

УДК 519.95

ЛОКАЛЬНЫЙ ПОИСК С ЗАПРЕТАМИ ДЛЯ ДИСКРЕТНОЙ ЗАДАЧИ О $(r|p)$ -ЦЕНТРОИДЕ

И. А. Давыдов

Аннотация. Для дискретной задачи о $(r|p)$ -центроиде разработан вероятностный метод локального поиска с запретами. Получены ограничения на список запретов, при которых алгоритм обладает следующим свойством: с ростом числа итераций вероятность отыскания глобального оптимума стремится к единице. Для оценки значений целевой функции используется метод лагранжевых релаксаций. Показано, что такая оценка не уступает оценке линейной релаксации. Исследуются различные алгоритмы субградиентной оптимизации для поиска оптимальных множителей Лагранжа. Проведены экспериментальные исследования на тестовых примерах из электронной библиотеки «Дискретные задачи размещения». Результаты экспериментов свидетельствуют о высокой частоте получения глобального оптимума разработанным методом.

Ключевые слова: конкурентные размещения, игры Штаккельберга, двухуровневое программирование.

Введение

В дискретной задаче о $(r|p)$ -центроиде заданы два конечных множества: множество клиентов J и множество предприятий I . Два игрока последовательно принимают решения об открытии предприятий. Сначала на рынок выходит первый игрок (лидер) и открывает свои p предприятий. Зная это решение, второй игрок (конкурент) открывает собственные r предприятий. Каждый клиент из открытых $p + r$ предприятий выбирает одно предприятие согласно собственным предпочтениям. В итоге множество клиентов делится между двумя игроками. Обслуживание каждого клиента приносит определённый доход. Каждый игрок стремится максимизировать свой доход. Задача состоит в том, чтобы найти p предприятий для лидера, которые дают ему максимальный доход при наилучшем ходе конкурента. Эта задача может быть сформулирована как задача линейного двухуровневого программирования или

как некооперативная игра Штаккельберга [6]. В [4] она исследовалась под названием конкурентной задачи о p -медиане.

Задача о $(r|p)$ -центроиде Σ_2^P -трудна, так как соответствующая ей задача распознавания полна в данном классе [19]. Этот класс является надмножеством класса NP, а рассматриваемая задача более сложна, чем широко известные NP-полные задачи. Полиномиально разрешимые частные случаи задачи и вычислительную сложность её различных подклассов можно найти в [15, 19].

Обозначим через X произвольное подмножество множества I , состоящее из p предприятий. Поиск наилучшего решения конкурента для заданного множества X назовём *задачей конкурента*. В [15] она называется *задачей о $(r|X)$ -медианоиде*. Её оптимальное решение позволяет вычислить максимальный доход конкурента, если лидер открывает все предприятия из множества X . Эта задача NP-трудна [15], что отчасти объясняет столь высокий сложностной статус исходной задачи. Кроме того, данное обстоятельство серьёзно затрудняет разработку численных методов. Если вычисление целевой функции лидера на произвольном допустимом решении является NP-трудной задачей, то поиск наилучшего допустимого решения представляется чрезвычайно непростым делом. В [8, 9] исследовались итерационные методы, в которых задача конкурента решалась жадными алгоритмами. Такой подход позволяет быстро находить неплохие решения лидера, но их погрешность сильно зависит от погрешности жадных эвристик. При малых значениях r ($r \leq 3$) такой подход даёт хорошие результаты [8], однако при больших значениях r доход лидера может оказаться существенно завышенным.

В [6, 17] исследовались верхние оценки оптимума. В [17] строится вспомогательная задача, исходя из предположений, что конкурент определённым образом сортирует предприятия по предварительно вычисляемой доходности и открывает ровно r предприятий согласно этому упорядочению. Если лидер захватывает какие-то из наиболее предпочтительных предприятий, то конкурент берёт следующие в том же порядке. Лидер знает это упорядочение и находит оптимальное решение против такой стратегии конкурента. Так как конкурент действует согласно выбранному упорядочению, а не ищет оптимальное решение, получаем верхнюю оценку на оптимум. В [6] показано, что такая верхняя оценка часто оказывается грубой, и предлагается принципиально другой подход, основанный на идее метода генерации столбцов. Исходная задача записывается как задача целочисленного линейного программирования с большим числом переменных и ограничений. Выбирая небольшое

подмножество ограничений и переменных и решая получившуюся подзадачу, получаем верхнюю оценку. Систематически пополняя это подмножество, можно получить глобальный оптимум. Именно таким методом получены оптимальные решения в тестовых примерах из электронной библиотеки «Дискретные задачи размещения» при $r = p \leq 7$ и $|I| = |J| = 100$, однако время работы такого метода оказывается неприемлемо большим. Для указанных размерностей оно часто составляет несколько часов работы на современном персональном компьютере.

В [1] исследовались простые эвристики, основанные на классической задаче о p -медиане. Показано, что игнорирование конкурента и стремление лидера поставить свои предприятия как можно ближе к клиентам приводит к допустимым решениям с малой относительной погрешностью. Для вычисления целевой функции лидера и решения задачи конкурента применяется метод ветвей и границ. В [4] разработан гибридный генетический алгоритм, в котором применяется вероятностный метод локального поиска с запретами к каждому элементу популяции. Для оценки соседних решений используется точное решение линейной релаксации задачи конкурента. При переходе от текущего решения лидера к соседнему решению значение целевой функции вычисляется точно методом ветвей и границ. Такой подход позволяет находить оптимальные решения для примеров из электронной библиотеки «Дискретные задачи размещения», но время счёта оказывается достаточно большим.

