

Введение Решение задач проектирования оптимальных ХТС должно проводиться с учетом существующей неопределенности в информации об условиях эксплуатации проектируемой ХТС. Наличие неопределенности в исходной информации обусловлено неточностями в математических моделях, используемых в процессе проектирования ХТС, неточных сведениях о внешних и внутренних условиях эксплуатации ХТС. Учет неопределенности в задачах проектирования оптимальных ХТС приводит к включению в число параметров, характеризующих ХТС, неопределенных параметров. При этом необходимость выполнения проектных ограничений на этапе функционирования ХТС приводит к появлению в формулировках задач оптимального проектирования бесконечного количества ограничений. Формулировки задач проектирования с учетом неопределенности в исходной информации принимают вид одноэтапной и двухэтапной задач. В зависимости от того, какие требования накладываются на выполнение проектных ограничений, задачи приобретают мягкие (выполнение условий в среднем или с заданной вероятностью) или жесткие ограничения, выполнение которых должно быть безусловным. Одноэтапная задача оптимизации формулируются для случаев, когда на этапе проектирования недостаточно сведений о условиях эксплуатации проектируемой ХТС. В таком случае на этапе проектирования не закладывается возможность подстройки управлений под изменяющиеся условия эксплуатации [1, 2, 3]. Для случая жестких ограничений задача имеет следующий вид [4] , (1) , , (2) где , – вектор конструктивных переменных, – вектор управляющих переменных, – вектор неопределенных параметров, , – математическое ожидание функции за весь период эксплуатации ХТС, – проектные ограничения. В случае наличия вероятностных ограничений задача примет вид [2] , (3) , , (4) где – вероятность выполнения ограничения . Формулировка двухэтапной задачи оптимизации предусматривает учет подстройки управляющих переменных под изменяющиеся условия эксплуатации ХТС [5,6]. В случае наличия жестких ограничений задача имеет вид [5] (5) . (6) Здесь (6) – требование выполнения ограничений на всей области неопределенности. При наличии вероятностных ограничений задача примет вид (7) , , . (8) Сложностью решения задач (1, 2), (3, 4), (7, 8) является необходимость проведения многомерного интегрирования для вычисления критерия и ограничений задачи на каждом шаге решения задачи оптимизации. Задача (5, 6) осложняется наличием требования работоспособности (6), описанного многоэкстремальной недифференцируемой функцией. Подходы к решению задач (3, 4), (5, 6), (7, 8) предложены в работах [2], [5], [6], [7], [8], [9] . Особенностью предложенных подходов является сведение исходной задачи оптимизации к последовательности задач полубесконечного программирования (semi-infinite problem, SIP). Задача полубесконечного программирования в общем виде может быть представлена как (9) , . (10) В этой задаче присутствует бесконечное количество ограничений (10) при конечном числе поисковых

переменных. Задачи полубесконечного программирования нашли применение в многих областях: оптимальное управление, оптимальное проектирование [1–9], [10], задачи контроля выбросов в атмосферу [11], задачах Чебышева-Стилтьеса, при интегрировании функций [12], в задачах с граничными условиями, задачах собственных значений [13, 14, 15] и многих других. Следует подчеркнуть, что с вычислительной точки зрения SIP задачи гораздо более трудны для решения, чем задачи с конечным числом ограничений. Основная причина заключается в трудности определения допустимости выбранного решения. Для конечной задачи: , необходимо вычислить только значений функций и проверить, все ли они являются неположительными. Для SIP проверка допустимости эквивалентна решению задачи поиска глобального максимума по переменной : и проверке, не нарушает ли глобальное решение условия [16]. Необходимо отметить, что даже для линейных полубесконечных задач (LSIP), задача , в общем случае, не является выпуклой. Поэтому ее решение может потребовать значительных вычислительных ресурсов. Задачи полубесконечного программирования изучаются начиная с 60-х годов. Хотя первоначально задачи SIP были связаны с проблемой Чебышевской аппроксимации, в классической работе Наага, посвященной линейным SIP [17], в условиях оптимальности Фрица Джона [18], сам термин полубесконечное программирование был введен в 1962 г. Charnes, Cooper и Kortanek в нескольких работах, посвященных линейным SIP [19, 20, 21]. Значительный вклад в развитие приложений SIP в экономике, теории игр, механике, статистических задачах и других областях внес Tschernikow в работе [22]. Методам решения задач полубесконечного программирования посвящено значительное число работ [23, 24, 25, 26, 27, 28, 29, 30, 31, 11, 12, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. Первые численные методы решения задач SIP в начале 1970-х были разработаны Gustafson и Kortanek в ряде работ, например в [51]. Теоретические основы задач полубесконечной оптимизации достаточно глубоко развиты в [23, 33]. Вычислительные особенности применения методов для решения задач аппроксимации и оптимального управления рассмотрены в задачах [11, 25, 31, 30, 12 и др.]