

Исследование преимуществ эволюционного синтеза робастного lqr-регулятора с вероятностной оценкой качества стабилизации

02, февраль 2016

Пупков К. А.¹, Пузанов А. Н.^{1,*}

УДК 004.94

¹Россия, МГТУ им. Н.Э. Баумана

[*ml-Alexandr@yandex.ru](mailto:ml-Alexandr@yandex.ru)

Введение

Значительный практический интерес в современной теории управления представляют подходы оптимального синтеза систем управления с учетом неопределенности моделей объектов и внешних воздействий. При этом задача оптимального синтеза системы заключается в определении её закона управления, обеспечивающего получение достаточно экстремального значения некоторого показателя эффективности. Когда в системе имеет место значительная неопределенность, необходимо чтобы цель управления достигалась при любом её допустимом значении и в общем случае с соблюдением некоторых ограничений.

Получение стабильных характеристик системы управления, удовлетворяющих исходным требованиям функционирования при различных (априори неизвестных) значениях её параметров таким образом, чтобы реализация её рабочего режима имела при любых требуемых значениях, небольшие отклонения, входит в задачу робастного управления.

Критерий эффективности работы системы, как правило, выражает некоторый показатель соотношения экономичности управления и «стоимости» фазовой траектории достижения цели, что, в частности, характерно для задач стабилизации. Присутствие неопределенности делает важным исследование эффективности работы системы на всей области её допустимых значений. Тогда хорошим результатом можно считать получение прототипа управляющего модуля системы, который соответствует достаточным значением критерия качества на всем требуемом диапазоне значений параметров в вероятностном смысле.

При синтезе стохастических динамических систем управления[1] можно также руководствоваться наихудшим значением показателя качества для самой неблагоприятной точки области параметров, однако, когда известна плотность распределения вероятности значений соответствующих диапазонов значений, можно допустить некоторый уровень риска ухудшения качества режима системы с целью расширения её рабочего диапазона, путем вероятностного усреднения этого показателя. Тогда учет значений показателя качества при менее вероятных значениях рабочего диапазона неопределенности отходит на

второй план, так что при нестрогом оптимальном синтезе есть риск, что обработка режима для менее учитываемых значений может оказаться грубой.

Детальное исследование устойчивости и робастных свойств систем с помощью ЭВМ основано на получении и оценке множества реализаций процессов управления. Такой подход подразумевает условный качественный расчет по результатам моделирования режимов системы для некоторого множества точек, достаточного для оценочного покрытия всего диапазона распределения, т. е. в каждой точке необходимо промоделировать режим и учесть поведение системы для получения результирующей характеристики по всему распределению. Типичным примером такого расчета является обход сетки возможных значений неопределенных параметров[1]. Использование данной процедуры для оценки возможных вариантов установки параметров или вычислительной конфигурации управляющего модуля при синтезе системы требует значительных вычислительных мощностей, так как необходимо получить закон управления, который обеспечит качественную обработку режимов практически на всех точках расчетной сетки.

Если структура системы определена, а неизвестными являются только параметры управляющего модуля, то имеем многовариантную задачу параметрической оптимизации с высокой вычислительной стоимостью расчета целевого функционала, которая как раз и рассматривается в данной работе.

Данная задача не имеет общего аналитического решения, а потому хорошо исследуется метаэвристическими и эволюционными методами глобальной оптимизации[6]. Среди таких методов наибольшее распространение получили генетические алгоритмы [3-5] как наиболее простой и универсальный метод-прототип эволюционного моделирования[3-5]. Генетические алгоритмы впервые были введены Холландом[3,6] и впоследствии были адаптированы для решения многих практических оптимизационных задач[6].

Вообще говоря, выделяют три тесно связанных, но независимо разработанных направления эволюционных вычислений[5]: генетические алгоритмы, эволюционные стратегии и генетическое программирование, однако, в последнее время появилась тенденция их интеграции и комплексирования, так что граница между ними становится все более условной.

Для решения задач параметрической оптимизации появилась тенденция освоения генетических алгоритмов с вещественным кодированием, что сопровождало появлению генетических алгоритмов нового вида. Генетические алгоритмы с вещественным кодированием [4] непосредственно работают с машинными представлениями чисел, так что потребность в операциях кодирования-декодирования отпадает автоматически, а все побитовые манипуляции с хромосомами в общем случае заменяются вещественными функциями со случайными составляющими. Однако помимо собственно несущих значений числовые хромосомы должны хранить информацию об ограничениях, которая используется при их инициализации и изменении.

В настоящей работе предлагается оригинальный параметрический генетический алгоритм с вещественным кодированием, который получает приспособленность генотипов особей – параметрических векторов через их привязку к модели задачи, т. е. посредством установки параметров модели соответствующими значениями хромосом из особей и про-

верки функциональных характеристик модели с этими параметрами. Так как рассматриваются модели с известным распределением их неопределенных параметров, результирующий функционал получается в виде взвешенной суммы функционалов по точкам сетки оценивания распределения. Таким образом, в данной работе предлагаемый алгоритм используется для параметрической оптимизации линейно-квадратичной стабилизации для схемы моделирования общего вида, так что каждая конкретная схема должна иметь, соответствующим образом формализованный вид регулятора для оптимизации (в данной работе, линейно-квадратичный регулятор соответствующей размерности).