В настоящей работе метод поиска с запретами применяется без связи со схемой генетических алгоритмов. Приводится доказательство асимптотической точности алгоритма: с ростом числа итераций вероятность найти глобальный оптимум стремится к единице. Для оценки соседних решений лидера используется метод лагранжевых релаксаций. Задача конкурента записывается в терминах целочисленного линейного программирования. Часть ограничений заносится в целевую функцию с множителями Лагранжа. Для поиска оптимальных множителей применяются методы субградиентной оптимизации. Рассмотрены три метода: классический метод с движением вдоль субградиента, метод с движением в направлении разности двух последовательных субградиентов и так называемый объёмный метод [7]. Показано, что несмотря на то, что все три метода в асимптотике дают одно и то же значение функционала, второй метод имеет определённые преимущества по точности при малом числе итераций. Показано, что оценки, полученные путем решения линейной релаксации задачи конкурента, совпадают с оценками лагранжевой релаксации. Исследована зависимость эффективно-

сти локального поиска с запретами от управляющих параметров. Проведены численные эксперименты на примерах из электронной библиотеки «Дискретные задачи размещения», которая находится по адресу: http://math.nsc.ru/AP/benchmarks/competitive/p_med_comp.html. Результаты этих экспериментов показывают, что новый метод заметно быстрее своих предшественников находит известные оптимальные решения лидера. Если же время решения задачи на персональном компьютере ограничить пятью минутами, то частота получения оптимума остаётся достаточно большой, а средняя относительная погрешность получаемых решений оказывается незначительной и не превосходит трёх процентов.

1. Постановка задачи

Обозначим через ω_j величину дохода, получаемого игроком от обслуживания j -го клиента. Будем предполагать, что это положительная величина для каждого $j \in J$. Предпочтения клиентов на множестве предприятий будем задавать матрицей (g_{ij}) . Если $g_{ij} > g_{kj}$, то клиент j из открытых предприятий i и k выбирает предприятие k . Для наглядности можно считать, что матрица (g_{ij}) задаёт расстояния от клиентов до предприятий и клиент предпочитает наименее удалённое предприятие независимо от того, какому игроку оно принадлежит. Введём переменные задачи:

$$x_i = \begin{cases} 1, & \text{если лидер открывает предприятие } i, \\ 0 & \text{в противном случае,} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{если конкурент открывает предприятие } i, \\ 0 & \text{в противном случае,} \end{cases}$$

$$u_j = \begin{cases} 1, & \text{если клиент } j \text{ обслуживается из предприятия лидера,} \\ 0, & \text{если клиент } j \text{ обслуживается из предприятия конкурента.} \end{cases}$$

Пусть вектор x задаёт решение лидера. Определим множество предприятий $I_j(x)$, позволяющих конкуренту захватить клиента j , т. е.

$$I_j(x) = \{i \in I \mid g_{ij} < \min_{l \in I} \{g_{lj} \mid x_l = 1\}\}.$$

Из этого определения следует, что в случае равных расстояний до предприятий лидера и конкурента клиент отдаёт предпочтение лидеру [15]. Другие варианты поведения клиентов можно найти в [10].

С использованием введённых обозначений задача о $(r|p)$ -центроиде может быть представлена как задача линейного двухуровневого программирования [4].

Найти $\max_x \sum_{j \in J} \omega_j u_j^*(x)$ при ограничениях

$$\sum_{i \in I} x_i = p, \quad x_i \in \{0, 1\}, \quad i \in I,$$

где $u_j^*(x), y_i^*(x)$ — оптимальное решение задачи конкурента:

найти $\max_{y, u} \sum_{j \in J} \omega_j (1 - u_j)$ при ограничениях

$$\sum_{i \in I} y_i = r, \quad 1 - u_j \leq \sum_{i \in I_j(x)} y_i, \quad y_i, u_j \in \{0, 1\}, \quad i \in I, \quad j \in J.$$

Целевая функция $W(x) = \sum_{j \in J} \omega_j u_j^*(x)$ задаёт суммарный доход лидера при условии, что он открывает ровно p предприятий. Эта величина зависит от оптимального решения задачи конкурента. Вектор x и множества $I_j(x)$ для всех $j \in J$ в этой задаче считаются известными. Целевая функция в задаче конкурента определяет его доход. Первое ограничение позволяет конкуренту открывать ровно r предприятий. Второе ограничение не разрешает конкуренту обслуживать клиента j , если в множестве $I_j(x)$ нет его предприятий.

Представление задачи в терминах двухуровневого программирования подсказывает естественный способ её решения: ограничить поиск пространством переменных x , вычисляя тем или иным способом значения функции $W(x)$. Так как точное значение вычислить достаточно трудно, а в ходе решения задачи это придётся делать многократно, далее сосредоточим основное внимание на приближённом решении задачи конкурента.

2. Лагранжева релаксация

Представим задачу конкурента эквивалентным образом, увеличив число переменных. Пусть новые переменные y_{ij} определяют поведение клиентов:

$$y_{ij} = \begin{cases} 1, & \text{если клиент } j \text{ выбирает предприятие } i, \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда задачу конкурента можно записать следующим образом:

$$\max_{y_i, y_{ij}} \sum_{j \in J} \omega_j \sum_{i \in I_j(x)} y_{ij} \tag{1}$$

при ограничениях

$$\sum_{i \in I_j(x)} y_{ij} \leq 1, \quad j \in J, \quad (2)$$

$$y_i \geq y_{ij} \geq 0, \quad i \in I, \quad j \in J, \quad (3)$$

$$\sum_{i \in I} y_i = r, \quad (4)$$

$$y_i, y_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J. \quad (5)$$

Поставим в соответствие ограничениям (2) вектор неотрицательных переменных λ_j и рассмотрим лагранжеву релаксацию задачи по этим ограничениям:

$$\text{LR}(\lambda) = \max_{y_i, y_{ij}} \sum_{j \in J} (\omega_j - \lambda_j) \sum_{i \in I_j(x)} y_{ij} + \sum_{j \in J} \lambda_j \quad (6)$$

при ограничениях (3)–(5).

Заметим, что при заданном векторе λ величина $\text{LR}(\lambda)$ может быть вычислена точно за полиномиальное время. Пусть множество $J_i(x)$ задаёт тех клиентов, которые предпочтут предприятие i любому из предприятий лидера в решении x , если это предприятие будет открыто конкурентом:

$$J_i(x) = \{j \in J \mid i \in I_j(x)\}.$$

Для каждого $i \in I$ вычислим доход конкурента от открытия этого предприятия, равный $\sum_{j \in J_i(x)} \max\{0, \omega_j - \lambda_j\}$, и выберем r наиболее доходных предприятий. Легко проверить, что такое решение даёт точное значение величины $\text{LR}(\lambda)$. Она является верхней оценкой на максимальный доход конкурента при любых неотрицательных значениях величин λ_j . Решение двойственной задачи

$$D = \min_{\lambda \geq 0} \text{LR}(\lambda)$$

даёт наилучшую из таких верхних оценок. Обозначим через LP оптимальное значение линейной релаксации задачи конкурента. Следующая теорема показывает связь верхней оценки D с соответствующей задачей линейного программирования (1)–(4).

