

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ЭВОЛЮЦИОННЫХ АЛГОРИТМОВ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ*

Предложена модификация эволюционных методов оптимизации, позволяющая решать задачи оптимизации с дискретными переменными без предварительного преобразования исходной задачи в задачу псевдодвоулевой оптимизации. Исследована эффективность эволюционных методов при решении задачи формирования кредитного портфеля банка.

Ключевые слова: генетический алгоритм, мутация, селекция, дискретная оптимизация.

Задачи оптимизации постоянно возникают в практической деятельности человека. Действительно, если существует возможность выбирать параметры технической или экономической системы, то разумно сделать это самым лучшим (в смысле некоторого критерия) образом. Все более актуальными становятся сложные задачи оптимизации, т. е. задачи, в которых целевая функция не обладает свойствами гладкости, унимодальности, выпуклости, позволяющими эффективно применять классические методы оптимизации. Зачастую для целевой функции не существует даже аналитического выражения.

В качестве примера задач такого типа можно назвать задачу формирования кредитного портфеля банка [1], заключающуюся в формировании кредитного портфеля, максимизирующего прибыль банка, при наличии жестких ограничений по суммам имеющихся в наличии свободных кредитных ресурсов, процентным ставкам на выдаваемые кредиты, срокам привлечения ресурсов и максимальному размеру кредита на одного заемщика.

Пусть N – количество заемщиков; k_j – сумма кредита, запрашиваемая j -м заемщиком; t_j – срок, на который j -й заемщик берет кредит; x_j – булева переменная, принимающая значение: 1, если кредит k_j выдается, и 0, если заявка на получение кредита отклоняется банком; y_j – целочисленная переменная, показывающая, в какой промежуток времени будет выдан j -й кредит; d_j – проценты за пользование j -м кредитом; P_j – вероятность невыполнения заемщиком обязательств по возврату кредита и процентов по нему; T_i – количество дней в i -м промежутке времени (месяце); ρ – ограничение на суммарную рискованность кредитного портфеля. В предлагаемой постановке задачи предполагается два варианта обслуживания долга заемщиком: 100%-й возврат суммы кредита и процентов по нему в установленный срок либо полное отсутствие платежей в погашение кредита и процентов по нему.

Целевая функция прибыли банка после выдачи и погашения всех кредитов будет выглядеть следующим образом:

$$E(x, y) = \sum_{i=1}^T \sum_{j=1}^N \left(\text{Income}(x_j, y_j, i, k_j, t_j, d_j) - \text{Outcome}(x_j, y_j, i, k_j) \right),$$

где T – количество промежутков времени (месяцев), на которых осуществляется планирование; N – количество кредитных заявок; $\text{Income}()$ – сумма, выплачиваемая j -м заемщиком банку в i -й промежуток времени; $\text{Outcome}()$ – сумма, выплачиваемая банком j -му заемщику в i -й промежуток времени.

Переменная Outcome принимает следующие значения: если $x_j = 1$ и $y_j = i$, то нужно вернуть сумму кредита k_j , в противном случае вернуть 0.

Переменная Income вычисляется следующим образом.

Шаг 1. Если $x_j = 0$, то $\text{Income} = 0$. Выйти из процедуры.

Шаг 2. Если $y_j > i$, то $\text{Income} = 0$. Выйти из процедуры.

Шаг 3. Вычислить количество оставшихся дней до истечения срока кредита по формуле

$$\text{rest} = t_j - \sum_{s=y_j}^{i-1} T_s.$$

Шаг 4. Если $\text{Rest} < 0$, то $\text{Income} = 0$. Выйти из процедуры.

Шаг 5. Вычислить количество дней в текущем i -м промежутке времени для уплаты кредита по формуле

$$\text{Period} = \min\{T_i, \text{Rest}\}.$$

Шаг 6. Вычислить количество денежных средств, которые выплачивает j -й заемщик банку в текущий i -й промежуток времени:

$$\text{Income} = k_j \cdot 0,01 \cdot d_j \cdot \frac{\text{period}}{365} + k_j \cdot \frac{\text{period}}{d_j}.$$

Средняя рискованность кредитного портфеля должна быть меньше некоторой заданной величины:

$$R(x) = \frac{\sum_{j=1}^N x_j \cdot P_j}{\sum_{j=1}^N x_j} \leq \rho.$$

*Работа выполнена при финансовой поддержке ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» (НИР 2011-1.9-519-005-042) и ФЦП «Научные и научно-педагогические кадры России» (НИР 2011-1.2.1-113-025, 2011-1.2.2-215-021).

