



М. А. Плаксин,

Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал);
Пермский государственный национальный исследовательский университет

ПРОПЕДЕВТИКА ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ШКОЛЬНОЙ ИНФОРМАТИКЕ. ПАРАЛЛЕЛЬНАЯ ФОРМА АЛГОРИТМА В ЗАДАЧАХ НА ОБХОД ГРАФА

Аннотация

Для освоения понятия «параллельная (ярусно-параллельная) форма алгоритма» предлагается использовать задачи на построение/обход «лабиринтов». В качестве примера дается разбор задания из конкурса «ТРИЗформашка-2015».

Ключевые слова: информатика, средняя школа, начальная школа, методика обучения, параллельные алгоритмы, параллельная форма алгоритма, ярусно-параллельная форма, обход графа, параллельные вычисления, пропедевтика, ТРИЗформатика, ТРИЗформашка.

Контактная информация

Плаксин Михаил Александрович, канд. физ.-мат. наук;

доцент кафедры информационных технологий в бизнесе Пермского филиала Национального исследовательского университета «Высшая школа экономики»; *адрес:* 614070, г. Пермь, ул. Студенческая, д. 38; *телефон:* (342) 205-52-50;

доцент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета; *адрес:* 614990, г. Пермь, ул. Букирева, д. 15; *телефон:* (342) 239-67-72;

e-mail: mapl@list.ru

M. A. Plaksin,

National Research University Higher School of Economics, Perm Branch;
Perm State National Research University

PROPAEDEUTICS OF PARALLEL COMPUTING IN SCHOOL INFORMATICS. PARALLEL FORM OF THE ALGORITHM IN GRAPH TRAVERSAL TASKS

Abstract

It is proposed to use the tasks of building/bypass "labyrinths" to learn the concept of "parallel form of the algorithm (multilevel structure)". As an example the investigation of task from the contests "TRIZformashka-2015" is given.

Keywords: informatics, secondary school, primary school, teaching methods, parallel algorithms, parallel form of algorithm, multilevel structure, graph traversal, parallel computing, propaedeutics, TRIZformatics, TRIZformashka.

Современный этап развития computer science связан с массовым распространением параллелизма вычислений на всех уровнях (многоядерные процессоры, многомашинные кластеры, многопроцессорные ЭВМ). Это делает актуальным включение пропедевтики параллельного программирования в школьный курс информатики.

В рамках работ над «пермской версией» пропедевтического курса информатики (авторский коллектив: М. А. Плаксин, Н. Г. Иванова, О. Л. Русакова; рабочее название курса — «ТРИЗформатика») [7, 8] разработка данной тематики начата в 2013 году. Эффективной площадкой для этого выступил конкурс «ТРИЗформашка» — ежегодный межрегиональный интернет-конкурс по информатике, системному анализу и ТРИЗ (теории решения изобретательских задач) для школьников и студентов [3–5, 9]. В марте 2018 года конкурс состоится в 18-й раз. Возраст участников — от первого класса школы до четвертого курса вуза. Среднее количество команд — около 100 (рекордное — 202). География конкурса — от Владивостока до Риги. Сайт конкурса: <http://www.trizformashka.ru>

В процессе подготовки конкурса «ТРИЗформашка» были определены типы задач, направленных на освоение базовых понятий параллельных вычислений, и придумано по несколько задач этих типов. Одним из таких типов являются задачи на параллельную (ярусно-параллельную) форму алгоритма. Понятие ярусно-параллельной формы (ЯПФ) описано в [1, 2, 6]. В качестве иллюстрации в [6] рассматривается задание из конкурса «ТРИЗформашка-2014», в котором ярусно-параллельная форма вводится в задаче о строительстве замка. В конкурсах 2015, 2016 и 2017 годов в задании на ЯПФ были использованы задачи на прохождение лабиринта (с точки зрения математики это задачи на обход графа). Такие задачи будут иметь некоторые важные отличия от «строительных» задач. Анализу этих отличий посвящена данная статья. В качестве иллюстрации рассматривается задача из конкурса «ТРИЗформашка-2015».

Ходы кривые роет подземный умный крот... (задача из конкурса «ТРИЗформашка-2015»)

Кроты копают подземный ход, начиная с точки А (рис. 1).

Подземный ход — узкий. Поэтому два крота не могут находиться рядом.

За один день один крот может прокопать расстояние от одного угла клеточки до другого (по горизонтали, вертикали или диагонали).

1. Сколько времени потребуется одному кроту, чтобы вырыть весь подземный ход? Ответ обосновать.

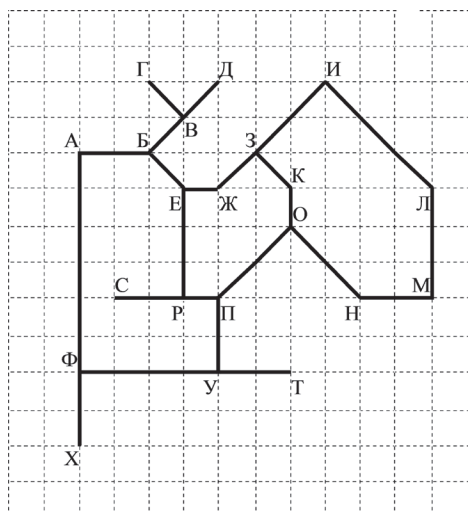


Рис. 1. Схема подземного хода
для задачи «Ходы кривые роет подземный умный крот...»

