

Проблема полной автоматизации при составлении школьного расписания

Пиликов Николай Петрович
www.mnogosmenka.ru

20 августа 2013 г.

Аннотация

Данная статья знакомит читателя с уникальным, недавно появившимся, алгоритмом составления школьного расписания. Сообщаются результаты тестирования единственной в мире программы, которая может не составлять, а составить такое расписание в полностью автоматическом режиме. Опираясь на результаты десятков миллионов тестов (построенных школьных расписаний) развенчивается миф о невозможности составления школьного расписания без участия человека. Делаются прогнозы дальнейшего развития этого программного средства. Обсуждается бизнес-модель SaaS его использования.

Для понимания основного содержания статьи не требуется какой-либо специальной математической подготовки, таким образом статья адресована широкому кругу заинтересованных читателей.

Содержание

1	Введение	2
2	Поколения ПО составления учебных расписаний	3
3	Нужны ли такие программы?	6
4	Немного о сложности решаемой задачи	8
5	Рынок ПО составления учебных расписаний	10
6	«Древнеегипетский» алгоритм решения задачи	13
7	Тестирование алгоритма	16
8	Логическая модель будущего ПО	18
9	Проблемы с бизнес-моделью	20
10	Бизнес-модель SaaS использования ПО	22
11	Вместо заключения	22

1 Введение

За последнее десятилетие в Российской Федерации было защищено не менее дюжины диссертаций по темам связанным с задачей составления учебных расписаний. За предыдущее, перед этим, десятилетие количество защищенных диссертаций не меньше. Хотя в основном диссертации защищаются на звание кандидата технических наук и рассматриваются задачи составления расписания занятий для высшего учебного заведения, тем не менее данный факт свидетельствует о том, что к задачам составления школьного расписания проявляет внимание все больше и больше исследователей. Возможно этот поток работ связан с постоянным прогрессом и всеобщей доступностью вычислительной техники. Действительно на наших глазах происходят по истине удивительные процессы. Еще каких-нибудь двадцать пять лет назад, такую электронно-вычислительную машину, как ЕС1066, могло себе позволить приобрести только крупное, как правило оборонное, предприятие. Располагался такой компьютер в помещении площадью до нескольких сот квадратных метров, оборудованном мощной системой бесперебойного энергопитания и системой поддержки микроклимата. Такие электронно-вычислительные машины в первую очередь предназначались для решения уникальных научно-технических задач, оказывающих влияние на обороноспособность страны.¹ Сегодня у многих дома на письменных столах стоят персональные компьютеры. Но только вдумайтесь. Оперативная память такого персонального компьютера по сравнению с выше упомянутым гигантом больше в 125 – 250 раз. Быстродействие выше, более чем в 1000 раз. И это не описка. Более чем в тысячу раз.

Сегодня для образованного человека, который никогда не сталкивался (или еще не сталкивался) с проблемой автоматического составления расписания в школе, включая людей которые называют себя IT-специалистами уровня эксперта, утверждение о том, что не существует программного обеспечения которое могло бы решить данную задачу (составить расписание в автоматическом режиме без участия человека) кажется просто абсурдным. Логика их рассуждений очень проста: — «При таком развитии техники, этого просто не может быть». Вторым аргументом, как им кажется бьющим «наповал», является отсылка к любой поисковой системе имеющейся в Интернете. По поисковому запросу: — «Программа для составления расписания уроков», открываются тысячи ссылок. Очень быстро находятся программы обещающие качественное решение данной задачи в полностью автоматическом режиме.

Таким образом, мы сегодня имеем —

Во-первых. Массовое заблуждение не специалистов (а точнее людей, не посвятивших несколько десятков часов, работе с данными программами) о том что эту задачу легко решить и рынок до отказа забит такими программами.

Во-вторых. Огромную армию завучей, попытавшихся применить такие программы в своей работе, и понявших, что их обманули, унизили, оскорбили, надругались и т.п.

В-третьих. Широко распространенный миф среди людей, которые считают себя специалистами в области разработки таких программ, о том, что задачу составления школьного расписания в автоматическом режиме не возможно решить в принципе. И это строго доказанный математический факт.

Данная статья посвящена, с одной стороны, развеиванию первого массового заблуждения о легкости решаемой задачи, и, с другой стороны, опровержению мифа о том, что данная задача не поддается решению в принципе.

¹Поэтому весь этаж где располагалась такая ЭВМ обтягивали мелкой металлической сеткой, дабы исключить возможность «электронного подглядывания» со стороны заклятых врагов советской власти.

2 Поколения ПО составления учебных расписаний

Первые публикации на тему применения вычислительной техники с целью автоматизировать составление расписания занятий появились в начале 60-х годов предыдущего столетия, таким образом задача составления учебного расписания с помощью вычислительной техники имеет достаточно продолжительную историю.² За почти 50 лет интенсивных исследований была проведена огромная интеллектуальная работа тысяч специалистов по всему миру.

Совершенно не удивительно то, что программы для составления школьного расписания появлялись и совершенствовались по мере развития вычислительной техники. Поэтому обратимся (естественно в телеграфном стиле) к весьма условным периодам этого развития.

Не вдаваясь сильно в исторические изыскания и не рискуя сильно ошибиться, появление компьютера (электронно-вычислительной машины — ЭВМ) можно к 1945 году. Это появление (опять же не рискуя сильно ошибиться) можно отнести на счет потребности в вычислениях для военных нужд. Одной из первых задач которые решались на первых ЭВМ была задача составления баллистических таблиц для артиллерии и авиации. Не последнюю роль в потребности военных играла задача изучения атомного и термоядерного взрыва. В силу выше указанных причин, сам факт существования ЭВМ и принципы ее работы по началу оставались засекреченными.

Примерно десять лет понадобилось на то, что бы довести сведения о «тактико-технических характеристиках» первых ЭВМ до широкого круга узких специалистов — математиков занимающихся численными методами. Результат не заставил себя долго ждать. С 1955 года наблюдался взрывообразный рост такой отрасли научных знаний, как прикладная математика. Сотни и тысячи практически важных задач стали предметом исследования математиков с применением электронно-вычислительной техники, что повлекло за собой разработку совершенно новых численных методов решения этих задач. По той причине, что стоимость компьютеров была совершенно не сопоставима с тем экономическим эффектом которые они могли принести для гражданского промышленного предприятия единственными пользователями этой техники были военные и весьма узкий круг ученых. Другими словами те люди которые не знали слов — дорого, затраты или словосочетания — экономический эффект.

Но время шло. Технологии производства и проектирования вычислительной техники развивались стремительными темпами. В результате, производительность компьютеров росла не бывальными шагами, а их стоимость быстро снижалась. Цены на компьютеры от астрономических неуклонно приближались к земным (хотя и еще заоблачным). К 1965 году весьма заметно вырос круг ученых которым для исследований была доступна вычислительная техника. К этому времени (начало шестидесятых годов), как было отмечено выше, и относятся первые публикации на тему составления на больших ЭВМ школьного расписания. Совершенно естественно, что работы по началу имели постановочный характер, а позже теоретический.

Около пятнадцати лет понадобилось на то, что бы придумать все что можно было легко придумать в отношении задачи составления школьного расписания. Этот период (с 1965 до 1980 года) вызывает резкие смешанные чувства. С одной стороны, были предложены красивые и оригинальные математические модели задачи составления школьного расписания (вершинная раскраска графов, реберная раскраска графов), а с другой сто-

²Самой же задаче составления учебного расписания (без помощи вычислительной техники) по всей видимости, уж не как не менее трехсот лет.

роны, вне всякого сомнения эти модели следует отнести к весьма упрощенному варианту задачи. Другими словами, полностью задача не была решена и даже не сформулирована в деталях. Более того, в 1976 году появилась работа израильских математиков где, по их мнению, была доказана принципиальная трудность решения задачи составления школьного расписания [20].

Итак, к 1980 году не смотря на то, что производительность ЭВМ постоянно повышалась а их стоимость постоянно снижалась, в результате чего уже и гражданские промышленные предприятия перешли в категорию активных пользователей вычислительной техники, наша задача по прежнему оставалась до конца не решенной, а вычислительная техника для главного пользователя — школ, оставалась не доступной. Вот пожалуй к этому периоду можно было бы отнести программы первого поколения для составления расписания занятий.

В силу выше указанных двух причин (труднорешаемость задачи и не доступность вычислительной техники для конечного потребителя) интерес к автоматическому составлению расписания занятий заметно ослабел (а может быть даже и полностью угас). Высшие учебные заведения, использующие данное программное обеспечение, взяли крен от собственно составления расписаний занятий к учету и контролю успеваемости студентов. Еще раз подчеркнем, что администрации школ в подавляющем большинстве даже и не догадывались о существовании таких программ.

Однако, к этому времени (естественно за рубежом) среди некоторых «яйцеголовых» студентов возникает мода на конструкторы из радиодеталей. Забрехала эра персональных компьютеров. мода оказалась весьма прилипчивой и круг «яйцеголовых» неуклонно расширялся. Весьма вероятно, что конструкторы из радиодеталей так и остались бы уделом кучки «не нормальных» если бы самый крупный в то время производитель пишущих машинок, а за одно и самых распространенных в то время ЭВМ, американская корпорация IBM, приблизительно в 1985 году, прозорливо не смекнула бы, что эти конструкторы, если им придать форму пишущей машинки, могли бы эти пишущие машинки заменить. Причем не просто заменить, а сделать из пишущей машинки сверх интеллектуальную пишущую машинку составив конкуренцию «свинцовым технологиям» в издательском деле. Разумеется в то время ни-кто, разве что кроме самых прозорливых, и предположить не мог, что конструкторы из радиодеталей когда нибудь смогут составить конкуренцию настоящим вычислительным устройствам.

