

УДК 621.395.7, 519.854

doi:10.15217/issn1684-8853.2015.3.99

# ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ МУЛЬТИСТАРТА И ПОИСКА С ЗАПРЕТАМИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАЗМЕЩЕНИЯ БАЗОВЫХ СТАНЦИЙ

Е. С. Скаков<sup>а</sup>, аспирантВ. Н. Малыш<sup>а</sup>, доктор техн. наук, профессор<sup>а</sup>Липецкий государственный педагогический университет, Липецк, РФ

**Постановка проблемы:** синтез топологической структуры беспроводной сети передачи данных подразумевает планирование территориального размещения базовых приемо-передающих станций на местах-кандидатах и подключение к ним клиентов. Недостатками существующих подходов к решению этой задачи являются использование методов, не показывающих высокую скорость расчета (метода ветвей и границ, эвристического метода Лагранжа и др.); отсутствие ограничений, учитывающих уровень затухания сигнала при распространении от базовой станции к клиенту и обратно, а также уровень межсотовых помех; использование всего одного типа базовых станций. Целью исследования является создание модели решения задачи размещения базовых станций, не имеющей указанных недостатков. **Результаты:** сформулирована задача размещения базовых станций с учетом уровня отношения сигнала к помехам для клиентов сети. Решение задачи представляется в виде вектора структур, каждая из которых хранит информацию об одном месте-кандидате (тип установленной базовой станции, список подключенных клиентов). Разработаны модификации алгоритмов вероятностного поиска с запретами и мултистарта, в основе которых лежит понятие окрестности текущего решения. Новое решение из окрестности текущего может быть получено при помощи одной из шести операций: смены типа одной станции на более дешевый/дорогой, переподключения одного клиента, удаления одной базовой станции, добавления одной станции, перемещения одной базовой станции. С целью избежать «застревания» в локальных оптимумах при поиске с запретами алгоритму запрещается просматривать решения из списка запретов. Новизна подхода заключается в том, что в список запретов добавляются не конкретные прошлые решения, а операции по изменению конфигурации сети, которые могут вернуть нас в старые локальные оптимумы. Сущность модифицированного алгоритма мултистарта состоит в следующем: используются всего две операции для получения нового решения (удаление базовой станции и смена типа на более дешевый), просматривается только часть окрестности, переход к новому решению осуществляется по принципу «первое улучшение», алгоритм поиска лучшего решения запускается несколько раз. Разработанные алгоритмы реализованы как программное обеспечение на языке Delphi. Показано, что новые алгоритмы демонстрируют лучшие результаты, чем метод локального поиска. **Практическая значимость:** разработанные модификации методов мултистарта и поиска с запретами позволяют находить решение задачи размещения базовых станций за приемлемое время, на много порядков быстрее точного метода полного перебора. Выявлена зависимость качества решения поставленной задачи методом вероятностного поиска с запретами от длины списка запретов и значения параметра рандомизации окрестности.

**Ключевые слова** — размещение базовых станций, алгоритм мултистарта, поиск с запретами, беспроводные сети, оптимизация, SIR.

## Введение

Как известно [1], задача проектирования сети (в том числе и беспроводной) может быть сведена к отысканию минимума функционала приведенной стоимости при наличии ограничений на вероятностно-временные и структурные характеристики сети и при требовании принадлежности множества вариантов архитектуры сети к области технически реализуемых решений.

Среди русскоязычных публикаций по проблеме размещения базовых станций (БС) отметим работу [2], посвященную сетям WiMAX. Подробная классификация алгоритмов размещения БС для сетей UMTS 3G дана в статье [3].

В данной работе сформулирована задача размещения БС, учитывающая уровень затухания сигнала при распространении от БС к клиенту и обратно, уровень межсотовых помех и наличие в сети нескольких типов БС. Поставленная задача решается метаэвристическими методами вероят-

ностного поиска с запретами (ПЗ) и мултистарта (МС), демонстрирующими высокую скорость и точность нахождения решения.

## Постановка задачи размещения БС

Задача размещения БС заключается в том, что у нас есть  $N_{tp}$  клиентов, каждого из которых необходимо подключить к БС. Базовая станция может быть установлена на одном из  $N_{ps}$  мест-кандидатов. Имеется  $N_{types}$  типов БС, отличающихся по своим характеристикам. Задача сводится к минимизации общей стоимости установленных БС при выполнении ряда ограничений.

Для описания решения задачи размещения БС нам необходимо воспользоваться такой единицей представления данных, как структура. Структура — композитный тип данных, позволяющий хранить совокупность переменных различного типа (полей), объединенных одним

именем [4]. В нашем случае структура содержит в себе три поля:

— переменную  $type$ , показывающую, какого типа БС установлена на данном месте-кандидате (если  $type = 0$ , то БС не установлена);

— переменную  $cl\_nbr$  — число клиентов, подключенных к данному месту;

— одномерный массив (вектор) целых чисел  $CL$  размерности  $cl\_nbr$ , содержащий номера клиентов, подключенных к данному месту.

Решение задачи мы будем представлять в качестве вектора  $Sol$ , элементами которого являются структуры, соответствующие местам-кандидатам (здесь и в дальнейшем запись вида  $A[j]$  означает обращение к  $j$ -му элементу вектора  $A$ ; элементы массивов нумеруются, начиная с 1). Обращение к полю обозначается при помощи символа «.». То есть  $Sol[4].type$  означает тип БС 4-го места-кандидата, а, например,  $Sol[5].CL[3]$  — номер 3-го клиента, подключенного к БС на месте № 5.

