Математические структуры и моделирование
2016. № 2(38). С. 43–59

УДК 519.16

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ ВЫБОРА ХАБА С ЗАДАННЫМ ЧИСЛОМ УЗЛОВ

А.В. Еремеев[1]
доцент, д.ф.-м.н., e-mail: eremeev@ofim.oscsbras.ru
Э.А. Тарасенко[2]
магистрант, e-mail: elfirat@yandex.ru

[1]Институт математики им. С. Л. Соболева Сибирского отделения РАН

[2]Омский государственный университет им Ф. М. Достоевского

Аннотация. Рассматривается задача выбора узлов хаба для рынка электроэнергии. Подмножество узлов электроэнергетической сети (хаб) используется для вычисления ценового индекса, который в каждый момент времени полагается равным среднему арифметическому цен электроэнергии по всем входящим в хаб узлам. В рассматриваемой задаче при известных значениях цен электроэнергии для каждого узла и каждого субъекта рынка за некоторый достаточно продолжительный период требуется выбрать хаб, состоящий из заданного числа узлов, с целью минимизации суммарного взвешенного среднеквадратического отклонения цен субъектов от ценового индекса хаба. Ввиду вычислительной сложности задачи для её решения предложены эвристические алгоритмы: метод локального поиска, генетический и гибридный генетический алгоритмы. Разработаны модификации предложенных алгоритмов для практически значимого варианта задачи с неопределёнными ценами в некоторые часы. Проведено тестирование алгоритмов на реальных данных и их сравнение с методами частично целочисленного программирования. Вычислительные эксперименты показали перспективность использования предложенных алгоритмов и выявили случаи, когда одни алгоритмы имеют преимущество перед другими.

Ключевые слова: поиск подмножества векторов, кластерный анализ, генетический алгоритм, локальный поиск, частично целочисленное программирование.

Введение

В современных рынках электроэнергии, основанных на узловом маржинальном ценообразовании, цена на электроэнергию не определяется единым значением, но зависит от узла электроэнергетической сети и периода времени [1]. При этом точное прогнозирование цены весьма проблематично, и участники рынка (оптовые продавцы и покупатели электроэнергии) заинтересованы в использовании *ценовых индексов* с целью снижения своих финансовых рисков посредством заключения фьючерсных контрактов по цене этих индексов. Подмножество узлов электроэнергетической сети, называемое *хабом*, служит для вычисления ценового индекса, аппроксимирующего в некотором смысле цены электроэнергии субъектов рынка. Индекс хаба в каждый момент времени полагается равным среднему арифметическому цен электроэнергии по всем входящим в хаб узлам. Использование индекса хаба позволяет субъектам рынка заключать фьючерсные контракты по цене электроэнергии в хабе с целью снижения финансовых рисков, связанных с колебаниями цен на электроэнергию. При выделении хаба заданной мощности из найденного на предварительном этапе множества узлов (кластера) возникает задача выбора узлов хаба.

В работе предложены и реализованы алгоритм локального поиска, генетический и гибридный генетический алгоритм для задачи выбора узлов хаба. Предложена математическая модель для варианта задачи с неопределёнными ценами в некоторые часы. Проведён вычислительный эксперимент на задачах различной размерности, в ходе которого проведено сравнение результатов работы эвристических алгоритмов и пакета CPLEX 11.0.

1. Постановки задачи выбора узлов хаба и их свойства

В [4, 5] сформулирована задача выделения заданного числа хабов из всего множества узлов энергетической сети как невыпуклая задача математического программирования. Данная модель является весьма сложной для стандартных методов невыпуклого программирования, поэтому вместо поиска ее точного решения может использоваться двухэтапный подход [4, 5]: сначала с помощью методов кластерного анализа все множество узлов разбивается на кластеры, в которых узлы имеют «близкие» цены, а затем в каждом из кластеров выделяются хабы. Последняя задача является предметом исследования в настоящей работе.

1.1. Задача выбора узлов хаба

Рассмотрим математическую модель задачи выбора хаба с заданным числом узлов из множества узлов кластера. С учётом специфики рынка электроэнергии России будем полагать, что требуется минимизировать суммарное взвешенное квадратичное отклонение индекса хаба от цен субъектов рынка. Пусть цены электроэнергии для субъектов известны за весь анализируемый период (например, за прошедший год или несколько лет). Число узлов искомого хаба задано. Этот параметр выбирается достаточно большим, чтобы обеспечить предсказуемость динамики индекса хаба и ослабить влияние на него возможных удалений отдельных узлов из системы (на практике размер хабов, как правило, составляет несколько десятков или сотен узлов).

Параметрами модели являются следующие величины:
n — число узлов заданного кластера;
m — число субъектов рынка, для которых выбирается хаб;
N — число узлов искомого хаба;

T — число часов в анализируемом периоде;
c_{it} — цена i-го узла в час t ($i = 1, \ldots, n$, $t = 1, \ldots, T$);
$w_{jt} \geqslant 0$ — суммарный объём покупки или продажи электроэнергии субъекта j ($j = 1, \ldots, m$), характеризующий значимость отклонения его цены от индекса хаба в момент времени t;
p_{jt} — цена электроэнергии, по которой её покупает или продаёт j-й субъект рынка в час t ($j = 1, \ldots, m$, $t = 1, \ldots, T$).

