

С. И. Попков, аспирант, e-mail: rslw25@gmail.com, ГБОУ ВПО Московский государственный психолого-педагогический университет, Москва, Россия

Метод внешней оптимизации для идентификации марковских процессов

Разработан численный метод для оптимизации идентификации марковских процессов. Метод применяется как вспомогательный в тех случаях, когда необходимо обеспечить высокую эффективность основного метода за ограниченное или реальное время. В качестве примера такого основного метода в данной статье выбран метод перебора значимых параметров. В основу программно-алгоритмической реализации метода внешней оптимизации положен шаблон проектирования "декоратор", позволяющий модифицировать входные и выходные данные преобразуемой функции в целях получения некоторых преобразованных данных, в данном случае — улучшенного значения критерия за ограниченное время. Предложенный подход позволил увеличить эффективность метода перебора значимых параметров более чем на 25 % (по сравнению с эмпирическим подходом к подбору параметров).

Ключевые слова: численный метод, декоратор, марковский процесс, перебор значимых параметров

Введение

В последние годы возросла актуальность задач, связанных с управлением группами подвижных роботов, которые должны координировать свое поведение для достижения заданного результата. Концепции, базовые понятия, принципы построения, перспективы практического применения и другие аспекты создания подобных систем управления были рассмотрены в ряде работ [1—9]. Однако достаточно развитый математический аппарат, пригодный для практического управления поведением группы агентов, в настоящее время все еще формируется.

Поведение ряда технических, диагностических и других систем удобно описывать, используя специально разработанные параметрические математические модели в форме систем обыкновенных дифференциальных уравнений. Подобная форма представления применялась для ряда моделей, среди которых можно выделить модели усталостного разрушения, адаптивного тестирования и различные модели прикладных многоагентных систем, построенных на базе марковских случайных процессов с дискретными состояниями и непрерывным временем [10—13]. Указанные многоагентные системы, в частности, решают задачу управления группами подвижных объектов, которые должны координировать свое поведение в пространстве и кооперироваться для достижения заданного результата. Область применения таких систем включает управление полетом и

выполнением миссии группы беспилотных летательных аппаратов, групповое управление роботами, а также других мобильных систем.

Важной особенностью практического использования марковских моделей применительно к указанным задачам является тот факт, что их параметры идентифицируются по результатам наблюдений, представленных наборами значений исследуемых функций в контрольных точках.

Марковские модели для описания динамики переходов между состояниями представляются ориентированными графами, в которых вершины соответствуют состояниям, а дуги — переходам, для которых выполняются свойства пуассоновских потоков событий. В этих потоках число событий X , попадающих в любой временной интервал длины τ , начинающийся в момент t , распределено согласно закону Пуассона:

$$P_{t,\tau}(X = n) = \frac{a(t, \tau)^n}{n!} e^{-a(t, \tau)},$$

где $P_{t,\tau}(X = n)$ — вероятность появления n событий в течение рассматриваемого интервала; $a(t, \tau)$ — среднее число событий, попадающих в интервал длины τ , начинающийся в момент времени t . Рассматриваются только стационарные потоки, в которых $a(t, \tau) = \eta\tau$, а $\eta = \text{const}$ есть интенсивность стационарного потока. Упомянутые выше предположения о свойствах потоков событий обычны для прикладных задач [14], так как эти потоки (или потоки, близкие к ним по свойствам) часто встречаются на

практике благодаря предельным теоремам для потоков событий.

Полагается, что:

- для указанных процессов с $n + 1$ дискретными состояниями и непрерывным временем заданы начальные распределения вероятностей и наблюдаемые частоты пребывания в состояниях процессов $\{F_{id}\}_{i=0, \dots, n}$ в моменты времени $\{t_d\}_{d=0, \dots, D-1}$, где D — число моментов времени, в которые фиксировались частоты F_{id} ; $0 \leq t_d \leq T$, где T — конечный момент времени;
- интенсивности переходов между состояниями полностью или частично являются неизвестными (свободными) параметрами.

Динамика изменения вероятностей пребывания в состояниях процесса определяется системой обыкновенных дифференциальных уравнений Колмогорова в матричной форме:

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{M}(\boldsymbol{\lambda})\mathbf{p},$$

где $0 \leq t \leq T$; $\mathbf{p}(t) = (p_0(t), \dots, p_n(t))^T$ — вероятности пребывания в состояниях процесса; $\boldsymbol{\lambda} = (\lambda_0, \dots, \lambda_m)^T$ — упорядоченный набор интенсивностей переходов между состояниями, $n > m$; \mathbf{M} — матрица интенсивностей переходов между состояниями порядка $n + 1$. Для указанной системы уравнений ставится задача идентификации набора параметров $\boldsymbol{\lambda}$. Значения параметров из этого набора определяются путем сравнения наблюдаемых и прогнозируемых гистограмм, описывающих распределения частот пребывания в состояниях модели, а именно: вычисляются значения, обеспечивающие наилучшее соответствие наблюдаемых и ожидаемых частот попадания в определенное состояние системы в заданные моменты времени.

