**Интеллект композитных алгоритмов поиска глобально оптимальных решений экстремальных задач с булевыми переменными**

**В. О. Гроппен**

groppen@mail.ru

1. **Введение**

Рост числа различных приложений систем искусственного интеллекта [1 - 3] повышает требования к используемым в рамках этой технологии оптимизационным алгоритмам. Это особенно важно применительно к задачам дискретной оптимизации, для которых отсутствуют эффективные процедуры, гарантирующие в общем случае глобально оптимальное решение [4, 8]. К классическим алгоритмам поиска глобально оптимальных решений экстремальных задач дискретной оптимизации можно отнести методы, развитые во второй половине прошлого века: полный перебор, методы типа ветвей и границ и динамическое программирование [5-8 ]. До последнего времени повышение их интеллекта было связано с распараллеливанием вычислений и совершенствованием способов вычисления оценок, адаптируя их применительно к специфике решаемых задач [3, 9, 10]. Ниже анализируется интеллект композитных алгоритмов, ориентированных на решение задач дискретной оптимизации с булевыми переменными. По аналогии с композитными материалами [12-15], под композитными алгоритмами ниже понимаются процедуры, в которых присутствуют компоненты не менее двух классических алгоритмов. Так, ниже процедура выбора направления спуска по дереву ветвлений в методах типа ветвей и границ дополняется технологией отсечения «плохих» планов, развитой в динамическом программировании, а процедуры отсечения, используемые в динамическом программировании, расширяются за счет применения оценок, присущих методам типа ветвей и границ.

Существуют различные способы сравнительной оценки «интеллекта» алгоритмов, используемых для принятия решений с помощью математических моделей [16, 17]. Применительно к процедурам, гарантирующим глобально оптимальные решения, мерой их интеллекта может служить время поиска решения: если алгоритм А находит решение i-й задачи за время t(A, i), а алгоритм В находит решение той же задачи за время t(B, i), причем справедливо:

 (1) то интеллект алгоритма В выше, чем А.

Далее исследуется сравнительный интеллект классических и композитных процедур поиска глобально оптимальны решений экстремальных задач с булевыми переменными, причем используются следующие обозначения, допущения и определения.

1. **Обозначения, допущения и определения**

Ниже анализируется сравнительный интеллект различных алгоритмов поиска решения однокритериальных задач с булевыми переменными вида:



где:   Сi и *bi,j* – коэффициенты, *aj*- константы, причем полагаем, что  Сi ≥0. При этом все численные примеры, иллюстрирующие эффективность предлагаемых подходов, ниже приводятся для случая, когда F → max, j = 1, что отвечает задаче о ранце вида [3]:

F = 7z1 +3z2 + 5z3 + 9z4 + 2z5 → max;

3z1 + 4z2 + 2z3 + 7z4 + 6z5 ≤ 10; (3)

zi = 1, 0; *i* = 1, 2, …,5.

Далее, применительно к методам неявного перебора, осуществляющим поиск решения на множестве частичных планов, используются следующие определения:

1. «Висячей» вершиной на любой итерации считается такая вершина построенной части дерева ветвлений, из которой не исходят дуги.
2. Корневая вершина дерева ветвлений считается «висячей» вершиной построенной части дерева на первой итерации.
3. «Кустом» с высотой h и с корнем в вершине  является ориентированный подграф, обладающий следующими свойствами:

* в каждую вершину куста, за исключением корневой, входит только одна дуга;
* из каждой вершины куста, за исключением множества висячих вершин, исходят две дуги;
* из вершины  в каждую висячую вершину куста существует только один путь, число дуг которого равно h;
* число вершин куста равно 2h+1-1, число висячих вершин равно 2h, число дуг куста определяется выражением: 2(2h - 1);
* все рассмотренные ниже примеры в разделах 3 – 6 используют случаи, когда h = 1, только в примере, рассмотренном в седьмом разделе, h = 2.

1. Ветвлением из заданной вершины  которой соответствует подмножество введенных в базис переменных I(xk), является построение «куста» с корнем в этой вершине, причем каждой из 2h висячих вершин этого куста соответствует │I(xk)│+ h введенных в базис переменных.
2. Верхняя оценка Δ(xk) величины F, отвечающей вершине  соответствующей вектору переменных задачи (3), в базис которого введены I1(xk) переменных, далее определяетсяследующим образом:

Δ(xk) =  (4)

где I – множество индексов всех переменных.