Переменные задачи:
$x_i = 1$, если i-й узел включается в хаб; иначе $x_i = 0$ ($i = 1, \ldots, n$);
c_t — ценовой индекс хаба в час t ($t = 1, \ldots, T$).

Требуется минимизировать суммарное квадратичное отклонение индекса хаба от цен электроэнергии субъектов рынка с учётом весовых коэффициентов. Математическая модель выглядит следующим образом [4]:

$$f = \sum_{t=1}^{T}\sum_{j=1}^{m}(c_t - p_{jt})^2 w_{jt} \to \min; \tag{1}$$

$$c_t = \frac{1}{N}\sum_{i=1}^{n} x_i c_{it},\ t = 1, \ldots, T; \tag{2}$$

$$\sum_{i=1}^{n} x_i = N; \tag{3}$$

$$x_i \in \{0, 1\},\ i = 1, \ldots N. \tag{4}$$

Запишем (1)–(4) в виде модели частично целочисленного линейного программирования [4]. Введём обозначение: $\Phi_{jt} = (c_t - p_{jt})^2$. Тогда

$$\Phi_{jt} = \left(\frac{1}{N}\sum_{i=1}^{n} x_i c_{it} - p_{jt}\right)^2. \tag{5}$$

Раскрывая скобки, получим:

$$\Phi_{jt} = \frac{1}{N^2}\sum_{i=1}^{n}(c_{it}^2 - 2Nc_{it}p_{jt})x_i + \frac{2}{N^2}\sum_{k=1}^{n}\sum_{l=1}^{k-1} c_{kt}c_{lt}x_k x_l + p_{jt}^2. \tag{6}$$

Введём переменные z_{kl}, $k = 1, \ldots, n$, $l = 1, \ldots, k-1$, подчинённые условиям

$$z_{kl} = x_k x_l,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1. \tag{7}$$

Последнее равенство можно заменить на систему линейных ограничений:

$$z_{kl} \leqslant x_k,\ z_{kl} \leqslant x_l,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1; \tag{8}$$

$$z_{kl} \geqslant x_k + x_l - 1,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1. \tag{9}$$

Запишем (1)–(4) с учётом (5)–(9).

$$C_0 + \sum_{i=1}^{n} C_i x_i + \sum_{k=1}^{n} \sum_{l=1}^{k-1} B_{kl} z_{kl} \to \min; \tag{10}$$

$$z_{kl} \geqslant x_k + x_l - 1,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1; \tag{11}$$

$$\sum_{i=1}^{n} x_i = N; \tag{12}$$

$$z_{kl} \leqslant x_l,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1; \tag{13}$$

$$x_i \in \{0, 1\},\ i = 1, \ldots, n; \tag{14}$$

$$z_{kl} \in \{0, 1\},\ k = 1, \ldots, n,\ l = 1, \ldots, k-1. \tag{15}$$

Для целевой функции (10) ограничения (8) реализуются автоматически. Поэтому можно оставить только условия (9). Рассматривая z_{kl} как непрерывные переменные, получим модель частично целочисленного линейного программирования:

$$C_0 + \sum_{i=1}^{n} C_i x_i + \sum_{k=1}^{n} \sum_{l=1}^{k-1} B_{kl} z_{kl} \to \min; \tag{16}$$

$$z_{kl} \geqslant x_k + x_l - 1,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1; \tag{17}$$

$$\sum_{i=1}^{n} x_i = N; \tag{18}$$

$$x_i \in \{0, 1\},\ i = 1, \ldots, n; \tag{19}$$

$$z_{kl} \geqslant 0,\ k = 1, \ldots, n,\ l = 1, \ldots, k-1. \tag{20}$$

Из (6) следует, что коэффициенты в целевой функции имеют вид:

$$C_0 = \sum_{j=1}^{m} \sum_{t=1}^{T} p_{jt}^2 w_{jt},$$

$$C_i = \frac{1}{N^2} \sum_{j=1}^{m} \sum_{t=1}^{T} (c_{it}^2 - 2N c_{it} p_{jt}) w_{jt},$$

$$B_{kl} = \frac{2}{N^2} \sum_{j=1}^{m} \sum_{t=1}^{T} c_{kt} c_{lt} w_{jt}.$$

1.2. Вычислительная сложность задачи выбора узлов хаба

В [3] установлено, что задача выбора узлов хаба является NP-трудной в сильном смысле. Этот же вывод может быть получен исходя из связи рассматриваемой задачи с задачей выбора подмножества векторов с кратчайшим средним при ограничении на мощность (Subset with the Shortest Average with cardinality restriction, SSA) [2]. Последняя формулируется следующим образом.

Дано: множество $\mathcal{Y} = \{y_1, \ldots, y_n\}$ точек (векторов) из $\mathbb{R}^T$ и натуральное число M.