Для решения этих проблем был предложен численный метод *перебора значимых параметров* [15], предназначенный для идентификации набора параметров $\boldsymbol{\lambda}$, определяющих матрицу \mathbf{M} . Этот метод представлен далее в версии, совместимой с задачей моделирования прикладных многоагентных систем, упомянутых выше.

Несмотря на то что скорость оптимизации с применением описанного алгоритма, как правило, близка к приемлемой, существует целый ряд задач, для которых одним из требований является мгновенное принятие решений. Цель данной работы — разработать и представить математический метод, позволяющий адаптировать параметры основного метода оптимизации

для повышения эффективности его работы, а также продемонстрировать реализацию нового метода в виде алгоритма вычислений и в качестве действующего программного инструмента.

Метод перебора значимых параметров. Алгоритм вычислений

1. Используя имеющиеся результаты наблюдений, вычислить в качестве начальных оценок параметров $\lambda_i (i = 0, \dots, m)$, обозначаемых $\lambda_i^0 (i = 0, \dots, m)$, случайно взятые значения интенсивностей переходов между парами соответствующих состояний марковского процесса в единицу времени. Полученные оценки рассматривать как начальные приближения к идентифицируемому параметрам.

2. Положить $j = 1, k = 1, \beta_1 = 1 + \gamma_1$, где γ_1 — параметр алгоритма.

3. Оценить чувствительность критерия X^2 в малой окрестности каждой из текущих оценок идентифицируемых параметров $\lambda_i^j (i = 0, \dots, m)$ на j -й итерации алгоритма, используя для этого разностные аппроксимации абсолютных значений частных производных

$$\delta_i(\varepsilon) = |X^2(\lambda_0^j, \dots, \lambda_i^j + \varepsilon, \dots, \lambda_m^j) - X^2(\lambda_0^j, \dots, \lambda_i^j, \dots, \lambda_m^j)/\varepsilon|,$$

где $i = 0, \dots, m$; ε — параметр алгоритма.

4. Выбрать l текущих оценок идентифицируемых параметров $\lambda_{i_q}^j$, где $i_q \in \{0, \dots, m\}$; $q = 1, \dots, l$; l — параметр алгоритма, соответствующий оценкам, имеющим наибольшие значения оценок чувствительности $\delta_i(\varepsilon)$.

5. Для каждой из выбранных на шаге 4 текущих оценок идентифицируемых параметров $\lambda_{i_q}^j$ на j -й итерации алгоритма, где $i_q \in \{0, \dots, m\}$, $q = 1, \dots, l$, вычислить верхнюю $\lambda_{i_q}^{j+}$ и нижнюю $\lambda_{i_q}^{j-}$ границу сдвига оценки по формулам:

$$\lambda_{i_q}^{j+} = \beta_j \lambda_{i_q}^j, \lambda_{i_q}^{j-} = \beta_j^{-1} \lambda_{i_q}^j.$$

6. Выполнив полный перебор всех вариантов текущих оценок набора параметров $\boldsymbol{\lambda}^j = (\lambda_0^j, \dots, \lambda_m^j)^T$, в которых каждый из его компонентов $\lambda_{i_q}^j$, где $i_q \in \{0, \dots, m\}$, $q = 1, \dots, l$, принимает только три возможных значения из множества $\{\lambda_{i_q}^{j-1}, \lambda_{i_q}^j, \lambda_{i_q}^{j+1}\}$, а оставшиеся компоненты — значение своей текущей оценки на j -й итерации алгоритма, выбрать из указанных вариантов одну из оценок $\boldsymbol{\lambda}^{j*} = (\lambda_0^{j*}, \dots, \lambda_m^{j*})^T$, обеспечивающих максимальное значение критерия X^2 , равное $X_{j, \max}^2$.

7. Если для всех компонентов выбранного набора $\lambda^{j,*}$ выполняется равенство $\lambda_i^{j,*} = \lambda_i^{j-1}$ ($i = 0, \dots, m$), то положить $k = k + 1$, $\gamma_k = \gamma_{k-1}/2$.

8. Положить $j = j + 1$, $\beta_j = 1 + \gamma_k$, $\lambda^j = \lambda^{j-1,*}$.

9. Если $|X_{j-1, \min}^2 - X_*^2| > \varepsilon_t$, где X_*^2, ε_t — параметры алгоритма, то перейти к шагу 3, иначе завершить вычисления.

Комментарий к представленному алгоритму вычислений

Для решаемых на практике типовых задач общее время вычислений зависит от параметра l монотонно, достигая наименьшего значения при $l = 1$. Временная сложность одной итерации модифицированного алгоритма при $l = 1$ есть $O(m)$, что свидетельствует об асимптотически линейной зависимости времени вычислений, необходимых для выполнения одной итерации, от числа идентифицируемых параметров (рис. 1).