Еще одним ограничением является неотрицательность наличных средств у банка в каждый промежуток времени. Пусть B_0 – собственные средства банка в начале срока, на который ведется планирование. Тогда собственные средства банка в i -й период можно определить по формуле

$$B_i(x, y) = B_{i-1} + \sum_{j=1}^N (I(x_j, y_j, s, k_j, t_j, d_j) - O(x_j, y_j, s, k_j))$$

и должно иметь место следующее неравенство:

$$\min_{i=1..T} \{B_i\} \geq 0.$$

Для решения сложных задач оптимизации, как правило, применяются методы прямого поиска. Одним из важнейших классов этой группы методов являются генетические алгоритмы (ГА) [2], имитирующие эволюционные и генетические процессы, происходящие в популяциях живых организмов. Генетический алгоритм работает на каждой итерации не с единственным решением, а с целым коллективом решений, или популяцией (в терминологии ГА).

Среди решений текущей популяции производится отбор (селекция) перспективных особей в промежуточную популяцию, которая затем используется для порождения особей новой популяции. Существует несколько основных методов селекции: пропорциональная, ранговая и турнирная. При пропорциональной селекции вероятность прохождения решением отбора пропорциональна значению целевой функции для этого решения, а поскольку вероятность не может быть отрицательной величиной, то данный метод требует, чтобы значения целевой функции были преобразованы в неотрицательные величины. При применении ранговой селекции вероятности прохождения отбора зависят не от значений целевой функции, а от их ранга в популяции. Метод турнирной селекции заключается в том, что турнирная группа формируется случайным образом и лучшее в смысле значения целевой функции решение становится победителем турнира и проходит отбор.

На основе отобранных в ходе этапа селекции решений с помощью генетических операторов скрещивания и мутации формируются новые решения.

В ходе скрещивания два родительских решения используются для формирования нового решения. Основными методами скрещивания являются одноточечное, двухточечное и равномерное скрещивание. При одноточечном скрещивании векторы родительских решений делятся случайным образом на две части, порождаемое решение получает первую часть от первого родителя, вторую – от второго. В процедуре двухточечного скрещивания векторы родительских решений разделяются на три части, а порождаемое решение получает начальную и конечную часть от первого решения, а среднюю – от второго. Равномерное скрещивание заключается в том, что каждый ген порождаемого решения переходит от случайно выбранного родительского решения.

В ходе мутации компоненты решения с небольшой вероятностью, которую называют *вероятностью мутации*, заменяются на противоположные. Основное предназначение оператора мутации – предотвращение преждевременной сходимости алгоритма.

На псевдокоде генетический алгоритм можно описать следующим образом.

Шаг 1. Создать и оценить начальную популяцию с равномерным распределением генов.

Шаг 2. Если выполнен критерий останова, то прекратить работу.

Шаг 3. Сформировать промежуточную популяцию с помощью выбранного метода селекции.

Шаг 4. Создать новую популяцию на основе промежуточной с использованием выбранного оператора скрещивания и мутации.

Шаг 5. Перейти к шагу 2.

Вероятностный генетический алгоритм (ВГА) – это попытка создать процедуру оптимизации, имеющую схему, похожую на схему стандартного генетического алгоритма [3]. В вероятностном генетическом алгоритме явным образом (в отличие от традиционного ГА) вычисляются компоненты вектора вероятностей и отсутствует оператор скрещивания (вместо него используется оператор порождения случайного решения с заданным распределением вероятностей компонент), но сохранены генетические операторы мутации и селекции.

На псевдокоде ВГА можно представить следующим образом.

Шаг 1. Создать и оценить начальную популяцию с равномерным распределением генов.

Шаг 2. Если выполнен критерий останова, то прекратить работу.

Шаг 3. Сформировать промежуточную популяцию с помощью выбранного метода селекции.

Шаг 4. Оценить распределение вероятностей значений генов в промежуточной популяции.

Шаг 5. Создать популяцию потомков с полученным распределением вероятностей.

Шаг 6. Применить к полученным решениям оператор мутации.

Шаг 7. Создать новую популяцию на основе популяции потомков.

Шаг 8. Перейти к шагу 2.