Теорема 1. $D = LP$.

ДОКАЗАТЕЛЬСТВО. Известно [12], что $D \leq LP$. Покажем обратное неравенство. Обозначим через λ^* оптимальный вектор множителей

Лагранжа, т. е. $D = \text{LR}(\lambda^*)$. Пусть $\overline{\text{LR}}(\lambda^*)$ — оптимальное значение задачи линейного программирования (6), (3), (4). В силу того, что

$$\overline{\text{LR}}(\lambda^*) \geq \min_{\lambda \geq 0} \overline{\text{LR}}(\lambda) = LP,$$

для доказательства теоремы достаточно показать, что $\text{LR}(\lambda) = \overline{\text{LR}}(\lambda)$. Другими словами, требуется установить, что среди оптимальных решений задачи линейного программирования (6), (3), (4) всегда найдётся оптимальное решение с целочисленным вектором y_i , $i \in I$. Предположим, что это не так, и пусть \bar{y}_i, \bar{y}_{ij} — оптимальное дробное решение с максимальным числом целочисленных компонент вектора \bar{y}_i . Поскольку r целое, дробных компонент как минимум две. Обозначим их через i_1 и i_2 , а соответствующие слагаемые в целевой функции — через

$$a_{i_k} = \sum_{j \in J_{i_k}(x)} \max(0, \omega_j - \lambda_j) \bar{y}_{ij}, \quad k = 1, 2.$$

Заметим, что $a_{i_1} = a_{i_2}$, так как в противном случае появляется возможность увеличить значение целевой функции, что невозможно в виду оптимальности решения. Предположим, что $y_{i_1} + y_{i_2} \leq 1$. Построим новое решение, отличающееся от старого только в компонентах i_1 и i_2 :

$$\begin{cases} y_{i_1} = \bar{y}_{i_1} + \bar{y}_{i_2} \\ y_{i_2} = 0 \end{cases} \quad \begin{cases} y_{i_1 j} = \bar{y}_{i_1 j} + \bar{y}_{i_2 j} \\ y_{i_2 j} = 0 \end{cases}, \quad j \in J.$$

В силу (3) это решение допустимо и оптимально, но имеет на одну целочисленную компоненту больше; противоречие. Случай $y_{i_1} + y_{i_2} \geq 1$ рассматривается аналогично. Теорема 1 доказана.

Ввиду совпадения оценок линейной и лагранжевой релаксации предпочтение будет отдано последней, так как при работе алгоритма используется приближённая оценка решения задачи конкурента. Применение лагранжевой релаксации позволяет существенно уменьшить трудоёмкость вычислений при локальном поиске, поскольку во время просмотра окрестности возможно использовать вектор λ , полученный на предыдущей итерации в качестве начального приближения. Заметим, что двойственная задача относится к классу задач недифференцируемой оптимизации. Для её решения хорошо зарекомендовали себя субградиентные методы. Последние достижения в данной области можно найти в обзорных статьях [11, 14, 18].

Методы субградиентной оптимизации предполагают построение последовательности неотрицательных векторов $\{\lambda^k\}$, начиная с некоторого начального вектора λ^0 . Пусть $y_i(\lambda^k), y_{ij}(\lambda^k)$ — оптимальное решение

релаксированной задачи при заданном векторе λ^k . Тогда невязка в ограничении (2), определяемая равенством

$$s_j = 1 - \sum_{i \in I_j(x)} y_{ij}(\lambda), \quad j \in J,$$

задаёт субградиент функции $\text{LR}(\lambda)$ [14]. В базовом варианте метода последовательность векторов задаётся следующим правилом:

$$\lambda^{k+1} = \max\{0, \lambda^k + t^k s^k\},$$

где t^k — длина шага в направлении субградиента s^k . Известно [20], что $LP(\lambda^k) \rightarrow D$ при таком выборе шага, что $t^k \rightarrow 0$ и $\sum_k t^k \rightarrow \infty$. Однако на практике используют другие правила выбора шага, например [16]:

$$t^k = \theta^k \frac{\text{LR}(\lambda^k) - D'}{\|s^k\|^2},$$

где D' — нижняя оценка на значение D , а θ^k — монотонно убывающая последовательность. Последовательность начинается с $\theta^0 = 2$, и через определённое число итераций, зависящее от размерности задачи, эта величина делится пополам. Критерием остановки служит выполнение неравенства $\theta^k < \varepsilon$ для достаточно малого положительного порога ε .

Наряду с базовым вариантом метода рассмотрим его модификацию, когда на каждой итерации вектор λ меняется в направлении разности двух последовательных субградиентов [21]. При заданном положительном параметре γ , не превосходящем единицы, выбор очередного вектора λ^{k+1} определяется правилом

$$\lambda^{k+1} = \max\{0, \lambda^k + \gamma t^k s^k - (1 - \gamma)t^{k-1} s^{k-1}\}.$$

Значение γ подбирается экспериментально, хотя значения из интервала $[0,6, 0,7]$ обычно приводят к хорошим результатам.

Наконец, третий метод, с успехом применённый в [7] для получения приближённых решений близкой по смыслу дискретной задачи размещения, также опирается на идеи субградиентной оптимизации. На каждой итерации этого метода используется аппроксимация оптимального решения задачи (1)–(4) для непрерывных переменных y_i, y_{ij} . Общая схема метода выглядит следующим образом.

ШАГ 1. Выбрать начальный вектор λ и найти решение y_i, y_{ij} релаксированной задачи, положить $k := 1$.

ШАГ 2. Положить $v^k := 1 - \sum_{i \in I_j(x)} y_{ij}$ и $\lambda^k := \lambda + t^k v^k$.

ШАГ 3. Решить релаксированную задачу, получить $\text{LR}(\lambda^k)$ и y_i^k, y_{ij}^k , положить $(y_i, y_{ij}) := \alpha(y_i^k, y_{ij}^k) + (1 - \alpha)(y_i, y_{ij})$.

ШАГ 4. Если $\text{LR}(\lambda^k) < \text{LR}(\lambda)$, то заменить вектор двойственных переменных $\lambda := \lambda^k$.