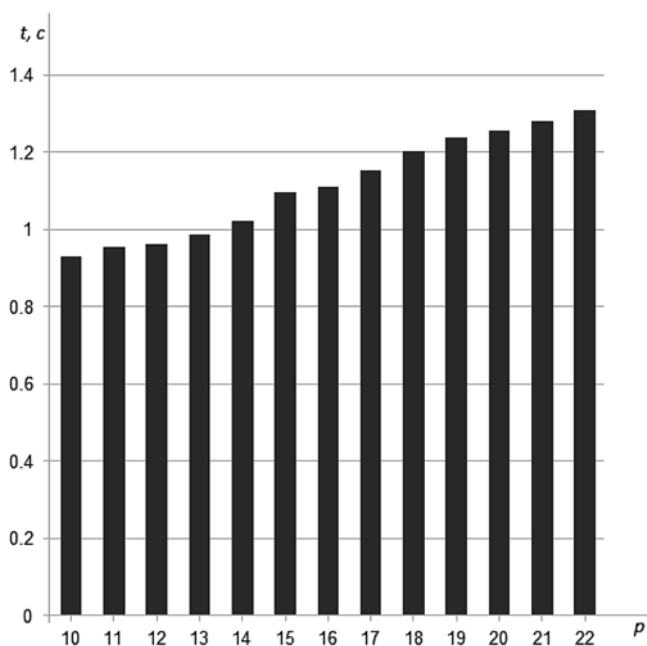


Рис. 1. Эмпирическая зависимость времени выполнения алгоритма t от числа параметров p

Введение терминологии для нового математического метода

Приведенный выше метод оптимизации (и соответствующий ему алгоритм) будем условно называть *основным*.

Параметры, существенно влияющие на качество и время работы основного алгоритма, но не относящиеся к числу оптимизируемых основным методом значимых параметров, будем называть *метапараметрами основного алгоритма*.

Эффективность основного алгоритма за одну итерацию можно рассматривать как отношение значения максимизируемого критерия ко времени, затраченному на работу алгоритма, при условии, что исходные значимые параметры остаются неизменными между итерациями (сохраняют свои изначальные значения в качестве входных параметров основного алгоритма). Тогда, адаптивно меняя метапараметры основного алгоритма оптимизации, оказывающие наибольшее влияние на рост его эффективности, можно повысить эффективность основного математического метода в целом.

Эмпирически было определено, что такими метапараметрами являются:

- M — число варьируемых значимых параметров;
- Δ — начальная поправка для значимых параметров.

Алгоритм-надстройку, осуществляющую модификацию метапараметров основного алгоритма в целях увеличения критерия эффективности, будем называть *внешним алгоритмом оптимизации*.

Для реализации соответствующего численного метода необходимо решить следующие задачи:

1. Каким образом будет осуществлено взаимодействие с основным методом?
2. Как алгоритмически будет представлен процесс адаптивного последовательного изменения метапараметров?

Взаимодействие между методами: шаблон "декоратор"

Алгоритмический шаблон проектирования "декоратор" описывает программную функцию высшего порядка, которая на основе получаемых значений аргументов, включая указатель на модифицируемую функцию, преобразует процесс работы последней, внося изменения в ее входные параметры и анализируя (с последующим внесением необходимых изменений) выходные данные.

В отношении описанного выше метода оптимизации можно рассматривать алгоритм метода

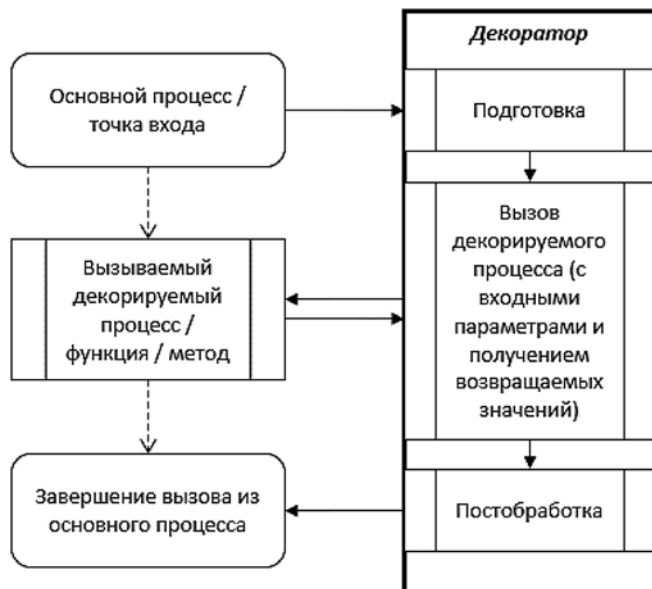


Рис. 2. Общий пример вызова декоратора из произвольного основного процесса. В качестве декорируемого процесса может выступать произвольная функция, метод или внешний процесс, которому временно передается управление из вызывающего процесса

перебора значимых параметров как модифицируемую функцию, для которой в качестве изменяемых входных данных задаются соответствующие значения изменяемых метапараметров.