Дальнейшим развитием идей ВГА стал асимптотический вероятностный генетический алгоритм (АВГА) [4], название которого связано с тем, что его операторы являются предельным случаем операторов вероятностного генетического алгоритма (при стремлении размера промежуточной популяции к бесконечности). В частности, процедура мутации, применяемая в АВГА, заключается в корректировке распределения (асимптотической мутации), а распределение значений генов для новой популяции вычисляется непосредственно по текущей популяции с учетом вероятностей прохождения селекции (асимптотической селекции) без статистического моделирования селекции по методу Монте-Карло. Работа АВГА осуществляется следующим образом.

Шаг 1. Создать и оценить начальную популяцию с равномерным распределением генов.

Шаг 2. Если выполнен критерий останова, то прекратить работу.

Шаг 3. Построить распределение с помощью оператора асимптотической селекции.

Шаг 4. Скорректировать распределение с помощью оператора асимптотической мутации.

Шаг 5. Создать новую популяцию с полученным распределением вероятностей. Оценить новую популяцию.

Шаг 6. Перейти к шагу 2.

Описанные выше алгоритмы являются методами псевдобулевой оптимизации, т. е. оптимизации функций с булевыми переменными и вещественными значениями. Однако ГА (или любой другой алгоритм псевдобулевой оптимизации) можно использовать для оптимизации функций дискретных и/или вещественных переменных, если совместить его с методом бинаризации, основанном на том, что дискретные и вещественные переменные кодируются (с некоторой погрешностью в случае вещественных переменных) с помощью стандартного бинарного кода или грей-кода [5], а генетические операторы мутации и скрещивания действуют на бинарное представление решений. Таким образом, метод бинаризации позволяет решать задачи оптимизации со смешанными переменными с помощью методов псевдобулевой оптимизации.

Двоичная кодировка обладает еще одним преимуществом: бинарное представление решения имеет наибольшую возможную длину среди всех избыточных представлений данного числа, а следовательно, наибольшую возможную мощность точек, отличающихся от заданной точки одним разрядом (однососедних точек). Если использовать для кодирования грей-код, то однососедние точки естественного представления переменных перейдут в однососедние точки двоичного кода, поэтому новые локальные экстремумы у функции с двоичным представлением переменных не появятся. С другой стороны, поскольку мощность системы окрестностей будет увеличена, то некоторые локальные экстремумы могут исчезнуть.

Эволюционные алгоритмы дискретной оптимизации. Если дискретная переменная может принимать N различных значений, то для хранения ее двоичного представления потребуется $n = \lceil \log_2(N) \rceil$ двоичных разрядов. И если при использовании метода бинаризации количество возможных значений дискретной переменной не является целой степенью двойки, то некоторым дискретным значениям будут соответствовать две различные бинарные строки. Эта особенность метода бинаризации приводит к следующим проблемам:

– так как некоторым дискретным значениям будет соответствовать одна бинарная строка, а некоторым – две, то отдельные дискретные значения при случай-

ном порождении будут появляться чаще, чем другие. Подобные неоднородности могут уменьшить надежность алгоритма в тех случаях, когда у оптимума будет одно бинарное представление, а у большей части малоперспективных точек – два;

– для хранения бинарного решения будет требоваться больше памяти, чем это теоретически необходимо. Конечно, избыток в 1 бит может показаться незначительным, но в современных ЭВМ бит не является адресуемой единицей, поэтому если не использовать побитовые операции, то дополнительные затраты памяти составят не менее байта. Если же служебная информация хранится для каждого гена (как, например, в вероятностном генетическом алгоритме), то дополнительных затрат памяти избежать не удастся даже с использованием манипуляций с отдельными битами, которые также увеличивают размер кода программы [6];

– большее количество генов увеличивает не только затраты памяти, но и время, затрачиваемое на работу алгоритма, так как генетические операторы мутации (при прямолинейной реализации этих операторов) и скрещивания требуют обхода всех битов бинарного представления, т. е. их сложность пропорциональна количеству генов. Кроме того, в современных вычислительных машинах увеличение затрат памяти само по себе может стать причиной уменьшения быстродействия.

Таким образом, для решения задач дискретной оптимизации перспективной является идея разработки генетического алгоритма без предварительной бинаризации. Рассмотрим, каким образом для этого нужно модифицировать основные этапы эволюционных алгоритмов: порождение решений с заданным распределением, селекцию, скрещивание и мутацию.