ШАГ 5. Если не выполнен критерий остановки $\theta^k < \varepsilon$, то $k := k + 1$ и перейти на шаг 2.

При выборе длины шага t^k используется новое правило изменения параметра θ^k . Выделяются три типа итераций.

Если $\text{LR}(\lambda^k) \geq \text{LR}(\lambda)$, то итерация называется *красной*. На таких итерациях размер шага следует уменьшать.

Если $\text{LR}(\lambda^k) < \text{LR}(\lambda)$, то вычисляется субградиент

$$s^k := 1 - \sum_{i \in I_j(x)} y_{ij}^k$$

и скалярное произведение $d = s^k v^k$. Если $d \geq 0$, то итерация называется *зелёной*. На таких итерациях размер шага следует увеличивать.

Если $\text{LR}(\lambda^k) < \text{LR}(\lambda)$ и $d < 0$, то итерация называется *жёлтой*. На таких итерациях размер шага можно не менять.

На каждой зелёной итерации величина θ умножается на 1,1. После двадцати последовательных красных итераций величина θ умножается на 0,6. Выбор этих значений определялся в ходе экспериментальных исследований [7].

Для вычисления скалярной величины α на шаге 3 используется вспомогательная задача:

$$\min \|\alpha s^k + (1 - \alpha)v^k\|,$$

$$\frac{b}{10} \leq \alpha \leq b,$$

для некоторой константы $b \leq 0,1$. Содержательный смысл этой задачи состоит в минимизации нормы вектора v^{k+1} со свободой в выборе значения α .

Характерно, что данный метод не имеет зигзагообразного поведения, свойственного базовому методу. На шаге 4 вектор двойственных переменных меняется только при улучшении верхней оценки оптимума. Промежуточные итерации предназначены для выбора подходящего направления и длины шага в этом направлении.

На рис. 1 представлены графики поведения методов субградиентной оптимизации для задачи конкурента на примере 111 из класса *Euclidean* при $p = r = 5$. Подробное описание примеров приведено ниже (см. разд. 4). По оси ОХ откладываются итерации. По оси ОУ — значения верхней оценки $LR(\lambda)$. Базовый метод (grad1) имеет ярко выраженную зигзагообразную кривую. Объёмный метод (vol) монотонно улучшает верхнюю оценку, но явно проигрывает второму методу с движением в направлении разности двух последовательных субградиентов (grad2) на первых пятидесяти итерациях. Кривая grad2 лежит ниже других и даёт хорошее приближение для оптимума двойственной задачи. Аналогичное поведение эти методы демонстрируют и на других тестовых примерах. Похожая картина наблюдается и при решении простейшей задачи размещения [7]. Именно второй метод используется далее для оценки оптимальных решений конкурента при локальных изменениях в решении лидера. Здесь и далее в численных экспериментах для вычисления величины D' используется формула $D' = 0,9LR(\lambda^k)$.

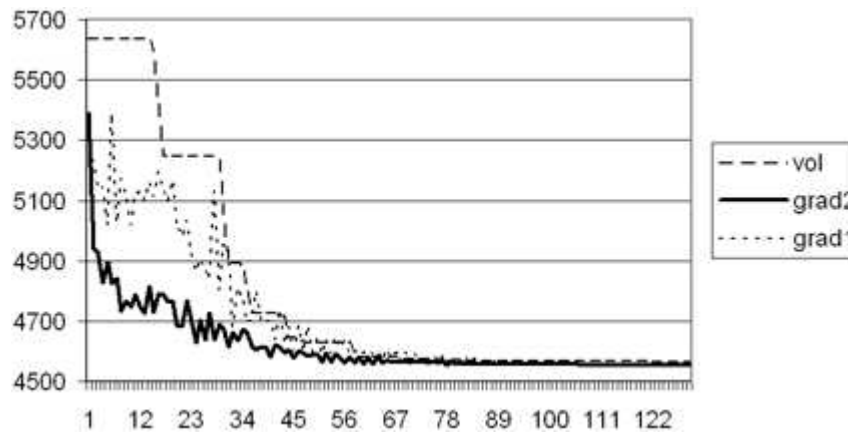


Рис. 1. Поведение методов субградиентной оптимизации

3. Локальный поиск с запретами

Данный метод, предложенный Гловером и сначала опробованный на задаче коммивояжёра, широко используется для решения различных NP-трудных задач [13]. В его основе лежит оригинальная схема локального поиска, позволяющая не останавливаться в точке локального оптимума, а «путешествовать» от одного оптимума к другому в надежде найти среди них глобальный оптимум. Основным механизмом, позволяющим методу выбираться из локального оптимума, является список

запретов. В этот список заносятся решения предшествующих итераций или фрагменты таких решений (рёбра графов, цвета вершин, номера заменяемых предприятий и т. п.). С помощью списка запретов из окрестности текущего решения удаляется часть соседних решений и выполняется переход в лучшее из оставшихся решений. Наряду со списком запретов важную роль играет рандомизация окрестности. Она также позволяет избежать заикливания, существенно сокращает время на одну итерацию и повышает эффективность поиска. При определённых ограничениях на длину списка запретов такой метод в асимптотике позволяет находить точное решение задачи [2].

Ниже приводится описание данного метода с адаптацией основных элементов для решения задачи лидера.

Обозначим через $N(x)$ множество всех решений, получающихся из решения x закрытием одного предприятия и открытием другого. Это множество будем называть *окрестностью решения x* , а его элементы — *соседними решениями* для x . Любое допустимое решение лидера имеет ровно $p(|I| - p)$ соседних решений. Рандомизированная окрестность $N_q(x)$ является подмножеством этой окрестности. Каждый элемент из множества $N(x)$ включается в множество $N_q(x)$ с заданной вероятностью q . Множество $N_q(x)$ может оказаться пустым или совпадать с исходным множеством $N(x)$.

