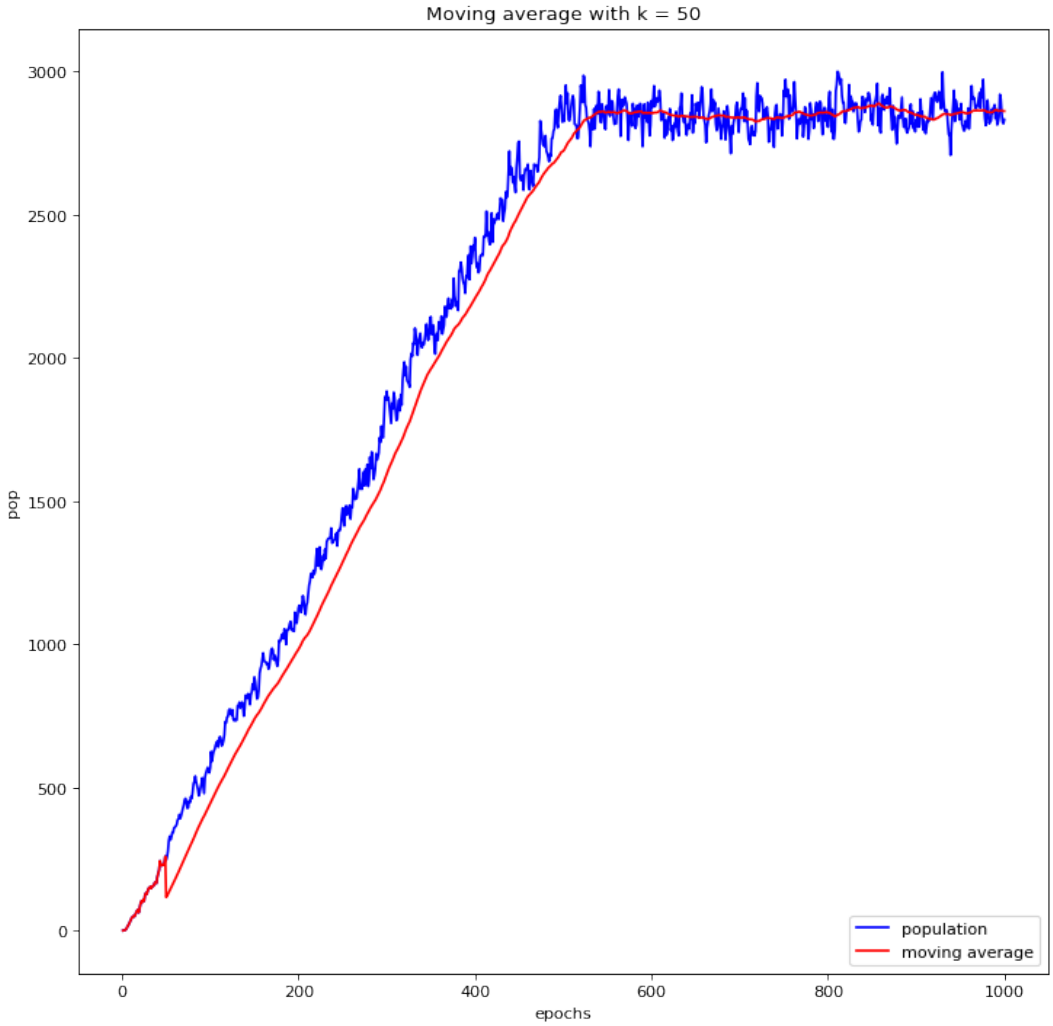


Рисунок 4 – Результат применения алгоритма скользящего среднего с размером окна 50 к данным симуляций.

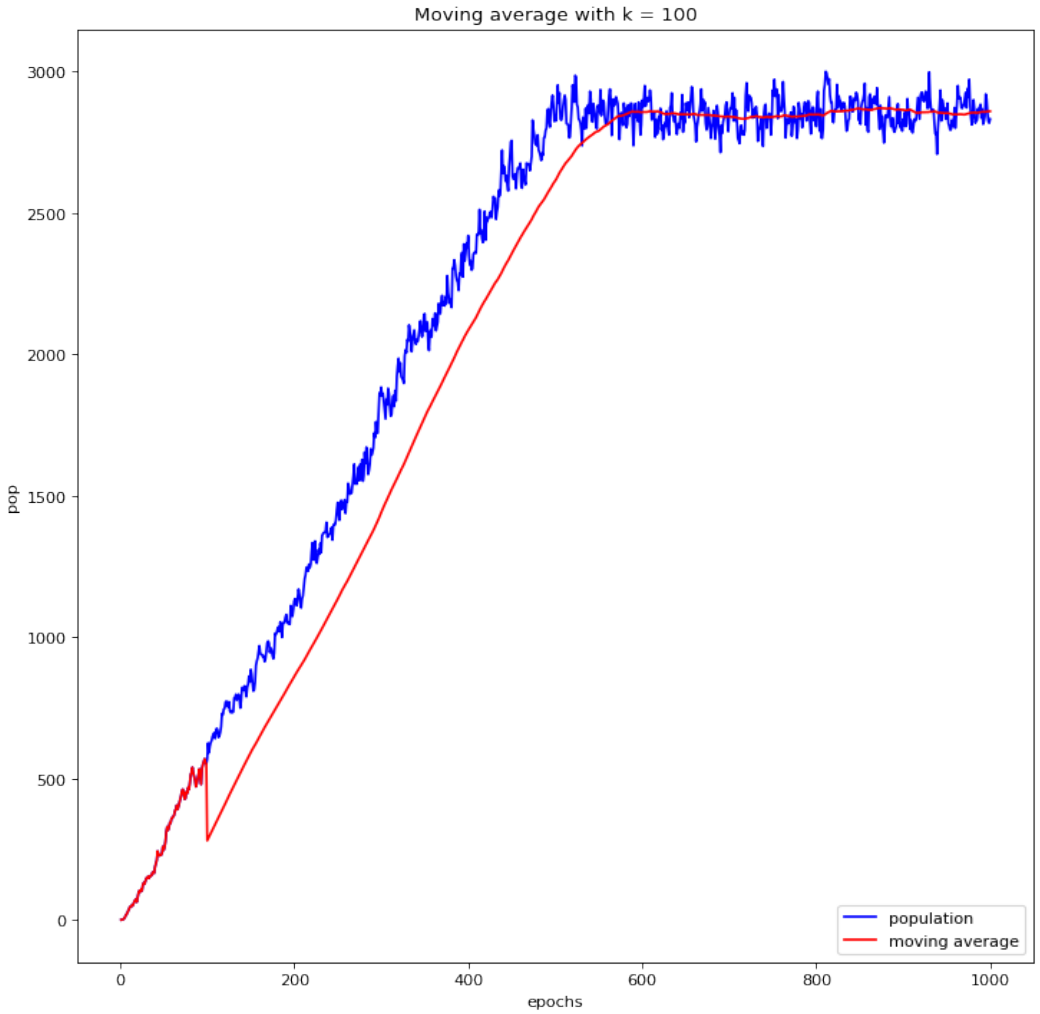


Рисунок 5 – Результат применения алгоритма скользящего среднего с размером окна 100 к данным симуляций.

Стоит заметить, что резкий скачек в начале графика вызван инициализацией значений первого окна соответствующими значениями симуляции и может быть устранен, например, взятием среднего из всех существующих на момент вычисления $\hat{y}_t$ элементов (t < k) для первого окна или другими модификациями алгоритма.

Визуальный анализ полученного при данном подходе к сглаживанию графика численности популяции позволяет заметить, что после применения алгоритма данные по-прежнему остаются зашумлены. Кроме того, еще одним существенным недостатком данного метода сглаживания данные является подбор параметра длины окна (k), который может зависеть от количества эпох симуляции и как видно по графикам сильно влияет на качество сглаживания. При этом в ходе экспериментов было выявлено значительное различие в длительности (количестве эпох перед выходом графика на плато).

Таким образом данный алгоритм является трудноreализуемым в режиме реального времени при отсутствии данных о длительности симуляции (количестве эпох) и

малоэффективным. По изложенным выше соображениям данный подход был исключён из рассмотрения.

3.2.3 Метод простого экспоненциального сглаживания

В следующем эксперименте был применен метод простого экспоненциального сглаживания (метод Брауна). При этом подходе новое значение ряда рассчитывается исходя из значения данных в текущий момент времени и результатов сглаживания за предыдущий моменты времени с затухающим коэффициентом ($\boldsymbol{a}$).

Формула для расчета значения ряда:

$$\widehat{\boldsymbol{y}}_{\boldsymbol{t}} = \boldsymbol{a} * \boldsymbol{y}_{\boldsymbol{t}} + (\mathbf{1} - \boldsymbol{a}) * \widehat{\boldsymbol{y}}_{\boldsymbol{t-1}}$$

При использовании данного подхода также необходимо подобрать значение одного параметра, но в отличие от скользящего среднего значение константы не зависит от длительности симуляции в эпохах.