. Особо следует выделить обзоры Polak E. [30], Hettich R., Kortanek K.O. [25] задач полубесконечной оптимизации с точки зрения недифференцируемой и гладкой оптимизации соответственно, обзор Reemtsen R., Gornier S. [37] численных методов для общих гладких задач SIP, обзор G. Still, M. Lopez [46]. Методы для решения задач полубесконечной оптимизации являются естественным продолжением методов для решения задач математического программирования. Рассмотрим класс алгоритмов, в которых необходимо найти решение задачи (9) с заданной точностью и проверить ее выполнение. Общая схема таких подходов состоит в процедуре дискретизации. Исходная задача полубесконечной оптимизации (9) заменяется последовательностью обычных задач оптимизации вида (11) , , , , (12) где – номер итерации в методе решения задач, множество

есть конечное множество значений переменной n , – размерность множества S . Задача (11) есть конечный аналог задачи SIP (9). Поскольку задача (11) является конечной задачей нелинейного программирования, ее можно решить с помощью известных алгоритмов нелинейного программирования, например последовательного квадратичного программирования. От способа дискретизации существенно зависит как «качество» аппроксимирующих задач, так и эффективность всего вычислительного процесса. Методы, основанные на дискретизации называются методами последовательной аппроксимации. В них на каждой итерации некоторые из исходных ограничений отсутствуют, и элементы последовательности S_k , будут недопустимыми по отношению к исходной области S . Поэтому такие алгоритмы относятся к методам внешней аппроксимации. Их можно разделить на два класса в зависимости от способа формирования множеств S_k . К первому классу методов последовательной аппроксимации принадлежат методы, для которых на каждой итерации k образом, на каждом шаге находятся новые ограничения и задача решается без учета старых ограничений. К методам последовательной аппроксимации, не наследующим элементов множества от итерации к итерации, можно отнести некоторые методы обмена [23], метод Eaves-Zangwill [26] и другие методы [52, 53]. Класс алгоритмов, в которых делится на активные и пассивные методы. В активных методах для поиска критических точек для формирования множества S_k сначала решаются внутренние задачи P_k , $k = 1, 2, \dots$. (13) Решением задачи (13) является $\{x^k\}$. В зависимости от того, является ли неотрицательным числом, изменяется способ вычисления направления движения метода. Если выполняется условие $\alpha_k > 0$, (14), то точка называют критической. Так как на каждой итерации множество S_k не включает точки множеств S , то обычно множества включают малое количество точек. Недостаток этих методов заключается в том, что часто трудно проверить является ли глобальным максимум, полученный при решении внутренних задач (13). К активным методам относятся, например, алгоритмы, основанные на методе квазиградиентов [52, 53], которые в то же время являются прямыми градиентными методами. В пассивных методах для формирования множеств нет необходимости каждый раз вычислять максимум функции f . Элементы для множеств находят с использованием случайных величин. Ко второму классу можно отнести методы, наследующие множества точек S_k , т.е. в которых $S_k \subset S_{k-1}$. В этом случае во множество S_k входят все или некоторые из точек множеств S_{k-1} . К методам этого класса относится большинство методов, используемых для решения задач SIP, такие как 1. методы дискретизации, например [23, 32, 33, 37], 2. полунепрерывные [33, 37], 3. методы, основанные на локальной редукции [54, 23, 37, 33]. Рассмотрим основные особенности этих методов. Методы дискретизации. Основной идеей методов дискретизации является решение задачи (11) на конечном подмножестве ограничений (12) бесконечного множества ограничений (10), и повторении этой процедуры для расширенного