Основная особенность алгоритма как эволюционного метода – универсальность применения, что позволяет не предусматривать ограничения на модель, объект стабилизации может быть любой. Время работы алгоритма не фиксировано, в силу его эвристичности, так что критерий его завершения, должен определяться предварительно. Получение около оптимальных решений при увеличении размерности задачи обычно требует значительно больше времени, но и рассмотрение областей неопределенности большей размерности также резко увеличивает время каждого шага оценивания, так что соотношения вычислительной стоимости оценивания для моделей выбранной постановки и соответствующей сложности пространство поиска, требований к обобщенному качеству решения, представляют собой материал для исследования и определения возможности или целесообразности применения соответствующих подходов.

1. Робастная линейно-квадратичная стабилизация

Положим необходимо стабилизировать систему, представленную уравнениями в пространстве состояний:

$$\dot{x}(t) = A(\Delta)x(t) + Bu(t), \quad x(0) = x_0 \quad (1)$$

где некоторая векторная неопределенность Δ входит в матрицу-множитель при векторе состояний $x(t) \in \mathbf{R}^n$, $u(t) \in \mathbf{R}^m$.

Стоимость линейно-квадратичного регулирования – стандартный квадратичный функционал

$$J = \int_0^\infty (x^T(t)Sx(t) + u^T(t)Ru(t))dt \quad (2)$$

где $R = R^T > 0$ и $S = S^T > 0$ – симметричные весовые матрицы. Хотя в общем случае матрица S может зависеть от неопределенности Δ , в то время как R – всегда фиксирована.

Общий вид закона управления регулятора, обеспечивающего стабилизацию системы (1) имеет вид:

$$u(t) = -Kx(t) \quad (3)$$

где $K \in \mathbf{R}^{m \times n}$, должен быть подобран так, чтобы он гарантировал стабильность системы (1) при любых допустимых значениях неопределенности Δ . Если K содержит только постоянные элементы, задача линейно-квадратичного регулирования [2] сводится к определению множества значений его элементов.

Для аппроксимации области возможных значений неопределенности, которые может принимать Δ , можно воспользоваться интервальной сеткой из конечного числа точек [2].

При этом будем предполагать, что неопределенность входит в качестве аддитивной добавки к некоторому фиксированному номинальному значению матрицы A

$$A(\Delta) = A(0) + \Delta \quad (4)$$

Сама область рассматриваемых значений неопределенности ограничивается некоторыми предельными значениями, между которыми её возможные значения некоторым образом распределено.

Для оценки установок регулятора K необходимо подчитать взвешенную сумму функционалов для каждой точки интервальной сетки:

$$J = \sum_{i=1}^N P_i \left(\int_0^{T_{\max}} f(x(t, \Delta_i), K, S, R, t) dt \right) \quad (5)$$

где f - подынтегральное выражение в (2); Δ_i - заданное значение неопределенности из интервальной сетки $\{\Delta_i\}_{i=1}^n$; P_i - вероятность того что значение неопределенности попадет в ближайшую окрестность Δ_i ; $T_{\max}$ - время отведенное на оценку процесса стабилизации.

Так как функционал (5) необходимо минимизировать, а генетический алгоритм использует функцию приспособленности, прирост значений которой требуется отыскивать, для оценки значений приспособленности установок K для алгоритма, эквивалентного критерию (5) можно использовать функционал:

$$Vital(K) = \begin{cases} MaxVital - pJ(K), & J(K) < \frac{MaxVital}{p}, \\ 0, & J(K) \geq \frac{MaxVital}{p}, \end{cases} \quad (6)$$

где $MaxVital$ - максимальная приспособленность (уставка); p - постоянный коэффициент, ориентировочно выбираемый исходя из примерных требований к качеству регулирования; $J(K)$ - функционал (5).

Кроме того, при вычислении функционала (5) необходимо убедиться, что процесс стабилизации завершился за время $T_{\max}$ для любых значений неопределенности из интервальной сетки $\{\Delta_i\}_{i=1}^n$, т. е. что все фазовые координаты пришли в ближайшую окрестность точки стабилизации. Если при каком-то значении неопределенности $\Delta_j, j \in \{1..n\}$ стабилизация не завершена, то необходимо задать нулевую приспособленность для данного варианта установок регулятора.

2. Оригинальный генетический алгоритм параметрической оптимизации с вещественным кодированием

Для решения задачи робастного синтеза установок управления в работе предлагается оригинальный генетический алгоритм с вещественным кодированием, реализующий механизм параметрической оптимизации функционалов заданного контекста и размерности. Рабочее множество алгоритма - популяция особей, каждая из которых содержит вектор числовых хромосом, размерность которого соответствует количеству искомых параметров задачи. Первым этапом подготовки алгоритма к работе является его связывание с функционалом цели, от которого он получает размерность постановки. Далее происходит стандартная подготовка популяции, и её случайная инициализация в соответствии с заданными

ми ограничениями, с возможностью внедрения каких-либо ранее полученных априорных решений.

У предлагаемого алгоритма можно выделить следующие особенности

- Использование оператора *мультикроссинговера* в паре с процедурой *виртуальной микроэволюции*, при этом мультикроссинговер с лучшей особью популяции рассматривается как его отдельная модификация. Также предусмотрена его модификация для задач повышенной размерности – блочный мультикроссинговер, при этом процедура виртуальной микроэволюции преобразуется соответствующим образом.
- Использование вложенных простых (однопараметрических) генетических алгоритмов ПГА с небольшим числом операторов и размером локальной популяции – монохромосомная микроэволюция. При этом для задач повышенной размерности, предусмотрена реализация вложенных генетических алгоритмов небольшой размерности (по нескольким параметрам)
- Индексирование модификаций операторов с целью оценки их текущей эффективности на основе предыдущей статистики их использования.
- Использование ранговой выборки при осуществлении выбора особей для бинарного партнерства; при этом в факторе формирования ранга второго партнера учитывается расстояние от него до первого.
- Использование априорных мутационных стратегий.