Тем не менее, жребий был брошен, началось массовое производство убийц пишущих машинок. Не долго себя заставили ждать и идеи производства сначала «два в одном» (пишущая машинка плюс помощник бизнесмена — электронная таблица), потом «три в одном» (еще плюс бухгалтерская программа), потом «четыре в одном», и так далее, и так далее. Вчерашние студенты по моноению волшебной палочки стали превращаться в миллиардеров, а бывшие конструкторы из радиодеталей все больше и больше стали походить на настоящие электронно-вычислительные машины. В технический и деловой язык вошла уважительная аббревиатура «Пи-Си» (PC), что означало персональный компьютер и уже в начале 90-х годов XX столетия ни у кого не вызывало сомнений, что у них на столе располагалась не игрушечная, а совершенно настоящая электронно-вычислительная машина.

Противоположные тенденции — взрывообразный рост производительности бывших игрушек, с одной стороны, и стремительное падение их цены, с другой стороны, сделали свое дело. В некоторых продвинутых школах на столе у руководителей появились, по сегодняшним меркам, здоровущие мониторы, которые как живой укор вопили: — «Наполни меня нужным программным обеспечением». Не мудрено, что вспомнилась уже казалось

бы совсем забытая идея составления расписания учебных занятий. Тысячи любителей легкой наживы бросились писать программы для школ, гарантируя полную автоматизацию всего, что только попадет под руку. К этому периоду, пожалуй можно отнести программы второго поколения, автоматизирующие процесс составления школьных расписаний.

В девяностых годах прошлого столетия индустрия персональных компьютеров переживала невероятный рост. Производительность персоналок увеличивалась вдвое чуть ли не каждый год и каждый год приносил инновационные программные продукты. У работающих в этой сфере «на ботинках рвались подметки». А программы по составлению школьных расписаний что-то ни-как не хотели правильно работать . . .

Сейчас конечно трудно сказать, знали или нет производители программ для составления школьного расписания о том наследстве которое им оставили их предшественники в 1965 – 1980-х годах прошлого столетия и о предупреждении израильских математиков 1976 года о том, что эта задача труднорешаемая, но факт остается фактом — администрация учебных заведений потихоньку списывала старые добрые пишущие машинки заменяя их персональными компьютерами. Расписание по прежнему, за не большими исключениями, составлялось в ручную.

К началу XXI века вместе с окончательным доминированием операционных систем с графическим пользовательским интерфейсом, приходит и конец программам второго поколения по составлению школьного расписания, которые использовали псевдографический интерфейс ушедшей операционной системы MS-DOS. Индустрия персональных компьютеров благополучно прекратила свое бурное развитие и перешла к пресловутой «стабильности». Персональная вычислительная техника преодолела рубеж производительности больших ЭВМ середины 80-х годов прошлого века, все было готово для разработки программ третьего поколения.

И действительно, в самом конце прошлого столетия не сметное количество производителей, в очередной раз но уже, как им казалось, на новом техническом и технологическом уровне взялись за разработку программ составления школьного расписания. На фоне прекращения заметного (хотя и плавного) роста производительности персональных компьютеров, стабилизации идей в области программного обеспечения развивались программы, которые можно было бы отнести к программам третьего поколения. Основной особенностью этих программ, как нам кажется, является то, что они могли разрабатываться учитывая как ошибки, так и оригинальные находки предшественников. Здесь в первую очередь имеются в виду разработчики девяностых. С математическими результатами шестидесятых, семидесятых и восьмидесятых дело обстоит проще. Если ты о них знаешь, то используешь, если не знаешь, то «придумываешь велосипед за ново». Другой особенностью является то, что эти программы разрабатывались с использованием нового на то время — графического интерфейса пользователя. Нет сомнений в том, что графический интерфейс предоставляет разработчику принципиально большие возможности по сравнению с псевдографическим (текстовым). Но в этом, в тоже время, кроется и опасность. Если мы начнем сравнивать имеющиеся на рынке (в использовании) программы составления школьного расписания, то обнаружим совершенно потрясающее разнообразие способов формирования (ввода) исходных данных, необходимых для расчета, хотя с математической точки зрения все программы делают (или по крайней мере, должны делать) совершенно одно и тоже. Таким образом, на качество программ составления школьного расписания стало оказывать существенное влияние логичность и удобство пользовательского интерфейса.

Сегодня (2013 год) стоит отметить, что по сравнению с программами девяностых,

программы третьего поколения (нулевых) весьма «поумнели». Оптимизма у разработчиков заметно поубавилось. Обещать полную автоматизацию всего, что попало под руку, уже ни-кто (или почти ни-кто) не берется.

Многие из проектов начатых на излете девяностых к настоящему времени прекратили свое существование в силу их не востребованности. Другие же продолжают развиваться и совершенствоваться. Третьи на протяжении последних десяти лет застыли в своем развитии. Но как ранее было отмечено, об окончательном и бесповоротном решении задачи составления школьного расписания сегодня говорить еще рано.

3 Нужны ли такие программы?

Обычно, говоря о пользе (необходимости) использования программы по автоматизированному построению расписания, указывают такой фактор, как — на порядок сокращение трудозатрат (времени) завуча при составлении учебного расписания. Часто указывают на то, что расписание с помощью компьютера может быть получено лучшего качества. Хотя этот довод, учитывая сказанное чуть ниже, является не без спорным. По нашему мнению следует согласиться с тем, что расчет расписания с помощью компьютера позволит, кроме экономии времени и получения лучшего качества расписания, с одной стороны, исключить субъективные оценки и личные симпатии завуча по отношению к учителю (части учителей), при составлении расписания, в том числе и при распределении педагогической нагрузки, а с другой стороны, позволит полностью исключить незаслуженные обвинения в адрес завуча со стороны учителей, в таких субъективных оценках и симпатиях, поскольку очевидно, что компьютер «лицо не заинтересованное» (во всем «виноват» компьютер). Тем самым, расчет распределения педагогической нагрузки и расписания на компьютере, может улучшить психологический климат в учительском коллективе (соблюсти принципы справедливости и равноправия), точно так же как улучшает настроение игроков футбольной команды судья матча после того как разыгрывает право первого удара по мячу с помощью жребия.

В 2001 году компанией «Хронобус» было проведено анкетирование почти 1000 московских школ по вопросу необходимости создания и внедрения АРМ(а) «Расписание» [18]. Результаты анкетирования показали, что у всех школ имеется искреннее желание пользоваться такой программой, но никто этого не делает. Причем причиной дружного игнорирования подобных средств автоматизации является вовсе не отсутствие необходимой техники или денег, а качество предлагаемых на рынке программ. Фраза: — «Если бы мне предложили увеличить зарплату в полтора раза, за то что я пользуюсь такой программой составления школьного расписания, то я бы от этого предложения отказалась» была не редкостью. Другими словами по мнению завучей программы по составлению школьного расписания являются программным обеспечением с отрицательной стоимостью.

Сегодня по прошествию двенадцати лет с момента выше указанного анкетирования у потенциальных пользователей программ по составлению учебных расписаний — завучей школ, к таким программам, еще в большей степени и не без основательно сформировалось стойкое негативное, а часто и агрессивное отношение.³

Вводящая в заблуждение реклама о навязываемом «информационном пространстве школы» формирует представление об авторах этого пространства, как о мошенниках сбывающих тухлый товар. По мнению завучей школ с большим стажем работы, *практи-*

³Зафиксированы случаи когда завучи — в общем то, культурные и воспитанные люди, услышав словосочетание: — «Программа для составления школьного расписания» мгновенно переходили на матершину.

ка показывает, что данные программы могут быть использованы только в качестве инструмента для первоначальной расстановки предметов с последующей ее ручной доработкой, а так же сохранения информации и вывода ее на печать. После автоматизированного распределения предметов (программа, как правило, расставляет от 40 до 70 %) учитывать гигиенические требования к расписанию уроков практически не возможно, так как приходится не только доставлять оставшиеся не расставленные предметы, но и существенно изменять (до 60 %) автоматизированную расстановку предметов по принципу «лишь бы расставить».

Умудренные опытом мастера своего дела, рекомендуют начинающим при составлении расписания учебных занятий пользоваться десятком – другим советом, проверенных многолетним опытом и практикой, используя при этом, вместо компьютера, макеты таблицы расписания уроков из листов картона, цветной бумаги, широкого прозрачного скотча, клея, кармашков и т.п. И они безусловно правы. Использование компьютера в режиме обычного редактора (как всем знакомого текстового редактора) или использование программ, заводящих процесс расстановки занятий в тупиковые ситуации, когда ни одного занятия не возможно пристроить в сетку расписания теоретически, ничего кроме не оправданных затруднений, неудобств и озлобленности принести не может.

Ожидание пользователей таких программ (завучей) сомнений не вызывает. По их мнению программы составления школьного расписания после ввода всех исходных данных, должны в полностью автоматическом режиме, составлять расписание по своему качеству превосходящее расписание составленное в ручную. Неадекватность ожиданий пользователей и получаемого результата от таких программ, порождает агрессивное отношение пользователей к этим программам и вместе с ними к автоматизаторам «раздвигающим информационное пространство школы».