множества ограничений, если требуется более высокая точность. Более подробно, идея заключается в последовательном вычислении «приближенного решения» дискретизированной задачи полубесконечной оптимизации (11) для по одному из алгоритмов конечномерной оптимизации. Используемые в методах дискретизации множества точек обычно задаются заранее. В ряде алгоритмов множество точек определяется адаптивно, с использованием на k -ой итерации дискретизации информации о точках, полученной на предыдущих шагах. Такой вариант метода наиболее эффективен и последовательно уточняет дискретизацию с учетом результатов предыдущей итерации. Метод дискретизации основан на построении сетки, построенной с некоторым шагом на области Ω . Алгоритм методов дискретизации в общем виде можно представить в виде [25] а) задается номер итерации k , предельное число итераций метода K , шаг h , некоторое множество точек S_k , n_k – целое, $n_k \geq 1$; б) находим x^k – решение задачи (11) на множестве S_k ; в) находим x^k ; г) строим новую сетку S_{k+1} , выбираем на ней новое множество S_{k+1} , учитывая x^k , n_{k+1} ; д) вычисляем решение задачи (11) на множестве S_{k+1} . Если удовлетворяет с заданной точностью задаче (11) на множестве S_{k+1} , переходим на шаг е). В противном случае выполняем шаг г). е) если номер итерации $k = K$, задача решена, стоп. В противном случае $k = k + 1$, переходим на шаг б). Методы дискретизации состоят не просто в замене бесконечного множества ограничений конечным подмножеством несвязных ограничений, но и в «запоминании» результатов в выбранной стратегической сетке. Пусть будет конечным подмножеством S_k и Ω_k . Тогда для достаточно малого h , решение на множестве S_k вместо x^k будет хорошей аппроксимацией для исходной задачи. Для доказательства сходимости методов дискретизации, необходимо, чтобы каждая предельная точка последовательности «решений» задач (11) была решением задачи SIP (9), где «решение» – это такая точка, к которой сходится соответствующий алгоритм решения задачи (11). Что касается решения аппроксимирующей задачи (11), то здесь подходят не все методы математического программирования, так как часто во многих из них требуется решение подзадач, которые имеют такое же число ограничений как и исходная задача. Следует отметить работу Polak E., Ne L. [55], в которой рассматривается проблема задания сетки и выбора числа n_k на шаге а), такого, чтобы увеличить сходимость метода решения задачи SIP. Для улучшения эффективности метода дискретизации важно, чтобы информация о решении задачи, полученная на одной итерации, передавалась на следующую итерацию, что означает, что приближенное решение задачи (11) на k -ой итерации может быть использовано в качестве начальной точки для решения задачи (11) на $k+1$ -ой итерации. Однако, такая начальная точка может быть недопустимой на $k+1$ -ой итерации, так как на этой итерации задача (11) включает дополнительные или различные с k -ой итерацией ограничения. Поэтому многие методы, которые нуждаются в допустимой стартовой точке слишком затратны. Однако методы дискретизации могут быть использованы для получения первого приближения к

решению, которое может быть улучшено на следующем шаге методом, обладающим хорошей локальной сходимостью. Полунепрерывные методы. К методам, основанным на последовательной аппроксимации относятся и полунепрерывные методы, или методы обмена. Они могут также работать с дискретизированными задачами полубесконечной оптимизации, поэтому многие из них дают базис методам дискретизации. Основная идея таких методов выглядит следующим образом: пусть на k -ой итерации задано конечное подмножество S_k и пусть будет решением задачи (11) на множестве S_k . Необходимо определить точки $x^1, \dots, x^m$, для которых достигается своего локального максимума, и выбрать в соответствии с некоторыми заданными правилами. В общем виде алгоритм методов обмена выглядит следующим образом: а) Задать некоторое конечное множество точек S_k , номер итерации k . б) Найти решение задачи (11) на множестве S_k . в) Провести решение задачи (12) для найденного x^* . В результате будут получены точки $x^1, \dots, x^m$, локальных максимумов ограничений. Если x^* не принадлежит S_k , то решение задачи (9) найдено, в противном случае строится новое множество S_{k+1} . г) , перейти на шаг б). Название «методы обмена» основано на том, что на каждой итерации ряд ограничений A_k , соответствующих точкам $x^1, \dots, x^m$, добавляются, а ряд ограничений для точек из множества удаляются из задачи, то есть происходит обмен ограничений. Следует отметить, что такие методы обычно используются для линейных и выпуклых задач. Подробнее о полунепрерывных методах можно найти в [25, 33]. К классу полунепрерывных методов относятся и неявные методы обмена, в которых и где обмен точек происходит неявно с помощью симплекс-метода. Методы локальной редукции. Методы, в которых бесконечное множество ограничений A , задачи (9) локально заменено в точке конечным множеством ограничений A_k , где $x^1, \dots, x^m$ – точки локальных максимумов относительно A_k , называются методами, основанными на локальной редукции (в [33] их называют непрерывными методами). Алгоритм методов редукции можно представить в виде а) Задаем S_k , номер итерации k . б) Определяем все локальные максимумы для ограничений задачи (9) A_k . Из найденных точек выбираем множество точек $x^1, \dots, x^m$, в которых найдены локальные максимумы ограничений A_k . Если множество S_k не пусто, решение найдено. В противном случае . в) Проводим решение задачи нелинейного программирования (11) на полученном множестве S_k . Получаем решение x^* . г) Увеличиваем счетчик итераций k . Переходим на шаг б). Основная идея методов редукции заключается в том, что если задана допустимая точка для задачи конечной оптимизации, тогда существует окрестность точки x^* , где допустимое множество может быть описано активными в точке ограничениями. Тогда если это (локальное) решение задачи, то оно также является (локальным) решением задачи в которой все ограничения, неактивные в x^* , уничтожены и наоборот. На самом деле, в случае задач полубесконечной оптимизации обычно это не выполняется, т. е. допустимое множество SIP задачи (9) не может быть локально представлено только посредством (обычно конечного множества)

активных ограничений (см. [37]). Тем не менее, при определенных предположениях, для (необязательно допустимой) существует конечное число заданных неявных неравенств и область, где допустимое множество, заданное этими ограничениями, совпадает с допустимым множеством исходной задачи SIP. Поэтому при определенных предположениях задача SIP может быть локально сведена к конечной задаче, по крайней мере, концептуально. Впоследствии такая конечная задача может быть решена методом Ньютона [33], методами последовательного квадратичного программирования (Sequential Quadratic Programming methods) [4], методом модифицированных функций Лагранжа [53] и другими. Подход, основанный на локальной редукции был предложен Хеттишем и Ионгеном [36] и затем был развит другими авторами, например [54]. Методы локальной редукции используются, если известно хорошее приближение к решению задачи в области выполнения условий Куна-Таккера для SIP. Проблемой использования подхода локальной редукции является то, что ни множество локальных максимумов задачи (12), ни допустимая область задачи SIP точно не известны. Поскольку на каждой итерации происходит накопление найденных точек в множестве, то при решении задачи (11) в области выполнения условий Куна-Таккера задачи SIP решение будет быстро найдено, в противном случае подход генерирует бесконечное множество точек. Методы обмена почти исключительно применяются к линейным или выпуклым SIP, но даже и в этих случаях, методы обычно дают низкую точность аппроксимации, эффективность которой повышается, если применять многократный обмен. У них медленная скорость сходимости и это серьезный недостаток, так как на каждом шаге необходимо решать задачу поиска глобального максимума на (фиксировано), что может занимать достаточно много времени. Их медленная сходимость не рекомендует задавать высокую точность. Методы дискретизации численно очень затратны, и время на решение задачи этими методами возрастает на каждой итерации при требовании высокой точности. Время для проверки допустимости точки по отношению к исходной задаче SIP (9) также сильно увеличивается с ростом мощности множества. Поэтому на практике могут использоваться только сетки с ограниченным числом узлов. Типичными являются сетки с 50000 - 100000 точками для задач с 100 переменными. Таким образом, методы дискретизации сильно проигрывают, когда пространство имеет высокую размерность, так как число активных ограничений стремится разрастись. В этом случае эти методы неэффективны. Хотя в общем, методы дискретизации с подходящими стратегиями уточнения сеток превосходят методы обмена вследствие простоты дискретизации и трудности поиска максимума на . Они обладают тем преимуществом, что работают исключительно с конечными подмножествами. Оба типа методов характеризуются тем, что на каждой итерации должны быть вычислены все локальные максимумы ограничений, . С этой стороны

выигрывают методы, основанные на локальной редукции, где глобальный поиск обычно ограничивается каждой третьей - четвертой итерацией. Редукционные методы также имеют преимущества в быстрой сходимости (при высокой точности), и в том, что количество активных ограничений в подзадачах сильно не увеличивается. Возможны и смешанные подходы, которые заключаются в том, что сначала используют метод редукции как основной метод, который дополняют обменом шагов как начальной фазой, и дискретизацией шага в случае, если появляются трудности в уменьшении шага методом редукции [25]. В свою очередь, наоборот, приближенное решение задачи полубесконечной оптимизации, полученное с помощью метода дискретизации, может быть улучшено методом, основанным на локальной редукции, когда первый метод становится неэффективным [37, 23]). Возможная трудность их связи заключается в том, что полученное решение даже может не быть близко к решению задачи SIP и, следовательно, не находится в области сходимости метода, основанного на локальной редукции. Несмотря на указанные недостатки, в основном, решение