Примеры экспоненциального сглаживания, примененного с коэффициентами $\boldsymbol{a}$ 0.1, 0.25, 0.5, 0.7:

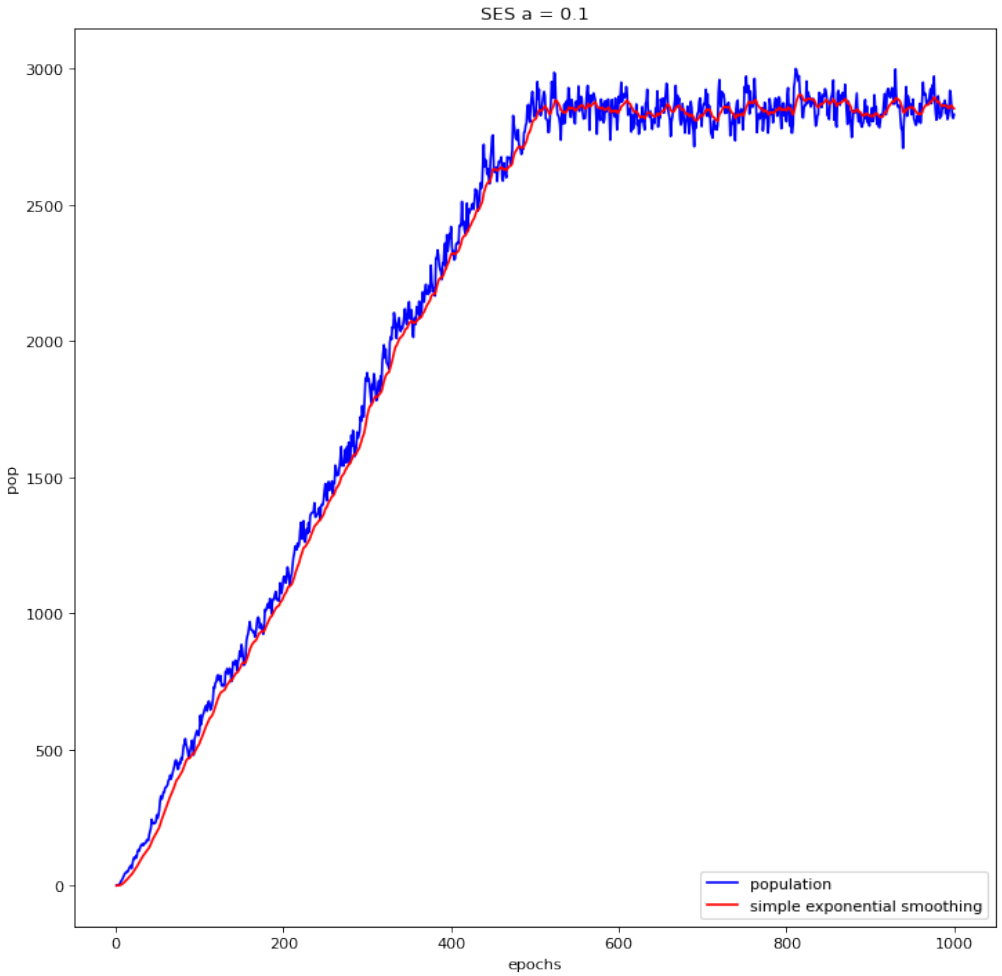


Рисунок 6 – Результат применения алгоритма простого экспоненциального сглаживания с коэффициентом a = 0.1 к данным симуляций.

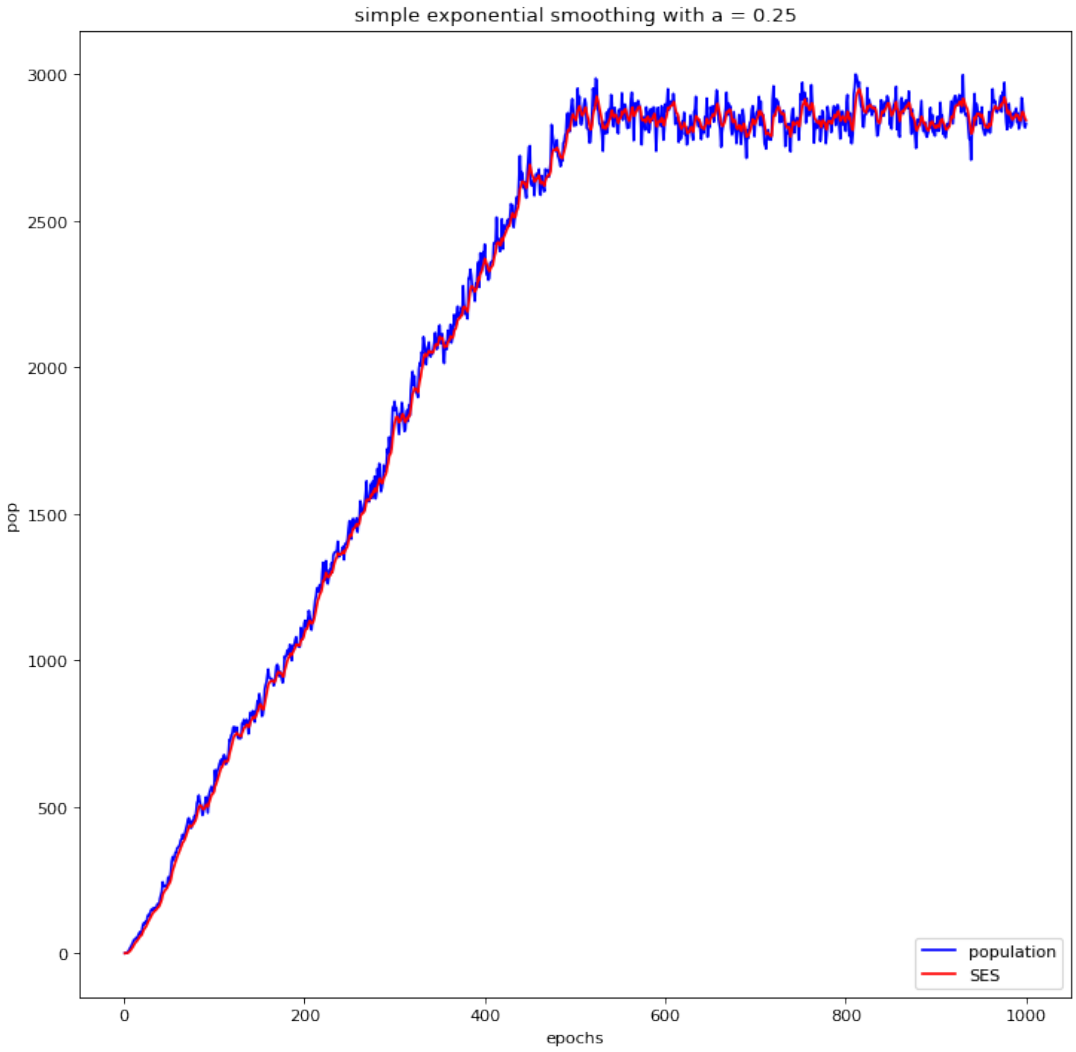


Рисунок 7 – Результат применения алгоритма простого экспоненциального сглаживания с коэффициентом a = 0.25 к данным симуляций.