Следует отметить, что разработчики программ по составлению школьного расписания в ходе «естественного отбора» поделились на три группы. Первая группа публично отстаивает точку зрения на то, что задачу автоматического расчета школьного расписания не возможно решить в принципе. И поэтому они «не будь бестолочами» даже и не пытаются этого сделать. А те кто пытается, по их мнению полные невежды. «У нас не программа расчета школьного расписания, а редактор школьного расписания. Мы не строим расписание вместо человека, а помогаем человеку построить самостоятельно (в ручном режиме) расписание» — с гордостью заявляют они.

Вторая группа разработчиков декларирует в качестве цели — полную автоматизацию построения школьного расписания, но в своих рекламных материалах и руководствах пользователя дипломатично умалчивает о достижении поставленной цели. «Наша программа может строить расписание в автоматической режиме, в ручном режиме и в смешанном (полуавтоматическом) режиме» — не обманывая пользователей констатируют они. Внимание потенциальных пользователей на том, что конь может пить воду из реки, но не может ее выпить, а программа может строить расписание в автоматическом режиме, но не может его построить, эти разработчики не акцентируют. По нашему мнению это весьма взвешенная и достойная позиция, которая, несмотря на маленькую хитрость, может вызвать только уважение. Или, по крайней мере, не вызывает агрессивного отношения к разработчикам со стороны пользователей.

И наконец, третья группа разработчиков. «Введите исходные данные, нажмите кнопку расчет, и через несколько минут вы гарантированно получите расписание с расстановкой всех без исключения занятий. Ограничений на размерность задачи никаких. Классов хоть — 99. Преподавателей хоть — 216. Совместителей хоть половина. Разделим класс на группы хоть до 256-ти групп. Ограничения для преподавателей и предметов любые.

Каждый преподаватель выбирает для себя удобные для него рабочие дни и часы. Никаких окон у преподавателей. Занятия по предметам проводятся только в разрешенные для этих предметов часы. Неукоснительное соблюдение параллелей. Каждому предмету присваиваются баллы сложности. Точное соответствие с санитарными нормами распределения суммарной сложности предметов во времени гарантировано.» — без стеснения заявляют они. Кстати говоря, на такой незатейливый ход идут разработчики наиболее беспомощных программ в плане автоматического составления расписания и к тому же неряшливых на вид (хотя есть одна, на вид весьма привлекательная). Такие программы в Microsoft метко окрестили — food dog — «собачья еда». Трудно сказать, что именно движет людьми, идущими на прямой и бесхитростный обман потребителей. Этот обман всегда становится очевидным при первом же вводе учебного плана школы в программу. По Российскому законодательству, в соответствии со ст. 179 Гражданского кодекса РФ, сделки совершенные под влиянием обмана могут быть признаны судом недействительными, при этом обманщик возвращает обманутому все полученные деньги, возмещает обманутому реальный ущерб и в добавок еще должен перевести в доход государства такую же сумму, что им была получена от продажи программы.

4 Немного о сложности решаемой задачи

Стоит сказать два слова о сложности решения задачи составления школьного расписания.⁴ Квалифицированным пользователям персонального компьютера, уверовавшим в его всемогущество, кажется, что задача составления школьного расписания не чуть не сложнее задачи создания, например, качественного редактора видеофильмов или звукового редактора. Однако, как упоминалось ранее, число исследователей⁵ так или иначе изучавших эту задачу трудно поддается подсчету. Среди них десятки докторов технических и физико-математических наук,⁶ сотни кандидатов наук, не только технических, но и физико-математических, не говоря уже о тысячах рядовых любителях математических головоломок, безусловно включающих в себя многочисленную армию студентов технического и физико-математического профиля обучения. Среди исследователей задачи составления школьного расписания можно упомянуть и двух академиков — В.С.Танаева⁷ и В.С.Михалевица,⁸ можно было бы назвать и зарубежных ученых с мировым именем.

⁴Здесь мы не будем останавливаться на теории NP-трудных задач, поскольку обсуждение этого вопроса увело бы читателя далеко в сторону от интересующей нас темы, а так же носило бы явно преждевременный и поверхностный характер. Заинтересованному же читателю, можно порекомендовать обратиться к пожалуй самому цитируемому в нашей стране изданию по данной теме [7]. Для полного понимания данной статьи под NP-трудными задачами можно понимать — практически нерешаемые задачи, хотя это не совсем точный «перевод».

⁵Имеются ввиду русскоязычные публикации, которых по сравнению с англоязычными не так много. Скорее всего их число не превышает общего вклада Российской Федерации в сфере высоких технологий, который оценивается в пределах 0.4 – 0.6 % (от нуля целых четырех десятых процента до нуля целых шести десятых процента) от общемирового.

⁶Правда, физико-математических наук на порядок меньше.

⁷Танаев Вячеслав Сергеевич (1940 – 2002) — белорусский математик, директор НИО «Кибернетика» НАН РБ, доктор физико-математических наук (1978), профессор (1980), действительный член Национальной академии наук Беларуси (2000). Область научных интересов: исследование операций, теория расписаний, методы оптимизации.

⁸Михалевиц Владимир Сергеевич (1930 – 1994) — украинский математик и кибернетик, академик АН Украины, академик РАН (1991; академик АН СССР с 1984). Труды по теории оптимальных статистических решений, системному анализу, теоретической и экономической кибернетике. Государственная премия СССР (1981).

Кроме ученых не обходили своим вниманием задачу составления школьного расписания выдающиеся бизнесмены⁹.

И тем не менее, не смотря на, без преувеличения, титанические усилия исследователей, говорить о полном и всеобъемлющем (или хотя бы удовлетворительном) решении задачи составления учебного расписания — не приходится.

В качестве подтверждения сказанного приведем цитату известного отечественного математика [14].

... Так как задача составления расписаний хорошо всем известна по школьному быту, то на каждом курсе находится один или несколько студентов, обуреваемых идеей алгоритмизации составления расписания занятий. Поэтому я вынужден предупредить, что это очень сложная задача. ... Существует особая наука - теория расписаний, изучающая и систематизирующая задачи такого рода, а так же различные приближенные методы их решения (на точные методы надежды почти нет). Особое место среди них занимают эвристические методы, в которых делаются попытки описать логику и технику действий диспетчера. ...

Интересно одно наблюдение. Но прежде приведем еще одну цитату [19].

Гипотезу четырех красок можно с полным основанием назвать «болезнью четырех красок», так как она во многом похожа на заболевание. Она в высшей степени заразна. Иногда она протекает сравнительно легко, но в некоторых случаях приобретает затяжной или даже угрожающий характер. Никаких прививок против нее не существует; правда, люди с достаточно здоровым организмом после короткой вспышки приобретают пожизненный иммунитет. Этой болезнью человек может болеть несколько раз, и она подчас сопровождается острой болью, но ни одного летального исхода зарегистрировано не было. Известен по крайней мере один случай передачи болезни от отца к сыну, так что, может быть, она наследственна.

Здесь выдающийся американский математик иронизирует над старинной задачей о раскраске политической карты в четыре цвета, где страны имеющие общую границу должны быть окрашены в разные цвета. Представляется, что все сказанное им, можно отнести и на счет задачи составления школьного расписания. Так, автору этих строк вздумалось, в меру своей возможности, отследить дальнейшую карьеру людей защитивших диссертацию по соответствующей теме. Казалось бы, новосостоявшемуся ученому «сам Бог» велел обратить свои научные достижения в деньги. То есть, каким-то образом вывести свое детище на рынок, поскольку практически всегда после защиты диссертации остается некая программа или часть автоматизированной системы по составлению расписания занятий. Так вот — нет. Все известные автору случаи защиты диссертации по этой теме заканчиваются одним — после защиты, диссертант бросает эту задачу и, как правило, начинает (или продолжает) карьеру преподавателя в вузе. Другими словами, приобретает к задаче составления учебного расписания пожизненный, стойкий иммунитет.

Заканчивая общие рассуждения о сложности решения задачи составления школьного расписания сошлемся еще на два мнения. Но прежде обратим внимание на то, кто это мнение высказывает.

Ни для кого не секрет, что некоторые школьные преподаватели информатики в припадках дидактических экспериментов поручают школьникам в качестве «домашнего задания» разработать программу составления расписания занятий для любимой школы.

⁹ Существует легенда [10] о том как, теперь самый богатый человек на планете, а тогда юный (шестнадцати летний) Билл Гейтс вместе со своим старшим другом Полом Алленом «решили» эту задачу, взамен получили за программирование машинное время и заработали 2400 долларов карманных денег.

Школьники, естественно, засучив рукава с энтузиазмом берутся за решение этой задачи. В качестве выхлопа от данной идеи в сети интернет можно встретить многочисленные рассуждения и теоретизирования по этому поводу выше указанного контингента. Чего только не придумывают и каких только мнений не высказывают пионеры . . . Не меньший ажиотаж данная тема вызывает и у людей с техническим образованием в попытках автоматизировать деятельность диспетчеров любимого вуза. Но эти мнения, мягко говоря, большого интереса не представляют. Профессиональные же математики, специалисты по теории расписаний, высказываются о проблеме составления учебного расписания чрезвычайно редко. Поэтому (или тем более) их мнение на этот счет представляется весьма интересным.

Итак. Сотсков Юрий Назарович, доктор физ.-мат. наук, профессор, главный научный сотрудник Объединенного института проблем информатики НАН Беларуси, Минск, один из виднейших специалистов в области теории расписаний, автор ряда монографий по теории расписаний. В своей статье [1] в частности пишет:

. . . С математической точки зрения задача построения оптимального расписания учебных занятий является достаточно сложной, поскольку она принадлежит классу так называемый NP-трудных задач. . . В данной статье показано, как раскраска вершин графа может применяться для построения расписания учебных занятий. . .