При переходе от любого допустимого решения к соседнему меняются с 1 на 0 и с 0 на 1 ровно две координаты. Чтобы воспрепятствовать движению в обратном направлении, будем хранить пары изменяемых координат и удалять соответствующих соседей из окрестности. Упорядоченный список таких пар будем называть *списком запретов*. Его длина l может существенно сказываться на эффективности поиска: при малой длине алгоритм может заикливаться, при большой из окрестности будет удаляться неоправданно много решений. Длина списка является параметром алгоритма. Она может меняться с целью интенсификации или диверсификации поиска. В [2] показано, что при использовании рандомизированных окрестностей влияние списков запретов падает и даже малые по длине списки приводят к быстрому обнаружению глобального оптимума или к нахождению приближённых решений с малой погрешностью.

Обозначим через φ список запретов, а через $N_q(x, \varphi)$ — множество решений из окрестности $N_q(x)$ без запрещённых элементов. Тогда общая схема локального поиска может быть представлена следующим образом.

ШАГ 1. Построить начальное решение x , положить $\varphi := \emptyset$, $W_A :=$

$W(x)$.

ШАГ 2. Пока не выполнен критерий остановки, делать следующее

ШАГ 2.1. Сформировать окрестность $N_q(x)$ и удалить из неё запрещённые элементы.

ШАГ 2.2. Если $N_q(x, \varphi) \neq \emptyset$, то найти в нём решение x' с наибольшей оценкой дохода лидера. Если $N_q(x, \varphi) = \emptyset$, то вернуться на шаг 2.1.

ШАГ 2.3. Вычислить точно значение $W(x')$. Если $W(x') > W_A$, то положить $W_A := W(x')$.

ШАГ 2.4. Обновить список запретов и положить $x := x'$.

ШАГ 3. Предъявить в качестве ответа наилучшее найденное значение W_A .

В качестве начального решения на шаге 1 выбирается приближённое решение задачи о p -медиане. Как уже отмечалось во введении, такой подход даёт неплохое начальное приближение. Предположим, что лидер игнорирует конкурента и стремится поставить свои предприятия как можно ближе к клиентам. Пусть переменные x_{ij} задают поведение клиентов при отсутствии конкурента: $x_{ij} = 1$, если клиент j обслуживается предприятием i , и $x_{ij} = 0$ в противном случае. Тогда задача о p -медиане записывается следующим образом:

$$\min_{x_i, x_{ij}} \sum_{j \in J} \sum_{i \in I} \omega_j g_{ij} x_{ij}$$

при ограничениях

$$\begin{aligned} \sum_{i \in I} x_{ij} &= 1, \quad j \in J, \\ x_i &\geq x_{ij}, \quad i \in I, j \in J, \\ \sum_{i \in I} x_i &= p, \\ x_{ij}, x_i &\in \{0, 1\}, \quad i \in I, j \in J. \end{aligned}$$

Как и для задачи конкурента, первое ограничение заносится в целевую функцию с множителями Лагранжа. При фиксированных множителях релаксированная задача легко решается точно. Для выбора оптимальных значений множителей Лагранжа применяется метод субградиентной оптимизации с движением в направлении разности двух последовательных субградиентов. На каждой итерации получается набор из p открываемых предприятий лидера. Лучшее из таких решений используется на шаге 1 в качестве начального решения для локального поиска.

Длина списка запретов и коэффициент рандомизации окрестности не меняются в ходе работы алгоритма. При переходе к очередной итерации в список запретов добавляется новая пара изменяемых координат в векторе x . Самая «старая» пара удаляется из списка. В качестве критерия остановки используется общее число выполненных итераций. На шаге 2.2 для оценки дохода лидера решается двойственная задача. Требуемая оценка вычисляется как разность между общим доходом $\sum_{j \in J} \omega_j$ и приближённым значением величины D . На шаге 2.3 используется метод ветвей и границ для вычисления точного значения величины $W(x')$.

Отметим, что в асимптотике алгоритм вероятностного поиска с запретами является точным. Для доказательства этого факта нам потребуется понятие конечной цепи Маркова. Известно [2], что вероятностный алгоритм поиска с запретами порождает цепи Маркова на конечном множестве состояний Ω всех пар вида (x, φ) , где x — допустимое решение и φ — список запретов. Если цепь неразложима, то алгоритм асимптотически точен.

Теорема 2. При $0 < l < n(n-1)/4$ и $0 < q < 1$ вероятностный алгоритм поиска с запретами порождает неразложимую цепь Маркова.

Доказательство. Пусть $(x, \xi), (y, \varphi) \in \Omega$. Построим путь от (x, ξ) до (y, φ) так, чтобы на каждом шаге вдоль этого пути не нарушались ограничения по списку запретов. Такой путь можно представить следующим образом. Рассмотрим случай, когда n нечётное. Выберем один из кратчайших путей из точки x в точку y и обозначим последовательность составляющих его пар через μ . Очевидно, что $|\mu| \leq n$. Обозначим через η множество всех пар (i, j) , $i < j$. На первом этапе переходим из точки x со списком запретов ξ в точку x со списком запретов ξ_1 , используя все шаги из η , где ξ_1 — любой список запретов, не содержащий l первых пар из μ . Отметим, что при последовательном переходе из точки x по всем парам из множества η мы попадём в точку x . Необходимо лишь выбрать правильный порядок перехода для соблюдения ограничений на список запретов. Для этого достаточно сначала сделать l шагов по парам из μ/ξ , а затем выбирать пары таким образом, чтобы среди последних l пар не было пар из μ (рис. 2). Так как $0 < l < n(n-1)/4$, такой переход возможен. Затем осуществляется переход из точки x в точку y по пути μ . Этот переход возможен, поскольку пары из пути μ не содержатся в списке запретов ξ_1 . На последнем этапе необходимо перейти из точки y со списком запретов φ_1 в точку y со списком запретов φ . Для этого осуществляется переход по всем парам, как на первом этапе. В случае

чётного n путь строится аналогично с использованием на промежуточных этапах точек $1 - x$, $1 - y$ и кратчайшего пути между ними. Теорема 2 доказана.

Следствие. При любом допустимом решении лидера x^0 , $0 < q < 1$ и $0 < l < n(n - 1)/4$

- (i) значение целевой функции W_A стремится к глобальному оптимуму W_{opt} почти наверное с ростом числа итераций,
- (ii) существуют константы $b > 1$ и $1 > c > 0$ такие, что

$$\Pr\{W_A \neq W_{\text{opt}}\} \leq bc^t.$$

ДОКАЗАТЕЛЬСТВО. Утверждения (i) и (ii) непосредственно следуют из свойства неразложимости и того факта, что величина W_A монотонно не убывает с ростом числа итераций. Следствие доказано.