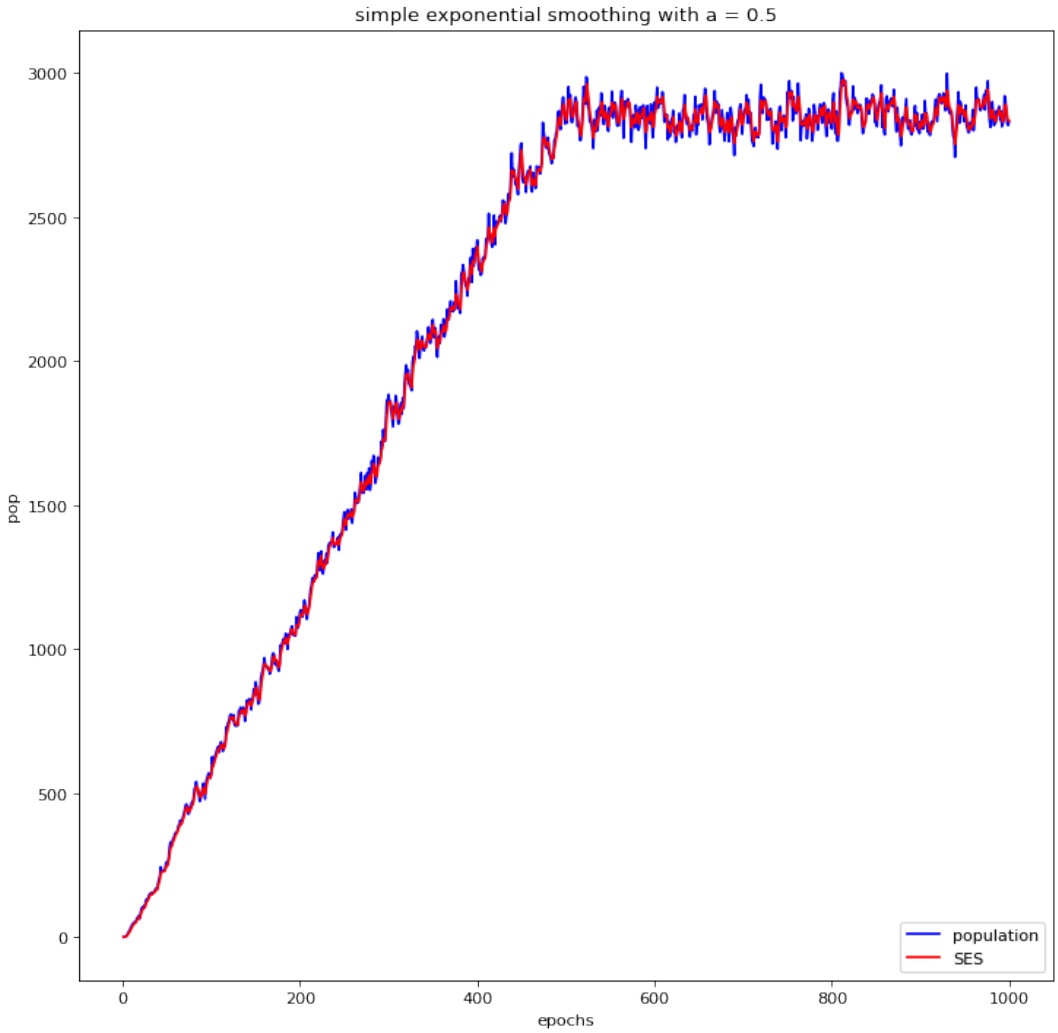


Рисунок 8 – Результат применения алгоритма простого экспоненциального сглаживания с коэффициентом a = 0.5 к данным симуляций.

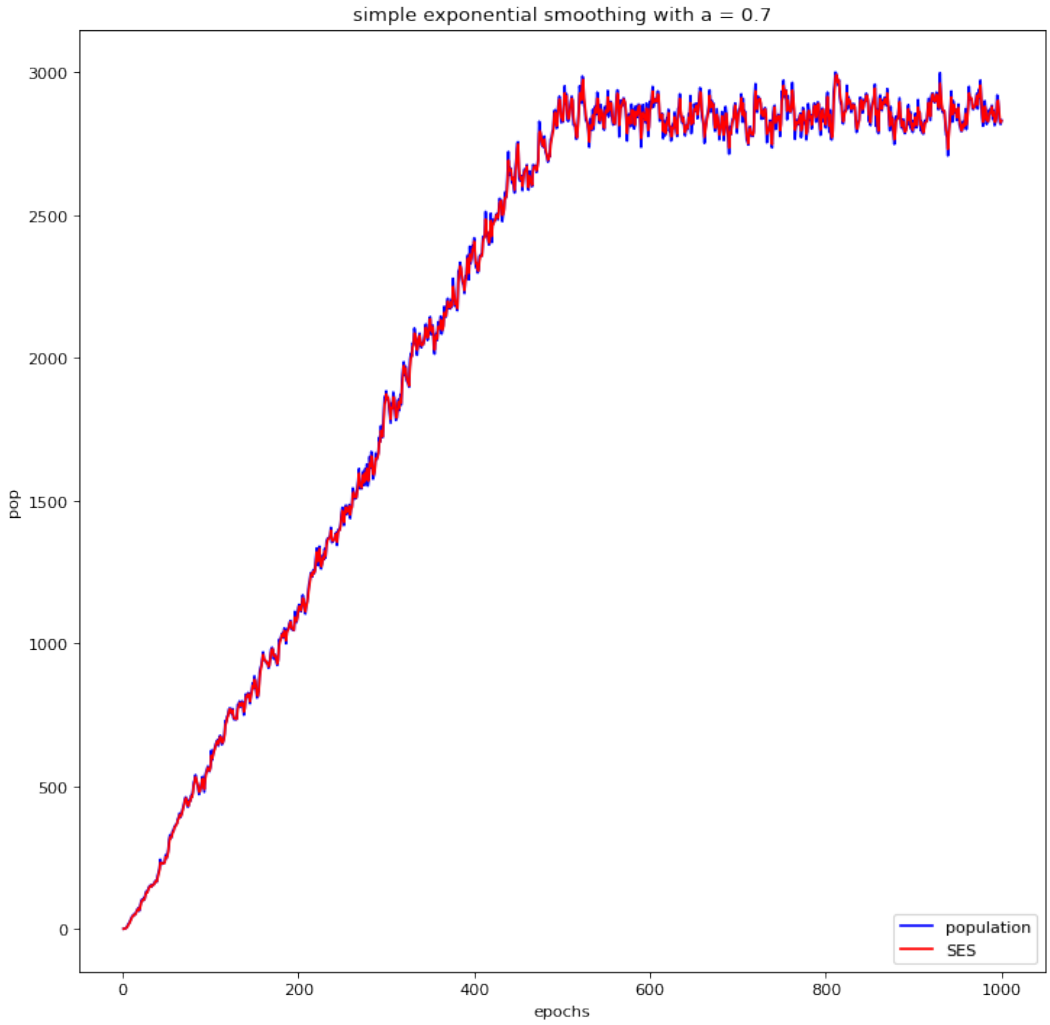


Рисунок 9 – Результат применения алгоритма простого экспоненциального сглаживания с коэффициентом a = 0.7 к данным симуляций.

Данного метода в ходе визуального анализа данных показал значительно превосходство над методом скользящего среднего. Наилучший результат достигается при значении a = 0.25. Кроме того данный алгоритм легко реализуем в условиях получения данных в реальном времени.

3.2.4 Метод последовательного простого экспоненциального сглаживания

Так как в ходе применение простого экспоненциального сглаживания рядов численности популяции была выявлена его высокая эффективность, был применен метод последовательного экспоненциального сглаживания. При этом подходе в течение n итераций применяется простое экспоненциальное сглаживание на первом этапе к исходному ряду значений, а на последующих к ряду, полученному на предыдущей

итерации. Формула вычисления значений ряда очередной итерации последовательного экспоненциального сглаживания:

$$\hat{y}_t^n = \begin{cases} \boldsymbol{a} * y_t + (\mathbf{1} - \boldsymbol{a}) * \hat{y}_{t-1}^n, & n = 1 \\ \boldsymbol{a} * \hat{y}_t^{n-1} + (\mathbf{1} - \boldsymbol{a}) * \hat{y}_{t-1}^n, & n > 1 \end{cases}$$

Были рассмотрено последовательное применение простого экспоненциального сглаживания применение из 10 итераций, так как дальнейшее применение сглаживание не давало видимого результата. Константа a была выбрана равной 0.25, так как при этом значении был достигнут лучший результат при применении простого экспоненциального сглаживания. Далее представлены график сравнения исходного ряда численности популяции симуляции и ряда полученного в ходе применения последовательного простого экспоненциального сглаживания в течение 10 итераций с a=0.25 и графики промежуточных значений численностей популяции симуляции рядов на каждой итерации алгоритма сглаживания.