1. Аналогично определяется нижняя оценка δ(I1) величины F:

δ(xk) =  (5)

В то время как нижняя граница, определяемая на основании (5) в силу сделанных выше допущений не зависит от специфики задачи, величина Δ(xk) непосредственно связана с этой спецификой: выражение (4) справедливо только для задачи о ранце. Примеры способов вычисления верхней оценки Δ(xk) применительно к ряду других задач, приведены в [11]. Далее полагаем, что время счета в первом приближении линейно зависит от числа вершин построенного в ходе поиска решения дерева ветвлений G(X,U), где Х – множество вершин, U – множество дуг.

Общей чертой приводимых ниже графических иллюстраций поиска решения системы (3) является кодировка вершин дерева ветвлений: «серые» вершины дерева отвечают значениям соответствующих переменных, равным единице, «белые» – равным нулю. Вершине последнего яруса с «жирным» контуром соответствует оптимальный вектор переменных Z.

1. **Методы неявного перебора на частичных планах, гарантирующие глобально оптимальное решение**

Ниже рассмотрено два метода такого рода: динамическое программирование и методы типа ветвей и границ.

* 1. Метод типа ветвей и границ, осуществляющий фронтальный спуск по дереву ветвлений

Содержательно работа такого алгоритма может быть представлена последовательным построением дерева ветвлений G(X,U):

Алгоритм 1

1. На множестве висячих вершинпостроенной части дерева ветвлений G(X,U) выбирается вершина xj с наилучшей оценкой. Если это осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева.
2. Если выбранной вершине отвечает равенство I1 = I, то перейти к шагу 5, в противном случае – к следующему шагу.
3. Ветвление осуществляется из выбранной на шаге 1 последней итерации вершины xj. Новое множество висячих вершин дерева вновь обозначаем Х1.
4. Вычисляются оценки, отвечающие висячим вершинам построенного на предыдущем шаге «куста». Для этого можно воспользоваться (4), если целевая функция системы (2) является максимизируемой и (5) – если целевая функция минимизируется. Перейти к шагу 1.
5. Алгоритм закончен. Вектор переменных, соответствующий выбранной вершине, является оптимальным.

|  |
| --- |
| 26 S 19 x1  26 23 19 16 x2   1. 21 23 18 19 14 x3   -∞ 17 - ∞ 12 -∞ 14 18 9 - ∞ 10  -∞ 15 - ∞ 16      z1  z2  z3  z4  z5 |
| Рис. 1. Дерево ветвлений G1(X1,U1), построенное алгоритмом 1 применительно к задаче (3). |

Оптимальный вектор переменных задачи (3), Z= {1,0,01,0}, соответствующее значение целевой функции F=16, число вершин дерева G1(X1, U1), │X1│=27.

* 1. Динамическое программирование

Содержательно решение задачи дискретной оптимизации с помощью динамического программирования может быть проиллюстрировано последовательным, ярус за ярусом, построением дерева ветвлений G2(X2,U2), стратегия которого существенно отличается от стратегии, описанной в приведенном выше алгоритме 1. Ниже приводится агрегированное описание процедуры такого рода при условии, что все коэффициенты системы (2) неотрицательны.

Алгоритм 2

1. На множестве висячих и не помеченных вершин построенной части дерева ветвлений G2(X2,U2) выбирается вершина xj. Если таковой вершины нет, то перейти к шагу 4. Если выбор осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева. 2. Осуществляется ветвление из выбранной на шаге 1 последней итерации вершины xj и вычисляются компоненты каждого вектора R(xk), отвечающего всем висячим вершинам построенного «куста» (величина h=1). Для определения первой компоненты этого вектора используется (5), остальные компоненты вычисляются по формуле:

 rj(xk) =  (6)

где xk – вершина, принадлежащая множеству висячих вершин куста, в которую заходит дуга из выбранной на шаге 1 последней итерации вершины xj. 3. Все вершины куста, построенного на шаге 2 последней итерации, помечаются, перейти к шагу 1. 4. Убираются все пометки подмножества висячих вершин дерева ветвлений. 5. Если на множестве висячих вершин существует вершина xk, для которой справедливо хотя бы одно из условий: 

а) I1(xk) = I1(xq);

б) δ(xk) «хуже» чем δ(xq);

в) rj(xq) ≥ rj(xk),

то эта вершина вычеркивается.