2. За какое минимальное время можно вырыть этот подземный ход, если в рытье может участвовать неограниченное количество кротов? Ответ обосновать.
3. Какое минимальное количество кротов должно участвовать в рытье, чтобы вырыть подземный ход за минимальное время? Ответ обосновать.
4. За какое время этот подземный ход смогут вырыть четыре крота? Ответ обосновать.
5. Сколько кротов надо, чтобы прорыть этот ход за 10 дней? Ответ обосновать.

Перед продолжением чтения читателю рекомендуется самостоятельно решить предложенную задачу.

Некоторые особенности задач на обход графов с точки зрения построения ЯПФ

В [6] понятие ярусно-параллельной формы было проиллюстрировано задачей на построение замка по заданному чертежу (рис. 2). Вершина графа ЯПФ в данном случае соответствует установке одного камня.

Для всех камней в замке однозначно определено, какой камень на какой опирается (или не опирается).

При построении ЯПФ применялось следующее правило:

Каждая вершина попадает на ярус, непосредственно следующий за ярусом, на котором расположены вершины, ей предшествующие. Расположение вершины на ярусе N означает, что последние необходимые для этого вершины были размещены на ярусе (N – 1).

Например, камень 15 будет установлен после камней 16, 17, 19. В ЯПФ строительства замка камень 16 располагается на ярусе 1, камень 19 — на ярусе 3, камень 17 — на ярусе 4. Значит, камень 15 будет расположен на ярусе 5.

ЯПФ, построенная по этому правилу, обладает минимальной высотой и максимальной шириной.

В данной статье рассматривается задача на построение/обход лабиринта. Лабиринт естественным образом трактуется как граф. Точки, обозначенные буквами, трактуются как вершины, коридоры между ними — как дуги. Задача на обход лабиринта представляется задачей на обход графа.

Сравним эту задачу со «строительной» задачей, рассмотренной в [6].

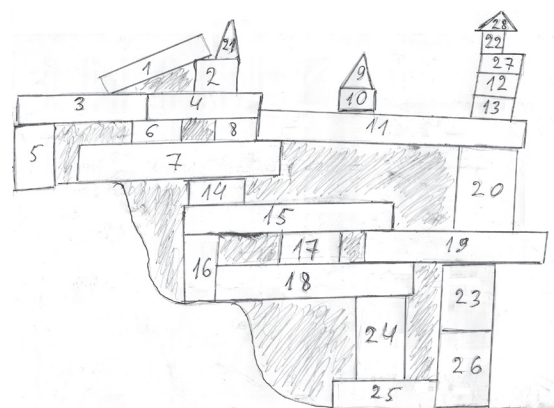


Рис. 2. «Чертеж дворца» для задачи из конкурса
«ТРИЗформашка-2014»

Для начала отметим, что аналогом укладки камней из «строительной» задачи в «лабиринтной» задаче будет являться прохождение коридоров лабиринта, т. е. дуг графа. Тем не менее из соображений удобства и наглядности далее мы будем говорить не только о дугах, но и о вершинах.

В «строительной» задаче строительство замка велось строго снизу вверх. Камень, укладываемый сверху, обязательно опирался на камни, уложенные снизу. Для каждого камня было однозначно определено его отношение с каждым другим камнем: предшествует, следует, не связан. Это отношение однозначно определяется статически, по внешнему виду рисунка, без учета динамики построения замка, без учета порядка укладки камней.

В задаче на обход лабиринта это не так. Здесь нет однозначного статически определенного отношения следования! Лабиринт можно пройти в разном порядке. Например, в вершину Б (см. рис. 1) можно прийти из вершины А (по дуге АБ), а можно из вершины Е (пройдя по пути АФУПРЕБ). В вершину О можно попасть из вершины К, можно из вершины П, можно из вершины Н. В зависимости от того, какая вершина окажется предшествующей, поменяются и вершины, являющиеся следующими. Так, для примера с вершиной Б в первом случае последующими окажутся вершины В и Е, во втором — вершины В и А. Для примера с вершиной О в первом случае последующими окажутся вершины П и Н, во втором — вершины К и Н, в третьем — вершины К и П. Лишь для вершин, лежащих на тупиковых ветвях лабиринта, предшествующая вершина определяется однозначно. Так, на рисунке 1 для вершины Х предшествующей является вершина Ф, для вершины Т — вершина У, для вершины С — вершина Р, для вершин Г и Д — вершина В.

Все вышесказанное можно без труда перевести с «языка вершин» на «язык дуг».

Дуга, соединяющая вершины А и Б, может либо не иметь предшествующей (в том случае, если с нее начнется обход графа), либо следовать за дугой ЕБ (после прохождения по одному из путей, начинающихся с АФУП: АФУПРЕБ, АФУПОКЗЖЕБ, АФУПОНМЛИЗЖЕБ). Дуга КО может предшествовать дугам ОП и ОН. Но дуга НО тоже может быть предшествующей — для дуг ОК и ОП, которые будут ей следовать. Наконец, и дуга ПО может оказаться предшествующей, а следовать за ней будут дуги ОК и ОН. (Буквы в названии дуги меняются местами, чтобы указать направление прохода по дуге.)