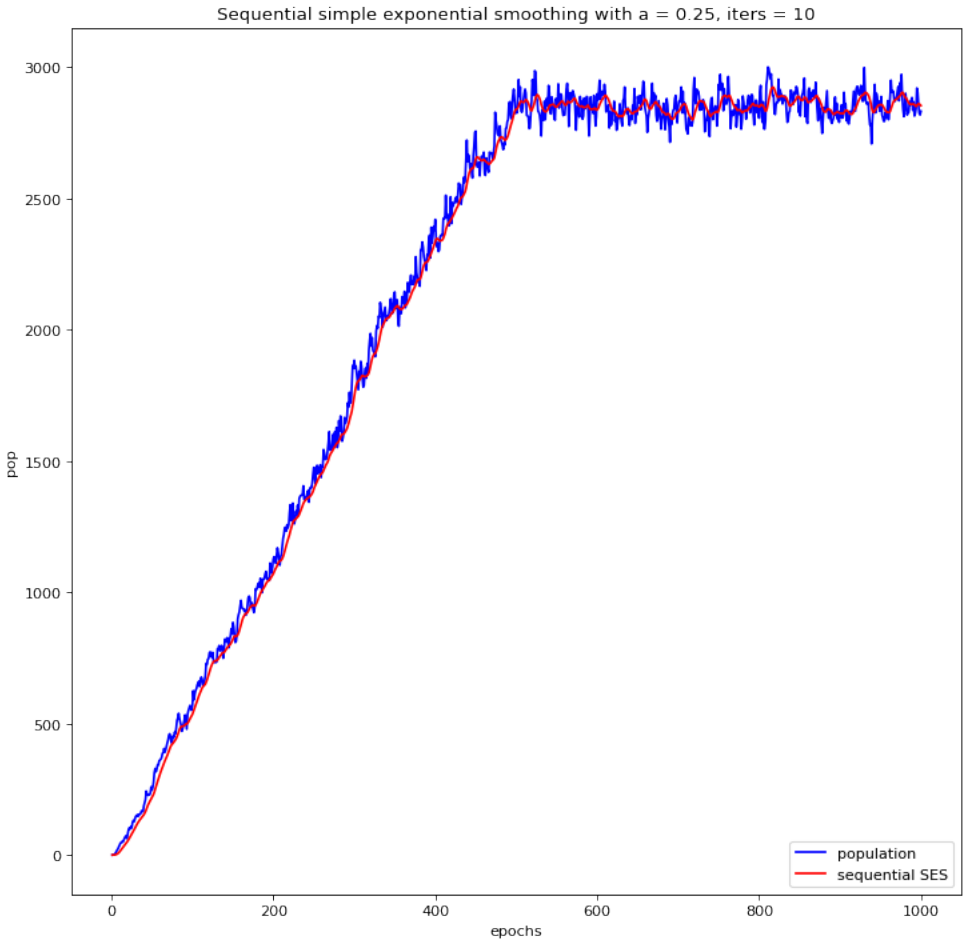
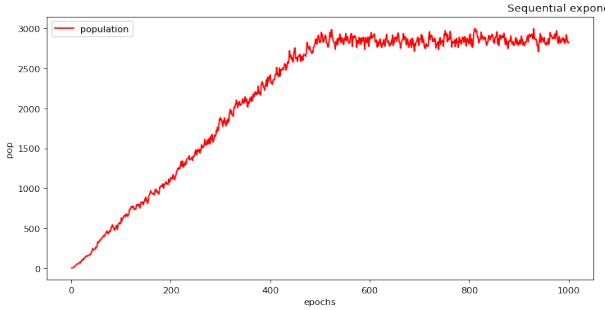
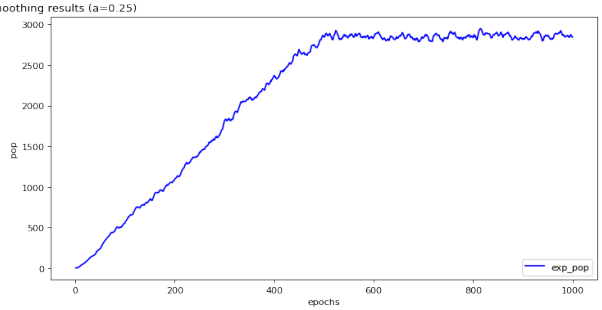
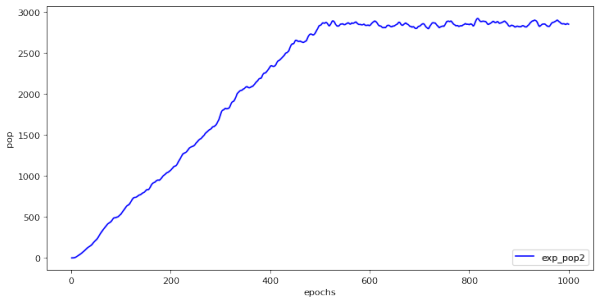


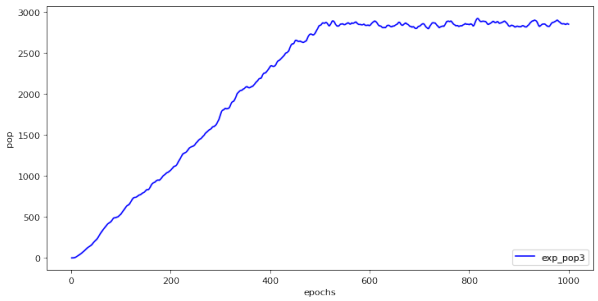
Рисунок 10 – Результат последовательного применения алгоритма простого экспоненциального сглаживания с коэффициентом a = 0.25 в течениие 10 итераций к данным симуляций.

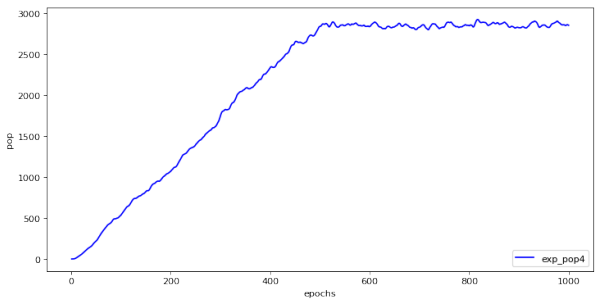
Sequential exponential smoothing results (a=0.25)

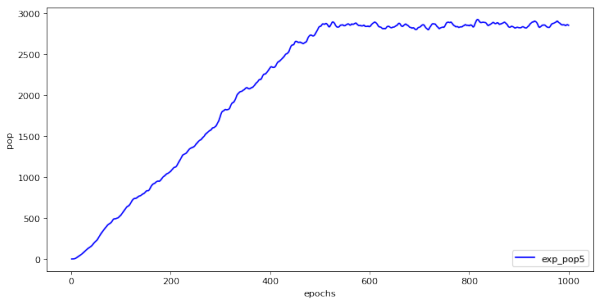


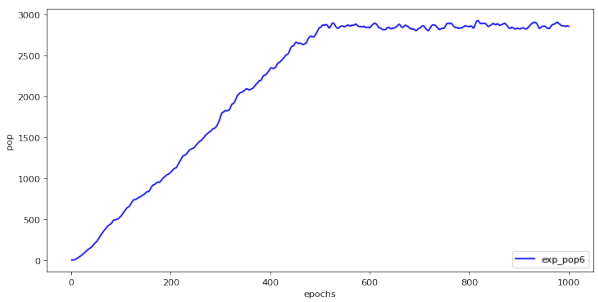


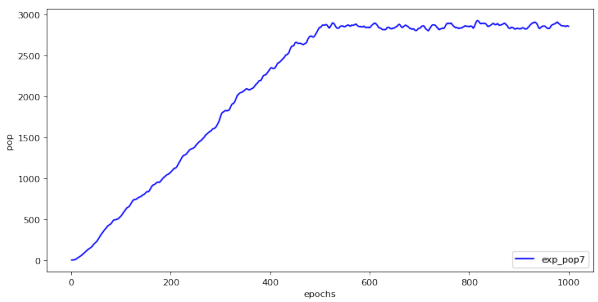


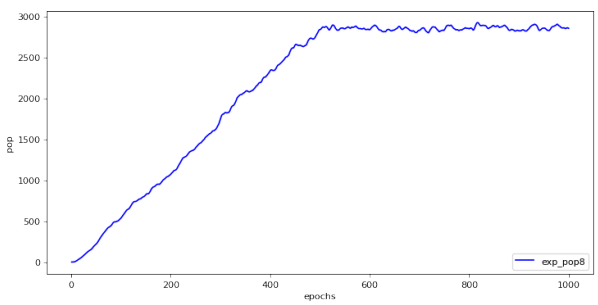


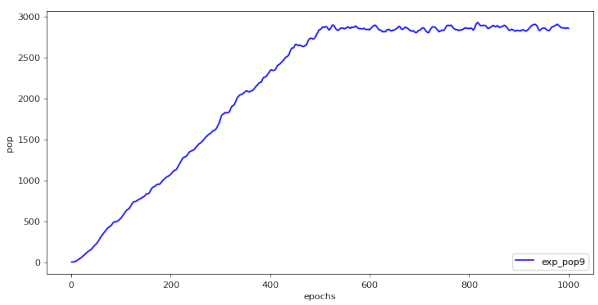












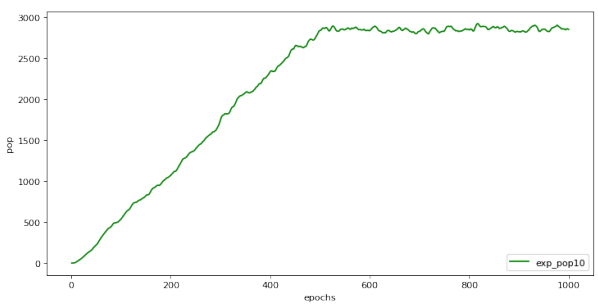


Рисунок 11 – Поэтапный результат последовательного применения алгоритма простого экспоненциального сглаживания с коэффициентом a = 0.25 в течениие 10 итераций к данным симуляций.

Визуальный анализ позволяет определить этот метод как наиболее подходящий среди всех рассмотренных. Кроме того, данный метод так же как и простое экспоненциальное сглаживание легко реализуем в алгоритме поиска момента выхода на плато при работе с данными режиме реального времени. Несмотря на незначительные изменения в графиках на последних итерациях (6 – 10) применения методе было принято решение о применении в алгоритме поиска выхода на плато 10 итераций, так как алгоритм сглаживания имеет линейную сложность и незначительно влияет на производительность симуляции, при этом на более зашумленных чем рассмотренный выше пример графиках наличие дополнительных итераций может повысить точность определения общего тренда симуляции. Исходя из вышеперечисленных преимуществ применения последовательного простого экспоненциального сглаживания, именно этот алгоритм использовался в дальнейшей работе с результатами симуляций в рамках данной курсовой работы.

3.3 Алгоритм поиска точки остановы на плато

3.3.1 Общее описание алгоритма поиска точки остановы плато

Во время работы исследовательской группы с результатами симуляций было выявлено, что большая часть графиков численности популяции симуляции содержит 3 основных участка: участок роста/ сокращения численности популяции, плато и участок перехода между предыдущими двумя участками – колено.