6. Если число введенных в базис переменных висячих вершин дерева ветвлений равно числу переменных решаемой задачи, то перейти к следующему шагу, в противном случае перейти к шагу 1. 7. На множестве не вычеркнутых висячих вершин выбирается вершина xk с наилучшей первой компонентой соответствующего ей вектора R(xk). 8. Алгоритм закончен. Вектор переменных Z, соответствующий выбранной на шаге 7 вершине xk, является оптимальным.

Ниже на рис. 2 изображено дерево G2(X2, U2), построенное с помощью алгоритма 2 применительно к задаче (3), векторы у вершин определены на шагах 3 каждой итерации.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S  7,7 0,10  7,7 10,3 3,6 0,10  12,5 7,7 15,1 10,3 5,8 0,10  12,5 16,0 7,7 15,1 5,8 0,10  12,5 16,0 9,1 7,7 15,1 7,2 5,8 2,4 0,10   |  | | --- | | i | | 1 | | 2 | | 3  4 | | 5 | |  | |
| Рис. 2. Дерево ветвлений G2(X2,U2), построенное c помощью алгоритма 2 применительно к задаче (3). |

На рис. 2 перечеркнуты вершины, вычеркнутые на шаге 6 алгоритма 2, и отсутствуют векторы перечеркнутых вершин, для которых справедливо условие в) шага 6, причем число вершин построенного дерева G2(X2, U2), │X2│=35.

1. **Композитные алгоритмы, реализующие неявный перебор на частичных планах, и гарантирующие глобально оптимальное решение**

Ниже приводятся модификации методов типа ветвей и границ, методика отсечения в которых включает технологии, используемые в динамическом программировании, а также модификации алгоритма, реализующего динамическое программирование с привлечением оценок, используемых методами типа ветвей и границ.

* 1. Композитная реализация метода типа ветвей и границ

В рамках реализации этого подхода каждой вершине xj строящегося дерева ветвлений G3(X3,U3) ставится в соответствие вектор V(xj), первой компонентой которого является оценка, используемая в традиционной реализации этого алгоритма, а остальные компоненты совпадают с компонентами вектора R(xj), описание которого приводится в предыдущем разделе, посвященном динамическому программированию. Ниже приводится содержательное описание такой процедуры.

Алгоритм 3

1. На множестве не вычеркнутых висячих вершинпостроенной части дерева ветвлений G(X,U) выбирается вершина xj с «наилучшей» первой компонентой вектора V(xj). Если это осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева.

2. Если выбранной вершине отвечает равенство I1 = I, то перейти к шагу 8, в противном случае – к следующему шагу.

3. Ветвление осуществляется из выбранной на шаге 1 последней итерации вершины xj. Новое множество висячих вершин дерева вновь обозначаем 

4. Для каждой висячей вершины построенного на предыдущем шаге «куста» xj формируется вектор V(xj). Для формирования первой компоненты этого вектора используется процедура, применявшаяся для вычисления соответствующей оценки в методе типа ветвей и границ, для определения второй компоненты вектора V(xj) используется (5), остальные компоненты вычисляются по формуле (6).

1. Если на множестве висячих вершин существует вершина xk, для которой справедливо хотя бы одно из условий: 

а) I1(xk) = I1(xq);

б) δ(xk) «хуже» чем δ(xq);

в) rj(xq) ≥ rj(xk),

то эта вершина вычеркивается.

1. Если существуют вершины и , для которых справедливо:

а) I1(xj) = I1(xq);

б) δ(xj) «лучше» чем Δ(xq),

то вершина xq вычеркивается.

1. Перейти к шагу 1.

8. Алгоритм закончен. Вектор переменных, соответствующий выбранной вершине, является оптимальным.

Ниже на рис. 3 изображено дерево G3(X3, U3), построенное с помощью алгоритма 3 применительно к задаче (3). Число вершин построенного дерева ветвлений G3(X3, U3), │X3│=23, причем однократно перечеркнуты вершины, отвечающие применению шага 5, а дважды перечеркнутые вершины отображают реализацию шага 6.