Для задач обхода графа отношение следования определяется динамически по мере выполнения алгоритма

обхода. По внешнему виду графа установить, в каком порядке будут пройдены дуги, вообще говоря, невозможно! (Очевидные вырожденные случаи нам сейчас не интересны.)

Необходимость строительства «снизу вверх», наличие статически определенного отношения следования делает порядок укладки камней вполне определенным, позволяет достаточно хорошо разграничить уже выполненную часть работы и еще не выполненную: уже уложенные камни расположены снизу, еще не уложенные — сверху. Отсутствие статически определенного отношения следования в задаче обхода графа порождает множество возможных путей обхода. Никаких признаков, по которым можно было бы отделить уже пройденную часть графа от еще не пройденной, здесь нет.

В «строительной» задаче каждый камень опирается на все «предшествующие» камни. Чтобы положить камень, надо сначала уложить все камни, которые ему «непосредственно предшествуют» (на которые он «непосредственно опирается»). В «лабиринтной» задаче для прохождения вершины достаточно подойти к ней по одной из дуг. С прохождением остальных дуг этот факт никак не связан. Если мы переходим в вершину О по дуге ПО, это ровно ничего не говорит о состоянии дуг ОК и ОН. Любая из этих дуг может быть уже пройдена, а может быть нет.

В «строительной» задаче после того, как мы положили некоторый камень, мы должны уложить на него все «непосредственно следующие» камни (т. е. камни, которые непосредственно опираются на данный камень). «Следующие» камни должны быть уложены обязательно все. В «лабиринтной» задаче порядок прохода вершин определяется динамически по мере обхода графа. Поэтому вполне может оказаться, что, когда мы дойдем до некоторой вершины (перейдем в нее по некоторой дуге), часть дуг, исходящих из этой вершины, уже пройдена.

В «строительной» задаче всегда известен камень, имеющий самый высокий уровень. Именно он и будет определять длину критического пути. В «лабиринтной» задаче место завершения обхода заранее не известно. В этой роли могут оказаться многие различные точки.

По этим причинам «лабиринтные» задачи первоначально представляются значительно более сложными, чем «строительные». Попытка решить «лабиринтную» задачу без использования механизма ярусно-параллельных форм вызывает значительные трудности.

Как уже было сказано, при обходе графа ни по внешнему виду, ни по расположению дуги ничего нельзя сказать о том, была она уже пройдена или нет, надо ли ее еще проходить или нет. Это можно только запомнить. А учитывая крайне малый объем сверхоперативной памяти человека (знаменитое «волшебное число 7 плюс/минус 2»), лучше записать. Ярусно-параллельная форма оказывается очень удобной формой фиксации проделанной части работы (пройденной части графа). Поэтому в случае применения ЯПФ практически все вышеуказанные трудности снимаются. В частности, в процессе построения ярусно-параллельной формы, ориентированной на минимальное время выполнения алгоритма, мы автоматически определим точку, наиболее отдаленную от входа в лабиринт.

В силу особенностей построения ярусно-параллельной формы для алгоритма обхода лабиринта (у каждой вершины — единственная входная дуга и сколько угодно выходных), ЯПФ для «лабиринтных» задач будет

деревом. Более точно: ярусно-параллельная форма будет одним из остовных деревьев исходного графа, т. е. деревьев, которые включают все вершины графа и дуги, минимально необходимые для обеспечения его связности [10].

Разбор задачи «Ходы кривые роет подземный умный крот...»

Вернемся к задаче про мудрого крота.

Для удобства обсуждения перенумеруем строки и столбцы на схеме подземного хода (рис. 3).

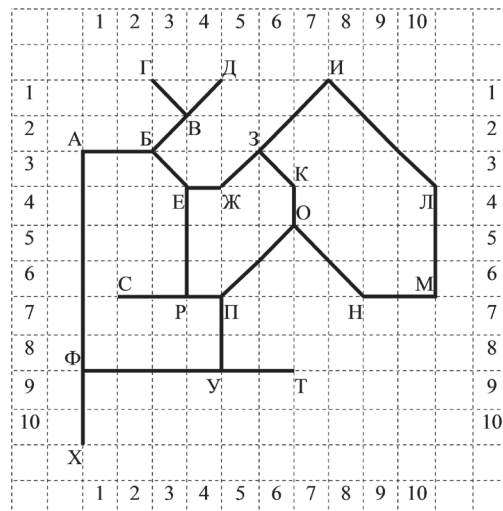


Рис. 3. Схема подземного хода с нумерацией строк и столбцов

Всего одиночных отрезков — 46.

Для ответа на поставленные в задании вопросы необходимо построить ярусно-параллельную форму алгоритма рытья подземного хода. Воспользуемся вышеприведенным правилом из [6] для построения графа минимальной высоты и максимальной ширины.

Граф ЯПФ будет выглядеть так, как представлено на рисунке 4.

В названии вершины графа двумя буквами обозначен отрезок, по которому движется крот. Число, стоящее после букв, — это номер клетки по вертикали или по горизонтали. Оно играет роль для отрезков, имеющих длину в несколько клеток. Ссылка на «номер по вертикали» или «номер по горизонтали» выбрана из соображений удобства без соблюдения каких-либо формальных правил и может быть изменена по желанию читателя.

Заметим, что вершинами графа ЯПФ являются не точки со схемы подземного хода, а «единицы работы», т. е. единичные отрезки, из которых состоит подземный ход.