. . . Задача раскраски вершин графа является NP-трудной, и, следовательно, ее обобщение, описанное в разд. 2, также является NP-трудной задачей. . .

Далее. Лазарев Александр Алексеевич, доктор физ.-мат. наук, профессор, главный научный сотрудник Института проблем управления им. В.А.Трапезникова РАН, Москва, один из виднейших специалистов в области теории расписаний, автор ряда монографий по теории расписаний. В своей статье [6] в частности пишет:

. . . Задача составления учебного расписания является хорошо известной задачей комбинаторной оптимизации «Составление временных таблиц» (timetabling). Даже нахождение допустимого расписания является NP-трудной в сильном смысле проблемой. Поэтому при ее решении необходимо использовать математические методы решения задач комбинаторной оптимизации. . .

Короче: — «Сливай воду, суши весла, туши свет . . .»

5 Рынок ПО составления учебных расписаний

Рынок ПО составления учебных расписаний, который складывался вместе с рынком любого ПО для персональных компьютеров, представляется просто уникальным,¹⁰ или уж по крайней мере удивительным, или на худой конец весьма странным. Так в чем же его

¹⁰Утверждение про уникальность рынка ПО составления школьного расписания, пожалуй все же не верно. Очень похожая ситуация сложилась на рынке ПО автоматического перевода текстов (в частности технических, написанных человеком для человека, тексты на языках программирования, естественно, не в счет) с одного языка на другой, например, с английского на русский. Точно так же, эта задача весьма внятно была сформулирована не менее пятидесяти лет назад. Точно так же, решалась усилиями многих тысяч исследователей. До настоящего времени в попытках решить задачу прилагаются фантастические усилия. Приблизительно те же результаты — близкие к нулевым. Если программе автоматического перевода не предлагать специально подобранный текст (как порой делают разработчики подобного ПО в своих демоверсиях) понять из перевода: — «Это вообще о чем?», практически не возможно.

В качестве другого примера, можно было бы привести ПО распознавания речи. Судя по программам обучения иностранному языку, кое-какие результаты в этом направлении все же имеются (отличить правильное произношение от неправильного — удается), но до окончательного решения проблемы еще очень и очень далеко. Мечта о вводе в компьютер данных или команд с голоса, все еще остается мечтой.

уникальность или странность?

Вам когданибудь приходилось видеть такую рекламу: — «Купите наш пылесос, который не может всасывать пыль». Или такую: — «Все кастрюли которые мы вам можем предложить — дырявые». Или такую: — «Наш телевизор уникален — он никогда, ничего не показывает». А вот рекламу: — «Купите нашу программу для составления школьного расписания, которая не может его составить, но зато может составлять», нам приходилось видеть сколько угодно. «Ну купите, купите, купите. Наша программа может же составлять расписание. Она вам расставит *почти* все занятия, ну а уж остальные, какнибудь сами. Выбираться из тупика, это же так интересно. Ну хоть за 15 долларов. Это же не большие деньги, мы ведь столько работали . . . ».

Так сколько же стоит пылесос, который не всасывает пыль, дырявая кастрюля или телевизор, который никогда, ничего не показывает? Прежде чем ответить на этот нелегкий вопрос попробуем оценить количество потенциальных покупателей и сравнить его с количеством школ (завучей) которые уже совершили свою покупку.

Демографами установлено, что около 16% населения развитых стран являются школьниками. Именно эта цифра используется при строительстве новых школ в новых районах застройки [16]. Далее будем производить арифметические вычисления на примере Российской Федерации (родина все-таки). Итак, население составляет примерно 140 млн. человек. Таким образом, школьников примерно 22 млн. Школ около 50 тыс. Значит среднее количество учеников в школе 440 человек. Но это среднее количество. Известно [16], что за последние 60 – 70 лет типовыми проектами школ, считались школы на 1000 – 1400 учеников. Отсюда вывод — имеется огромное количество школ с количеством учеников на много меньшим, чем наша средняя цифра — 440 человек. Очевидно, что это школы в сельской местности или в очень не больших городах. Отсюда же, более сильный вывод — огромному количеству школ, программы для составления расписания учебных занятий не нужны в принципе.

Оценить количество школ, которым такие программы не нужны в принципе, конечно же весьма трудно. Тем не менее, внимательно посмотрев на потолок увидим там цифру — 70%. Из чего следует, что 30% школ имеют количество учеников от 500 и более, а таким школам программа, которая, не может составлять школьное расписание, а может его составить, не помешала бы. Получаем окончательную цифру — 15 тыс. школ. Вот это, пожалуй, потенциальная емкость рынка для Российской Федерации.

А что же мы имеем на сегодня в реальности? Вопрос не простой. Достоверная статистика отсутствует. Прежде всего на ум приходит одна программа, которая для всех школ РФ была им «впендюрина бесплатно». Начало разработки этой программы относится к 1998 году, а окончание (последняя версия) к 2003 году. По внешнему виду, тем более для своего времени, программа конечно же не плохая. По сравнению с другими аналогичными программами, имеет весьма логичный и продуманный интерфейс пользователя. На наше субъективное мнение, лучший пользовательский интерфейс. Однако, хотя там и имеется кнопка **Составить расписание**, программа абсолютно беспомощна в плане автоматического (без участия человека) составления расписания. Она не способна решить даже тех простых подзадач, с которыми другие программы легко справляются. Судя по отзывам в интернете, этой программой почти никто не пользуется. Так что, будем считать ее «радиационным фоном» не влияющим на общую рыночную ситуацию.

Поехали дальше. Поставим такой вопрос. Имеются ли на рынке программы которые

Третьем примером могло бы быть ПО распознавания рукописного текста (не путать с распознаванием типографского текста, это принципиально различные задачи). Многолетние усилия огромной армии исследователей пока-что увенчались практически нулевым результатом.

могут оказать завучу хоть какую-то помощь при составлении расписания? Например, многие завучи в ручную составляют расписание в два этапа. На первом этапе по их выражению: — «Разбираются с иностранцами». Другими словами, составляют расписание для преподавателей и классов при изучении иностранного языка. Вторым этапом идет все остальное. По крайней мере две программы, представленные на рынке, с этим, первым этапом, справляются на зависть прекрасно. Здесь же можно спланировать время проведения элективных курсов. При этом расставляется от 10 до 40 процентов занятий. Так что, кое-какая польза от применения компьютера, оснащенного этими программами, конечно же есть. Причем одна из этих программ весьма агрессивно и настойчиво пытается достроить расписание. В некоторых случаях, пускай и редких, это ей удается. Другая же, при достраивании расписания, абсолютно беспомощна.

Так сколько же народу на сегодня пользуется ПО для составления расписания учебных занятий в Российской Федерации? Некоторые производители такого ПО, на своих сайтах публикуют информацию о своих клиентах. Правда, к этой информации стоит относиться весьма осторожно. Как было отмечено выше, некоторые производители в «припадаках маркетинга» идут на весьма бесхитростный обман потенциальных клиентов. И тем не менее, отделив зерна от плевел получим цифру — порядка 1500 школ. Что составляет около 10% емкости потенциального рынка. Следовательно 90% потенциальных клиентов еще не окучены.

Теперь обратим свой взор на мировой рынок. Как следует из предыдущих вычислений, весьма удобным способом расчета количества потенциальных клиентов, является такой способ. Берем население страны, отбрасываем четыре нуля, получаем количество потенциальных клиентов. Так и поступим.

Европа — 500 млн. человек.

США — 300 млн. человек.

Канада — 30 млн. человек.

Япония — 125 млн. человек.

Австралия — 20 млн. человек.

Другие развитые страны — 25 млн. человек.

Вот он — «Золотой миллиард». Отбрасываем четыре нуля. Получаем — 100 тыс. потенциальных клиентов. Теперь вопрос: — «Сколько же школ из этого золотого миллиарда используют ПО для составления школьного расписания?» Применяем ту же методику, отделив зерна от плевел, что и для Российской Федерации. Получаем цифру — порядка 30-ти тыс. школ. Что составляет 30% рынка. При этом 70% открыты для агрессивного маркетинга (окучивания).

Теперь остается перевести количество в качество. То-есть, умножить количество потенциальных клиентов на цену одной лицензии ПО. Другими словами, оценить емкость мирового рынка в американских рублях. Но для этого нужно знать цену такой лицензии.

Интересно, приходилось ли читателю держать в руках толстую книжку с примерно таким названием: — «Стоимость программного обеспечения». А нам приходилось. На самом деле, формула очень проста. Программное обеспечение, какой бы сложности и объема оно ни было, стоит ровно столько, сколько за него платит клиент (пользователь). Ярчайшим примером тому, является операционная система Windows от Microsoft. Наверное мало кто задумывался о том, что по количеству труда, таланта, знаний и т.п., высадка человека на Луну, по сравнению с этой операционной системой, является детскими шалостями. И тем не менее, сто пятьдесят баксов на бочку, и ты легальный пользователь.

Проблема только в том, что количество потенциальных клиентов — пользователей операционной системы и программы для составления школьного расписания не сопо-

ставимо, ни в первом, ни во втором приближении. Отсюда вывод: — «Не смотря на то, что некоторые просят по 15 долларов за дырявые кастрюли, программа которая действительно могла бы решить большинство проблем завучей, должна стоить дорого». Остается только ответить на вопрос: — «Что такое дорого?»