Найти: подмножество $\mathcal{C} \subseteq \mathcal{Y}$ такое, что $|\mathcal{C}| \geqslant M$ и

$$\frac{1}{|\mathcal{C}|}\|\sum_{y\in\mathcal{C}} y\| \to \min .$$

Любая индивидуальная задача SSA сводится к последовательности из M индивидуальных задач выбора узлов хаба с теми же значениями n и T, что и в SSA, с узловыми ценами $c_{it} = y_{it},\ t = 1, \ldots, T$, с одним субъектом рынка, имеющим цены $p_{1t} = 0,\ t = 1, \ldots, T$ и $N = 1, \ldots, M$. Поскольку, как показано в [2], для задачи SSA не существует приближенных алгоритмов с какой-либо гарантированной оценкой относительной погрешности при P$\neq$NP, то такое же утверждение справедливо и для задачи выбора узлов хаба.

1.3. Задача выбора узлов хаба при неопределённых ценах в некоторые часы

На практике исходные данные не всегда содержат информацию о ценах в узлах и ценах субъектов рынка для *каждого* часа. Постановка, рассматриваемая в данном параграфе, отличается от предыдущей тем, что она учитывает возможность отсутствия в некоторые часы данных о ценах в узлах и ценах субъектов.

Заданы множества:

$\mathcal{N}_t$ — множество узлов кластера, определённых в час t, $|\mathcal{N}_t| = n_t$;

$\mathcal{T}_j$ — множество часов, когда определена цена электроэнергии субъекта j, $T_j = |\mathcal{T}_j|$.

Требуется минимизировать суммарное квадратичное отклонение индекса хаба от всех цен покупки или продажи электроэнергии по субъектам рынка с учётом весовых коэффициентов. Построенная модель выглядит следующим образом:

$$\sum_{j=1}^{m}\sum_{t\in\mathcal{T}_j}(c_t - p_{jt})^2 w_{jt} \to \min; \tag{21}$$

$$c_t = \frac{\sum\limits_{i\in\mathcal{N}_t} x_i c_{it}}{\sum\limits_{i\in\mathcal{N}_t} x_i},\ t = 1, \ldots, T; \tag{22}$$

$$\sum_{i=1}^{n} x_i = N; \tag{23}$$

$$x_i \in \{0, 1\},\ i = 1, \ldots, n; \tag{24}$$

$$c_t \geqslant 0,\ t = 1, \ldots, T. \tag{25}$$

В отличие от моделей, рассмотренных в п 1.1 и п. 1.2, данная модель содержит дробно-линейные ограничения, что существенно осложняет её решение, однако, для практики эта постановка задачи имеет большое значение, поскольку данные о ценах узлов и субъектов по техническим причинам не всегда определены. Для решения поставленной задачи целесообразно применять эвристические алгоритмы.

2. Эвристические алгоритмы решения задачи выбора узлов хаба

Ввиду того, что задача выбора узлов хаба является NP-трудной в сильном смысле и неаппроксимируемой при P≠NP, для её решения целесообразно использовать эвристические алгоритмы, показавшие эффективность для решения задач с большой вычислительной сложностью. В настоящем разделе для решения задачи выбора узлов хаба (1)–(4) и варианта задачи с неопределёнными ценами в некоторые часы (21)–(25) предложены алгоритм локального поиска, генетический алгоритм и комбинация двух этих методов.

2.1. Алгоритм локального поиска

Пусть D — множество допустимых решений. Окрестностью $\Psi(x_0) \subseteq D$ решения x_0 называется множество допустимых решений, в некотором смысле близких к данному. Идея алгоритма локального поиска состоит в следующем. Имеется допустимое решение задачи x. Строится окрестность $\Psi(x)$. Из окрестности выбирается решение x', дающее улучшение целевой функции. Различают несколько вариантов улучшения текущего решения: из окрестности может выбираться решение, дающее лучшее значение целевой функции, либо из окрестности выбирается первое улучшение целевой функции. После перехода в решение x' строится окрестность $\Psi(x')$, и процесс продолжается. Если улучшить текущее решение не удаётся, то оно является локальным оптимумом.

Рассмотрим алгоритм локального поиска для задачи выбора узлов хаба. Расстоянием Хэмминга $h(x, x')$ для булевых векторов x и x' называется число несовпадающих компонент. Окрестностью k-*swap* вектора x называется множество таких точек x', что $h(x, x') = 2k$, где $h(x, x')$ — расстояние Хэмминга для булевых векторов x и x'. Рассмотрим окрестность 1-*swap*, в эту окрестность включаются те решения, которые получаются из текущего исключением из хаба одного узла и включением другого узла. Допустимым решением будет булев вектор: $\mathbf{x} = (x_1, \ldots, x_n)$ такой, что $\sum\limits_{i=1}^{n} x_i = N$, где x_i=1, если узел i включили в хаб. В предложенном алгоритме в окрестности текущего решения выбирается решение, дающее наибольшее улучшение целевой функции.

При переходе от текущего решения $\mathbf{x}$ к новому допустимому решению $\mathbf{x}'$ используется быстрый пересчёт значения целевой функции для задачи (1)–(4):

$$F(\mathbf{x}') = F(\mathbf{x}) - C_i + C_j - \sum_{u:x_u=1} B_{ui} + \sum_{u:x'_u=1} B_{uj}.$$

В случае задачи (21)–(25) значение функции $F(\mathbf{x}')$ вычисляется следующим образом:

$$F(\mathbf{x}') = \sum_{t=1}^{T}\sum_{j=1}^{m}(c_t(\mathbf{x}') - p_{jt})^2 w_{jt},$$

где

$$c_t(\mathbf{x}') = \frac{c_t(\mathbf{x}') - c_{it} + c_{jt}}{\sum\limits_{i\in\mathcal{N}_t} x'_i}.$$

2.2. Генетический алгоритм

Генетический алгоритм (ГА) представляет собой метод оптимизации, основанный на эволюционных принципах. Рассмотрим задачу оптимизации в общем виде:

$$\max\{f(x) : x \in X\},$$

где X — пространство решений.