Высота ЯПФ равна 12, ширина — 8. Это ширина яруса 8.

Напомним, что в ЯПФ, построенной по правилам из [6], высота графа определяет минимальное время выполнения алгоритма, а ширина — максимальную степень распараллеливания.

То есть минимальное время прохождения лабиринта (прокапывания подземного хода) равно 12. Для этого требуется восемь кротов.

Можно ли сжать ЯПФ? Сделать ее более узкой, и тем самым уменьшить количество кротов, необходимых для построения лабиринта?

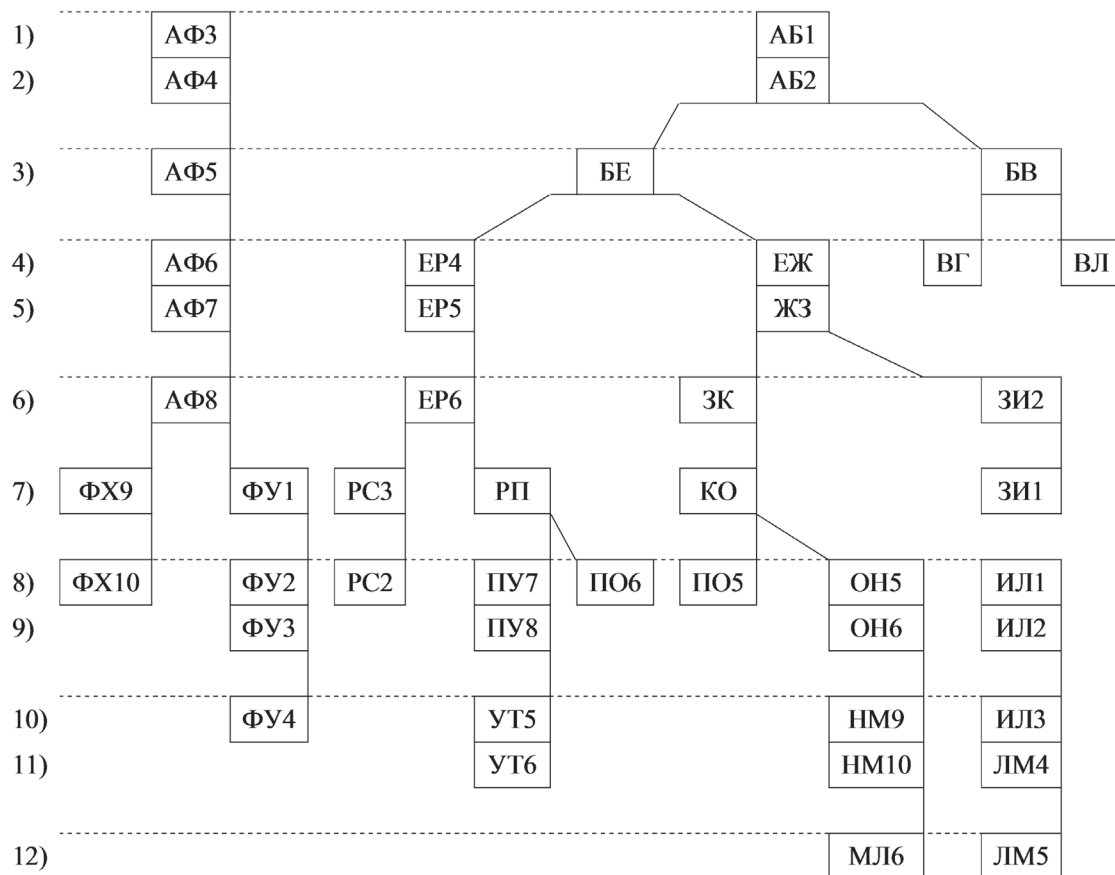


Рис. 4. Ярусно-параллельная форма для задачи про кротов

Граф ЯПФ содержит 46 вершин. При высоте 12 граф может иметь ширину 4 ($46 : 12$, округленное вверх). Значит, возможность сжатия будет определяться распределением вершин по ярусам. Существует ли возможность спустить вершины с перегруженных ярусов вниз? Да.

Вершину ФХ10 можно спустить на 9-й, или 10-й, или 11-й, или 12-й ярус. Ширина ЯПФ станет равна 7. Это ширина яруса 8. Остальные ярусы уже. Результат см. на рисунке 5.

Продолжим сужение.

Тройку вершин ФУ2—ФУ3—ФУ4 можно сместить на один или два яруса вниз (на ярусы 9—10—11 или 10—11—12). Ширина ЯПФ станет равна 6. Это ширина ярусов 7 и 8. Остальные ярусы уже. Результат см. на рисунке 6.

С этих двух ярусов (7-го и 8-го) надо сместить вниз по одной вершине. Это возможно. Ниже находится один ярус шириной 3 и три яруса шириной 4. Например, пару вершин РС3—РС2 сместить вниз на два, три или четыре яруса. Сместим их на ярусы 9—10. Результат см. на рисунке 7.

Теперь ширина ЯПФ стала равна 5. Это ширина ярусов 4, 7, 8, 10.

Дальнейшее сужение возможно только для яруса 4. С него можно спустить вниз вершину ВГ или ВЛ.