Общая иллюстрация способа применения шаблона "декоратор" в ходе вызова декорируемой функции представлена на рис. 2.

Условно процесс можно разделить на три части: подготовка к вызову декорируемого процесса, непосредственно вызов и постобработка.

Подготовка заключается в осуществлении действий, предвещающих запуск вызываемого (также называемого декорируемым) процесса. Это может быть вывод сообщения для отладки, предварительное сохранение значимых данных перед осуществлением транзакции, представленной декорируемым процессом, подготовка промежуточных вычислений на основе входных аргументов и параметров, получаемых процессом локально или из глобального контекста, а также другие процедуры, осуществляющие инициализацию модифицированного окружения и контекста вызываемого процесса для достижения конкретной поставленной перед декоратором задачи.

Вызов декорируемого процесса может сопровождаться как изменением входных аргументов и возвращаемых значений, так и их сохранением в случае, если воздействие на

окружение процесса осуществлено каким-то другим образом (например, через вызов другого процесса во время подготовки декоратора).

Постобработка — это процесс модификации выходных данных либо применения их в промежуточных вычислениях для анализа и извлечения результатов, определяющих работу декоратора.

Примером автономной программной реализации такого шаблона можно считать приложение, вызывающее другое приложение, библиотечную функцию или системный вызов.

Описание процесса адаптивного изменения метапараметров

Надстройка алгоритма является самостоятельной функцией, выполненной в форме шаблона проектирования "декоратор" для объектно-ориентированного метода, представляющего собой реализацию алгоритма оптимизации для заданного марковского процесса либо группы процессов. Среди всех задаваемых аргументов функции последовательно изменяются параметры модели, отвечающие за число изменяемых параметров и значение начального сдвига параметров.

Представим "сетку" — двумерное пространство, в котором по одной оси с фиксированным шагом h_1 задается число одновременно изменяемых параметров от 1 до N , а по другой также с фиксированным шагом h_2 определяется начальный сдвиг искомых параметров (от h_2 до 1). Полученное пространство позволяет определить значение критерия эффективности метода в каждой ячейке "сетки" — в зависимости от конкретного значения модифицируемых параметров.

Описываемую сетку можно представить в виде, показанном на рис. 3.

Время, затрачиваемое на эмпирический поиск наиболее оптимального решения (например, посредством полного перебора), значительно возрастает с увеличением сложности модели, делая такой поиск непригодным на практике. В связи с этим наблюдением возникает необходимость в численном методе, позволяющем качественно улучшить эффективность основного метода, при этом адаптирующемся под предельно допустимое время, как выделяемое для самого процесса поиска оптимального решения, так и затрачиваемое на каждый шаг алгорит-

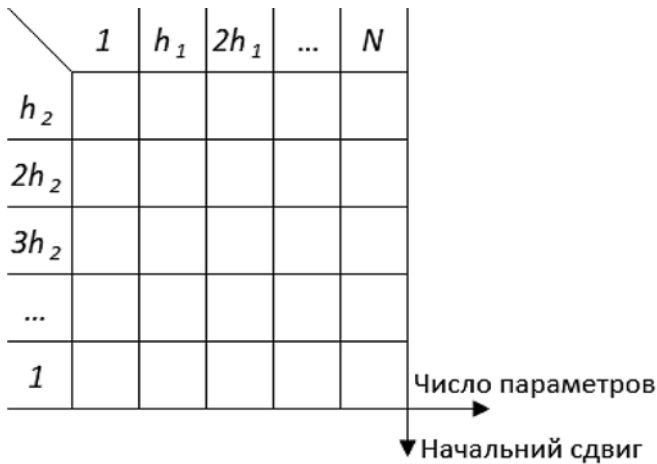


Рис. 3. Двумерное пространство параметров для проведения процедуры оптимизации

ма оптимизации, для которого осуществляется подбор оптимальных параметров запуска.

Такой метод был разработан, его алгоритм представлен ниже.

Метод внешней оптимизации. Алгоритм вычислений

1. Задается минимальное число изменяемых параметров ($N \geq 1$), максимальное число изменяемых параметров ($U \leq$ число параметров), шаг сетки по параметрам (S), шаг сетки (D), предельный срок вычислений (H), максимальное время (T).

2. Генерируется сетка параметров M, d (число изменяемых элементов, начальный шаг), $N \leq M \leq U$, $0 \leq d \leq 1$, шаг $M = S$, шаг $d = D$.

3. На сетке выбирается произвольная начальная позиция M_0, d_0 и инициализируется словарь пути Z и массив оптимального результата R , хранящий данные о координатах и значениях словаря пути Z .

4. В окрестностях M_0, d_0 выбирается множество точек $\{M_x, d_y\}$, такое, что каждая координата M_x принадлежит отрезку $[M_{0-1} \dots M_{0+1}]$, каждая координата d_y принадлежит отрезку $[d_{0-1} \dots d_{0+1}]$.

5. Инициализируется массив результатов измерений P для хранения вещественных значений с числом элементов, соответствующим числу точек во множестве $\{M_x, d_y\}$.