Рассмотрим каждую из них в отдельности:

2.1. Мультикроссинговер и виртуальная микроэволюция

Мультикроссинговер – оператор, реализующий процедуру получения n пар потомков от двух особей, имеющих по n хромосом, причем каждая k -ая пара $k \in \{1, \dots, n\}$ получается из исходной родительской пары путем применения одного из стандартных операторов кроссинговера к k -ым хромосомам этой пары, остальные хромосомы пары потомков передаются от пары родителей без изменений.

Структурная схема оператора мультикроссинговера представлена на рис. 1.

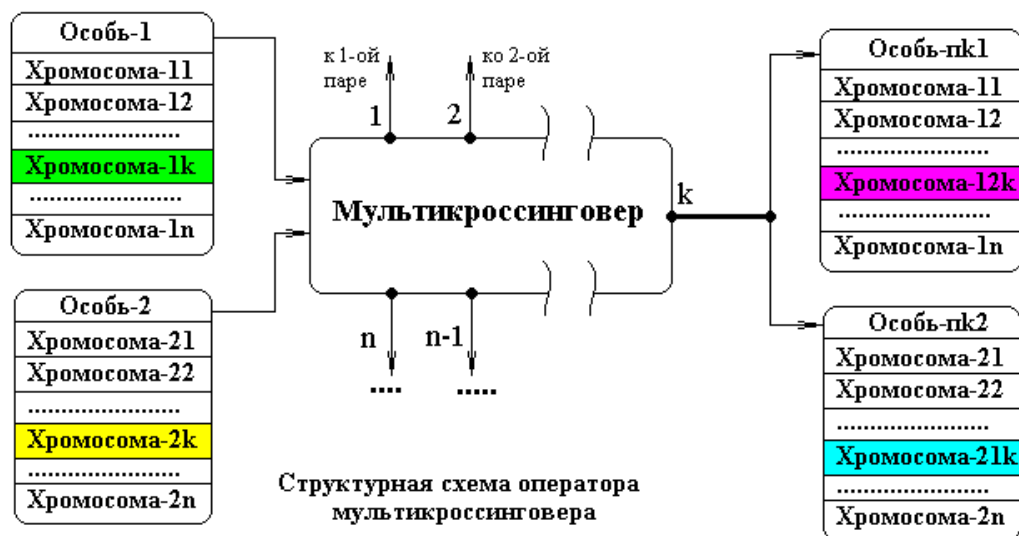


Рис. 1. Структурная схема оператора мультикроссинговера

Первичные потомки, полученные в результате мультикроссинговера, претерпевают процедуру виртуальной микроэволюции. Виртуальная микроэволюция (микроэволюция первого рода) – процедура, применяемая для поэтапного отбора и улучшения потенциальных потомков, полученных в результате мультикроссинговера.

Основные этапы микроэволюции:

1) Сортировка потенциальных потомков по витальности.

2) Тиражирование лучшей текущей пары потомков в зону мутации (все особи с витальностями ниже, чем у лучших двух) с вытеснением менее приспособленных.

3) Цикл мутации (повторяется заданное число раз):

3.1) Применения оператора мутации заданной интенсивности к особям, находящимся в зоне мутации

3.2) Сортировка, измененного в нижней части, множества потенциальных потомков по витальности

3.3) Тиражирование лучшей текущей пары потомков в зону мутации с вытеснением менее приспособленных.

4) Возврат лучшей пары потомков в основную популяцию.

В качестве стандартного кроссинговера хромосом в мультикроссинговере используется модификация плоского кроссинговера [4].

При использовании оператора мультикроссинговера, необходимо иметь в виду, что его эффективность достигает своего максимума для задач средней размерности. При решении задач высокой размерности его эффективность с её ростом начинает заметно снижаться.

Главной причиной снижения эффективности мультикроссинговера с ростом размерности задачи является резкое увеличение его вычислительной стоимости, связанное с разрастанием множества потенциальных потомков, без учета виртуальной микроэволюции, обращение к функционалу цели осуществляется минимум $2n$ раз, где n – размерность задачи. Обработка множества потенциальных потомков в процессе виртуальной микроэволюции также становится еще более затратной процедурой, а эффективность осуществляемых преобразований, напротив, падает. Это связано с тем, что преобразования отдельной выбранной, даже чувствительной хромосомы при решении задач многоэкстремального высоко размерного поиска, как правило, не может обеспечить существенных позитивных сдвигов всего эволюционного процесса. Поэтому для таких задач целесообразно ввести особую модификацию данного оператора. Если разбить исходное множество хромосом особей задачи на отдельные блоки, то мультикроссинговер можно осуществлять не по отдельным хромосомам, а сразу по таким блокам. Данную модификацию оператора будем называть блочным мультикроссинговером.

Блочный мультикроссинговер – оператор реализующий процедуру получения p пар потомков от двух особей имеющих по n хромосом, множество которых разбито на p ($p < n$) блоков, причем каждая k -ая пара $k \in \{1, \dots, p\}$ получается из исходной родительской пары

путем применения одного из стандартных операторов кроссинговера к одноименным хромосомам k -го блока этой пары, остальные блоки хромосом пары потомков передаются от пары родителей без изменений.