Конечно же, у каждого представления о «Дорого» свои. Но наверное, для завуча (или аналогичной должности, если мы говорим о мировом рынке), дорого это его месячная зарплата. То-есть, от 1000 до 5000 долларов США. Что собственно, мы и наблюдаем, или по крайней мере ранее наблюдали, в реальности. Первое время, на мировом рынке эти программы именно столько и стоили. Падение цен, как нам представляется, произошло именно из-за того, что вдруг обнаружилось — за 5000 долларов была куплена дырявая кастрюля. Теперь, те же дырявые кастрюли, предлагают по 500 долларов США.

И наконец, перемножив количество на цену, получим примерную емкость мирового рынка ПО для составления школьного расписания — от 100 до 500 млн. долларов США. То-есть, рынок не менее денежно-емкий, чем, скажем, рынок различных систем автоматизированного проектирования в промышленности и строительстве. И к стати говоря, не менее науко-емкий.

6 «Древнеегипетский» алгоритм решения задачи

Весной 2012 года один ученый-археолог обратился к знакомым программистам со странной просьбой. С его слов при расшифровке древнеегипетских рукописей он наткнулся на описание алгоритма составления школьного расписания. Авторство алгоритма приписывалось египетской жрице по имени Ануш. Собственно его просьба заключалась в том, что бы проверить на современном компьютере действительно ли этот алгоритм способен построить школьное расписание.

По началу друзья его подняли на смех. Но внимательно ознакомившись со странными записями все же решили их проверить. Итак, приступаем к описанию идеи этого алгоритма, по сути к краткому изложению перевода древней рукописи. Предварительно скажем, что сама терминология этого алгоритма и организация древнеегипетской школы представляет отдельный исторический интерес, но поскольку данная статья предназначена не для историков, излагать алгоритм будем в современной и привычной для живущего сейчас человека терминологии.

Основное отличие древнеегипетского алгоритма (далее слово древнеегипетский будем опускать) от современных подходов заключается в том, что задача разбивается на части, или точнее, на ряд последовательно решаемых задач, при этом каждая решенная задача на предыдущим шаге является ограничением для задачи подлежащей решению на следующем шаге. В современной терминологии — применен метод декомпозиции решаемой задачи.

Следует отметить, что каждая в отдельности из задач, которые последовательно решаются в ходе алгоритма, не является NP-трудной (не решаемой). Это и позволяет с помощью последовательного решения серии легко решаемых задач, решить всю задачу составления школьного расписания целиком¹¹.

На первом шаге следует выбрать режим работы учебного заведения, а именно, определиться с тем сколько дней в неделю будет работать школа (5 или 6) и определиться с количеством проводимых за учебный день уроков (7 или 6 соответственно). Так

¹¹Слава тебе, Господи, что ты создал все нужное не трудным, а все трудное — не нужным. (Сковорода Григорий Саввич, 1722 – 1792) Сковороду Г.С. называют «первым философом Российской империи».

же нужно задать количество классов обучаемых в школе. Далее нужно поставить запрещения на те часы по которым уроки не проводятся. Это последние часы в каждом учебном дне. Для младших классов (в нашей терминологии это начиная с 5-го) таких запретов больше, для средних классов меньше, а для самых старших (11-е классы) эти запрещения вовсе отсутствуют. Что соответствует нашим санитарным нормам. Таблица запретов на проведение уроков, которая далее будет использоваться на протяжении всего алгоритма запоминается.

На втором шаге строится расписание для совместителей. Оказалось, что и древнеегипетские учебные заведения не гнушались работой совместителей. Основная особенность этой задачи заключается в том, что совместителям позволено в ультимативной форме заявлять дни по которым они будут работать. Кроме того, некоторым совместителям позволено отказаться от работы в первый урок всех рабочих дней когда они работают. Видимо этими совместителями были женщины и они не могли рано приходить в школу.

Задача решается с помощью алгоритма предписанной раскраски вершин обыкновенного графа. С этой математической моделью можно подробно познакомиться с помощью уже упоминавшейся статьи [1] или с помощью других многочисленных журнальных статей, например, [4, 11], а так же познакомившись с книгами [8, 9].

Далее для каждого занятия (класс, преподаватель, время) с помощью алгоритма решения задачи о назначении подбирается помещение для проведения этого занятия. Алгоритм решения задачи о назначении описан во множестве современных учебников, в частности с ним можно познакомиться по книге [13].

Окончанием второго шага является операция по объединению таблицы запретов на проведение уроков построенной в соответствии с санитарными ограничениями и полученным расписанием для совместителей. Таким образом, мы получаем новую таблицу запретов на проведение уроков, которая будет одним из ограничений для следующего шага алгоритма.

Третий шаг состоит из решения задачи о проведении занятий по выбору учащихся (в нашей терминологии элективных курсов). Особенностью этой задачи является то, что некоторое количество классов, в некоторый учебный час, объединяются в потоки, что бы затем в этот час разойтись по своим элективным курсам. Построение расписания будет заключаться в том, что каждому потоку будет назначено время в которое будут проводиться элективные курсы, но преподаватели будут назначаться уже после того как все расписание будет окончательно построено. То есть, на этом шаге преподаватели для проведения элективных курсов не назначаются.

При построении расписания соблюдается правило — для любого потока в один учебный день может быть назначено не более одного учебного часа для проведения элективного курса. Кроме того, соблюдается другое правило — в любой момент времени не могут быть запланированы элективные курсы более чем для одного потока. Это правило (ограничение) представляется вполне разумным, поскольку при проведении элективных курсов резко возрастает потребность в помещениях для проведения занятий. Оно введено именно с той целью, что бы не возникло ситуации когда нескольким потокам в одно и тоже время требуется большое количество свободных помещений.

Помещения для проведения элективных курсов, на этом шаге, так же как и преподаватели не подбираются, они будут подбираться вместе с преподавателями после построения всего расписания.

Алгоритмом решения задачи о проведении элективных курсов, является алгоритм предписанной раскраски верши обыкновенного графа, на который мы указывали при

описании предыдущего шага.

Новая таблица запрещений на проведения уроков строиться точно так же, как и на предыдущим шаге. Полученное расписание объединяется с таблицей запрещения.

На четвертом шаге алгоритма строиться расписание для уроков по изучению иностранного языка. Особенностью этой задачи является то, что класс может быть разделен на группы. Преподаватели не могут в ультимативном порядке заявлять о том в какие дни они будут работать. Однако для преподавателей с небольшой нагрузкой гарантирован один или два выходных, которые им будут предоставлены. Точно также как и на втором шаге алгоритма некоторые преподаватели, обучающие иностранному языку, могут потребовать освободить их от уроков в первый час рабочего дня когда они работают.

Задача о расписании преподавателей/классов по изучению иностранного языка, точно также как и на втором и третьем шаге решается с помощью алгоритма предписанной раскраски вершин обыкновенного графа.

Точно так же, как на и втором шаге с помощью алгоритма о назначении каждому занятию, а точнее, каждой группе учеников и их преподавателю подбирается помещение для его проведения.

Окончанием четвертого шага, так же как второго и третьего, является операция по объединению таблицы запрещения на проведения уроков с полученным расписанием. Таким образом, мы получаем новый вариант этой таблицы, который будем использовать на шестом шаге.

После окончания 4-го шага алгоритма, в зависимости от учебного плана школы, составляет обычно от 15% до 40% всей учебной нагрузки, предусмотренной этим планом.

На пятом шаге подсчитывается нагрузка, определенная учебным планом, на помещения которые для школы являются дефицитными. Такими помещениями, как правило, являются — спортзалы, мастерские для проведения уроков труда (технологии), кабинеты оснащенные компьютерами для проведения уроков информатики. Этот подсчет осуществляется с целью максимально возможной загрузки (минимального «простоя») таких помещений.

На шестом шаге строиться расписание для всех оставшихся предметов кроме тех, которые проводятся в дефицитных помещениях. У преподавателей нет возможности заявить ультиматум о том в какие дни они будут работать, но для тех преподавателей у которых низкая нагрузка гарантирован один или два выходных, а для части преподавателей есть возможность отказаться от работы в первый урок.

Эта задача решается с помощью алгоритма предписанной раскраски ребер двудольного мультиграфа. С идеей этого алгоритма можно познакомиться по книге [15] или по журнальным статьям [2, 3, 5, 12, 17].

Построенное расписание состоит из четверок — класс, преподаватель, предмет, время. На этом же шаге всем четверкам с помощью алгоритма решения задачи о назначении сопоставляются помещения где будут проводиться эти занятия (четверки).

После окончания этого шага вся сетка расписания заполнена, за исключением занятий проводимых в дефицитных помещениях. Однако, оставшиеся «дыры» в расписании, это и есть расписание для проведения занятий в дефицитных помещениях. Таким образом, можно считать, что на этом — шестом шаге, в некотором смысле, одновременно строятся два расписания — для обычных преподавателей/классов и для дефицитных помещений/классов.

На седьмом шаге осуществляется разделение классов на группы по предметам которые будут проводится в дефицитных помещениях. Как правило, по таким предметам как физкультура, труд (технология), информатика классы делятся на группы. Если множе-

ство преподавателей для которых построено расписание на предыдущем шаге, пересекается, с множеством преподавателей проводящих занятия в дефицитных помещениях, то тогда, формируется таблица на запрещенные часы работы преподавателей, являющихся пересечением этих множеств. С помощью алгоритма решения задачи о назначении осуществляется подбор преподавателей для каждой группы.