Селекция не требует каких-либо модификаций, так как на этом этапе роль играют только значения целевой функции (или их ранги), а не представления решений. Процедуры случайного порождения решений, скрещивания и мутации, напротив, действуют на конкретное представление решения, поэтому их необходимо модифицировать. При проектировании этих операторов следует учитывать, что дискретные переменные могут быть измерены в различных шкалах: номинальной или ранговой [7].

В данной статье мы ограничимся случаем номинальных переменных, так как любую шкалу можно рассматривать как номинальную (с соответствующим округлением результата). Кроме того, в случае очень большого количества возможных значений различия между номинальной и ранговой шкалами будут несущественными.

Оператор мутации для номинальных переменных разумно определить следующим образом: с некоторой вероятностью (вероятностью мутации) переменной присваивается случайно выбранное значение, причем все значения выбираются с одной и той же вероятностью, так как номинальная шкала не позволяет судить

о степени близости двух значений. Оператор мутации должен вносить небольшие изменения в решения, поэтому вероятность мутации, так же как и в случае булевых переменных, следует выбирать достаточно малой.

В случае номинальных дискретных переменных можно определить аналоги всех трех основных операторов скрещивания стандартного генетического алгоритма: так как значения номинальных переменных можно только проверять на равенство, то каждый компонент решения должен переходить от одного из родительских решений.

Предложенный подход можно распространить и на вероятностный генетический алгоритм. Оценка распределения и генерация решений в соответствии с построенным распределением является довольно прямолинейной задачей, а операторы селекции и мутации ВГА не отличаются от соответствующих операторов ГА.

Исследование исходных алгоритмов и их модификаций на тестовых функциях, взятых из [3], показало преимущество эволюционных алгоритмов, использующих небинарное кодирование. Кроме небольшого увеличения средней надежности это преимущество выражалось в более точном представлении решения: в некоторых случаях этими алгоритмами было получено точное (без погрешности) решение, в то время как бинарные эволюционные алгоритмы останавливались в точках, достаточно близких к истинному оптимуму, но не совпадающих с ним.

Численные данные для апробации на практической задаче формирования кредитного портфеля банка были взяты из [1]. В качестве альтернативы ГА и ВГА был рассмотрен мультистарт локального поиска: алгоритм локального поиска дискретной оптимизации запускался заданное число раз из случайно выбранной точки. Численные эксперименты производились при следующих настройках: размер популяции – 100; число итераций – 100; число запусков для усреднения

результатов – 100; селекция – турнирная (размер турнирной группы – 2); мутация – слабая; скрещивание (в генетическом алгоритме) – равномерное. В алгоритмах, которым необходима бинаризация переменных задачи, использовался грей-код. Для учета ограничений применялся следующий метод: если доля допустимых решений в популяции была меньше определенного порога, то выполнялась минимизация по степени нарушения ограничений; если же допустимых решений было достаточно, то в качестве значения пригодности использовалось значение целевой функции, а недопустимые решения не переходили в новую популяцию.

Результаты решения задач, полученные различными методами дискретной оптимизации, приведены в таблице. Для ГА, ВГА и АВГА в скобках указан способ кодирования: двоичные гены – стандартный генетический алгоритм, троичные и десятичные гены – предложенные модификации, использующие для кодирования целочисленных переменных соответственно троичную и десятичную систему счисления.

Задача формирования кредитного портфеля банка также была решена с помощью методов PBIL (Population Based Incremental Learning – последовательное популяционное обучение) [8] и PSO (Particle Swarm Optimization – оптимизация роем частиц) [9].

Анализ данных таблицы показывает, что и генетический алгоритм, и вероятностный генетический алгоритм дискретной оптимизации по эффективности решения задачи и быстродействию превосходят свои аналоги, использующие бинаризацию. Кроме того, оба эти метода существенно эффективнее мультистарта локального поиска и метода PSO. Наилучшие результаты показал АВГА, использующий троичное кодирование. Полученные результаты позволяют сделать вывод о том, что предложенный подход к решению задач дискретной оптимизации без предварительной бинаризации объектных переменных является состоятельным.