Клиент может быть подключен к месту-кандидату, только если на нем уже установлена БС, т. е. для каждого места, у которого  $Sol[s].cl\_nbr \neq 0$ , должно выполняться неравенство  $Sol[s].type \neq 0$ .

Пусть  $\mathbf{b}$  — вектор, элементами которого являются требуемые полосы пропускания для клиентов;  $\beta$  — вектор, элементами которого являются максимальные производительности БС разного типа. Для каждой установленной БС суммарный требуемый трафик не должен превосходить максимально возможную производительность оборудования:

$$\sum_{i=1}^{Sol[s].cl\_nbr} \mathbf{b}[Sol[s].CL[i]] \leq \beta[Sol[s].type] \quad \forall s \in \{1, 2, \dots, N_{ps}\}. \quad (1)$$

Пусть  $\mathbf{P}_{BS}^{\max}$  — вектор, элементами которого являются максимальные мощности БС разного типа;  $\mathbf{P}_{TP}^{\max}$  — вектор максимальных мощностей клиентов;  $\mathbf{P}_{BS}^{\text{tar}}$  — вектор, элементами которого являются чувствительности БС разного типа;  $\mathbf{P}_{TP}^{\text{tar}}$  — вектор, элементами которого являются чувствительности клиентов;  $\mathbf{G}$  — двумерный массив (матрица) размерности  $N_{tp} \times N_{ps}$ , каждый элемент которого  $0 \leq G[i][s] \leq 1$  отражает уровень затухания между  $i$ -м клиентом и местом установки  $s$ .

Несмотря на затухание сигнала на пути от БС к клиенту, мощность, доходящая от приемника к передатчику, должна превышать минимальную целевую мощность:

$$\frac{G[Sol[s].CL[i]][s] \cdot \mathbf{P}_{BS}^{\max}[Sol[s].type]}{\mathbf{P}_{TP}^{\text{tar}}[Sol[s].CL[i]]} \geq 1 \quad \forall s \in \{1, 2, \dots, N_{ps}\}, \quad \forall i \in Sol[s].CL. \quad (2)$$

Ограничение (2) составлено для режима downlink. Для режима uplink аналогичное ограничение выглядит следующим образом:

$$\frac{G[Sol[s].CL[i]][s] \cdot \mathbf{P}_{TP}^{\max}[Sol[s].CL[i]]}{\mathbf{P}_{BS}^{\text{tar}}[Sol[s].type]} \geq 1 \quad \forall s \in \{1, 2, \dots, N_{ps}\}, \quad \forall i \in Sol[s].CL. \quad (3)$$

Целевая функция, которую необходимо минимизировать, имеет вид

$$\Phi = \sum_{s=1}^{N_{ps}} \text{Cost}[Sol[s].type] + \sum_{s=1}^{N_{ps}} \sum_{i=1}^{Sol[s].cl\_nbr} SIR_{dB}^{si} \cdot k, \quad (4)$$

где  $\text{Cost}$  — вектор стоимостей (включая установку) БС разного типа.

Первое слагаемое (4) — суммарная стоимость комплекса. Второе слагаемое отвечает за учет уровня SIR для всех клиентов системы. SIR — отношение уровня сигнала к уровню помех (signal-to-interference ratio) [5], в общем случае оно рассчитывается так:

$$SIR_{dB} = 10 \lg(P_{\text{signal}}/P_{\text{interference}}). \quad (5)$$

В нашей задаче мы рассчитываем SIR отдельно для каждого клиента. В числителе фигурирует получаемый сигнал от той БС, к которой подключен клиент, в знаменателе — сигналы от остальных БС, которые создают помехи:

$$SIR_{dB}^{si} = \frac{G^{si} \cdot \mathbf{P}_{BS}^{\max}[Sol[s].type]}{\sum_{q=1}^{N_{ps}} \sum_{z=1}^{Sol[q].cl\_nbr} G^{qz} \cdot \mathbf{P}_{BS}^{\max}[Sol[q].type] \cdot w_{siqz}}, \quad (6)$$

$$G^{si} = G[Sol[s].CL[i]][s],$$

$$G^{qz} = G[Sol[q].CL[z]][q],$$

где  $w_{siqz} = 0$ , если  $s = q$  и  $i = z$ , иначе  $w_{siqz} = 1$ .

Коэффициент  $k$  обеспечивает одновременный учет в целевой функции затрат на создание сети и уровня SIR клиентов.  $k$  имеет размерность стоимости комплекса БС, тем самым обеспечивая сохранение размерности в формуле (4), так как SIR — безразмерная величина.  $k$  должен обеспечивать «штраф» за низкий уровень SIR и «награду» за высокий. В данной работе принято  $k = -10$  условных единиц стоимости ( $k$  имеет отрицательное значение, так как мы решаем минимизационную задачу, а значит, хороший высокий уровень SIR должен уменьшать целевую функцию).

Необходимо отсортировать типы БС по возрастанию цены. В дальнейшем мы будем исходить из того, что  $i$ -й тип БС дороже  $(i - 1)$ -го и дешевле  $(i + 1)$ -го.