Популяцией $\Pi = (\xi^1, \xi^2, \ldots, \xi^l)$ численности l называется вектор пространства $B^l, B = \{0, 1\}^u$, координаты которого называются генотипами особей. Целевая функция f исходной задачи заменяется в генетическом алгоритме на функцию приспособленности генотипа F. С помощью данной функции можно задать для каждой особи среднее число оставляемых ею потомков или вероятность того, что эта особь оставит потомка. Алгоритм начинает свою работу с некоторой начальной популяции Π_0. Шагом генетического алгоритма является переход от текущей популяции к следующей, то есть получение новой популяции Π^{t+1} из Π^t. В генетическом алгоритме новые решения строятся как комбинации элементов решений текущей популяции с использованием операторов селекции, скрещивания (кроссинговера) и мутации.

Оператором селекции $Sel : B^l \to \{1, \ldots, l\}$ называется одноместный оператор, действие которого заключается в выборе номера родителя в популяции. Оператор селекции особей на пространстве популяций имеет то же значение, что и естественный отбор в природе.

Оператором скрещивания $Cross : B \times B \to B \times B$ называется двуместный оператор, действие которого заключается в построении генотипов потомков путём комбинации участков родительских генопипов.

Широко используемый одноточечный оператор кроссинговера [6] действует следующим образом. Пусть генотипы родителей имеют вид:

$$\xi = (\xi_1, \xi_2, \ldots, \xi_u),$$

$$\eta = (\eta_1, \eta_2, \ldots, \eta_u),$$

тогда результат кросинговера $(\xi', \eta') = Cross(\xi, \eta)$ формируется в виде:

$$\xi' = (\xi_1, \xi_2, \ldots, \xi_\chi, \eta_{\chi+1}, \ldots, \eta_u),$$

$$\eta' = (\eta_1, \eta_2, \ldots, \eta_\chi, \xi_{\chi+1}, \ldots, \xi_u),$$

где случайная позиция χ выбрана с равномерным распределением от 1 до $u-1$. Оператор кроссинговера применяется в алгоритме с вероятностью P_{cross}. В случае, когда оператор кроссинговера не применяется, родительские генотипы копируются в новую популяцию.

Оператором мутации называется одноместный оператор $Mut : B \to B$, который вносит случайные изменения в генотип особи. Оператор мутации [6] в каждой позиции аргумента с заданной вероятностью P_{mut} заменяет её содержимое на другой элемент алфавита {0, 1}.

Предлагаемый генетический алгоритм для решения задачи выбора узлов хаба описывается следующим образом. Генотипом является вектор $\mathbf{x} \in B^n$, причём $\sum\limits_{i=1}^{n} x_i = N$. Популяцией численности e является вектор $\Pi = \{x_1, \ldots, x_e\}$. Функция приспособленности генотипа

$$F(c_t) = -f(c_t) = -\sum_{t=1}^{T}\sum_{j=1}^{m}(c_t - p_{jt})^2 w_{jt}.$$

Генетический алгоритм

1. Случайным образом строится начальная популяция Π_0.

2. Пока не превышено максимальное число итераций, выполняется:

    2.1. Оператор селекции Sel'. Из текущей популяции выбираются k особей с равномерным распределением, из них выбираются две особи ξ и η с наибольшим значением функции приспособленности.

    2.2. Оператор скрещивания $Cross'$. Два потомка ξ' и η' строятся с помощью оператора кроссинговера, описываемого ниже.

    2.3. Оператор мутации Mut'. К потомкам ξ' и η' применяется оператор мутации с вероятностью P_{mut}, который также описан ниже.

    2.4. Из двух особей потомков ξ' и η' и двух особей родителей ξ и η выбираются две особи с наибольшим значением функции приспособленности и помещаются в популяцию Π'.

Рассмотрим оператор кроссинговера $Cross'$, предложенный для рассматриваемой задачи. Особенностью данного оператора является то, что потомки должны наследовать свойство родителей $\sum\limits_{i=1}^{n} x_i = N$. Сохранение этого свойства обеспечивается следующим правилом формирования потомков:
1) если в потомке, полученном в результате применения оператора кроссинговера, описанного в п. 2.2.1, имеется N' единиц, причём $N' > N$, то из множества позиций, содержащих единицы, с равномерным распределением выбираются $N' - N$ позиций, которые заполняются нулями;
2) если в потомке, полученном в результате применения оператора кроссинговера, имеется N' единиц, причём $N' < N$, то из позиций, в которых стоят

единицы в родительских особях, а в потомках стоят нули, случайным образом выбираются $N - N'$ позиций, которые заменяются единицами.