Фактически, утверждение (i) означает, что алгоритм получит оптимальное решение за достаточно большое число шагов, т. е. он является асимптотически точным. Утверждение (ii) говорит о сходимости по вероятности величины W_A к оптимальному решению задачи со скоростью геометрической прогрессии.

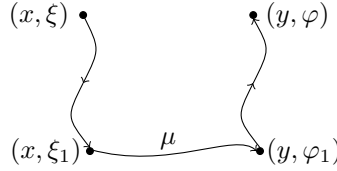


Рис. 2. Путь от (x, ξ) до (y, φ)

4. Экспериментальные исследования

Разработанный алгоритм запрограммирован в среде Delphi 7.0 и тестировался на компьютере Fujitsu–Siemens Amilo Pro V2040 1,6 ГГц. Все тестовые примеры имели одинаковую размерность $|I| = |J| = 100$. Они разбиты на два класса: *Euclidean* и *Uniform*. В первом классе элементы матрицы предпочтений клиентов (g_{ij}) — расстояния между точками на плоскости. В квадрате 7000 на 7000 выбиралось 100 точек случайным образом с равномерным распределением. Величины g_{ij} соответствовали расстояниям между точками i и j . Предполагалось, что множества I и J совпадают. Во втором классе каждый элемент матрицы (g_{ij}) выбирался независимо от других элементов случайным образом с равномерным распределением в интервале от 0 до 10^4 . В каждом классе рассматривалось

два типа доходов. В первом случае $w_j = 1$ для всех $j \in J$. Во втором случае эти величины выбирались независимо друг от друга случайным образом с равномерным распределением в интервале от 1 до 200. Порог точности ε во всех экспериментах принимался равным 10^{-6} .

Первый численный эксперимент связан с выбором параметров метода: коэффициентом рандомизации окрестности q и длиной списка запретов, которую далее будем обозначать через l . На рис. 3 показано типичное поведение метода при $q = 0,2$ и длинах списка запретов $l = 10$ и $l = 100$ на тестовом примере 111 из класса *Euclidean*, $p = r = 5$. На оси ОХ указано число итераций локального поиска, а на оси ОУ — значения целевой функции текущего решения.

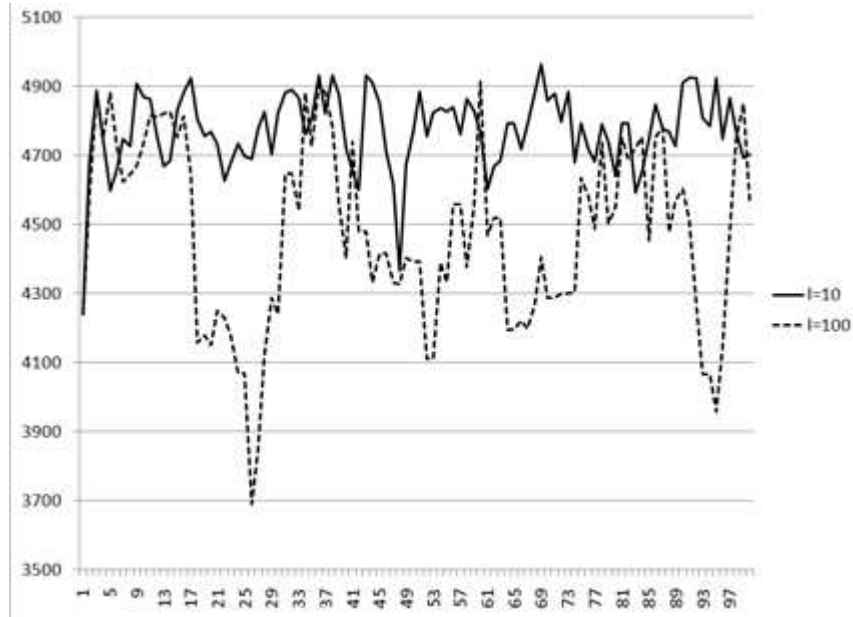


Рис. 3. Типичное поведение метода

Зигзагообразный характер кривых на этом рисунке подтверждает немонотонность локального поиска по целевой функции. При увеличении длины списка запретов поведение метода меняется. Большое количество допустимых решений оказывается запрещённым, что приводит к снижению эффективности поиска в среднем. При $l = 100$ кривая лежит значительно ниже кривой, соответствующей $l = 10$. При малых значениях параметра q для эффективной работы алгоритма достаточно коротких списков запретов. В дальнейших экспериментах длина списка запретов равна 10.

Следующий эксперимент касается выбора доли рандомизации окрестности. В [2] при рассмотрении многостадийной задачи размещения показано, что правильный выбор значения q существенным образом сказывается на эффективности локального поиска. На рис. 4 представлены результаты серии тестов со значениями параметра $q = 0,2; 0,3; 0,4$ на примере из класса *Uniform*. Проведено 100 испытаний алгоритма с каждым из трёх значений. Алгоритм останавливался после 15 минут работы. Такой критерий уравнивает шансы, так как при доле рандомизации $q = 0,2$ алгоритм совершает вдвое больше итераций, чем при $q = 0,4$. По оси абсцисс отложены значения целевой функции, а по оси ординат — число попаданий в эти решения. Как видно из рисунка, для данного класса примеров превосходство какого-либо из предложенных значений не является очевидным. Это может свидетельствовать о том, что данный класс примеров, по-видимому, не является сложным для вероятностного поиска с запретами, ввиду чего вопрос о доле рандомизации окрестности не является принципиально важным. Класс *Euclidean* является более простым с вычислительной точки зрения, потому те же рассуждения применимы и к примерам этого класса. В дальнейшем используется значение параметра $q = 0,3$.