Такое строение графиков симуляций позволяет предложить для поиска плато алгоритм, основанный на сравнении дисперсий временных отрезков с предварительным применением последовательного простого экспоненциального сглаживания. Дисперсия на отрезке имеет самое большое значение на этапе роста/падения графика, при этом на плато она минимальна. Соответственно 2 соседних отрезка с максимальным отношением дисперсий будут находиться на этапе выхода графика на плато. Стоит заметить, что при разбиении графика на небольшие временные промежутки может понадобиться сравнение не соседних отрезков, а через один друг от друга. Пока данный порог не будет пройден симуляция не будет считаться достигшей плато.

В описанном выше алгоритме есть 3 параметра: коэффициент сглаживания, который был подобран на предыдущем шаге исследования, порог выхода на плато и длина отрезков рассматриваемых в алгоритме.

Код реализации данного алгоритма на языке R приведен в репозитории курсового проекта. (Приложение В)

3.3.3 Подбор длин временных отрезков в алгоритме

Для подбора переменных порога и длины окна в алгоритме были рассмотрены тяжелые, длительные симуляции идентичные тем, которые рассматривает наша научная группа. Результаты симуляций приведены на google drive (Приложение A) и визуализированы в блокноте (Приложение D)

Для подбора длин окон симуляции были разбиты на отрезки разной длительности: 100, 300, 500, 1000, 1500, 2000, 5000 эпох. Затем на этих отрезках были посчитаны дисперсии.

Графики дисперсий для каждой из симуляций. Пример для одной из симуляций

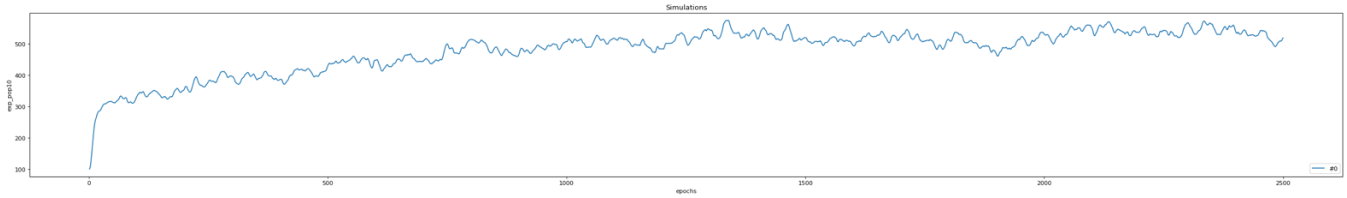


Рисунок 12 – Сглаженный график симуляции пример 3.

Дисперсии временных отрезков различной длительности.

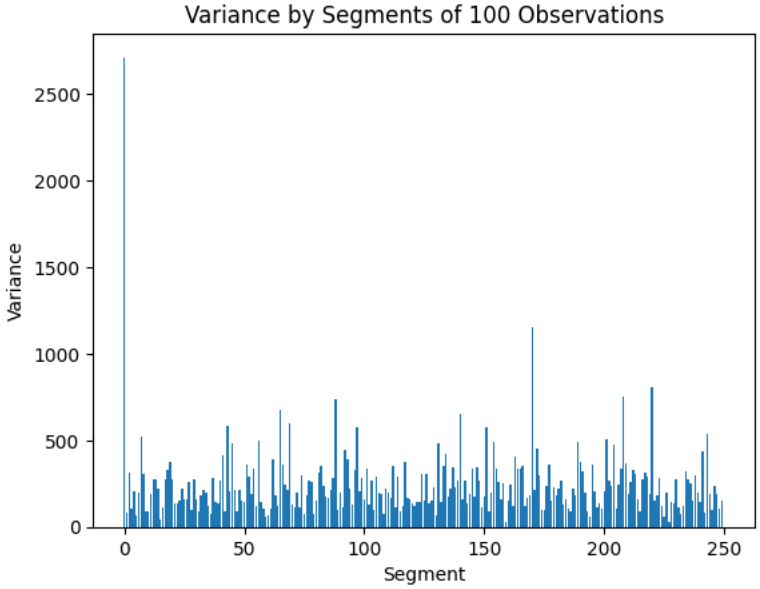


Рисунок 13 – Дисперсии временных отрезков длины 100 для симуляции пример 3.

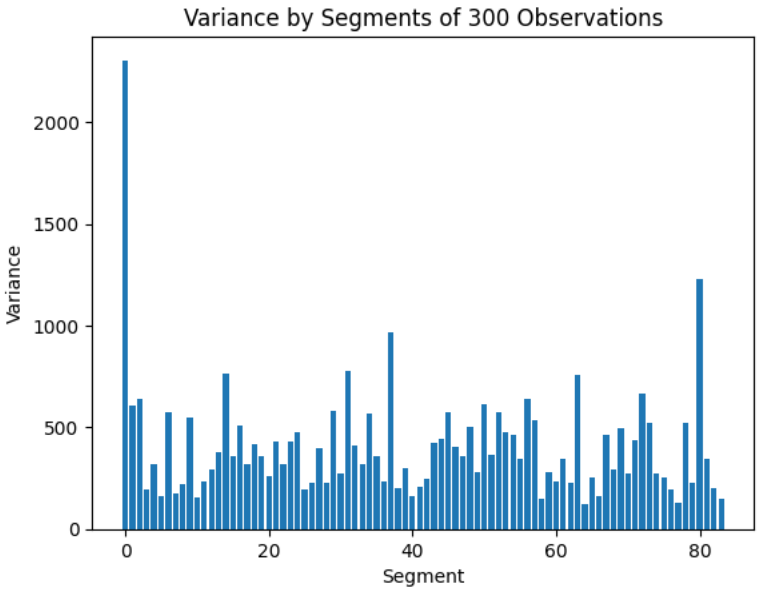


Рисунок 14 – Дисперсии временных отрезков длины 300 для симуляции пример 3.

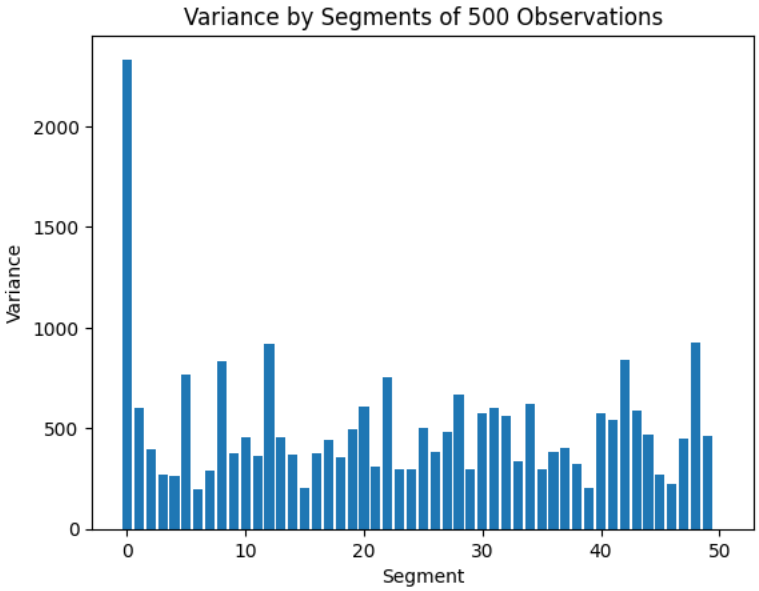


Рисунок 15 – Дисперсии временных отрезков длины 500 для симуляции пример 3.

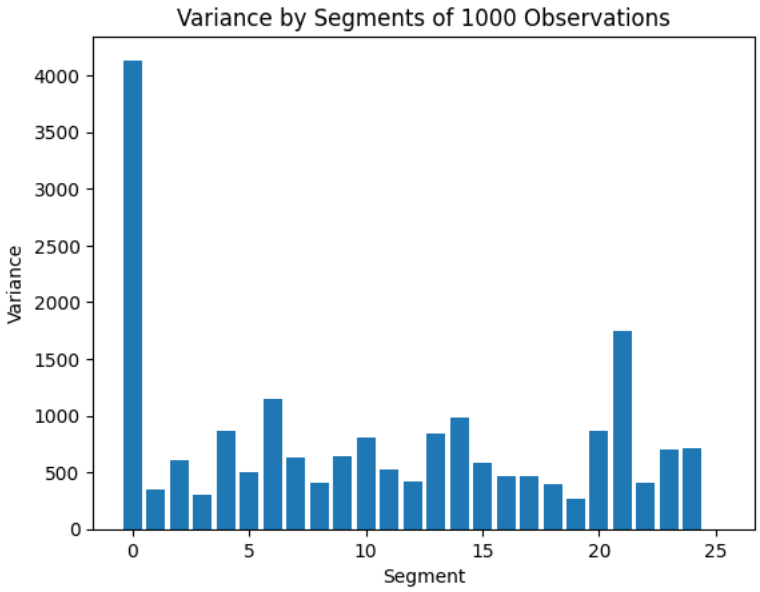


Рисунок 16 – Дисперсии временных отрезков длины 1000 для симуляции пример 3.