Рассмотрим оператор мутации Mut', применяемый в предложенном алгоритме. Отличие данного оператора мутации от оператора Mut классического генетического алгоритма состоит в том, что сохраняется свойство родительских решений $\sum\limits_{i=1}^{n} x_i = N$. Для этого каждый элемент вектора с вероятностью P_{mut} заменяется на противоположный, при этом подсчитывается $Q_{1\to 0}$ — количество замен 1 на 0 и $Q_{0\to 1}$ — количество замен 0 на 1, затем $|Q_{1\to 0} - Q_{0\to 1}|$ раз выполняются обратные замены для $|Q_{1\to 0} - Q_{0\to 1}|$ случайно выбранных позиций из числа тех, которые подвергались замене.
Рассмотрим вычисление функции приспособленности генотипа для задач (1)–(4) и (21)–(25). Пусть $\mathbf{x}$ — генотип, для которого известно значение функции приспособленности $F(\mathbf{x})$, значение индекса хаба $c_t(\mathbf{x})$ и число узлов хаба $N_t(\mathbf{x}), t = 1, \ldots, T$. Вычисляем $F(\mathbf{x}')$ для задачи (1)–(4) следующим образом:

$$c_t(\mathbf{x}') = \frac{1}{N}\left(c_t(\mathbf{x}) + \sum_{i\in I_1} c_{it} - \sum_{i\in I_{-1}} c_{it}\right),$$

$$F(\mathbf{x}') = \sum_{t=1}^{T}\sum_{j=1}^{m}\left(c_t(\mathbf{x}') - p_{jt}\right)^2 w_{jt}.$$

Для задачи (21)–(25) вычисление $F(\mathbf{x}')$ происходит следующим образом:

$$c_t(\mathbf{x}') = \frac{c_t(\mathbf{x}) + \sum\limits_{i\in I_1} c_{it} - \sum\limits_{i\in I_{-1}} c_{it}}{N_t(\mathbf{x}) + \sum_{i=1}^{n}(x_i - x_i')},$$

$$F(\mathbf{x}') = \sum_{t=1}^{T}\sum_{j=1}^{m}(c_t(\mathbf{x}') - p_{jt})^2 w_{jt},$$

где $I_1 = \{i : x_i - x_i' = 1\}, I_{-1} = \{i : x_i - x_i' = -1\}$.

2.3. Комбинация генетического алгоритма с алгоритмом локального поиска

Для решения задачи выбора узлов хаба (1)–(4) и варианта задачи с неопределёнными ценами в некоторые часы (21)–(25) был также предложен гибридный генетический алгоритм. Идея гибридных алгоритмов заключается в сочетании генетического алгоритма с некоторым другим алгоритмом решения, в котором учитывается специфика задачи. В данном случае в качестве второго метода использовался алгоритм локального поиска. На каждом поколении полученный потомок оптимизировался этим методом, после чего добавлялся в популяцию.

Рассмотрим схему гибридного генетического алгоритма для задачи выбора узлов хаба (1)–(4) и варианта задачи с неопределёнными ценами в некоторые часы (21)–(25). Для каждой из двух полученных особей на итерации генетического алгоритма применяется алгоритм локального поиска, полученные локальные оптимумы помещаются в популяцию Π'. Данная схема в англоязычной литературе носит название «memetic algorithm».

3. Вычислительный эксперимент

В настоящем разделе приведены результаты тестирования предложенных выше алгоритмов для задачи выбора узлов хаба. Для проведения экспериментов алгоритмы были реализованы на языке C++ и тестировались на ЭВМ Athlon II X4 с тактовой частотой процессора 2,9 ГГц и объёмом оперативной памяти 2 Гб. В качестве тестовых примеров использовались задачи, построенные с использованием исходных данных о ценах электроэнергии «PJM Interconnection» (США), доступных на сайте `http://www.pjm.com`, а также реальные данные рынка электроэнергии России.

Для решения задачи выбора узлов хаба (1)–(4) и варианта задачи с неопределёнными ценами в некоторые часы (21)–(25) реализованы алгоритм локального поиска, генетический алгоритм и гибридный генетический алгоритм.

В качестве тестовых использовались задачи, представленные в табл. 1. В данной таблице содержится описание параметров задач, таких как число узлов заданного кластера n, число субъектов рынка m, число узлов хаба N, число часов T, также в последнем столбце обозначена принадлежность соответствующего кластера США или России. Задачи 1–8 представляют данные для хабов США и содержат полную информацию о ценах в узлах и ценах в субъектах рынка, они соответствуют модели (1)–(4). Задачи 9–14 представляют данные из европейской и сибирской ценовых зон соответственно, в некоторые часы данные не содержат информацию о ценах в узлах или в субъектах, эти задачи соответствуют модели (21)–(25).

3.1. Выбор параметров генетического алгоритма

Процедура подбора параметров генетического алгоритма решения задачи выбора узлов хаба (1)–(4) проводилась по следующей схеме.

Этап 1. Подбор вероятности мутации P_{mut} с шагом 0.1 (таким образом рассматриваются все варианты $P_{mut} = 0, 0.1, \ldots, 1$), при этом вероятность запуска оператора кроссинговера $P_{cross} = 0$.

Этап 2. Подбор размера популяции. Рассматриваются варианты: 50, 100, 200, 300, 400, 500. При этом вероятность мутации фиксирована и равна лучшему значению параметра, полученного на этапе 1.

Этап 3. Подбор размера турнира k. Осуществляется путём запуска алгоритма с числом участников турнира $k = 2, 5, 20$.