В соответствии с принятыми выше допущениями применительно к задаче (3) относительное улучшение интеллекта метода типа ветвей и границ, осуществляющего фронтальный спуск по дереву ветвлений, в результате привлечения технологий отсечения, присущих динамическому программированию, определяется выражением:

 (7)

|  |
| --- |
| z1  z2  z3  z4  z5  26,7,7 S 19,0,10    26,10,3 23,7,7 19,3,7 14,3,6 16,0,10  26,15,1 21,10,3 23,12,5 18,7,7 19,8,4 14,3,6  -∞ 17,15,1 - ∞ 14,12,5 18,16,0 9,7,7  -∞ 15,15,1 -∞ 16,16,0 |
| Рис. 3. Дерево ветвлений G3(X3,U3), построенное алгоритмом 3 применительно к задаче (3). |

* 1. Композитная реализация динамического программирования

Предлагаемый далее подход отличается от приведенного выше алгоритма 2 следующими параметрами:

1. Каждой вершине xj дерева ветвлений ставится в соответствие вектор V(xj), аналогичный тому, который использовался в алгоритме 3.
2. Если на i-й итерации i-у ярусу построенной части дерева ветвлений принадлежат две такие вершины xj и xq , у которых вторая компонента вектора V(xj) «лучше» первой компоненты вектора V(xq), то вершина xq вычеркивается. Практически эта операция совпадает с шагом 6 алгоритма 3.

Ниже приводится содержательное описание композитной реализации динамического программирования, включающей алгоритм 2 и приведенные выше параметры.

Алгоритм 4

1. На множестве висячих, не вычеркнутых и не помеченных вершин построенной части дерева ветвлений G4(X4,U4) выбирается вершина, которую обозначаем, как xj. Если таковой вершины нет, то перейти к шагу 4. Если выбор осуществляется на первой итерации, то такой вершиной *a priori* считается корневая вершина дерева G4(X4,U4).
2. Осуществляется ветвление из выбранной на шаге 1 последней итерации вершины xj и вычисляются компоненты каждого вектора V(xk), отвечающего всем висячим вершинам построенного «куста». Процедура вычисления компонент этого вектора совпадает с процедурой, описанной на шаге 4 алгоритма 3.
3. Пометить все полученные на шаге 2 последней итерации висячие вершины куста и перейти к шагу 1.
4. Убрать пометки всех висячих вершин построенной части дерева ветвлений G4(X4,U4).
5. Если на множестве висячих вершин дерева G4(X4,U4) существует вершина xk, для которой справедливо хотя бы одно из условий: 

а) I1(xk) = I1(xq);

б) δ(xk) «хуже» чем δ(xq);

в) rj(xq) ≥ rj(xk),

то вершина xk вычеркивается.

1. Если на множестве висячих вершин построенной части дерева G4(X4,U4) существуют две такие вершины xj и xq , у которых вторая компонента вектора V(xj) «лучше» первой компоненты вектора V(xq), то вершина xq вычеркивается.
2. На множестве не вычеркнутых висячих вершин выбирается вершина xk с наилучшей первой компонентой соответствующего ей вектора V(xk).
3. Если выбранной вершине отвечает число переменных, введенных в базис, равное числу переменных решаемой задачи, то перейти к шагу 9, в противном случае перейти к шагу 1.
4. Алгоритм закончен. Вектор переменных Z, соответствующий выбранной на шаге 9 вершине xk, является оптимальным.

Ниже на рис. 4 изображено дерево G4(X4, U4), построенное с помощью алгоритма 4 применительно к задаче (3), векторы у вершин определены на шагах 3 каждой итерации. В соответствии с принятыми выше допущениями применительно к задаче (3) относительное улучшение интеллекта динамического программирования, в результате привлечения технологий отсечения, присущих методам типа ветвей и границ, определяется выражением:

 (8)

где Х2 – множество вершин дерева G2(X2, U2) (см. выше рис. 2), Х4 - множество вершин дерева G4(X4, U4) (см. ниже рис.4).

|  |
| --- |
| S  19,0,10 26,7,7  16,0,10 19,3,6 23,7,7 26,10,3  11,0,10 16,5,8 18,7,7 23,12,5 21,10,3 26,15,1  7,5,8 16,14,1 9,7,7 18,15,0 14,12,5 -∞ 17,15,1 -∞    16,16,0 - ∞ 15,15,1 - ∞  Z1  Z2  Z3  Z4  Z5 |
| Рис. 4. Дерево ветвлений, иллюстрирующее поиск решения задачи (3) алгоритмом 4. |

Знаком « - ∞» помечены вершины, отвечающие векторам переменных, нарушающим ограничения системы (3), однократно перечеркнуты вершины, отвечающие применению шага 6, а дважды перечеркнутые вершины отображают реализацию шага 7.