6. Для каждой точки, принадлежащей множеству $\{M_x, d_y\}$, выполняется следующая последовательность действий:

6.1. Если координата выбираемой точки принадлежит диапазону соответствующих зна-

чений, определенных на шаге 1, то перейти к шагу 6.4.

6.2. Если координата выбираемой точки имеет значение меньше нуля, то присвоить такой координате значение, равное максимальному значению из допустимого диапазона; повторить для каждой координаты.

6.3. Если координата выбираемой точки имеет значение больше максимально допустимого для соответствующего диапазона, то присвоить такой координате значение, равное минимальному значению из допустимого диапазона; повторить для каждой координаты.

6.4. Если точка уже была занесена в словарь пути Z в качестве ключа, то считать значение с соответствующим ключом из словаря и перейти к шагу 6.8.

6.5. Используя имеющийся основной алгоритм оптимизации с параметрами, соответствующими текущему положению сетки в заданной точке, выполнить процедуру оптимизации до критерия.

6.6. Время, потраченное на выполнение алгоритма в пункте 6.2, а также достигнутый уровень значения критерия в ходе выполнения алгоритма фиксируются в словаре пути Z , где в качестве ключа словаря используется искомая точка.

6.7. Потраченное время прибавить к общему времени H_0 , затраченному на выполнение алгоритма оптимизации во всех искомых точках.

6.8. Занести значение критерия для текущих координат в массив результатов измерений P для соответствующего значения координат заданной точки.

6.9. Если все точки из множества $\{M_x, d_y\}$ перебраны, то переход на шаг 7; иначе — переход на шаг 6.1.

7. Осуществляется поиск наибольшего значения критерия в массиве результатов измерений P и вычисляются соответствующие данному значению координаты M_{\max}, d_{\max} .

8. Для значения координат M_{\max} и d_{\max} , используемых в качестве ключей словаря пути Z , извлекаются соответствующие значения затраченного времени и уровня критерия; если затраченное время превышает порог максимального времени T , отводимого на один прогон алгоритма оптимизации с параметрами, то переход на шаг 11.

9. Внешний критерий определяется как функция f от затраченного времени T_0 на выполнение вложенного алгоритма оптимизации и значение полученного критерия C_0 : $f(C_0, T_0) = C_0/T_0$.

10. Если массив оптимального результата R пуст или значение найденного внешнего критерия в массиве R не превышает соответствующего значения критерия для словаря пути Z с ключом (M_{\max}, d_{\max}) , то новый набор значений массива R задается эквивалентным совокупности ключей словаря пути Z и соответствующему значению внешнего критерия для координат M_{\max} и d_{\max} .

11. Если $M_{\max} == M_0$ и $d_{\max} == d_0$, то переход на шаг 12, иначе переход на шаг 15.

12. Задаются случайные координаты сетки M_r и d_r .

13. Если словарь пути Z содержит в качестве ключа координаты сетки M_r и d_r , а размерность пространства ключей словаря не превышает значений соответствующих параметров, заданных на шаге 1, то перейти на шаг 12.

14. Присвоить $M_0 = M_r$, $d_0 = d_r$.

15. Если время H_0 , затраченное на выполнение алгоритма оптимизации на всем ансамбле измеренных параметров, превышает пороговое значение предельного срока вычислений H , то перейти на шаг 16, иначе перейти на шаг 4.

16. Если массив оптимального результата R пуст, то перейти на шаг 18.

17. Вернуть в качестве результата выполнения алгоритма массив оптимального результата R и перейти на шаг 19.

18. Вернуть внутренний код сообщения об ошибке "Начальные параметры заданы неверно".

19. Завершение работы алгоритма.

Комментарий к алгоритму

В терминах шаблона проектирования "декоратор" описанный выше алгоритм можно разбить на три части, подробно описанные выше: подготовка (шаги 1—6.4), вызов декорируемого процесса (шаг 6.5) и постобработка (шаги 6.6—19).

Подготовка перед запуском декорируемого процесса — основного метода оптимизации — заключается в инициализации сетки допустимых значений изменяемых параметров декорируемого процесса перед его вызовом на основе параметров для процесса-декоратора, описывающего "внешний" метод оптимизации. Инициализируются служебные структуры хранения данных для фиксации траектории движения внешнего метода оптимизации по сетке параметров в целях достижения оптимального результата работы вызываемого процесса на основе принципов, представленных ранее. За-

пускается цикл с постусловием, позволяющий осуществлять проход по пространству возможных значений параметров, которое ограничено узлами предварительно сгенерированной сетки. Устанавливается число одновременно изменяемых параметров и начальный сдвиг искомых параметров в соответствии с условиями, определенными для текущей итерации. Также выполняются действия, программная реализация которых варьируется в зависимости от среды выполнения (архитектуры вычислительной системы, операционной системы и т. п.), суть которых сводится к обеспечению работы таймера для проведения измерения временного интервала, который необходим для работы основного метода оптимизации при заданных параметрах.