На этапе 1 для задачи (1)–(4) относительное отклонение от лучшего известного значения функции оказалось наименьшим при вероятности мутации $P_{mut} = 0.1$. На этапе 2 лучшие значения целевой функции для всех рассматриваемых задач получены при размере популяции равном 100. Исключение со-

Таблица 1. Параметры тестовых примеров

№	n	m	N	T	Географическое расположение
1	43	5	20	24	США, PJM
2	152	10	20	24	США, PJM
3	199	20	10	24	США, PJM
4	199	30	100	24	США, PJM
5	233	100	5	24	США, PJM
6	408	5	100	24	США, PJM
7	642	30	10	24	США, PJM
8	642	5	100	24	США, PJM
9	63	204	50	20472	Россия (европейская ценовая зона)
10	95	181	76	20472	Россия (европейская ценовая зона)
11	259	266	207	20472	Россия (европейская ценовая зона)
12	411	656	330	20472	Россия (европейская ценовая зона)
13	49	52	40	20472	Россия (сибирская ценовая зона)
14	80	52	64	20472	Россия (сибирская ценовая зона)

ставляла задача 6, для которой лучшие результаты были получены при размере популяции равном 50. На этапе 3 лучшее значение целевой функции для всех рассматриваемых задач получено при $k = 20$. Процедура подбора параметров генетического алгоритма для варианта задачи с неопределёнными ценами в некоторые часы дала аналогичные результаты.

3.2. Решение задач выбора узлов хаба

Рассмотрим решение задачи выбора узлов хаба (1)–(4). Данная задача была решена с помощью алгоритма локального поиска, генетического алгоритма, а также гибридного генетического алгоритма. Для каждого набора тестовых данных 1–8 алгоритм локального поиска применялся 20 раз, в качестве результата выбиралось лучшее из 20 решений.

Для гибридного генетического алгоритма применены те же настройки параметров, что и для генетического алгоритма. Для решения задач генетическим алгоритмом и гибридным генетическим алгоритмом задавалось максимальное время решения, равное времени решения задачи алгоритмом локального поиска. Результаты эксперимента для всех рассмотренных алгоритмов представлены для задач 1–8 в табл. 2. В таблице указано число узлов заданного множества, число субъектов рынка, число узлов хаба, а также полученные рассматриваемыми алгоритмами значения целевой функции и время счёта τ в секундах. Лучшие известные значения целевой функции в этой и последующих таблица обозначены жирным шрифтом.

Таблица 2. Результаты эксперимента по сравнению эвристических алгоритмов

№	n	m	N	**Локальный поиск**		**Генетический алгоритм**	**Гибридный алгоритм**
				f	τ, сек.	f	f
1	43	5	20	**4.993**	1	**4.993**	**4.993**
2	152	10	20	0.1857	1	0.1857	0.1857
3	199	20	10	**0.0478**	3	**0.0478**	**0.0478**
4	199	30	100	3.338	21	3.338	3.338
5	233	100	5	**61.057**	4	61.123	**61.057**
6	408	5	100	1315.440	309	1315.869	1315.468
7	642	30	10	4413.308	17	**4413.279**	4413.317
8	642	5	100	**0.0049**	582	0.0062	0.0052

Из представленной таблицы видно, что рассматриваемые алгоритмы получали одинаковое значение целевой функции для задач 1–4. На задачах 5, 6, 8 гибридный генетический алгоритм получал значения целевой функции меньше, чем генетический алгоритм за то же самое время, кроме того на этих задачах алгоритм локального поиска получал значение целевой функции меньше, чем гибридный генетический алгоритм. Значение функции, полученное генетическим алгоритмом для задачи 7 меньше, чем значение функции, полученное алгоритмом локального поиска и гибридным генетическим алгоритмом. Результаты проведённого вычислительного эксперимента позволяют сделать вывод, что алгоритм локального поиска наряду с гибридным генетическим алгоритмом успешно решают задачи малой размерности.

Для сравнения полученных результатов работы алгоритма локального поиска с коммерческим пакетом прикладных программ была составлена модель частично целочисленного линейного программирования (10)–(15) с помощью пакета GAMS 22.6, а также модель квадратичного программирования (1)–(4). Все тестовые задачи решены с помощью пакета CPLEX 11.0, входящего в состав GAMS. При решении задач квадратичного частично целочисленного программирования использовался параметр CPLEX `miqcp=1` (решение задачи квадратичного программирования в каждом узле дерева ветвей и границ).

Решение задач с помощью пакета CPLEX 11.0 для каждой модели было проведено в два этапа:

1. Задавалось максимальное время решения, равное времени решения задачи алгоритмом локального поиска ($\tau_{max} = \tau_L$).

2. Задавалось максимальное время решения, равное 30 минутам. В обоих случаях максимальное число итераций задавалось равным 10^8, но в ходе тестирования предел числа итераций не превышался.

Результаты экспериментов приведены в сводных таблицах (в табл. 3 — результаты сравнения алгоритма локального поиска и пакета CPLEX в режиме частично целочисленного линейного программирования, в табл. 4 — того же алгоритма и пакета CPLEX в режиме квадратичного программирования). В обеих таблицах для каждой задачи указано число узлов заданного множества, число субъектов рынка, число узлов в хабе, а также полученные алгоритмом локального поиска и пакетом CPLEX 11.0 значения целевой функции и время счета τ в секундах. Символ «*» обозначает оптимальные решения. В скобках указано время доказательства оптимальности пакетом CPLEX.