Для ярусов 7, 8, 10 сужение невозможно. Спустить с них вершины можно только на ярусы, которые имеют ширину 4. Спуск туда еще одной вершины приведет к увеличению их ширины до 5.

Таким образом, дальнейшее сужение ЯПФ невозможно.

Минимальная ширина ЯПФ равна 5. Следовательно, минимальное количество кротов, которые могут построить такой лабиринт за 12 ходов, равно 5.

Ответы на вопросы задания выглядят так:

1. Сколько времени потребуется одному кроту, чтобы вырыть весь подземный ход? Ответ обосновать.

Ответ. 46. Сложить длину всех отрезков. Одному кроту придется пройти их все по очереди.

2. За какое минимальное время можно вырыть этот подземный ход, если в рытье может участвовать неограниченное количество кротов? Ответ обосновать.

Ответ. 12 (или 13 при невнимательном чтении условий).

Обоснование.

См. ярусно-параллельную форму на рисунке 4.

Задание намеренно сформулировано таким образом, что при невнимательном чтении может спровоцировать ошибку понимания. Из формулировки следует, что два крота не могут находиться рядом. Но не следует, что два крота не могут одновременно рыть один и тот же отрезок с двух сторон. Если эту возможность не учитывать, то критический путь будет равен 13, если учитывать, то 12.

Самая дальняя точка от входа в подземелье — точка на отрезке ЛМ на одну клетку выше точки М.

Длина пути АБЕЖЗИЛМ = 13. Длина пути АБЕЖЗКОМ = 11.

Остальные пути — меньше или равны:

АФХ = 8,

АФУТ = 12,

АБЕРПУТ = 12,

АБЕРПО = 9,

АБЕЖЗКО = 7.

Через 10 дней один из кротов выйдет в точку Л сверху и двинется вниз.