## Метод локального поиска

Для начала мы предлагаем рассмотреть алгоритм локального поиска (ЛП), так как именно он лежит в основе большинства метаэвристических методов оптимизации. Данный алгоритм является итеративным методом. На каждом шаге рассматривается некоторое множество соседних решений — так называемая окрестность текущего решения. В качестве следующего из нее выбирается решение, доставляющее максимум оценочной функции. Процесс продолжается до тех пор, пока в окрестности имеются решения лучшие, чем текущие по отношению к оценочной функции [6].

Ниже представлен простейший алгоритм ЛП для задачи минимизации [7]. Предполагается, что вектор  $x$  — решение некоторой оптимизационной задачи. Множество всех возможных векторов обозначим  $X$ . Пусть требуется минимизировать некоторую функцию  $f(x)$  на множестве  $X$ :

1.  $x_0$  — начальное решение.
2. Текущее решение  $x = x_0$ .
3. Построить окрестность соседних решений  $N(x)$ .
4. Найти такое решение  $z \in N(x)$ , что  $f(z) \leq f(y)$  для всех  $y \in N(x)$ .
5. Положить  $x = z$ .
6. Если окрестность соседних решений не содержит ни одного решения, улучшающего целевую функцию, то работа алгоритма прекращается,  $x$  — результат, иначе перейти к шагу 3.

## Метод поиска с запретами

Метод поиска с запретами (Tabu Search) является одним из наиболее эффективных метаэвристических методов. Предложил его в 80-е гг. Ф. Гловер (F. Glover) [8, 9]. Отличительная черта этого метода заключается в процессе введения и снятия некоторых искусственных ограничений задачи в ходе поиска решения [6].

Основным недостатком метода ЛП является его остановка при достижении локального оптимума. Мы же ищем глобальный оптимум. Очевидно, что глобальный оптимум есть также и локальный, поэтому для успешного поиска решений мы должны как-то переходить от одного локального оптимума к другому [10].

В методе ПЗ с целью преодолеть вышеописанный недостаток вводится так называемый список запретов (Tabu List). Этот список хранит некоторое количество предыдущих решений, и при выборе нового решения запрещается выбирать из окрестности решения, содержащиеся в списке запретов [10]. Данный прием помогает избегать «застревания» в локальном оптимуме, расширяет пространство поиска, позволяя алгоритму поиска с запретами находить лучшие решения, чем метод ЛП.

## Метод мультистарта

Как уже было сказано, глобальный оптимум является в то же самое время одним из локальных оптимумов. Соответственно, если просмотреть все локальные оптимумы и выбрать из них лучший, то это даст нам окончательное решение задачи. Метод МС нацелен на то, чтобы обойти как можно большее число локальных оптимумов [11]. Суть метода проста: алгоритм ЛП (вернее, его модификация) запускается несколько раз. Лучшее решение, полученное при каждом запуске поиска, сохраняется в памяти. По окончании времени поиска лучший из локальных оптимумов возвращается в качестве решения задачи.

Метод даст хорошие результаты при выполнении двух условий [11]:

— каждый запуск алгоритма (будем называть его старт) должен стартовать из нового начального решения;

— сама процедура ЛП должна быть выполнена так, чтобы каждый новый старт по возможности приводил в новый локальный оптимум. Это необходимо для расширения пространства поиска.

Метод МС показывает результаты, сопоставимые с другими метаэвристиками, и поэтому часто используется для решения NP-трудных задач [12, 13].

## Окрестность решения

Понятие «окрестность» является самым интересным в метаэвристиках, основанных на ЛП. Оно плохо формализовано и для каждой конкретной задачи оптимизации обладает своей спецификой. В случае задачи размещения БС мы имеем решение, представленное в виде ряда БС с подключенными к ним клиентами. Необходимо определить, какие решения являются наиболее близкими к нашему. Нами был предложен метод формирования окрестности решения посредством осуществления небольших изменений в текущем решении. Новое решение из окрестности текущего решения можно получить одним из шести способов (операций).

1. *Смена типа одной БС на более дешевый.* Подобного рода операция ( $Sol[s].type = Sol[s].type - 1$ ) возможна для каждого элемента массива  $Sol$ , у которого  $Sol[s].type > 1$ . При новой конфигурации БС мы должны проверить решение  $Sol$  на соответствие требованиям (1)–(3).

2. *Смена типа одной БС на более дорогой.* Данная операция ( $Sol[s].type = Sol[s].type + 1$ ) возможна для каждого элемента массива  $Sol$ , у которого  $Sol[s].type \neq N_{types}$  и  $Sol[s].type \neq 0$ .

3. *Переключением одного клиента* (т. е. подключение к другой БС). Данная операция возможна для каждого клиента. Если мы ищем

новую БС для  $i$ -го клиента, то мы должны последовательно [проверяя на соответствие ограничениям (1)–(3)] пробовать подключить его к одному из мест-кандидатов (кроме его текущего места), начиная с самого ближнего к клиенту  $i$ . Очевидно, что место  $w$ , к которому мы хотим подключить клиента, должно иметь активную БС ( $\text{Sol}[w].type \neq 0$ ).