Таблица 3. Сравнение алгоритма локального поиска и пакета CPLEX в режиме частично целочисленного линейного программирования

№	n	m	N	**Локальный поиск**		**CPLEX 11.0**			
				f	τ, сек.	$\tau_{max} = \tau_L$		$\tau_{max} = 1800c$	
						f	число итер.	f	число итер.
1	43	5	20	4.993	1	20.337	3031	7.711	974854
2	152	10	20	0.1857	1	0.1874	412	0.1858	604440
3	199	20	10	0.0478	3	0.051	251	0.048	453946
4	199	30	100	3.338	21	3.478	1874	**3.334**	32189
5	233	100	5	**61.057***	4	78.339	59	61.939	4396
6	408	5	100	**1315.4***	309	3290.99	7789	3290.98	19555
7	642	30	10	4413.3	17	5245.243	96	5245.242	2011
8	642	5	100	0.0049	582	11.972	4396	0.0109	7234

Как видно из табл. 3, на первом этапе CPLEX 11.0 во всех задачах получал решение со значением целевой функции, большим, чем алгоритм локального поиска. На втором этапе CPLEX 11.0 за время $\tau = 30$ мин. не доказал оптимальность решения для задач 1–8 и получал значение целевой функции, большее, чем алгоритм локального поиска.

Как видно из табл. 4, для задач 1, 5, 6, 8 с помощью пакета CPLEX 11.0 установлена оптимальность найденных решений. В задачах 5, 6 алгоритм локального поиска получил оптимальные решения. В задачах 2, 3, 5, 7 значения целевой функции, полученные на первом этапе пакетом CPLEX, оказались больше значений, полученных алгоритмом локального поиска.

Проведённое сравнение алгоритма локального поиска и пакета CPLEX показывает, что алгоритм локального поиска, наряду с пакетом CPLEX в режиме квадратичного программирования, может эффективно применяться для решения задачи выбора узлов хаба. Однако использование коммерческого пакета

Таблица 4. Сравнение алгоритма локального поиска и пакета CPLEX в режиме квадратичного программирования

№	n	m	N	**Локальный поиск**		**CPLEX 11.0**	
				f	τ	$\tau_{max} = \tau_L$	$\tau_{max} = 1800c$
						f	f
1	43	5	20	4.993	1	**4.992*** (1 с)	**4.992*** (1 с)
2	152	10	20	0.1857	1	0.1858	**0.1856**
3	199	20	10	0.0478	3	0.0479	0.0478
4	199	30	100	3.338	21	3.338	3.338
5	233	100	5	**61.057***	4	61.085	**61.057*** (13 с)
6	408	5	100	**1315.4***	309	**1315.4*** (3 с)	**1315.4*** (3 с)
7	642	30	10	4413.3	17	5168.2	4413.3
8	642	5	100	0.0049	582	0.0039* (4 с)	0.0039* (4 с)

CPLEX 11.0 в режиме частично целочисленного линейного программирования оказалось нецелесообразным.

3.3. Решение задач выбора узлов хаба при неопределённых ценах в некоторые часы

Для решения варианта задачи с неопределёнными ценами в некоторые часы (21)–(25) реализован алгоритм локального поиска, генетический алгоритм, а также гибридный генетический алгоритм, описанные в разделе 2. В качестве данных о ценах в узлах, ценах субъектов рынка и объёмах покупки или продажи электроэнергии используются реальные данные рынка электроэнергии России. В качестве тестовых задач используются четыре кластера из европейской ценовой зоны (задачи 11–14) и два кластера из сибирской ценовой зоны (задачи 9, 10) — см. `http://www.mosenex.ru/`. Алгоритм локального поиска запускается один раз, число итераций генетического алгоритма равно 2000. Максимальное время работы гибридного генетического алгоритма задаётся равным времени решения задачи генетическим алгоритмом. Локальный поиск в гибридном алгоритме запускается с вероятностью $P_{local} = 0.2$. Результаты экспериментов представлены в табл. 5, которая содержит информацию о числе узлов заданного множества, числе субъектов рынка, числе узлов хаба, а также полученные рассматриваемыми алгоритмами значения целевой функции и время счета τ в секундах.

Во всех задачах генетический алгоритм и алгоритм локального поиска получали близкие значения целевой функции, однако, время работы алгоритма

Таблица 5. Решение задач выбора узлов хаба при неопределённых ценах в некоторые часы

№	n	m	N	**Локальный поиск**		**Генетический алгоритм**		**Гибридный алгоритм**
				f	τ, сек.	f	τ, сек.	f
9	63	204	50	**2.5600e12**	1917	2.5601e12	2213	**2.5600e12**
10	95	181	76	**4.3826e12**	6748	4.3839e12	3239	4.3839e12
11	259	266	207	**2.3320e12**	160650	2.3343e12	18677	2.3358e12
12	411	656	330	**3.4374e12**	518400	3.4386e12	71430	3.4386e12
13	49	52	40	6.1329e11	211	6.1329e11	454	**6.1328e11**
14	80	52	64	**6.5781e11**	970	6.5842e11	721	**6.5781e11**

локального поиска даже на одной итерации оказалось больше, чем время работы генетического алгоритма, исключение составляют задачи 9, 13. Для задач 9, 14 гибридный алгоритм получал такое же значение целевой функции, что и алгоритм локального поиска, для задачи 13 гибридный алгоритм получал меньшее значение функции, чем алгоритм локального поиска.