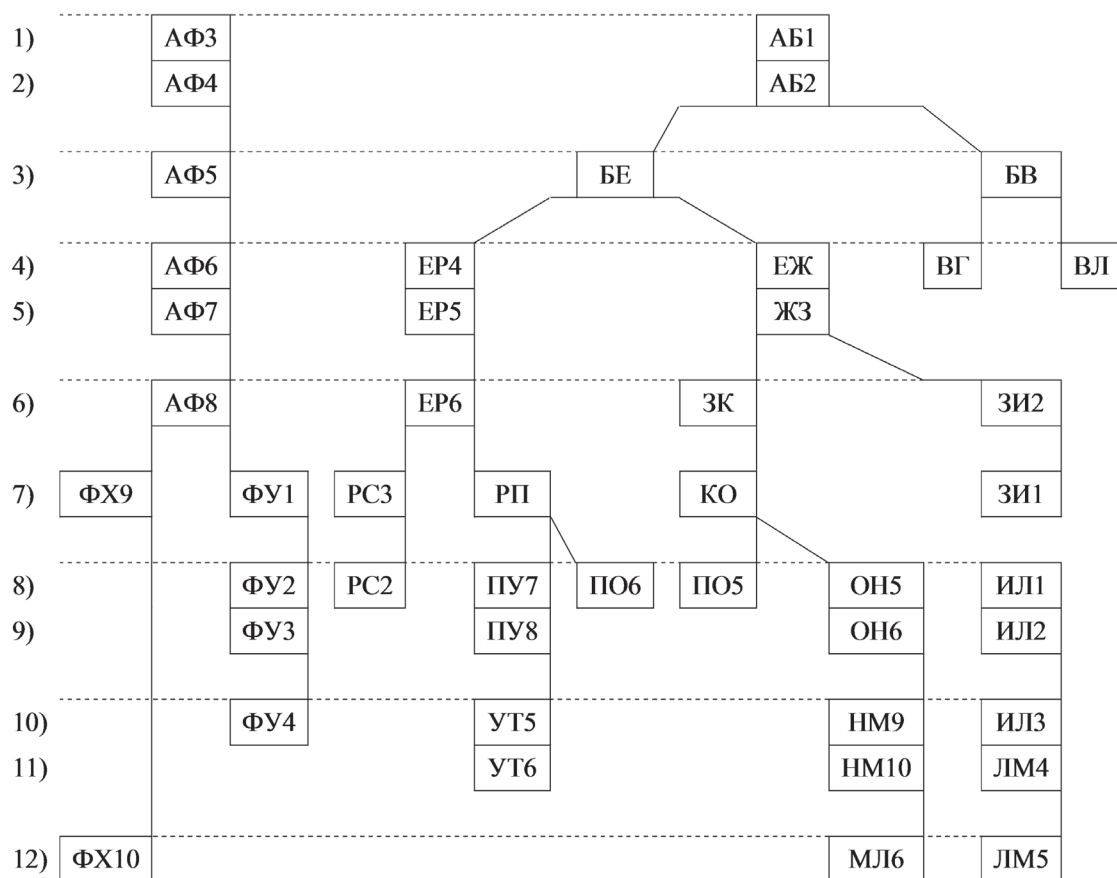


Рис. 5. Ярусно-параллельная форма для задачи про кротов, сжатая до ширины 7

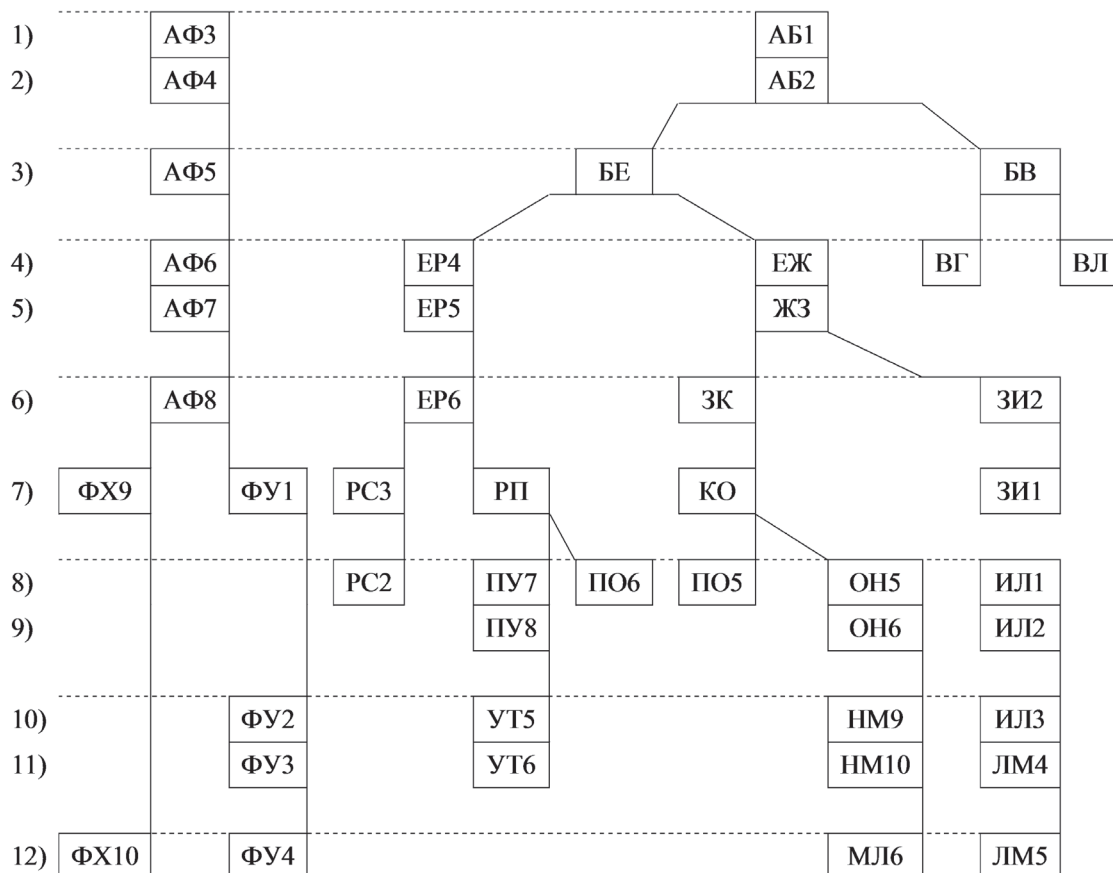


Рис. 6. Ярусно-параллельная форма для задачи про кротов, сжатая до ширины 6

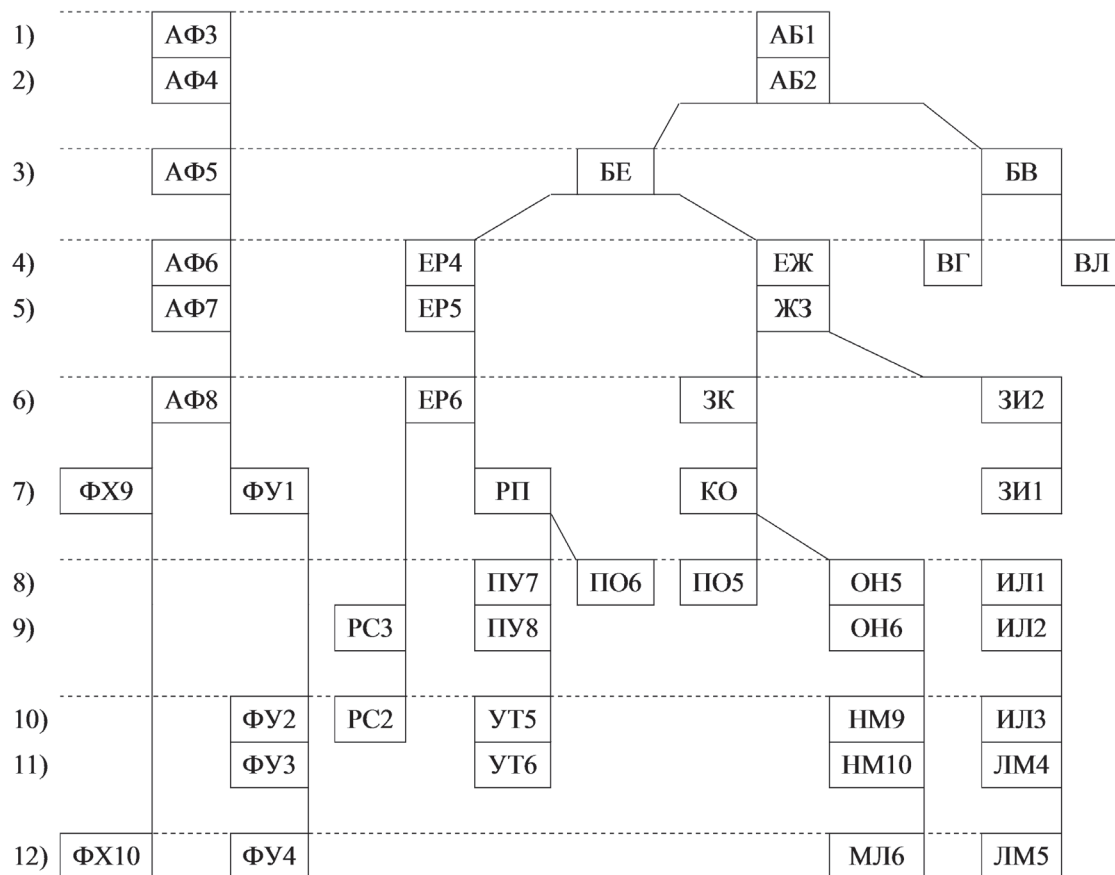


Рис. 7. Ярусно-параллельная форма для задачи про кротов, сжатая до ширины 5

Через 11 дней другой крот выйдет в точку М слева и двинется вверх.

К этому времени верхний крот уже пройдет одну клетку вниз. Встретятся два крота через день в точке, лежащей на одну клетку выше точки М.

Таким образом, длина критического пути равна 12.

Если возможность одновременного копания одного отрезка с двух сторон не учитывать, то критическим будет путь АБЕЖЗИЛМ = 13. При наличии обоснования этот ответ может быть принят, но оценка за него должна быть снижена.

Дополнительная сложность связана с тем, что самая дальняя точка сознательно не обозначена буквой.

3. Какое минимальное количество кротов должно участвовать в рытье, чтобы вырыть подземный ход за минимальное время? Ответ обосновать.

Ответ. Достаточно пяти кротов.

Обоснование. См. ЯПФ на рисунке 7.

4. За какое время этот подземный ход смогут вырыть четыре крота? Ответ обосновать.

Ответ. За 13 дней.

Обоснование. Для ответа на этот вопрос надо ярусно-параллельную форму с рисунка 7 сузить до ширины 4 и определить ее новую высоту. Покажем, как это можно сделать.

Содержательные рассуждения будут выглядеть следующим образом.