Структурная схема блочной модификации оператора мультикроссинговера показана на рис. 2.

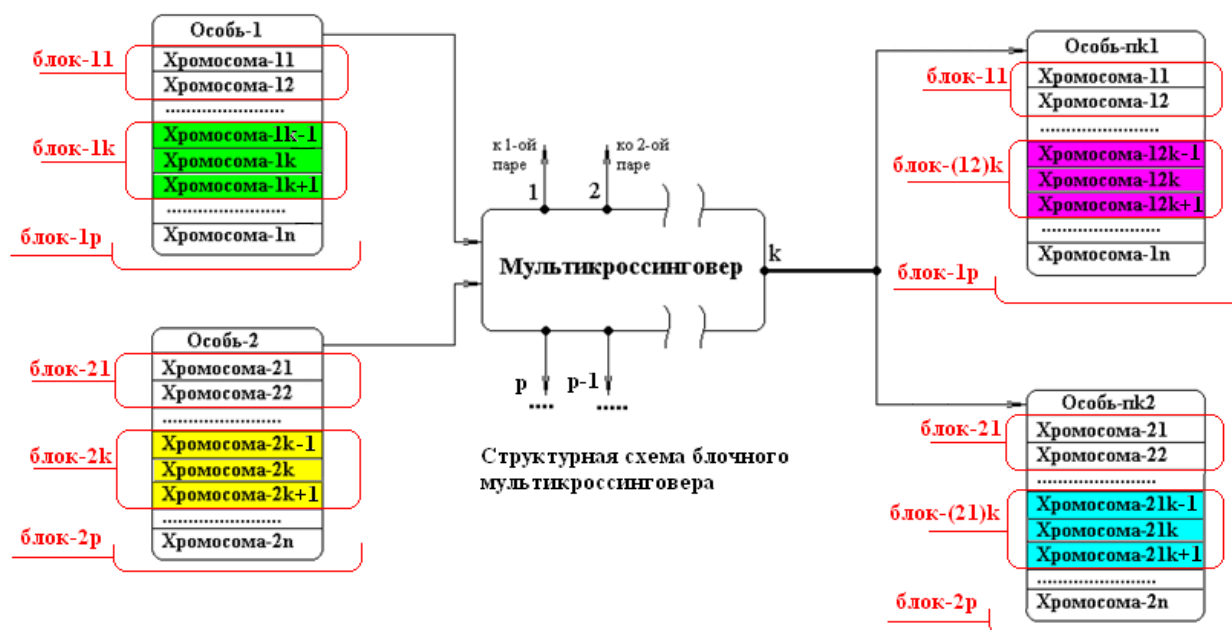


Рис. 2. Структурная схема блочного мультикроссинговера

2.2. Вложенные ПГА (монокроссомная микроэволюция)

Сущность монокроссомной микроэволюции заключается в организации вложенного простого генетического алгоритма над множеством особей локальной мини-популяции на основе выбранной базовой особи. Монокроссомная микроэволюция (МХМЭ) имеет смысл проводить не более одного раза за репродукцию, в первую очередь из-за её высокой вычислительной стоимости. В качестве базовой особи обычно целесообразно выбирать текущую лучшую особь. Сущность вложенных ПГА заключается в следующем: выбранная особь тиражируется на всю локальную популяцию, после чего у всех тиражированных копий, за исключением одной, случайным образом инициализируется одна и та же предварительно выбранная хромосома. Далее особи участвуют в простом генетическом алгоритме преобразований по данной хромосоме, остальные хромосомы на протяжении всего процесса остаются замороженными. На выходе, после заданного числа поколений, имеем лучшую особь, возвращаемую в главную популяцию. На рис. 3 представлена иллюстрация процесса монокроссомной микроэволюции. Монокроссомную микроэволюцию, проводимую по различным хромосомам можно рассматривать как её отдельную модификацию, что позволит осуществлять выбор хромосомы для нее на основе опытного контроля.