Последний шаг — восьмой. На этом шаге осуществляется объединение всех полученных ранее расписаний, то есть, формирование финального расписания. Для осуществления этого шага никаких алгоритмов не требуется, достаточно простых арифметических операций.

После получения финального расписания каждый преподаватель может решить для себя когда ему будет удобно проведение элективных курсов. Время для них было зарезервировано на шаге 3 алгоритма. И если этот преподаватель сможет набрать себе группу учеников, то он самостоятельно поставит свой элективный курс в расписание, вместе с подобранным им же самим помещением.

Общим правилом для всех ранее описанных шагов, кроме пятого, является правило — у каждого класса в один день не может быть более одного урока по какому либо предмету.

Кроме того, общим правилом для преподавателей является то, что каждый преподаватель может проводить занятия по нескольким предметам, в том числе и у одного класса.

7 Тестирование алгоритма

Как видно из предыдущего раздела ничего сложного для понимания в работе алгоритма построения школьного расписания нет. Одна за другой решаются, связанные между собой, отдельные легко решаемые (не NP-трудные) задачи до тех пор пока все они не будут исчерпаны. Тем не менее с уверенностью утверждать, что каждая из этих задач может быть решена, не имелось никаких оснований. В отсутствие каких-либо теоретических обоснований алгоритма проверить его работоспособность можно было только экспериментально, тем более, что именно такая задача и была поставлена ученым-археологом наткнувшимся на древнюю рукопись и сделавшим ее перевод.

Вполне естественно то, что первая мысль которая пришла в голову программистам — это создать обычное приложения для операционной системы Windows. Но что такое обычное win-приложение? Будучи активированным (запущенным на выполнение) оно ожидает поступления событий от пользователя, например, ввода исходных данных. А каким образом можно эти исходные данные получить, а позже занести в программу?

Слава богу, а точнее США, в настоящее время, мало-мальски уважающая себя школа открыла свой сайт в интернете и первое что появляется на этом сайте, не считая фотографий с различных праздничных мероприятий, так это учебный план школы. Остается только его скопировать и ввести в программу в качестве исходных данных для расчета расписания. Вопрос. Сколько нужно для этого времени? Практика использования ныне предлагаемых рынком программ составления школьного расписания показала, что для ввода учебного плана вместе с формированием таблицы распределения педагогической нагрузки требуется от 8-ми до 10-ти часов, мягко говоря кропотливого труда. Предположим, что этот учебный план введен, а таблица распределения педагогической нагрузки сформирована, и о чудо... расписание построено. О чем это говорит. Ровным счетом ни о чем. Ни-какой гарантии того, что следующая задача будет решена нет. Вот если бы расписание не было построено, то это говорило бы о многом, а именно, о том, что

алгоритм не решает поставленной задачи. Другими словами, обычное win-приложение, в некотором смысле, практически не поддается тестированию. Как же быть?

Опять же — слава богу, а точнее слава Microsoft, в современных версиях операционной системы Windows поддерживается так называемый режим консольного приложения. Кстати говоря, для некоторых молодых людей это является полным откровением, им никогда не доводилось видеть черных окошек с бегущими внутри этих окошек строчками текста. Действительно, это стиль больших ЭВМ из далекого прошлого и давно покинувшей сцену — MS-DOS. Но у этих окошек есть одно преимущество. Они могут висеть на экране компьютера, производя нужные вычисления, без какого либо участия человека и день, и месяц, и. . . не берусь утверждать сколько. Именно это и требовалось для проверки алгоритма.

Далее ход рассуждений был следующий. Написание генератора исходных данных (грубо говоря, учебного плана типовой школы и таблицы распределения педагогической нагрузки) конечно же займет некоторое время, но зато, будучи один раз написанным он позволит получить не ограниченное количество тестовых заданий для проверки алгоритма, достаточно будет только лишь после решения очередной задачи передать управление этому генератору для построения нового (следующего) задания. Появится возможность получить статистически достоверные данные о качестве тестируемого алгоритма. Например, 80 процентов задач решено, а 20 нет, или наоборот. Нужно только лишь, число решаемых задач сделать достаточно большим. Именно так и предстояло поступить — консольное приложение, таким виделся выход из создавшейся ситуации.

Как говорить, быстро сказка сказывается, да не быстро дело делается. Придумать генератор исходных данных, адекватно отражающий все практические ситуации, пусть даже и типовой школы, оказалось не таким простым занятием. Но однажды сбылись мечты сумашедшие. . . , рано или поздно. . . , сколько веревочка не вейся. . . Генератор исходных данных закончен, древнеегипетский алгоритм запрограммирован, «все ошибки исправлены»,¹² ловушки на ошибки расставлены, проверки результатов вычислений установлены.

В начале программе предлагалось для составления расписания небольшое количество классов — от 9 до 14 (маленькая школа). Решения выскакивали как из пулемета. При увеличении количества классов — от 15 до 21 (средняя школа) решения выстреливались быстро, но уже не как из пулемета. . . скорее как из пистолета. Далее. Вот она. . . большая школа, до четырех классов в параллеле, общее количество классов от 22 до 28. Явно включились тормоза. . . Процесс стал напоминать ленивую утку переваливающуюся с ноги на ногу. Но одно радовало — строчка: «Количество не решенных задач = » постоянно показывала ноль.

Стало ясно. Для получения статистически достоверных данных, подтверждающих возможность решения любой разумной задачи в полностью автоматическом режиме, одного компьютера не достаточно. Небольшие арифметические расчеты показывали — что бы оперировать числами от шести и более знаков о количестве решенных задач, требуется не менее десятка компьютеров. А для десятка компьютеров (можно прикинуть количество выделяющегося от этих компьютеров тепла и издаваемый постоянный шум от вентиляторов) требуется отдельная комната. Но ничего, нас не остановишь. . . Десяток не десяток, а семь четырех-ядерных компьютеров вскоре были введены в эксплуатацию.

В результате по прошествии года «насильственных действий» древнеегипетского алгоритма по отношению к достопочтенной четырех-ядерной семерки, и после десятков миллионов решенных задач, можно с уверенностью утверждать: — «Любая, без исклю-

¹²Ха, ха. . .

чений, разумная, корректно заданная задача¹³ по расчету школьного расписания, может быть решена без участия человека в полностью автоматическом режиме». При этом, суммарное время расчета 1000 задач примерно следующее: для группы задач от 9 до 14 классов = 20 минут, для группы задач от 15 до 21 класса = 40 минут, для группы задач от 22 до 28 классов время расчета составляет от 6 до 8 часов, т.е. для этой группы в среднем, примерно, по пол минуты на одну задачу.

Таким образом, более чем годовой эксперимент, по проверке (тестированию) алгоритма составления школьного расписания в полностью автоматическом режиме, без участия человека, за который было решено десятки миллионов тестовых заданий, был с успехом завершен. Практически для всех тестовых заданий (исходных данных) было полностью построено расписание, удовлетворяющее всем ограничениям.

8 Логическая модель будущего ПО

После завершения годового тестирования алгоритма составления школьного расписания встал вопрос: — «И что дальше?».

Прежде всего, бросается в глаза то, что консольное приложение, никого не сможет убедить в том, что действительно решается задача составления школьного расписания. . . разве что, самого программиста который написал это приложение. Создать черное окошко, с время от времени появляющимися там строчками типа: — «Количество решенных задач = 12547564» подсилу даже плохоуспевающему пятикласснику. Таким образом, нормальный человек, просто не поверит такой, с позволения сказать, программе, и правильно сделает. Обойтись без полноценного win-приложения, никак не получится.

Но, сначала не плохо было бы, определиться с целями создания такого приложения. По крайней мере, на виду две таких цели. Это — создание полноценного ПО со всеми вытекающими из этого последствиями, и — создание приложения демонстрирующего работу алгоритма, которое хуже или лучше сможет убедить человека в том, что его не обманывают. И ежику понятно, что по трудоемкости эти два проекта просто не сопоставимы. Вполне естественно, что было принято решение пойти по легкому пути.

Хорошо: — «Что требуется от такого win-приложения — демонстрации?». Прежде даже можно поставить другой вопрос: — «Какое оно должно быть?».

Во-первых. Сразу же снимается головная боль об удобном, понятном, практичном и красивом пользовательском интерфейсе. Для такой демонстрашки вполне достаточно примитивнейшего интерфейса. Важно лишь то, что бы пользователю были видны те исходные данные которые предлагаются программе для расчета (сгенерированные естественно случайным образом) и результаты этого расчета. По крайней мере, теоретически у пользователя будет возможность проверить соответствие исходных данных и полученного с помощью программы результата. Сложна ли такая проверка? . . . Ответ однозначный: — «Да уж не проста. . . ». Особенно если знать какое количество ловушек и проверок содержится в консольном приложении для постоянной верификации получаемых результатов, а так же размер кода этих проверок и ловушек. Есть ли другие способы убеждения? . . . Разве что, передача всем интересующимся. . . исходных текстов программы. Но вот, например, в Microsoft это не принято¹⁴.

¹³Что такое разумная задача, более менее понятно, а вот что такое корректно заданная задача, вопрос очень и очень не простой. В первом приближении можно считать задачу корректно заданной, если ее решение существует. А вот вопрос: — «При каких условиях решение задачи существует?» к сожалению на сегодня остается без ответа.

¹⁴Впрочем, передача кода генератора исходных данных и кода проверки правильности составленного

Во-вторых. Снимается проблема файла помощи, руководства пользователя, и прочих бантиков и прибабасов совершенно необходимых для полноценного программного обеспечения.