4. *Удаление одной БС.* Данная операция возможна для каждого активного места-кандидата. Для каждого из клиентов удаляемой станции  $s$  мы запускаем операцию «Переподключение одного клиента». Если все клиенты  $s$ -й станции можно подключить к другим БС, значит удаление  $s$ -й станции возможно.

5. *Добавление одной БС.* Данная операция возможна для каждого пустого места ( $\text{Sol}[s].type = 0$ ). Мы устанавливаем на место  $s$  новую станцию и пробуем подключить к ней самого близкого клиента (пусть его номер  $i$ ). Если такое подключение возможно [для  $s$ -го места выполняются ограничения (1)–(3)], то новое решение  $\text{Sol}$  является допустимым. Клиента с номером  $i$ , естественно, необходимо отключить от его старой БС.

6. *Перемещение одной БС.* Данная операция возможна для каждого активного места-кандидата. Пусть мы перемещаем БС с места  $s$ . Тогда мы должны последовательно пробовать поместить ее на пустые места-кандидаты, начиная с самого ближнего к месту  $s$ , с учетом того, что на новом месте  $w$  должны выполняться ограничения (2) и (3).

### Алгоритм вероятностного поиска с запретами для решения задачи размещения БС

Рассмотрим рандомизированную окрестность  $N_p(\mathbf{x}) \subseteq N(\mathbf{x})$ , где каждый элемент окрестности  $N(\mathbf{x})$  включается в множество  $N_p(\mathbf{x})$  с вероятностью  $0 \leq p \leq 1$  ( $p$  — параметр рандомизации окрестности) независимо от других элементов [14]. С ненулевой вероятностью множество  $N_p(\mathbf{x})$  может совпадать с  $N(\mathbf{x})$ , может оказаться пустым или содержать ровно один элемент. Алгоритм поиска с запретами, представленный в данном разделе, осуществляет вероятностный ЛП по рандомизированной окрестности (поэтому он называется вероятностный ЛП), совершая шаги как улучшающие, так и ухудшающие целевую функцию, что позволяет алгоритму перемещаться от одного локального оптимума к другому в целях найти среди них лучшее решение.

Использование рандомизированной окрестности  $N_p(\mathbf{x})$  дает лучшие результаты по сравнению с просмотром полной окрестности  $N(\mathbf{x})$  [14–16]. Очевидно, что при  $p = 1$  окрестность  $N_p(\mathbf{x})$  совпадает с  $N(\mathbf{x})$ .

Основным механизмом, позволяющим алгоритму покидать локальные оптимумы, является

список запретов  $TL$ . Список строится по предыстории поиска, т. е. по нескольким последним итерациям, и запрещает часть окрестности  $N(\mathbf{x})$  текущего решения [7]. В классическом методе поиска с запретами в  $TL$  добавляются те решения, которые были признаны лучшими на соответствующей итерации и к которым был применен переход как к новым текущим решениям.

Обновление списка запретов состоит из двух этапов: удаление самого «старого» элемента списка (если текущая длина списка равна максимальной длине  $l$ ) и добавление в список нового элемента. Однако специфика нашей задачи такова, что для нее характерно очень большое пространство поиска, в котором легко «застыть» в локальной окрестности, принадлежащей одному пику, даже если использовать очень большой список запретов. Возможных решений может быть слишком много.

Поэтому мы предлагаем в список запретов добавлять не решения, а операции, которые нельзя будет осуществлять в течение  $l$  следующих итераций. То есть каждое изменение, примененное к решению, порождает запрет на операцию, которая может вернуть нас в это самое решение. Ниже приведены пары «изменение» — «запрет» (далее фразой типа « $s$ -я БС» мы для краткости обозначаем БС, расположенную на  $s$ -м месте):

- 1) «смена типа  $s$ -й БС на более дешевый» — «смена типа  $s$ -й БС на более дорогой»;
- 2) «смена типа  $s$ -й БС на более дорогой» — «смена типа  $s$ -й БС на более дешевый»;
- 3) «переподключение  $i$ -го клиента» — «переподключение  $i$ -го клиента»;
- 4) «удаление  $s$ -й БС» — «добавление  $s$ -й БС»;
- 5) «добавление  $s$ -й БС» — «удаление  $s$ -й БС»;
- 6) «перемещение  $s$ -й БС на  $w$ -е место» — «перемещение  $w$ -й БС».

Представим общую схему вероятностного поиска с запретами для задачи размещения БС.

1. Построить начальное решение **Curr**.

Введем обозначения:  $p$  — параметр рандомизации окрестности;  $TL$  — список запретов; **Best\_neigh** — лучшее решение в текущей окрестности;  $cost\_best\_neigh$  — целевая функция решения **Best\_neigh**; **Best** — лучшее решение задачи.

2.  $TL = \emptyset$ .

3. **Best = Curr**.

4. Построим окрестность решения **Curr** при помощи шести вышеописанных операций (кроме тех, которые есть в списке запретов  $TL$ ).

5.  $cost\_best\_neigh = \infty$ .

6. Цикл. Для каждого решения **Sol** из построенной окрестности выполним:

6.1. Пусть  $w$  — случайное число из диапазона  $[0; 1]$ . Если  $w > p$ , то переход к следующей итерации цикла.

6.2. Если решение  $Sol$  — недопустимое [не удовлетворяет одному из ограничений (1)–(3)], то переход к следующей итерации цикла.