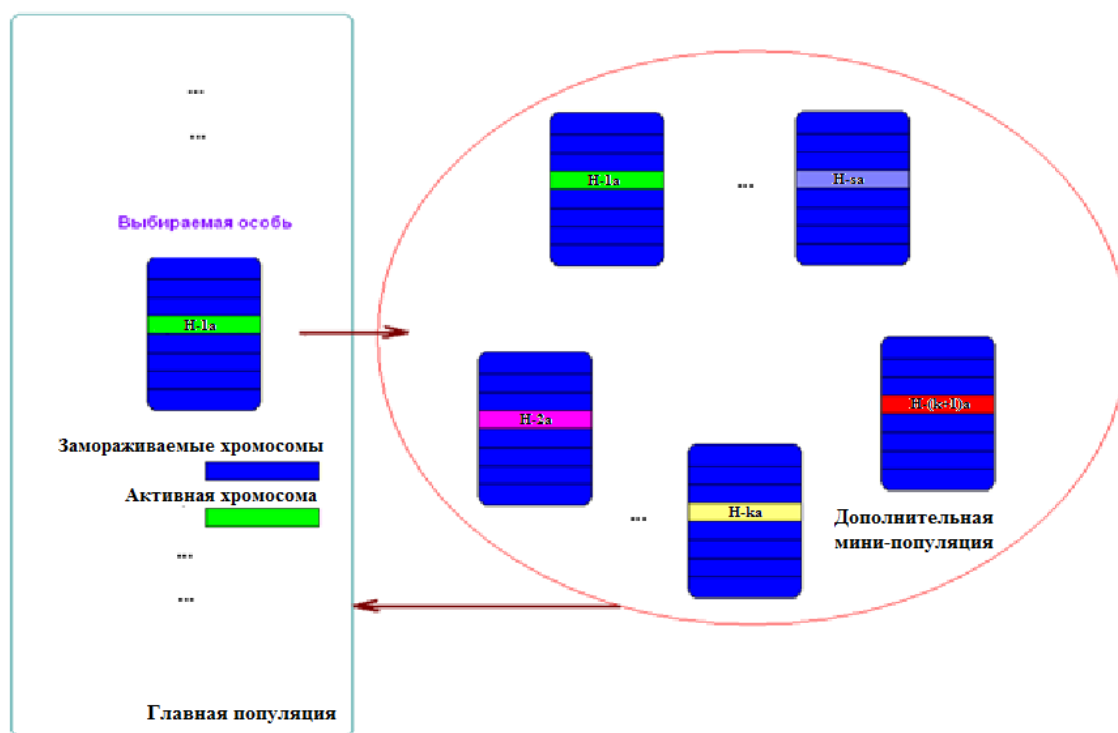


Рис. 3. Иллюстрация использования вложенных простых генетических алгоритмов (монохромосомной микроэволюции)

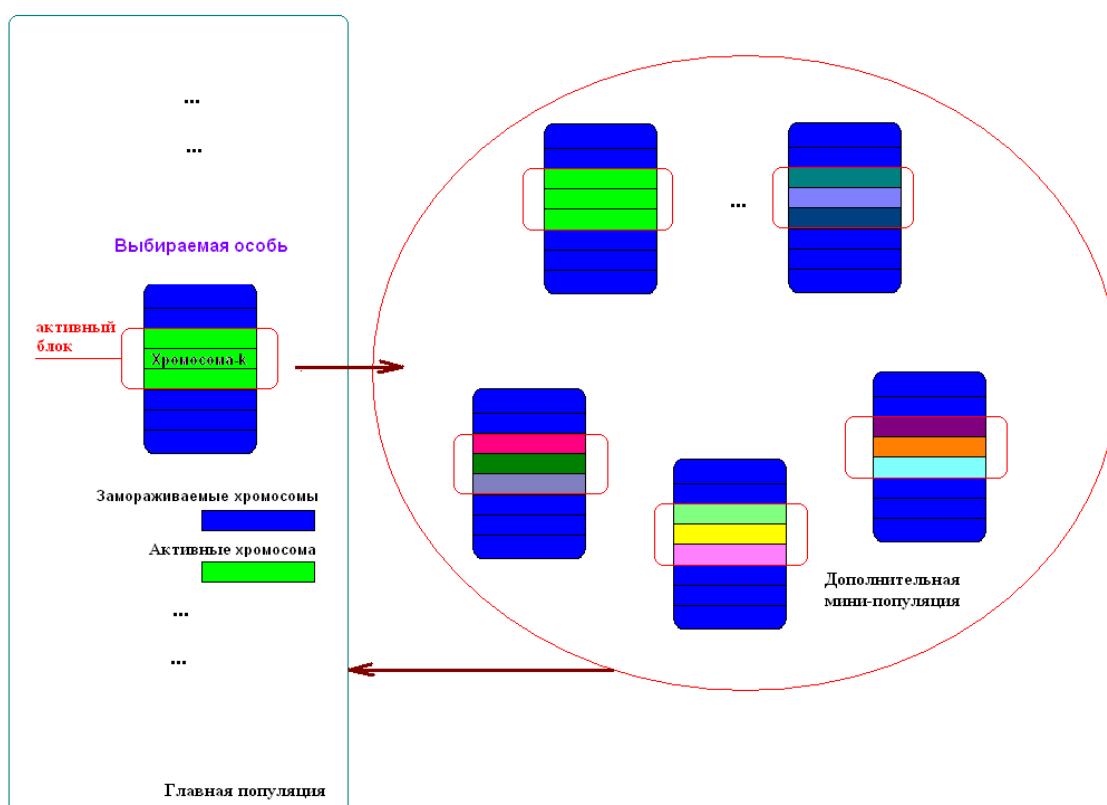


Рис. 4. Иллюстрация использования вложенных генетических алгоритмов по подмножеству хромосом (полихромосомной микроэволюции)

Как отмечалось выше, невысокая эффективность монохромосомных преобразований для задач высокой размерности делают процедуру монохромосомной микроэволюции малоприменимой для использования в таких случаях. Решения данной проблемы можно найти аналогичным способом, как это было сделано для оператора мультикроссинговера: при сохранении общей логики работы подпроцесса мы можем перейти от использования одной активной хромосомы микроэволюции к использованию некоторого их набора. В этом случае процедура монохромосомной микроэволюции преобразуется в полихромосомную микроэволюции (вложенный ГА небольшой размерности). Тогда эволюции осуществляется по некоторому набору активных хромосом и в её ходе могут быть использованы такие операторы, как например мультикроссинговер. Иллюстрация полихромосомной микроэволюции приведена на рис. 4.