Так и сделали¹⁵. На главную форму приложения наклеили двадцать с лишним кнопок, из которых на каждом этапе расчета активна только одна, не считая кнопок типа — *О программе*, *Начать новую задачу*, *Закрой меня*. Нажимаешь на такую кнопку, появляется окошко с кнопкой *Генерация данных*. Нажимаешь *Генерация данных*, в окошке на белом фоне появляются построенные данные. Закрываем окошко. Кнопка на которую только что нажали, тухнет (перестает быть активной), следующая на которую следует нажать становится активной. Жмем. Открывается следующее окошко. А там кнопка *Построить расписание*. Нажимаем на *Построить расписание*, появляется построенное расписание. Желаящий может проверить — правильно построено расписание или нет. И так до тех пор, пока не будут пройдены все шаги алгоритма. А дальше можно нажать на большую кнопку *Начать новую задачу*. И так по кругу. Или нажать кнопку *Закрой меня*.

На первый взгляд может показаться: — «Вся эта демонстрационная программа — мартышкин труд». Но это не так. По крайней мере по трем причинам.

Во-первых. В ходе разработки демонстрации была решена довольно важная задача разработки будущей архитектуры полноценного ПО. А именно. Требовалось жесточайшим образом отделить «мозги» от «туловища». Если выразаться более понятно, отделить код алгоритма расчета расписания от кода генератора исходных данных и кода пользовательского интерфейса. Весь код алгоритма расчета расписания сосредоточен в динамически подключаемой библиотеке, таким образом, пользовательский интерфейс, как клиент, может обращаться с заданиями к динамической библиотеке, которая выступает в качестве сервера, на построение различных расписаний, составляемых на различных шагах алгоритма. Это позволит в будущем, не трогая кода алгоритма расчета расписания, проводить эксперименты с различными вариантами интерфейса до полного и окончательного удовлетворения пользователей.

Во-вторых. Не смотря на свою примитивность, демонстрационный пользовательский интерфейс представляет из себя логическую модель будущего удобного, понятного, практичного и красивого интерфейса пользователя. Так например, в нем реализована возможность возврата к предыдущему шагу алгоритма, а эта возможность, в свою очередь повлияла на структуру данных программы. Кроме того, демонстрационный интерфейс поддерживает такую особенность алгоритма, как продвижение от шага к шагу в строгой последовательности, что обеспечивает целостность данных и их защиту от неверных изменений.

В-третьих. Опять же, повторимся, не смотря на свою примитивность, имеющийся пользовательский интерфейс пригоден для анализа математической модели практических ситуаций, которые возникают при составлении школьного расписания, принятой в данной программе. Такой анализ или экспертизу могли бы осуществить специалисты хорошо знакомые с темой, например, завучи с достаточным опытом работы, преподающие в школе математику. Разобраться в деталях расчета, конечно же их квалификации не достаточно (да и не у кого не возникнет такого желания), зато в силу полученной общей математической культуры, разглядеть явные упущения в постановке задачи они могут гораздо лучше любого профессионального математика, который знаком с работой школы только по наслышке или по различного рода публикациям.

расписания вполне возможна, поскольку этот код никакой коммерческой ценности не представляет.

¹⁵В честь древнеегипетской жрицы Ануш программу, на русский манер, назвали — Аннушка.

«И что же дальше?». А дальше разработка полноценного ПО по всем законам и правилам программной инженерии, которое теперь уже, по сложности, не превосходит обычное ПО для ERP-систем.

Вот только не спрашивайте: — «Сколько для этого потребуется времени и какова трудоемкость разработки такого ПО?...». И тем более не спрашивайте: — «Во сколько обойдется такая разработка?...».

9 Проблемы с бизнес-моделью

Как ранее было подсчитано, общемировой рынок ПО составления школьного расписания, в полностью автоматическом режиме, составляет от 100 млн. до 500 млн. долларов США.¹⁶ Однако, этот рынок, как выражаются венчурные инвесторы, еще нужно «Поднять». И здесь, совершенно отчетливо вырисовываются, по меньшей мере, две проблемы. Одна проблема, это: — «Дорого». На ней мы уже останавливались. И другая, на наш взгляд более серьезная, это: — «Репутация подобного ПО».

Если прибегнуть к метафоре, то репутация подобного ПО напоминает загаженную, сильно унавоженную и дымящуюся, как после битвы на Куликовом поле, мусорную свалку. Причем дым настолько едкий, что хочется закрыть глаза и прекратить дыхание. Как уже упоминалось ранее, при разговоре с потенциальными клиентами ПО составления школьного расписания, этот разговор с легкостью переходит на матершину. «Достали. . . своей автоматизацией, информационным пространством школы и электронными дневниками, дайте спокойно работать. . .».

Что можно сделать для изменения репутации подобного ПО и отношения к нему завучей с враждебного, до хотя бы, нейтрального? Про положительный образ мы пока не заикаемся.

Лет десять назад, еще можно было говорить о том, что компьютеры в кабинетах завучей стоят для мебели, как непрременный аксессуар учености и прогрессивности. Что в лучшем случае, компьютер используется вместо пишущей машинки (хотя, как было отмечено ранее именно это обстоятельство и послужило такому бурному расцвету индустрии персональных компьютеров). В настоящее время обстановка изменилась. Уже многие попробовали. . . Результаты таких проб мы только что обсуждали.

Остается все начать с начала. А именно. С бизнес-модели распространения подобных программ. Даже не очень внимательно приглядевшись, можно увидеть, что за последние 15 лет эта бизнес-модель практически не изменилась. Найдите сайт программы, скачайте демонстрационную версию, выпишите счет на оплату. . .

Со счетом на оплату вроде все понятно. Без сайта программы, тоже никак не обойтись. А вот что с демо-версиями?

А с демо-версиями все по разному. **Вариант первый.** Наша демо-версия ничем не отличается от рабочего варианта программы, вот только нельзя сохранять введенные данные, и нельзя выводить на принтер полученные результаты. А так, все работает. Можно ли с помощью такой демо-версии оценить все достоинства и недостатки программы?

16

И даже. . . может быть. . . Но что! мечта пустая.

Не будет этого никак.

Судьба завистливая, злая!

Ах, отчего я не табак! . . .

А. С. Пушкин

Как уже было отмечено ранее, для ввода всех исходных данных, что бы там не пискляво гундела реклама про час, максимум полтора, реально требуется минимум 8 – 10 часов непрерывной и кропотливой (до чертиков скучной) работы. Нормальный человек, а тем более пользователь впервые приступающий к работе с программой, когда ему нужно одновременно и с программой учиться работать и аккуратно, без ошибок вводить гору исходных данных, за один раз этого сделать не сможет. Требуется как минимум два, а то и три дня (раза). Теперь вообразите страх начинающего перед тем, что сейчас непременно отключится питание или чтонибудь перезагрузится. Ну ... не возникнет у нормального человека желания воспользоваться такой демо-версией. Так что, либо решиться на покупку «кота в мешке», зная о «припадках маркетинга» некоторых разработчиков, либо, что чаще всего и бывает, с горечью за бесцельно потраченное время нажать на клавишу Del.

Справедливости ради нужно отметить, что те же разработчики придумали другой вариант. Сделали для своей программы «ломалку». Ничего не подозревающий, добродушный пользователь, предварительно отключив небольшим ключиком свою совесть, скачивает нелегальную копию (дема + ломалка). Устанавливает, ломает, и ... все работает ... Как говорится, пользуйтесь на здоровье ... Правда, примерно через пол года, программа вам объявит, что переходит в демо-режим, а для сохранения своих данных, будьте так любезны ..., обратитесь к разработчику за выпиской счета ...

Глядя со стороны на такие уловки, этот вариант представляется — в конце концов, более честным. Хотя конечно, пользователь пытается обмануть производителя, производитель обманывает пользователя ..., кстати говоря обещая ему, что через несколько минут после ввода всех исходных данных он получит готовое расписание. С уверенностью можно утверждать, что подавляющее большинство пользователей, так никогда и не узнают о том, что их данные подвергались реальной угрозе. Потратив 15 – 20 часов на работу с программой и убедившись в ее бесполезности, с криком: — «Все программы, как и мужики, такие ...», потенциальные покупатели гневно удаляют эту программу со своего компьютера. А через час – полтора, успокоившись, отдышавшись, говорят себе: — «Какая же я ... все-таки умная, что не заплатила денег за это ..., говорила мне мама — "Не бери кота в мешке"».

Вариант второй. Наша демо-версия ничем не отличается от рабочего варианта, ограничение только одно, максимальное количество классов — пять. А так, все работает. В результате на форуме появляется такое заявление. «Видал я вапу, с позволения сказать программу. И ввел то, всего ничего — четыре класса. А она мне: — "Не могу составить расписание". Можете засунуть ее себе в ... Спекулянты проклятые.»

Здесь мы сталкиваемся со случаем когда разработчики, нашли на свою «... (голову)» приключений. Те кто думают, что составить расписание для школы с четырьмя классами намного легче чем, например, с двадцатью, глубоко заблуждаются. Именно по этому при тестирования «Древнеегипетского» алгоритма составления расписания, было принято решение — при генерации тестовых данных, за минимум количества классов, выбрать число — девять. Это объясняется, порой, невозможностью в автоматическом режиме составить таблицу распределения педагогической нагрузки. Проще говоря, распределить нагрузку между мизерным количеством классов и соответственно, мизерным количеством учителей. По всей видимости, такие фокусы может показывать только лишь очень опытная рука (или глаз, если угодно) человека.