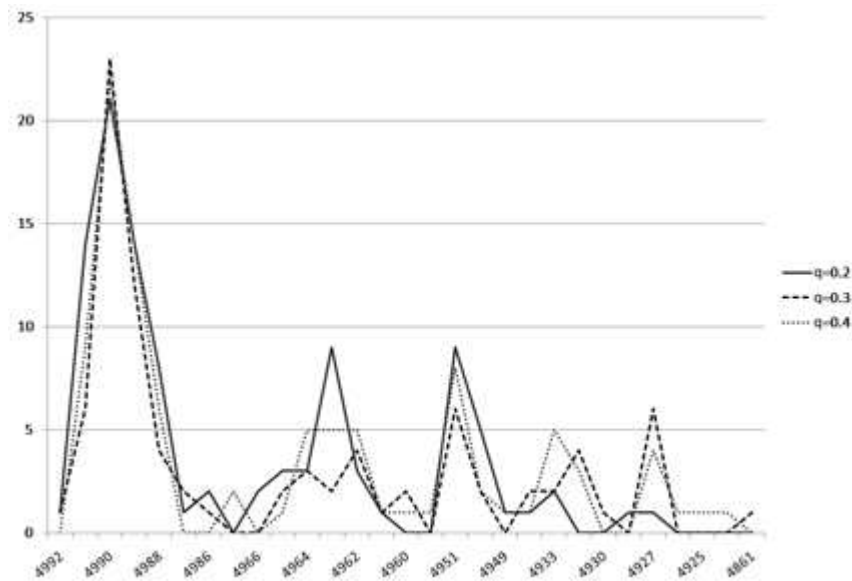


Рис. 4. Влияние рандомизации окрестности на эффективность поиска

Наряду с окрестностью $N(x)$, описанной в разд. 3, можно использовать и другие окрестности, например, окрестность *FM1* [3, 5]. Соседние

решения в этой окрестности строятся в два этапа. На первом этапе из множества открытых предприятий лидера выбирается одно и закрывается. На втором этапе для открытия выбирается любое из закрытых предприятий, исключая предприятие, закрытое на первом этапе. Мощность этой окрестности линейна и не превосходит $|I|$. Обе окрестности хорошо зарекомендовали себя при решении различных NP-трудных задач. Применительно к данной проблеме при использовании линейной окрестности *FM1* несмотря на значительный выигрыш в трудоёмкости алгоритм даже за вдвое большее время в среднем демонстрирует значительно более низкое качество получаемых решений, чем при использовании квадратичной окрестности $N(x)$. Возможно, причина кроется в том, что на этапе удаления предприятия лидера происходит значительная перестройка решения конкурента. Это приводит к неадекватной оценке принимаемого лидером решения. В дальнейших расчётах будет использоваться только окрестность $N(x)$.

В табл. 1 приведены результаты расчётов для класса *Euclidean* при $p = r = 5$, $\omega_j \in [1, 200]$. На каждом примере проводилось 10 испытаний алгоритма. Алгоритм останавливался после выполнения 100 итераций.

Т а б л и ц а 1

Результаты для класса *Euclidean*, $p = r = 5$

Код примера	Оптимум	Попаданий в оптимум	Средняя погрешность
111	4139	8	0,42%
211	4822	8	0,53%
311	4215	7	0,49%
411	4678	4	1,66%
511	4594	2	1,23%
611	4483	5	1,01%
711	5153	2	2,66%
811	4404	6	0,96%
911	4700	2	2,01%
1011	4923	5	1,82%

Как видно из табл. 1, на всех примерах алгоритму часто удаётся находить оптимум, а средняя погрешность невелика. Время счёта не превосходит 5 минут. Для сравнения в [4] приводятся результаты аналогичных расчётов для гибридного генетического алгоритма. Среднее время счёта составляет 7 часов на PC Pentium Intel Core 2, 1.87 GHz, 2 Gb RAM.

Аналогичные расчёты проведены для случая при $\omega_j = 1$ при $p = r = 10$. Значение глобального оптимума для этих параметров неизвест-

но, поэтому в качестве критерия останова использовалось достижение наилучшего известного решения из электронной библиотеки «Дискретные задачи размещения».

Т а б л и ц а 2

Результаты для класса *Euclidean*, $p = r = 10$

Код примера	Лучшее найденное	Число итераций	Время счёта
111	4361	38	342
211	5310	45	405
311	4483	91	819
411	4994	30	270
511	4906	44	396
611	4595	56	504
711	5586	23	207
811	4609	67	603
911	5302	33	297
1011	5005	37	333

При $\omega_j = 1$ алгоритм всегда находит наилучшее известное решение за 15 минут работы. Если же время счёта увеличить в 10 раз, то на всех примерах этого класса не удаётся улучшить ни одного значения. По-видимому, эти значения оптимальны, и следует разрабатывать специальные методы для проверки этого предположения.

Т а б л и ц а 3

Результаты для класса *Euclidean*, $p = r = 15$

Код примера	Лучшее найденное	Попаданий в лучшее	Время счёта
111	4596	8	545
211	5373	2	12021
311	4800	13	3126
411	5064	3	1442
511	5123	6	1928
611	4881	9	781
711	5827	2	2043
811	4675	2	1088
911	5158	1	1322
1011	5195	5	2403

Аналогичные результаты получены для случая $p = r = 10$, $\omega_j \in [1, 200]$. При ограничении в 1500 итераций алгоритм на всех примерах

находит наилучшие известные решения. В третьей колонке табл. 2 указан номер итерации, на которой впервые найдено лучшее решение. Четвёртая колонка характеризует время счёта в секундах для получения рекордного значения целевой функции. Как видно из табл. 2, наилучшее решение алгоритм находит на первых 100 итерациях. Тот факт, что на последующих итерациях это решение не улучшается, позволяет предполагать, что найденное решение оптимально.

В табл. 3 и 4 приводятся наилучшие найденные решения за 1000 итераций на примерах класса *Euclidean* с $p = 15$ и $p = 20$. В третьей колонке указано количество попаданий в лучшее найденное решение за время поиска, а в четвёртой — время счёта в секундах до нахождения рекорда.

Т а б л и ц а 4

Результаты для класса *Euclidean*, $p = r = 20$

Код примера	Лучшее найденное	Попаданий в лучшее	Время, счёта
111	4512	5	6008
211	5432	3	4922
311	4893	2	9365
411	5209	2	5165
511	5334	2	7922
611	4952	11	13081
711	5893	7	6000
811	4858	4	8526
911	5459	1	1023
1011	5399	2	2347

В табл. 5 приведены результаты расчётов для класса *Uniform*, $p = r = 10$, $\omega_j \in [1, 200]$. В качестве критерия останова выбрано количество итераций, не более 100.