Результаты решения задачи формирования кредитного портфеля банка

Алгоритм	Среднее значение	Выборочное среднеквадратическое отклонение значения целевой функции	Время работы, с
Локальный поиск	5 855	206,19	0,20
ГА (двоичные гены)	6 285	27,47	7,18
ГА (троичные гены)	6 290	21,78	6,85
ГА (десятичные гены)	6 287	25,46	7,16
ВГА (двоичные гены)	6 306	19,34	7,19
ВГА (троичные гены)	6 308	20,15	7,60
ВГА (десятичные гены)	6 307	22,39	8,10
АВГА (двоичные гены)	6 316	18,13	7,60
АВГА (троичные гены)	6 318	18,70	7,38
АВГА (десятичные гены)	6 318	19,81	7,81
PBIL	6 309	16,97	7,33
PSO (дискретный)	6 267	21,27	7,85

Библиографические ссылки

1. Пуртиков В. А. Оптимизация управления формированием кредитного портфеля банка : дис. ... канд. техн. наук. Красноярск, 2001.
2. Goldberg D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Mass. : Addison-Wesley, 1989.
3. Семенкин Е. С., Сопов Е. А. Вероятностные эволюционные алгоритмы оптимизации сложных систем // Тр. Междунар. науч.-техн. конф. «Интеллектуальные системы» и «Интеллектуальные САПР». В 3 т. М. : Физматлит, 2005. Т. 1.
4. Галушин П. В., Семенкин Е. С. Асимптотический вероятностный генетический алгоритм // Вестник СибГАУ. 2009. Вып. 4 (25). С. 37–42.
5. Кнут Д. Э. Искусство программирования. Т. 4, вып. 2. Генерация всех кортежей и перестановок. М. : Вильямс, 2008.
6. Страуструп Б. Язык программирования C++. Спец. изд. М. : Бином-пресс, 2007.
7. Перегудов Ф. И., Тарасенко Ф. П. Основы системного анализа : учебник. 2-е изд., доп. Томск : НТЛ, 1997.
8. Baluja S. Population-Based Incremental Learning: a Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Pittsburgh, Pa: Carnegie Mellon Univ., 1994.
9. Kennedy J., Eberhart R. Particle Swarm Optimization // Proc. of IEEE Intern. Conf. on Neural Networks. IV. Perth, Australia, 1995. P. 1942–1948.

P. V. Galushin, O. E. Semenina

DEVELOPMENT AND EVALUATION OF EVOLUTIONARY ALGORITHMS FOR DISCRETE OPTIMIZATION

This paper proposes a modification of evolutionary optimization algorithms, which can be used to solve optimization problems with discrete variables without transformation of original problem into pseudo-boolean problem. It also evaluates efficiency of proposed methods applied for solution of problem of bank credit portfolio building.

Keywords: genetic algorithm, mutation, selection, discrete optimization.

© Галушин П. В., Семенкина О. Э., 2011

УДК 519.8

Д. А. Дегтерев, А. С. Дегтерев

МАТЕМАТИЧЕСКИЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ СОСТАВЛЕНИЯ ЦЕХОВЫХ РАСПИСАНИЙ

Проведен анализ современных систем, математических моделей и методов планирования и управления производством. Рассмотрены критерии оптимизации при составлении производственных расписаний, оценена вычислительная сложность задач оптимизации расписаний. Предложена модель задачи многокритериальной оптимизации цеховых расписаний дискретного производства, учитывающая все технологические ограничения.

Ключевые слова: математическое, динамическое и линейное программирование, имитационное моделирование.

Базовым понятием для задач составления расписаний (Machine Scheduling, MS) является операция. Для задания операции, как правило, необходимы два агента: объект операции, или ее материальный носитель, – то, над чем производят операцию, и субъект операции, или исполнитель, – тот, кто (или что) выполняет операцию. В качестве материального носителя операции может выступать отдельная деталь или сборочная единица создаваемого изделия. Носителем операции может быть человек, обслуживаемый другими людьми (приборами). Субъектами операций являются станки (инструменты, люди), необходимые для выполнения этих операций. В моделях MS они обычно называются машинами.

Некоторые подмножества операций организационно (и технологически) объединяются в работы. Например, если завод выпускает сложные изделия (машины, самолеты и т. п.), то все операции, относящиеся к одному изделию, составляют одну работу. На некоторые работы может быть наложен жесткий директивный срок их окончания, которому соответствует английский термин Deadline (смертельная линия), и/или желаемый срок окончания Due Date, отклонение от которого наказывается штрафом; при этом Deadline и Due Date одной и той же работы могут не совпадать. Моменты $\{C_j\}$ окончания работ $\{J_j\}$ (C – от Completion Time) играют существенную