6.3. Если  $\Phi(Sol) < cost\_best\_neigh$ , то  $Best\_neigh = Sol$  и  $cost\_best\_neigh = \Phi(Sol)$ .

7. Если  $cost\_best\_neigh = \infty$ , то переход к п. 11.

8.  $Curr = Best\_neigh$ .

9. Обновить  $TL$ .

10. Если  $\Phi(Curr) < \Phi(Best)$ , то  $Best = Curr$ .

11. Если время работы алгоритма  $time$  истекло, то переход к п. 12, иначе к п. 4.

12. Вернем  $Best$  в качестве решения задачи.

### Алгоритм мултистарта для решения задачи размещения БС

Введем понятие жадного алгоритма. В жадном алгоритме делается выбор, который является самым лучшим в данный момент, т. е. производится локально оптимальный выбор в надежде, что он приведет к оптимальному решению глобальной задачи [17]. В предлагаемом нами методе мултистарта используется жадная эвристика «первое улучшение» [11]. Суть ее заключается в том, что при ЛП мы рассматриваем одно случайное соседнее решение  $z$  из окрестности текущего решения  $x$ . Если  $f(z)$  лучше, чем  $f(x)$ , то текущим решением становится  $z$ . Подобный переход к первому же лучшему решению расширяет пространство поиска, что в сочетании со стартами из разных начальных решений позволит попасть в большее количество разных локальных оптимумов.

Новые потенциальные решения мы будем генерировать не из окрестностей  $N(x)$  или  $N_p(x)$ , а из окрестности  $N_{greed}(x)$ .  $N_{greed}(x)$  — окрестность, получаемая при помощи операций «удаление одной БС» и «смена типа одной БС на более дешевый» к  $x$ . Таким образом,  $N_{greed}(x)$ , в отличие от  $N(x)$ , содержит только решения с целевой функцией, лучшие, чем у  $x$ .

Ниже представлена общая схема алгоритма мултистарта для задачи размещения БС.

1.  $i = 0$ .

Введем обозначения:  $N\_iter\_max$  — максимальное число итераций алгоритма подряд без улучшения целевой функции;  $n\_iter$  — число итераций подряд без улучшения целевой функции.

2.  $i = i + 1$ ;  $n\_iter = 0$ .

3. Построить начальное решение  $Curr_i$ .

4. Сгенерируем  $Sol$  — решение из окрестности  $Curr_i$ , полученное при помощи операции «Удаление одной БС» или «Смена типа одной БС на более дешевый».

5. Если решение  $Sol$  — допустимое [удовлетворяет каждому из ограничений (1)–(3)], то переход к п. 6. Если время работы алгоритма истекло, то переход к п. 9, иначе к п. 4.

6. Если  $\Phi(Sol) < \Phi(Curr_i)$ , то  $Curr_i = Sol$  и  $n\_iter = 0$ , иначе  $n\_iter = n\_iter + 1$ .

7. Если время работы алгоритма  $time$  истекло, то переход к п. 9.

8. Если  $n\_iter = N\_iter\_max$ , то переход к п. 2, иначе к п. 4.

9. Вернем лучшее из решений  $Curr_i$  в качестве окончательного решения задачи.

### Компьютерное моделирование

Разработанные алгоритмы реализованы как программное обеспечение в среде Embarcadero Delphi XE5. С его помощью был проведен ряд вычислительных экспериментов по нахождению оптимального расположения БС и подключения к ним клиентов. Моделирование проводилось на компьютере с процессором Intel Core i5-3470 и оперативной памятью 6 ГБ.

Первая серия вычислительных экспериментов была посвящена исследованию быстродействия и точности предложенных модификаций методов ПЗ и МС путем сравнения их с методом полного перебора (ПП). Принцип алгоритма ПП очень прост: мы должны перебрать все возможные решения задачи, отсеять все недопустимые решения, а среди оставшихся выбрать лучшее (с точки зрения значения целевой функции). Так как метод ПП не позволяет решать задачу за полиномиальное время, сравнение проводилось на задачах малой и средней размерности.

Пусть  $N_{types} = 2$ . Эксперименты проводились на 9 тестовых задачах (по 3 значения  $N_{tp}$  и  $N_{ps}$  порождают 9 комбинаций). Зафиксируем параметры алгоритма ПЗ:  $l = 50$ ,  $p = 0,15$ ,  $time = 0,1$  с. Параметры алгоритма МС:  $N\_iter\_max = 50$ ,  $time = 0,1$  с. Результаты работы алгоритмов приведены в табл. 1. Каждая ячейка табл. 1 содержит три строки: верхняя — это время работы алгоритма ПЗ в секундах, в скобках — значение соответствующей целевой функции в условных единицах, средняя и нижняя — аналогичные

■ Таблица 1. Сравнение результатов алгоритмов поиска с запретами и мултистарта с методом полного перебора