Первые два дня будут задействованы по два крота, на третий день — три. На четвертый день будут работать все четыре крота, и при этом образуется «задолженность»: одна вершина ЯПФ будет не отработана. Но на пятый

день эта задолженность будет ликвидирована: три крота отработают вершины пятого яруса, а один — непройденную вершину четвертого яруса.

Все последующие ярусы имеют ширину 4 или 5. Так что при их прохождении будут постоянно заняты все четыре крота. При этом при прохождении ярусов, имеющих ширину 4, будет образовываться «задолженность» в один блок на каждом ярусе. Таких ярусов три: 7-й, 8-й и 10-й. Для ликвидации «задолженности» в три блока четырем кротам понадобится еще один день.

Формальные преобразования графа ЯПФ таковы.

Вершина ВЛ смещена с яруса 4 на ярус 5.

С яруса 7 на ярус 12 вниз смещена вершина ФХ9. Чтобы освободить ей место, вершина ФХ10 была сдвинута вниз на вновь образованный ярус 13.

С яруса 8 на вновь образованный ярус 13 была смещена вершина ПО5, а с яруса 10 — вершина РС2.

Результат сужения — ЯПФ шириной 4 — показан на рисунке 8.

5. Сколько кротов надо, чтобы прорыть этот ход за 10 дней? Ответ обосновать.

Ответ. Нисколько. Это невозможно. Длина критического пути равна 12.

В настоящей статье мы обсудили особенности задач на обход графа по сравнению со «строительными» задачами (отсутствие статически определенного отношения следования вершин/дуг) и разобрали применение ярусно-параллельных форм для решения таких задач. В данном случае ЯПФ показали себя как очень эффективный меха-