2.3. Индексирование операторов

Используя имеющиеся множество различных операторов и их модификаций, алгоритм на протяжении своей работы накапливает определенную статистику их использования, которая может быть им использована для периодической самоподстройки. Индекс оператора – оценка его текущей эффективности, которую целесообразно учитывать во время очередной самоподстройки.

После применения оператора срабатывает счетчик, подсчитывающий общее число его применений с момента старта алгоритма. Далее, если произошла эволюция лучшего решения индекс пользы оператора увеличивается на определенную уставку, если же просто произошло увеличение средней витальности аргументов оператора, индекс пользы увеличивается на 1. Приведенный индекс оператора рассчитывает, как индекс его пользы деленной на общее число его использований. Хотя без учета информации об окрестности пространства поиска, в которой применяется оператор, такой индекс является весьма условной оценкой его эффективности.

2.4. Априорные мутационные стратегии

Мутационная стратегия – зависимость интенсивности мутации, применяемой в качестве служебного компонента в той или иной области алгоритма (например, во время виртуальной микроэволюции), от номера поколения и параметров популяции. Например, можно выделить экспоненциальные стратегии, различной грубости:

$$\begin{aligned} \text{MCMI}_{\text{Light}} &= \text{MCMI}_{\text{Nominal}} \cdot \exp(-\tau/T) \cdot f(\dots) \\ \text{MCMI}_{\text{SoftNormal}} &= \text{MCMI}_{\text{Nominal}} \cdot \exp(-\tau/T^2) \cdot f(\dots) \\ \text{MCMI}_{\text{Normal}} &= \text{MCMI}_{\text{Nominal}} \cdot \exp(-\tau/T^3) \cdot f(\dots) \\ \text{MCMI}_{\text{RudeNormal}} &= \text{MCMI}_{\text{Nominal}} \cdot \exp(-\tau/T^4) \cdot f(\dots) \\ \text{MCMI}_{\text{Rude}} &= \text{MCMI}_{\text{Nominal}} \cdot \exp(-\tau/T^5) \cdot f(\dots) \end{aligned} \quad (7)$$

где τ – номер поколения; T – постоянная затухания мутации, а $\text{MCMI}_{\text{Nominal}}$ – номинальная интенсивность мутации.

2.4. Особенности выбора особей для партнерства

Во многих реализациях генетических алгоритмов в качестве основного метода отбора особей для участия в репродуктивных операторах часто используется метод колеса рулетки (метод пропорционального отбора). Однако данный способ имеет выраженный недостаток, – в случае, когда значения функции приспособленности для одной-двух особей оказывается существенно больше соответствующих значений для остальных членов популяции или же группа особей с относительно высокой приспособленностью сосредоточена в окрестности некоторого локального экстремума, вероятность выбора последних может оказаться пренебрежимо малой величиной, что может обусловить застойные эффекты в эволюции, так как образуется достаточно мощный центр притяжения, так что вся популяция постепенно локализуется в окрестности достаточно выраженного экстремума.

Наиболее простым и универсальным способом, позволяющим решить данную проблему, является использование рангового отбора. При этом каждой особи присваивается ранг, соответствующий её положению в последовательности, сортированной по фактору формирования ранга, а это обычно приспособленность. Если выборка происходит из N особей, их можно ранжировать натуральными числами $\{1..N\}$. Генетический алгоритм, использующий ранговую селекцию, более направлен на экстенсивное освоение пространства поиска: он медленнее осваивает текущие окрестные холмы и поэтому дольше сходится, нежели чем алгоритм, использующий пропорциональный отбор, но в итоге осуществляет более глобальный обзор и лучше подходит для сложных многоэкстремальных задач. Пример сравнения ранговой и пропорциональной селекции представлен на рис. 5.

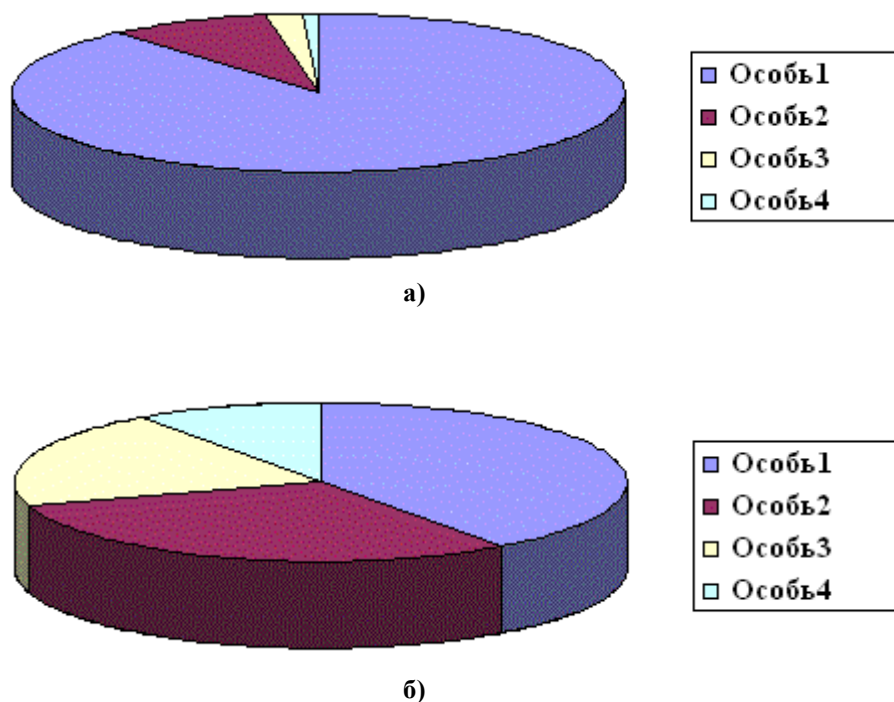


Рис. 5. Сравнение выборки методом рулетки и ранговой селекции

Разработанный алгоритм использует модифицированную ранговую селекцию. При выборе первого партнера ранги особей множества выбора формируется по витальности. При выборе второго партнера учитывается также и расстояние от него до первого, предпочтительными полагаются более удаленные от первого партнера особи.