После проведения этих действий осуществляется вызов декорируемого процесса, которым является основной метод оптимизации. Эта процедура необходима для получения значения возвращаемого критерия при заданных параметрах и возможности провести измерение временного интервала, в ходе которого осуществляется искомый вызов.

Постобработка заключается в фиксации значений, полученных в качестве результата вызова декорируемого процесса в целях последующего отбора наилучших результатов в ходе работы цикла, а также определения критериев останова алгоритма. Также в ходе постобработки фиксируются возможные ошибки при задании начальных параметров работы внешнего алгоритма оптимизации, который сообщает пользователю о недостижимости поставленной цели при текущих условиях выполнения.

Данный метод может быть применен не только для улучшения представленного метода оптимизации, но и для произвольного численного метода, быстроедействие и эффективность которого зависят от подчиняющихся модификации параметров. Метод может быть расширен для N -мерного пространства параметров за счет увеличения числа шагов, обеспечивающих поиск вокруг текущей точки.

Пример практического применения

Пример запуска и работы программной реализации метода на практическом примере показаны на рис. 4.

Задается предел времени (в секундах) для всех итераций поиска и для каждого шага. На

```

C:\Windows\system32\cmd.exe
Time Limit: 40.0 ; Time Limit per pt: 10.0
Start point: (0.1, 1) with Criteria = 10.836506758034245
Time to go: 41.285477876000016
Destination point: (0.2, 1) with Criteria = 13.68880788605269
The method quality has been improved by 26.321 %

```

Рис. 4. Пример программной реализации (основная кросс-платформенная версия без графического интерфейса)

основе этого, согласно описанному ранее алгоритму, проводится поиск оптимального решения. Значение оцениваемого критерия и затраченное время выполнения демонстрируются в соответствующем окне консоли. В частности, при переходе от произвольно выбранной начальной точки в пространстве "сетки" метопараметров (сдвиг = 0,1, число параметров = 1) к точке (сдвиг = 0,2, число параметров = 1) критерий эффективности увеличивается с 10,84 до 13,69 (на 26 %), при этом время поиска составило 41,3 с (при увеличении лимита допустимого времени качество может увеличиться). Метод дает гарантию, что основная процедура оптимизации при найденных метопараметрах для начальных условий, в которых проводилась оптимизация, не превысит 10 с (на практике полученное время одной итерации в среднем не превышает 0,06 с).

Конечная и начальная точка пути поиска наравне с показателем эффективности, выраженном в процентах, представлена выводом консольного окна и может быть применена при запуске программной реализации модели путем перенаправления стандартного канала вывода в файл или другое хранилище данных, совместимое с форматом данных, возвращаемых методом.

Применение разработанного метода в качестве надстройки над основным методом перебора значимых параметров увеличивает эффективность последнего на 26,32 %, что является значимым результатом с точки зрения увеличения общей эффективности решения ввиду того, что основной метод применяется не один раз, а на каждом шаге рассматриваемой прикладной задачи (в ходе работы вероятностной модели поведения прикладной многоагентной системы), как показано на рис. 5.

При необходимости может быть осуществлена сериализация промежуточных данных в формат XML, совместимый с большинством современных редакторов электронных таблиц (LibreOffice Calc, Microsoft Excel, WPS Spreadsheets и др.), что позволяет осуществлять необходимые расчеты и анализировать промежу-

точные результаты поиска оптимального пути по сетке.

Например, можно оценить среднее значение критерия в зависимости от роста изменяемого параметра алгоритма метода, например, начальный сдвиг (рис. 6) или относительное значение времени, затрачиваемого на поиск, при увеличении числа параметров (рис. 7).

Данные могут быть получены для произвольного алгоритма оптимизации.

На основе всего вышесказанного можно сделать вывод о том, что представленный ма-



Рис. 5. Пример действующей многоагентной системы, использующей разработанный метод. В процессе моделирования метод оптимизации был вызван 57 раз (при выборе направления движения в сторону цели)

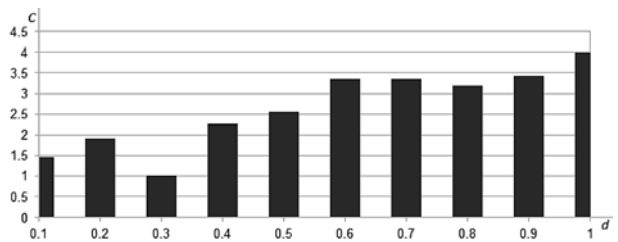


Рис. 6. Среднее значение критерия c в зависимости от роста изменяемого параметра алгоритма d

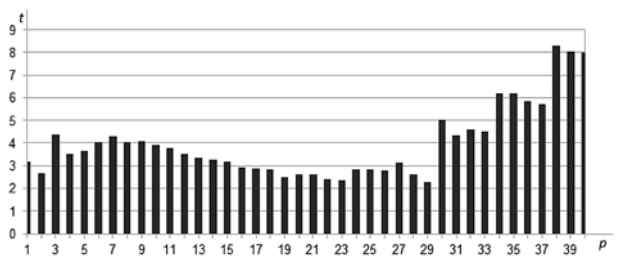


Рис. 7. Относительное значение времени t , затрачиваемого на поиск, при увеличении числа параметров p

тематический метод может повысить эффективность работы произвольного алгоритма оптимизации за счет адаптивного изменения его метапараметров.