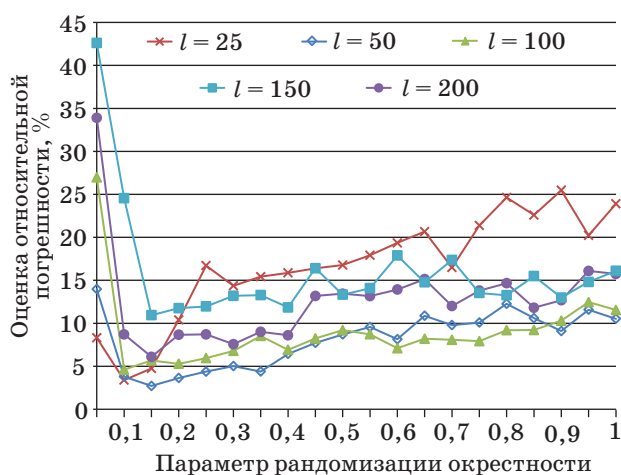
$N_{tp}$	$N_{ps}$		
	5	7	10
3	0,100 с (27 382)	0,100 с (26 784)	0,100 с (26 516)
	0,100 с (27 382)	0,100 с (26 784)	0,100 с (26 516)
	0,030 с (27 382)	0,710 с (26 784)	65,19 с (26 516)
5	0,100 с (26 776)	0,100 с (26 570)	0,100 с (26 476)
	0,100 с (26 776)	0,100 с (26 570)	0,100 с (26 476)
	0,606 с (26 776)	31,59 с (26 570)	5140,7 с (26 476)
7	0,100 с (26 458)	0,100 с (26 396)	0,100 с (26 230)
	0,100 с (26 458)	0,100 с (26 396)	0,100 с (26 230)
	14,65 с (26 458)	1390 с (26 396)	455798 с (26 230)

величины для МС и ПЗ соответственно. Время решения задачи для методов ПЗ и МС приводится как среднее за 10 запусков алгоритма, значение целевой функции — как лучшее за 10 запусков алгоритма. Представленные в табл. 1 данные свидетельствуют о том, что на задачах малой размерности предложенные алгоритмы обеспечивают получение точных значений целевой функции, как и метод ПП, который дает точное решение при каждом запуске алгоритма. При этом алгоритмы ПЗ и МС справляются с решением задач размещения БС за очень малое время.

Далее был проведен ряд вычислительных экспериментов с целью выявить влияние параметра рандомизации окрестности  $p$  и длины списка запретов  $l$  на качество получаемых решений методом ПЗ. В качестве тестового примера была выбрана задача следующей размерности:  $N_{tp} = 100$ ,  $N_{ps} = 100$ ,  $N_{types} = 3$ . Результаты эксперимента представлены на рисунке. По оси абсцисс отложены значения  $p$  в диапазоне [5 %; 100 %] с дискретностью 5 %. По оси ординат отложены значения оценки относительной погрешности решений, которая считается как среднее отклонение полученных решений от лучшего из известных решений задачи. Для каждой пары  $p$  и  $l$  приведены усредненные результаты за 50 запусков,  $time = 20$  с.

Как видно из графиков, на данных примерах алгоритм ПЗ достигает наибольшей эффективности при  $p \in [0,1; 0,35]$ , что в целом согласуется с результатами работы [14]. Также выявлено, что при значениях длины списка запретов, равных 50–100, алгоритм поиска с запретами показывает наилучшие результаты. Наилучшее значение оценки относительной погрешности получено при  $l = 50$  и  $p = 0,15$  (2,709 %).

Также был проведен ряд вычислительных экспериментов с целью сравнить эффективность методов ЛП, ПЗ и МС при решении задач разной



■ Влияние рандомизации окрестности и длины списка запретов на относительную погрешность решений

■ Таблица 2. Сравнение результатов локального поиска, алгоритма поиска с запретами и метода мултистарта

Размерность задачи ( $N_{tp} \times N_{ps} \times N_{types}$ )	Метод		
	ЛП	ПЗ	МС
50×50×3	0,191 с 32,775 %	0,191 с 20,598 %	0,191 с 13,828 %
100×100×3	2,535 с 31,690 %	2,535 с 13,194 %	2,535 с 9,170 %
150×150×3	11,933 с 38,305 %	11,933 с 22,039 %	11,933 с 13,299 %
200×200×3	36,389 с 24,008 %	36,389 с 16,130 %	36,389 с 13,700 %

размерности. Зафиксируем параметры алгоритма ПЗ:  $l = 50$ ,  $p = 0,15$ . Параметры алгоритма МС:  $N_{iter\_max} = 50$ . Моделирование производилось следующим образом: сначала каждую из задач мы решали методом ЛП (усредняя результаты за 50 запусков). Далее мы решали ту же задачу методами ПЗ и МС (те же 50 запусков для каждого метода), ограничивая их работу временем, которое понадобилось методу ЛП для решения задачи. Результаты эксперимента приведены в табл. 2. Каждая ячейка таблицы содержит две строки: верхняя — это время работы алгоритма, нижняя — значение оценки относительной погрешности решений, получаемых данным алгоритмом. Из таблицы видно, что методы ПЗ и МС решают задачу размещения БС лучше, чем алгоритм ЛП. При этом алгоритм МС показывает лучшие результаты, чем вероятностный ПЗ.

### Заключение

Проведенное компьютерное моделирование позволяет сделать следующие выводы.