$$Key(j) = Vital(j) \cdot \left(1 + \eta \cdot \frac{\rho(i, j)}{\rho_{nom}} \right) \quad (8)$$

где $Vital(j)$ - витальность второго партнера;

$\rho(i, j)$ - евклидово расстояние между первым и вторым партнером;

ρ_{nom} - некоторое характерное расстояние;

η - коэффициент учета различия;

Критерий формирования ранга второй особи (8) целесообразно использовать для рельефов задач, имеющих значительное число областей экстремумов, так как это повышает вероятность выбора достаточно приспособленных особей, принадлежащих разным областям, что дает возможность исследовать пространство между ними.

3. Эволюционное моделирование задачи линейно-квадратичной стабилизации модели кренового движения самолета (пример задачи для решения и исследования предлагаемым алгоритмом).

Рассмотрим задачу оптимизации регулятора кренового канала самолета приведенную в [2]. Уравнение динамики модели кренового движения самолета в пространстве состояний имеет вид:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & L_p & L_\beta & L_r \\ g/V & 0 & Y_\beta & -1 \\ N_{\dot{\beta}} \cdot (g/V) & N_p & N_\beta + N_{\dot{\beta}} Y_\beta & N_r - N_\beta \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & -3.91 \\ 0.035 & 0 \\ -2.53 & 0.31 \end{bmatrix} u(t)$$

где x_1 - угол крена, x_2 - производная угла крена, x_3 - угол скольжения, x_4 - угловая скорость рысканья; u_1 - отклонение рулей направления, u_2 - отклонение элеронов.

Будем рассматривать следующие номинальные значения параметров модели:

$L_p = -2.93$; $L_\beta = -4.75$; $L_r = 0.78$; $g/V = 0.086$; $Y_\beta = -0.11$; $N_{\dot{\beta}} = 0.1$; $N_p = -0.042$; $N_\beta = 2.601$ и $N_r = -0.29$;

и начальные условия

$$x(0) = \begin{bmatrix} \frac{\pi}{6} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Если допустить неопределенность сразу по многим параметрам, то расчет функционала (5) потребует большого объема вычислений, а поскольку при выполнении алгоритма он вызывается многократно, время его работы окажется чрезмерно большим и оценивание гарантировано только для заданных начальных условий. В этом случае более целесообразно рассматривать робастное адаптивное управление.

Поэтому введем неопределенность только для первых трех параметров модели L_p , L_β и L_r : номинальное значение каждого из этих параметров считается возмущенным относительной неопределенностью, которая может достигать 15 %. Также положим, что неопределенность распределена равномерно и выберем весовые матрицы критерия (5): $S = 0.1I$ и $R = I$.

Модель задачи выполнена в виде схемы Matlab Simulink, которая предварительно загружается, перед тем как matlab функция (заранее подготовленная функция на языке matlab) может рассчитывать значение функционала (5) на основании моделирования схемы во всех точках интервальной сетки за время оценивания $T_{max} = 10$ с.

Параметры управляющего модуля в законе управления (3) задаются матрицей

$$K = \begin{bmatrix} K_1 & K_2 & K_3 & K_4 \\ K_5 & K_6 & K_7 & K_8 \end{bmatrix}$$

Область поиска:

$$-50 \leq K_i \leq 50, \quad i \in \{1..8\}$$

Предлагаемый генетический алгоритм, реализован на языке программирования C++/CLI (C++ для .Net Framework), а расчет значений приспособленности особей (6) выполняется через класс-функционал, взаимодействующий со средой Matlab на основе COM – интерфейса.

Для сокращения времени вычислений при расчете функции приспособленности будем учитывать неопределенность только по первым двум параметрам, а проверку полученного решения будем делать с расчетом на неопределенность по всем трем.

В результате работы алгоритма с основной популяцией из 100 особей через 20 поколений было получено решение:

$$K^* = \begin{bmatrix} -32.517 & -24.2163 & 24.3699 & -19.504 \\ -33.0304 & -41.5889 & -31.3845 & 31.4114 \end{bmatrix}$$

Его приспособленность по 1000 – бальной шкале:

$$Vital(K^*) = RobustLQRVital(K^*) = 938.0411$$

1000 – бальный тест решения для неопределенности по трем параметрам:

$$TRobustLQRVital(K^*) = 937.9444$$

Процессы стабилизации с полученными параметрами управляющего модуля при различных значениях неопределенности приведены на рис. 6,7.