Основные результаты и выводы

1. Исследован метод перебора значимых параметров в реализации, ориентированной на решение задачи моделирования прикладных многоагентных систем. Определены достоинства и недостатки метода, проиллюстрированы эмпирическими результатами.

2. Предложен и разработан новый математический метод, позволяющий адаптировать метапараметры основного метода оптимизации для повышения эффективности его работы.

3. Изучен и применен шаблон проектирования "декоратор", позволяющий использовать новый метод в качестве надстройки над произвольным методом оптимизации (в частности — над методом перебора значимых параметров, рассмотренным выше) для увеличения критерия эффективности при минимизации временных затрат на поиск оптимального критерия.

4. Создан и применен подход к адаптивному изменению метапараметров алгоритма оптимизации на основе "сетки" — двумерного пространства для определения значения критерия эффективности метода. Проанализированы проблемы, связанные с эмпирическим поиском оптимального значения критерия эффективности.

5. Продемонстрирована реализация нового метода в виде алгоритма вычислений и результатов работы соответствующей программной реализации.

6. Проанализированы подходы к возможному практическому применению нового метода в качестве действующего программного инструмента.

7. Представлены иллюстрации, отражающие влияние адаптивных параметров на результат работы алгоритма метода перебора значимых параметров с точки зрения эффективности.

8. Реализована поддержка ряда современных редакторов электронных таблиц для проведения детального анализа промежуточных результатов работы метода за счет механизма сериализации в специальный формат XML, совместимый с LibreOffice Calc, Microsoft Excel, WPS Spreadsheets и другими редакторами.

9. Анализ результатов проведенных вычислительных экспериментов показал, что применение разработанного метода в качестве надстройки над методом перебора значимых параметров увеличивает эффективность последнего на 26,32 %.

Автор выражает благодарность своему научному руководителю, д. т. н. проф. Льву Семеновичу Куравскому.

Список литературы

1. **Городецкий В. И., Карсаев О. В., Самойлов В. В., Себряков С. В.** Прикладные многоагентные системы группового управления // Искусственный интеллект и принятие решений. 2009. № 2. С. 3—24.
2. **Осипов Г. С.** Динамические интеллектуальные системы // Искусственный интеллект и принятие решений. № 1. 2008. С. 47—54.
3. **Осипов Г. С.** Методы искусственного интеллекта. М.: Физматлит, 2011. 296 с.
4. **Aras R., Dutech A., Charpillet F.** Cooperation through communication in decentralized Markov games // International Conference on Advances in Intelligent Systems — Theory and Applications — AISTA'2004. Luxembourg-Kirchberg/Luxembourg, 2004.
5. **Boutilier C.** Planning, learning and coordination in multiagent decision processes // Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge, Morgan Kaufmann Publishers Inc. 1996. P. 195—210.
6. **Claus C. & Boutilier C.** The dynamics of reinforcement learning in cooperative multiagent systems // Proceedings of the National Conference on Artificial Intelligence. John Wiley & Sons Ltd. 1998. P. 746—752.
7. **Miikkulainen R.** Creating Intelligent Agents in Games (2006). The Bridge, 5—13, 2006.
8. **Owen G.** Game Theory. Academic Press, 1995.
9. **Read M., Möslinger Ch., Dipper T., Kengyel D., Hilder J., Thenius R., Tyrrell A., Timmis J., Schmickl T.** Profiling Underwater Swarm Robotic Shoaling Performance using Simulation // Proceedings of TAROS 2013. 2013. P. 456—462.
10. **Куравский Л. С., Марголис А. А., Юрьев Г. А., Мармалюк П. А.** Концепция системы поддержки принятия решений для психологического тестирования // Психологическая наука и образование. 2012. № 1. С. 56—65.
11. **Куравский Л. С., Мармалюк П. А., Алхимов В. И., Юрьев Г. А.** Математические основы нового подхода к построению процедур тестирования // Экспериментальная психология. 2012. Т. 5, № 4. С. 75—98.
12. **Куравский Л. С., Марголис А. А., Мармалюк П. А., Юрьев Г. А., Думин П. Н.** Обучаемые марковские модели в задачах оптимизации порядка предъявления психологических тестов // Нейрокомпьютеры: разработка и применение. 2013. № 4. С. 28—38.
13. **Марковские модели в задачах диагностики и прогнозирования: учеб. пособ. / Под ред. Л. С. Куравского.** М.: Изд-во МГППУ, 2017. 203 с.
14. **Овчаров Л. А.** Прикладные задачи теории массового обслуживания. М.: Машиностроение, 1969. 324 с.
15. **Куравский Л. С., Мармалюк П. А., Юрьев Г. А., Думин П. А.** Численные методы идентификации марковских процессов с дискретными состояниями и непрерывным временем // Математическое моделирование. 2017. № 5. С. 133—146.