1. Предложенные модификации алгоритмов поиска с запретами и мултистарта способны решать задачу размещения БС.
2. Для задач малой и средней размерности результаты методов ПЗ и МС совпадают с результатами метода ПП. При этом предложенные методы решают задачу на много порядков быстрее, чем точный метод ПП.
3. Программное обеспечение, разработанное на базе предложенных алгоритмов, может находить решение задач средней и большой размерности за время, не превышающее 1 мин.
4. Рекомендованными параметрами алгоритма ПЗ, позволяющими находить хорошее решение за приемлемое время, являются следующие:  $l = 50-100$ ,  $p = 0,1-0,35$ .
5. Предложенные модификации методов ПЗ и МС решают задачу размещения БС лучше, чем алгоритм ЛП. При этом алгоритм МС показывает лучшие результаты, чем ПЗ.

## Литература

1. Вишнеvский В. М. Теоретические основы проектирования компьютерных сетей. — М.: Техносфера, 2003. — 512 с.
2. Ермолаев С. Ю. Оптимальное размещение базовых станций // Telecommunication Sciences. 2010. № 1. С. 349–355.
3. St-Hilaire M., Chinnek J. W., Chamberland S., Pierre S. Efficient Solution of the 3G Network Planning Problem // Computers & Industrial Engineering. 2012. N 4. P. 819–830.
4. Архангельский А. Я. Программирование в Delphi 7. — М.: Бином-Пресс, 2003. — 1152 с.
5. Koskie S., Gajic Z. A Nash Game Algorithm for SIR-based Power Control in 3G Wireless CDMA Networks // IEEE/ACM Transactions on Networking. 2005. N 5. P. 1017–1026.
6. Усманова А. Р. Метод поиска с запретами для задачи упаковки в контейнеры: дис. ... канд. физ.-мат. наук. — Уфа, 2002. — 101 с.
7. Скаков Е. С. Метод поиска с запретами для решения оптимизационных задач // Новое слово в науке и практике: гипотезы и апробация результатов исследований: тр. 15-й Междунар. конф., Новосибирск, 23 января 2015 г. Новосибирск, 2015. С. 166–171.
8. Glover F. Tabu Search. Part I // ORSA Journal on Computing. 1989. N 3. P. 190–206.
9. Glover F. Tabu Search. Part II // ORSA Journal on Computing. 1990. N 1. P. 4–32.
10. Luke S. Essentials of Metaheuristics. — Lulu, 2013. — 242 p.
11. Gendreau M., Potvin J. Y. Handbook of Metaheuristics. — Springer, 2010. — 669 p.
12. Brønmo G., Christiansen M., Fagerholt K., Nygreen B. A Multi-Start Local Search Heuristic for Ship Scheduling — a Computational Study // Computers & Operations Research. 2007. N 3. P. 900–917.
13. Martí R., Resende M. G. C., Ribeiro C. C. Multi-Start Methods for Combinatorial Optimization // European Journal of Operational Research. 2013. N 1. P. 1–8.
14. Руднев А. С. Вероятностный поиск с запретами для задачи упаковки кругов и прямоугольников в полосу // Дискретный анализ и исследование операций. 2009. № 4. С. 61–86.
15. Гончаров Е. Н., Кочетов Ю. А. Вероятностный поиск с запретами для дискретных задач безусловной оптимизации // Дискретный анализ и исследование операций. 2002. № 2. С. 13–30.
16. Кочетов Ю. А. Методы локального поиска для дискретных задач размещения: дис. ... д-ра физ.-мат. наук. — Новосибирск, 2009. — 267 с.
17. Гончаров Е. Н., Кочетов Ю. А. Поведение вероятностных жадных алгоритмов для многостадийной задачи размещения // Дискретный анализ и исследование операций. 1999. № 1. С. 12–32.

UDC 621.395.7, 519.854

doi:10.15217/issn1684-8853.2015.3.99

## Multi-Start and Tabu Search Algorithms for Base Station Location Problem

Skakov E. S.<sup>a</sup>, Post-Graduate Student, wallkirya@mail.ruMalysh V. N.<sup>a</sup>, Dr. Sc., Tech., Professor, vmalysh@mail.ru<sup>a</sup>Lipetsk State Pedagogical University, 42, Lenina St., 398020, Lipetsk, Russian Federation

**Purpose:** The synthesis of a topological structure for a wireless data network implies planning territorial distribution of the base transceiver stations on potential sites, and connecting to the clients. The existing approaches to this problem have shortcomings: using solution methods with low computing speed (branch and bound method, heuristic method of Lagrange, etc.); lack of restrictions which would take into account the level of signal fading during its propagation from the base station to the client and back and also the level of inter-cell interference; using only one type of base stations. The goal of this research is creating a model for base station location which would be free from the above-mentioned drawbacks. **Results:** The problem of base station location has been formulated, taking into account the signal-to-interference ratio for the network clients. A problem solution is represented as a vector of structures, each storing the information about one potential site (the installed base station type, the list of connected clients). Modified algorithms of probabilistic tabu search and multi-start have been developed, based on the concept of current solution neighborhood. A new solution from the neighborhood of the current solution can be obtained using one of the following six operations: changing the type of a station to a more cheap/expensive one, reconnecting a client, removing a base station, adding a station, moving a base station. In order to avoid getting stuck in local optima during the tabu search, the algorithm is disabled to browse solutions from the tabu list. The novelty of the approach is that the tabu list is filled not with real decisions made in the past, but with operations to change the network configuration which can bring us back to old local optima. The essence of the modified multi-start algorithm is as follows: only two operations are used to obtain a new solutions (removing a base station and changing the type of a station to a cheaper one); only a part of the neighborhood is browsed; the transition to a new solution is performed on the "first improvement" principle; the algorithm of finding the best solutions runs several times. The developed algorithms are implemented in Delphi programming environment. It has been shown that the new algorithms show better results than the local search method. **Practical relevance:** The developed modifications of tabu search and multi-start methods help to find the solution for the base station location problem in a reasonable time, many orders of magnitude faster than the exact method of exhaustive search. It has been found that the quality of the problem solution obtained by probabilistic tabu search depends on the length of the tabu list and on the value of the neighborhood randomization parameter.