В соответствии с принятыми выше допущениями применительно к задаче (3) относительное улучшение интеллекта динамического программирования, в результате привлечения технологий отсечения, присущих методам типа ветвей и границ, определяется выражением:

 (8)

1. **Поиск глобально оптимального решения на полных планах**

Поиск глобально оптимальных решений экстремальных задач с булевыми переменными на полных планах связан с последовательным анализом 2n их вариантов, где n – число переменных. Сокращение объема перебора может быть достигнуто разбиением всего множества полных планов на 2h подмножеств (h<n), каждое из которых анализируется на предмет отсечения с помощью рассмотренных в предыдущем разделе методов. Графически это можно интерпретировать построением дерева, содержащего два яруса вершин, причем первому ярусу соответствует только корневая вершина, а второму – 2h вершин, каждая из которых отвечает «своему» подмножеству полных планов с совпадающими значениями первых h переменных (см. рис. 5 ниже). При этом каждой вершине второго яруса xj, jϵ{0÷(2h-1)}, ставится в соответствие вектор V(j)={v1,j, v2,j, v3,j …} формирование которого описано в шаге 3 приводимого ниже алгоритма 5.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S  0 1 2 3  h=2     |  |  |  |  | | --- | --- | --- | --- | | 0 0 | 0 | 0 | 0 | | 0 | 0 | 1 | | 0 | 1 | 0 | | 0 | 1 | 1 | | 1 | 0 | 0 | | 1 | 0 | 1 | | 1 | 1 | 0 | | 1 | 1 | 1 |  |  |  |  |  | | --- | --- | --- | --- | | 0 1 | 0 | 0 | 0 | | 0 | 0 | 1 | | 0 | 1 | 0 | | 0 | 1 | 1 | | 1 | 0 | 0 | | 1 | 0 | 1 | | 1 | 1 | 0 | | 1 | 1 | 1 |  |  |  |  |  | | --- | --- | --- | --- | | 1 0 | 0 | 0 | 0 | | 0 | 0 | 1 | | 0 | 1 | 0 | | 0 | 1 | 1 | | 1 | 0 | 0 | | 1 | 0 | 1 | | 1 | 1 | 0 | | 1 | 1 | 1 |  |  |  |  |  | | --- | --- | --- | --- | | 1 1 | 0 | 0 | 0 | | 0 | 0 | 1 | | 0 | 1 | 0 | | 0 | 1 | 1 | | 1 | 0 | 0 | | 1 | 0 | 1 | | 1 | 1 | 0 | | 1 | 1 | 1 | |
| Рис. 5. Разбиение полных планов на подмножества для случая, когда n=5, h=2. |

В приводимом ниже cодержательном описании композитной версии полного перебора выражение .означает, что объект «а» лучше чем объект «b».

Алгоритм 5

1. Величине W присваивается значение, равное +∞, если целевая функция задачи (2) минимизируется, и значение, равное -∞ в противном случае.
2. Множество всех полных планов разбивается на 2h подмножеств, где h – число зафиксированных значений вектора булевых переменных Z (см. рис. 5).
3. Каждому полученному на предыдущем шаге подмножеству присваивается номер j:

 (9)