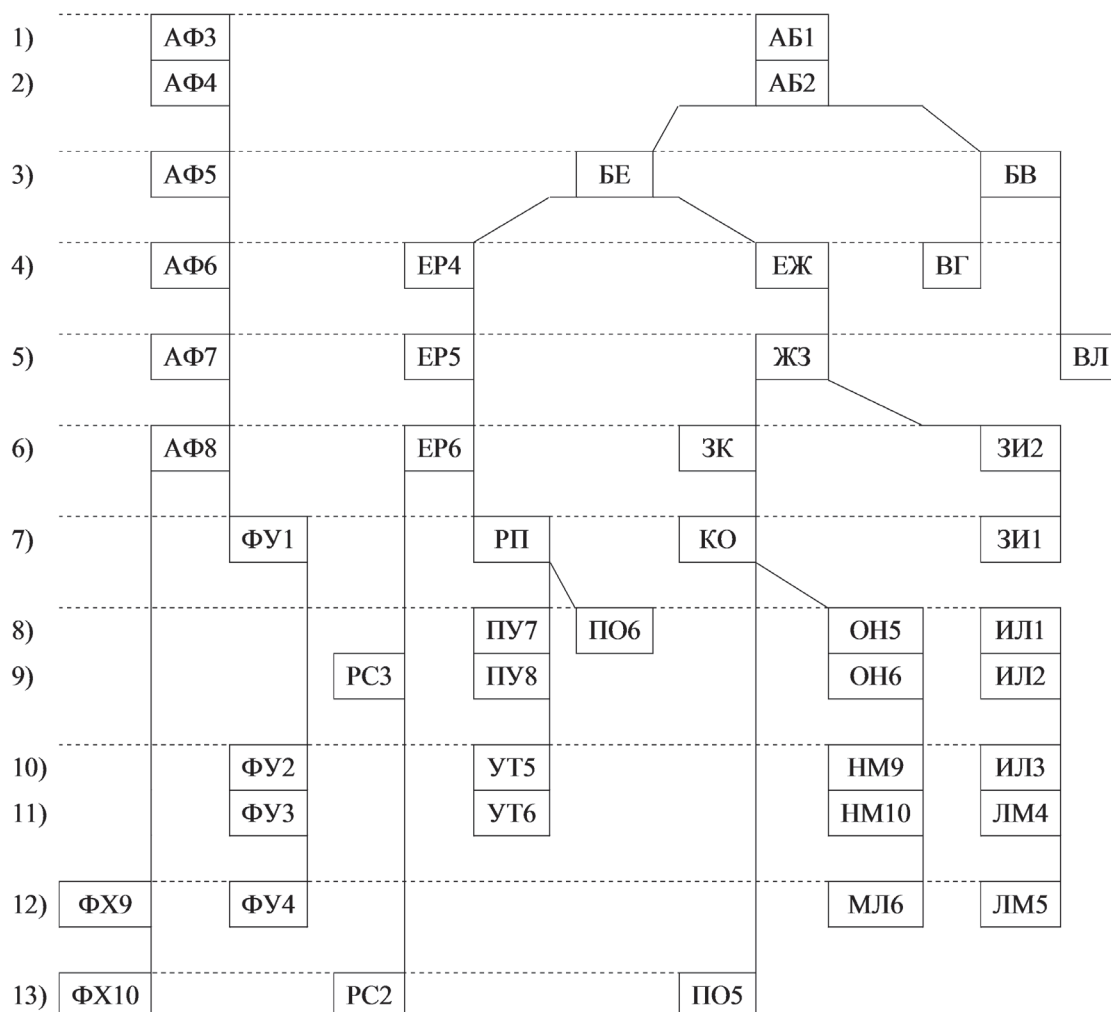


Рис. 8. Ярусно-параллельная форма для задачи про кротов, сжатая до ширины 4

низм. Построение параллельной формы алгоритма автоматически сняло все трудности, которые имелись в задаче.

В конкурсах «ТРИЗформашка-2016» и «ТРИЗформашка-2017» в качестве задач на ЯПФ также использовались «лабиринтные» задачи. Но со своими особенностями-усложнениями. В «ТРИЗформашке-2016» усложнением стало введение в задачу исполнителей с разной производительностью труда и переход от однородного графа к неоднородному (разные точки внутри лабиринта неравноправны). В «ТРИЗформашке-2017» усложнение выразилось в переходе от одноуровневого лабиринта к многоуровневому. Разбор этих задач — тема отдельного разговора.

Список использованных источников

1. Воеводин В. В. Вычислительная математика и структура алгоритмов: 10 лекций о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно, чтобы успешно преодолевать эти трудности. 2-е изд., стереотип. М.: Изд-во Московского университета, 2010.
2. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
3. Иванова Н. Г., Плаксин М. А., Русакова О. Л. Задачи на параллельное программирование в конкурсе «ТРИЗформашка-2013» // Информационные технологии в образовании. XXIII Международная конференция-выставка: Сборник трудов.

Ч. П. М.: Издательский отдел факультета ВМК МГУ имени М. В. Ломоносова, 2013.

4. Иванова Н. Г., Плаксин М. А., Русакова О. Л. Конкурс «ТРИЗформашка» как площадка для апробации заданий на параллельное программирование // Информатика в школе: прошлое, настоящее и будущее. Материалы Всероссийской научно-методической конференции по вопросам применения ИКТ в образовании, 6–7 февраля 2014 г. Пермь: Перм. гос. нац. иссл. ун-т, 2014.

5. Иванова Н. Г., Плаксин М. А., Русакова О. Л. ТРИЗформашка // Информатика. 2010. № 5.

6. Плаксин М. А. Пропедевтика параллельных вычислений в школьной информатике. Параллельная форма алгоритма // Информатика в школе. 2016. № 10.

7. Плаксин М. А., Иванова Н. Г., Русакова О. Л. Информатика: учебник для 3 класса: в 2 ч. М.: БИНОМ. Лаборатория знаний, 2013.

8. Плаксин М. А., Иванова Н. Г., Русакова О. Л. Информатика: учебник для 4 класса: в 2 ч. М.: БИНОМ. Лаборатория знаний, 2013.

9. Плаксин М. А., Иванова Н. Г., Русакова О. Л. Набор заданий для знакомства с параллельными вычислениями в конкурсе «ТРИЗформашка» // Преподавание информационных технологий в Российской Федерации. Материалы Тринадцатой открытой Всероссийской конференции «ИТ-Образование — 2015» (г. Пермь, 14–15 мая 2015 г.). Пермь: Перм. гос. нац. иссл. ун-т, 2015.

10. Романовский И. В. Дискретный анализ. 3-е изд., перераб. и доп. СПб.: Невский Диалект; БХВ-Петербург, 2003.