Вариант третий. Ну хорошо. Пользуйтесь нашей программой. Но, две недели. А через две недели все, шабаш. «Воду отключим ...». Можно ли за две недели освоить программу и оценить ее все достоинства и недостатки? Положа руку на сердце ска-

жем: — «Пожалуй, что можно . . . ». Но при одном условии. Нужно перестать заниматься всем остальным. А любимое слово завучей: — «Занята». «Ой, занята. Так занята, что ни вдохнуть, ни . . . некогда». Бросит ли завуч на две недели все на свете и погрузиться ли на этот срок в программу для составления расписания? Как говорят ученые: — «Сказать трудно . . . ».

Короче говоря, все плохо . . . И так плохо, и эдак не удобно . . .

Где же искать выход? Может быть . . . аренда?

10 Бизнес-модель SaaS использования ПО

Изначально вся компьютерная отрасль использовала арендную бизнес-модель — первые компьютеры стоили огромных денег и их вычислительные мощности сдавались заказчикам. С появлением Интернета старая бизнес-модель была реанимирована, но на принципиально иной технологической базе. **SaaS** (англ. *software as a service* — программное обеспечение как услуга) — бизнес-модель продажи и использования программного обеспечения, при которой поставщик разрабатывает веб-приложение и самостоятельно управляет им, предоставляя заказчику доступ к программному обеспечению через Интернет.

Главным отличием SaaS от старой модели является то, что раньше заказчики получали доступ к компьютерам напрямую, а не с помощью глобальных сетей. Поскольку модель SaaS ориентирована на предоставление услуг с помощью Интернета, ее развитие непосредственно связано с развитием глобальной сети. Первые компании предлагавшие программное обеспечение как услугу, появились в западных странах в 1997 – 1999 годах, а акроним SaaS вошел в широкое употребление в 2001 году.

Представляется, что в нашем «тяжелом случае», эта бизнес-модель является наиболее оптимальной, а может быть даже, единственно приемлемой. Она избавит потенциальных клиентов рисковать относительно крупной суммой денег при оплате программного продукта из группы продуктов с почти безнадежно испорченной репутацией. Используя арендную бизнес-модель заказчик сможет спокойно и постепенно убедиться в том, что предлагаемый продукт, ему действительно нужен, и что его ожидания от использования продукта совпадают с тем, что он в действительности получает. Об ожиданиях завучей от подобного рода программ, мы ранее говорили достаточно подробно.

11 Вместо заключения

Иногда, некоторые с ехидцей спрашивают: — «А бизнес-план у вас есть? . . . » Есть. И при том, очень простой. «Последовательно решать возникающие проблемы по мере их поступления. . . ». В крайнем случае, можно будет воспользоваться моделью SaaS (бизнес-план — по требованию). Если комунибудь понадобится, можно будет все подробно и в деталях спланировать, ни один бухгалтер не придерется!

Другие с не меньшей ехидцей спрашивают: — «А экспертизу вы своего бизнес-проекта делали? . . . » Нет не делали, и сейчас объясним почему.

Прежде всего нужно определиться с тем, какие вопросы могут быть заданы экспертам. Здесь довольно четко прослеживается два направления. Это во-первых, вопросы связанные с емкостью рынка и его состоянием, и во-вторых вопросы связанные с достоверностью декларируемых результатов.

Что касается рынка, то вопросы могли бы быть сформулированы следующим образом:

1. Верна ли методика оценки емкости рынка ПО составления школьного расписания?

2. Действительно ли, имеющиеся в настоящее время на рынке программы, совершенно беспомощны при решении задачи автоматического построения школьного расписания без участия человека?

На первый вопрос могли бы ответить специалисты по венчурным инвестициям. На второй вопрос, мог бы дать ответ любой более, менее грамотный пользователь компьютера, но лишь при том условии, что этот пользователь посвятит не менее 15 – 30 часов работы по изучению каждой программы составления школьного расписания. При этом количество проверенных программ должно быть, хотя бы, не менее трех, наиболее успешных и представительных. Другими словами, это человек должен посвятить от 50 до 100 часов своего личного времени изучению этого ПО. Или переводя в рабочие дни он должен заниматься этим около месяца.

Остается теперь ответить на вопрос: — «Где взять человека, который согласится в течении месяца заниматься сизифовым трудом?»

Теперь, что касается декларируемых результатов. Вопросы могли бы быть сформулированы следующим образом:

1. Известны ли Вам научные работы (реклама коммерческих продуктов исключается) декларирующие решение задачи о составлении школьного расписания в полностью автоматическом режиме? Другими словами, гарантирующие решение задачи о составлении школьного расписания, при наличии всех необходимых исходных данных для ее решения.
2. Какова по Вашему мнению может быть степень доверия к тексту, изложенному в разделе 6 и 7 экспертируемой статьи? В частности:
 - Может ли задача о предписанной раскраске вершин обыкновенного графа служить математической моделью задачи о составлении расписания для совместителей, работающих в школе, для составления элективных курсов, для составления расписания учителей/классов, изучающих иностранный язык?
 - Может ли задача о предписанной раскраске ребер двудольного мультиграфа служить математической моделью задачи о достраивании школьного расписания?
 - Может ли задача о наилучшем назначении являться математической моделью задачи о распределении занятий (занятие — четверка: класс, учитель, предмет, время) по школьным помещениям?
3. Какова по Вашему мнению сложность задачи о составлении школьного расписания в полностью автоматическом режиме? В частности, задачи о предписанной раскраске вершин обыкновенного графа, задачи о предписанной раскраске ребер двудольного мультиграфа (сложность задачи о наилучшем назначении хорошо известна любому студенту — это легко решаемая задача).

Далее возникает вопрос: — «Кто бы мог дать ответы на эти вопросы?» Совершенно очевидно, что этот человек должен быть достаточно искушенным в математике (по классификации ВАК — «Дискретная математика и математическая кибернетика», именно по этой специальности защищаются диссертации по теории расписаний). Таких специалистов в стране не много, и будут ли они заниматься подобной экспертизой из любви к науке, совершенно не ясно . . .

Список литературы

- [1] Балтак С.В., Сотсков Ю.Н. Построение расписания учебных занятий на основе раскраски вершин графа // Информатика, 2006, № 3, с. 58 – 69.
- [2] Бородин О.В. Раскраски и топологические представления графов // Дискретный анализ и исследование операций. 1996, Том 3, № 4, с. 3 – 27.
- [3] Бородин О.В. Обобщение теоремы Коцига и предписанная раскраска ребер плоских графов // Математические заметки. 1990, Том 48, Выпуск 6, с. 22 – 28.
- [4] Визинг В.Г. Раскраска вершин графа при мажоритарных ограничениях на используемые цвета // Дискретный анализ и исследование операций. 2009, Том 16, № 4, с. 21 – 30.
- [5] Визинг В.Г. О связной раскраске графов в предписанные цвета // Дискретный анализ и исследование операций. 1999, Серия 1, Том 6, № 4, с. 36 – 43.
- [6] Гафаров Е.Р., Лазарев А.А. Математические методы оптимизации при составлении учебного расписания // Новые информационные технологии в образовании. Сборник научных трудов. — М.: 1С-Публишинг, 2013, Часть 2, с. 51 – 55.
- [7] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.
- [8] Дистель Р. Теория графов: Пер. с англ. — Новосибирск: Изд-во Ин-та математики, 2002. — 336 с.
- [9] Емеличев В.А., Мельников А.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. — М.: Наука. Гл. ред. физ.-мат. лит., 1990. — 384 с.
- [10] Ичбана Д., Кнеппер С. Бил Гейтс и сотворение Microsoft. — Ростов-на-Дону: Издательство «Феникс», 1997. — 352 с.
- [11] Карпов Д.В. Динамические правильные раскраски вершин графа. // Записки научных семинаров ПОМИ. 2010, Том 381, с. 47 – 77.
- [12] Магомедов А.М., Магомедов Т.А. Интервальная на одной доле правильная реберная 5-раскраска двудольного графа // Прикладная дискретная математика. 2011. №3(13), с. 85 – 91.
- [13] Пападимитру Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. Пер. с англ. — М.: Мир, 1985. — 512 с.
- [14] Романовский И.В. Дискретный анализ. Учебное пособие для студентов, специализирующихся по прикладной математике и информатике. — Издание 2-е, исправленное. — СПб.: Невский диалект, 2000. — 240 с.
- [15] Свами М., Тхуласираман К. Графы, сети и алгоритмы: Пер. с англ. — М.: Мир, 1984. — 455 с.

- [16] Смирнов В.В. Перербургские школы и школьные здания. История школьного строительства в Санкт-Петербурге — Петрограде — Ленинграде 1703 — 2003 гг. — СПб.: Издательство «Русско-Балтийский информационный центр "БЛИЦ"», 2003. — 144 с.
- [17] Стеценко О.П. Об одном виде раскраски ребер графа в предписанные цвета // Дискретная математика. 1997. Том 9, выпуск 4, 92 — 93.
- [18] Урнов В.А. Расписание — наиболее востребованный АРМ в образовании // Информатика и образование. 2001, № 4, с. 47 — 52.
- [19] Харари Ф. Теория графов. — М.: Мир, 1973. — 302 с.
- [20] Even S., Itai A., Shamir A. On the complexity of timetable and multicommodity flow problems // SIAM J: Comput. Vol. 5, No. 4, December 1976, 691-703