1. Каждому j-у подмножеству ставится в соответствие вектор V(j), компоненты которого определяются следующим образом: для формирования первой компоненты используется процедура вычисления оценки, применяемая в методах типа ветвей и границ, при условии, что первые h компонент вектора переменных Z фиксированы и определяются двоичным представлением числа j. Для определения второй компоненты вектора V(j) используется (5), остальные компоненты вычисляются по формуле (6).
2. Если существуют два таких вектора j и k, для которых справедливо:  то k-е подмножество исключается из числа рассматриваемых.
3. Осуществляется упорядочение π = {j1, j2, …} оставшихся после реализации шага 4 подмножеств по мере ухудшения первой компоненты вектора V(j): 
4. q = 1.
5. Исследуется q-е подмножество полных планов перестановки π. Фиксируется наилучший найденный полный план Zq и соответствующее значение целевой функции Fq. Если  то прежнее значение W забывается, а новое, равное Fq, запоминается.
6. Если существует вектор V(j), отвечающий необследованному подмножеству полных планов, для которого справедливо: то j-е подмножество полных планов исключается из числа рассматриваемых, а индексы всех следующих за ним векторов уменьшаются на единицу.
7. Если существует не анализировавшееся подмножество полных планов, то перейти к шагу 10, в противном случае – к шагу 11.
8. q = q+1, перейти к шагу 7.
9. Конец алгоритма. Оптимальное значение целевой функции равно W.

Пример применения алгоритма 5 при решении приведенной ранее задачи (3) приводится ниже. При этом следует учитывать, что первый шаг иллюстрируется рисунком 5.

Шаги 2 и 3 – на них формируются векторы V(j), j=0,1,2,3, которые представлены ниже на рис. 6 а)-г) ниже :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| j = 0 | |  | j = 1 | |  | j = 2 | |  | j = 3 | |
| 0 0 | 16,0,10 |  | 0 1 | 19,3,6 |  | 1 0 | 23,7,7 |  | 11 | 26,10,3 |

а) б) в) г)

Рис. 6. Векторы, характеризующие подмножества полных планов задачи (5), образованные частичными планами с фиксированными двумя первыми переменными (h=2).

Шаг 4: Поскольку все компоненты вектора V(2) лучше одноименных компонент вектора V(1), все полные планы, первые две компоненты которых равны, соответственно, «0» и «1», исключаются из числа анализируемых.

Шаг 5. Перестановка π оставшихся векторов: π = V(3), V(2), V(0).

Шаги 6 и 7. q = 1. Результаты анализа всех полных планов, первые две компоненты которых равны единице: Fmax= 15, Zopt = 11100, R=15.

Шаги 10 и 7. q = 2. Результаты анализа всех полных планов, первые две компоненты которых соответственно равны единице и нулю: Fmax= 16, Zopt = 10010, R=16.

Шаг 8. Поскольку первая компонента вектора V(0) не больше величины R, подмножество полных планов, первые две компоненты которых являются нулевыми, исключается из рассмотрения. Поскольку все множество полных планов исчерпано, переход к шагу 11.

Шаг 11. Конец алгоритма. Полученные оптимальные значения целевой функции и вектора переменных: Fmax= 16, Zopt = 10010.

Относительное приращение интеллекта алгоритмом 5 при решении системы (3) по сравнению полным перебором всех полных планов этой системы равно η5:



где N – число всех различных полных планов решаемой задачи; N5 – число полных планов, принадлежащих одному из 2h подмножеств, выделенных на первом шаге алгоритма 5: 

Иными словами, применительно к задаче (3) композиционный алгоритм 5 продемонстрировал интеллект, который вдвое превышает «наивную» реализацию полного перебора.

1. **Заключение**

Предложенный выше подход позволяет, в рамках сделанных допущений, повысить интеллект алгоритмов поиска глобально оптимальных решений задач дискретной оптимизации с булевыми переменными. При этом легко прослеживается взаимосвязь между приведенными выше композитными алгоритмами: обладая аналогичными процедурами отсечения «плохих» планов, они разнятся лишь стратегиями ветвления. Так, применительно к алгоритму 5, неоднократное повторение шагов 1 – 6, при котором разбиение на подмножества полных планов на шаге 1 охватывает лишь планы, соответствующие первой компоненте перестановки π, получаемой на текущей итерации, сближает такую модификацию этого алгоритма с алгоритмом 3. Если же повторяются шаги 1 – 4, причем шаг 1 заменяется следующим: 1. Все ранее полученные подмножества полных планов одновременно разбиваются на 2h подмножеств, где h – число зафиксированных значений вектора булевых переменных Z (h=2), то такая модификация алгоритма 5 оказывается близка к алгоритму 4. Увеличение числа тестов на отсечение подмножеств «плохих» планов в композитных алгоритмах по сравнению с соответствующими традиционными процедурами в соответствии с (1) *a priori* свидетельствует об их более высоком интеллекте. При этом следует учитывать, что допущение о линейной зависимости времени поиска решения от числа вершин построенного дерева ветвлений применительно к методам типа ветвей и границ, осуществляющим фронтальный спуск по дереву ветвлений G(X,U), справедливо для сочетания задач сравнительно небольших размерностей с эффективным способом вычисления оценки: с ростом числа висячих вершин дерева ветвлений G(X,U) основные затраты времени на каждой итерации смещаются с вычисления компонент векторов V(xj), xjϵX, к сравнению их первых компонент. Таким образом, дальнейшее развитие предлагаемого подхода может быть связано с:

а) экспериментальным анализом границ его эффективности применительно к методам поиска глобально оптимальных решений задач дискретной оптимизации;

б) априорным и экспериментальным анализом его эффективности применительно к детерминированным и рандомизированным методам локальной оптимизации.

Литература

1. Лорьер Ж. Л*.* Системы искусственного интеллекта, М.: Мир, 1991.  568 с.
2. G. F. Luger. Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 4-th edition, Williams, 2005, pp. 864 с. [ISBN 5-8459-0437-4](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/5845904374).
3. Handbook of Operations Research, v.2: Models and Applications. Edited by Joseph J. Moder and Salah E. Elmaghraby, North Carolina State University, 1978, 677p.
4. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн, К*.* Алгоритмы: построение и анализ, 2-e издание. М.: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2005.  Стр. 833 – 839.
5. A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems, Econometrica, Vol. 28, No. 3. (Jul., 1960), pp. 497-520.
6. Беллман Р. Динамическое программирование. — М.: Изд-во иностранной литературы, 1960.
7. [V. Chvátal](https://ru.wikipedia.org/w/index.php?title=%D0%A8%D0%B2%D0%B0%D1%82%D0%B0%D0%BB,_%D0%92%D0%B0%D1%81%D0%BB%D0%B0%D0%B2&action=edit&redlink=1) et al. Selected combinatorial research problems.  Technical Report STAN-CS-72-292. Computer Science Department, Stanford University, 1972.
8. [Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование.](http://www.google.ru/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwj6n4Kuu67RAhUEFywKHb9JBMcQFggaMAA&url=http%3A%2F%2Fwww.twirpx.com%2Ffile%2F86280%2F&usg=AFQjCNG86Mx0elVaw5s0mgWzg289tPSCeg&bvm=bv.142059868,d.bGg) М.: Наука, 1969. 368 с.
9. Гроппен В.О. Повышение эффективности методов типа ветвей и границ для комбинаторных задач с булевыми переменными. ‘’Автоматика и телемеханика’’ N 5, 1978г., с. 105-112.
10. Гроппен В.О. Эффективная организация комбинаторных алгоритмов на однородных вычислительных средствах. Труды 5-го Международного семинара «Прикладные аспекты теории автоматов», Варна, 1979г., с. 358-364.
11. Гроппен В.О. Экстремальные задачи на взвешенных графах. Изд. Фламинго, Владикавказ, 2012г., 90 с.
12. Кербер М. Л., Полимерные композиционные материалы. Структура. Свойства. Технологии. — СПб.: Профессия, 2008, 560 с.
13. **Перепелкин К.Е. Армирующие волокна и волокнистые полимерные композиты,** СПб.: Научные основы и технологии, 2009 **г., 380 с.**
14. Мелешко А.И. Половников С.П.Углерод, углеродные волокна, углеродные композиты, 2007, ISBN 5-88070-119-0, стр. 192.
15. Васильев В. В., Механика конструкций из композиционных материалов, М.: Машиностроение, 1988, 272 с.
16. Федунов Б. Е. Методика экспресс - оценки реализуемости графа решений оператора антропоцентрического объекта на этапе разработки спецификаций алгоритмов бортового интеллекта. М.: Известия РАН, Теория и системы управления (ТиСУ), №1, 2002.
17. А. А. Разборов. [О сложности вычислений](http://www.mccme.ru/free-books/matpros/i4127141.pdf.zip) //[МЦНМО](https://ru.wikipedia.org/wiki/%D0%9C%D0%A6%D0%9D%D0%9C%D0%9E), [Математическое просвещение](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D0%B2%D0%B5%D1%89%D0%B5%D0%BD%D0%B8%D0%B5), № 3, 1999, стр. 127-141.