полученное методом дискретизации удовлетворяет практическим целям. Как и методы дискретизации, полунепрерывные методы могут быть использованы на первой фазе и при некоторых предположениях связываться с методом, основанным на локальной редукции, у которого обычно хорошая сходимость к локальным экстремумам [37]. Такие комбинации называют двухфазовыми или гибридными методами. Двухфазовый метод с техникой дискретизации и методом, основанным на локальной редукции, до сих пор остается наиболее реальным и эффективным подходом к решению малых и небольших задач SIP. Существенной трудностью таких двухфазовых подходов является определение момента перехода от первой ко второй фазе, если, как это часто случается, метод второй фазы сходится только локально. Переключение может произойти слишком рано, так, что полученное приближенное решение не будет находиться в области сходимости второго метода. В этом случае гибридный метод должен вернуться в первую фазу. С другой стороны, возвращение в первую фазу может означать конец всей процедуры, так как продолжение первого метода может быть невыгодно при высоких вычислительных затратах, как в случае с методами дискретизации, или из-за неустойчивости численного метода, что иногда случается в некоторых методах внешних аппроксимаций. Поэтому для получения начальной точки для второй фазы можно вести решение задачи на основе подхода дискретизации с грубо заданной сеткой. В рамках недифференцируемой оптимизации Polak E. в [30] показывает, что большой класс инженерных задач, проектирование конструкций, управление роботами и другие задачи проектирования могут быть сведены к задачам полубесконечной оптимизации. В работе рассматриваются методы спуска, как общеприменимый класс методов для недифференцируемой оптимизации. Следует отметить, что

эти методы могут быть применимы и в случае, когда недифференцируема по . С другой стороны, методы спуска требуют большого числа вычислений функции цели. Вычислительная процедура становится очень затратной, поскольку для решения внутренней задачи (13) необходимо вычисление глобального максимума. Более того, для вычисления достаточно хороших направлений спуска опять может понадобиться использование бесконечного числа ограничений. В версиях алгоритмов, разработанных в [30], область заменяется конечной сеткой с тонким дроблением, гарантирующим сходимость к решению задачи полубесконечной оптимизации. В основном, эти методы очень дороги по затратам времени вычислений. Другие обзоры, посвященные методам спуска, можно найти в [4, 56]. Для невыпуклых задач полубесконечной оптимизации, в основном, существуют два основных вида алгоритмов, на основе которых построено множество других. Первый - из класса возможных направлений представленный В.Ф. Демьяновым [57]. Вторым - это класс внешних аппроксимаций [58, 57, 29, 34], развитый в работах Е.С. Левитина и Б.Т. Поляка [58], Eaves B.C., Zangwill W.I. [26], Blankenship J.W., Falk J.E. [29]. В обоих направлениях мало практических реализаций предложенных алгоритмов: в [34] авторы использовали идеи Демьянова для случая одномерного пространства, в [29] был представлен алгоритм для выпуклого случая. Среди методов, предназначенных для решения выпуклых задач (функции , – непрерывно дифференцируемы и выпуклы по) следует отметить методы «отсекающих плоскостей» (cutting plane methods), которые изначально были предназначены для решения конечных выпуклых задач. Среди них метод Kelley-Cheney-Goldstem [59, 60], который является достаточно надежным методом для решения линейных и выпуклых задач SIP и дает высокую точность решений для задач с числом переменных более тысячи. Метод имеет и ряд недостатков как теоретического, так и практического характера. Наибольший недостаток представляет собой генерация алгоритмом последовательности недопустимых точек. Сходимость к оптимальному решению можно гарантировать, только если допустимая область выпуклая, хотя при помощи метода Келли был решен и ряд невыпуклых задач. Методы «отсекающих плоскостей» чувствительны к точности решения подзадач, поскольку при низкой точности последовательно вводимые отсекающие плоскости в окрестности точки оптимума почти параллельны. Метод Келли и другие методы «отсекающих плоскостей», например, метод Veinott-Elzinga и метод Moore [61] были очень распространены и модифицированы также и для выпуклых задач полубесконечной оптимизации. В [38, 39] Tichatschke R., Schwartz B. представили методы для выпуклых задач полубесконечной оптимизации, которые комбинируют адаптивную схему дискретизации с подходом возможных направлений, используемым для решения задач конечномерной оптимизации. Kaplan A., Tichatschke R. в [40, 41, 42] предлагают несколько вариантов численного подхода к выпуклым