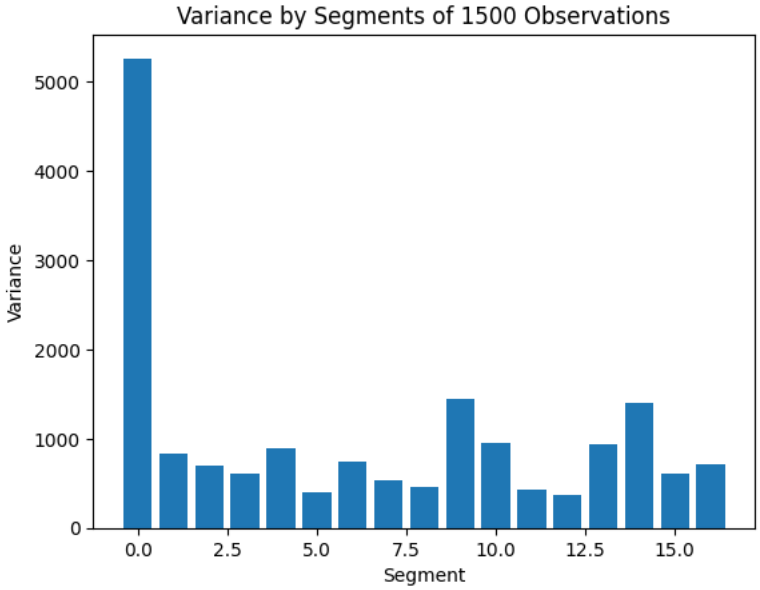


Рисунок 17 – Дисперсии временных отрезков длины 1500 для симуляции пример 3.

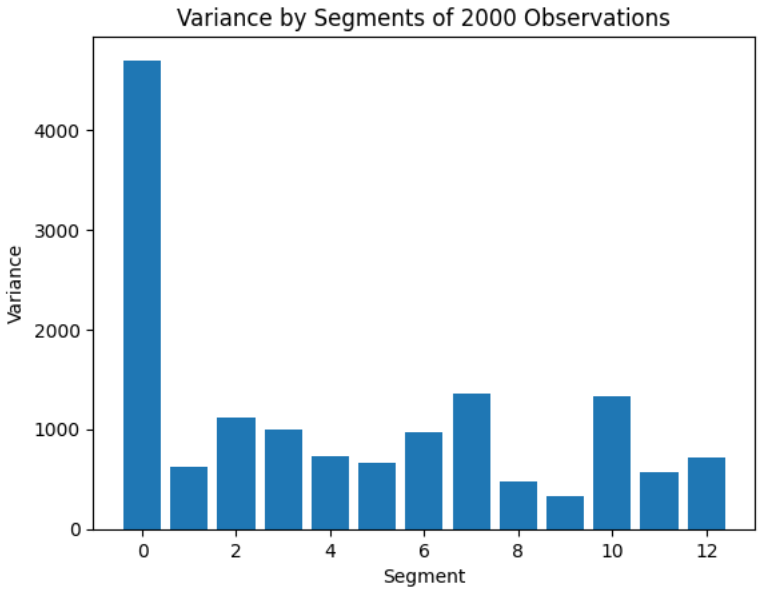


Рисунок 18 – Дисперсии временных отрезков длины 2000 для симуляции пример 3.

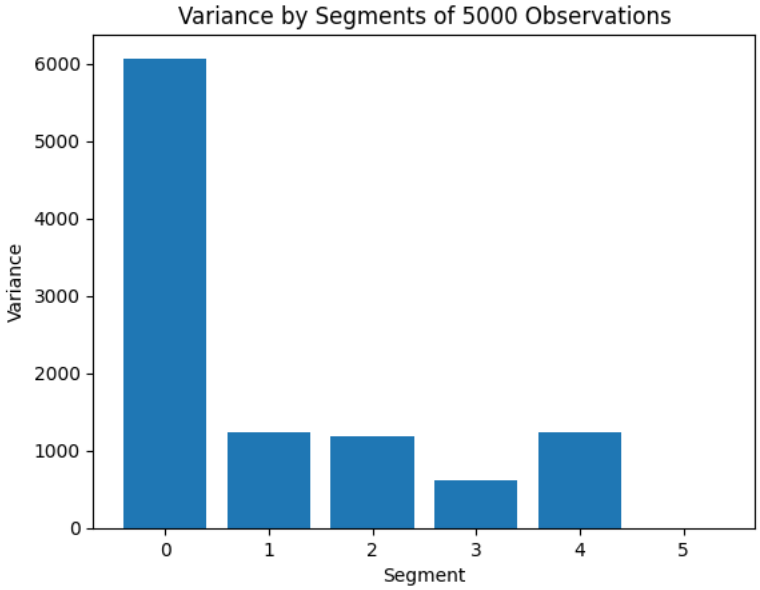


Рисунок 19 – Дисперсии временных отрезков длины 5000 для симуляции пример 3.

Визуально спад дисперсии после этапа роста отличим при любом размере окна из рассматриваемых, однако при маленьком размере окна график может быть остановлен на скачках на этапе роста или на этапе колена. Однако слишком большие временные отрезки требуют больших затрат на генерацию симуляций в тот момент, когда график может быть уже остановлен на плато.

Исходя из вышеописанных особенностей симуляций было решено рассматривать в каждой симуляции сразу два временных отрезка независимо и останавливаться, если точка остановы была достигнута хотя бы на одном из них. Во избежание остановки на скачках графика на этапе роста и на этапе колена минимальным размером для этих окон стали 500 и 1000 эпох соответственно. Кроме того, для регулирования размера окон в зависимости от длительности симуляции в эпохах, размеры временных отрезков задаются следующим образом:

$$size1 = \max\left(count_{epochs}\ div\ 5;\ 1000\right),$$

$$size2 = \max\left(count_{epochs}\ div\ 10;\ 500\right)$$

При изменении типа симуляций также остается возможность менять размер и правила рассматриваемых окон в коде алгоритма поиск плато.

3.3.2 Подбор порога выхода на плато для алгоритма

После выбора и фиксации длин временных отрезков возникает необходимость подбора порога выхода на плато. Более точно порог выхода на плато можно подбирать в зависимости от типа симуляций, с которыми проводится исследование, однако в этой курсовой работе приведена попытка подбора общего значения порога для симуляций с различными параметрами.

Для этого на всех сгенерированных на момент подбора симуляциях (без остановки на плато и без остановки по причине преодоления предела реального времени был вычислено максимально отношение дисперсий для размеров окон, заданных ранее описанным образом. Этот алгоритм реализован в блокноте подбора порога (Приложение D).

При сравнении данных популяции получилась статистика значений:

Таблица 6 – Распределение максимумов отношений дисперсий соседних участков на графике численности популяций для выбранных размеров окон на тестовых симуляциях.

Количество симуляций	470
mean	460.89
std	1653.76
min	1.03
25%	2.62
50%	106.02
75%	360.42
max	22294.09

Также была посчитана по значениям вторых максимумов, которые были встречены до нахождения первого, т. е. тем значениям, на которых симуляция может ошибочно остановиться, приняв их за выход на максимум:

Таблица 7 – Распределение вторых максимумов отношений дисперсий соседних участков на графике численности популяций для выбранных размеров окон на тестовых симуляциях.

Количество симуляций	111
mean	1.89
std	2.53
min	1.01
25%	1.09
50%	1.26
75%	1.73
max	18.81

В результате проведенного исследования в алгоритме поиска точки остановы на плато в режиме реального времени на симуляции по умолчанию применен порог доверия должен быть больше 18.81: 1 и точно меньше 106:1. Для отношения дисперсий отрезков выбран порог 20 к 1. Данное значение порога не гарантирует 100% вероятность нахождения выхода на плато, однако исходя из располагаемых данных минимизирует риск неправильного нахождения места выхода на плато и при этом максимизирует вероятность обнаружения выхода на плато.