**Keywords** — Base Station Location, Multi-Start Algorithm, Tabu Search, Wireless Networks, Optimization, SIR.

## References

- Vishnevskii V. M. *Teoreticheskie osnovy proektirovaniia komp'iuternykh setei* [Theoretical Bases of Designing Computer Networks]. Moscow, Tekhnosfera Publ., 2003. 512 p. (In Russian).
- Ermolaev S. I. Optimal Base Station Location. *Telecommunication Sciences*, 2010, vol. 1, no. 1, pp. 349–355 (In Russian).
- St-Hilaire M., Chinnek J. W., Chamberland S., Pierre S. Efficient Solution of the 3G Network Planning Problem. *Computers & Industrial Engineering*, 2012, vol. 63, no. 4, pp. 819–830.
- Arkhangel'skii A. I. *Programmirovaniie v Delphi 7* [Programming in Delphi 7]. Moscow, Binom-Press Publ., 2003. 1152 p. (In Russian).
- Koskie S., Gajic Z. A Nash Game Algorithm for SIR-based Power Control in 3G Wireless CDMA Networks. *IEEE/ACM Transactions on Networking*, 2005, vol. 13, no. 5, pp. 1017–1026.
- Usmanova A. R. *Metod poiska s zapretami dlia zadach upakovki v konteynery*. Dis. kand. fiz.-mat. nauk [Tabu Search Method for Bin Packing Problem. PhD phys. and math. sci. diss.]. Ufa, 2002. 101 p. (In Russian).
- Skakov E. S. Tabu Search for Solving Optimization Problems. *Trudy 15 Mezhdunarodnoi konferentsii "Novoe slovo v nauke i praktike: gipotezy i aprobatitsia rezul'tatov issledovaniia"* [Proc. 15th Int. Conf. "The New Word in Science and Practice: Hypotheses and Testing of the Results of Researches"]. Novosibirsk, 2015, pp. 166–171 (In Russian).
- Glover F. Tabu Search. Part I. *ORSA Journal on Computing*, 1989, vol. 1, no. 3, pp. 190–206.
- Glover F. Tabu Search. Part II. *ORSA Journal on Computing*, 1990, vol. 2, no. 1, pp. 4–32.
- Luke S. *Essentials of Metaheuristics*. Lulu, 2013. 242 p.
- Gendreau M., Potvin J. Y. *Handbook of Metaheuristics*. Springer, 2010. 669 p.
- Brønmo G., Christiansen M., Fagerholt K., Nygreen B. A Multi-Start Local Search Heuristic for Ship Scheduling — a Computational Study. *Computers & Operations Research*, 2007, vol. 34, no. 3, pp. 900–917.
- Martí R., Resende M. G. C., Ribeiro C. C. Multi-Start Methods for Combinatorial Optimization. *European Journal of Operational Research*, 2013, vol. 226, no. 1, pp. 1–8.
- Rudnev A. S. Probabilistic Tabu Search Algorithm for the Packing Circles and Rectangles into the Strip. *Diskretnyi analiz i issledovanie operatsii*, 2009, vol. 16, no. 4, pp. 61–86 (In Russian).
- Goncharov E. N., Kochetov I. A. Probabilistic Tabu Search for Discrete Unconstrained Optimization. *Diskretnyi analiz i issledovanie operatsii*, 2002, vol. 9, no. 2, pp. 13–30 (In Russian).
- Kochetov I. A. *Metody lokal'nogo poiska dlia diskretnykh zadach razmeshcheniia*. Dis. dokt. fiz.-mat. nauk [Local Search Methods for Discrete Facility Location Problems. Dr. phys. and math. sci. diss.]. Novosibirsk, 2009. 267 p. (In Russian).
- Goncharov E. N., Kochetov I. A. The Behavior of Probabilistic Greedy Algorithms for Multi-stage Facility Location Problem. *Diskretnyi analiz i issledovanie operatsii*, 1999, vol. 6, no. 1, pp. 12–32 (In Russian).

К статье А. В. Назарова, В. Л. Якимова, В. А. Авдеева «Алгоритм максимизации энтропии обучающей выборки и его использование при синтезе моделей прогноза дискретных состояний нелинейных динамических систем» (ИУС, 2015, № 2, с. 57–66).

На с. 58 формула (2) напечатана в следующем виде:

$$\bar{G} = -\bar{P}_{\text{ош}} = -\frac{1}{h_3} \sum_{h=1}^{h_3} P_{\text{ош } h}, \quad (2)$$

Следует читать

$$\bar{G} = 1 - \bar{P}_{\text{ош}} = 1 - \frac{1}{h_3} \sum_{h=1}^{h_3} P_{\text{ош } h}, \quad (2)$$

Редакция приносит извинения за допущенную ошибку.