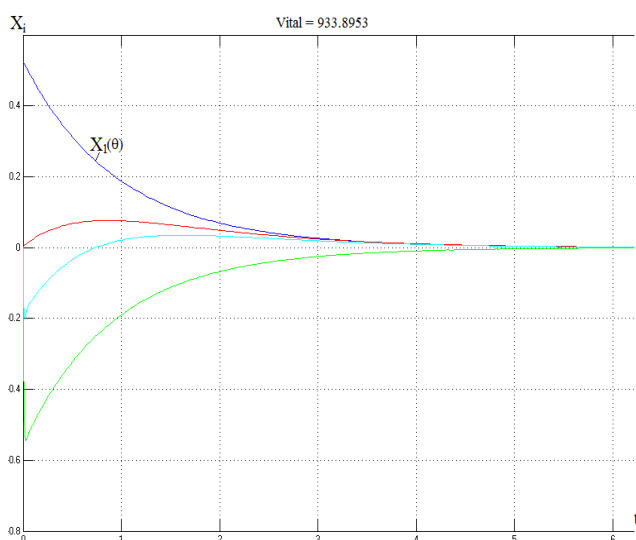


Рис. 6. Процесс стабилизации при $L_\beta = -5.4625$; $L_p = -3.3695$; $L_r = 0.78$
приспособленность $Vital = 933.8953$

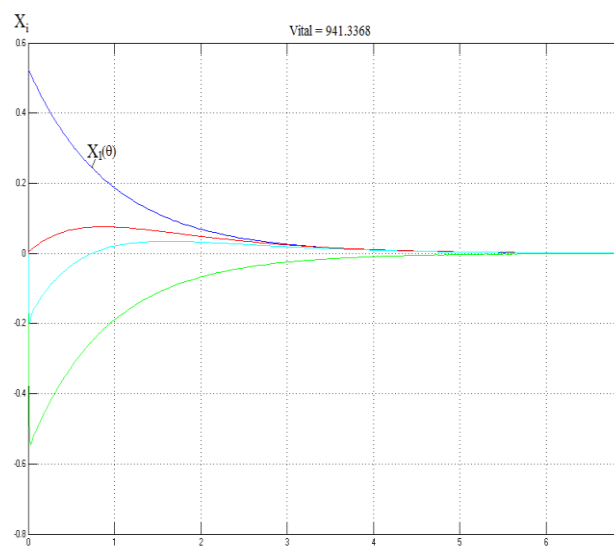


Рис. 7. Процесс стабилизации при $L_\beta = -4.0375$; $L_p = -2.4905$; $L_r = 0.8970$
приспособленность $Vital = 941.3368$

Как видно из представленных рисунков, фазовые траектории стабилизации при заданных крайних значениях первых трех параметров модели, почти не отличаются, что позволяет говорить о робастности полученного решения в рассматриваемой области возможных вариаций.

Заключение

Предложенный генетический алгоритм с вещественным кодированием позволяет синтезировать параметры линейно-квадратичного регулирования, робастно обеспечивающие качественную стабилизацию для широкого диапазона значений неопределенных параметров, в общем случае, не предусматривая ограничения на модель объекта, что показано на примере линейной системы. Время работы алгоритма для приемлемого эффективного освоения области возможных решений оказывается достаточно, когда количество рассматриваемых параметров неопределенности невелико, однако, при рассмотрении более широких рабочих диапазонов, вычислительные затраты алгоритма обычно резко увеличиваются, поэтому для сокращения вычислений, как правило, достаточно учитывать неопределенность только по тем параметрам, которые наиболее подвержены изменениям или к которым система наиболее чувствительна. На примере линейной системы, в работе также показано, что при проверке характерного качества решения при расширении области допустимых значений неопределенности до большей размерности, достаточная оптимальность результатов, полученных в ходе эволюции по меньшему числу параметров, обеспечивает тенденцию к обобщению – неоцененный достаточный запас робастности качества при учете возможности дополнительной вариации параметров, т. е. достаточно оптимальные решения, полученные для небольшого числа параметров, могут покрывать гораздо

более широкие области функционирования, но, так или иначе, требуют аттестационных проверок на расширенных диапазонах.

Таким образом, в работе показано, что эволюционные методы, в общем случае могут обеспечить получения робастных законов управления для систем произвольной сложности, однако специфика вариантов их выбора и подготовки определяет примерный уровень возможно ожидаемых вычислительных затрат по отношению к каждой конкретной системе, что позволяет выявить насколько постановка задачи может быть упрощена для поиска эффективных решений исходной постановки, либо оценить насколько применение метода, может быть, использовано для её исследования или решения.

Список литературы

- [1]. Пупков К.А., Фам Суан Фанг. Вероятностная неопределенность в стохастических динамических системах: устойчивость и управление. М.: РУДН. 2014. 118 с.
- [2]. Polyak B.T., Tempo R. Probabilistic robust design with linear quadratic regulators // Systems & Control Letters. 2001. Vol. 43. Is. 5. P. 343-353. doi:10.1016/S0167-6911(01)00117-7
- [3]. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. М.: ФИЗМАТЛИТ. 2003. 432 с.
- [4]. Дударов С.П. Математические основы генетических алгоритмов: учеб. пособие. М.: РХТУ им. Д. И. Менделеева. 2012. 56 с.
- [5]. Thomas Back, Ulrich Hammel, Hans-Paul Schwefel. Evolutionary Computation: Comments on the History and Current State // IEEE Transactions on evolutionary computations. 1997. Vol. 1. Is. 1. DOI: 10.1109/4235.585888
- [6]. Thomas Weise. Global Optimization Algorithms “Theory and Application” 2nd ed. / Version: 2009-06-26. 2009. 821 p. Режим доступа: <http://www.it-weise.de/projects/book.pdf> (дата обращения: 7.01.2015)