3.3.4 Результаты применения алгоритма на симуляциях

Для демонстрации результатов работы алгоритма были проведены контрольные симуляции. Подробное описание параметров симуляций и код проверки результатов работы алгоритма представлен в блокноте (Приложение Е)

Для каждого набора параметров было проведено 2 симуляции в одной из них применялся алгоритм поиска точки остановы на плато, в другой алгоритм останавливался после достижения максимального количества эпох. Примеры парных симуляций:

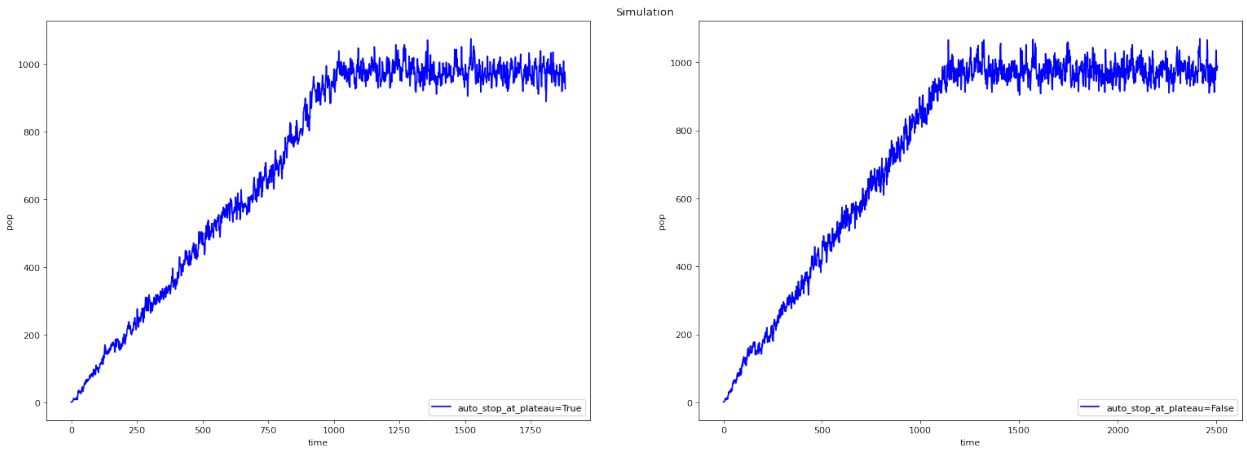


Рисунок 20 –Пример парных симуляций 1.

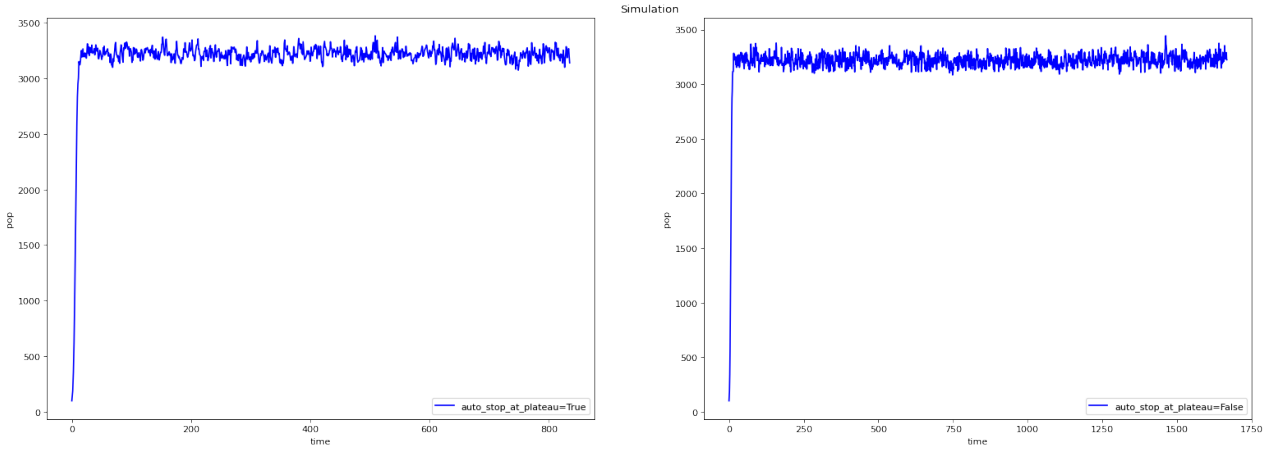


Рисунок 21 –Пример парных симуляций 2.

Данные симуляций, у которых остановка на плато произошла непосредственно перед завершением симуляции по достижению предела выполнения в эпохах, дополнительно проверялись вручную. Также дополнительную проверку проходили бы симуляции, для которых средние значения последних 100 наблюдений обоих вариантов запуска отличались более чем на 0.5% (Таких симуляций не было обнаружено).

Таблица 8 – Статистика отношений средних последних 100 наблюдений парных симуляций.

Количество пар симуляций	96
mean	1.00025
std	0.0053
min	0.987
25%	0.996
50%	0.999
75%	1.003
max	1.0147

В результате тестовой выборке не было обнаружено симуляций, на которых алгоритм работал бы неверно.

Однако существуют графики популяции, на которых алгоритм будет работать некорректно.

Пример график симуляции со следующими параметрами:

Таблица 9 – Параметры симуляции Рис21.

Параметр симуляции	**Значение**
sd_b	1
sd_d	0.8
b	1
d	0
dd	0.8
death_r	5
area_length_x	100
initial_pop	1300

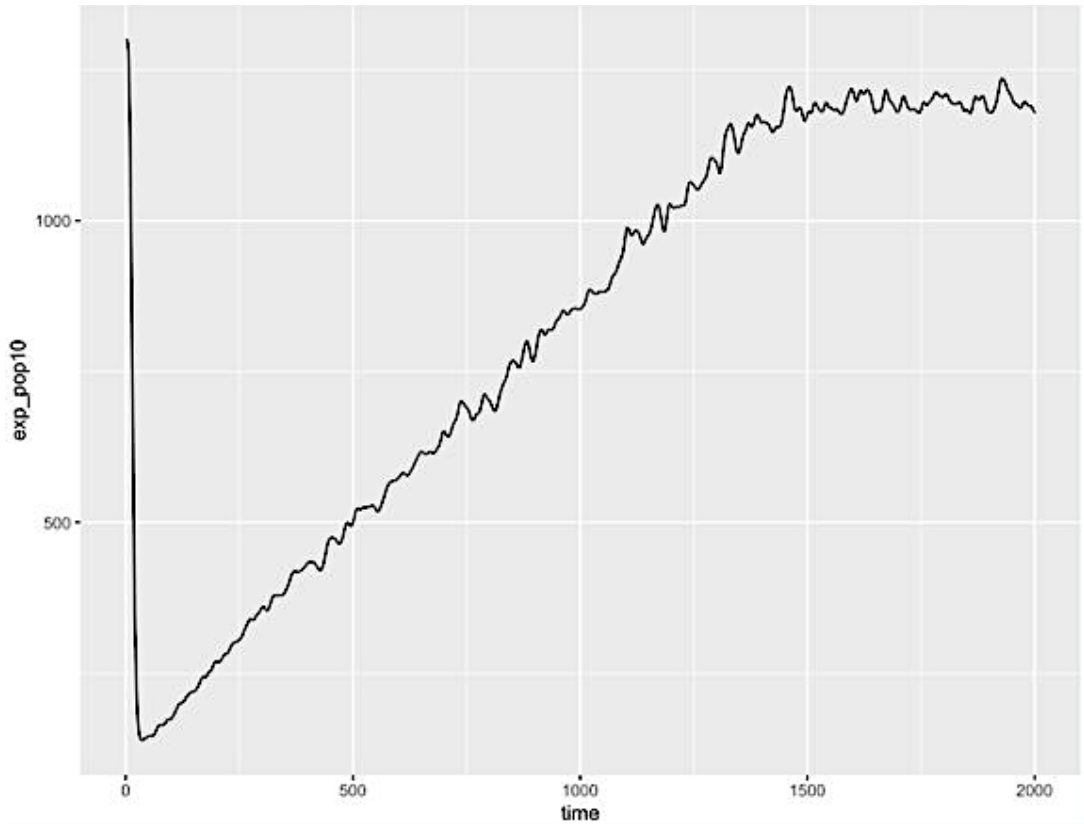


Рисунок 22 – График численности популяции пример 4.

Некорректная работа алгоритма на данном примере обусловлена резким падением численности популяции с последующим ростом (изменением тренда). Причиной такого падения становится «перенаселение» на начальном этапе симуляции при высокой конкуренции (dd) между особями. Для избежания подобных ситуаций и обеспечения корректной работы алгоритма необходимо задавать умеренное число индивидов, существующих в начальный момент времени.

4. РАБОТА С МОДЕЛЯМИ

4.1 Генерация датасета

После того, как в рамках данной курсовой работе был описан и реализован алгоритм остановки симуляции на плато численности популяции, появилась возможность проведения большого количества симуляций и генерации датасета на их основе.

Так как генерация данных времязатратна и требует определенных вычислительных ресурсов в рамках данной курсовой было принято решение рассматривать только симуляции в одномерном пространстве размера 100. При генерации был применен описанный выше алгоритм остановки симуляции на плато с подобранными в предыдущем разделе курсовой работы параметрами размеров окон и порога выхода на плато. Также для ускорения процесса генерации датасета было выставлено максимальное количество эпох – 4000 и предел реального времени исполнения симуляции - 1 минута.

Так как непосредственно данные симуляций занимают большое количество памяти на google drive приведена только таблица с параметрами симуляций и числовым значением численности популяции на плато, которое вычислялось как среднее арифметическое последних 100 значений численности популяции с предварительно примененным к ней в течение 10 итераций последовательным экспоненциальным сглаживанием. В приведенном на диске файлы записаны результаты всех проведенных во время генерации датасета симуляций, при этом для обучения и рассмотрения сетки параметров использовались только результаты, в которых симуляции не были остановлены по достижению предела реального времени или ограничения в количество эпох и были проведены в одномерном пространстве длинной 100. Скрипты запуска симуляций для генерации датасета и формирования .csv файла на основе полученных данных представлены в репозитории проекта (Приложение G).

Всего было проведено 290525 симуляций из них для обучения было отобрано 221711.