Сравнивая результаты расчётов для классов *Uniform* и *Euclidean*, можно заметить, что класс *Uniform* более сложен. Алгоритму реже удаётся найти оптимум, хотя средняя погрешность незначительно превосходит аналогичный показатель класса *Euclidean*. Объясняется это тем, что на примерах из класса *Euclidean* при решении задачи конкурента, по-видимому, отсутствует разрыв двойственности, что позволяет методу лагранжевых релаксаций находить точное решение.

5. Заключение

Рассмотрена задача о $(r|p)$ -центроиде. Для её решения разработан итерационный метод локального поиска с запретами. Доказана асимптотическая точность разработанного алгоритма. Для оценки значений целевой функции допустимых решений используется алгоритм лагранжевых релаксаций. Показано, что такая оценка не уступает оценке линейной релаксации. Релаксированная задача решается методом субградиентной оптимизации. Использование техники лагранжевых релаксаций позволяет избежать многократного обращения к трудоёмкой процедуре вычисления точного решения задачи конкурента. Численные эксперименты показали, что новый метод заметно быстрее своих предшественников находит известные оптимальные решения лидера. Если же время решения задачи ограничено, то частота получения оптимума остаётся достаточно большой, а средняя относительная погрешность получаемых решений оказывается незначительной.

Т а б л и ц а 5

Результаты для класса *Uniform*

Код примера	Наилучшее известное	Лучшее найденное	Попаданий в наилучшее	Средняя погрешность (%)
123	5009	4904	0	2,645
223	5459	5459	9	0,016
323	5019	5019	1	1,030
423	4908	4846	0	1,344
523	5208	5206	0	0,412
623	5032	5032	9	0,196
723	5055	5003	0	1,776
823	4951	4951	1	1,874
923	5127	5127	5	0,653
1023	5084	5025	0	1,465

ЛИТЕРАТУРА

1. Алексеева Е. В., Кочетова Н. А. Верхние и нижние границы для конкурентной задачи о p -медиане // Тр. XIV-й Байкальской междунар. школы-семинара «Методы оптимизации и их приложения». Т. 1. — Иркутск: РИО ИДСТУ СО РАН, 2008. — С. 563–569.
2. Гончаров Е. Н., Кочетов Ю. А. Вероятностный поиск с запретами для дискретных задач безусловной оптимизации // Дискрет. анализ и исслед. операций. Сер. 2. — 2002. — Т. 9, № 2. — С. 13–30.

3. Кочетов Ю. А., Пащенко М. Г., Плясунов А. В. О сложности локального поиска в задаче о p -медиане // Дискрет. анализ и исслед. операций. Сер. 2. — 2005. — Т. 12, № 2. — С. 44–71.
4. Alekseeva E., Kochetova N., Kochetov Yu., Plyasunov A. A hybrid memetic algorithm for the competitive p -median problem // Proc. INCOM (Moscow, Russia, June 3–5, 2009). — Moscow: ICS RAS Publ., 2009. — P. 1516–1520.
5. Alekseeva E., Kochetov Yu., Plyasunov A. Complexity of local search for the p -median problem // Eur. J. Oper. Res. — 2008. — Vol. 191. — P. 736–752.
6. Alekseeva E. V., Kochetova N. A., Kochetov Yu. A., Plyasunov A. V. A heuristic and exact methods for the discrete $(r|p)$ -centroid problem // Evolutionary computation in combinatorial optimization. Proc. 10th Eur. Conf. EvoCOP 2010 (Istanbul, Turkey, April 7–9, 2010). — Berlin: Springer-Verl., 2010. — P. 11–22. (Lect. Notes Comp. Sci.; Vol. 6022.)
7. Barahona F., Chudak F. A. Near-optimal solutions to large-scale facility location problems // Discrete Optimization. — 2005. — Vol. 2. — P. 35–50.
8. Benati S., Laporte G. Tabu search algorithms for the $(r|X_p)$ -medianoid and $(r|p)$ -centroid problems // Locat. Sci. — 1994. — Vol. 2, N 2. — P. 193–204.
9. Bhadury J., Eiselt Y., Jaramillo J. An alternating heuristic for medianoid and centroid problems in the plane // Comput. Oper. Res. — 2003. — Vol. 30. — P. 553–565.
10. Drezner T., Eiselt H. A. Consumers in competitive location models // Facility location. Applications and theory. — Berlin: Springer-Verl., 2004. — P. 151–178.
11. Frangeoni A. About Lagrangian methods in integer optimization // Ann. Oper. Res. — 2005. — Vol. 139. — P. 163–193.
12. Geoffrion A. M. Lagrangian relaxation for integer programming // Math. Program. Study, Ser. — 1974. — Vol. 2. — P. 82–114.
13. Glover F., Laguna M. Tabu search. — Boston: Kluwer Acad. Publ., 1997. — 408 p.
14. Guinard M. Lagrangian relaxation // TOP. — 2003. — Vol. 11. — P. 215–228.
15. Hakimi S. L. Locations with spatial interactions: competitive locations and games // Discrete Locat. Theory. — New York: Wiley, 1990. — P. 439–478.
16. Held M., Wolfe P., Crowder H. P. Validation of subgradient optimization // Math. Program. — 1974. — Vol. 6, N 1. — P. 62–88.
17. Kochetov Yu., Kononov A., Plyasunov A. Competitive facility location models // Comput. Math. Math. Physics. — 2009. — Vol. 49, N 6. — P. 994–1009.
18. Lemaréchal C. Lagrangian relaxation // Computational combinatorial optimization. — Heidelberg: Springer-Verl., 2001. — P. 115–160.
19. Noltemeier H., Spoerhase J., Wirth H.-C. Multiple voting location and single voting location on trees // Eur. J. Oper. Res. — 2007. — Vol. 181. —

P. 654–667.

- 20. Poljak B. T.** Subgradient methods: a survey of Soviet research // Nonsmooth optimization. IIASA Proc. Ser. — 1977. — Vol. 3. — P. 5–30.
- 21. Shor N. Z.** Minimization methods for non-differentiable functions. — New York: Springer-Verl., 1985. — 162 p.

Давыдов Иван Александрович
e-mail: iadavydov@math.nsc.ru

Статья поступила
31 марта 2011 г.

Переработанный вариант —
26 декабря 2011 г.