The Method of External Optimization for Methods of Markov Processes Identification

A numerical method for optimization of the identification of Markov processes is developed. The method is used as an additional one, in cases where it is necessary to ensure high efficiency of the primary method for a limited or real time. As an example of such primary method in this article the method of brute force of significant parameters is chosen. The basis of algorithmic and software implementation of the method of external optimization is based on the design pattern "decorator" allowing to modify the input and output of transformed function for the purpose of receiving the transformed data, such as the improved value of the criteria in a given time. The new approach allowed to increase the efficiency of the method of brute force of significant parameters by more than 25 % (compared with the empirical approach to the selection of parameters).

Keywords: numerical method, decorator, Markov process, brute force of significant parameters

DOI: 10.17587/it.24.633-641

References

1. Gorodeckij V. I., Karsaev O. V., Samojlov V. V., Serebrjakov S. V. Prikladnye mnogoagentnye sistemy gruppovogo upravlenija (Applied multiagent systems with group control), *Iskusstvennyj Intellekt i Prinjatje Reshenij*, 2009, no. 2, pp. 3–24 (in Russian).
2. Osipov G. S. Dinamicheskie intellektual'nye sistemy (Dynamic intelligent systems), *Iskusstvennyj Intellekt i Prinjatje Reshenij*, 2008, no. 1, pp. 47–54 (in Russian).
3. Osipov G. S. *Metody iskusstvennogo intellekta* (Methods of Artificial Intelligence), Moscow, Fizmatlit, 2011, 296 p. (in Russian).
4. Aras R., Dutech A., Charpillat F. Cooperation through communication in decentralized Markov games, *In International Conference on Advances in Intelligent Systems — Theory and Applications — AISTA'2004*, Luxembourg-Kirchberg/Luxembourg, 2004.
5. Boutilier C. Planning, learning and coordination in multiagent decision processes, *In Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, Morgan Kaufmann Publishers Inc., 1996, pp. 195–210.
6. Claus C. & Boutilier C. The dynamics of reinforcement learning in cooperative multiagent systems, *In Proceedings of the National Conference on Artificial Intelligence*, John Wiley & Sons Ltd., 1998, pp. 746–752.
7. Miikkulainen R. Creating Intelligent Agents in Games, *The Bridge*: 5–13, 2006.
8. Owen G. *Game Theory*, Academic Press, 1995.
9. Read M., Möslinger Ch., Dipper T., Kengyel D., Hilder J., Thenius R., Tyrrell A., Timmis J., Schmickl T. Profiling Underwater Swarm Robotic Shoaling Performance using Simulation, *In Proceedings of TAROS 2013*, 2013, pp. 456–462.
10. Kuravskij L. S., Margolis A. A., Jur'ev G. A., Marmaljuk P. A. Konceptija sistemy podderzhki prinjatija reshenij dlja psihologicheskogo testirovanija (Conception of decision-making support system for psychological testing), *Psihologicheskaja Nauka i Obrazovanie*, 2012, no.1, pp. 56–65 (in Russian).
11. Kuravskij L. S., Marmaljuk P. A., Alhimov V. I., Jur'ev G. A. Matematicheskie osnovy novogo podhoda k postroeniju procedur testirovanija (Mathematical basics of new approach for building of testing procedures), *Jeksperimental'naja Psihologija*, 2012, vol. 5, no. 4, pp. 75–98 (in Russian).
12. Kuravskij L. S., Margolis A. A., Marmaljuk P. A., Jur'ev G. A., Dumin P. N. Obuchaemye markovskie modeli v zadachah optimizacii porjadka pred#javenija psihologicheskikh testov (Learning Markov models in tasks of order optimization for psychological tests presentation), *Nejrokomp'jutery: Razrabotka i Primenenie*, 2013, no. 4, pp. 28–38 (in Russian).
13. Kuravskii L. S. ed. *Markovskie modeli v zadachah diagnostiki i prognozirovanija: Uceb. posobie* (Markov models in tasks of diagnostics and prognosis: a tutorial / ed. by L. S. Kuravsky), Moscow, Publishing house of MGPPU, 2017, 203 p. (in Russian).
14. Ovcharov L. A. *Prikladnye zadachi teorii massovogo ob-sluzhivaniya* (Applied tasks in queueing theory), Moscow, Mashinostroenie, 1969, 324 p. (in Russian).
15. Kuravskij L. S., Marmaljuk P. A., Jur'ev G. A., Dumin P. A. Chislennye metody identifikacii markovskih processov s diskretnymi sostojanijami i nepreryvnym vremenem (Numerical methods of identification for Markov processes with discrete states and continuous time), *Matematicheskoe modelirovanie*, 2017, no. 5, pp. 133–146. (in Russian).