В полученном датасете параметры имеют следующие распределения.

Таблица 10 – Распределение значений параметров симуляций в датасете.

	mean	std	min	25%	50%	75%	max
b	0.674	0.228	0.1	0.55	0.75	0.85	0.95
d	0.362	0.221	0.1	0.2	0.3	0.5	0.9
death_r	5.361	2.961	0.1	3	5	8	10
dd	0.549	0.287	0.1	0.3	0.55	0.8	1
sd_b	0.725	0.167	0.5	0.6	0.75	0.9	1
sd_d	0.725	0.167	0.5	0.6	0.75	0.9	1
initial_pop	49.062	40.651	1	1	50	100	100

В результате симуляций получилось следующее распределение численностей популяций.

Таблица 11 – Распределение значений численности популяций на плато симуляций в датасете

mean	134.51
std	757.02
min	0.00
25%	0.00
50%	0.00
75%	0.00
max	34,227.56

Видно, что большом количестве симуляций популяции вымирали, это объяснимо, так как в датасете рассмотрено большое количество при d > b.

4.2 Обучение моделей предсказания популяций

4.2.1 Общее описание обучения

После генерации датасета появилась возможность обучения моделей, предсказывающих численность популяции на плато по заданным параметрам с использованием методов машинного и глубинного обучения.

Для обучения данные были разделены случайным образом на обучающую (70%), тестовую (18%) и валидационную (12%) выборки. Обучающей выборка использовалась для обучения моделей, тестовая для отбора лучшей модели и подбора гиперпараметров, а валидационная для оценки лучшей модели.

Результаты вычислений представлены в блокноте (Приложение F).

В качестве функции потерь была выбрана MSE.

4.2.2 Линейная регрессия

В первую очередь к данным была применена модель линейной регрессии. Так как параметры имеют разные порядки, была предварительно выполнена нормализация как всех параметров, так и целевой переменной. В ходе нормализации исходные данные (x) преобразуются следующим образом: $z = \frac{x-u}{s}$, где u – среднее значение параметра на обучающей выборке, а s – стандартное отклонение параметра на обучающей выборке. При подсчете MSE после обучения модели к целевой переменной была применена обратная нормализации операция для сравнимости результатов с другими экспериментами.

В результате обучения MSE на тестовой выборке составила около 520263, что является большим значением. Это также говорит о том, что численность популяции на плато не является линейно зависимой от параметров симуляции, поэтому для решения задачи необходимо применять модели, способные находить более сложные зависимости.

4.2.3 Решающие деревья

Одной из моделей, способных находить нелинейные зависимости целевой переменной от параметров являются решающие деревья. Модель можно представить в виде бинарного дерева, где в листьях хранятся множества предсказаний целевой переменной, а в узлах предикаты, разбивающие подмножество пространства признаков узла на 2 подмножества (дочерних узла) по порогу по одному из признаков. Построение такого дерева регуляризируется двумя гиперпараметрами: глубиной дерева и минимальным количеством целевых переменных обучающей выборки в одном листе. Однако данная модель легко переобучается, из-за чего имеет смысл применять ее только в ансамблях.

Для обучения была выбрана модель случайного леса, которая является ансамблем на основе алгоритма бэггинга. Для случайного леса с учетом баланса длительности обучения и качества результата было подобрано число решающих деревьев равное 300.

Для алгоритма была проверена сетка гиперпараметров, задающих минимальное количество элементов обучающей выборки в листе и максимальную глубину дерева. В результате лучший результат на тестовой выборке MSE = 42105 продемонстрировал случайный лес с максимальной глубиной дерева 3 и минимальным количеством значений в листе 3.

4.2.4 Нейронные сети

В третьем эксперименте для решения поставленной задачи была использована нейронная сеть, состоящая из трех полносвязных слоев с функцией активации ReLU. Для нее был подобран learning rate = 0.01.

После обучения в течение 10 эпох MSE модели на тестовой выборки составил 34621.

Данная модель показала лучший результат из всех ранее обученных. Качество сохранилось и на валидационной выборке (MSE = 34702).

Данное качество модели является приемлемым для первичного эксперимента, проводимого в рамках данной курсовой. При дальнейших исследованиях стоит провести больше экспериментов с различными архитектурами модели, увеличить размеры датасета и расширить сетку параметров. Также следует при наличии достаточной выборки экспериментировать с обучением моделей только на симуляциях, где d <= b, и, как следствие, меньше шанс вымирания популяции.

5. ЗАКЛЮЧЕНИЕ

В ходе исследовательской была изучены модель Дикманна-Лоу и реализация стохастических симуляций стационарных сообществ, основанная на этой модели.

На первом этапе были изучены результаты симуляций и подобран оптимальный алгоритм сглаживания. На следующем этапе был предложен и реализован алгоритм остановы на плато в режиме реального времени. Для описанного алгоритма были подобраны оптимальные параметры: длина временного отрезка и порог. Работоспособность алгоритма и корректность алгоритма были проверены на тестовом наборе симуляций.

Внедрение алгоритма остановы на плато обеспечило возможность генерации датасета симуляций, что было выполнено на следующем этапе исследования. На полученном датасете были проведены первичные эксперименты, связанные с обучением моделей машинного и глубинного обучения решать задачу регрессии по определению численности популяции на плато по ее параметрам.

В результате исследования была расширена база результатов симуляций для дальнейших исследований и оптимизированы методы ее дальнейшего масштабирования, а также показана потенциальная эффективность подхода к решению задачи поиска численности популяции на плато методами глубинного обучения.

СПИСОК ИСТОЧНИКОВ

1) Richard Law David J. Murrella, Ulf Dieckmannb. On moment closures for population dynamics in continuous space. Journal of Theoretical Biology 229, pages 421–432, 2004.
2) Poisson point process simulation of spatial population dynamics [Электронный ресурс] / GitHub. Режим доступа: https://github.com/YegorGalkin/RcppSim свободный (дата обращения 4.04.2023).
3) Dieckmann U., Law R. Moment approximations of individual-based models // The Geometry of Ecological Interactions: Simplifying Spatial Complexity / Ed. by U. Dieckmann, R. Law, J. Metz. Cambridge University Press, 2000.
4) Dieckmann U., Law R. Relaxation projections and the method of moments // The Geometry of Ecological Interactions: Simplifying Spatial Complexity / Ed. by U. Dieckmann, R. Law, J. Metz. Cambridge University Press, 2000.
5) Murrell D. J., Dieckmann U. On moment closures for population dynamics in continuous space // J. Theor. Biology. 229. 2004.
6) U. Dieckmann, R. Law. Relaxation Projections and the Method of Moments // Cambridge University Press, 2000
7) Николаев М. В., Никитин А. А. Исследование интегрального уравнения равновесия с ядрами-куртозианами в пространствах различных размерностей // Вестник Московского университета. Сер 15: Вычисл. матем. и киберн. 3, 2018.
8) Николаев М. В., Дикман У., Никитин А. А. Применение специальных функциональных пространств к исследованию нелинейных интегральных уравнений, возникающих в равновесной пространственной логистической динамике. // Доклады Академии Наук (в печати), 2021.
9) Аббасов М.Э. Методы оптимизации: Учебное пособие. — СПб.: Издательство “ВВМ”, 2014.
10) Куркин М. Л. Оптимизация параметров численного метода для модели экологических сообществ // Выпускная квалификационная работа, 2021. Режим доступа: закрытый
11) Куркин М. Л. Оптимизация параметров численного метода для модели экологических сообществ // Выпускная квалификационная работа, 2021. Режим доступа: закрытый
12) Михайлова К. Д. Исследование стационарной модели биологических сообществ// Курсовая работа, 2022. Режим доступа: закрытый

ПРИЛОЖЕНИЕ А

Результаты симуляций и датасет

https://drive.google.com/drive/u/1/folders/17URsRm9K5kYYqHnyPldIftXDf_RzqF-4

ПРИЛОЖЕНИЕ В

Реализация алгоритма остановки на плато в режиме реального времени.

https://github.com/sokanaid/CourseProject2022BiologicalMath/blob/main/run_simulation.R

ПРИЛОЖЕНИЕ С

Блокнот подбора метода сглаживания.

https://github.com/sokanaid/CourseProject2022BiologicalMath/blob/main/data_smoothins.ipynb

ПРИЛОЖЕНИЕ D

Блокнот подбора параметров для алгоритма поиска плато.

https://github.com/sokanaid/CourseProject2022BiologicalMath/blob/main/find_plateau_threshold.ipynb

ПРИЛОЖЕНИЕ Е

Блокнот проверки результатов работы алгоритма поиска плато.

https://github.com/sokanaid/CourseProject2022BiologicalMath/blob/main/plateau_search_results.ipynb

ПРИЛОЖЕНИЕ F

Блокнот применения методов машинного и глубинного обучения для предсказания численности популяции на плато.

https://github.com/sokanaid/CourseProject2022BiologicalMath/blob/main/model_learning.ipynb

ПРИЛОЖЕНИЕ G

Общий репозиторий со скриптами запуска симуляций.

https://github.com/sokanaid/CourseProject2022BiologicalMath