некорректным полубесконечным задачам. Они объединяют техники метода штрафов с дискретизацией и итеративной прох-регуляризацией. Первыми работами, посвященными методам последовательной аппроксимации, были [59, 60], где методы были представлены в форме методов «отсекающих плоскостей». В 1966 году Е.С. Левитин и Б.Т. Поляк [58] представили методы внешних аппроксимаций в более абстрактной форме. Основное отличие предложенных ранее методов сведения бесконечного числа ограничений к конечному состояло в выборе метода поиска критических ограничений. В первых работах, посвященных этим методам, авторы столкнулись с проблемой разрастания множества ограничений. С ростом очень быстро задачи (11) становились такими же сложными, как и исходная задача (9). Поэтому эффективность метода внешних аппроксимаций зависит от того, насколько успешно уничтожаются ограничения. Далее, после некоторого перерыва развитие данного метода получило во многих работах, например, [26, 35], в которых для ограничения скорости роста мощности множеств были предложены схемы отсеивания некоторых ограничений. Так от методов с монотонным ростом мощности множеств ограничений состоялся переход к методам с адаптивным формированием множеств активных ограничений для аппроксимирующих задач. В [27] авторы обобщили разрозненные алгоритмы и систематизировали их, используя для формирования множеств следующие правила: Добавление точек. Для заданной точки последовательное получение точек x^k , осуществляется до тех пор пока не перестанут выполняться необходимые условия оптимальности в точке следующей задачи: (14) $x^k \in X$, (15) $x^k \in X^*$. Отбрасывание точек. Отсеивание части точек из множества X , в которых не нарушаются ограничения $f_j(x) \leq 0$. В [28] для подбора множеств ограничений использовалась схема пассивного случайного поиска. Основным новшеством предложенной схемы была зависимость между значением квазиоптимальной функцией и числом экспериментов случайного поиска активных ограничений. Такая связь ограничивает число экспериментов рандомизации, выполняемых в каждой точке. Вообще, использование только пассивного случайного поиска не очень эффективно, вследствие того, что найденные точки не обязательно будут критическими. В [62] предложено заменить задач максимизации (13) одной задачей максимизации с целевой функцией следующим образом, (16) $\max_{x \in X} f(x)$, где есть некоторый параметр. Авторы доказали, что $f^* = f^*$. Однако, функция может быть невогнутой, даже если функции являются вогнутыми по переменным x . Поэтому задача (16) может стать многоэкстремальной. В связи с этим в работе [4] предложен подход, опирающийся на аппроксимацию ограничений задачи (9) кусочно-линейными по функциями, уточняемыми на каждой итерации метода внешней аппроксимации. В предположении выпуклости ограничений задачи (9) по x , такая постановка задачи дает нижнюю оценку исходной задачи. Для улучшения нижней оценки авторами [4] предложено проводить разбиение исходной области на подобласти,

разбивая на каждой итерации метода те подобласти, на которых аппроксимация ограничений была наихудшей. В то же время, для снижения числа точек в множестве предложено разбивать те подобласти области, в которых решение внутренней задачи (13) дало положительное значение максимума ограничения. В [4] показана эффективность такого подхода, позволяющая получать глобальное решение задачи (9) с небольшими вычислительными затратами. Выводы Задачи полубесконечной оптимизации нашли применение во многих областях, в частности, задачи оптимального проектирования с учетом неопределенности в исходной информации, задачи анализа работоспособности ХТС сводятся во многих случаях к задачам полубесконечного программирования. Обзор численных методов, предлагаемых за последние десятилетия для решения этих задач показал, что для их решения предлагаются множество методов, в основе которых лежит переход от задачи (9), имеющей бесконечное количество ограничений, к задаче (11) с конечным числом ограничений (12). Переход осуществляется за счет дискретизации области с бесконечным числом точек и заменой области конечным множеством, состоящим из набора точек. Основное различие методов решения задач полубесконечного программирования состоит в способе формирования конечного множества. Анализ существующих методов показывает, что наилучшим способом будет использование смешанных подходов последовательной аппроксимации, опирающихся на наследование критических точек, накапливаемых от итерации к итерации метода. Основным недостатком таких подходов является необходимость нахождения глобальных максимумов при решении внутренней задачи (13) на каждой итерации метода, что требует значительных вычислительных затрат. Показано, что в работе [4] предложен подход, основанный на методе внешней аппроксимации [31], дающий глобальное решение задачи (9) и требующий значительно меньших вычислительных затрат, по сравнению с известными методами