Проведённый вычислительный эксперимент показывает, что для варианта задачи с неопределёнными ценами в некоторые часы (21)–(25) может эффективно применяться алгоритм локального поиска и гибридный генетический алгоритм для задач небольшой размерности. Для задач большой размерности целесообразно применять генетический алгоритм, в то время как применение алгоритма локального поиска или гибридного генетического алгоритма нецелесообразно в связи с большой трудоёмкостью локального поиска (быстрый пересчёт значения целевой функции в данном случае не применим).

4. Заключение

В данной работе рассмотрены две постановки задачи выбора узлов хаба, предложена математическая модель варианта задачи с неопределёнными ценами в некоторые часы (21)–(25). Предложены и реализованы алгоритм локального поиска, генетический и гибридный генетический алгоритмы. Все алгоритмы были реализованы и протестированы на задачах различной размерности. Проведено сравнение алгоритмов между собой.

Полученные результаты позволяют сделать вывод о перспективности применения предложенных эвристических алгоритмов для приближенного решения задачи выбора узлов хаба большой размерности. В случае, когда число субъектов мало, точные решения могут быть получены пакетом CPLEX в режиме квадратичного частично целочисленного программирования. Использование пакета CPLEX в режиме частично целочисленного линейного программирования представляется нецелесообразным.

Для приближённого решения задачи с неопределёнными ценами в некоторые часы целесообразно применять генетический алгоритм, в то время как использование алгоритма локального поиска или гибридного генетического алгоритма оказалось менее эффективно в связи с большой трудоёмкостью локального поиска.

Благодарности

Работа выполнена при поддержке Программы фундаментальных научных исследований государственных академий наук на 2013-2020 годы, п. I.5.1.5. «Исследование и решение задач комбинаторной оптимизации с использованием целочисленного программирования».

Литература

1. Давидсон М.Р., Догадушкина Ю.В., Крейнес Е.М., Новикова Н.М., Удальцов Ю.А., Ширяева Л.В. Математическая модель конкурентного оптового рынка электроэнергии в России // Изв. РАН. Теория и системы управления. 2004. № 3. С. 72–83.
2. Еремеев А.В., Кельманов А.В., Пяткин А.В. О сложности некоторых евклидовых задач оптимального суммирования. Доклады Академии наук. 2016. т. 468, N 4. С. 372-375.
3. Тарасенко Э.А. О сложности задачи выбора узлов хаба // Труды региональной научной студенческой конференции «Молодежь III тысячелетия». Омск : ОмГУ, 2010. С. 45–48.
4. Borisovsky P.A., Eremeev A.V., Grinkevich E.B., Klokov S.A., Vinnikov A.V. Trading Hubs Construction for Electricity Markets // In: Optimization in the Energy Industry / Kallrath J., Pardalos P.M., Rebennack S. Springer : Berlin, 2009. P. 27–51.
5. Hartshorn A., Hastings D. Midwest ISO Trading hubs development. Cambridge, MA : LECG, LLC, 2003. URL:~https://www.misoenergy.org/_layouts/miso/ecm/redirect.aspx?id=30607 (дата обращения: 12.02.2016).
6. Holland J. H. Adaptation in Natural and Artifical systems. Ann Arbor, MI : University of Michigan Press, 1975.
7. Reeves C. Genetic algorithm for the operations researcher // INFORMS Journal on Computing. 1997. V. 9, № 3. P. 231–250.

ALGORITHMS FOR SOLVING THE HUB CONSTRUCTION PROBLEM WITH GIVEN NUMBER OF HUB NODES

A.V. Eremeev[1]
Associate Professor, Dr. Sci. (Phys.-Math.), e-mail: eremeev@ofim.oscsbras.ru
E.A. Tarasenko[2]
MSci. Student, e-mail: elfirat@yandex.ru

[1]Omsk Branch of Sobolev Institute of Mathematics SB RAS
[2]Dostoevsky Omsk State University

Abstract. The trading hub construction problem for electricity markets is considered. A subset of nodes of an electricity grid (called a hub) is used for computing the price index, which at every moment is assumed to be equal to the arithmetic mean of electricity prices over all nodes of the hub. Given historical prices for all nodes and for all market participants over a sufficiently long period of time, the problem is to choose a hub consisting of a given number of nodes so as to minimize the sum of squared differences between the hub price and the prices of participants. In view of problem complexity, the heuristic algorithms are proposed for its solution: a local search method, a genetic algorithm and a memetic algorithm. Modifications of these algorithms were developed for a practically significant version of the problem where the prices for some hours are undefined. The algorithms are tested and compared to the integer programming methods on the real-life data. Computational experiments indicated prospectiveness of usage of the proposed algorithms and displayed the cases where some algorithms are in advantage to others.

Keywords: vectors subset search, cluster analysis, genetic algorithm, local search, mixed integer programming.